

Component, Context, and Manufacturing Model Library 1 (C2M2L-1) – Technical Area 2

Final Report

28 February 2013

Reporting Period: 11/1/2011 – 5/1/2013

Contract Number: D12PC00241

Prepared For:

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Prepared by:

BAE Systems Land & Armaments L.P.
4800 East River Road
Minneapolis, MN 55421-1498

Distribution Statement C: Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document shall be referred to the DARPA Technical Information Office

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 28-02-2013	2. REPORT TYPE Final Report		3. DATES COVERED (From - To) November 2011 - February 2013		
4. TITLE AND SUBTITLE Component, Context, and Manufacturing model Library 1 (C2M2L-1), Technical Area 2 Final Report			5a. CONTRACT NUMBER D12PC00241		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Challou, Daniel; Pukite, Paul; St. Louis; Polston, Roger; Ames, Jeremy; Wentland, Christopher			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) BAE Systems, Land & Armaments 4800 East River Road Minneapolis, MN 55421-1498				8. PERFORMING ORGANIZATION REPORT NUMBER 0001002059	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Project Agency 3701 North Fairfax Drive Arlington, VA 22203-1714				10. SPONSOR/MONITOR'S ACRONYM(S) DARPA	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Distribution Statement C: Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document shall be referred to the DARPA Technical Information Office					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The role of environmental context models in the Adaptive Vehicle Make program is to help verify that the design solution (i.e., model of an amphibious infantry fighting vehicle) produced by design teams participating in the FANG challenges is capable of meeting or exceeding the mobility and drivetrain related requirements as specified by DARPA's AVM program management team. The BAE System's team employed an iterative, requirements-driven approach to develop the environmental context models, examples of their use, and web-based repository. Our extensive knowledge in the area of military ground vehicle development and testing and the unclassified documentation describing the standard tests and procedures used by the United States Army Test and Evaluation Command (ATEC) were the foundation for capturing a cohesive collection of context models and examples.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF: a. REPORT U		17. LIMITATION OF ABSTRACT SAR		18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Stephen F. Panciocco 19b. TELEPHONE NUMBER (Include area code) 763-572-6124

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39-18

Table of Contents

1.	Introduction and Overview.....	1
2.	Creating and Using Context Models	2
2.1	Role of Context Models in AVM.....	3
2.2	Deriving the Requirements for Context Models	5
2.2.1	Bottom-up derivation of Requirements for Environmental Context Models.....	5
2.2.2	Top-Down Derivation of Requirements for Environmental Context Models.....	9
2.3	Creating Context Models	9
2.3.1	Discussion	10
2.3.2	Technical Challenges	11
2.3.3	Technical Results	11
2.3.4	Findings and Conclusions	12
2.3.5	Implications for Further Research.....	12
2.4	Applying (Representing/Storing and Using) Context Models.....	13
2.4.1	Technical Challenges	13
2.4.2	Context Model Scope and Representations	13
2.4.3	Technical Results	14
2.4.4	Findings and Conclusions	17
2.4.5	Implications for Further Research.....	17
2.5	Examples of Context Model Use in Design Verification.....	17
2.6	Storing and Accessing Context Models and Examples.....	19
2.6.1	Technical Challenges	20
2.6.2	Overview of Development Methodology	20
2.6.3	Technical Results	22
2.6.4	Findings and Conclusions	22
2.6.5	Implications for Further Research.....	23
3.	Results	23
3.1	Technical Challenges.....	23
3.2	Discussion	24
3.3	Technical Results	24
4.	Conclusion	27
4.1	Avenues for Further Research.....	28
5.	References	29

List of Figures

Figure 1: Overview of the Scope of our Effort for Environmental Context Modeling	1
Figure 2: Context Models Support the META Verification Process. Interactions illustrate the use of the context model library and environmental context models in the verification process.	3
Figure 3: Description of a simulation for computing 6-watt power for a model of a road wheel of an Amphibious Infantry Fighting Vehicle (e.g., a Quarter-Car), and the interaction between the model of the design solution (the road wheel) and the environment (terrain derived from a stochastic representation).	4
Figure 4: Land Use-Cases mapped to required contexts	7
Figure 5: Aquatic Use-Cases mapped to required contexts	7
Figure 6: Atmospheric Environmental Context Models required to carry out mobility testing on an IFV.	8
Figure 7: Land Environmental Context Models required to carry out mobility testing on an IFV.....	8
Figure 8: Aquatic Environmental Context Models required to carry out mobility testing on an IFV.....	9
Figure 9: Outline of Process for Creating Models	10
Figure 10: Integration of stochastic modeling with the semantic web	12
Figure 11: State-space for model-based verification	13
Figure 12: Ontological organization of Context Models.....	14
Figure 13: Terrain representation at a gross topographic resolution, suitable for long range analysis, and a fine resolution suitable for vehicle suspension analysis.....	16
Figure 14: Usage of context model in a vehicle model checker	17
Figure 15: Track Builder: A Unity-based Application for Validating Context and Vehicle Concept Models Performance in Various Land and Sea Contexts. More detail on the application and example can be found in Appendix L.....	18
Figure 16: Extension to Track Builder for Validating Vehicle Concept Performance in Surf Zones. More detail on the application and example can be found in Appendix L	19
Figure 17: Filter Example to Demonstrate interaction between Atmospheric Contexts and Vehicle Subsystems. More details on the example can be found in Appendix O	19
Figure 18: Ontological System for Context Artifacts and Resources (OSCAR) Architectural Overview. OSCAR is described in Detail in Appendix E	20
Figure 19: Dynamic Context Server Home Page. The Dynamic Context Server is described in detail in Appendix D.....	21
Figure 20: Overview of the Context Modeling Effort in relation to the Appendices	27

List of Tables

Table 1: Brief Description of each of the steps in our Environmental Context Modeling Effort depicted in Figure 1	2
Table 2: Data Sources used to extrapolate requirements for Vehicle Test Cases (i.e., “Use Cases”), and environmental context models	5

List of Symbols, Abbreviations, and Acronyms

Symbol, Abbreviation, Acronym	Definition
AVM	Adaptive Vehicle Make
C2M2L	Component, Context, and Manufacturing Model Library
CAD	Computer-Aided Design
CML	Component Model Library
CSIR	Council for Scientific and Industrial Research
DARPA	Defense Advanced Research Project Agency
DCS	Dynamic Context Server
FANG	Fast, Adaptive, Next-Generation
FMI / FMU	Functional Mockup Interface / Functional Mockup Unit
IFV	Infantry Fighting Vehicle
ITAR	International Traffic in Arms Regulations
JPL	Jet Propulsion Laboratory
MBE	Model Based Engineering
NRMM	NATO Reference Mobility Model
OSCAR	Ontological System for Context Artifacts and Resources
PCC	Probabilistic Certificate of Correctness
PDF	Probability Density Function
PoC	Probability of Correctness
PSD	Power Spectral Density
SWEET	Semantic Web for Earth and Environmental terminology
TA	Technical Area
TOPS	Test Operating Procedures

1. Introduction and Overview

The goal of the Defense Advanced Research Project Agency's (DARPA) Adaptive Vehicle Make (AVM) program is to develop a radically innovative approach to the development of military vehicles, resulting in significant reductions in the cost and schedule required to develop such vehicles. To achieve that goal, a novel approach to the model-based verification of vehicles is required. Thus, in response to DARPA's call to construct a Component, Context, and Manufacturing Model Library (C2M2L) to help achieve their goal, we developed a comprehensive suite of environmental context models necessary to verify the model-based mobility and drivetrain systems of a land-based or an amphibious Infantry Fighting Vehicle (IFV).

More specifically, the objectives of our efforts for the Component, Context, and Manufacturing Model Library (C2M2L-1) were as follows:

1. To develop a comprehensive suite of environmental context models that can be utilized to thoroughly evaluate model-based representations of the mobility and drivetrain systems of land-based or amphibious Infantry Fighting Vehicles for their ability to meet or exceed their performance requirements.
2. To ensure that the models we developed would be correctly and effectively used for their analysis tasks, we created examples of how to use them for requirements verification, and made the models, examples of their use, and information about the models accessible (searchable and retrievable) via a web-based Ontological System for Context Artifacts and Resources (OSCAR). We also formulated and delivered a spreadsheet detailing model and example locations in a Subversion repository as an alternate means of locating and utilizing many of the models and examples.

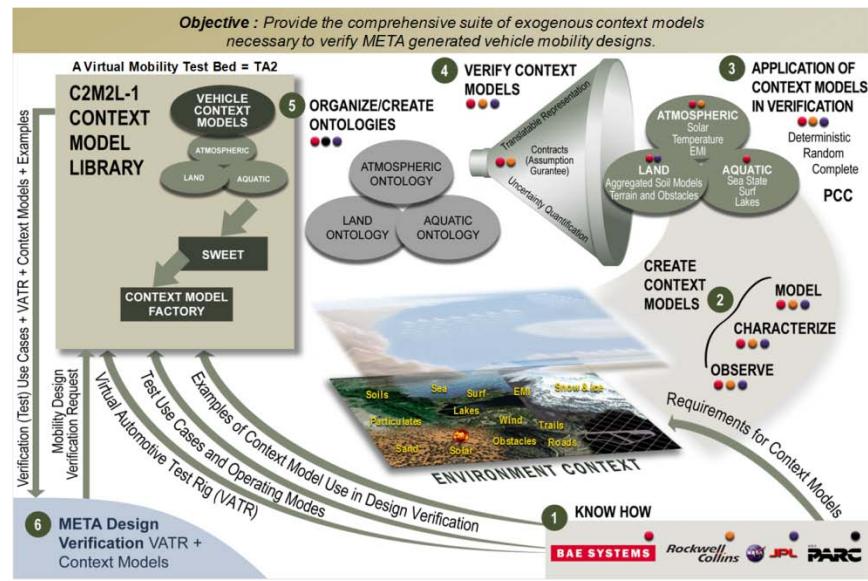


Figure 1: Overview of the Scope of our Effort for Environmental Context Modeling

Figure 1 outlines the steps we followed in order to meet our objectives. **Table 1** presents a brief overview of each of the six steps outlined in **Figure 1**.

Table 1: Brief description of each step in our Environmental Context Modeling flow depicted in **Figure 1**

1.	Derived the set of use cases/mobility verification requirements for testing land-based and amphibious mobility systems and drivetrains. Derived the requirements for environmental context models from the use-cases / mobility verification requirements.
2.	Created the set of environmental context models necessary to thoroughly test the mobility and drivetrain system and subsystem designs constructed by META toolset from their requirements using novel and adaptable algorithms and methods
3.	Synthesized novel (stochastic) and other representations to facilitate the application of Context Models for different classes of verification.
4.	Created executable examples to verify the context models and to illustrate the use of each of the context models in design verification scenarios.
5.	Created a semantic web-based executable architecture based on the Semantic Web Earth and Environmental (SWEET) ontology populated with translatable representations of the context models, information used to construct the models, examples of model use, and methods for translating the models
6.	Created a template of a surrogate mobility and drivetrain system (A Virtual Automotive Test Rig (VATR)) for use with context models and utilized it, along with the context models, in the executable examples we formulated.

Table 1 briefly describes each of the steps we carried out over the course of our context modeling effort. The following sections describe our approach/methodology, challenges, results, and findings as applicable to each of the steps in the table.

Section 2 provides a more detailed discussion on creating and using context models (Steps 1-6 illustrated in **Figure 1** and described in **Table 1**);

Section 3 discusses the technical challenges of the overall modeling effort and provides an overview and pointer to the detailed results documented in the appendices.

Section 4, reviews our notable developments over the course of this project in the areas of model development, examples of context model use, and ontologically-based web services.

Finally, accompanying this report is a series of appendices that provide greater technical detail on the results of our efforts. The appendices are listed and summarized in Section 3.3, Technical Results, on page 24.

2. Creating and Using Context Models

We employed an iterative, requirements-driven approach to develop the environmental context models, examples of their use, and web-based repository. We determined what environmental models would be required for evaluating mobility and drivetrain designs by drawing on BAE's extensive knowledge in the area of military ground vehicle development and testing; by reviewing unclassified documentation describing the standard tests and procedures used by the United States Army Test and Evaluation Command (ATEC) to evaluate the ability of military vehicles to meet their performance requirements, and by reviewing the requirements for the Fast, Adaptive, Next-Generation (FANG) infantry fighting vehicle (IFV) supplied by DARPA's Adaptive Vehicle Make (AVM) program leadership. Once we had the requirements, our task was fairly straightforward. Collectively, the requirements directly or indirectly

specify what environments vehicles will be evaluated in, and they also specify the use-cases for constructing the examples of vehicle use. Thus, we proceeded as follows.

First, we synthesized environmental models from relevant data sources and put them into representations that were most suitable for storing and utilizing the data. In cases where data describing the environment was already available, test course data documented in publications such as ATEC's Test Operations Procedures (TOPs), we obtained (i.e., scanned) the data from the documentation and put it into representations that best facilitated its use.

As the models of the environment became available, we began to synthesize examples illustrating the use of the environmental models in requirements verification scenarios.

As the models and examples were completed, we stored them in a Subversion Repository. Once the Ontological System for Context Artifacts and the Dynamic Context Server became operational, we made the models, examples, and information on the models available through these mechanisms.

In the remainder of this section, we address the following

- Role of Context Models in AVM (Section 2.1, Page 3)
- Deriving the requirements for Context Models (Section 2.2, Page 5)
- Creating Context Models (Section 2.3, Page 9)
- Application of Context Models (Section 2.4, Page 13)
- Examples of Context Model Use in Design Verification (Section 2.5, Page 17)
- Storing and Accessing the Context Models and Examples (Section 2.6, Page 19)

2.1 Role of Context Models in AVM

The role of environmental context models in the Adaptive Vehicle Make program is to help verify that the design solution (i.e., model of an amphibious infantry fighting vehicle) produced by design teams participating in the FANG challenges is capable of meeting or exceeding the mobility and drivetrain related requirements as specified by DARPA's AVM program management team. **Figure 2** below depicts a high level view of the role of environmental models in AVM and design solution development.

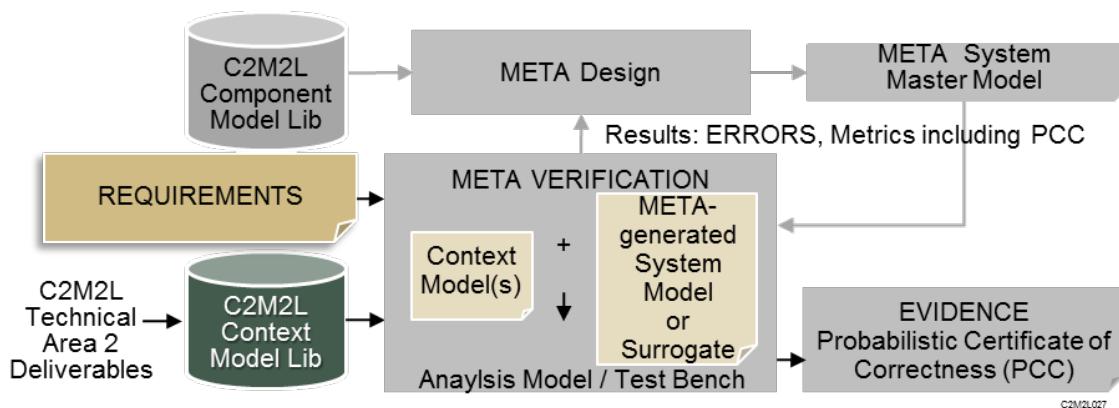


Figure 2: Context Models Support the META Verification Process. Interactions illustrate the use of the context model library and environmental context models in the verification process.

Currently, as implied by **Figure 2**, META verification processes being developed by the META performers use the environmental context models to place a “load” on the design solutions formulated by

their tools. They are accomplishing this by creating simulations which couple the model of the design solution formulated by their toolset to the context model data that will interact with the design solution during the execution of a use case.

For example, one typical military ground vehicle requirement, and one of the FANG IFV mobility requirements, is to determine if the load on the driver's seat will exceed six watts of power as the vehicle is being driven across a test course¹. **Figure 3** outlines how this would be modeled and simulated to compute an estimate of the input to the six watt power calculation for a single vehicle road wheel. Once this is done for a single road wheel, it is relatively straightforward to accomplish for n road wheels. The environmental input is depicted as a green section of terrain depicted in the lower right side of **Figure 3**; the actual input is an X, Z profile of a course that the quarter car is driving over.

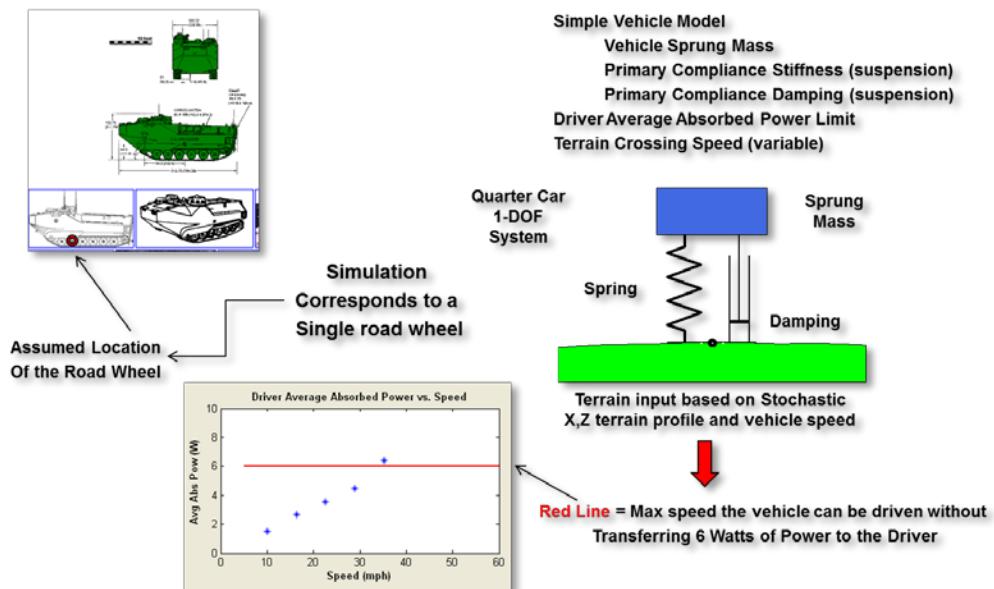


Figure 3: Description of a simulation for computing 6-watt power for a model of a road wheel of an Amphibious Infantry Fighting Vehicle (e.g., a Quarter-Car), and the interaction between the model of the design solution (the road wheel) and the environment (terrain derived from a stochastic representation).

To facilitate the synthesis of the simulations to verify design performance, which are referred to as “test benches” by AVM performers, we created over 20 executable examples that incorporate each of the context models we synthesized or obtained. The examples illustrate how to construct simulations of the interactions between the each of the context models and a surrogate design solution. The executable examples illustrating the use of context models are discussed in more detail in Section 2.5. In the next section we discuss how we identified the context models that were required for evaluating IFV mobility and drivetrain designs.

¹ “... the tolerance limit for representative young American males is approximately 6 watts of continuously absorbed power, ...” NATO Reference Mobility Model, Edition I, User’s Guide Oct 1979 (Referencing: Pradko, F., Lee, R. A., and Kaluza, V., “Theory of Human Vibration Response”, Paper 66-WA-BHF-15, presented at ASME Meeting, Nov 1966)

2.2 Deriving the Requirements for Context Models

We pursued two avenues for identifying the set of context models required to test mobility and drivetrain designs. Since the requirements for the FANG vehicle were not available at the beginning of our effort, we generated the candidate use-cases for a typical vehicle in a bottom up fashion. Specifically, we used BAE's in-house experience in the development of military ground vehicles and descriptions of tests and the environmental contexts in which they are carried out – which are effectively the use-cases for the vehicle. Once the use cases were documented, it was straightforward to derive the context models required to carry them out. This is discussed in more detail in section 2.2.1.

DARPA Program Management made the draft set of FANG requirements available in the July timeframe. The FANG requirements essentially specify the use-cases for the vehicle, and we worked with personnel from Ricardo and Vanderbilt to specify what contexts would be required to test the requirements for the first FANG Challenge. This is discussed in more detail in section 2.2.2.

2.2.1 Bottom-up derivation of Requirements for Environmental Context Models

While the tests that the ATEC community uses vary, the scope of the potential tests, i.e., the vehicle use cases, and the environments in which use cases occur, are relatively invariant. Moreover, the tests and detailed descriptions of the environments in which they are carried out are well-documented. Thus, the requirements for context models can be derived in “bottom up” fashion as follows. By using BAE’s long time expertise in the design, evaluation, delivery and support of amphibious and land-based IFV’s, and reviewing the information on tests and environments contained in the sources that are enumerated in **Table 2**, we were able derive the mobility centric use-cases for an IFV. The mobility centric use cases for land and aquatic mobility testing we derived are listed in the left-hand column in **Figure 4** and **Figure 5**. The columns on the right side of the **Figure 4** and **Figure 5** indicate the general properties of the environment required to carry out each test / use case. The columns on the top right hand side of the figures list the characteristics of the environment necessary to execute each use case.

Table 2: Data Sources used to extrapolate requirements for Vehicle Test Cases (i.e., “Use Cases”), and environmental context models

Description of Data	Reference
Military specifications, regulations, and test operating procedures	<ul style="list-style-type: none"> ■ Mil-Std-810G ■ AR-70-38, Research, Development, Test and Evaluation of Materiel for Extreme Climactic Conditions ■ TOP-1-1-010, U.S. Army Test and Evaluation Command Test Operations Procedure “Vehicle Test Course Severity (Surface Roughness)” ■ TOP-1-1-011, U.S. Army Test and Evaluation Command Test Operations Procedure “Vehicle Test Facilities at Aberdeen Test Center and Yuma Test Center” ■ TARADCOM Signal Analysis Program ■ TOP-1-1-014 Ride Dynamics ■ APG Report No. APG-MT-3635, Special Study of Technique and Study of Automotive Test Course Index, Phase I and II, TECOM Project No. 9-CO-011-000-019, P. Paules and S. Harley, August 1970. ■ APG Report No. APG-MT-4105, Special Study of Establishment of Quality Control Charts for Test Course Inspection Roughness, TECOM Project No. 9-CO-011-000-055, John P. Sobczyk, July 1972. ■ APG Report No. APG-MT-4533, Special Study of Technique and Study of Automotive Test Course Index, Phase III, TECOM Project No. 9-CO-081-000-007, W. Scott Walton, October 1974.

Description of Data	Reference
	<ul style="list-style-type: none"> ■ APG Report No. APG-MT-5882, Final Report on Profiling the Yuma Proving Ground Mid-East Desert Analog Test Course, TECOM Project No. 7-CO-RD3-AP1-012, T. Shrader and W. Connon III, Sept. 1983.
US National Weather Service (NWS) data	www.weather.gov
American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) handbook data	2009 ASHRAE Handbook – Fundamentals
National Solar Radiation Data Base (NSRDB)	http://www.nrel.gov/rredc/solar_data.html http://rredc.nrel.gov/solar/old_data/nsrdb/
Digital Elevation Model (DEM)	http://ned.usgs.gov/

Land Operating Mode and Use Cases		Vehicle Type Applicability	Required Contexts					
			Air Properties	Features	Contaminants	Surface Character	Discrete Obstacles	Terrains
		Atmosphere	Land	Key:				
Unpowered Performance (Ground Clearance, Wheel Loads)		All	I I I	R N R	R	R	R	R
Level Ground Static		All	I I I	R N R	R	R	R	R
Longitudinal Slope		All	I I I	R N R	R	R	R	R
Side-to-Side Slope		All	I I I	R N R	R	R	R	R
Lifting Load (Cranes, Winches, Booms)		All	I I I	R N R	R	R	R	R
Transitioning (Unpowered to Powered)								
Engine Start		All	R I R	I N R				
Engine Idle		All	R I I	I N R				
Engine Tactical Idle		All	R I I	I N R				
Engine Quiet Watch		All	R R I	I N R				
Powered Performance								
Ride Quality								
6-Watt Absorbed Power Curves		All	I I I	R N R				
Average Absorbed Power at Average Speed at $\leq 2.5g$		All	R I I	R N R				
Average Speed at $\leq 2.5g$		All	R I I	R R R				
Cross Country Course at Speed		All	R I R	R N R				
Turning and Steering (Left and Right)								
Straight Ahead		All	I I I	R R R				
Pivot (Retard inside track, power outside track)		Skid Steer	I I I	R N R				
Neutral Axis (Power inside & outside track in opposite direction)		Skid Steer	I I I	R N R				
Moderate Turns		All	I I I	R R R				
High-g Turns		All	I I I	R N R				
Slalom Course		All	I I I	R N R				
Lane Change		All	I I I	R N R				
Turning Performance								
Curb-to-Curb		All	N N N	R N R				
Wall-to-Wall		All	N N N	R N R				
Minimum Inside Wheel Turn Radius		All	N N N	R N R				
Automotive Performance								
Level Ground Constant Speed		All	R R R	R N R				
Acceleration		All	R R R	R N R				
Braking		All	R R R	R R R				
Panic Stop		All	R R R	R R R				
Top Speed		All	R R R	R N R				
Creeping Speed		All	R I R	R N R				
Tractive Effort (TE)								
Max TE Both Sides		Skid Steer	R N R	R N R				
Max TE Per Side		Skid Steer	R N R	R N R				
Continuous TE		All	R N R	R N R				
Differential TE		Skid Steer	R N R	R N R				
Fuel Economy		All	R R R	R N R				
Towing (TE, Braking, Clearances)		All	R I R	R R R				
Towed (TE, Braking, Clearances)		All	R I R	R R R				

Figure 4: Land Use-Cases mapped to required contexts

Aquatic Operating Mode and Use Cases		Vehicle Type Applicability	Required Contexts					
			Air Properties	Features	Contaminants	Surface Character	Discrete Obstacles	Water Properties
		Atmosphere	Land	Aquatic	Key:			
Unpowered Performance								
Static Buoyancy Reserve (Hydrostatics)	Amphibious	I I I	N N N	R R R	R R R	R R R		
Self-righting Buoyancy	Amphibious	I I I	N N N	R R R	R R R	R R R		
Powered Performance (Drag and Hydrodynamics)								
Forward (Transition, Fording, Swim)	All	R R R	R N R	R R R	R R R	R R R		
Reverse	All	R R R	N N N	R R R	R R R	R R R		
Turning and Steering (Left and Right)	All	R R R	N N N	R R R	R R R	R R R		
Towing	Amphibious	R R R	N N N	R R R	R R R	R R R		
Towed	Amphibious	R R I	N N N	R R R	R R R	R R R		

Figure 5: Aquatic Use-Cases mapped to required contexts

After we identified the set of tests necessary to verify the performance of an IFV, we revisited the documents in **Table 2** to determine the contexts required to carry out each use case, and then enumerated them. **Figure 6**, **Figure 7**, and **Figure 8** list the environmental context models required to carry out the mobility centric use cases. With these requirements in hand, we began synthesizing the context models required to thoroughly analyze the mobility performance of an IFV design upon inception of our C2M2L-1 TA2 contract.

Descriptions of Atmospheric Contexts Delivered	
Atmospheric Environment	
Air Properties	
Pressure	
Density	
Moisture	
Temperature (Arctic, Cold, Normal, Hot)	
Temperature (Locally Induced)	
Atmospheric Features	
Wind	
Solar Radiation	
Contaminants	
Corrosive Components (Salt spray, SO ₂ , NO _x)	
Particulates (Dust, Sand, Volcanic Ash, Rain, Snow, Ice Crystals)	
Electro Magnetic Interference (EMI)/ <i>Electro Magnetic Pulse (EMP)</i>	

Figure 6: Atmospheric Environmental Context Models required for mobility testing on an IFV.

Descriptions of Land Contexts Delivered	
Land Environment	
Surface Characteristics (for Depth of Interest)	
Concrete	
Paved	
Dirt	
Sand	
Wet	
Mud	
Snow	
Ice	
Discrete Obstacles (Forward and Reverse, and at Angles)	
Step Climb	
Step Descend	
Gap Crossing	
V-Ditch	
Half-Round	
Curb	
Features found in MOUT (Military Operations in Urban Terrain)	
Jersey Barrier (Highway Divider)	
Improvised Obstacles (e.g., passenger cars)	
Terrains	
Terrains of varying roughness (Flat to 5" in rms)	
Longitudinal Grades (Forward and Reverse)	
Side-to-Side Slopes (Either side up-hill)	
Combined Grade and Slope (Fore-Aft and Side-to-Side)	
Curvature (Turns, Crown, Trough)	

Figure 7: Land Environmental Context Models required for mobility testing on an IFV.

Descriptions of Aquatic Contexts Delivered	
Aquatic Environment	
Water Properties	
Density	
Temperature	
Viscosity	
Thermal Conductivity	
Specific Heat	
Water Body Features	
Depth	
Calm	
Surf	
Currents	
Sea-State	
Contaminants	
Salt	
Particulates (Sand, Volcanic Ash)	
Debris (Vegetation, Spills)	

Figure 8: Aquatic Environmental Context Models required for mobility testing on an IFV.

2.2.2 Top-Down Derivation of Requirements for Environmental Context Models

Once the performance requirements for the mobility and drivetrain system to be constructed for the FANG-1 challenge became available (in the July 2012 timeframe), we worked directly with the authors of the requirements to derive the environmental context models required to test the requirements in top-down fashion. This was accomplished via a series of weekly meetings with system engineering representatives from Ricardo, the authors of performance requirements for the FANG-1 challenge, and META tool chain implementers from Vanderbilt (CyPhy) and CyDesign. For the most part, the team met on a weekly basis from the end of July 2012 through December 2012.

By the time the FANG performance requirements were made available, our C2M2L-1 TA2 efforts were well underway as our participation in the program began in the March 2012 timeframe. Fortunately, the requirements that we derived in bottom-up fashion listed in **Figure 6**, **Figure 7**, and **Figure 8**, which we used to guide efforts to synthesize context models and examples, yielded a set of context models compatible with the requirements that we derived in top-down fashion from the FANG performance requirements. As of this writing, a subset of the models we produced is being utilized by the CyPhy and CyDesign toolset implementers to perform verification in the toolsets being utilized for the FANG-1 challenge.

In the next section, we discuss the methods we employed for synthesizing environmental context models based upon the requirements we derived, our results, and our findings and conclusions.

2.3 Creating Context Models

As a general approach to creating models, we use the process shown in **Figure 9** below. The process entails observation, characterization and modeling.

In **Figure 9**, the *observations* essentially supply the set of data that we use to formulate a model. The *information* step is a process of characterizing the data to determine whether it could be used in deterministic “go/no go” test settings (such as an obstacle profile), or whether it is a stochastic variate that requires a full probability measure of its state space (such as significant wave height prediction for sea-state modeling).

The last *knowledge* capture step uses the results of characterization to create the necessary numerical model for the environmental context being tested against. For high-fidelity models, this is often the most unwieldy step, as it may require multi-disciplinary physics to adequately represent. On the other hand,

the consideration of statistical physics or thermodynamic laws may allow the model a concise mathematical representation which agrees with the empirical data.

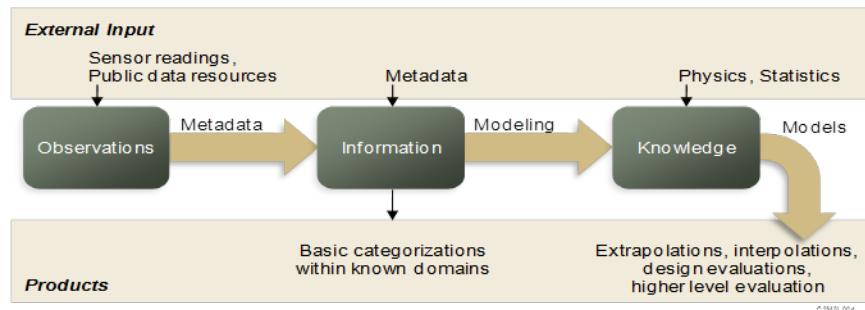


Figure 9: Outline of Process for Creating Models

The observations were typically taken from previously collected and published data, while the characterization step was used to determine whether we could apply any patterns of use from other modeling domains.

The pattern that we often employed leveraged the maximum entropy (MaxEnt) principle [1] to fit empirical distributions based on a key set of characterized statistical moments. For example, we could often fit empirical distributions based only on the characterized statistical mean or mode of an observed variate (this worked well for wind, rain, particle sizes, among other domains).

2.3.1 Discussion

Several of the modeling domains that were required for environmental contexts were either lacking in characterization detail or pedigree, or had limitations placed on their use (e.g. proprietary IP, classified-use, etc.). Based on the extensive collection of environmental modeling domains that a vehicle context typically involves, we had to apply several shortcut heuristics and basic algorithms to ensure we had enough fidelity to cover all the cases. A goal of modeling is also to reduce the complexity and magnitude found in raw data, and we have high-resolution evidence (courtesy of data collected by our C2M2L colleagues at CSIR [2]) as to where this information reduction is applicable.



An effective strategy we used was to develop models that were founded on essential stochastic patterns observed in nature [3]. From this foundation, a general pattern that we relied on involved creating probability density functions (PDF) based on applying the maximum entropy principle to the observed

data. This was an approach predicated on limited information or knowledge that leveraged Jayne’s information theory concepts [1].

Another strategy was based on applying heuristics to restricted-access (ITAR) information. For vehicle dynamics context models, we were restricted to indirectly using power spectral density (PSD) plots of terrains which could then generate an equivalent real-space terrain profile.

2.3.2 Technical Challenges

The paths of developing context content involved (1) creating models from scratch, (2) extending models from public data sources, or (3) applying heuristics on related data that would retain a substantial portion of the information content.

In some cases, we had to process significant amounts of data from available environmental repositories; this was necessary to generate adequate statistical precision for the low-probability cases that we were interested in (e.g., gathering significant wave height statistics for severe sea-state classifications).

In one class of cases, we needed to apply screen-scraping to extract the non-classified parts of critical test data, and then apply a key heuristic to produce tolerable fidelity that would be suitable for focused crowd-sourced² contextual use.

The ongoing challenge is to produce environmental models for the state-space that a vehicle interacts with. The issue with detailed proprietary physics models is that they are often considered valuable intellectual property (IP) and so to simplify even some of these models in a fashion suitable for open development has enormous benefits.

2.3.3 Technical Results

For the domain of terrain profiling we evaluated a number of approaches. A heuristic model based on PSD data sets produced adequate results for nondescript/random terrain. A semi-Markov approach that we developed during the course of this project worked very well to model recent high-resolution data produced by CSIR [2]. These terrains included semi-periodic structures such as Belgian block cobblestone and potholes. Another promising approach developed during this project uses a kernel transform that is an outgrowth of shock and vibration modeling [4].

The set of results is more completely covered in the following appendices: Appendix A covers the stochastic modeling approach and the results from applying MaxEnt to a number of environmental domains [5]. Appendix B covers fine terrain and gross terrain (topography) and evaluates the effectiveness of heuristics[6]. Appendix C covers stochastic growth models that find application to thermal and corrosion modeling [7]. Appendix H covers the kernel approach to terrain profiling [4].

For compliant modeling, Appendix I describes a library containing nominal climate conditions and a thermal design example, and Appendix J and Appendix K describe approaches for granular and soft soil terrain.

² This is not crowd-sourcing in the current use of the term, but focused on technically proficient engineers who often can offer non-traditional design experience.

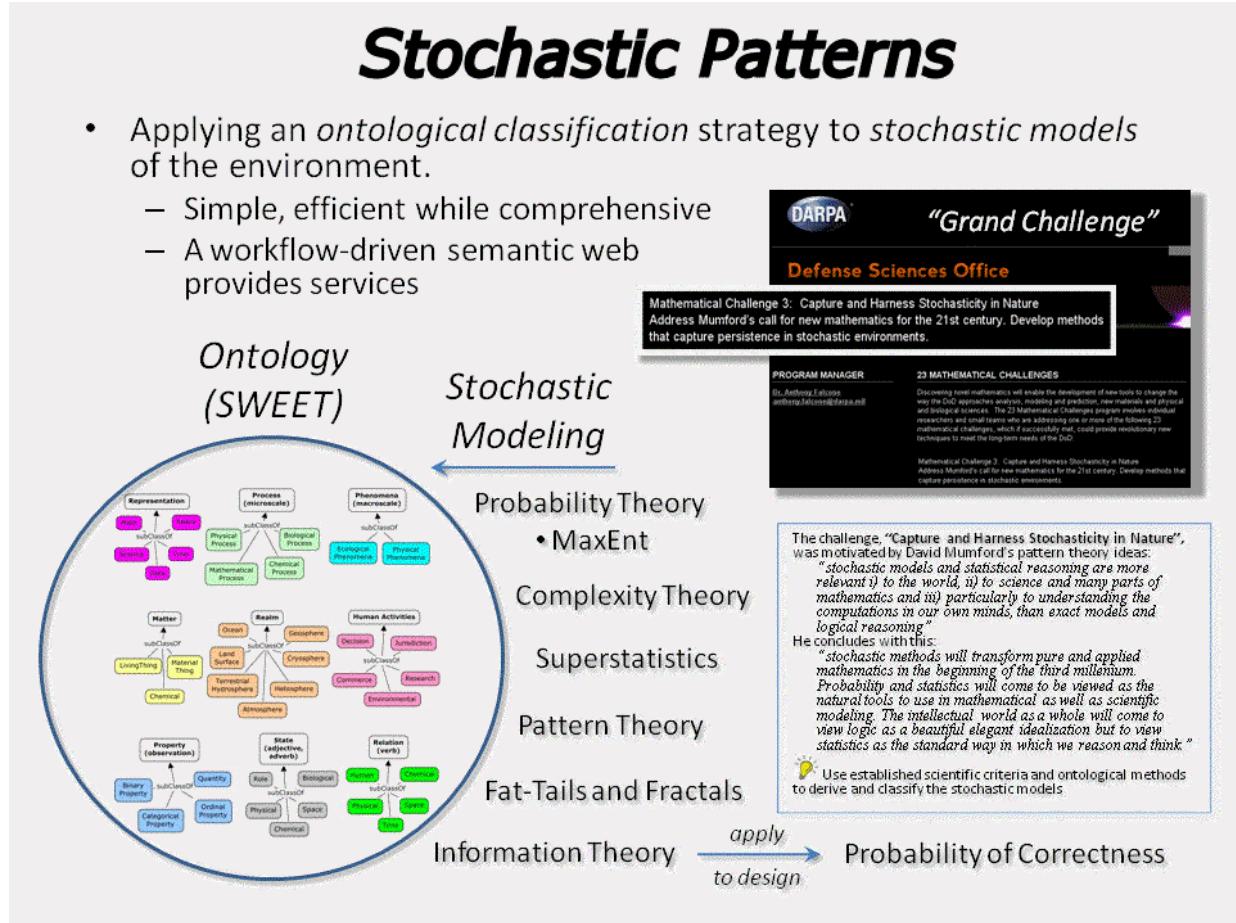


Figure 10: Integration of stochastic modeling with the semantic web

2.3.4 Findings and Conclusions

The most frequent result we came across when characterizing and modeling domains was how often we could apply previously observed patterns, and especially those derived from MaxEnt principles.

The combination of stochastic patterns mapped onto ontology allowed us to classify the patterns as well as the environmental domains (see Appendix D).

Within the domain of terrain modeling, data from state-of-the-art terrain profiling data collected by CSIR benefited from the models we are currently developing. The spatial resolution was high enough that the stochastic and periodic features of the track features could be clearly delineated and reduced to compact parametric models. By doing this we can reduce data that could occupy potentially gigabytes of storage into a representation of a few parameters.

These findings resulted in several research papers that we either have submitted, or are submitting, for peer-reviewed publication.

2.3.5 Implications for Further Research

The approach we took was partly inspired by a DARPA mathematical grand challenge, "Capture and Harness Stochasticity in Nature", which seeks to address "Mumford's call for new mathematics for the

21st century, methods that capture persistence in stochastic environments” [8]. As **Figure 10** shows, we embraced and utilized Mumford’s ideas, and believe that this strategy holds great promise for future modeling work.

2.4 Applying (Representing/Storing and Using) Context Models

Context models can fill the role of verification for a vehicle design and for evaluation of fitness-of-use within a specific operating environment. This has the potential to be an unbounded exercise — as the possible environmental conditions are combinatorially unlimited — yet we can address the problem with alternative strategies. The alternate approaches can involve estimating probability densities for model outputs and/or generating Monte Carlo simulations based on modeling observed stochastic behaviors. These become suitable for evaluating correctness (i.e. PCC) of vehicle designs when used in an environmental context. This section describes the classes of context models available to implement for use in a model library.

2.4.1 Technical Challenges

The challenge of verification remains one of essentially proving that a given vehicle design will meet its requirements over all the environmental and climatic states it may be subjected to. The options for this class of state-space evaluation are shown in **Figure 11** below. The simplest case is to consider *deterministic* paths through the state-space. This is typically taken as a path through a specific test-track[9] or the operation under a nominal set of climatic conditions[10].

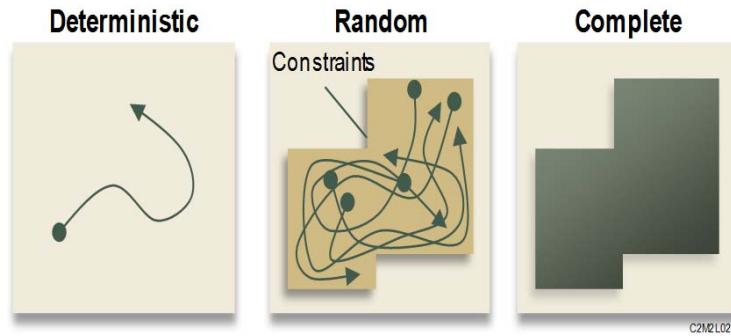


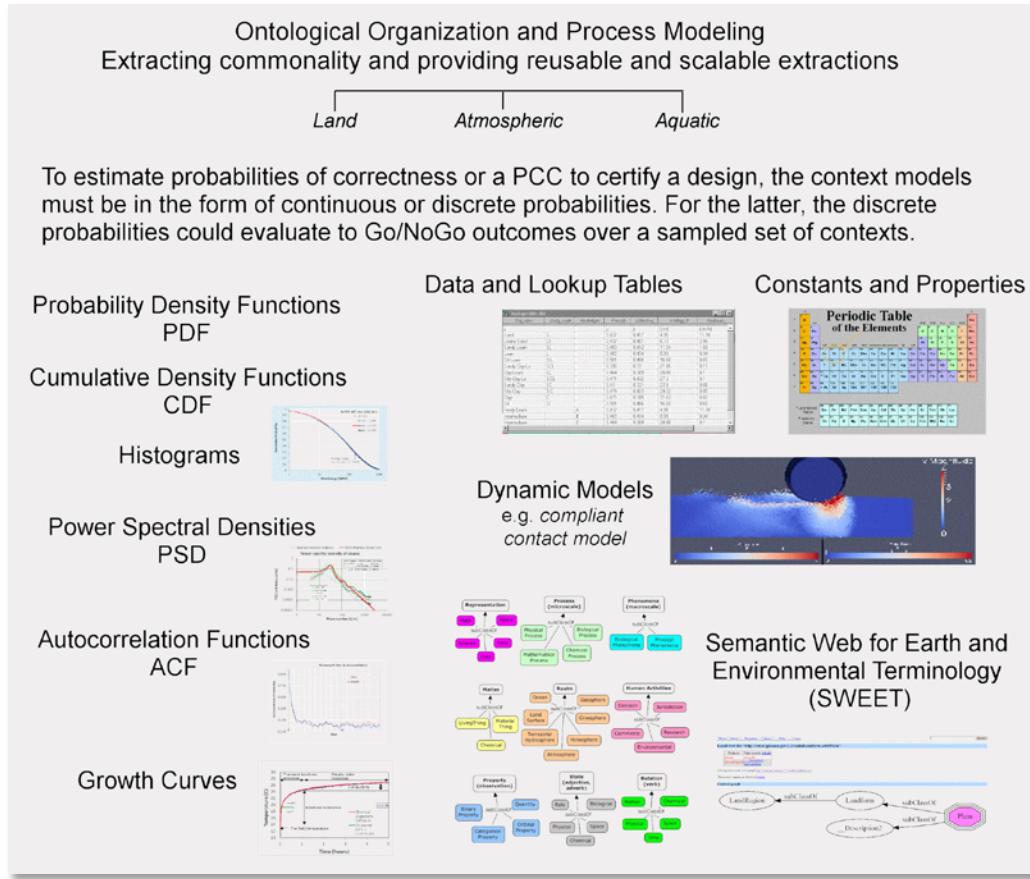
Figure 11: State-space for model-based verification

On the other side of the spectrum, it becomes almost intractably complex to assume that all possible states of the environmental context can be covered. This is considered the *complete* situation, yet this approach is often the only option if one needs to prove some conjecture in formal terms.

In between the range of complexity from deterministic to complete is the *random* or stochastic space of probabilistically-based states; with the outcomes also specified as probabilities. The challenge is to create models of sufficient fidelity so as to represent the environment context adequately. In this case, the most critical states are the ones that occur rarely or are considered extreme values. Since these occur on the tail of the probability curve (i.e., they occur with low probability), they are more difficult to estimate.

2.4.2 Context Model Scope and Representations

The hierarchy of **Figure 12** describes the range of models that we utilized to represent and store environmental context models. Models cast as PDFs or CDFs (and even raw histograms) can be used to generate Monte Carlo samples for use in simulations, once they are inverted.

**Figure 12:** Ontological organization of Context Models

2.4.3 Technical Results

The list below is a sampling of results on the set of terrain models we have either evaluated or formulated as part of this effort. This set reveals the various levels of abstraction that a modeler can apply to terrains:

- 1. Heuristics:** Terrain profiles can be recovered from a PSD through the Fourier series superposition of sine waves method. Although not necessarily precise due to loss of information [11], these are adequate to express the root-mean-square (RMS) of the course profile, which is vital for vehicle suspension and ride comfort modeling. This was implemented as a functional mockup unit (FMU) to interface to clients of the terrain model integrated within a vehicle test-bench via the Functional Mockup Interface (FMI), and it was also used to interface to a 3D rendering package.
- 2. Improved Heuristics:** To improve on the Fourier transform heuristic, a kernel-based spectral decomposition approach was formulated during this project. The algorithm derived from a shock-and-vibration analysis application which was then applied to modeling fine-grain terrain profiles[4] (see Appendix E).
- 3. Stochastic PDF:** Given enough data, such as is available with large topographic data sets, we can extract probability density over many orders of magnitude. In the case of characterizing terrain slopes, we modeled the PDF with a single mean value and specific profile shape over the USA lower 48 states (see Appendix B).

4. **Markov model:** Where we have good data on topography, and can perform pair-correlation on variants such as terrain elevation, we can provide models based on Markov step transitions. Covered fully in Appendix B, we were able to automatically assess and parameterize a marginal probability model for a given region. The simulated terrain relief could be recovered by applying a random walk algorithm to the Markov model parameters.
5. **Semi-Markov model:** Where we have very precise data on fine terrain relief and which may also include specific features, we can apply a semi-Markov model that was developed during the course of this project. This stochastic approach is able to parameterize partially-ordered or quasi-periodic phenomenon such as cobblestones, pot-holes, and wash-boarding features on track profiles. We can recover the realistic detail by taking the parameters of the semi-Markov model and then feeding those into a variation of a random walk simulator (see Appendix B).
6. **Compliant Physics-based models:** We have two detailed reports, on the modeling of compliant environments - “Massively Parallel Discrete Element Modeling of Wheeled Mobility” on Granular terrain in Appendix J, and “Simulation of tracked wheels on soft soil substrates” in Appendix K. Compliant environments are models of the environment (e.g., sandy soil) that change during the course of simulated vehicle interaction, while non-compliant environments are not affected by their interaction with the vehicle during the course of simulation. As appendices J and K indicate, the complexity of a simulation increases significantly when compliant terrains are used in simulation, but they produce more accurate results than simulations that utilize non-compliant environments.
7. **Formally Complete:** The Markov model of topographic elevation changes described in (4) was applied to a formal tool called Prism. A state-machine vehicle description was integrated with a Markov chain representation to prove correctness over a range of possible slopes and elevation changes (see Appendix F).
8. **Deterministic:** Obstacle profiles were provided for specific test courses, including cross-tie, side-slope, staircase, etc.

Depending on the application and the resolution required, we can choose the appropriate terrain model. For example, **Figure 13** below illustrates the need for a resolution at two levels, a fine detail to represent roughness (as a spectral model) and a gross detail to represent elevation changes.

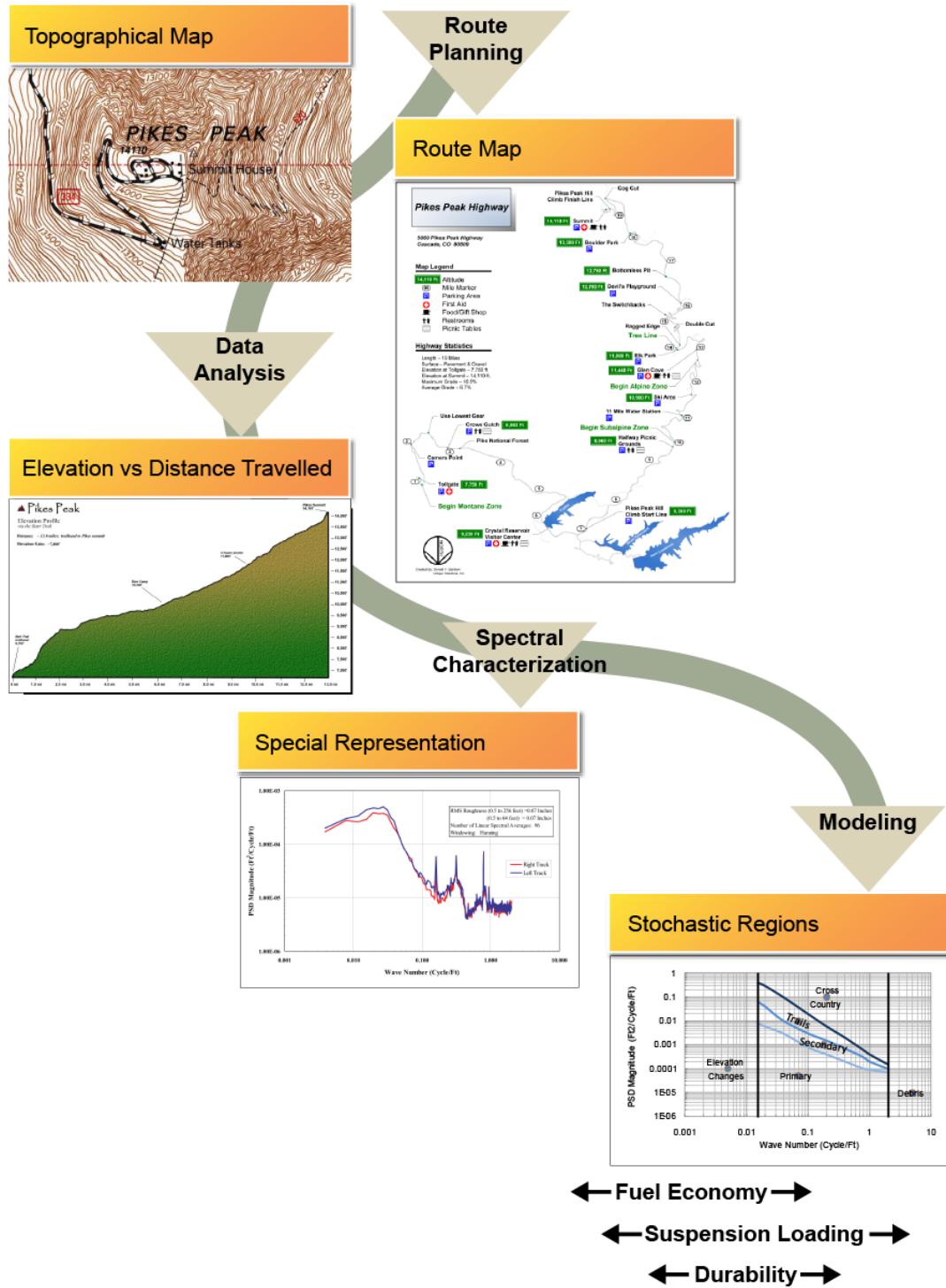


Figure 13: Terrain representation at a gross topographic resolution, suitable for long range analysis, and a fine resolution suitable for vehicle suspension analysis.

A similar classification of models was done for models in the aquatic and atmospheric realms.

2.4.4 Findings and Conclusions

The potential for the generation of unlimited simulation outcomes from a stochastic model allows a systematic coverage of the environmental state-space to be performed. As the set of outcomes remains probabilistic, it is not yet considered complete, but is nevertheless useful for evaluation of a PCC and other measures of correctness.

In Appendix F, we demonstrate how a Markov terrain elevation model is used to provide constraints on a vehicle model. This is essentially an example of integration with formal verification tools. This approach is illustrated in **Figure 14**.

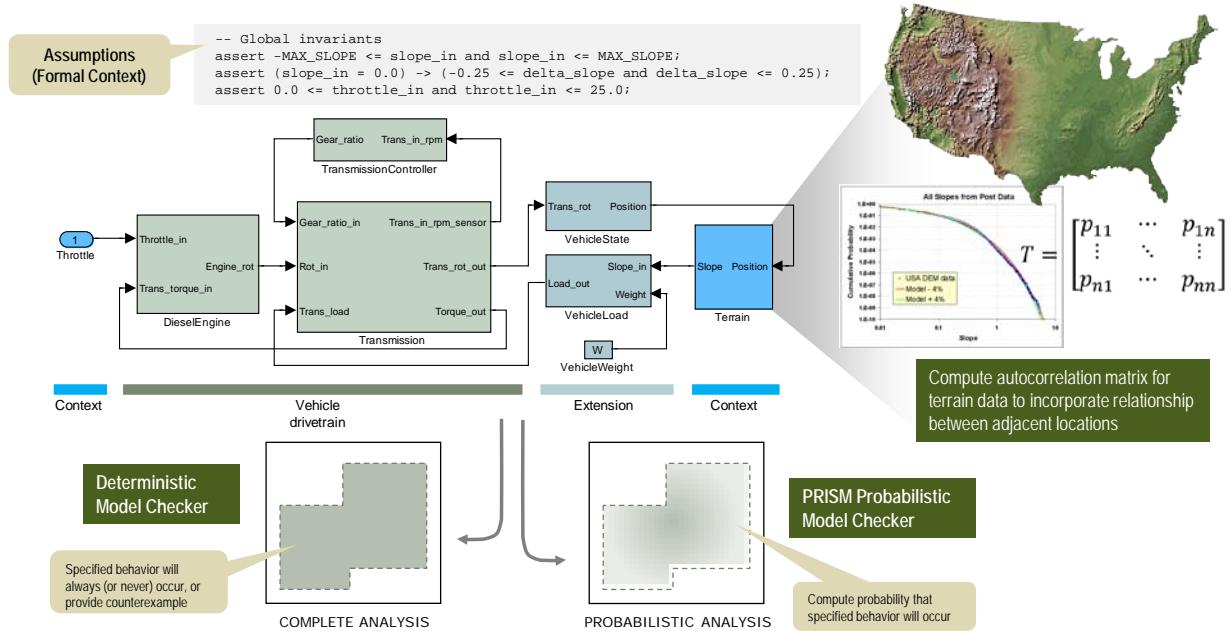


Figure 14: Usage of context model in a vehicle model checker

Considering the span of models we have investigated, which can range from the application of heuristic approaches, to more formal Markov and semi-Markov models, and then to detailed physics models, we have made in-roads toward formulating a comprehensive set of usage patterns.

2.4.5 Implications for Further Research

In terms of design problems that require a probabilistic certificate of correctness, the probabilistic and complete approaches discussed above are of vital importance.

2.5 Examples of Context Model Use in Design Verification

We created over 20 examples to verify the context models and to provide illustrations of how to use each instance of a model in a simulation to validate a typical IFV performance requirement. Appendices G, K, L, M, O, and P provide more detail on each of the examples using non-compliant environments (i.e., environments that do not change during interaction with the vehicle). Appendices J and K provide details on examples that have contexts which are compliant (i.e., environments that do change during interaction with the vehicle). Finally, Appendix Q discusses the examples we formulated and delivered illustrating

how to interface our context models to designs simulated in Open Modelica via the Functional Mockup Interface.

We created executable examples illustrating the use of each context model in each of the 45 classes of environmental context models listed in **Figure 6** (10 Classes of Atmospheric Context Models), **Figure 7** (25 Classes of Land Context Models), and **Figure 8** (13 Classes of Aquatic Context Models) in requirements verification scenarios. Appendix Z, the Context Model Map, contains a mapping of the 45 classes of context models to each instance of a context model in a class, and examples of their use in verification scenarios.

Figure 15, **Figure 16**, and **Figure 17** provide a sampling of the how environmental contexts in the Land, Aquatic, and Atmospheric realms are coupled with designs to validate a design’s ability to meet its performance requirements. **Figure 15** depicts a view of the Unity-Based simulation we formulated to facilitate the visualization of terrains generated from Spectral representations of the Army test courses we obtained, and the effect of those courses (i.e., terrains) upon a conceptual design when the design interacts with the terrain as it is driven across that terrain. In this case, the context models we synthesized provide insight into how the design will perform against its requirements to traverse terrains of varying roughness and frictional coefficients.

Figure 16 depicts a view from that same Unity Based engine simulating the interaction between a Conceptual Vehicle design and water, as well as a surf zone. In this case, the context models provide insight as how the design will perform against its requirements to navigate through various sea-states and when traversing surf zones. Note, the vehicle design model pictured in **Figure 16** is the surrogate design that we use in the majority of our examples, which we refer to as a Virtual Automated Test Rig (VATR). It is based on an actual amphibious vehicle available from Hydratek (www.hydratek.com). A detailed description of the model, including a computer aided design rendering in the tool Pro-Engineer can be found in Appendix O under the discussion labeled Amphibious Context Model.

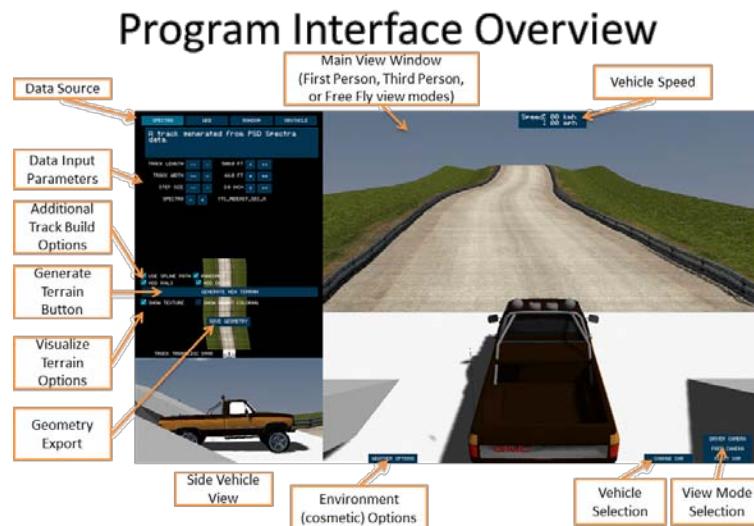


Figure 15: Track Builder: A Unity-based Application for Validating Context and Vehicle Concept Models Performance in Various Land and Sea Contexts. More detail on the application and example can be found in Appendix L

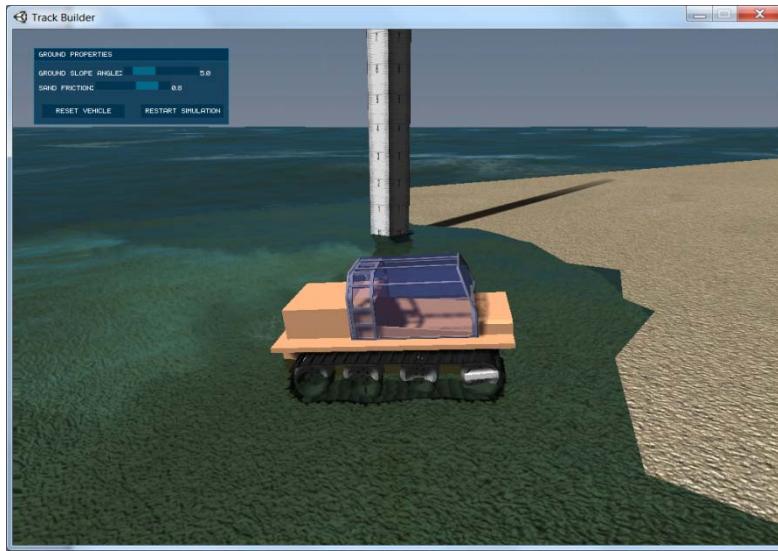


Figure 16: Extension to Track Builder for Validating Vehicle Concept Performance in Surf Zones. More detail on the application and example can be found in Appendix L

Figure 17 illustrates interaction between that atmosphere and a portion of a vehicle design, an air filter on an intake blower to a vehicle's engine. In this case, the simulation is used to determine the effect on vehicle performance when the intake flow decreases due to the build-up of particles on the air filter. Specifically, the simulation can be used to determine how long it will take the vehicle flow rate to drop below the required performance when filtering particles of various sizes.

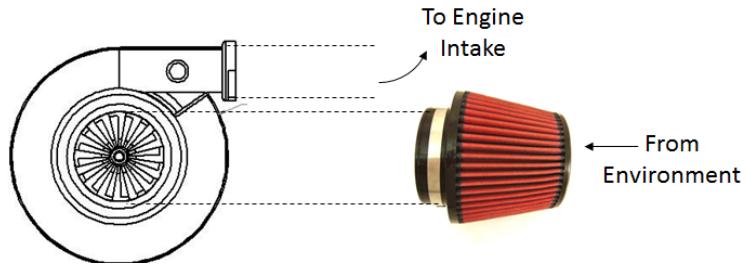


Figure 17: Filter Example to Demonstrate interaction between Atmospheric Contexts and Vehicle Subsystems. More details on the example can be found in Appendix O

2.6 Storing and Accessing Context Models and Examples

As with any library, a Context Model Library requires a certain level of organization and categorization to make it usable and to keep it maintainable. It must also provide a reliable means of delivering models and related artifacts, as well as providing documentation where needed. The executable context models and their documentation, as well as the context models associated with the examples are available via the Subversion system that we used for revision control as we developed the context models and examples of their use. Appendix Z, lists the location of the context models and their examples of use within the SVN (SubVersion Network) repository.

We then engineered and delivered a library delivery system based on a semantic web organization[12] and classified according the Semantic Web for Earth and Environmental Terminology (SWEET) ontology. The system was built as an interactive web server that consisted of two major parts, a model

delivery service called Ontological System for Context Artifacts and Resources (OSCAR) and a flexible interactive knowledgebase called Dynamic Context Server (DCS). As architected, OSCAR and DCS perform as portals for serving context models and knowledge which can meet the needs of vehicle design and test. In addition to the SVN locations of the Context Models and Examples, Appendix Z also lists the means for locating and accessing context models and examples of their use using OSCAR and DCS.

2.6.1 Technical Challenges

The challenges in creating a semantic web server for C2M2L related mainly to the breadth of coverage required in managing models that crossed the land, aquatic, and atmospheric environmental realms. The scope of effort was large enough that the library required various means of searching and selecting models. Part of the challenge was to make it automated so that the user could provide project requirements and the built-in library reasoner could find and present choices of models that the user could select and that would meet the need of an environmental context requirement.

2.6.2 Overview of Development Methodology

The approach taken was to apply state-of-the-art semantic web development processes to build OSCAR and the DCS. For OSCAR, we used a Java-based production model, while for DCS we employed a first-order logic language (Prolog) that was cleanly coupled to a triple-store knowledgebase. The mix of an established architecture (OSCAR) with a next-generation declarative knowledge-based approach (DCS) allowed us to experiment and prototype a wide range of service mechanisms. We applied patterns and archetypes to optimize the amount of reuse in both context models (which often had similarities across physical domains) and in the semantic web building blocks.

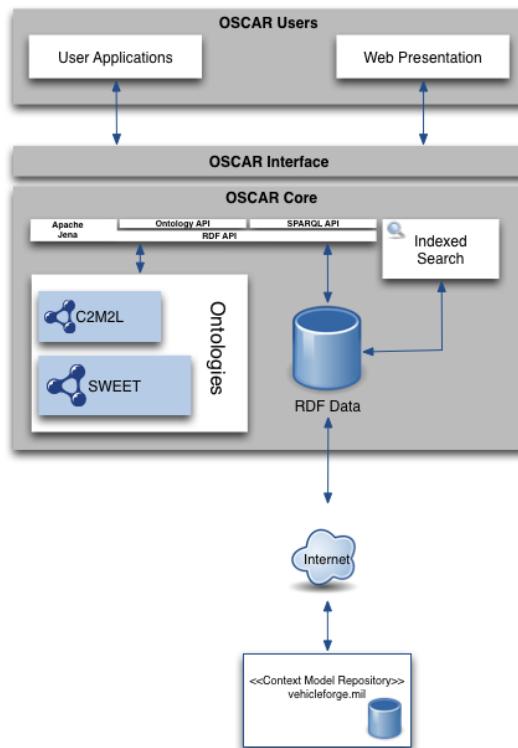


Figure 18: Ontological System for Context Artifacts and Resources (OSCAR) Architectural Overview.
OSCAR is described in Detail in Appendix E



Figure 19: Dynamic Context Server Home Page. The Dynamic Context Server is described in detail in Appendix D

For serving models and artifacts, we organized services into several categories discussed in the following sections.

2.6.2.1 Standalone or embedded models

The model is delivered as executable code that is linked to a vehicle simulation test-bench. After C2M2L-1 TA2 program inception, in the July timeframe, AVM program management selected the Functional Mockup Interface[13] as the preferred interface mechanism, although other approaches include S-functions in Matlab, general compiled C-code, or interpreted embedded Python.

The standalone/embedded models can be potentially generated from a reasoner which uses information on algorithms and data to construct the necessary code. We used OSCAR as a vehicle for this.

2.6.2.2 Served models

The model resides on a web server hosted on a cloud computing environment. Information is served for Monte Carlo simulations run by clients that can interface through a web service. Organization is provided by the DCS semantic web back-end which keeps track of the models available.

Most of the environmental context models have similar patterns, in that they possess salient characteristics that follow standard representations such as probability density functions (PDF), power spectral densities (PSD), and amplitude profiles. Since these are standard formulations, they can be aggregated and distilled through a common pattern and reuse framework.

2.6.2.3 Data file models

Data files are either delivered through a web service or file server, with the semantic web service again providing organization above that a file server will provide. Both DCS and OSCAR supply data files.

2.6.2.4 Model artifacts (graphs, visuals, etc.)

Artifacts for models can include interactive PSD and PDF charts, and generated Monte Carlo profiles. These are useful for the user to understand the ranges, extreme values, and severity of the various models. More elaborate are context model visualizations which can include 3D rendering of terrains and aquatic surfaces. Both DCS and OSCAR supplied artifacts.

2.6.2.5 Model metrics

Similar to artifacts, metrics contain auxiliary meta-information on models such as maturity and potentially complexity, which will be useful for model curation activities.

2.6.3 Technical Results

The context server covered approximately 10 physical domains spread across the land, aquatic, and atmospheric realms. The main feature classifications included fine terrain (for vehicle suspension contexts), gross terrain (for vehicle range), corrosion, thermal, precipitation, aquatic wave, inland water, particulate, EMI, and wind models.

The main usability aspects related to providing search, browsing, requirements, technical resource access (such as standards tables and a comprehensive reference and citation database), and workflow interfaces to the featured domains, with context-sensitive help guiding the process. Most of these usability features took advantage of semantic features such as terminology look-ahead and sophisticated pattern-matching based on similar modeling archetypes.

Examples across the domains were provided for how to use models of sea-state, rough track profiles, obstacle courses, temperature and thermal diffusion, topography, fording, EMI clutter, corrosion, buoyancy, droplet size, and atmospheric properties. The examples included generation of artifacts such as probability density functions (PDFs) and power spectral density (PSD) plots, and use cases where constraints or environmental extremes played a role.

2.6.4 Findings and Conclusions

The delivery of models based on requirements worked well and we were able to create rules that would be robust against churn in how requirements were specified.

The dynamic context server was deployed on an evaluation computing cloud and found to work well, with plans to include it on a Vehicle Forge site for test development.

Appendix D titled “Knowledge-Based Environmental Context Modeling” goes more deeply into architectural decisions made and utility of the overall approach.

Appendix E titled “Ontological System for Context Artifacts and Resources” provides a supplemental view for user manipulation of models (registering, modifying, and downloading).

2.6.5 Implications for Further Research

The performance and scalability afforded by state-of-the-art semantic web development architectures indicates that the OSCAR and DSC servers can evolve -- simply by the addition of knowledge to the triple-store. By adding a more comprehensive set of geospatial location-specific models and models outside the domains evaluated for this project, we can extend the functionality, while the pattern-matching and reasoning afforded by a knowledgebase-aware server does the rest of the heavy lifting.

As with many semantically-based systems, as long as users and curators maintain ties to an ontology, the approach can continue to grow and expand into any realm. For example, by tying in a component ontology (which we prototyped with the XSB component ontology), we can associate available component materials[14] with environmental models of corrosion and thus better integrate the design and verification space.

3. Results

The primary objective of this project was to provide environmental context models for use in vehicle design efforts.

3.1 Technical Challenges

The objective presented dual challenges in delivering context models for practical use, as we needed to (1) provide comprehensive coverage across the land, aquatic, and atmospheric realms within scope for this effort and (2) maintain sufficient accuracy and fidelity such that they are both usable and adequate for vehicle simulation needs.

Concerning accuracy, any empirical, analytical, or computational data that comprises the foundation of a context model will necessarily have some uncertainty associated with it³. This can be either aleatory uncertainty, which exists as part of natural variation, or epistemic uncertainty, associated with our inability to quantify the data precisely. To certify the correctness (to a given confidence level) of a system design within the constraints of the environment requires uncertainty quantification of the context models.

Concerning fidelity, the objective should be to trade-off the level of accuracy in the environmental characterization and its resultant context model, with extra flexibility necessary to adjust the models in light of changing or emerging accuracy requirements. As this extra flexibility leads to more choices, the challenge is to provide automated ways to search, manipulate, and translate context models to aid in the use and maintenance of the model library. Any documentation and pedigree data needs to go with each context model.

As is the case with component models, the context models need to be supplied in a suitable domain-specific or multi-domain modeling language with well-defined, formal semantics and interface specifications. The choice of language needs constructs and operations that are grounded in mathematical terms. This enables us to more easily prove or verify that a modeling behavior holds true, while also making it easier to interface with clients of the models. The ultimate challenge is to prove that model constraints and conditions are met and that intended states are reachable and undesirable states avoidable,

³ unless we deal with the most simple of contexts and phenomena, such as the practical measure of gravitational force, which can be made accurate for terrestrial use

which is a necessary ingredient to formulating criteria with regard to the correctness and completeness of a vehicle design.

3.2 Discussion

The models we developed were based on the initial aim of the project, which was to evaluate an amphibious vehicle's drive-train and mobility subsystems with respect to their ability to meet specified performance requirements. In addition to a nominal list of domains that were pre-selected based on past needs, specific environmental features were modeled based on vehicle requirements developed during the course of the project. This included discussions with the test-bench teams that were responsible for evaluating vehicle designs.

An ontology-based semantic web-server was selected to provide organization of models at various levels of fidelity.

3.3 Technical Results

The classes of context models we synthesized and delivered are listed in **Figure 6**, **Figure 7**, and **Figure 8**. Details of the technical results are provided in several appendices, which are described as follows:

Appendix	Description
A Stochastic Modeling	Fundamental principles of thermodynamics and statistical physics can be used to create compact parameterized models capable of statistically capturing the patterns exhibited in wide range of environmental contexts. Such models allow more efficient and systematic assessment of the strengths and weaknesses of candidate designs than is possible using benchmark datasets or test tracks, and this can play an important role in utilizing computer simulation to produce better designs of complex cyber-physical systems more quickly, affordably, and reliably.
B Terrain Spectroscopy	For terrains the underlying behavior is rarely controlled by a completely ordered process, and any model characteristics will carry along with it a level of aleatory uncertainty governed by the natural disorder. This appendix applies novel uncertainty characterization approaches to classes of topographic models to demonstrate how to quantify the natural order and distinguish from artificial (man-made) order.
C Uncertainty Quantification in Growth Models	Environmental models by their nature contain a great deal of uncertainty. Since the underlying behavior of the model is rarely controlled by an ordered process, all model characteristics will carry along with it a level of aleatory uncertainty governed by the natural disorder. This appendix applies novel uncertainty quantification approaches to classes of diffusion problems such as corrosion which illustrate the benefit of assuming natural variability.
D Dynamic Context Server	This appendix describes a semantic web architecture based on patterns and logical archetypal building-blocks well suited for comprehensive environmental modeling framework. The patterns span a range of features that cover specific land, atmospheric and aquatic domains intended for amphibious vehicles. The modeling engine contained within the server relied on knowledge-based inferencing capable of supporting formal terminology (through the SWEET ontology and a domain specific

		language) and levels of abstraction via integrated reasoning modules.
E	OSCAR	OSCAR (Ontological System for Context Artifacts and Resources) is a knowledgebase library system for context models. We implemented an archival system that is driven by the backend ontology designed for environmental context along with concepts for common archival transactions.
F	Formal Methods and Model Checking	Describes work on probabilistic context models and associated probabilistic analysis methods. This covers (1) developing a probabilistic ground vehicle drive train model to interact with terrain context models, (2) creation of probabilistic terrain context models suitable for model verification, (3) developing analysis tools for assume/guarantee-style contracts in probabilistic models.
G	EMI Modeling	This appendix presents a set of modeling methods and general context models for predicting susceptibility of electronic equipment to MIL-STD-461F environments with respect to reception of interference via the cable/connector assemblies. This covers three areas – system electromagnetic environment flow-down to equipment and cable/connector assemblies, modeling method for cable/connector assemblies, and a detailed example.
H	Novel Synthesis of Shock & Vibration Profiles with Potential for Terrain Profiling	An approach is presented to synthesize a base shock-and-vibration acceleration time-history that is compatible with a prescribed power spectral density (PSD). The synthesized acceleration is developed using the summation of sinusoids with frequencies that cover the frequency range of the PSD. The peak amplitude of the envelope is sized to match the total power of the prescribed PSD to that of the PSD that results from the synthesized base acceleration. Examples are presented to illustrate the process for shock-and-vibration data and extended for potential use in terrain roughness analysis.
I	C++ Library	A C++ class library of nominal and extreme climate-based environmental models drawn from the AR 70-38 and ASHRAE specifications.
J	Compliant Modeling	Quantitative understanding of wheeled mobility on granular terrain such as sand or gravel is critical for design and effective operations of ground vehicles. This presents findings of massively parallel discrete element modeling of wheeled mobility on granular media such as sand, resulting from parametric studies with varying levels of both wheel penetration and mobility conditions. The important details of the problem are retained by simulating granules of the size encountered in real terrain to overcome the fidelity limited issues of other comparable methods that use much larger granules.
K	Tracked Wheels on Soft Soil Simulation	Multiple analysis strategies were applied to the problem of simulating a tracked wheel on soft soils, with the ultimate goal of generating results that could be compiled into a context model. The strategies included Finite Element Analysis, FEA using eroding solid elements, SPH (Smoothed Particle Hydrodynamics) technology, and SPH technology utilizing adaptive solid elements.
L	Surf to Shore	The mixed surf and shore environment, due to its chaotic nature, is perhaps the most difficult context to model. The vehicle is simulated using

	physics provided by the Unity 3D visualization rendering engine. Physical properties are assigned to the geometry in the scene. Properties include static and dynamic frictions. Forces and torques are applied to a rigid body, which cause the vehicle to move. Buoyancy forces act while the body of the vehicle is in water. Friction properties react between the wheels and linkages making the track, as well as the track with the ground when it is in contact.
M Test Case Amphibious Vehicle Model	A vehicle model to serve as a platform for evaluating terrain models was provided.
N Ecto-VATR	The Early Concepting Tool – Virtual Automated Test Rig (ECTO-VATR) is a system design tool that enables editing of a model of vehicle design primarily through the hierarchical assembly and manipulation of components from the Component Model Library (CML). It is focused on empowering a designer operating in an early design phase to be able to incorporate and manipulate major design drivers and rapidly assess the quality of system concepts. Resultant concepts can be used as the basis for more detailed design. It illustrates a working concept and our vision of how context models should be integrated into the design verification process in system development.
O Examples of Context Model use with Vehicles	This section describes context models used with stand-alone executable vehicle simulations. The simulations were developed in Matlab or C++, either directly linked to context models, or run-time linked via FMU interfaces.
P Amphibious buoyancy, range, and stability	Several different attributes make up the amphibious context model but the main results are reserve buoyancy, range and stability. An excel spreadsheet was put together to contain all of these results based on inputs of a vehicle's basic dimensions and weight information. An example vehicle was chosen in order to prove out the calculations; this example was created using Pro/engineer and basic information from a public domain website, Hydratrek (www.hydratrek.com).
Q FMU usage	Examples of context models within Functional Mockup Units

In summary, appendices A, B, C, H, I, J, K describe the novel approaches that we have devised to create and verify models (steps **2****3****4** in **Figure 20** below). Appendices D and E describe the approach to organizing and maintaining models using an ontology (step **5**). Appendices F, G, L, M, N, O, P, and Q describe the application of the models to META design challenges (step **6**). The entire process is initiated in step **1** as requirements knowledge from the customer and design community flows into a completed project.

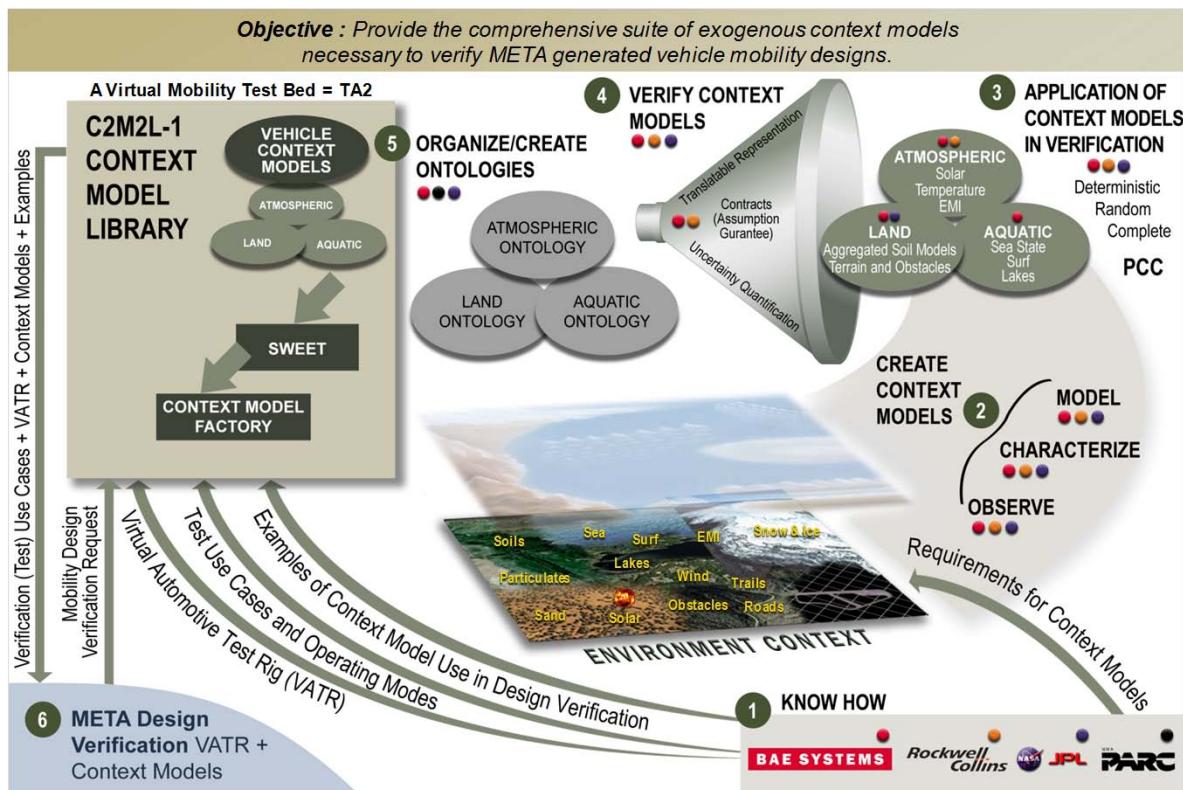


Figure 20: Overview of the Context Modeling Effort in relation to the Appendices

The delivery of models was organized along two paths. The Ontology path consisted of all semantic elements used to deliver the dynamic context server and OSCAR. The Model path consisted of all stand-alone or library elements used in traditional simulation integration. An adjunct to the latter was a demonstration of a simulated vehicle test-bed developed to show the suitability of the context models.

4. Conclusion

Our notable developments over the course of this project were in the areas of model development, developing the examples of context model use, and developing the ontologically-based web services to facilitate access to the context models and examples of their use.

Our important developments in the area of model development include:

1. Devising a spectral decomposition algorithm to model semi-Markov terrain features
2. Devising a terrain elevation correlation approach and applying it to the modeling of geospatial regions
3. Devising a pattern-based stochastic approach to model sample spaces
4. Using the principle of Maximum Entropy to synthesize model distributions with very concise formulations. These concise formulations can be easily inverted to provide sampling for Monte-Carlo simulations
5. Applying a depth-corrected wave height model to synthesize models of various bodies of water
6. Devising and diffusion model of oxidation and corrosion applying uncertainty quantification
7. Devising a novel heuristic method for Synthesis of a PSD Compatible Acceleration Time-History.

Our important developments in the area of formulating examples of context model use in design verification include:

1. Synthesis of over 20 examples, which demonstrate the use of each context model in vehicle simulations for the purposes of both verifying model correctness and for illustrating how the models can be used to verify the ability of a vehicle design to meet its performance requirements. Thus, each of the context models belonging to each of the 45 classes of context models listed in **Figure 6**, **Figure 7**, and **Figure 8** has been used in an example.
2. Formulation of standard approaches to interfacing context model units with external simulations via a number of development and interface platforms (The Functional Mockup Interface, MATLAB, C/C++, XML, and Java)
3. The synthesis of the Ecto-VATR, a concept synthesis tool that utilizes test benches that incorporate a subset of the context models we developed. The Ecto-VATR is a prototype of our vision for integrating context models into model-based development and verification processes.
4. Reuse of the context models on DOD development and research projects. In particular, we used context models obtained from the TOPS to perform ride quality and gun stabilization analysis for the U.S Army's Ground Combat Vehicle Program; We used the thermal network solver we devised as an example for the U.S Navy's Long Range Land Attack Projectile (LRLAP) Pallet analysis of the 40 degree functional test and slow cook off, and the prototype compositional verification tools developed by Rockwell Collins are the basis for Rockwell Collin's work on the DARPA's High-Assurance Cyber Military Systems program.

Our important developments in our efforts developing the ontologically-based web services to facilitate access to the context models and examples of their use include:

1. Organizing the models semantically using the SWEET ontology
2. Design and implementation of a semantic web server to facilitate model search and interaction
3. Design and implementation of a logic-based domain specific language for formal specification and model generation.
4. Design and implementation of a scheme for Phrase-based matching of requirements to context models

Finally, we completed and documented a thorough study to aid in Predicting Susceptibility via Cable/Connector Assemblies in a MIL-STD-461F Environment

4.1 Avenues for Further Research

The semantic organization will scale for other contexts. Among the missing contexts which were deemed out of scope for this effort include that of the human context and reliability contexts. For the human context, such factors as response times and unpredictable behaviors have a significant effect on vehicle design and performance objects. Although we only scratched the surface with models of corrosion, the impacts of demanding environments would fit in well as models of reliability, maintainability, and availability.

5. References

- [1] E. T. Jaynes and G. L. Bretthorst, *Probability theory: the logic of science*. Cambridge Univ Pr, 2003.
- [2] C. Becker and P. S. Els, “GEROTEK data from CSIR for C2M2L program.” .
- [3] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis Of Real-World Signals*. A K Peters, Ltd., 2010.
- [4] J. E. Alexander, “Synthesis of a PSD Compatible Acceleration Time-History,” *Submitted to Shock & Vibration Symposium (November 5-9, 2012)*.
- [5] P. R. Pukite, S. Bankes, and D. Challou, “stochastic_analysis.docx.”
- [6] P. R. Pukite, S. Bankes, and D. Challou, “terrain_characterization.docx.”
- [7] P. R. Pukite, S. Bankes, and D. Challou, “diffusive_growth.docx.”
- [8] D. Mumford, “The dawning of the age of stochasticity,” *Mathematics: Frontiers and Perspectives*, pp. 197–218, 2000.
- [9] Automotive Directorate, “Test Operations Procedure (TOP) 1-1-010 Vehicle Test Course Severity (Surface Roughness),” ATEC, Aberdeen Proving Ground, MD, Final TOP 1-1-010, 2006.
- [10] Army, “Research, Development, Test and Evaluation of Materiel for Extreme Climatic Conditions,” APD, Headquarters, Dept of the Army, Army Regulation 70-38, Sep. 1979.
- [11] A. Steinwoff and W. H. Connon III, “Limitations of the Fourier transform for describing test course profiles,” *Sound and Vibration*, vol. 39, no. 2, pp. 12–17, 2005.
- [12] S. Bankes, D. Challou, T. Haynes, H. Holloway, P. Pukite, J. Tierno, and C. Wentland, “META Adaptive, Reflective, Robust Workflow (ARRoW),” BAE Systems, Final Report TR-2742, 2011.
- [13] Wikipedia, “Functional Mock-up Interface - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Functional_Mock-up_Interface. [Accessed: 11-Feb-2013].
- [14] D. S. Warren, T. L. Swift, T. Vidrevich, I. Ramakrishnan, L. R. Pokorny, A. Beggs, C. Rued, M. Epstein, H. Singh, and H. Davulcu, “Methods for determining the similarity of content and structuring unstructured content from heterogeneous sources.” Google Patents, 02-Jun-2009.

Stochastic Analysis for Context Modeling

Paul Pukite, Steve Bankes, Dan Challou
BAE Systems

Abstract: Models of the physical environment play an important role in supporting Model Based Engineering (MBE). This paper describes how fundamental principles of thermodynamics and statistical physics can be applied to create compact parameterized models capable of statistically capturing the patterns exhibited in wide range of environmental contexts. Such models will allow more efficient and systematic assessment of the strengths and weaknesses of candidate designs than is possible using benchmark datasets or test tracks, and this can play an important role in using computer simulation to produce better designs of complex cyber-physical systems more quickly, affordably, and reliably. This is the first in a series of papers on the statistical description of natural processes. Subsequent papers in this series will treat specific examples in greater detail than provided in this overview.

Preface

Models of natural and man-made environmental contexts are important for a variety of purposes, in particular to support computational assessment of the performance of candidate designs for engineered systems. Vehicles perform on roads or other terrain, and both mobile and static engineered structures must contend with wind, waves, precipitation, and corrosion. Much of the value of engineered systems comes from their performance in the context of the environments in which they operate. Consequently, reliable assessment of candidate engineering designs requires models of potential operational contexts that accurately capture both the variability and overarching patterns presented by these environments.

Without computer simulation, candidate designs must be physically tested on sample deployed environments, for example test tracks. One aspiration for Model Based Engineering (MBE) is to achieve cost and other savings by replacing physical testing with testing through simulation. This requires validated models of the context environment that perform at least as well as a physical test in revealing design problems. However, simulation based testing can provide value beyond the elimination of physical testing. A significant cause of design failure often relates to providing an insufficient safety margin to deleterious environmental factors and contextual variables, such as an exceedingly rough terrain or corrosive atmospheric elements. Environmental context models that parametrically span the range of possible environmental challenges can provide a means for establishing the level of challenge at which a design will fail, generating considerably more information than produced by benchmark tests.

Environmental models that can replicate data derived from a wide range of actual environments are highly valuable, both through potential reuse across many design activities, and by supporting the creation of more robust and adaptable systems. Many of the patterns observable in contextual environments stem from fundamental processes derived from statistical mechanics and thermodynamics. Consequently, significant insight can be gained by approaching the patterns from first-principles physics. A combination of data and physical theory can result in models that extrapolate and accurately infer behavior outside of the data sets used for modeling. This approach results in models that are much more reliable for prediction and verification tasks than can be achieved through generic statistical modeling uninformed by physical considerations.

This paper introduces a set of research results relating to stochastic modeling of environmental contexts. We first provide some background to probability theory and for formulating the building blocks for stochastic analysis — in particular, that of *probability density functions* and Markov and semi-Markov processes. These building blocks comprise the core fundamentals of stochastic modeling, whereby we can reason about probabilities, sampling, and uncertainty. The more sophisticated techniques of autocorrelation and power spectrum densities are elaborated in an associated report on terrain

characterization, and models of dispersive growth and uncertainty propagation are described within an associated report on diffusion process context models.

Background: Environmental models for cyber-physical systems

Two criteria are central in crafting probabilistic models for observed behavior: (1) the importance or impact of observed events and (2) the associated likelihood of the event. For MBE, context models are needed for phenomena that play a significant role either through their frequency of occurrence or the severity of their impact. Context models portray the distribution of a metric of the phenomenon of interest. Metrics must be measurable or countable and could involve extensive variables such as volumes or consist of ratios such as rate (i.e. volume over time).

Table 1 : The use of probabilities can describe high likelihood and high impact events.

High Likelihood		High Impact
Terrain	RMS roughness	Steep slopes
Rainfall	Humid climates	Heavy downpours
Wind	Prevailing winds	Gusts
Waves	Chop and swell	Rogue waves
Particulates	Aerosols	High density volcanic dust

The fundamental building block in the creation of context models is the use of probability density functions (PDF) to model sample spaces. These facilitate the characterization and modeling of natural phenomena that are prevalent in human environmental contexts, including distribution of terrain slopes, wind velocities, rainfall amounts, etc. These can both model the high likelihood events through sampling of the meat of the distribution curves, but also provide for the rare cases through the concept of exceedance probabilities [1]. Models that can be expressed as fairly simple analytical forms will be more broadly useful. For disordered systems and data containing uncertainty, techniques such as the *maximum entropy principle* (MaxEnt)[2] and *superstatistics*[3] will be applied; these often have a more formal basis than the heuristic fractal models[4] often employed.

To discover patterns in data, data analysis techniques such as rank histogram plots are useful. A set of data, binned according to frequency of occurrence for the parameter of interest reveals most of the structure of the probability density. These views can be manipulated or marginalized against conditional or joint probabilities.

One of the significant observations that one can make about typical environmental parameters is in the extent of their randomness. On occasion an environmental parameter, such as temperature, can exist within a narrow range of values and thus become well-suited for a normal Gaussian distribution model, but more commonly, skewed (i.e. asymmetric) and fat-tail distributions are much more applicable. In these cases, the data along with some physical reasoning will direct the modeler away from a normal distribution toward a higher variance distribution.

Environmental context modeling relies on knowledge of exogenous behavior – that behavior that exists outside the confines of the vehicle or other engineered system we are designing. Any behavior that we have little control over needs to be regarded as uncertain, and will in general require stochastic models. Recent advances in our understanding of stochastic phenomena have benefited greatly from the availability of data from a variety of sources. In the past, modeling of physical behavior has often been hampered by the lack of sufficient statistics to substantiate the original formulation. In combining stochastics and information elements for modeling, we can incorporate probability and information theory (Jaynes[2], Shannon[5]), pattern theory (Mumford[6], Grenander[7]), fat-tail statistics (Mandelbrot[8],

Taleb[9], Sornette[10]), and superstatistics and complexity theory (Beck[3], Gell-Mann[11]) and then apply these contemporary ideas to the characterization of environmental contexts.

Probability theory as advanced by E.T. Jaynes[2] suggests using probability as an extended logic, and we should consider inference and plausible reasoning under various levels of uncertainty. The key idea of Jaynes is to meld Shannon's information theory concept of entropy together with the statistical mechanics definition of entropy. Many important and non-trivial applications exist where Jaynes' maximum entropy principle is the only tool we may need, as it describes the minimal application of prior knowledge when appropriate — often a mean value is all that is required.

Pattern theory as advanced by Mumford[6] and Grenander[7] seeks to identify the hidden variables of a data set, characterizing real world observations as patterns. The approach uses the observed patterns to infer information about the unobservable factors, formulating prior distributions for those too complex or difficult to observe. If we can determine efficient inferential models for the observed variables by modeling observations as partly stochastic and partly deterministic, and apply the randomness and variability of these distributions along with considering their natural structures and constraints (i.e. symmetries, independences of parts, and marginals on key statistics) we can create classes of stochastic models by applying transformations to patterns. We can then synthesize (sample) from the models, and the stochastic factors affecting an observation exhibit strong conditional independence, making it easily decomposable. We will see this approach demonstrated when we consider terrain characterization.

The analysis of fat-tail statistics as advanced by Sornette[10] and Taleb[9] has shown promise for the prediction of crises and extreme events in complex systems and risk management, both for social[12] and natural systems. The general theory encompasses scale-free properties, fractals, and power-laws, and provides an alternative to normal or Gaussian statistics. This introduces black and gray swan terminology and the idea of rare dragon-kings which relates to extreme value analysis (EVA)[13] , and evidenced via the scarcity of 100-year events.

Considerations of complexity theory as advanced by Gell-Mann[11] leads to the idea that : “*when defining complexity, it is always necessary to specify a level of detail up to which the system is described, with finer details being ignored*”. Seemingly complex representations can often be represented by rather concise descriptions and we can apply concepts such as dispersion and coarse graining to simplify the complexity. One such idea is that of superstatistics[3], which ties in closely to the ideas of maximum entropy [14]. The essential approach here is to admit that randomness can exist on different scales and by combining these scales, the underlying real-world statistical distributions are revealed.

The awareness that we can indeed use probability to characterize larger scale phenomena has often been fought tooth-and-nail by opposing schools of thought. For example, Mumford[15] describes how classical statisticians opposed contextual Bayesian modeling when they claimed that “*this approach denies that statistical inference can have anything to do with real thought because real-life situations are always buried in contextual variables and cannot be repeated.*” In this case, the contextual variables appear to get in the way of our understanding of the desired effect, whereas they should become part of the understanding of the *system*: i.e. the context of the vehicle within the environment. In reality, we can and have created very useful models by incorporating prior contextual knowledge to infer possible and potential behaviors. The key is that many of the contextual variables are governed by properties of nature that repeatedly occur under conditions of the thermodynamic arrow of time, which always leads to greater amounts of entropy. In this sense, nature can perhaps be more predictable than we think, and at worst, we can expect that it remain predictable in its unpredictability, and thus we can make progress by applying a stochastic characterization to the empirical observations.

Maximum Entropy Principle Modeling

To apply context modeling effectively one has to approach the environmental domains pragmatically. We want to find solutions with the minimal amount of complexity that conversely generates the most general benefit. A model with a very detailed representation will typically apply only in specific cases and be of little use. So we first seek the simplest possible approach and find out if that has general applicability.

Fortunately, nature helps us out with the bookkeeping, through its implicit use of information theory, or as it is known in the physical sciences world – statistical mechanics. Thermodynamics and the essential notion of entropy both derive from statistical physics. The essential idea behind information theory is to try to describe an observed phenomenon with the least amount of words as possible.

A simple example of connectivity patterns drawn from Gell-Mann[11] demonstrates this point. Consider Figure 1 below, a series of networks showing increasing levels of seeming complexity. Graph **A** seems the least complex as it has no connections, while Graph **F** seemingly shows the most amount of complexity as it has all nodes interconnected.

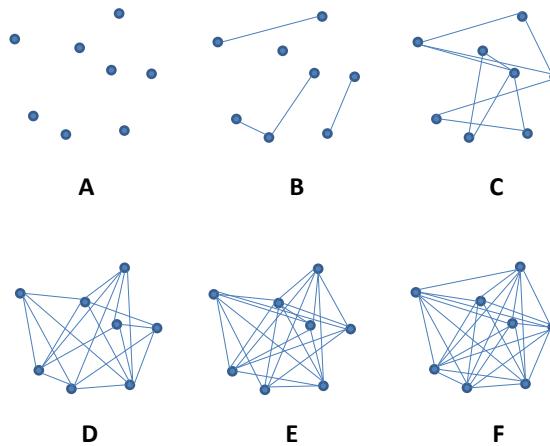


Figure 1: Gell-Mann's connectivity patterns. Increasing levels of complexity shown as an interconnected graph.

Using information theory, we can estimate the actual complexity by algorithmically describing each of the connectivity arrangements. Graph **A** can be described as “*no nodes connected*”, while Graph **F** can be described as “*all nodes connected*”. These therefore have very close to the same information content, because they can be reduced to an equally concise description. The graphs in between these end-points have higher complexity because they will require a more elaborate description.

In physics, this approach has long been applied to describe topics such as the ideal gas law, which stems from seminal ideas of statistical mechanics. Ensembles of particles, though complex at a micro-scale, can be described more concisely at the macro-scale, with the connecting thread elegantly applied from entropy and information theory.

These ideas can be applied to other natural phenomena. The key idea that we propose to apply is that of the Maximum Entropy Principle (MaxEnt)[2]. The principle of maximum entropy states that those configurations described by the least amount of constraints and moments (such as mean, variance, and higher-order moments) will tend toward probability density functions that maximize entropy. The MaxEnt solution is essentially to maximize $S = \text{entropy}$ subject to constraints of probability, $f(p_i)$.

$$\max_{f(p_i)} S = - \sum p_i \cdot \ln p_i \quad (1)$$

In practice, MaxEnt gets routinely applied, even without the knowledge of the practitioners. For example, with the assumption of only a mean and a standard deviation in a set of empirical observations, the conventional approach is to apply the standard Gaussian or normal statistical distribution. That indeed coincides with the MaxEnt derived probability distribution for that particular set of moments. The canonical example of this is a manufactured component with a tight tolerance; this would routinely show a Normal distribution in the measurements of the produced parts. (A description of this example for an RLC¹ circuit using ratio distributions can be found in a report from DARPA's Adaptive Vehicle Make META program[16]).

Another simple example is that of a dye mixing in a glass of water. The eventual spatial distribution of the dye in the volume will approach a uniform level over time. This is straightforwardly modeled as a MaxEnt calculation given the boundary conditions of the volume. There is no preference for the dye occupying one subvolume over another so that we can only apply the known physical constraints. The barometric pressure law with altitude is one of the simplest examples of adding an energy constraint to Jaynes' formulation; and this gives the characteristic damped exponential decrease of pressure with altitude.

In addition, countless other phenomena, especially those showing great amounts of disorder, do not follow either a Normal distribution, or the uniform distribution or exponential just described. This includes many of the so-called “fat-tail” distributions, popularized by recent events[9]. Further, some of the fat-tails come about not from any intrinsic physical property, but from a derived property of the measure space (such as the ratio distribution mentioned parenthetically above).

Uncertainty Quantification

In addition to the natural random variation in the values of a particular environmental observable (**Type 1**: aleatoric uncertainties), our limited abilities to quantify the numbers can lead to a further uncertainty spread (**Type 2**: epistemic or systemic uncertainties). Whether the epistemic uncertainties allow us to capture the underlying randomness depends on the strength of the characterization and modeling. For example, we can have uncertainty in the actual measurements due to calibration or precision errors, uncertainty due to counting statistics, and uncertainty in the applicability of a model.

If the observable has a wide dynamic range and is readily quantified, the uncertainty in measurement is often absorbed in the natural randomness and has minimal impact on the data characterization. For counting statistics, uncertainty is minimized by drawing from a sufficient sample size that represents the complete dynamic range of the observable. By accumulating data over time, using approaches such as Bayes rule via additional prior knowledge, the additional evidence will reduce this uncertainty.

The applicability of the model for describing a natural phenomenon is referred to as the *likelihood* of the model and of its parameters. The certainty or confidence we have in a particular model is ultimately best gauged by comparison to an alternative model in which we can apply standard inference metrics such as conditional entropy, maximum likelihood, log-likelihood, and information entropy criteria such as Akaike Information Criterion (AIC) or Bayes Information Criterion (BIC) [17]. The information criteria techniques are valuable because they penalize models that contain too many fitting parameters. Models based on first-principles with minimal parameterization (such as those derived from the Maximum

¹ RLC = resistive-inductive-capacitive. A configuration of RLC components will show a resonant frequency sensitive to the choice of parameter values.

Entropy Principle) will always score higher than, for example, a naïve high-order polynomial fit that contains many adjustable parameters.

The goal of context modeling is to converge to only **Type 1** uncertainty, where we can apply what we consider the most likely model and use its probability distribution function to provide Monte Carlo or importance sampling for design verification. In previous work[16] we applied propagation of uncertainty in combination with various physical and information-entropy-based models to arrive at estimates of a probabilistic certificate of correctness (PCC) for a given design. This included exogenous artificial effects and sets of metrics dealing specifically with exogenous variables, those variables whose value is determined outside the model in which it is used:

- Manufacturing variance (the RLC example)
- Semantic network links
- Travel dispersion
- Wireless signal latency
- Human reaction times

This approach is based upon application of the maximum entropy formulation and a careful consideration of the measure space². By applying minimal information to stochastic models of various behaviors, that we can infer the essential probability distributions, and therefore concisely model contextual behaviors suitable for conversion (in reverse) to uncertainty bounds — necessary for tasks such as verifying vehicle or system environmental suitability. This choice results in a modeling approach for disordered systems and data containing uncertainty, using techniques such as the maximum entropy principle and superstatistics, has a more formal basis than the heuristic or fractal models conventionally used to empirically fit data.

In the following sections, we describe the basics of this approach. More sophisticated analyses can be developed on this foundation, for example:

- Growth curves
- Generalized correlation functions in the real-space domain
- Combining correlation functions with spectral representations, both in the spatial and temporal domains of system context.

These analyses will be covered in subsequent volumes (see Appendices B, C, D).

Characterization to Modeling

Fundamentally, context modeling involves applying the scientific method to describe the physical world. Figure 2 below shows the process going from (1) initial observations, to (2) characterization, and then to (3) modeling.

² See .This approach is not as common as one would think, considering the amount of detailed analysis undertaken with conventional statistical techniques (by non-domain experts who have a knowledge of statistics). We want to occupy the pragmatic middle ground and borrow insight from both physics and statistical camps.

*Figure 2: Process of Modeling*

Stage 1 represents gathering the initial data from observations and measurements.

Stage 2 transforms the raw data to usable information —characterization imparts deeper categorical qualities and meaning to the data. In characterizing the initial data, we may not yet know the process or theory explaining how the data was generated, but inferences can be made from information made available, through for example a graphical form or as a lookup table. Such characterization of categorized data will allow the differentiation from other sets of data. For example, by charting the frequency of slopes in an environmental terrain data set, we can tell if one region is on average steeper than another region.

Stage 3 transforms the information to real knowledge via scientific modeling. By modeling we can bring a much deeper understanding to the information at hand than someone who only has lookup tables. To provide one benefit, we can extrapolate to regions outside the scope of our data ranges. Then, we can not only say how much steeper one area is than another but we can explain the reason and thus have additional knowledge to infer from. In other words, we can extend our reasoning capability.

We will treat these models in as concise a formulation as possible to avoid undue complexity, and we apply physical first principles to justify their description

Context Modeling of Environmental Domains

To better convey how this final modeling step is performed, we now provide examples applying maximum entropy principles and superstatistics to several practical examples: (1) wind speed, (2) terrain slopes, (3) rainfall intensity, (4) wave crest heights, and (5) electro-magnetic signal clutter. These all share a common foundation, as each draws from energy from the environment to give a range in intensity of some measure. In each case, the intensity is a readily measured observable, with the general trend of higher intensity values occurring less frequently than the low intensity values. The zero energy state is indeed the most common in all these measures.

An initial set of questions to consider from a context modeling perspective is what general characterization can we make from these observables, and then what universal modeling approaches can we apply?

Drawing from the perspective of maximum entropy, we first consider that the probability of a high energy configuration is typically treated as a variation of a Boltzmann factor:

$$e^{-E/E_A}$$

where E_A is the activation energy for occurrence. In this approach, the likelihood of higher energies becomes exponentially damped, scaled by the activation energy constant. The connection to maximum entropy is that we can treat this factor as a probability and then apply it as a probability density function of the measure of interest:

$$p(E)dE = \frac{1}{E_A} e^{-E/E_A} dE$$

The choice of the exponential in terms of MaxEnt is that it is the least biased estimator considering that E_A is the **average** energy of the configuration. If we happened to know the variance of the ensemble configuration, this would lead to a normal, Gaussian PDF. Yet, since we in general lack this knowledge, we need to rely on the least amount of information available, and this is the exponential, or Boltzmann factor.

Next we consider the application of the activation energy for the individual cases. For wind distributions, the kinetic energy is related to the square of the wind speed, v . This turns into the Rayleigh distribution.

$$\begin{aligned} E &\sim kv^2 \\ dE &= 2kdv \\ p(v) &\sim 2kv e^{-kv^2} \end{aligned}$$

Electromagnetic signal clutter follows a similar derivation as the energy is the square of the amplitude of the electric signal.

For terrain distributions, we can to first order suggest that a potential energy is directly proportional to the terrain slope.

$$p(s) \sim e^{-ks}$$

For rainfall intensity, we consider the potential energy associated with a volume of water under gravitational forces. The larger the volume, the greater the encapsulated energy, which gets released scaled to the rate intensity of the rainfall.

Finally, for aquatic wave crests, the energy of the waves is proportional to square of the crest height. This works only to some level, as shoaling and non-linear fluid mechanics can prevent or attenuate taller waves.

Now consider that in each of these cases, the value of the activation energy can vary depending on regional or environmental conditions. For the case of rainfall, the intensity of the rainfall can be predicated on other conditions besides the volume of the water vapor alone. This leads to the idea of a super-statistical distribution. Here we not only apply the exponential PDF to the measures of interest, but we grant that probability an extra layer of uncertainty. That uncertainty would commonly apply to the value of E_A or to some other constant of proportionality.

Based upon activation energy proportionality or some other variant measure, different kinds of statistical distributions can be derived. For example, by considering variations in capacity and growth time, such phenomena as cloud sizes, lake sizes, and particulate sizes can be modeled. These become fat-tail distributions due to the weighting of the rate calculation, as a strong variant situated in the denominator of a stochastic ratio turns into a heavy weighting in the tail of a distribution.

The foregoing are exemplars of the general superstatistical approach we apply to natural context domains. The table below describes several of the practical examples. For many of these natural phenomena, empirical data sets can be used as samples from the observational space. By inferring information from a model that more accurately represents the underlying behavior than can a generic statistical distribution

more fundamental insights may be inferred. This has a number of benefits, including conciseness of representation and potentially better estimation of rare events.

Table 2: List of stochastic models

Stochastic Metric	Elements	Description	Data
Wind	PDF, ME, SS	Model of wind speeds	Bonneville Power Authority[18]
Rainfall	PDF, ME, SS	Model of rainfall amount	Hydrometeorology Lab University of Iowa[19]
Clutter	PDF, ME, SS	Model of EMI	
Clouds	PDF, ME	Model of cloud sizes	NASA Goddard[20]
Lakes	PDF, ME	Model of lake sizes	Global Lakes and Wetlands Database[21]
Particles	PDF, ME	Model of particle sizes	NASA JPL [22]
Waves	PDF, ME	Model of crest heights	CDIP[23] and US Army Corps of Engineers[24]
Terrain slopes	PDF, ME, SS	Model of inclination	USGS DEM [25]

Examples

We now provide examples of almost predictable unpredictability that can arise in many natural environmental contexts, such as variability in terrain slopes, as well as in artificially man-made situations, such as a large highly-interconnected network [26]. These models can be applied in either analytic form or in a form suitable for Monte Carlo-type simulations.

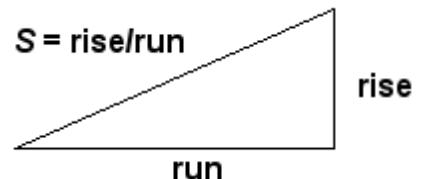


Slope modeling and Topography

The first example is of a longitudinal slope context distribution. The macroscopic slope model of USA terrain follows closely that described by the Maximum Entropy Principle (MaxEnt).

How do we model and thus characterize disorder in the earth's terrain? Can we actually understand the extreme variability we see? If we consider that immense forces cause upheaval in the crust then we can reason that the energy can also vary slope topography widely.

The process that transfers potential energy into kinetic energy to first order has to contain elements of randomness. To the huge internal forces within the earth, generating relief textures equates to a kind of Brownian motion in relative terms — over geological time, the terrain amounts to nothing more than inconsequential particles to the earth's powerful internal engine.



We take the terrain slope S as our random variable (defined as rise/run).

The initial premise is the higher the slope, the more energetic the terrain. Applying the Maximum Entropy Principle to a section of terrain, we can approximate the local variations as a MaxEnt conditional probability density function, where E is the local mean energy and c is a constant of proportionality. But we also assume that the mean E varies over a larger area that we are interested in, as in the superstatistical sense of applying a prior distribution, where k is another MaxEnt measure of our uncertainty in the energy spread over a larger area.

The final probability is an integral over the marginal distribution consisting of the conditional multiplied by the prior. This integrates as a modified BesselK function of the zero order, K_0 .

$$p(S) = \frac{2}{S_0} \cdot K_0(2\sqrt{S/S_0})$$

The average value of the terrain slope for this distribution is simply the value S_0 .

The validity of the model thus derived can be assessed by comparing to a large set of data. The digital elevation model (DEM) data for the 1 degree quadrangles (aka blocks/tiles) in the USA from the USGS web site was characterized. This consists of post data at approximately 90 meter intervals (i.e. a fixed value of run) at 1:250,000 scale for the lower 48 USA and some spillover into Canada. From individual DEM files, we calculate the slopes between adjacent posts yielding an average slope (rise/run) of 0.039, approximately a 4% grade or 2.2 degrees pitch. The characterization takes the absolute values of all slopes so that the average is not zero.

The cumulative plot of terrain slopes for all 5 billion calculated slope points appears on the following chart[26]. The cumulative probability distribution of the BesselK model is plotted with the calculated average slope as the single adjustable parameter.

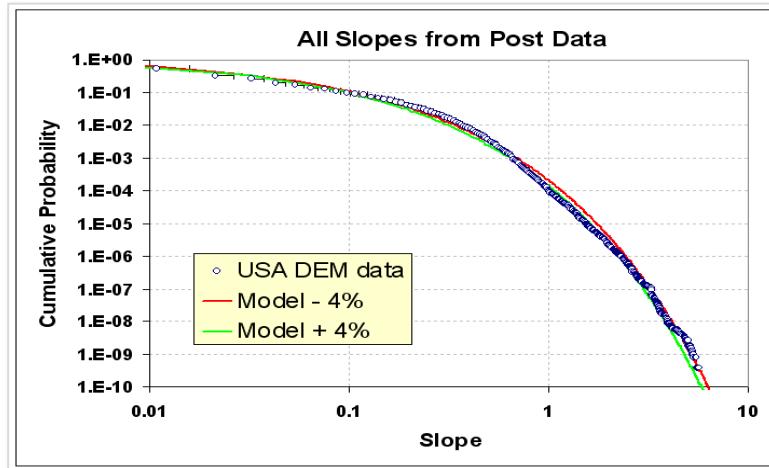


Figure 3: Longitudinal slope cumulative distribution function for the USA. The BesselK model with a small variation in S_0 (+/-4% about the average 0.037 rise/run) demonstrates sensitivity to the fit.

The good agreement occurs because random forces contribute to maximizing the entropy of the topography. Enough variability exists for the terrain to reach an ergodic limit in filling the energy-constrained state space. As supporting evidence, we can generate a distribution that maps well to the prior by estimating the average slope from the conditional PDF of each of the 922 quadrangle blocks and then plotting this aggregate data set as another histogram.

For context modeling, a library function is used to generate Monte Carlo sample draws for the BesselK model without requiring a probability inversion. The resulting algorithm turns out surprisingly simple. First, draw two independent random samples from a uniform [0.0 .. 1.0] interval, then apply the natural log to each, multiply them together, and then multiply by the BesselK S_0 scaling constant.

This random draw algorithm will give the following cumulative if done 5 billion times, which is the same sample size as the real USA DEM data sample (see Figure 4). The only statistical noise is at the 10^{-9} level, which is roughly the same as in the DEM data set.

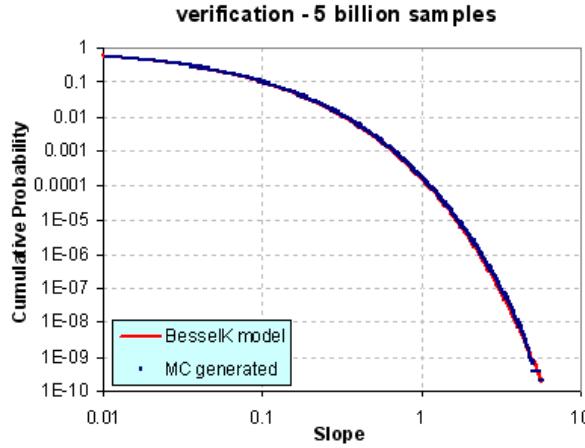


Figure 4: Drawing random samples to approximate slope distribution

Practically speaking, we see the variability in slopes expressed at the two different levels: the entire USA at the integrated (BesselK model) level and the aggregated regions at the localized (exponential prior) level. These remain consistent as they agree on the single adjustable parameter S_0 .

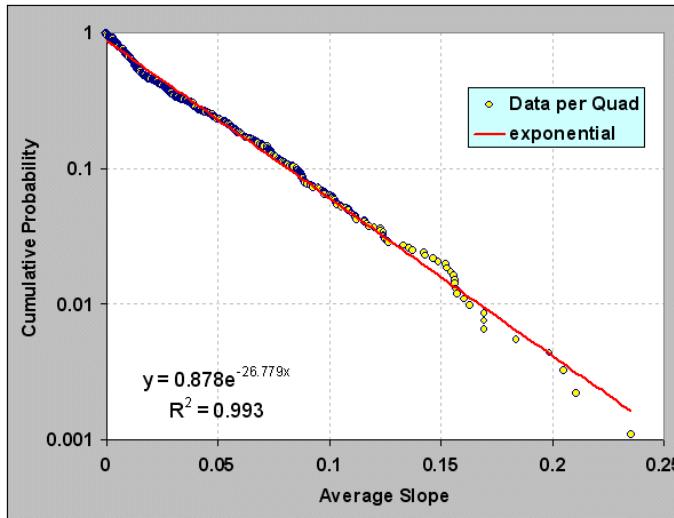


Figure 5: Generation of the prior distribution by taking the average slope of each of nearly 1000 quadrangles. The best fit generates a value of S_0 ($1/27=0.037$) close to that used in the previous figure.

The modeled distribution has many practical uses for analysis, including mobility studies and fuel efficiency planning (see Appendix F). Obviously, vehicles traveling up slopes use a significant amount of energy and a context modeler would need a model to base an analysis on without having to rely on the raw data by itself. Further, spatial correlations also exist and will prove useful as well. A survey of recent research gives no indication that others have discovered this rather simple model (See further Vico[27] Gagnon[28], Gonçalves[29], Guarnieri[30])



Wind Distribution

Wind velocities demonstrate a wide dynamic variability, ranging from calm to gale force. However intuitive the concept of “windiness”, we may often miss the underlying mathematical simplicity behind wind speed variability. The complexity of the earth’s climate and environment actually contributes to this simplicity as it generates more states for the system to exist within (see the Gell-Mann argument), which can also increase the likelihood of variability. With minimal knowledge as to the origin of the wind variance, we can apply the maximum entropy principle to its energy content.

The derivation of wind dispersion follows a few straightforward steps. We start with the premise that every location on Earth has a mean or average wind speed. This speed has a prevailing direction but assume that it can blow in any direction. Next we safely assume that the kinetic energy contained in the aggregate speed goes as the square of its velocity. If we assume only temporally-averaged mean wind energy and then relate the energy, E , as the square of the wind speed, v^2 , the resultant maximum entropy probability distribution matches the Rayleigh distribution.

$$p(v) = p(E) \cdot \frac{dE}{dv} = 2cv \cdot e^{-cv^2}$$

This comes about from the Newtonian kinetic energy law $\frac{1}{2}mv^2$ and it shows up empirically as the aeronautical drag law (i.e. wind resistance) which also goes as the square of the speed.³ Then apply the principle of maximum entropy to the possible states of energy that exist and come up with a probability density function that has no constraints other than a mean value (with negative speeds forbidden). In the equation above c is a constant and $1/c$ defines the mean energy (i.e. essentially acting as the Boltzmann activation energy). This describes a declining probability profile, with low energies much more probable than high energies. To convert to a wind dispersion PDF we substitute velocity for energy and simplify.

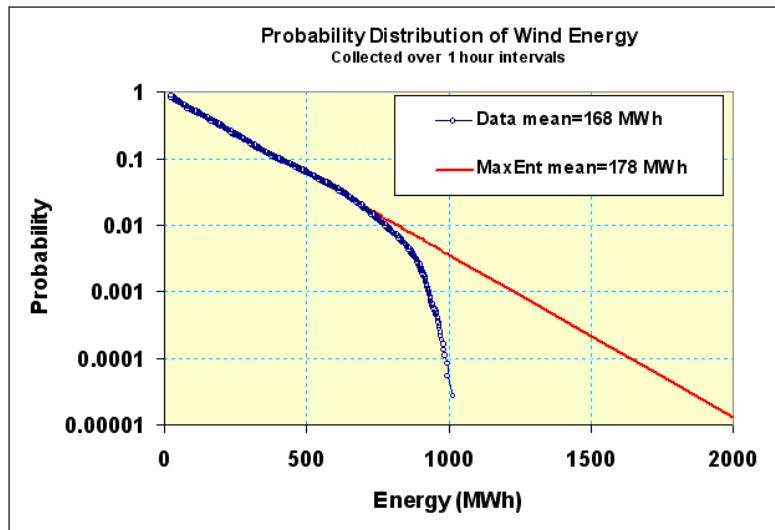


Figure 6: Wind distribution follows Rayleigh distribution closely. This data was collected from an archive of wind energy statistics collected in Ontario. The upper level cut-out is due to turbine governor regulation at high wind speeds.

³ Note that we can consider E as an energy or modified slightly as a power, since the energy is sustained over time

Figure 6 shows an empirically observed wind speed distribution, showing a peak away from zero wind speeds and a rapid decline of frequency at higher velocity. Heuristically, many scientists refer to the model as following a Rayleigh or Weibull distribution. The Rayleigh comes out as the simpler model because it derives from first principles and any deviation from the quadratic exponent works as a refinement. The first data set shown consisted of about 36,000 sequential hourly measurements in terms of energy (kilowatt-hours) for Ontario.

By adding more data to our knowledge on wind dispersion, we can observe how dispersion in wind speeds has a universal character. The second data set (Figure 7) comes from northwest Germany and consists of wind power collected at 15 minute intervals.

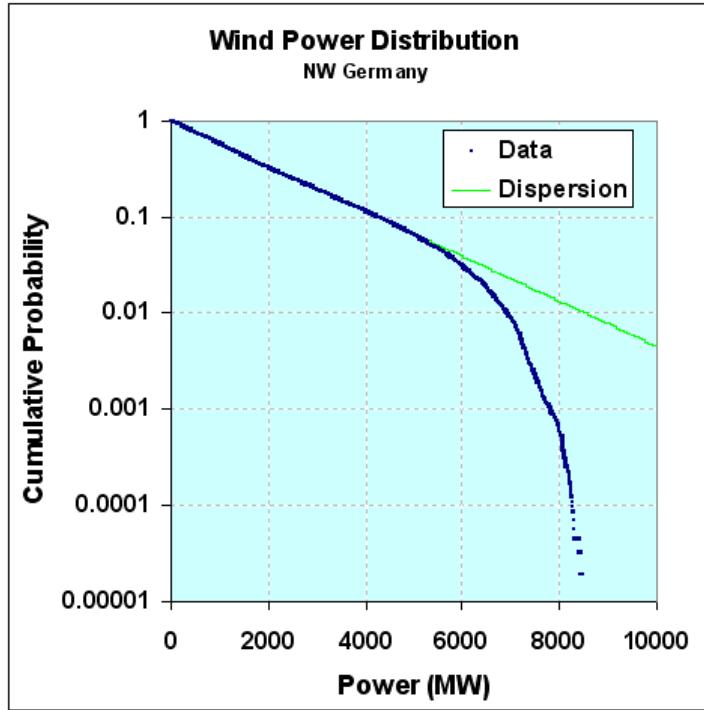


Figure 7: Wind variations for Germany. The curve has all the same characteristics as that for Ontario, demonstrating the universal behavior in wind speed variability.

Note that the same wind dispersion holds as for Ontario. Both curves display the same damped exponential probability distribution function for frequency of wind power (derived from wind speed). We also see the same qualitative cut-out above a certain power or wind energy level.

Adding More Variability. Since E_A can vary from region to region, we leave it as a conditional, and then set that as a maximal entropy estimator as well

$$p(E_i) = \alpha \cdot e^{-\alpha E_i}$$

then we integrate over the conditional's range according to standard practice and arrive at a cumulative.

$$P(E) = \int_0^{\infty} P(E|E_i)p(E_i)dE_i$$

This results in a simple lookup in your favorite comprehensive table of cataloged integration formulas, which leads to the following solution:

$$P(E) = 2 \sqrt{\frac{E}{\pi}} K_1 \left(2 \sqrt{\frac{E}{\pi}} \right)$$

where K_1 is the modified BesselK function of the second kind, in this case of order 1, which is found in any spreadsheet program (such as Excel). Note that this is the same function that we used for the distribution of terrain slopes. The order 1 is the variant used for the *cumulative* distribution function

We tested this formulation against wind data from [Bonneville Power Administration](#), which has over 20 meteorological stations set up around northern Oregon. The download consisted of over 2.5 million data points collected at 5 minute intervals, archived over the span of a little less than 2 years.

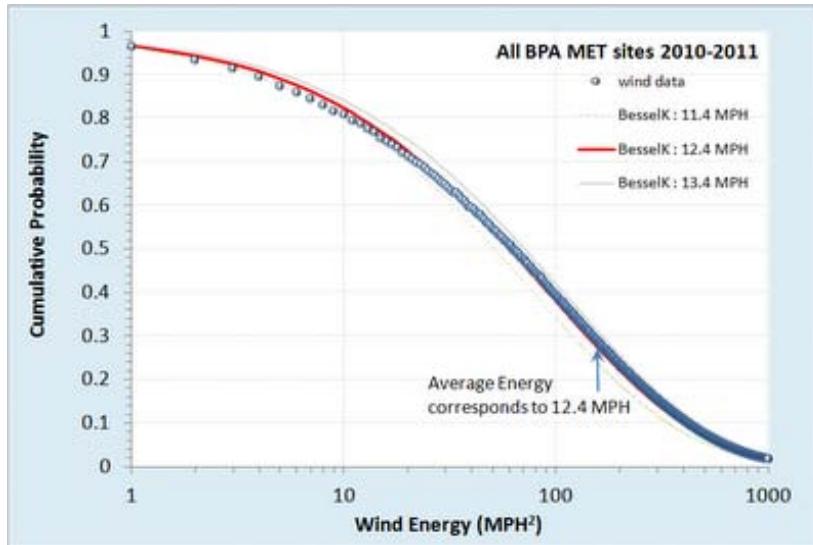


Figure 8: Cumulative distribution function of wind energies from Bonneville with model fit.

For the fit of this curve, the average energy was derived directly by computing the mean over the entire set of data separately. This corresponded to a value of 12.4 MPH, and placed a pair of positive and negative tolerances to give an idea of the sensitivity of the fit.

As this is a single parameter model, the only leeway we have is in shifting the curve horizontally along the energy axis, and since this is locked by an average, the fit becomes essentially automatic with no room for tweaking and little for argument. The probabilities are automatically normalized.

Figure 9 shows the log-log plot, which reveals a departure at high wind speeds. This shows that excessive gale force winds (greater than 60 MPH) did not occur over the extended region during the span of two years data collection.

Wind dispersion analysis has obvious applications for context modeling. Fuel efficiency is impacted by aerodynamics and drag goes up as the square of the wind speed. Vehicle cooling also is impacted by convection due to local winds. Applying this approach for context modeling has the benefit of allowing simple sampled data generation for verification and PCC bounding, similar to that applied for terrain slopes.

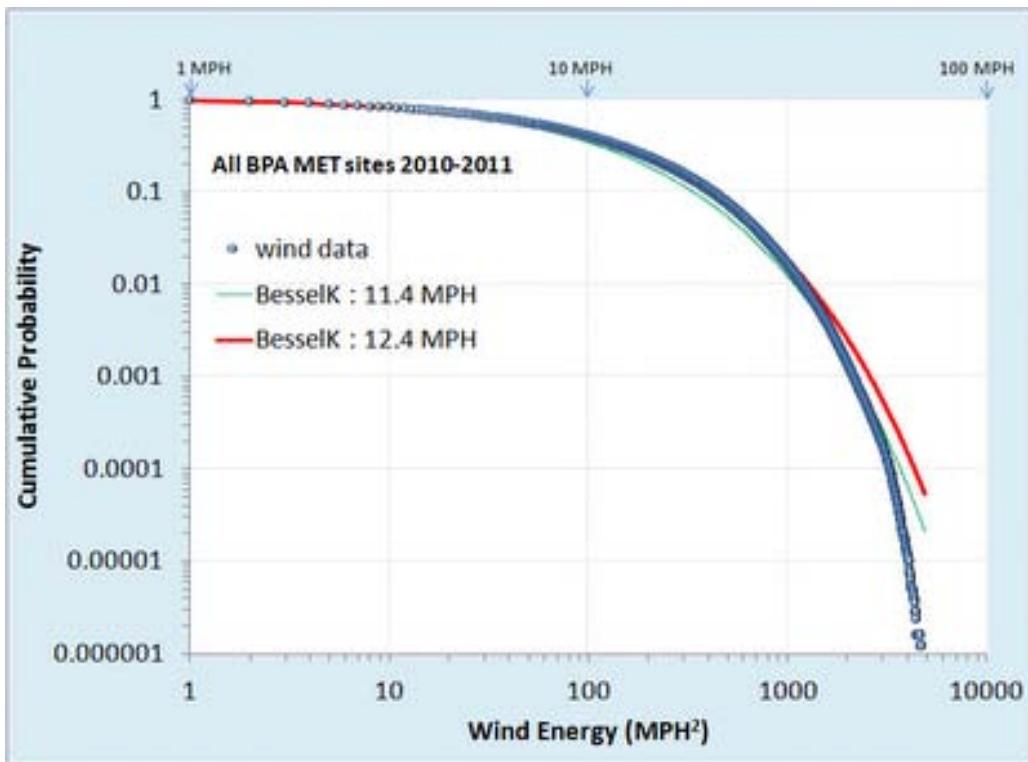


Figure 9: Cumulative distribution function of wind energies on a log-log plot.

The previous maximum entropy derivation assumed only a known mean of wind energy levels (measured as power integrated over a fixed time period). From this simple formulation, one can estimate or extrapolate a wind speed probability. Knowing the probability of wind speed, one can also perform all kinds of interesting extrapolations — for example, we can project the likelihood of how long it would take to accumulate a certain level of energy[31].



Aquatic Waves

Ocean waves exist in as disordered and unpredictable state as the wind. We may not always notice this as the scale of waves is smaller and often takes the form of a regular lapping of swells. In practice, the wind and wave energy distributions relate via similar maximum entropy disorder considerations. The following derivation assumes a deep enough water such that the wave troughs do not touch bottom

First, we make a maximum entropy estimation of the energy of a one-dimensional propagating wave driven by a prevailing wind direction. The mean energy of the wave is related to the wave height by the square of the height, H . This makes sense because a taller wave needs a broader base to support that height, leading to a scaled pseudo-triangular shape of a *gravity* wave, as shown in Figure 10 below.

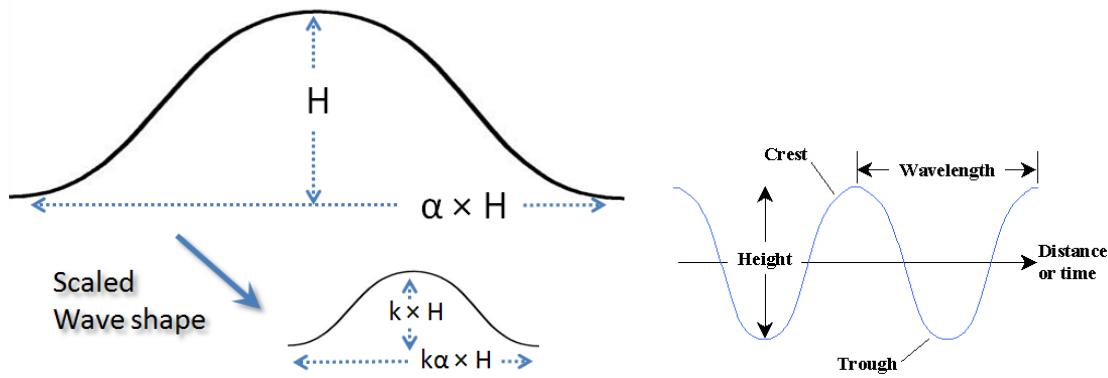


Figure 10: Total energy in a directed wave goes as the square of the height, and the macroscopic fluid properties suggest that it scales to size. This leads to a dispersive form for the wave size distribution

Since the area of such a scaled triangle goes as H^2 , the MaxEnt cumulative probability is:

$$P(H) = e^{-\alpha H^2}$$

where α is related to the mean energy of an ensemble of waves. This relationship is empirically observed from measurements of ocean wave heights over a sufficient time period. This looks at height alone.

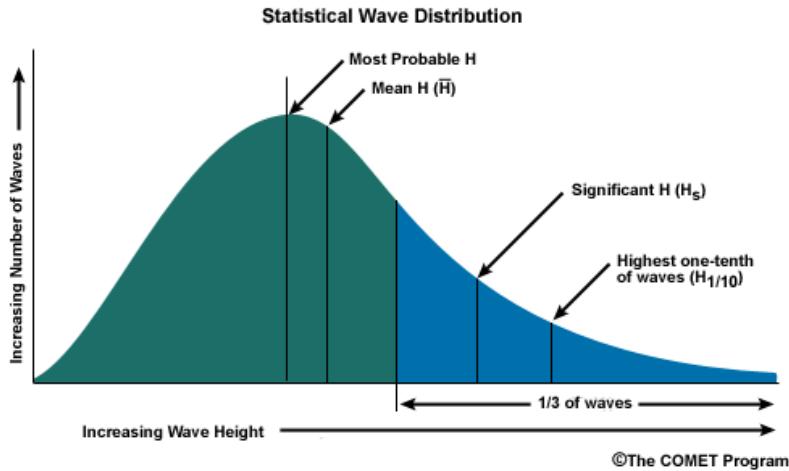


Figure 11: Statistical Wave Distribution (source: NOAA UCAR COMET Program)

Long-lived ocean and lake measuring stations have recorded historical records of wave crest data over the span of decades. From the US Army Corps of Engineer's Wave Information Studies project[24], the following figure collects chop and swell data from over several hundred million data points on Lake Michigan:

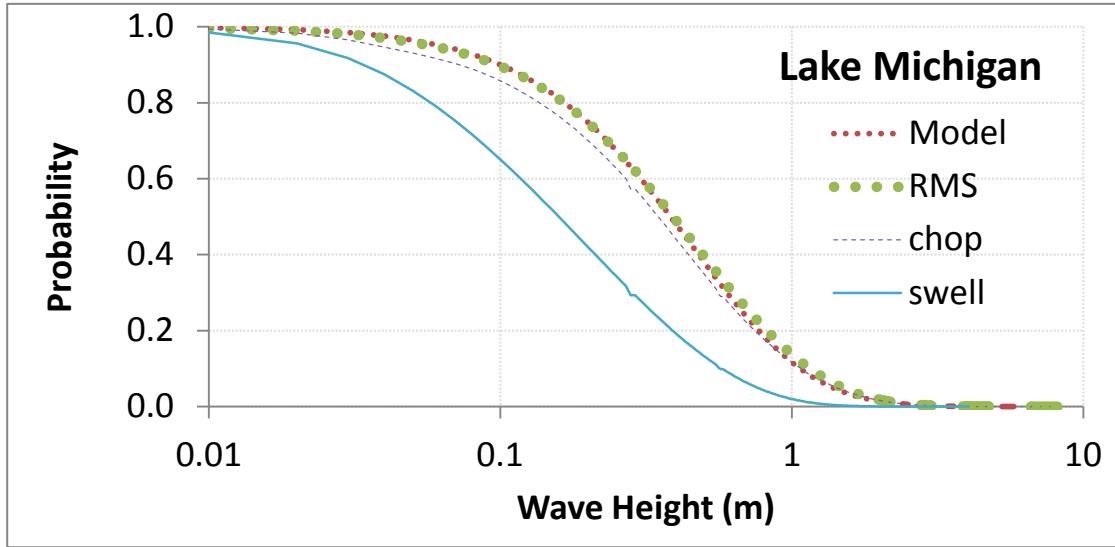


Figure 12: Lake Michigan wave height PDF on log-linear scale.

The probability density function for significant wave height empirically follows a modified Bessel function of the second kind. This essentially derives from a maximum entropy distribution for a wave height with a mean energy value, which is then dispersed again by maximizing the uncertainty in the energy. With precisely the same pattern that we derive the Bessel function from a range in wind speed values, we can derive the PDF for wave energy.

The Bessel fit works well for small wave heights but then starts to diverge when the wave height starts to exceed a critical level. This critical level is essentially the cresting limit of a wave given the average depth of the water. At this point we can apply an empirical correction factor first proposed by Jahns and Wheeler[32] and further analyzed by Haring [33]. This factor is essentially a 2nd-order polynomial which gradually suppresses the wave height from exceeding the critical cresting value.

$$1 - 4.37 \frac{H}{d} \left(0.57 - \frac{H}{d} \right)$$

The rationale for the factor arises from the remote likelihood of a wave height from exceeding the average water depth (d) in a region.

Based on data collected from coastal waters of two large lakes, Superior and Michigan, and that along the eastern USA seaboard of the Atlantic Ocean, we can see the characteristic bend on the PDF at approximately 10 meter height.⁴

According to Figure 13, the same basic Jahns/Wheeler correction is applied across the bodies of water. Both Michigan and Superior use a Bessel function of order 1, while the Atlantic uses a Bessel of order 2, which is generated by assuming an uncertainty that is not a maximum entropy exponential in the mean, but a MaxEnt that is a gamma of order 2 (i.e. two exponentials convolved which reduces the variance in $\frac{1}{2}$)

⁴ The data from Lake Superior was not as extensive as the other two bodies of water

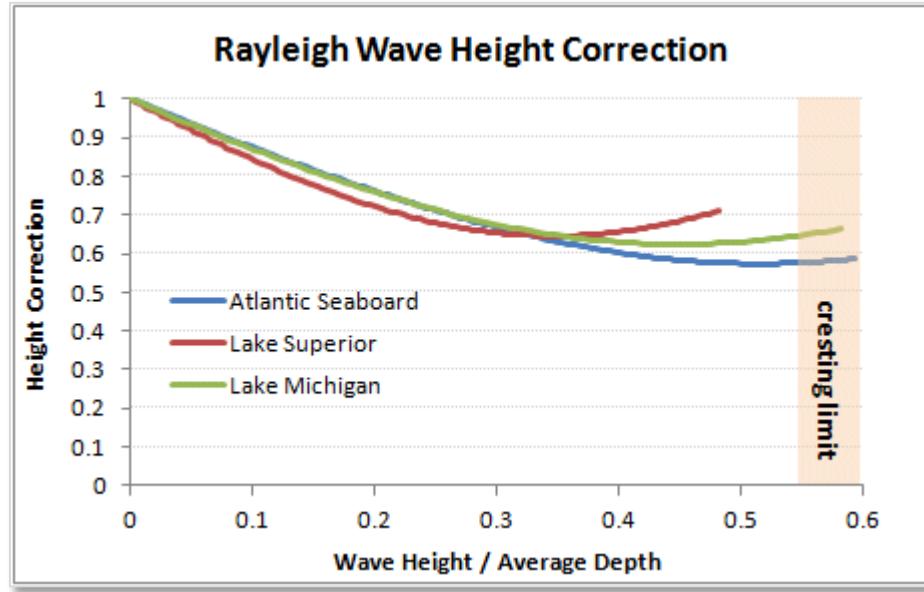


Figure 13: The Jahns/Wheeler correction applied to various bodies of water. The Lake Superior used coefficients closer to the heuristic J/W values (see Table 3) but Atlantic and Michigan appeared to asymptotically merge at the Nelson cresting limit of 0.55 for the wave height / average depth ratio.[34] [35].

Table 3: Heuristic Jahns/Wheeler correction used different coefficients to achieve the best fit.

	Atlantic Seaboard	Lake Superior	Lake Michigan	Haring
Inferred Average Depth	25.13 m	22.82 m	13.75 m	-
Haring Coefficient 1	2.52	4.86	3.03	4.37
Haring Coefficient 2	1.03	0.69	0.90	0.57

To evaluate the fit across the extreme values, Figure 14 shows the model profile on a log-log scale. The long tails are important for evaluating the probabilities of high sea-state values.

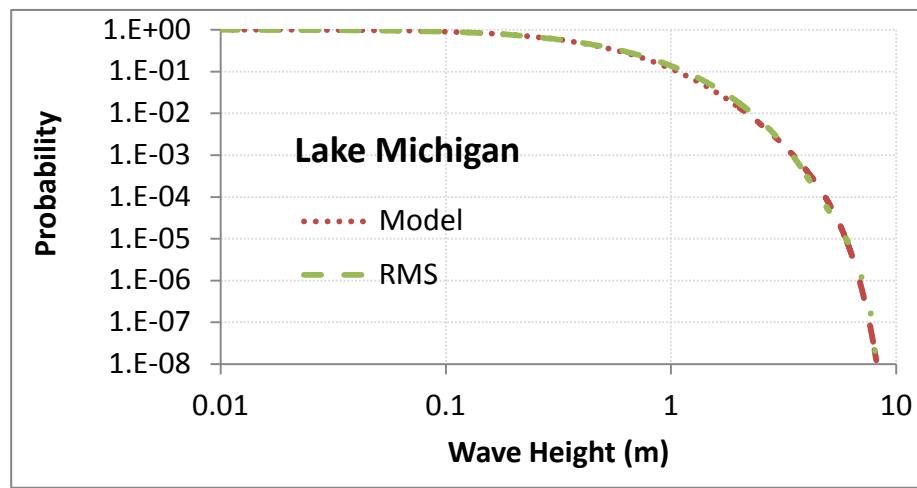


Figure 14: Lake Michigan log-log scaled version of wave height PDF

Lake Superior measurements (Figure 15) were sparse, yet the same profile is observed.

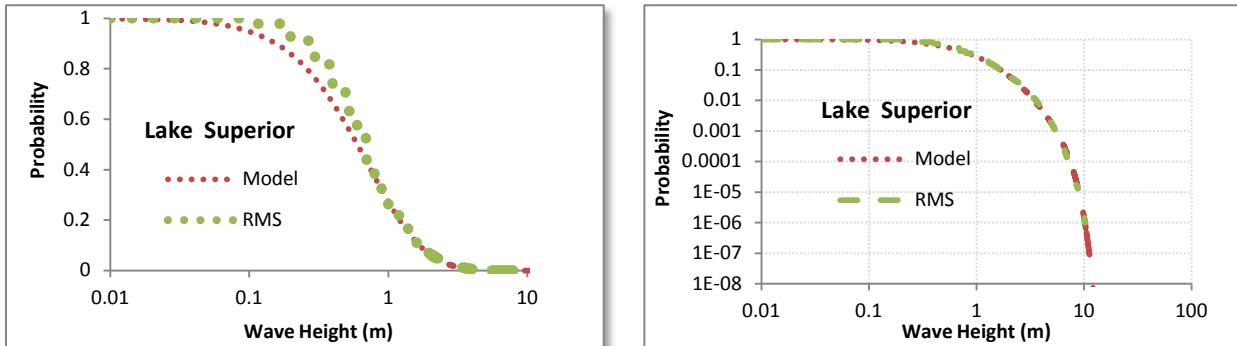


Figure 15: Lake Superior wave height PDF log-linear (left) and log-log (right)

Wave measurements from the Atlantic Coast along the length of the USA were less widely dispersed (see below). It is much more unlikely to find very calm waters in the data set. In this case, a higher order BesselK function was used to model the wave height distribution.

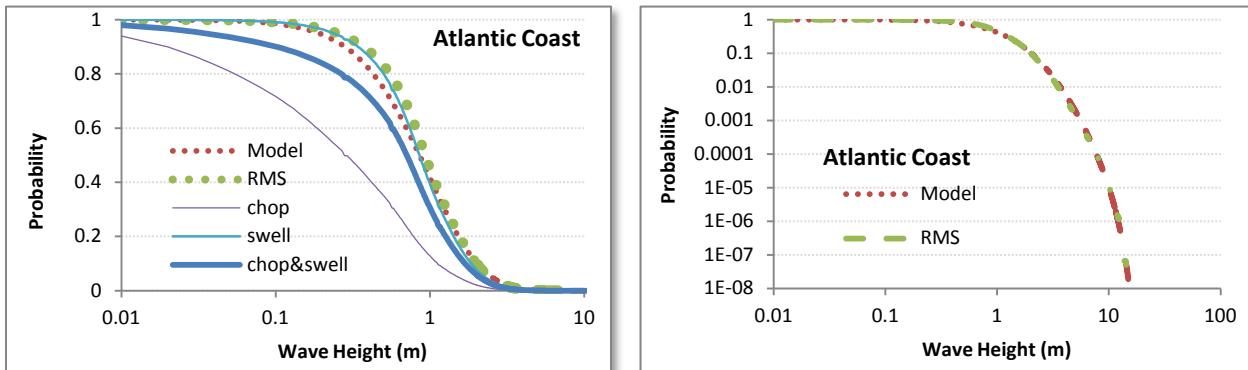


Figure 16: Atlantic Coast wave height PDF plotted as log-linear (left) and log-log (right)

In general, the divergence between the BesselK maximum entropy estimator for wind and that for waves is due to nonlinear effects at large wave-height. The heuristic Jahns/Wheeler correction factor is physically modeled by wave cusping, which generate larger heights than the triangular base wave predicts. This essentially rationalizes the sharpened crests and flattened troughs before it hits the cresting limit. We used a heuristic, but other corrections are available, such as derivations from the Rayleigh-Stokes (Tayfun model[36] [37]) process which is classified as a narrow-banded random process[38].

For use in simulations, other factors also play in such as, the probability of consecutive waves and the fact that phase velocity increases with the increase in wave steepness[39].

Even with the complexity inherent in modeling turbulence, the modeling at the PDF level has some predictive power. For example, using the Atlantic model parameters we can estimate the wave height for the Mediterranean coast of Greece [40]. In this sense, waves have universal characteristics.

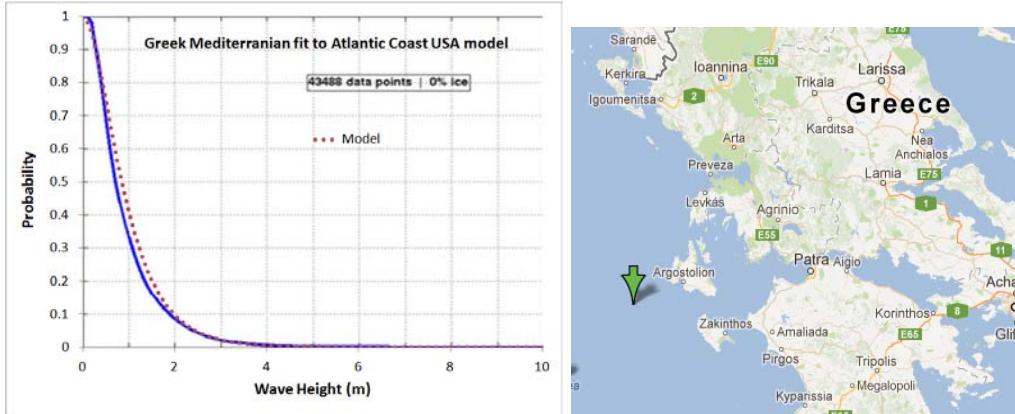


Figure 17: Mediterranean sea wave height assuming Atlantic coast model



Rainfall Intensity

As a premise, we want to consider whether a simple stochastic model can generate statistical patterns of rainfall events. We postulate that a critical point exists for rain to fall. The volume and density at which nature decides it reaches this critical point has much to do with the rate at which a cloud develops in its intensity and payload.

We assume a maximum entropy probability distribution function which assumes an average energy of rainfall rate for a storm, i :

$$P(E|E_i) = e^{-E/E_i}$$

The rationale for this is that the rainfall's energy is proportional to the rate of the rainfall, since that amount of moisture had to be held aloft by gravity.

$$\text{Rate}_i \sim E_i$$

Yet we know that the E_i can vary from storm to storm, so we leave it as a conditional, and then set that as a maximal entropy estimator as well

$$P(E_i) = \alpha e^{-\alpha E_i}$$

then integrating over the conditional's range.

$$P(E) = \int_0^{\infty} P(E|E_i)p(E_i)dE_i$$

This leads to the following solution:

$$P(E) = 2 \sqrt{\frac{E}{\pi}} K_1 \left(2 \sqrt{\frac{E}{\pi}} \right)$$

where K_1 is the modified BesselK function of the second kind, in this case of order 1. This is the same general derivation as we performed for wind speed.

This analysis was compared against this recent paper by Papalexio: "[Can a simple stochastic model generate rich patterns of rainfall events?](#)" [19], and graphed as shown below in Figure 18. The green points constitute the BesselK fit which lies right on top of the blue empirical data set.

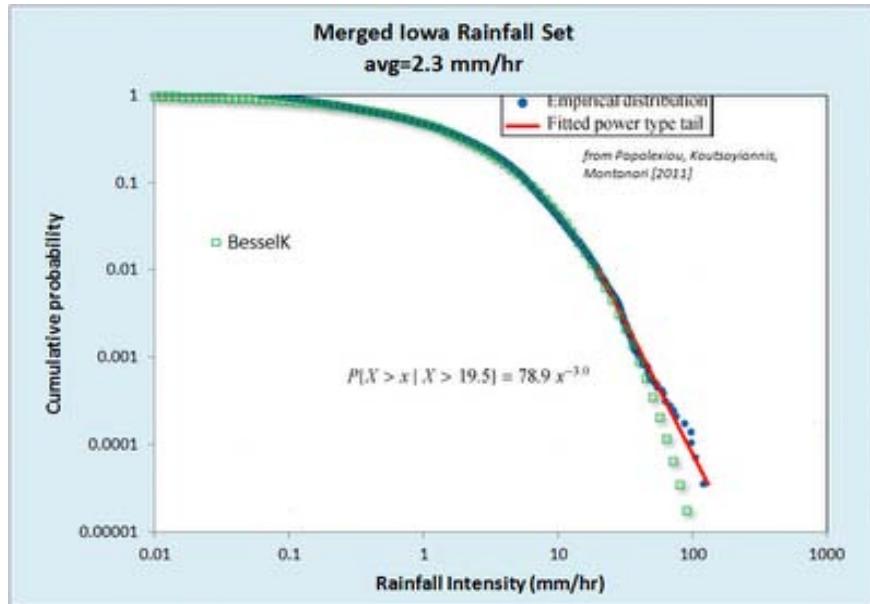


Figure 18: Cumulative rainfall distribution statistically gathered from several storm events.

From the table below reproduced from the paper, we can see that the mean value used in the BesselK distribution was *exactly the same* as that from the standard statistical moment calculation

Table 4: Rainfall moments

Event#	1	2	3	4	5	6	7	All
SampleSize	9697	4379	4211	3539	3345	3331	1034	29536
Mean(mm/h)	3.89	0.5	0.38	1.14	3.03	2.74	2.7	2.29
StandardDeviation(mm/h)	6.16	0.97	0.55	1.19	3.39	2.2	2	4.11
Skewness	4.84	9.23	5.01	2.07	3.95	1.47	0.52	6.54
Kurtosis	47.12	110.24	37.38	5.52	27.34	2.91	-0.59	91
HurstExponent	0.94	0.79	0.89	0.94	0.89	0.87	0.97	0.89

In contrast, Papalexio tried to apply Hurst-Kolmogorov statistics to the problem in search of a power law solution, claiming significance for the finding of a power law tail of -3. Instead we suggest that the slight deviation in the tail region is likely caused by insufficient sampling in the data set. A slight divergence starts to occur at the 1 part in 5,000 resolution level and since there are only 30,000 points in the merged data set, indicating that statistical fluctuations could account for the difference. See Figure 19 below which synthesizes a BesselK distribution from the same sample size, and can clearly duplicate the deviation in the tail.

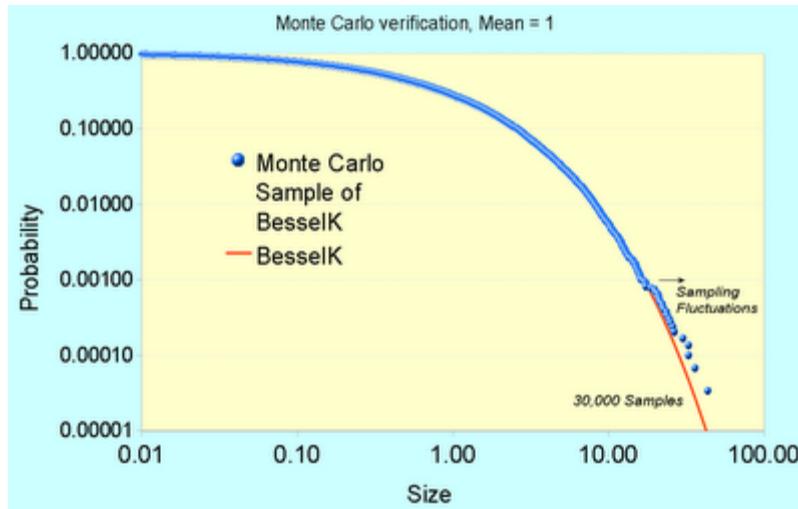


Figure 19: Monte Carlo sampling from a synthetic BesselK distribution reveals the same departure at low probability events.

Fluctuations of 1 in 30,000 are clearly seen at this level, so care must be taken to not read fat-tail behavior that may not exist.

The bottom-line is that this simple stochastic model outperforms Papalexio's fractal model, which doesn't seem as well-grounded with respect to first-principles once we apply an elementary uncertainty analysis to the data. The only physical premise needed to derive the physics grounded model specification is the intuition that rainfall buildup is a potential energy argument. This translates to a Boltzmann/Gibbs probability with a large spread in the activation energy. The large spread is modeled by maximum entropy — since the activation energy is unknown we assume a mean and let the fluctuations about that mean vary to the maximum amount possible. That is the maximum entropy activation which is proportional to a mean rainfall rate — the stronger the rate, the higher the energy.

That essentially becomes the premise for predicting the probability of a given rainfall rate within the larger ensemble defined by a mean. Note that Papalexio[19] seems to imply that the PDFs can go beyond the variability of a BesselK and into fatter tail territory (such as the MiejerG distribution, which is exponentially mixed prior with the Bessel). That is certainly acceptable as it will simply make it into a higher probability of outlier distribution, and if a fatter tail matches the empirical observations on a grander ensemble scale, then that might be a better characterization of the natural process [41].



The intermittent nature of wind power has a fundamental explanation based on entropy arguments. This same entropy-based approach explains some other related noisy and intermittent phenomena that modelers have to deal with. One case involves the use of mobile wireless gadgets such as WiFi devices, cell phones, and global positioning system (GPS) navigation aids in an imperfect (i.e. noisy) environment crowded out by electro-magnetic interference (EMI).

These wireless devices are often used in cluttered environments where ideal transmitted power mixed with EMI noise results in frustrating fadeouts that we need to patiently wait out. An example of Rayleigh fading appears in Figure 20 below. Signal interference-based explanations for why this happens can be

found, originating via the same intentional phase cancellations that occur in noise-cancelling headphones. The electronics in noise-cancelling headphones flip the phase so all interferences turn destructive, but for wireless devices, the interferences turn random, some positive and some negative, so the result gives the random signal shown. In the limit of a highly interfering environment the amplitude distribution of the signal shows a Rayleigh distribution, the same observed for wind speed.

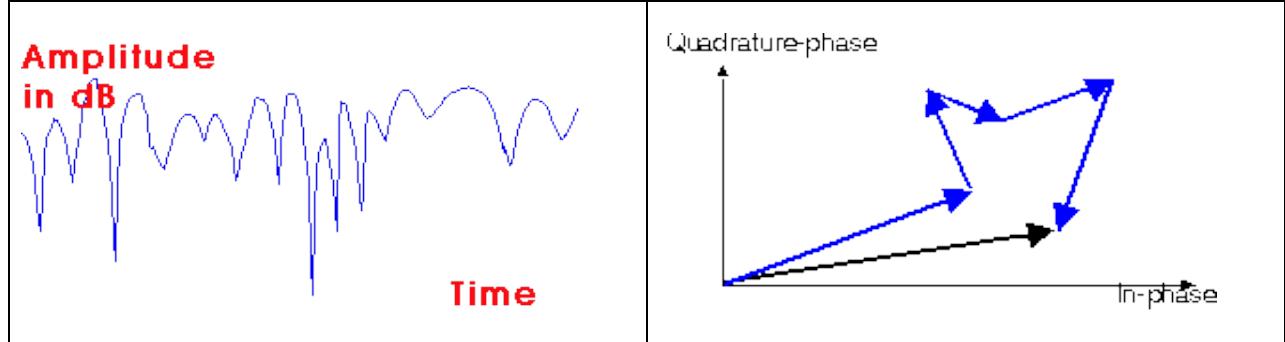


Figure 20: Noisy variation of signal amplitudes (left) is typically explained as a quadrature summation of randomly directed vector amplitudes (right). The variation in noise amplitude levels with time and the effect this has on a signal is known as Rayleigh fading.

Our knowledge of the situation reduces to that of knowing only the average power level of the signal. In that case, we can use Maximum Entropy Principles to estimate the amplitude from the energy stored in the signal, just like one can derive it for wind speed. So, as a starting premise, if we know the average power alone, then we can derive the Rayleigh distribution. The following figure shows the probability density function of the correlated power measured from a GPS signal. Since power in an electromagnetic signal relates to energy as a flow of constant energy per unit time, then we would expect the energy or power distribution to look like a damped exponential, in line with the maximum entropy interpretation. And it does exactly match a damped exponential.

$$p(E) = k \cdot e^{-kE}$$

This matches the observation of noise power level as shown in Figure 21.

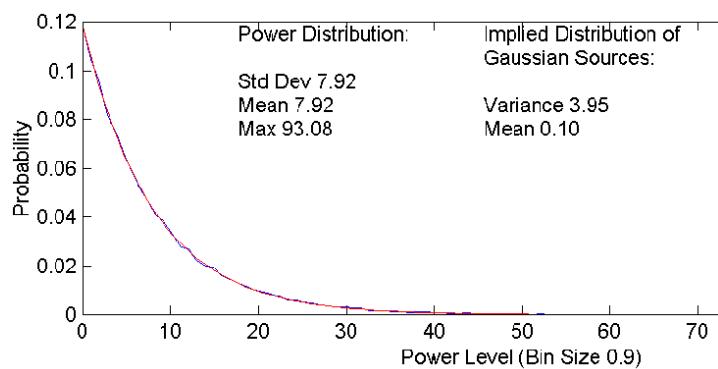


Figure 21: Note that the standard deviation equals the mean, a clear indication of a damped exponential. (from Watson[42])

Yet since power (E) is proportional to Amplitude squared, we can derive the probability density function by invoking the chain rule.

$$p(A) = p(E) \cdot \frac{dE}{dA} = e^{-kA^2} \frac{d}{dA}(A^2) = 2kr \cdot e^{-kA^2}$$

This precisely matches the Rayleigh distribution, implying that Rayleigh fits a Maximum Entropy (MaxEnt) distribution. So too does the uniformly random phase in the destructive interference process qualify as a MaxEnt distribution, which will range from 0 to 360 degrees (which gives an alternative derivation of Rayleigh). So all three of the distributions, the Exponential, Rayleigh, and Uniform, work together; and this provides a rather parsimonious application of the maximum entropy principle.

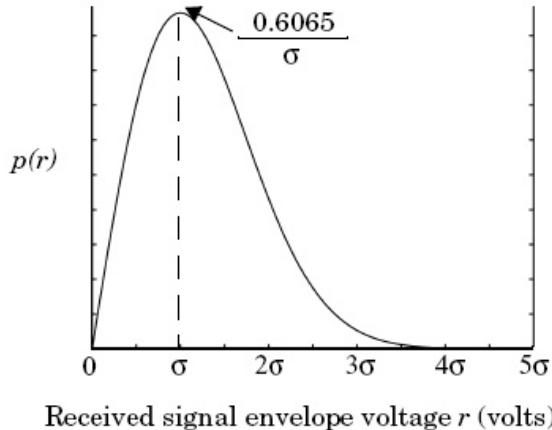


Figure 22: The Rayleigh Distribution of signal strength.

The most interesting implication of an entropic signal strength environment relates to how we deal with this power variation in our electronic devices. In a GPS, a vehicle's navigation system will experience this when trying to acquire a GPS signal from a cold-start. The amount of time it takes to acquire GPS satellites can range from seconds to minutes, and sometimes we don't get a signal at all, especially if we have tree cover with branches swaying in the wind.

Explaining the variable delay in GPS comes out quite cleanly as a fat-tail statistic if we consider how the GPS locks into the set of satellite signals. The solution assumes the entropy variations of the signal strength and integrating this against the search space that the receiver needs to lock-in to the GPS satellites. Since the search space involves time on one axis and frequency in the other, it takes in the limit $\sim N^2$ steps to decode a solution that identifies a particular satellite signal sequence for a particular unknown starting position. This gets reduced because of the mean number of steps needed on average in the search space. Dynamic programming matrix methods and parallel processing (perhaps using an FFT) can be used to get this to order N , so the speed-up for a given rate is t^2 . So this will take a stochastic amount of time according to the MaxEnt criteria.

However, due to the Rayleigh fading phenomenon, we don't know how long it will take to integrate our signal with regard to the rate R . This rate has a density function proportional to the power level distribution, then according to the rules for marginal distributions the conditionals line up to give the probability of acquiring a signal within time t .

$$P(t < T) = \frac{1}{1 + \left(\frac{T}{a}\right)^2}$$

This leads to the dispersion result where a is an empirically determined number derived from k and c . This is not considered an extremely fat tail because the acceleration of the search by quadrature tends to mitigate very long times.

Data Analysis. Data was collected from a GPS project that has a goal to speed up wild-fire response times by cleverly using remote transponders[43]. They published data for cold-start times as shown in the histogram below. Note that the data shows many times that approach 1000 seconds. The single parameter entropic dispersion fit ($a=62$ seconds) appears as the blue curve, and it fits the data quite well.

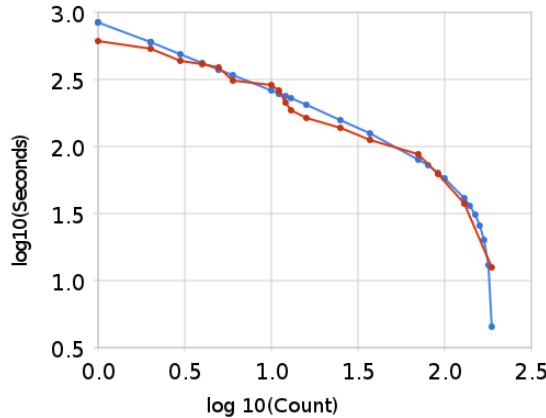


Figure 23: Fit of GPS acquisition times to a dispersive model

Beyond the Rayleigh, we will also find the BesselK PDF, as that is what describes the K-distributed clutter which appears in large regions collected by radar stations against random terrestrial and oceanographic features. The heterogeneous nature of that environment will generate clutter with longer tails than the Rayleigh [7]. The fatter tails of K-distributed clutter has implications for triggering false alarms when trying to distinguish signal from noise [44].

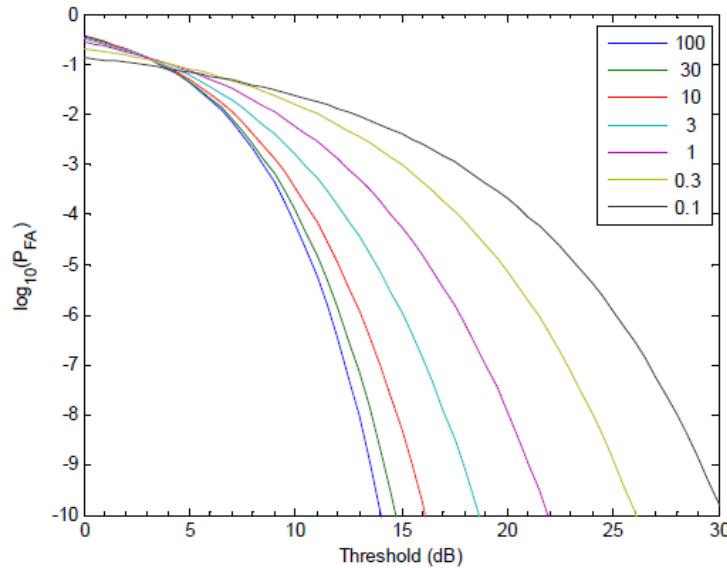


Figure 24: Probability of false alarm for a single pulse return from K-distributed clutter, for different values of the shape parameter v . Adapted from Bocquet [44]



If we assume that clouds develop by some sort of preferential attachment, then the uncertainties at which the preferential attachment process increases with time balanced against the uncertainties in the critical point contribute to the dispersion. This is an analysis based on work by Yuan at NASA Goddard, “Cloud macroscopic organization: order emerging from randomness” [20]

The cloud size distribution follows a variant of the Zipf-Mandelbrot law, which also fits several other natural characterizations, such as oil field volume[26] and lake size distributions (see next section).

$$P(\text{Size}) = \frac{1}{1 + \frac{\text{Median}}{\text{Size}}}$$

We first assume that water vapor disperses through the atmosphere freely. The rate r at which it does this we treat as a stochastic variable with a probability density

$$p(r|g) = \frac{1}{g} \cdot e^{-\frac{r}{g}}$$

This introduces two concepts at once: the idea that we do not assume a single rate (i.e. assume instead *dispersion*) together with the idea that we can only assume at best a mean (as the growth rate g) *and* to treat the standard deviation as equivalent to the mean. This type of assumption makes the least presuppositions as to what has happened — we know we have a mean value but beyond that, the rate can vary to the maximum entropy limit.

If we next assume that a collection of these rates can act to sweep out a selected uniform unit of volume, then over time we can imagine that a cloud will capture this moisture.

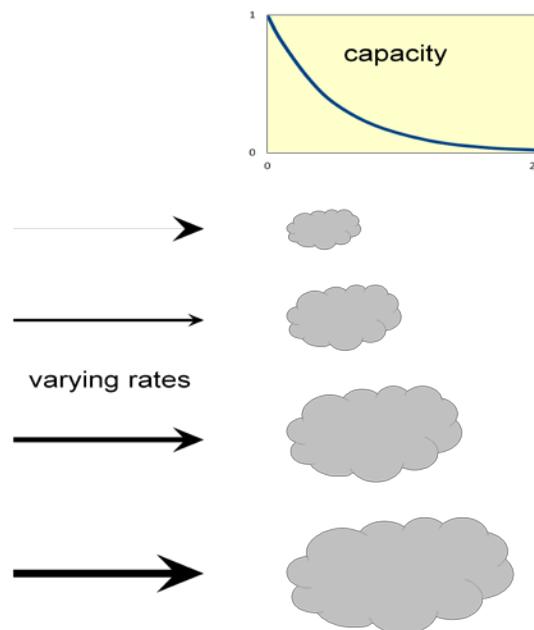


Figure 25: Cloud formation process leads to dispersion in sizes

Suppose that the water vapor diffuses outward, so for a given time period t , the moisture will diffuse over a distance $x = r t$, a simple variable change gives

$$p(r|g) = \frac{1}{gt} \cdot e^{-\frac{x}{gt}}$$

Over time, the probability that some moisture will migrate at least a length x_0 distance is:

$$p(x > x_0 | g, t) = \int_0^\infty p(x|g, t) dx = e^{-\frac{x_0}{gt}}$$

Alternatively, the following relation tells us the cumulative probability of the distance covered by material after time t . This again assumes a distance travelled $x = r t$.

$$P(x_0 | g) = \int_{r=x_0/t}^\infty p(r) dr = e^{-\frac{x_0}{gt}}$$

This relation also crops up in terms of the *population balance equation*. It basically relates a conservation of particles law, in that we do not lose track of any material due to a flow.

So next we have to accumulate this over a volume or depth at which a cloud develops. The simplest approximation assumes that the water droplets get distributed to a mean height (H) with a similar exponential distribution — this is like a capacity for the cloud formation (see Figure 25):

$$f(x|H) = \frac{1}{H} e^{-\frac{x}{H}}$$

Combining the two relations turns into an *a priori* probability for the expected cumulative transfer after time t through the volume. Integrating over the atmospheric layer over which clouds can form, gives the average water volume accumulated:

$$\begin{aligned} C(t|H) &= \int_0^\infty f(x|H) \cdot P(x|g) dx = \int_0^\infty f(x|H) \cdot e^{-\frac{x}{gt}} dx \\ C(t|H) &= \frac{1}{1 + \frac{gt}{H}} \end{aligned}$$

For the last assumption, we note that if t gets evenly spread from the over time, then the value gt becomes the effective collected thickness W of a distribution of clouds, where we add a factor k to indicate collection efficiency. Alternatively, we can interpret the stochastic variable W as the maximum net cloud thickness that would develop over a diffusion time t . The term kH sets the potential maximum net thickness achievable turning it into a hyperbolic discounting probability distribution.

$$C(W|H) = \frac{1}{1 + \frac{kH}{W}}$$

Or as we first surmised:

$$P(Size|Median) = \frac{1}{1 + \frac{Median}{Size}}$$

The agreement to the collected data is remarkable over several orders of magnitude..

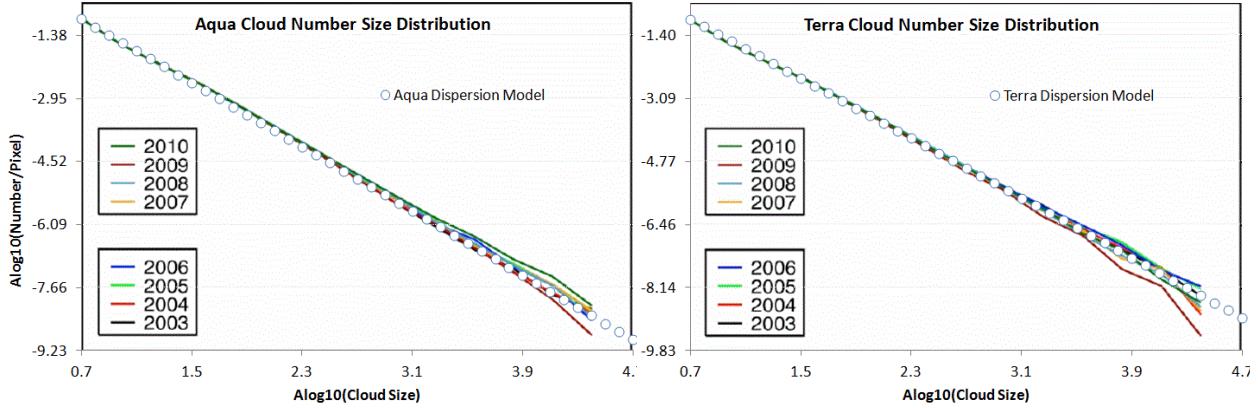


Figure 26: Cloud statistics using two different telemetry systems[20]. The same power law holds over several orders of magnitude.

Cloud size modeling may have some limited utility for nominal weather prediction (such as expected durations of shade and estimating the potential for lightning), but it does demonstrate well the wide aleatory uncertainty in a natural phenomena, and how widespread it is.



Lake sizes

Our environment shows great diversity in the size and abundance in natural structures. Freshwater lakes accumulate their volume in a dispersive manner. Over geologic time, water drifts into a basin at various rates and over a range in collecting regions. As lakes capture most of their volume through water drainage, one can imagine that the rate of influx plays a factor in how large a lake can become. The Maximum Entropy prediction of the size distribution leads to the following expression, exactly the same as for cloud formation:

$$P(\text{Size}) = \frac{1}{1 + \frac{\text{Median}}{\text{Size}}}$$

Surveys of lake-size show the same reciprocal power law dependence, with the exponent usually appearing arbitrarily close to one. In the Figure 27 below, the data plotted on a ranked plot clearly shows this dependence over several orders of magnitude.

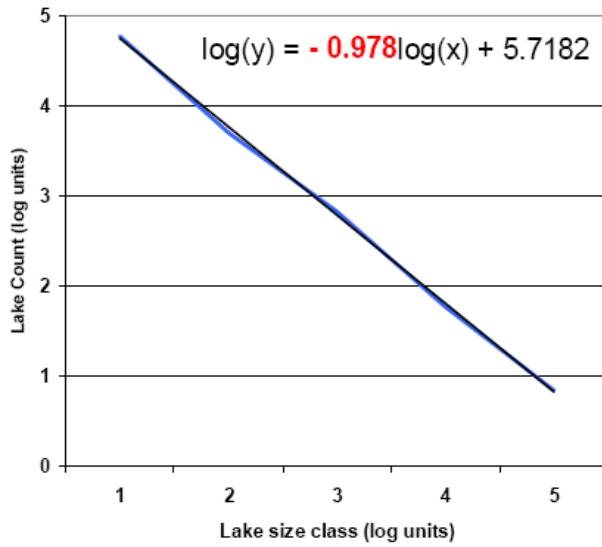


Figure 27: Northern Quebec lakes map and size distribution adapted from Telmer [45].

More revealing is that in Figure 28 below, we can observe the bend in the curve that limits the number of small lakes in exact accordance to the equation shown above. The agreement with such a simple model suggests that of a universal behavior.

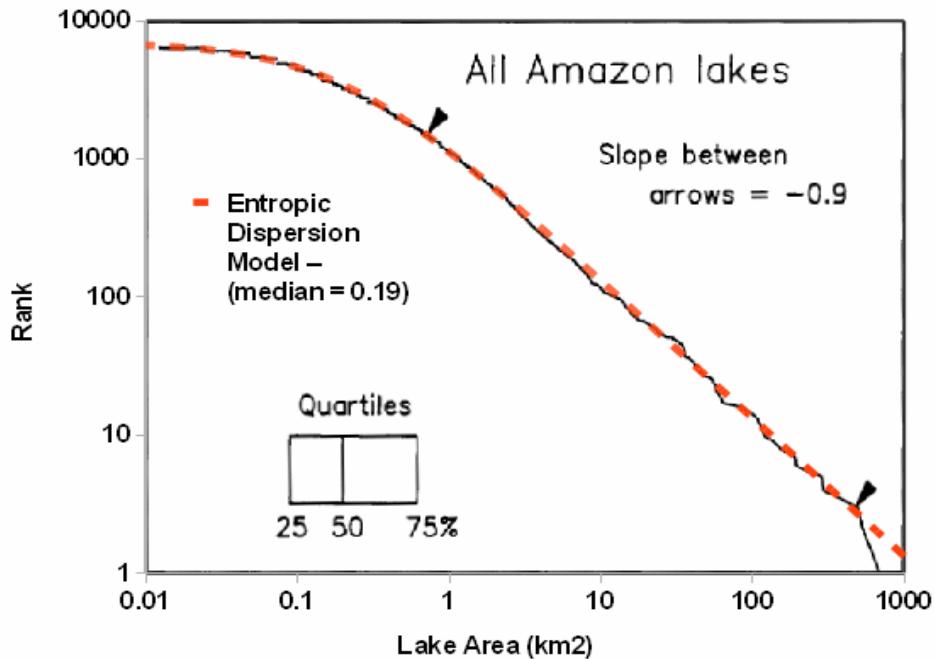


Figure 28: Amazon lake size distribution from “Estimation of the fractal dimension of terrain from Lake Size Distributions”[46]

This general trend is repeated for lakes around the world. All that is required is to have a median value for lake area and the rest of the distribution will roughly scale to this factor.

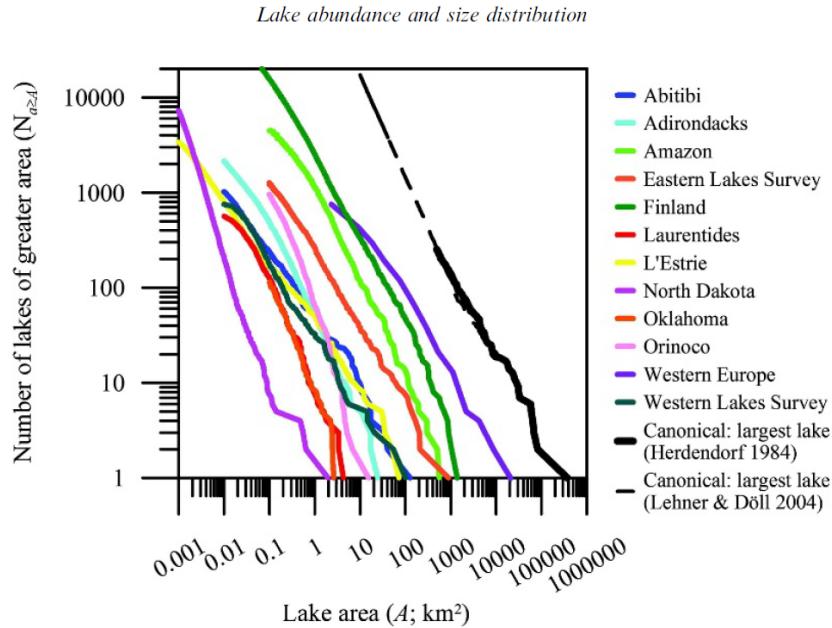


Figure 29: Near universal scaling of lake size distributions across the world [21]

Models of lake size distribution may have utility for vehicle fording and amphibious vehicle simulations.



The role of entropy overrides other factors enough so that some simple dispersion arguments can explain the size distribution completely. Take as an example the formation of ice crystals in a cirrus cloud. Depending on the surrounding temperature, a crystal nucleates on some foreign particle and then starts growing. The atmospheric conditions have enough variety that the growth rate will disperse to the maximum entropy amount given a mean rate value. The end state for volumetric growth will also show the same amount of variation, where x is the size variate and S is the mean size.

The following particle size distribution (PSD) graph, Figure 30, shows measurements taken from high altitude cloud experiments [47][48][22]. The size gets measured along a single length dimension and the density of the particles takes the place of a probability.

The main profile follows a power-law in volumetric growth, where S is the median volume (see [26]):

$$p(x) = \frac{x}{(x + S)^2}$$

Crystal sizes get reported as a length and we have to convert that to a volume. This means the derivative has to include a chain rule to convert the volume x to a length parameter L , $x \sim L^3$ generates $dx/dL \sim L^2$.

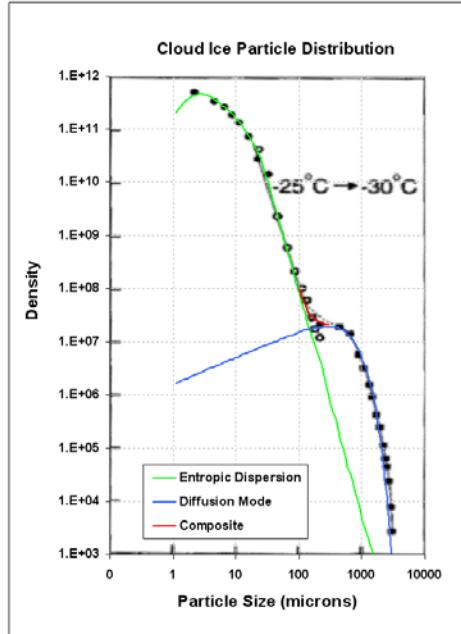


Figure 30: Cloud ice particle number density $n(D)$ vs. the long dimension of particles as observed at the temperature range of -25C and -30 C (from Platt[47]). It shows a bimodal structure in the ice crystal distribution with the second peak at ~500 microns.

The data fits the dispersion model nicely (green line), but notice at low density that an extra mode shows up as the blue line. This clearly has a sharp exponential drop so likely has a non-dispersive origin. In terms of the higher density entropic model, this stands out as an ordered nucleation regime in the midst of a sea of disordered ice crystal growth modes. Applying a prior distribution is similar to integrating profiles of different weight (aka *superstatistics*) to come up with a power-law as shown below. By comparing this to the empirical sized distribution one can see how a distinct peak could occur.

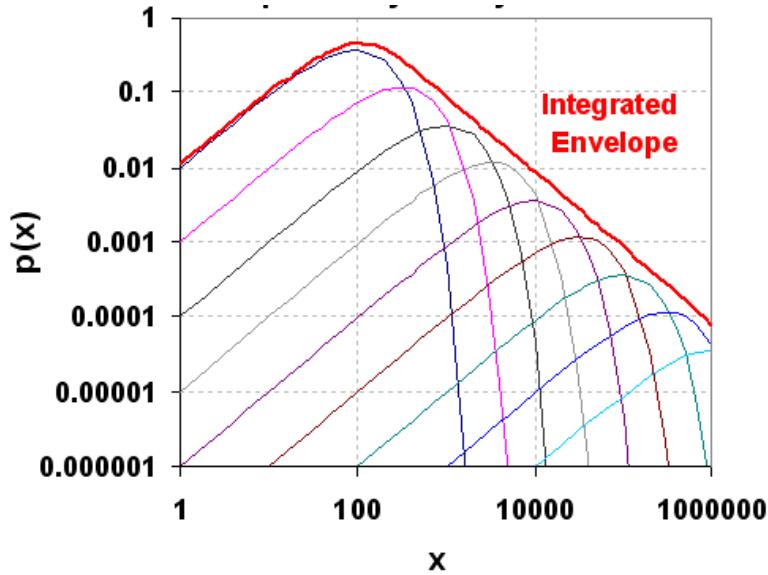


Figure 31: Superposition of exponential density profiles leads to an entropic dispersion envelope. The maximum entropy distribution as a prior distribution acts to smooth the envelope, but a strong component could stand out. Indeed this is typical in particle size distributions.

Physically why does this bump occur? Some unknown nucleation process has provided an optimal growth environment for these crystals to deviate from the entropic distribution. Hypothesizing, this could take the form of a catalyst or an accommodating growth substrate. With a power tail of -3/2 this might well have a strong diffusive growth component. However, the nuclei occur rarely enough so they do not drown out the much more common random or spontaneously occurring growth centers. It thus shows up as a clear non-dispersive growth mode in a sea of non-uniformity.

On a micro-level, we do have a population of reproducible structured shapes to bind against — as the airborne particulate world shows some uniformity in its density. This original analysis may prove of some help to those looking at particle size distributions of volcanic ash (see the figure below). Researchers routinely apply a log-normal fit to the data — yet one that uses a MaxEnt dispersion formulation with the appropriate volume/diameter exponent often can work just as well. Below, we use a root 1/2 dispersive growth rate on volume which may indicate a diffusion-controlled rate.

To account for wear/tear/corrosion modeling, one could apply particulate and ash models (Figure 32 and Figure 33, the latter featuring multi-modal size features) that feed into models for physical breakdown.

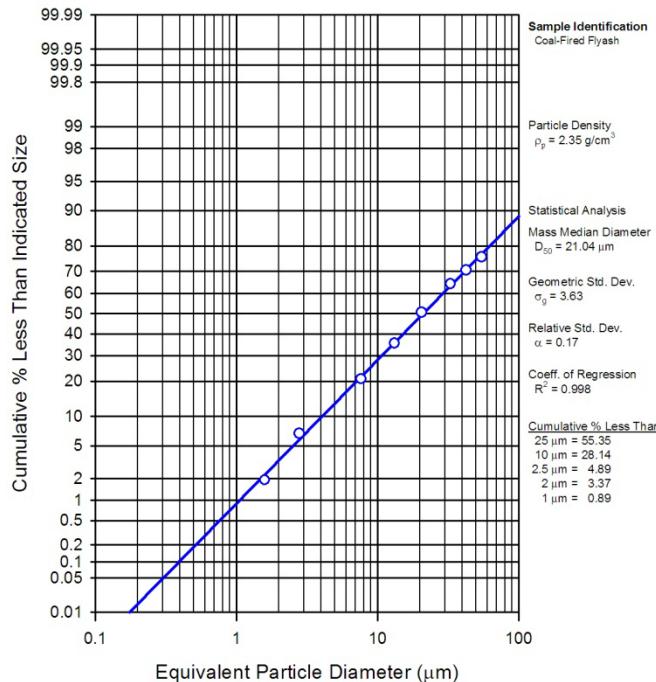


Figure 32: Wiki data for fly ash size distribution[49].

The signature feature of volcanic ash plumes is that they have dense populations of certain sized particulates (Figure 33), having derived from a homogeneous environment.

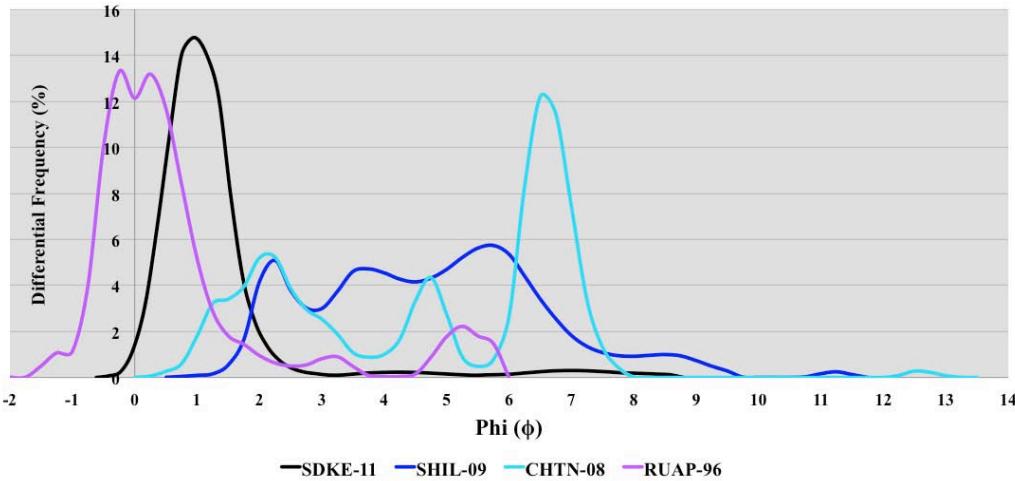


Figure 33: Volcanic Ash [50]. This is on the Krumbein phi (ϕ) particle size scale [51] which is negative of the log base 2 of the particle diameter. That means smaller diameters are to the right side of the scale.

Discussion

This study has described the characterization and modeling of natural phenomena in a form suitable for contextual use. We have provided examples of slope distribution, wind speed, rainfall rates, lake sizes, etc. Most of the models reduce to fairly simple analytical forms.

The probability measures derived from theory and applied to the data are often snapshots in time of a continuous growth process. Physical processes can accumulate or discard material following rates of material flow. While this may seem overly simplistic to consider, it does set the stage for revealing patterns in behavior.

For example, a statistically measured volume could be the result of various rates operating over varying time spans. Along with this variation, the concept of an average or mean value plays a role into modeling the behavior.

Generally speaking, two categories of models exist:

- *Models of variations around a mean energy:* Measures are bound by uncertainty in the mean energy value of a measured value, such as wind, rainfall rate, clutter, terrain slopes, wave crest height; these all generate the same universal curve, the ubiquitous modified Bessel of the second kind.
- *Models of variations around a rate and enclosed volume:* Several measures are bound by considerations of the rate of accumulation and the uncertainty of the eventual volume: such as clouds, particles, and lake sizes give the same entropic power law.⁵

We also described how the statistical spreads come about, via the concept of dispersion and maximum entropy. One of the outcomes of modeling is that a significant amount of data, 5 billion in the case of USA terrain and several hundred million for just the Lake Michigan wave data, can be expressed in a

⁵ Plus oil reservoirs.

much more concise form.⁶ This has a practical implication of allowing a terse model to summarize voluminous data to usefully assess candidate engineering designs. Further, as these models are parameterized by physical variables such as mean energy, the concise formulation allows exploration of extreme conditions extrapolated from but not exemplified by existing data sets.

Sampling approaches for simulation are often very straightforward for these distributions. We described several efficient applications of random sampling for transcendental functions such as the BesselK. We can further apply techniques such as importance sampling to reduce the sample size needed or to reduce the variance of the statistics.

In certain cases it may be possible to invert the stochastic models in order to support assessment of assume-guarantee conditions on component engineering models and to derive Probabilistic Certificates of Correctness (PCCs). The classical case is for normal distributions, but for fat-tail distributions such as the hyperbolic power law, this is also rather straightforward. To make sampling efficient, particularly for rare events, techniques such as importance sampling may be required. If a probability for a certain state is 10 orders of magnitude more rare than the most frequent state, the rule of thumb is that 10^{10} more samples may be required to catch this event. Importance sampling can cut the required number of samples down if the density function is well-characterized.

The fact that many of these models follow similar patterns makes them useful for model automation. Abstract interfaces for common distributions such as the BesselK and hyperbolic power law can be generated and are of broad utility (see Appendix D).

Other characterization techniques such as the multiscale entropy measure[52] can be applied to temporal and spatial scales covering a wide dynamic range. This reveals the amount of disorder and uncertainty in the data, which is important as a quick characterization metric.

Summary and path forward

In general, support for context modeling and the general notion of Internet-based data collection allows us to leverage a vast amount available data and reduce its dimensionality and scope by careful characterization and modeling. Certain elementary characteristics related to probability density functions were covered in this paper.

Frequency characteristics and correlations will be described in the second paper in this series, “Terrain Characterization” (see Appendix B).

The third paper in this series, “Diffusive Growth” uses universal principles to apply the same pattern to characterize other disparate natural phenomenon, such as corrosion and thermal dispersion (see Appendix C).

Furthermore, by applying ontology-based approaches for organizing models and techniques we can set the stage for broader collections of such models discoverable by a general community of designers and analysts. Together with standard access protocols for context modeling these innovations provide the promise of making environmental context models generally available and reusable, significantly assisting the routine application of model based engineering. That is the scope of the fourth paper in this series (see Appendix D).

⁶ Archived data often gets stale, and eventually gets deleted via some bureaucratic measure. Possessing a model that can potentially condense 4 billion data points into a single function containing a single parameter has clear benefits. It amounts to a 4 Billion to 1 reduction in storage size and encapsulated domain knowledge.

References

- [1] F. E. Pritchard, C. C. Easterbrook, and G. E. McVehil, "Spectral and Exceedance Probability Models of Atmospheric Turbulence for Use in Aircraft Design and Operation," DTIC Document, 1965.
- [2] E. T. Jaynes and G. L. Brethorst, *Probability theory: the logic of science*. Cambridge Univ Pr, 2003.
- [3] C. Beck and E. Cohen, "Superstatistics," *Physica A: Statistical Mechanics and its Applications*, vol. 322, pp. 267–275, 2003.
- [4] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet mathematics*, vol. 1, no. 2, pp. 226–251, 2004.
- [5] C. E. Shannon, W. Weaver, R. E. Blahut, and B. Hajek, *The mathematical theory of communication*, vol. 117. University of Illinois press Urbana, 1949.
- [6] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis Of Real-World Signals*. A K Peters, Ltd., 2010.
- [7] U. Grenander and A. Srivastava, "Probability models for clutter in natural images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 4, pp. 424–429, 2001.
- [8] E. W. Montroll and M. F. Shlesinger, "Maximum entropy formalism, fractals, scaling phenomena, and 1/f noise: a tale of tails," *Journal of Statistical Physics*, vol. 32, no. 2, pp. 209–230, 1983.
- [9] N. N. Taleb, *The black swan: The impact of the highly improbable*. Random House Inc, 2010.
- [10] D. Sornette, *Critical phenomena in natural sciences: chaos, fractals, selforganization, and disorder: concepts and tools*. Springer Verlag, 2004.
- [11] M. Gell-Mann, *The Quark and the Jaguar: Adventures in the Simple and the Complex*. St. Martin's Griffin, 1995.
- [12] N. Taleb, *Fooled by randomness: The hidden role of chance in life and in the markets*. Random House Inc, 2005.
- [13] E. J. Gumbel and J. Lieblein, "Some applications of extreme-value methods," *The American Statistician*, vol. 8, no. 5, pp. 14–17, 1954.
- [14] R. Hanel, Thurner, and M. Gell-Mann, "Generalized entropies and the transformation group of superstatistics." [Online]. Available: <http://www.pnas.org/content/108/16/6390.full>. [Accessed: 15-Mar-2012].
- [15] D. Mumford, "The dawning of the age of stochasticity," *Mathematics: Frontiers and Perspectives*, pp. 197–218, 2000.
- [16] S. Bankes, D. Challou, T. Haynes, H. Holloway, P. Pukite, J. Tierno, and C. Wentland, "META Adaptive, Reflective, Robust Workflow (ARRoW)," BAE Systems, Final Report TR-2742, 2011.
- [17] D. J. Noakes, A. I. McLeod, and K. W. Hipel, "Forecasting monthly riverflow time series," *International Journal of Forecasting*, vol. 1, no. 2, pp. 179–190, 1985.
- [18] BPA, "BPA Meteorological Information," *Meteorological Information from BPA Weather Sites*. [Online]. Available: <http://transmission.bpa.gov/business/operations/wind/MetData.aspx>. [Accessed: 31-May-2012].
- [19] S. M. Papalexiou, D. Koutsoyiannis, and A. Montanari, "Can a simple stochastic model generate rich patterns of rainfall events?," *Journal of Hydrology*, 2011.
- [20] T. Yuan, "Cloud macroscopic organization: order emerging from randomness," *Atmos. Chem. Phys*, vol. 11, pp. 7483–7490, 2011.
- [21] J. A. Downing, Y. T. Prairie, J. J. Cole, C. M. Duarte, L. J. Tranvik, R. G. Striegl, W. H. McDowell, P. Kortelainen, N. F. Caraco, J. M. Melack, and others, "The global abundance and size distribution of lakes, ponds, and impoundments," *Limnology and Oceanography*, pp. 2388–2397, 2006.
- [22] D. L. Wu, "MLS Cloud Ice Measurements." [Online]. Available: <http://mls.jpl.nasa.gov/dwu/cloud/index.html>. [Accessed: 27-Apr-2012].
- [23] CDIP, "CDIP Homepage." [Online]. Available: http://cdip.ucsd.edu/?units=metric&tz=UTC&pub=public&map_stati=1,2,3. [Accessed: 24-Apr-2012].
- [24] C. of E. US Army, "Wave Information Studies." [Online]. Available: <http://wis.usace.army.mil/hindcasts.shtml>. [Accessed: 31-May-2012].
- [25] USGS, "USGS/EROS Find Data/Products and Data Available/DEM." [Online]. Available: http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/DEM. [Accessed: 24-Apr-2012].
- [26] P. R. Pukite, *The Oil Conundrum: Vol. 1 Decline, Vol. 2 Renewal*, vol. 1,2, 2 vols. Daina, 2011.
- [27] G. Vico and A. Porporato, "Probabilistic description of topographic slope and aspect," *J. Geophys. Res*, vol. 114, p. F01011, 2009.

- [28] J. S. Gagnon, S. Lovejoy, D. Schertzer, and others, "Multifractal earth topography," *Nonlinear Processes in Geophysics*, vol. 13, no. 5, pp. 541–570, 2006.
- [29] G. Gonçalves and J. Santos, "Propagation of Dem Uncertainty: An Interval Arithmetic Approach.," in *XXII International Cartographic Conference, Spain*, 2005.
- [30] A. M. Guarneri, "SAR interferometry and statistical topography," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 40, no. 12, pp. 2567–2581, 2002.
- [31] D. Carrier, "Renewable Energy Analysis for Afghanistan," RDECOM AMSAA, 2010.
- [32] H. Jahns and J. Wheeler, "Long-term wave probabilities based on hindcasting of severe storms," *Journal of Petroleum Technology*, vol. 25, no. 4, pp. 473–486, 1973.
- [33] G. Z. Forristall, "Wave crest distributions: Observations and second-order theory," *Journal of Physical Oceanography*, vol. 30, no. 8, pp. 1931–1943, 2000.
- [34] R. C. Nelson, "Depth limited design wave heights in very flat regions," *Coastal Engineering*, vol. 23, no. 1, pp. 43–59, 1994.
- [35] J. Fenton, "Nonlinear wave theories," *The Sea*, vol. 9, no. 1, pp. 3–25, 1990.
- [36] M. A. Tayfun, "Narrow-band nonlinear sea waves," *Journal of Geophysical Research*, vol. 85, no. C3, pp. 1548–1552, 1980.
- [37] H. Socquet-Juglard, K. Dysthe, K. Trulsen, H. E. Krogstad, and J. Liu, "Probability distributions of surface gravity waves during spectral changes," *Journal of Fluid Mechanics*, vol. 542, no. -1, p. 195, Oct. 2005.
- [38] A. H. Izadparast and J. M. Niedzwiecki, "Empirical Moment-Based Estimation of Rayleigh-Stokes Distribution Parameters," presented at the Proc. 21st International Ocean and Polar Engineering Conference, 2011.
- [39] R. H. Stewart, *Introduction to physical oceanography*. A & M University, 2003.
- [40] C. DMI, "DMI / COI [Wave statistics]." [Online]. Available: <http://ocean.dmi.dk/wavestat/index.uk.php>. [Accessed: 20-Sep-2012].
- [41] D. Koutsoyiannis, "The scaling properties in the distribution of hydrological variables as a result of the maximum entropy principle," presented at the Geophysical Research Abstracts, 2005, vol. 7, p. 03781.
- [42] J. R. A. Watson, *High-sensitivity GPS LI signal analysis for indoor channel modelling*. University of Calgary, Department of Geomatics Engineering, 2005.
- [43] M. Chen, C. Majidi, D. Doolin, S. Glaser, and N. Sitar, "Design and construction of a wildfire instrumentation system using networked sensors," *Network Embedded Systems Technology (NEST) Retreat, Oakland California. Retrieved April*, vol. 5, 2004.
- [44] S. Bocquet, "Calculation of Radar Probability of Detection in K-Distributed Sea Clutter and Noise," DTIC Document, 2011.
- [45] K. Telmer, M. Costa, and J. MacGregor, "K&C Science Report–Phase 1 Global Lake Census."
- [46] S. K. HAMILTON, J. M. MELACK, M. F. GOODCHILD, and W. Lewis, "Estimation of the fractal dimension of terrain from lake size distributions," *Lowland floodplain rivers: Geomorphological perspectives*. Wiley, pp. 145–163, 1992.
- [47] C. M. R. Platt, "A parameterization of the visible extinction coefficient of ice clouds in terms of the ice/water content," *Journal of the atmospheric sciences*, vol. 54, no. 16, pp. 2083–2098, 1997.
- [48] D. L. Wu, J. H. Jiang, and C. P. Davis, "EOS MLS cloud ice measurements and cloudy-sky radiative transfer model," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 5, pp. 1156–1165, 2006.
- [49] Wikipedia, "Particle-size distribution - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Particle-size_distribution. [Accessed: 31-May-2012].
- [50] Wikipedia, "Volcanic ash - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Volcanic_ash. [Accessed: 31-May-2012].
- [51] Wikipedia, "Particle size (grain size) - Wikipedia, the free encyclopedia." [Online]. Available: [http://en.wikipedia.org/wiki/Particle_size_\(grain_size\)](http://en.wikipedia.org/wiki/Particle_size_(grain_size)). [Accessed: 31-May-2012].
- [52] P. Pukite and S. Banks, "Entropic Complexity Measured in Context Switching," in *Applications of Digital Signal Processing*, vol. 17, InTech, 2011.

Unified characterization of surface topography for vehicle dynamics applications

Paul Pukite, Steve Bankes, Dan Challou: *BAE Systems*

Carl Becker: *CSIR/University of Pretoria*

Richard Bradford: *Rockwell Collins*

Abstract: Environmental models by their nature contain a great deal of uncertainty. Since the underlying behavior of the model is rarely controlled by an ordered process, any model characteristics will carry along with it a level of aleatory uncertainty governed by the natural disorder. This paper applies novel uncertainty characterization approaches to classes of topographic models to demonstrate how to quantify the natural order and distinguish from artificial (man-made) order.

Preface

Comprehensive vehicle testing requires *environmental context* models to evaluate a design's robustness and fitness for use. For a ground vehicle designed for both on-road and off-road use, interactions with the terrain provide an important testing context. For amphibious vehicles, the environmental context extends to sea states and aquatic currents. As an overriding factor, being able to traverse and navigate successfully various surface topographical features plays a large role in verifying that a vehicle's requirement specifications can be met.

To streamline the verification process, terrains need to be first well characterized and then modeled comprehensively enough that probability of correctness approaches can be applied. Without the ability to quantify for the probability of occurrence for a specific environmental effect, any requirements based on topography, such as traverse grade, can't be propagated to a virtual test of the vehicle's design — and thus we cannot substantiate with any degree of certainty or accuracy whether the vehicle meets the customer's needs.

This paper applies general methods for stochastic modeling of environment contexts described in a companion paper (volume 1 of this series) "Stochastic Analysis for Context Modeling" (see Appendix A).

Primer on Characterization and Modeling

To develop some breadth, the models described here will cover both land and aquatic topographies. The landform aspects we refer to as *terrain modeling* [1] and the aquatic as *wave* (or *wave energy*) *modeling*[2].

For terrains, we consider a rigid model of altitudes as a function of lateral dimension. Often, a one-dimensional path, with provision for left and right tracks, is sufficient to characterize a terrain. The interaction of the vehicle with this type of terrain path provides enough of a context to virtually verify requirements such as absorbed power at speed and fuel efficiency on inclined grades.

Stochastic models[3] work very effectively to describe natural as well as artificial, man-made terrain. Enough disorder and randomness exists in the natural world, and even within an artificial world of cobblestones, chatter bumps, rumble strips, and wave machines, that a synthetic model using probability concepts appears indistinguishable from an empirically measured set of real-world data[4].

The stochastic tools of the trade include probability distribution functions, autocorrelation functions, power spectral densities, and moment distributions [5].

Probability density functions (PDF) model sample spaces. A single PDF graph facilitates the characterization of natural phenomena that are prevalent in context modeling, including terrain slope distributions as well as wind, rainfall, lakes, cloud, etc.[6] Many of these models reduce to fairly simple analytical forms. For disordered systems and data containing aleatory uncertainty[7], we use techniques such as the *maximum entropy principle*[8] and *super-statistics*[9], which have a more formal basis than heuristic models can provide¹. For narrow distributions, we can apply Normal or Gaussian statistics.

Autocorrelations (ACF) and *power spectral densities* (PSD) are tied closely together. In basic terms, an autocorrelation describes the amount of similarity between two points separated spatially or temporally [10]. For static terrain this is a purely spatial consideration, but at a constant vehicle speed, an autocorrelation turns into a temporal characteristic, as time equates to motion across a terrain. A PSD contains the same information as an ACF but works in the inverse space, either the wavenumber (k) space in the spatial domain, or frequency (ω) space in the time domain. The decision to select an autocorrelation function or power spectral density will depend on the application. A spectral representation can easily provide reconstructed instances for either spatial or temporal domains in a general system context. The power of the real space autocorrelation is that it can recover a synthetic representation of the local terrain as a random walk model, while the PSD can approximate a varied terrain as a superposition of sine waves.

In addition, certain *multiscale* metrics[11] are useful to further characterize and in particular disambiguate potentially degenerate model descriptions. In particular, the multiscale entropy measure can be applied to temporal and spatial scales covering a wide dynamic range, and multiscale variances are sensitive to asymptotic behavior. What this tells us is the amount of disorder and uncertainty in the data, which is important as a concise supporting characterization metric.

Table 1: Stochastic Model Categories

Stochastic Model	Includes and Related		
Probability Density Function	PDF	Cumulative Distribution Function, Histogram	Position Independent
Autocorrelation Function	ACF	Correlation, Autocovariance	Relative
Power Spectral Density	PSD	Power Spectrum, Periodogram, Fourier Spectrum	Relative
Multiscale Analysis	MSA	Multiscale Variance, Multiscale Entropy	Relative

Table 1 above categorizes the stochastic models used to characterize terrain and waves. These can work together to model some sophisticated profiles. For example, a probability density function can model a length distribution, which then gets applied to a semi-Markov formulation for an autocorrelation analysis. By then applying a *Fourier transform* to the autocorrelation, we can arrive at a very concise representation. We will describe this in the analysis section.

Data Characterization

The rules for construction of probability density functions follow from some elementary principles. Samples from experimental data are ranked, and then normalized to the largest (i.e. scarcest) sample. This establishes a cumulative probability of unity when integrated over the sampled data space. The rank histogram is then converted to a cumulative distribution function by interpolating across a continuum, and then a PDF derives from the first derivative with respect to the random variate. Strictly speaking, a PDF is a discrete (binned) form while a probability mass function (PMF) is the continuum, but we will uniformly use the term PDF, even though we generally assume the continuum.

¹ The sample size needed to reduce the variance of the statistics can be reduced by efficient applications of random sampling such as importance sampling.

Given that X is some variate of interest, then a PDF described by $f(X)$ can be used to generate moments of the distribution, such as the expected value $\mathbf{E}[X]$.

$$\mathbf{E}[X] = \bar{X} = \int_{-\infty}^{\infty} X f(X) dX$$

Autocorrelation functions consider the pair-wise expected value of all samples separated by a distance measure. This describes the affinity for localized interactions that a PDF lacks sensitivity to. The computational complexity of calculating an autocorrelation can go as N^2 , but a full computation is not always necessary for determining only closely separated distance correlations. Due to the *Wiener-Kinchin* theorem, the Fourier Transform of the data itself, calculated as a power spectrum of magnitude squared, is equal to the Fourier Transform of the autocorrelation, so that an inverse Fourier Transform of the directly calculated power spectrum will recover an autocorrelation. Due to the efficiency of a FFT, the computational time is order $N \times \log(N)$, so this is often used to produce an ACF and a PSD with only frequency spectrum tools.

The power spectrum and autocorrelation are related by the following equation:

$$S_{xx}(f) = \int_{-\infty}^{\infty} r_{xx}(\tau) e^{-2\pi i f \tau} d\tau$$

Where the autocorrelation is defined as

$$r_{xx}(\tau) = \mathbf{E}[x(t) \cdot x^*(t - \tau)]$$

The double xx indicates that the expected value is between pairs of points along the terrain, with the τ serving as the sample-to-sample distance measure. (The asterisk indicates the complex conjugate, which does not apply for these signals.)

Foundation of Stochastic Model Analysis

The analysis of real-world PDF's is aided by the fact that independence is assumed in the sample space. Each draw from a PDF is by definition independent and *stationary* with respect to the sampling point. We don't always have to understand and assert independence but enough empirical studies have been done to understand when the premise will work and when it doesn't.

For distributions based on energetic processes, the *principle of maximum entropy* often results in parsimonious fits to collected data (later will show extended examples for ocean waves and terrain). The selection of the variate and how to apply the prior is important, as the shape of the distribution can be thin-tailed (exponential or Normal distributed) or fat-tailed (a power-law) depending on its modeled derivation. For example, a timing distribution may be fat-tailed because the analyzed variate is velocity (which would ostensibly give a thin-tail) but since time appears in the denominator with respect to velocity, the eventual PDF generates much more weight in the tail (i.e. a ratio distribution [3]).

For an ACF, the typical realization is via a random-walk model, often described by a *Markov* or *semi-Markov* process [11]. Higher order localized interactions beyond that, such as the near-neighbor memory-less Markovian random walk, can generate smoothed/filtered or ordered/periodic profiles, depending on the signs and strengths of the interactions. This has significance for decoding both an ACF and a PSD. An important consideration for autocorrelations is the concept of *coherence* length. A coherence length is that distance at which long-range correlations cease to factor in. Beyond that point, the state of the system could just as easily be determined by a draw from a PDF. This has significant implications on whether to apply a Fourier Transform on a model to match a power spectrum or to simply treat the PSD as an inverted probability density function in the frequency spectrum.

The interplay between the use of PDF and ACF profiles for characterization is abetted by some very practical aspects of working in the spatial frequency domain. One novel way of looking at the problem derives from the world of diffraction spectroscopy, where because of the micro-scales involved, the only way possible to get insight is to immerse oneself in the inverse of the spatial domain, into the spatial frequency domain, in what is known as *reciprocal space*. [12]

We will work in reciprocal space here, not because we can't detect the spatial features with measurements, but because the convenient mating of stochastic processes combined with powerful spectral algorithms, allows us to work out an analysis with greater rigor and statistical accuracy. This has huge implications for generating synthetic terrains based on limited real-world data (for example, where the actual course is classified but the PSD is not).

Unified Autocorrelation Analysis

Our analysis based on characterizing stationary profiles. This approach is fully documented elsewhere[12] in which we derive the *pair correlation* (autocorrelation) functions of arbitrary surfaces. The derivation assumes a path in essentially a 2-dimensional slice. One dimension is the distance along the path of traversal and the other dimension is the elevation of the point along that path.

A surface is constructed with N_c rigid cells arranged in the x- and z-directions. In a discrete approximation, the cell spacing in the x-direction is a , and in the z-direction is d . This surface is described by a function $f(x, z)$ which is equal to 1 if there is a surface cell at coordinates (x, z) and 0 otherwise. The Fourier power spectrum is defined as:

$$I(\vec{S}) = \left| \sum_{\vec{r}} f(\vec{r}) e^{-i\vec{S} \cdot \vec{r}} \right|$$

Where \vec{S} is the vector wavenumber defined in the reciprocal space of (x, z) as (S_x, S_z) . This can be rewritten as the equivalent Weiner-Khinchin relation:

$$I(\vec{S}) = N_c \left| \sum_{\vec{u}} C(\vec{u}) e^{-i\vec{S} \cdot \vec{u}} \right|, \quad N_c \rightarrow \infty$$

Where \vec{u} is a real-space vector and

$$C(\vec{u}) = \frac{1}{N_c} \sum_{\vec{u}} f(\vec{r}) \cdot f(\vec{r} + \vec{u})$$

is the pair correlation function on the surface. It is the probability of finding two cells on the surface separated by a vector \vec{u} . By including the array of cells as a sum of delta functions, this equation can be changed to the integral form:

$$I(\vec{S}) = N_c \int_{-\infty}^{\infty} dx e^{-iS_x x} \sum_l e^{-iS_z l d} \left[\sum_n \delta(x - na) C(x, ld) \right]$$

Where $C(x, ld)$ is the continuum portion of $C(\vec{u})$ written expressed along the two dimensions of interest. This can be rewritten making use of the *convolution theorem* as:

$$I(\vec{S}) = N_c \left[\int_{-\infty}^{\infty} dx e^{-iS_x x} \delta(x - na) \right] \otimes \left[\sum_l e^{-iS_z ld} \int_{-\infty}^{\infty} dx e^{-iS_x x} C(x, ld) \right]$$

Where \otimes is the convolution operator, which expands the embedded summations. By taking the Fourier transform of the first term:

$$I(\vec{S}) = \left[\frac{2\pi N_c}{a} \sum_n \delta(S_x - \frac{2\pi n}{a}) \right] \otimes \left[\sum_l e^{-iS_z ld} \int_{-\infty}^{\infty} dx e^{-iS_x x} C(x, ld) \right]$$

The second term repeats along the periodicity of the cell spacing, which is the result of a discrete representation, and is what we are interested in for a continuous system. Expanding the pair correlation term $C(x, ld)$, we first consider a surface in which the surface cells are allowed to be on any of an infinite number of levels as illustrated below. (Note that infinite specifically applies to the number of levels; for finite level systems, only the x-distance is infinite.)

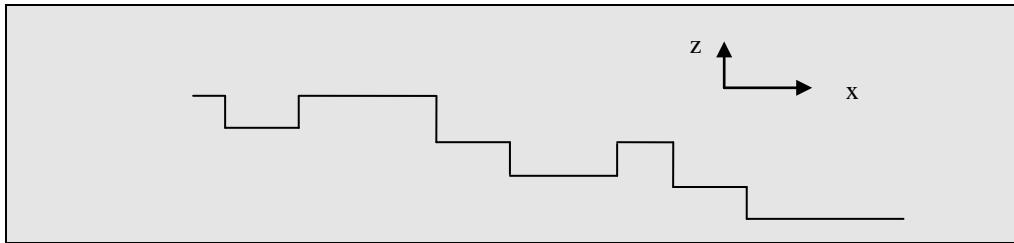


Figure 1: A surface composed of terraces separated by levels

The probability of finding a cell at the origin and at a position (x, ld) away is the sum of the probabilities of all possible configurations of steps that separate these two points. For example, assume for the moment that all steps are same height and are always in the same direction as in the classic descending stair. And suppose that a sequence of levels with lengths (L_1, L_2, \dots, L_n) separate a cell at an origin that is L_0 away from the first step from a cell at (x, ld) which is L_n away from the last step. Then the probability of this configuration is

$$P_0(L_0) P_1(L_1) \cdots P_{n-1}(L_{n-1}) P_n(L_n) \Delta L_0 \Delta L_1 \cdots \Delta L_{n-1} \Delta L_n$$

Where the $P_i(L_i)$ with $i = 1, 2, \dots, n-1$ are the probabilities per unit length of a terrace of length L_i occurring on the i^{th} level of the surface. The function $P_0(L_0)$ is the probability per unit length that there is a cell at the origin that is L_0 before the first step. The function $P_n(L_n)$ is the probability that there is a cell L_n away from the last step. The middle $n-1$ $P_i(L_i)$ are identical and equal to $P(L_i)$ for the infinite surface. This equation implicitly assumes that the “terrace” lengths are statistically independent from one another. The length of one does not determine the length of subsequent terraces except in so far as the sum of the L_i equals the total x-coordinate and that the number of steps is l , which is equal to n for the strictly descending staircase. To determine the pair correlation function, one must integrate over all possible configurations of steps that satisfy these constraints.

If in addition, we allow the step heights to be any multiple of the level separation, either positive or negative, with probability distribution $H(h_i)$ for a step of height $h_i d$, then the correlation function $C(x, ld)$ is

$$\begin{aligned}
 C^+(x, ld) = & N_l \left\{ P_{0f}(x) \delta_{l,0} \right. \\
 & + \sum_{n=1}^{\infty} \sum_{h_i=}^{\infty} \cdots \sum_{-\infty}^{\infty} \int_{L_i=}^{\infty} \cdots \int_0^{\infty} P_0(L_0) \times H(h_1) P(L_1) \cdots P(L_{n-1}) H(h_n) P_f(L_n) \\
 & \left. \times \left[\delta\left(x - \sum_{i=0}^n L_i\right) \delta\left(x - \sum_{i=1}^n h_i\right) dL_0 \cdots dL_n \right] \right\}
 \end{aligned}$$

This formulation is equivalent to a convolution integral of all the possible configurations of steps. The right hand side of this equation integrates over all possible configurations of terrace lengths, L_i , and sums over all possible step heights, $h_i d$, which will add to give a displacement (x, ld) from the origin. The plus sign in the superscript indicates the correlation function for the positive x and z -directions; to obtain the negative directions, simply replace x by $-x$ and z by $-z$. N_l is the total number of levels involved and is required to approach infinity. The function $P_{0f}(x)$ is the probability that there is a cell at the origin and one x away on the same terrace with no jumps in between.

The two distribution functions $H(h)$ and $P(L)$ are normalized according to

$$1 = \sum_{h=-\infty}^{\infty} H(h). \quad 1 = \int_{\tau=L}^{\infty} P(\tau) d\tau$$

The functions $P_0(L)$ and $P_f(L)$ can be written in terms of the terrace length distribution function $P(L)$. $P_0(L)$ is the probability that there is a cell at the origin and the first step is L away. It is given by the product of the probability that there is a cell at the origin, which is given by the coverage θ , and the number of configurations that allow the first step to be at L . For the infinite staircase, the coverage is $1/N_l$ and

$$P(L) = \frac{\theta}{\langle L \rangle} \int_{\tau=L}^{\infty} P(\tau) d\tau$$

Where

$$\langle L \rangle = \int_{\tau=L}^{\infty} \tau P(\tau) d\tau$$

To find $P_f(L)$ it is important to realize the asymmetry of the method. Since the probability of being at step n is 1 by construction, $P_f(L)$ is simply given by the probability that the step is followed by a terrace of length $\tau > L$. Thus it is given as

$$P_f(L) = \int_{\tau=L}^{\infty} P(\tau) d\tau$$

For the important case of no level changes, or the $n = 0$ portion of the summation, $P_0(L_0)$ and $P_f(L_n)$ must be replaced by

$$P_{0f}(L) = \frac{\theta}{\langle L \rangle} \int_{\tau=x}^{\infty} (\tau - x) P(\tau) d\tau$$

as this is the correlation function of a single level. Similar derivations of the preceding probabilities applied to a microscopic domain can be found in [12].

To calculate the PSD, we want the Fourier transform of the correlation function for $C^+(x, ld)$. For this, first substitute the delta function representations

$$\delta\left(x - \sum_{i=0}^n L_i\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dS_x e^{-iS_x(x - \sum_{i=0}^n L_i)}$$

$$\delta\left(l - \sum_{i=0}^n h_i\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dS_z e^{-iS_z(l - \sum_{i=0}^n h_i)}$$

into the long equation and define $P(S_x)$ and $H(S_z)$ by

$$P(S_x) = \int_0^{\infty} P(L) e^{-iS_x L}$$

$$H(S_z) = \sum_{-\infty}^{\infty} H(h) e^{-iS_z h}$$

to be the Fourier transforms of the individual probability density functions. Also, the Fourier transforms of the other probability functions can be written in terms of $P(S_x)$ according to

$$P_0(S_x) = \frac{\theta}{iS_x \langle L \rangle} [1 - P(S_x)]$$

$$P_f(S_x) = \frac{1}{iS_x} [1 - P(S_x)]$$

$$P_{0f}(S_x) = \frac{\theta}{S_x^2 \langle L \rangle} [1 - P(S_x)] + \frac{\theta}{iS_x}$$

Then the correlation function becomes

$$C^+(x, ld) = N_l \left\{ P_{0f}(x) \delta_{l,0} + \frac{1}{4\pi^2} \sum_{n=1}^{\infty} \left[\int_{-\infty}^{\infty} dS_z e^{-iS_z ld} [H(S_z)]^n \right] \times \left[\int_{-\infty}^{\infty} dS_x e^{-iS_x x} P(S_x) (P(S_x))^{n-1} P(S_x) \right] \right\}$$

Taking the transform to generate a PSD, we obtain:

$$I_0(S_x, S_z) = 2 N_l \operatorname{Re} \left\{ P_{0f}(S_x) + \sum_{n=1}^{\infty} [H(S_z)]^n P_0(S_x) [P(S_x)]^{n-1} P_f(S_x) \right\}$$

Here, the subscript 0 on the power spectrum intensity indicates the Fourier transform of the continuum part of the correlation function. Calculating the geometric sum, the result is

$$\begin{aligned} I_0(S_x, S_z) &= 2 N_l \operatorname{Re} \left\{ P_{0f}(S_x) + H(S_z) P_0(S_x) P_f(S_x) \sum_{n=0}^{\infty} [H(S_z) P(S_x)]^n \right\} \\ &= 2 N_l \operatorname{Re} \left\{ P_{0f}(S_x) + \frac{H(S_z) P_0(S_x) P_f(S_x)}{1 - H(S_z) P(S_x)} \right\} \end{aligned}$$

$$= \frac{2 N_l}{S_x^2 \langle L \rangle} \operatorname{Re} \left\{ 1 - P(S_x) - \frac{H(S_z) (1 - P(S_x))^2}{1 - H(S_z)P(S_x)} \right\} \quad (1)$$

One simple realization of a surface forms a staircase of steps. Each step descends or ascends in one direction. For this case, the transform of the step height distribution function, $H(S_z)$, becomes $e^{-iS_z d}$. We also assume that the staircase is not perfectly regular so that the sum in the equation always converges.

$$I_0(S_x, S_z) = \frac{2}{S_x^2 \langle L \rangle} \frac{1 - |P(S_x)|^2}{|1 - P(S_x)e^{-iS_z d}|^2} \quad (2)$$

If we then flatten the staircase, this emulates a sawtooth pattern of topography, either a set of breaking waves or a graded terrain with that structure. To level the reciprocal space, all we need to do is follow a path along the equality $x+z/\theta$, as shown in the figure below. This maps the average terrain slope onto an affine transformed coordinate system that is flat from the perspective of the viewer. In reciprocal coordinates, the transformed path is $S_x + S_z \theta$

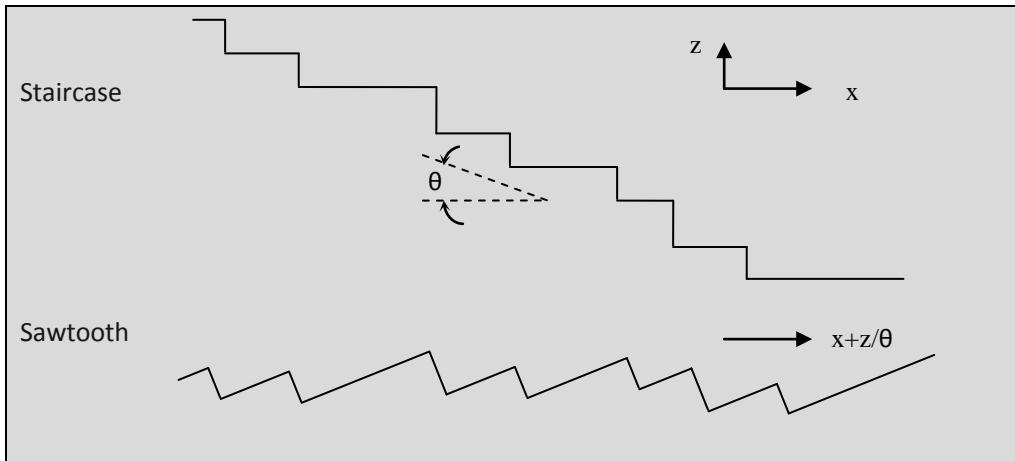


Figure 2: A descending staircase is used to derive an autocorrelation and then the coordinate system is rotated to model a sawtooth terrain profile.

The general approach falls under the categorization of a *semi-Markov* analysis. We allow the distribution for $P(S_x)$ and $H(S_z)$ to take on any form. To recover the elementary Markov spectrum, we apply a maximally disordered distribution to $P(L)$.

$$P(L) = \frac{1}{\langle L \rangle} e^{-L/\langle L \rangle}$$

This selection results in the Fourier transform for $P(S_x)$

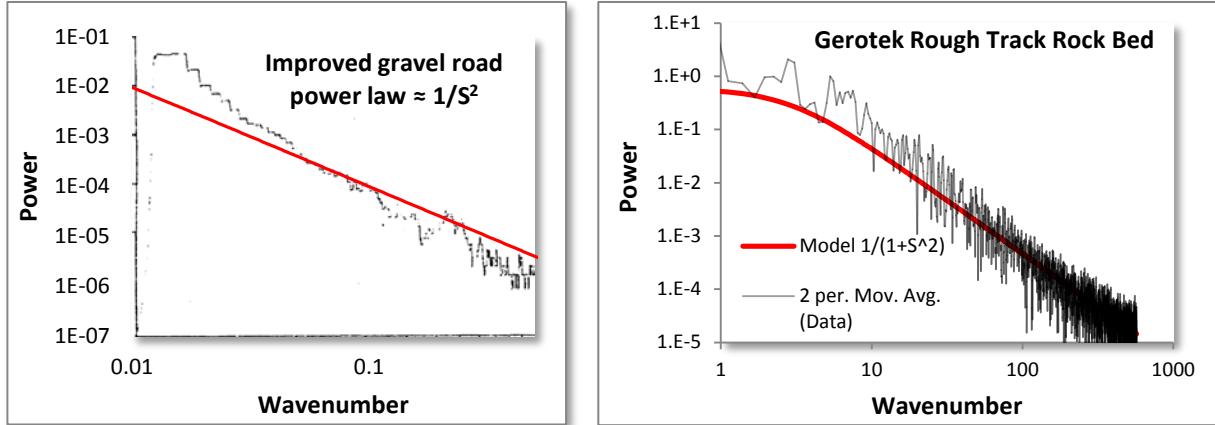
$$P(S_x) = \frac{1}{1 + iS_x \langle L \rangle} \quad (3)$$

Applying the same transform to the step height, the resulting PSD is the Cauchy or Lorentzian profile:

$$I_0(S_x) = \frac{1}{4 + \langle L \rangle S_x^2} \quad (4)$$

This has the classic PSD second-order power-law fall-off of a random terrain, shown below. The deviations from this fall-off are due either to further order in the terrain relief or filtering of the data set.

This particular data was extracted from an unclassified military test course and from data gathered by one of the authors[13][14] on the Gerotek facility vehicle testing track².



*Figure 3: (left) Power Spectral Density (PSD) plot from a declassified military test course, with a random walk terrain profile overlaid as $1/S^2$.
 (right) High Resolution PSD of Gerotek rough track rock bed course, with Cauchy model overlay*

The data and spectra from Figure 3 are taken from a historical unclassified test document. Figure 4 presents data from the present day “rough road” test track data from a Mercedes-Benz course[15], note the reduced fluctuation noise and wide dynamic range in the data.

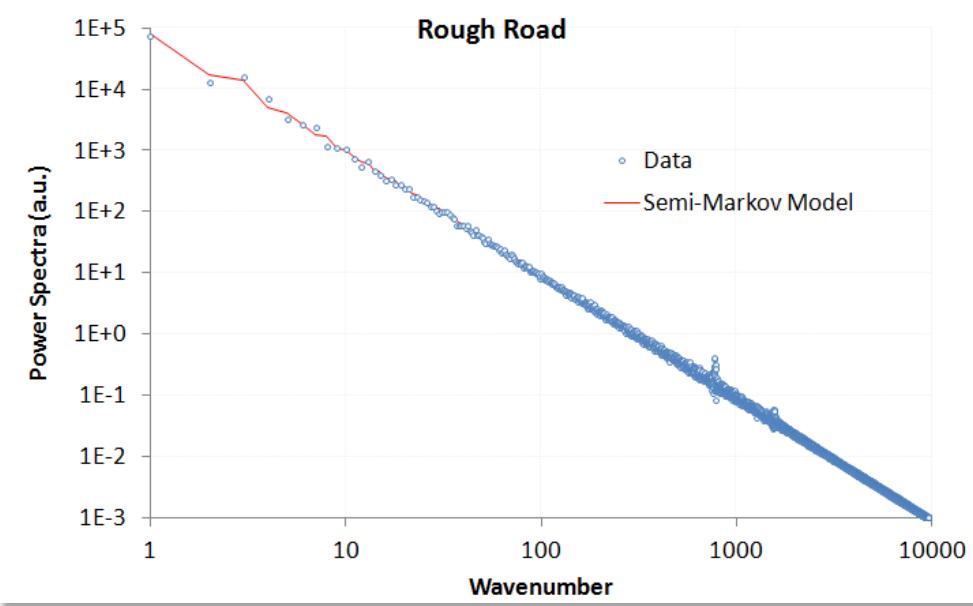


Figure 4: PSD of data collected from a higher resolution and more extensive terrain data set from a Mercedes-Benz test track. Note the weak high frequency spike with a definite even harmonic signal. The overall tendency of the terrain profile follows a random semi-Markov model very well.

² <http://www.gerotek.co.za/>, Pretoria, South Africa

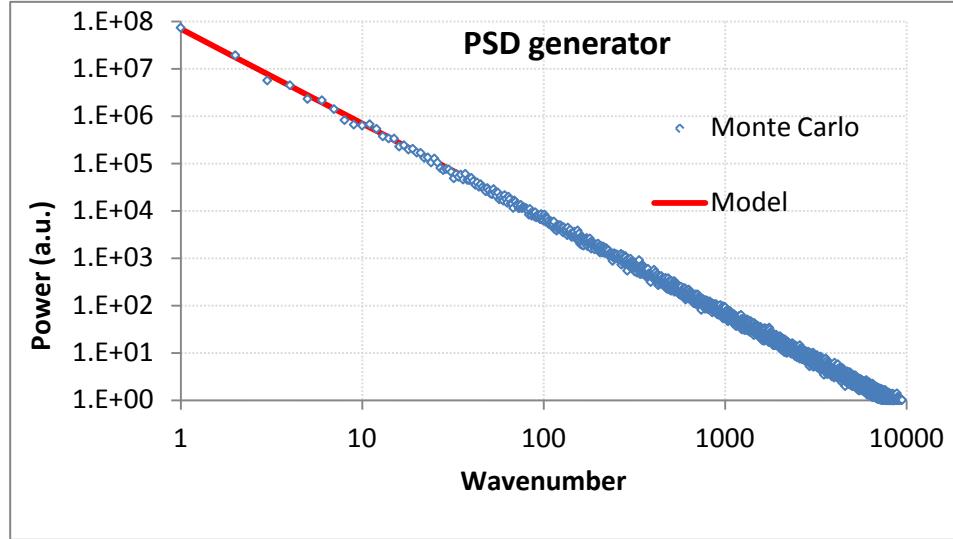


Figure 5: Monte Carlo generated course based on random walk model

The power of the analysis is that it can detect asymmetries in the surface relief. For example, a terrain that shows sawtooth waves will contain even harmonics (note the weak harmonic in Figure 4), while a terrain that contains symmetric square or rolling pseudo-sinusoidal waves will contain odd harmonics.

Additionally, for the surface shown above, much detail may be obscured by a long-wavelength trend in the underlying terrain. The slope for the “rough road” example of Figure 4 is fairly large and this contributes to a strong $1/S^2$ tail above a certain wave number, i.e. a linear slope in real space generates an inverse squared response in reciprocal space according to Fourier analysis. This can also be inferred by the reduction of noise at high wave numbers, since a static slope is deterministic and dominates over the stochastic fluctuations. This also explains why a course profile needs to be “detrended” for it to be practical for determining actual roughness. A non-detrended course will generate a $1/S^2$ and provide less value in extracting the smaller scale roughness.

Regular features: If on occasion, we come across a terrain structure with extremely regular or periodic features (such as a highway rumble strip or a lengthy grated section), the PSD calculation simplifies to a Fourier series.

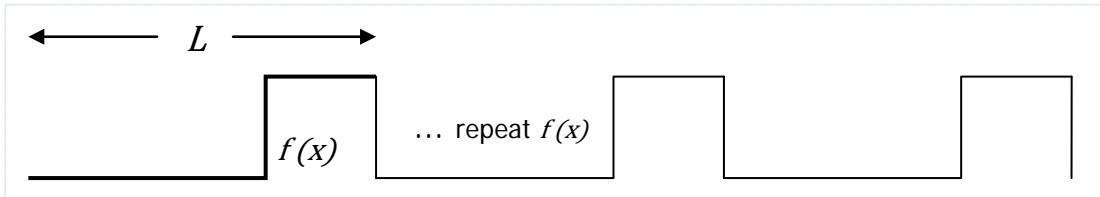


Figure 7: A perfectly ordered sequence of steps of periodicity L

This reduces to the following intensity, with the PSD described as a series of harmonic peaks of very narrow width modulated by an envelope determined by the Fourier transform of one cycle

$$I(S_x) = |F(S_x)|^2 \sum_{i=0}^{\infty} \delta(S_x - \frac{2\pi}{L}i) \quad \xrightarrow{\text{where}} \quad F(S_x) = \int_0^L f(x) e^{-is_x x} dx$$

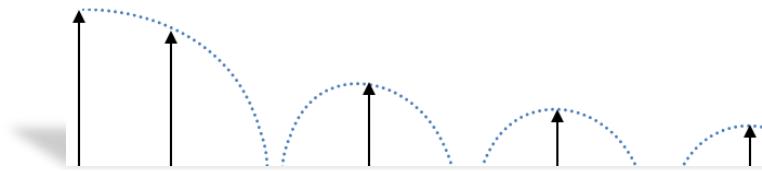


Figure 8: The PSD of ordered steps has delta-function harmonic peaks, modulated by the Fourier transform envelope of a single period.

These insights derive from a the Fourier series decomposition of periodic waveforms, but our goal is to apply this technique to stochastically varying waveforms – as stochastic profiles are very prevalent in artificial (see the following figure) and natural terrain. In the following sections, we will look more closely at the spectral features of disordered aquatic and land terrain.



Figure 9: Examples of roads with random and pseudo-periodic cracks and patches

Application to Aquatic Waves

We first consider aquatic wave spectra because these have a more consistent character than land terrain. Gravity plays a critical role as it automatically detrends the data to maintain a level profile over the distances of interest. Sea waves also are very sensitive to long-range *coherence*, that is, the ability to maintain phase relationships over many wavelengths such as seen with capillary waves[16][17] or over a significant distance as with cnoidal swells[18]. The term coherence is defined as correlation when applied specifically to wave-like properties [19]. This is noteworthy also when one considers that a dispersion relation holds between the spatial frequencies and temporal frequencies of aquatic waves.

Spatially Incoherent Wave Spectra

In the wild, waves usually display little coherence over long spatial scales. In other words, the knowledge that one has about one wave has limited implications regarding another wave separated by several undulations.

We make a maximum entropy estimation of the energy of a one-dimensional propagating wave driven by a prevailing wind direction. The mean energy of the wave is a function of the square of the wave height, H .[20] This makes sense because a taller wave needs a broader base to support that height, leading to a scaled pseudo-triangular shape, as shown in the figure below.

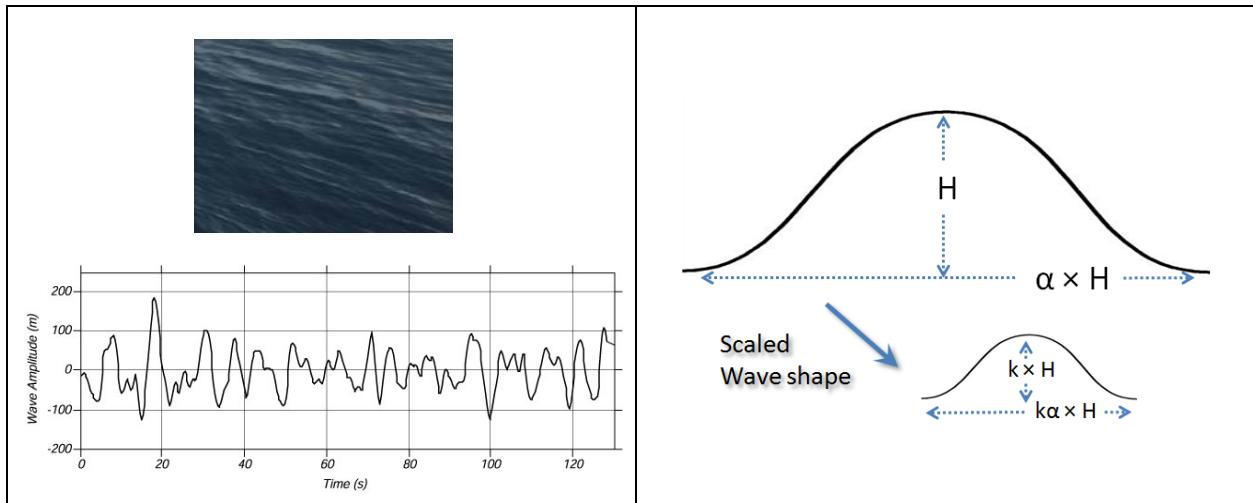


Figure 10: (left) The goal is to model the spectral density of waves. Enough disorder exists in open water that periodic coherence between spatially separate waves is not maintained, measurement from wave buoy in north Atlantic.(from [2]). (right) The initial assumption is that waves contain energy proportional to their width and height. This proportionality scales independent of volume. Total energy in a directed wave goes as the square of the height, and the macroscopic fluid properties suggest that it scales to size. This leads to a dispersive form for the wave size distribution.

Since the area of such a scaled triangle goes as H^2 , the MaxEnt cumulative probability is:

$$P(H) = e^{-aH^2}$$

where a is related to the mean energy of an ensemble of waves. This relationship is empirically observed from measurements of ocean wave heights over a sufficient time period. However, we can proceed further and try to derive the dispersion results of wave frequency, which is the typical oceanography measure. So we consider — based on the energy stored in a specific wave — the time, t , it will take to drop a height, H , by the Newton's law relation:

$$t^2 \sim H$$

and since t goes as $1/f$, then we can create a new PDF from the height cumulative as follows:

$$p(f)df = \frac{dP(H)}{dH} \frac{dH}{df} df$$

where[2]

$$H \sim \frac{1}{f^2}$$

$$\frac{dH}{df} \sim -\frac{1}{f^3}$$

then

$$p(f) \sim \frac{1}{f^5} e^{-c/f^4} \quad (5)$$

which is just the Pierson-Moskowitz wave spectra that oceanographers have observed for years (developed first in 1964, variations of this include the JONSWAP, Bretschneider and ITTC wave spectra[21]).

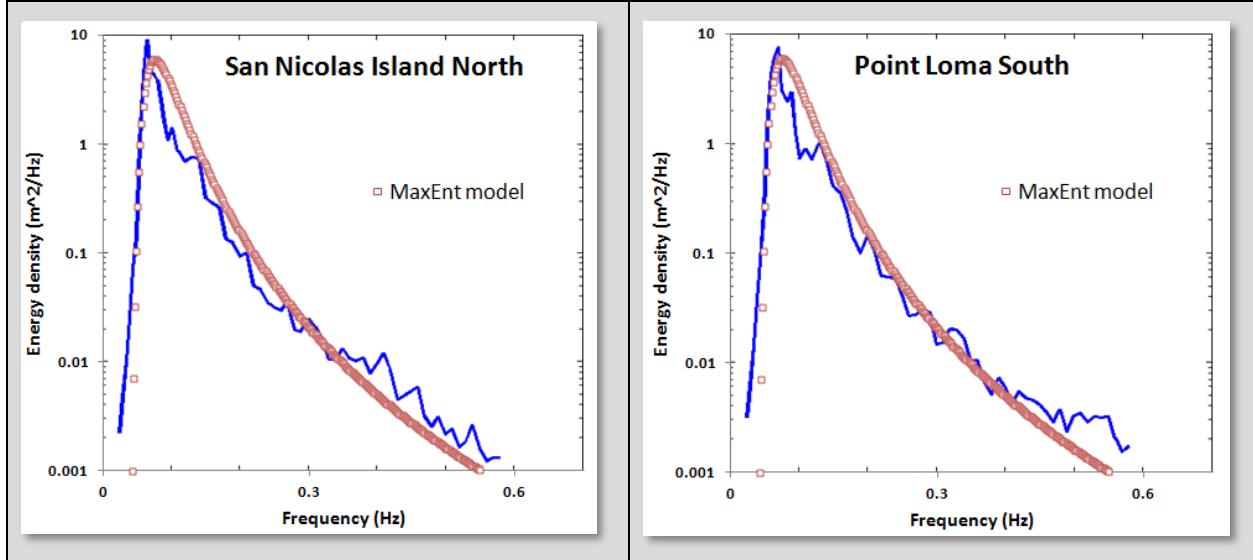
This concise derivation works well despite not applying the correct path of calculating an auto-correlation from $p(f)$ and then deriving a power spectrum from the Fourier Transform of $p(f)$. As smooth ocean waves approach a sinusoidal ideal, a single wave will contribute a Fourier component at that frequency alone and the spectrum can be evaluated almost by inspection. This convenient shortcut remains useful in understanding the simple physics and probabilities involved.

The utility of this derived form can be demonstrated using data[22] obtained from public-access stations. The following data is extracted from a pair of measuring stations located off the coast of San Diego [23].



Figure 11: The CDIP data source provided archival wave energy statistics for assessing models.

The data shown here is averaged wave spectra for the entire day 1/1/2012. The red points in Figure 12 correspond to best fits from the derived MaxEnt algorithm applied to the blue data set.³



*Figure 12: Wave energy spectra from two sites off the San Diego coastal region.
The Maximum Entropy estimate is in red.*

Like many similar spectra such as wind and EMI, the wave spectrum derives simply from maximum entropy conditions.

Note that we did not need to invoke the full spectral decomposition model presented earlier in this paper. The correlations are short over the sinusoidal shape of an individual wave and so those frequency components show up strongly in the energy density PSD. The implications are that the driving forcing function needed to provide order in the case of natural waves is missing, and this makes sense as the main stimulus, wind energy, on its own is highly disordered. The wind simply stimulates the wave to maximize its entropy subject to the kinetic and potential energy that are provided.

On the other hand, under controlled or reinforcing (positive interference) conditions, more order can be supplied to make the waves appear more coherent over a spatial distance.

Spatially Coherent Wave Spectra

Coherent waves are easier to generate in the laboratory than in nature, apart from the occasional cnoidal swells peculiar to a region. A large wave tank, with waves generated by a consistent wind will clearly expose any ordered features in the PSD [24].

In Figure 13 below, taken from [25], the measured PSD clearly shows clear harmonic peaks strongly suggestive of longer-range order in the wave periodicities. Superimposed on the data profile (shown in black) is a model adapted from the derivation of the previous section of this paper. Note that from the spatial/temporal dispersion relation, the time- domain frequency scale with the spatial-domain frequencies, as short, choppy waves have much smaller periods or cycle-times than the large, rolling

³ The dataset is available from :
http://cdip.ucsd.edu/?nav=historic&sub=data&units=metric&tz=UTC&pub=public&map_stati=1,2,3&sln=167&stream=p1&xymo=201201&xitem=product25

waves. This means that we can apply a spatial analysis to infer the temporal characteristics with some generality (and vice versa).

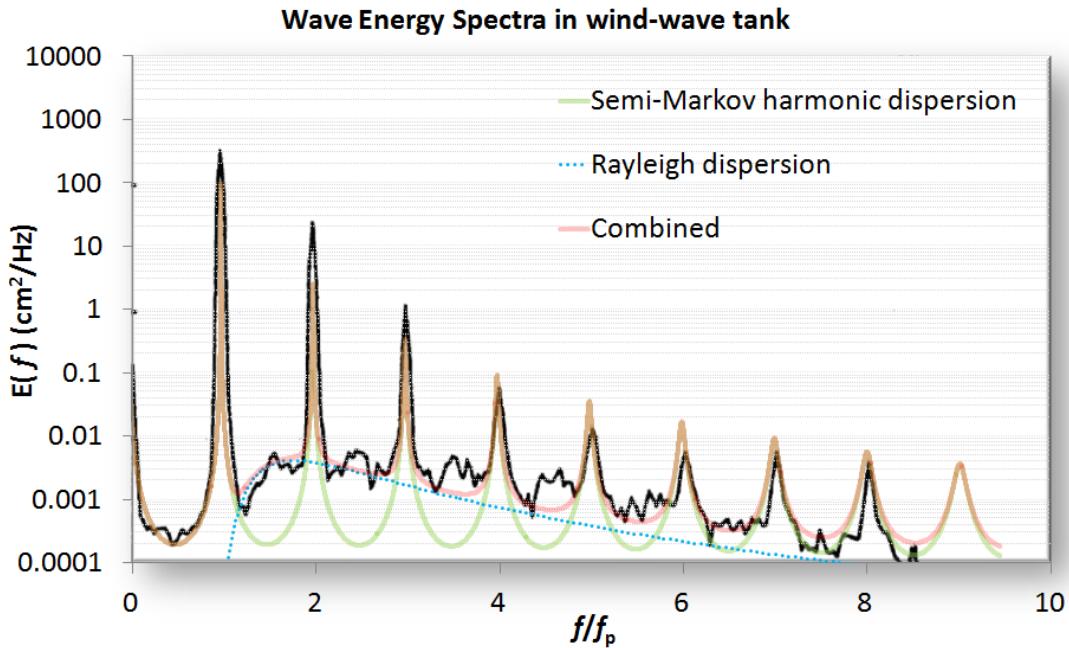


Figure 13: Highly ordered waves with all harmonics generated in a wave tank. This is in terms of temporal frequency and the spatial frequency can be recovered via the dispersion relation.

The basic model is to choose a $P(L)$ wave density such that it reproduces the spacing and envelope of the harmonics. One of the simplest density functions is known as the shifted exponential. This maintains a minimum spacing with the possibility of an occasional longer wavelength:

$$P(L) = \begin{cases} 0, & L < L_0 \\ e^{-\alpha(L-L_0)}, & L > L_0 \end{cases} \quad (6)$$

This generates a Fourier transform

$$P(S_x) = \frac{e^{-iS_xL_0}}{1 + iS_x/\alpha}$$

And then the full power spectrum assuming a coherent relationship exists (see **Equation 1**):

$$I(S_x) = \frac{1}{(S_x - \alpha \sin(S_x L_0))^2 + \alpha^2(1 + \cos(S_x L_0))^2} \quad (7)$$

$$I(S_x) = \frac{1}{(S_x + \alpha \sin(2S_x L_0))^2 + \alpha^2(1 - \cos(2S_x L_0))^2} - \frac{1}{S_x^2(1 + \alpha L_0)^2} \quad (8)$$

These two formulations, scaled accordingly, generate the semi-Markov harmonic dispersion model shown in Figure 13. The first equation (7) generates an odd-harmonic waveform due to a symmetric view of up and down steps. The second equation (8) assumes an all-harmonic composition derived by assuming a surface with a saw-tooth set of steps, which is likely nearer the real situation for stimulated waves.

Note that as L_0 approaches zero, then the Markov random walk is recovered (see equation 4). Thus a simple parametric model with the two coefficients related by a *characteristic period*, $\langle L \rangle$, can emulate the continuum of order to disorder:

$$\langle L \rangle = L_0 + 1/\alpha$$

As L_0 approaches $\langle L \rangle$ the spectrum will show strong harmonics. As L_0 approaches 0, the random aspect will suppress the harmonics. As an example of this continuum consider Figure 14 which shows very weak harmonics and higher disorder than the waves of Figure 13: [26]

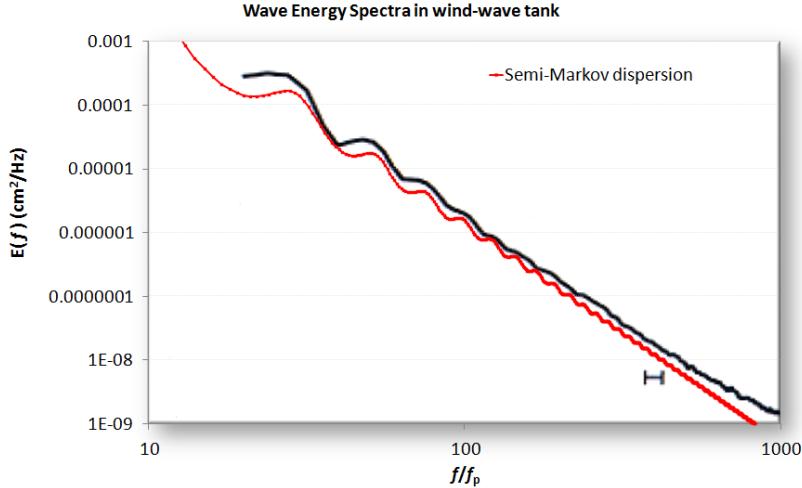


Figure 14: A PSD from a wave tank showing ripples

Note that the dispersion model does, by definition, generate square or sharp edges. Unless there are breakers in the waves, the actual profiles will get rounded. By applying a low-pass filter to the model, the actual waveform will reveal reduced high frequency components and give better agreement with the spectral data. Another example, fit from a simulated wave [27] is shown in Figure 15:

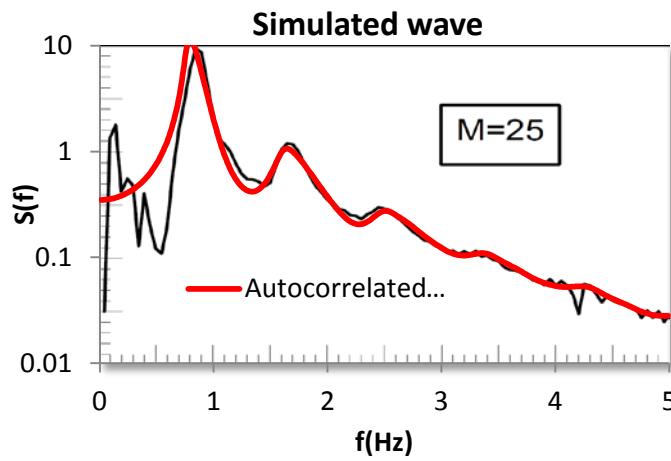


Figure 15: A simulated wave which emulates crest-focused dynamics shows intermediate order

In general, aquatic wave coherence showing strong harmonics occurs under more controlled conditions than normally exist in the wild.

Application to Landforms and Terrain

For terrains we use the same approach as for wave spectra. The significant geometric scale distinction between terrain profiles and aquatic wave profiles is that the former can have very long range variations, stretching to the scale of mountain ranges.

Incoherent Terrain Correlations

Statistics over an extensive large-scale terrain database provides the best example of incoherent variations in surface topography. Figure 16 derives from a detailed terrain slope case study reported in a previous study [6]. The derivation assumed that maximally sloped regions had larger potential energy and a maximum entropy (MaxEnt) statistical distribution would follow. The chart in Figure 16 is from a single region and that of Figure 17 from the entire country. The former data follows an exponential distribution and the latter super-statistical data follows a Bessel distribution.

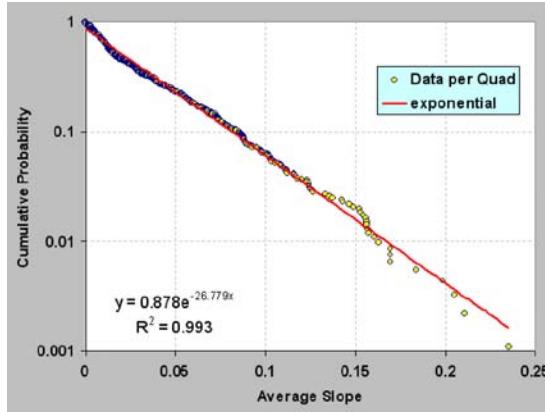


Figure 16 : Probability density function of terrain slopes for an isolated region

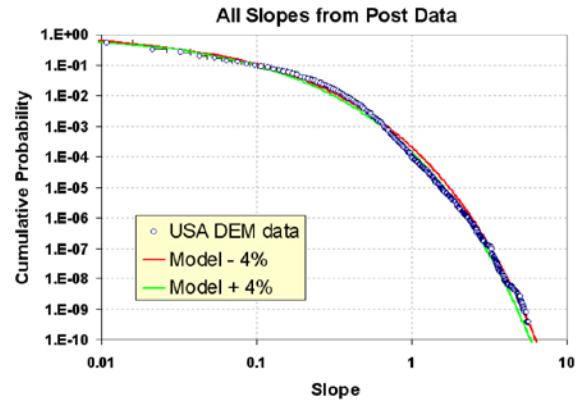


Figure 17 : Probability density function for a wide area

In this case, the term incoherent implies that the slopes at two points separated by a distance can lose correlation at wide spatial separations. In other words, the slope at one terrain point is independent of the value at another point. This is certainly an approximation that does not hold as the two points decrease in spatial separation and merge onto a single location.

Markov Terrain Correlations

We can use a random walk model as a very useful first-order approximation to characterize terrain correlations. Random walk applied in this way falls into the category of a Markov model, where the correlations occur between near-neighbor steps. In general, a pure random walk model featuring stochastic up-and-down steps will show an unbounded variance. This means that the probability of two points separated by any elevation is uniformly distributed between zero and infinity — *given a large enough surface separation*.

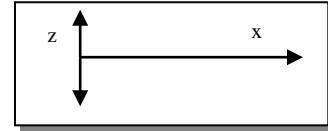
A pure random walk defined in this way is impractical and unphysical, as elevation differences are in fact finitely bounded. To allow for the boundedness of real terrains, a random walk process that has *reversion to the mean* properties is required. The classic reversion-to-the-mean variation to the random walk model is known as the Ornstein-Uhlenbeck process.

If one assumes the mean is the average terrain elevation for a localized region, the Ornstein-Uhlenbeck is the simplest stochastic behavior that will retain both Markov properties and a Gaussian spread in

elevation changes. It also generates the commonly observed damped exponential pair-correlation (or autocorrelation) function.

Given a stationary run of random walk transitions, there are known solutions to the Ornstein-Uhlenbeck process, calculated from solving the Fokker-Planck equation; this solution turns out fairly concise in terms of representation.

For a transition probability of elevation change (z) given a surface translation (x)



$$p(z|x) = \sqrt{\frac{\theta}{2\pi D(1 - e^{-2\theta x})}} \cdot e^{\left\{-\frac{\theta}{2D} \left[\frac{(z-\xi e^{-\theta x})^2}{1-e^{-2\theta x}} \right]\right\}}$$

Here, θ represents a drag term for reversion to the mean, and D is the random walk diffusivity or a relative hopping rate in a discrete simulation. The ξ term is an initial condition necessary for representing a starting point away from the mean. If we assume a stationary run, the ξ term disappears and the expression reduces to:

$$p(z|x) = \sqrt{\frac{\theta}{2\pi D(1 - e^{-2\theta x})}} \cdot e^{\left\{-\frac{\theta}{2D} \left[\frac{z^2}{1-e^{-2\theta x}} \right]\right\}}$$

The factor $(1 - e^{-2\theta x})/\theta$ serves as an asymptotic limit which prevents the elevation (z) excursions from getting too large, as a non-linear scaling factor gets applied to create an *effective* surface translation.

A normalized view of the marginal translation probability is shown in Figure 18 below, where y takes the place of z . Note that the contour profile shows a roughly parabolic shape and this shape begins to asymptotically approach a horizontal along the x -axis. In contrast, for a classic random walk which does not have Ornstein-Uhlenbeck reversion-to-the-mean tendencies, this profile would continue to rise with increasing x , in keeping with a Fickian square-root growth law characteristic of a pure diffusional process. It is this asymptotic behavior which keeps the variance of z -excursions bounded.

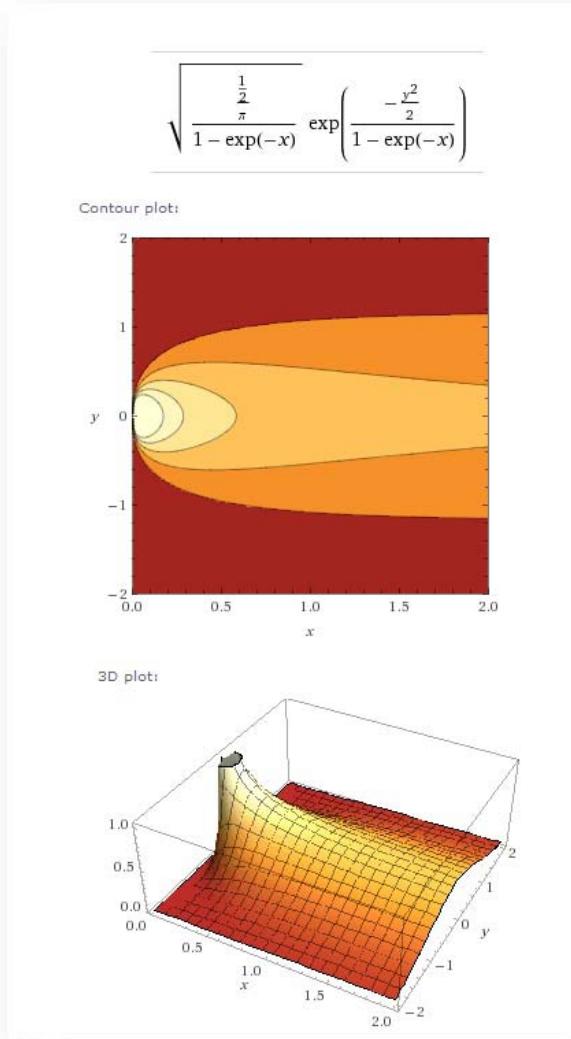


Figure 18: Contour plots of transition probability for the Ornstein-Uhlenbeck process

For an alternative view of what is happening, consider that the marginal transition probability for a pure random walk is given by:

$$p(x, z) = \frac{1}{\sqrt{2\pi D x}} e^{-\frac{z^2}{2Dx}}$$

An Ornstein-Uhlenbeck random walk assumes the transformation

$$x = \frac{(1 - e^{-2\theta \tilde{x}})}{2\theta}$$

This provides a reversion-to-the-mean property, which prevents large or infinite excursions from occurring in a pure unconstrained random walk.

For small x relative to $1/\theta$, this reverts to pure random walk, where

$$\tilde{x} = x$$

For the following derivation, we leave this in the pure random walk formulation and mix in a probability density function reflecting uncertainty in D . We initially assume maximum entropy uncertainty, where we characterize the value of D by its mean value $\langle D \rangle$.

$$p(D) = 1/\langle D \rangle \cdot e^{-D/\langle D \rangle}$$

Then by integration over marginal probabilities, we can estimate a “fuzzy” Ornstein-Uhlenbeck random walk PDF.

$$\begin{aligned} p(x, z) &= \int_{D=0}^{D=\infty} p(x, z|D) \cdot p(D) dD \\ p(x, z) &= \int_{D=0}^{D=\infty} \frac{1}{\sqrt{2\pi D x}} e^{-\frac{z^2}{2Dx}} \cdot \frac{1}{\langle D \rangle} \cdot e^{-D/\langle D \rangle} dD \end{aligned}$$

This looks like a complicated integral, not amenable to closed form evaluation since the variate D shows up in both the numerator and denominator of an exponential, as well as within a square root. Yet, this does indeed reduce to a very manageable expression thanks to a key integral identity, available from any comprehensive integration table reference. The resultant expression is

$$p(x, z) = \frac{1}{2\sqrt{\langle D \rangle x}} e^{-\frac{|z|}{\sqrt{\langle D \rangle x}}}$$

where $\langle D \rangle$ replaces the original D in the original random walk formulation, assuming the role of the mean diffusivity. No new parameters are added to the original, as we have simply swapped out certainty in D with uncertainty characterized by $\langle D \rangle$. To get to the Ornstein-Uhlenbeck model we simply apply the x -transformation.

$$x = \frac{(1 - e^{-2\theta \bar{x}})}{2\theta}$$

To mix-in any other probabilities, we simply need to factor in marginal prior probabilities for different values of $\langle D \rangle$. So for instance, assume that very low diffusion, D_0 , areas are mixed in by a fraction f , where $0 < f < 1$. Then

$$\begin{aligned} p'(x, z) &= f \cdot p(x, z) + (1 - f) \cdot p_0(x, z) \\ p'(x, z) &= \frac{f}{2\sqrt{\langle D \rangle x}} e^{-\frac{|z|}{\sqrt{\langle D \rangle x}}} + \frac{(1 - f)}{2\sqrt{D_0 x}} e^{-\frac{|z|}{\sqrt{D_0 x}}} \end{aligned}$$

In general, a model fit would require parametric values for $\langle D \rangle$ and θ , and potentially f and D_0 , if an extra inhomogeneity is suggested from the empirical data. The prime example of this would be flat regions that occur due to absorbing boundary conditions via a natural random walk, or due to non-random features such as bodies of water, graded roads, or flatter agricultural regions.

This foundational analysis gives us some room to work with when we try to characterize actually topographies. The premise is that the Ornstein-Uhlenbeck formulation with a single diffusivity D may map well to terrains with a homogeneous random character, while the maximum entropy “fuzzy” Ornstein-Uhlenbeck formulation works better with terrain regions that have mixed heterogeneous character, in the sense that the diffusivity $\langle D \rangle$ has a (prior) range in values which match the variability of the region.

We tested out the two formulations with respect to a set of digital elevation models (DEM) representing the lower 48 states. Each DEM file corresponded to a section that was 1° in latitude by 1° in longitude. This corresponded to approximately 90 meter post spacing in the north-south direction and a variant amount below that value in the east-west direction for increasing latitudes. Each “post” contains an elevation value with respect to sea-level reported to the nearest meter (there are 2401×2401 posts in total per DEM section).

What we wanted to demonstrate was how well an Ornstein-Uhlenbeck model works to describe correlated elevation transitions for relatively small surface translations (< 40 post spacings, or < 4 kilometers). The data from a DEM section was sampled uniformly to capture good counting statistics.

In Figure 19 below, raw histogram counts are shown for an area around Yuma, CA (the El_Centro DEM section). What is readily apparent is the trend toward a contour profile such as that shown in Figure 18.

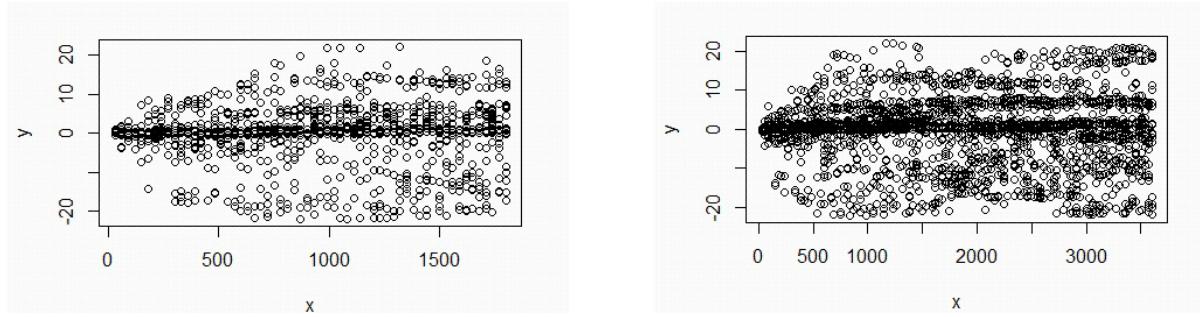


Figure 19: Raw histogram counts for a DEM section around Yuma, California.

A relief plot is shown below, along with a log-scaled contour plot:

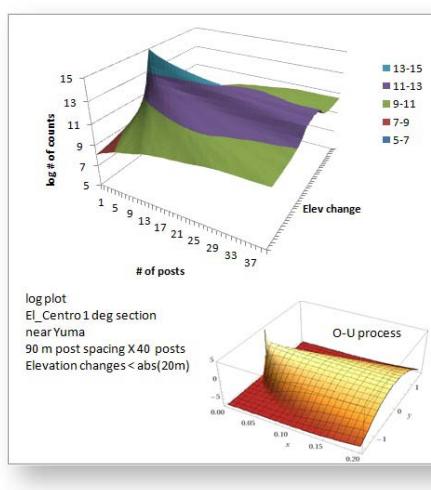


Figure 20: Relief plot

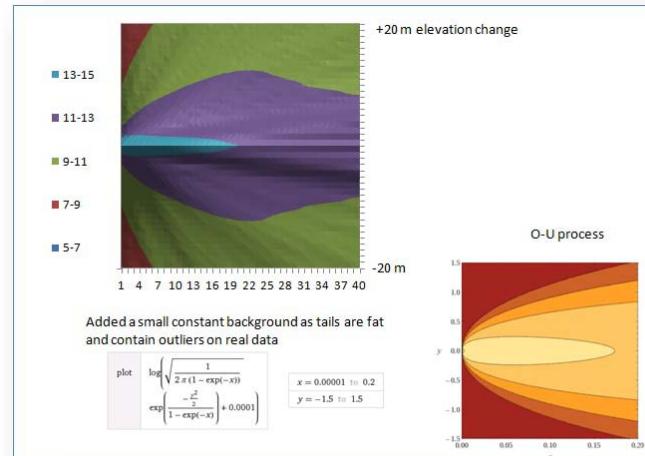


Figure 21: Contour plot of marginal probability

To demonstrate the relative fitting abilities of the conventional Ornstein-Uhlenbeck model against the maximum entropy O-U model, we searched the DEM database for examples that gave good qualitative fits to each model.



Figure 22

The *Andalusia-E* DEM section is located in mid-south Alabama, with terrain that is in between the flatness of the Gulf lowlands and the hilly terrain of the start of the southern Appalachians.

The *Eau Claire-W* DEM section is located in southwestern Wisconsin, with terrain that features hills and valleys that are remnants of one of the few un-glaciated areas of the upper Midwest. To get an appreciation for the topography, the following image is converted from a portion of this DEM located in southeastern Minnesota.

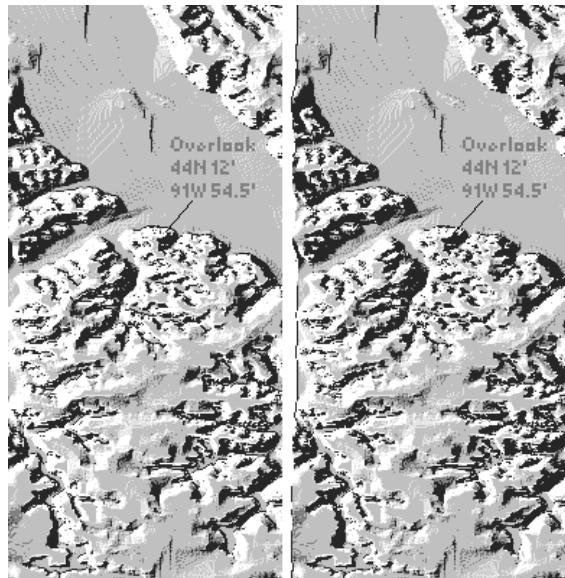


Figure 23 : A stereoscopic rendering of the southwest portion of the *Eau Claire-W* section. The GPS reading is next to the Upper Mississippi River National Wildlife and Fish Refuge with a view northeast across to Wisconsin.

The fitting of the appropriate O-U model to each data set was accomplished by minimizing the log of the error between the model and the data for each separation and elevation. For the pure Ornstein-Uhlenbeck process, Figure 24 illustrates the salient features, which includes a rounded profile in the relief plot, indicating a single diffusivity. Note that the plots are one-sided as the negative elevation excursion is removed due to symmetry.

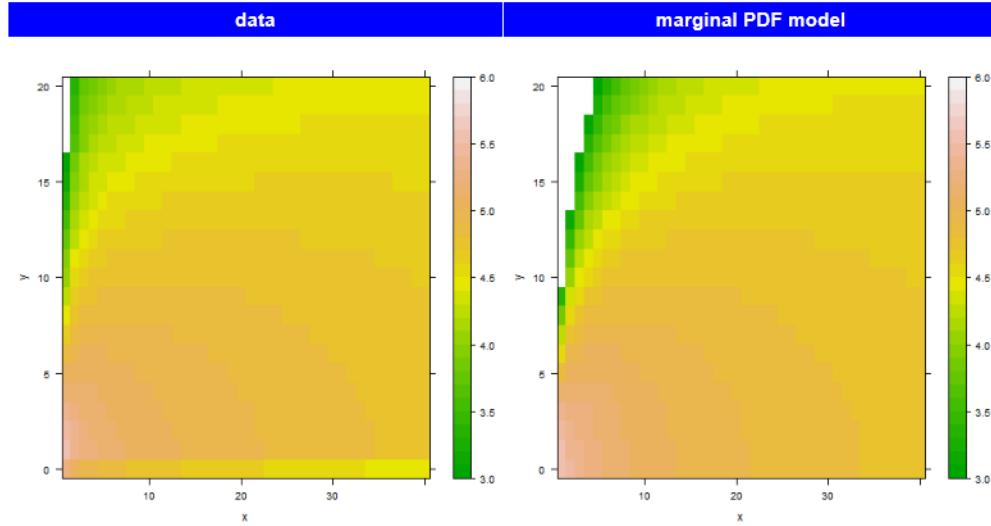


Figure 24: Andalusia-E contour plot

For the maximum entropy Ornstein-Uhlenbeck process, the profile shown in Figure 25 has more of a tail, indicating a spread in diffusivities. This indicates that the unglaciated region of the Eau Claire section has greater variability in its terrain makeup.

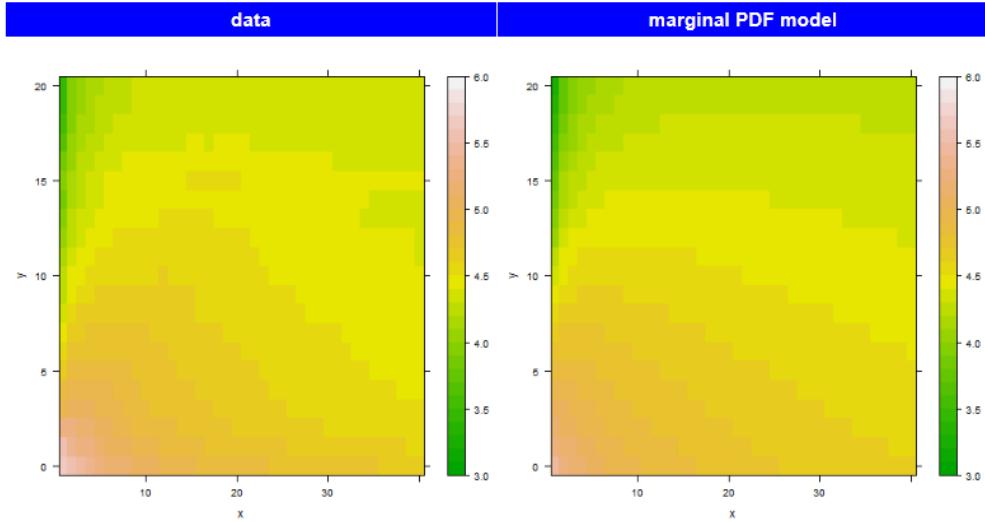


Figure 25: Eau Claire-W contour plot

In Figure 26, we place the data plots along the center to indicate which way the profiles tend toward. For the upper Andalusia contour, the color contours indicate closer agreement to the Ornstein-Uhlenbeck model, while for the lower Eau Claire data, the contour lines map more closely to the Max Entropy prior of the Ornstein-Uhlenbeck model.

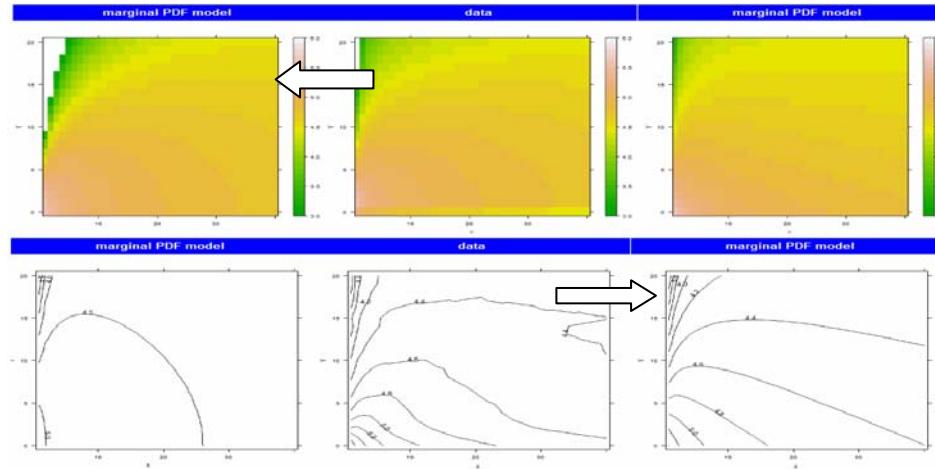


Figure 26: Regions trend either toward an Ornstein-Uhlenbeck profile (left column) or a Maximum Entropy prior of the O-U profile (right column).

A very strong case can be made for the effectiveness of the maximum entropy model in remote topography regions that contain little human terrain modification.

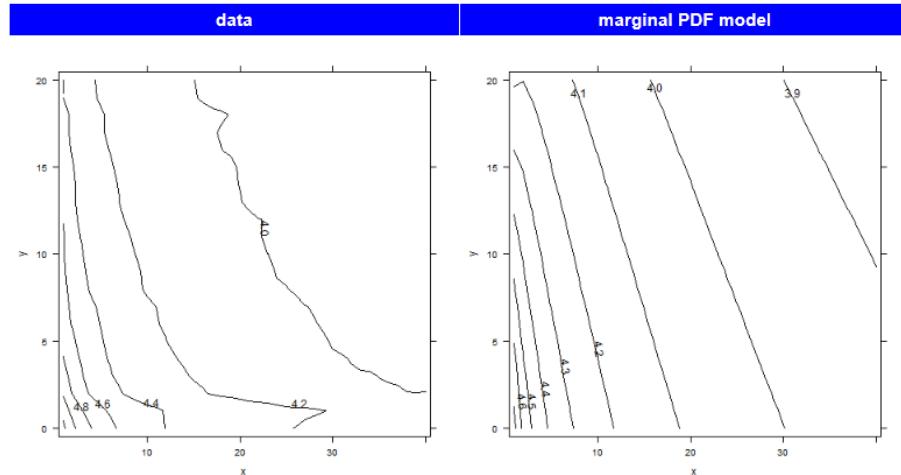


Figure 27: Knoxville-W contour plot of elevation difference probabilities. This is in the middle of the Great Smoky Mountains region, and has very low integrated error against the Maximum Entropy prior of the Ornstein-Uhlenbeck model.

We can use the maximum entropy formulation directly to calculate the censored (or truncated) variance and mean for the data sets and compare that to the theoretical values. A censored variance means that only the collected data points go into the calculation and those that contribute to the true variance ignored. Figure 28 shows the RMS deviation in elevation and the mean elevation as a function of surface translation for the *Knoxville-W* data set.

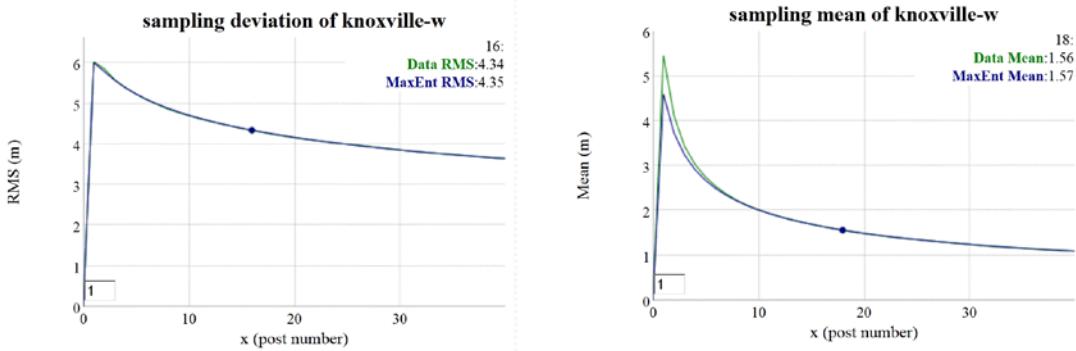


Figure 28: Sampling RMS deviation (left) and mean (right) of Knoxville section data against the Maximum Entropy prior of the Ornstein-Uhlenbeck model. The RMS deviation for both model and data align precisely enough that the curves cannot be distinguished.

The alignment between data and model of this measure mirrors that of the contour plots. This supports the idea that the terrain in these locations is well suited to a MaxEnt-varied random walk model. The simulation of the disordered random walk involves drawing a sample of a diffusion coefficient and then applying that to an Ornstein-Uhlenbeck random walk (see next section).

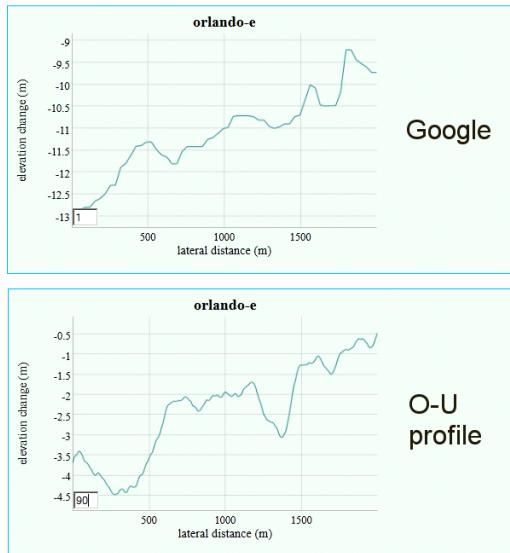


Figure 29: Simulated random walk applied to the Orlando DEM section. Above is an extracted Google elevation profile from a random location within that section. The scale of the excursions match the diffusivity of the random walk.

In addition, the effects of systemic errors and non-stochastic anomalies are readily apparent from the marginal probability density. If artificial correlations exist, they will appear as high density regions in the contour plot (see the strong horizontal bar at elevation changes of $\sim 15\text{m}$ shown in the right inset of Figure 30). The correlations in question come about from a propensity of surveying crews to report elevations to the nearest 50 foot round-off. For example, one can see that red dotted lines in Figure 30 passing through 1800', 1850', 1900', and 1950' contour levels map to elevation areas that tend to flatten out over short intervals. This can only be ascribed to poor data collection or data translation and is systematically observed in DEM data sets that were taken over the relatively flat terrain of the Midwest USA .

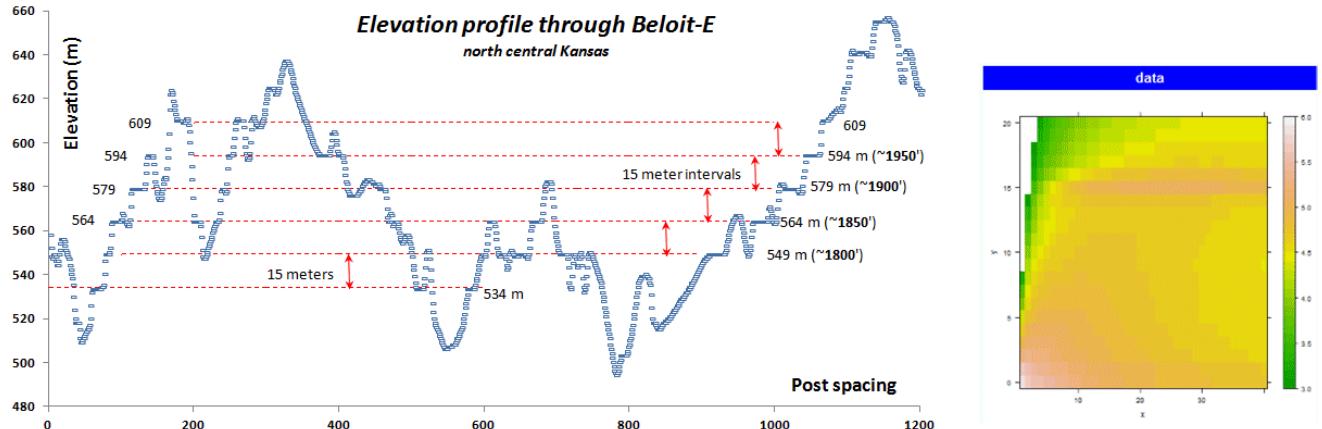
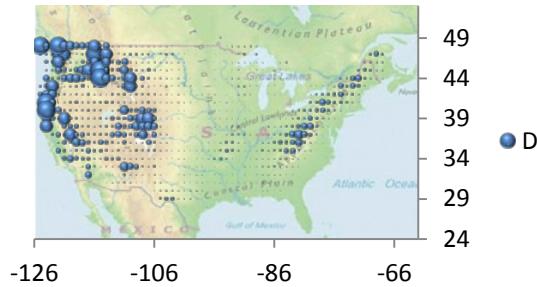


Figure 30: Determining anomalies in terrain elevation modeling. Notice the systemic correlations between elevations separated by rounded 50 foot elevation contours.

The utility of this approach is dependent on the quality and the availability of the data. Mapping out the lower 48 states into 1° sections, we can clearly discern the high diffusivity regions of the country's terrain, bounded by the western and eastern mountains.



We have used this approach to infer models which we can use for formal verification and model checking, as further detailed in Appendix F.

Semi-Markov Terrain Correlation

Terrain correlations also exist on a much shorter scale, which can also show subtle periodic features. The objective is to model the fine terrain, both random roughness as described in Figure 4, and more periodic features such as cobblestones[28] and washboard[29] as shown in Figure 31 below: [30]

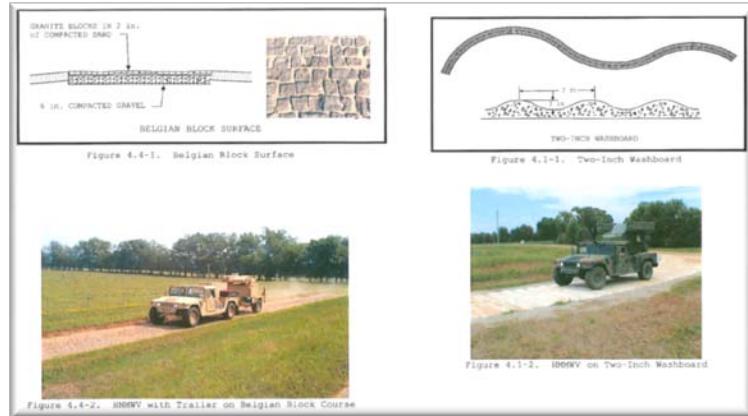


Figure 31: (left) Cobblestone test track and (right) Washboard test track

The Mercedes-Benz test track data used in Figure 4 was also applied to a particular style of road cobblestone paving known as *Belgian Block*. This data was supplied in *OpenCRG* format[15], and covered the track shown in the Google Earth snapshot shown below in Figure 15.



Figure 32: The OpenCRG terrain format features a header which indicates the geospatial location of the data set. This winding Belgian Block course was located underneath an overpass on the Mercedes-Benz campus in Stuttgart, with the start of the path indicated by the green arrow.

A typical one-dimensional profile trace is shown below in Figure 33. Clearly visible are two levels of terrain variability. One variation on the order of 10 cm height contributes to swales on the test track and the other, a fine-grained roughness on the order of 1 cm, are caused by the cobblestone blocks themselves.

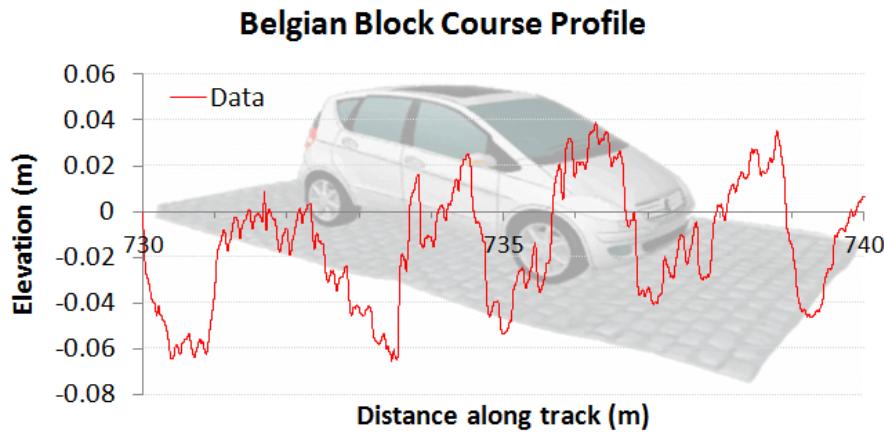


Figure 33: A profile trace along a short path of the test track indicates the roughness.

The fine resolution of the data is shown in Figure 34, which is on the order of 1 cm in the lateral dimension.

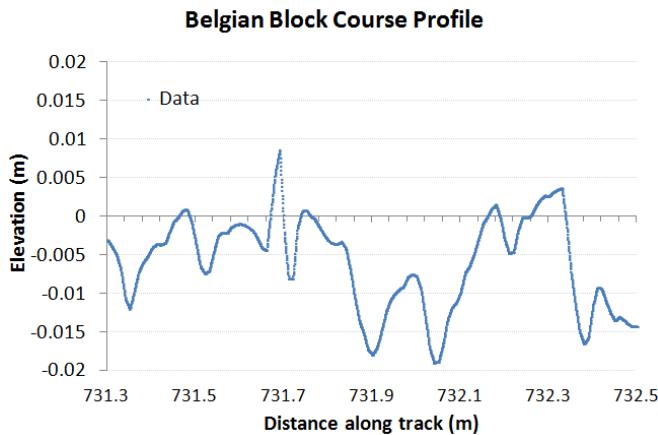


Figure 34 : A very short profile along the test track showing the Belgian Block variability.

The PSD calculated for this course is shown in Figure 35. Although the harmonics are not as striking as that shown for the aquatic wave PSD of Figure 13, their positions correlate well with a model of the terrain consisting of varying degrees of order (shown by the blue curve). The low-frequency odd harmonics correspond to washboard-like swales on the order of 1.5 meters. These are odd-harmonics because they show a tendency to an asymmetric slant or tilt. The higher frequency even harmonics are caused by the Belgian block perturbations which are about 10 cm in lateral spacing.

The overall envelope suggests that a strong random walk component also exists, with a low density filter cutting off the high frequency sharpness in the terrain roughness. Over time, the cobblestones will wear down their sharp edges, and this is observed in the filtered trace.

The idea of using a superposition of various profiles from **Equation 7** makes intuitive sense since the various periodicities have very little mutual coherence, and the pair correlations are additive in contributing to the power spectrum.

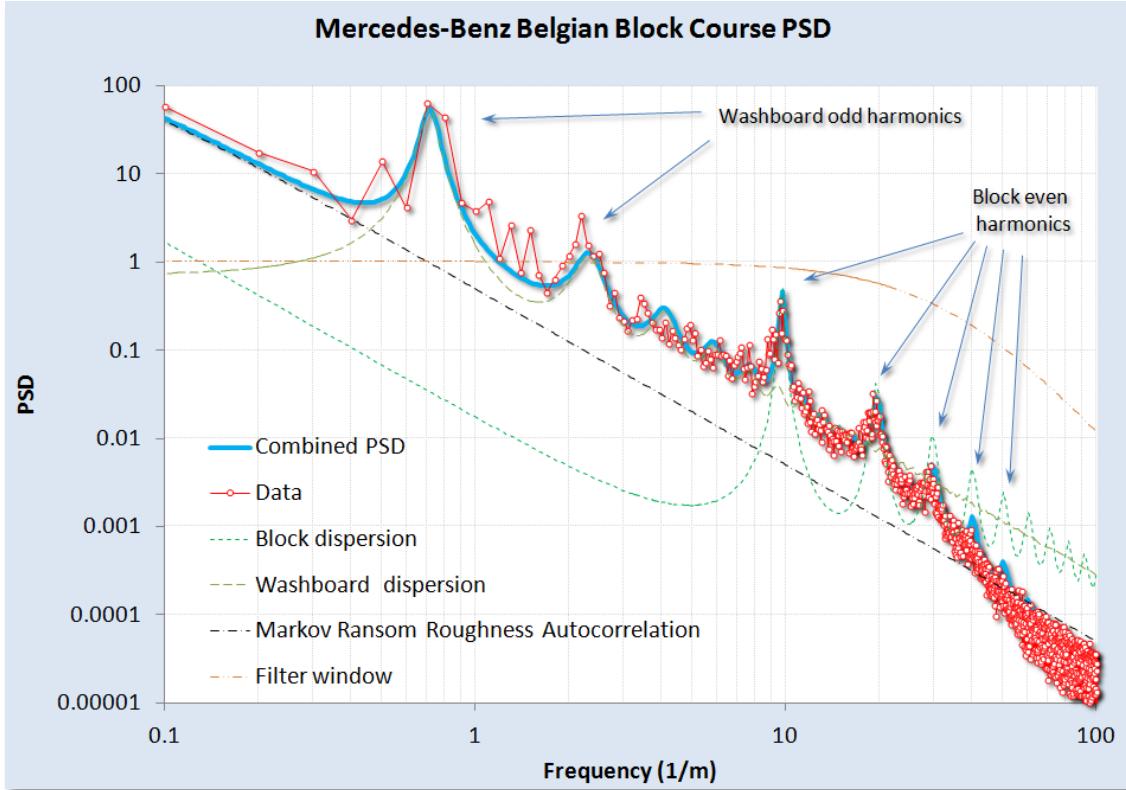


Figure 35: A model fit to the Belgian Block course PSD. A rich variety of surface textures can be gleaned from the harmonics. At high wavenumbers, a change in slope is detected, likely due to a smoothing spatial filter applied to the fine terrain relief and partly due to interpolation for the OpenCRG data set

For the Belgian block course, the data was not detrended prior to computing a spectrum. A slight tilt does exist in the data as can be gleaned from Figure 36, where several sections of 2D Fourier Transform PSD's for simulated (left and right) and real (center) terrains. The general approach of treating both the Z and X dimensions in a full power spectrum allows us to sensitively test for stepped or staircase surfaces.

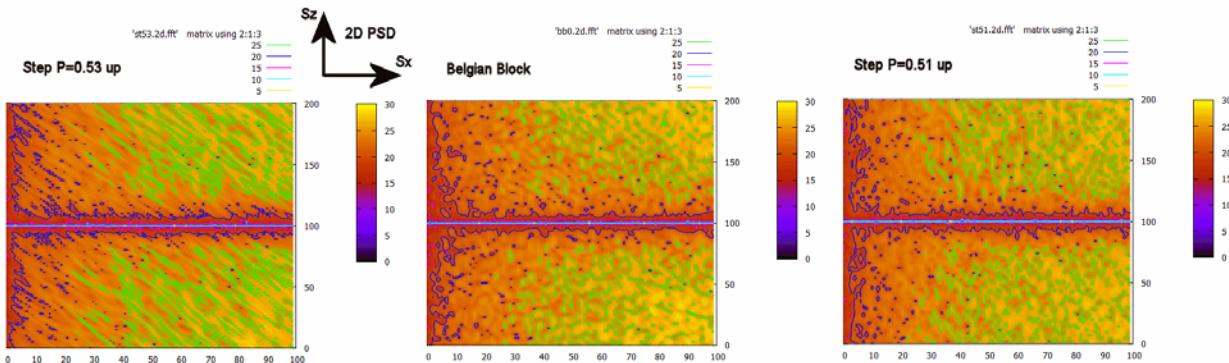


Figure 36: Using the full 2D power spectrum, we can discern subtle changes in slope from any staircase pattern by looking at slanted texture. The center is the real data, and the two side maps show a slightly higher slope (left) and a slightly lower slope (right). The logarithm was taken of the PSD after the transform was multiplied by S_x^2 and S_z^2

As a control study, we also collected PSD data from a military test course and compared the results in Figure 37. This data was detrended prior to use, and was the only unclassified data available for analysis. Similar features are observed as with the Mercedes-Benz Belgian Block course, with the same washboard and block harmonics. These are slightly shifted in period, an effect likely due to different road construction and maintenance procedures.

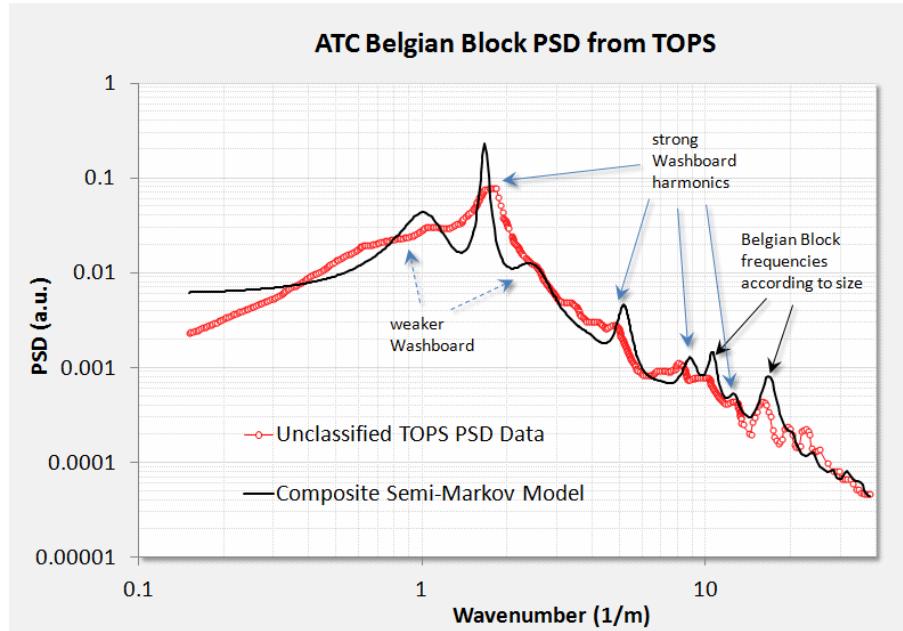


Figure 37: Belgian Block course from a military test operational procedure (TOP) document.

A few other profiles are worth considering. For Figure 38, from [31], we aligned an even-harmonic semi-Markov PSD with the empirically calculated PSD. The strong harmonics of a washboard road are expected as the repeated travel of vehicles over the course reinforces the washboard resonant frequency.

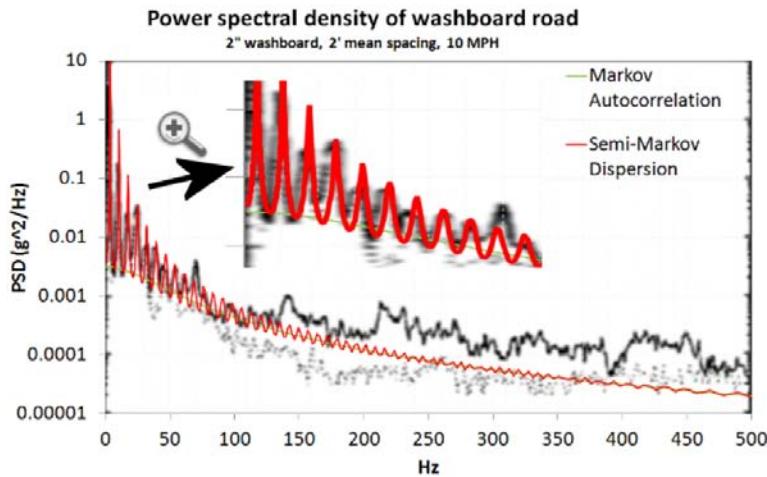
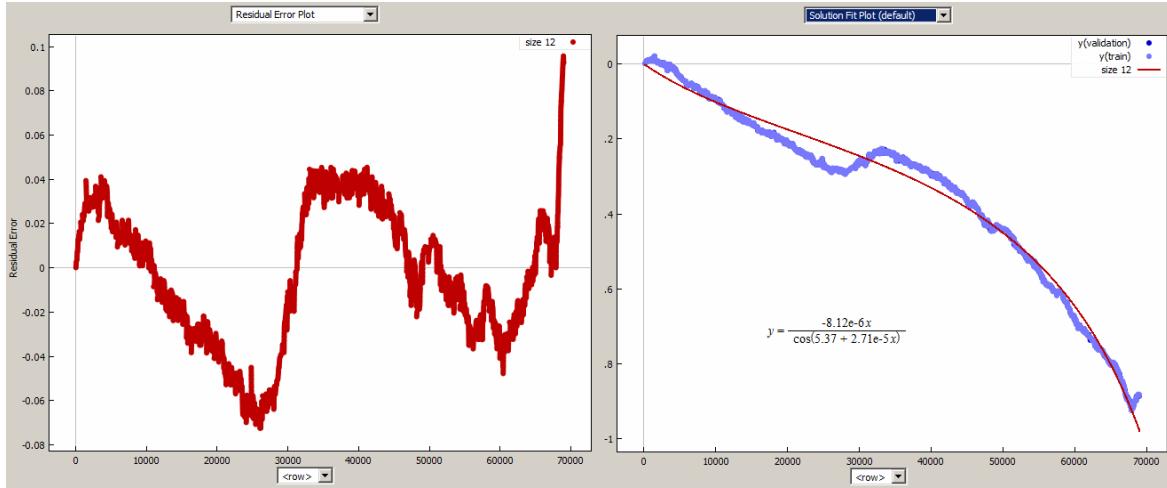


Figure 38: A model fit to a pure washboard. The x-axis is expressed in Hz to indicate the test vehicle was maintaining travel at a constant speed.

We can also return to the rough road power spectrum of Figure 4. We noted that this course likely contained a long-wavelength (deterministic) slope. The data was detrended as shown in Figure 39 below.



*Figure 39: The elevation trend in the rough road data set (right) was removed by a curve fitting program (**Eureqa**). The detrended residuals are shown to the left.*

After detrending the terrain profile, the rough-road PSD was calculated and fit to a semi-Markov model with one low-frequency component and a high-frequency periodic component as shown in Figure 40. The harmonics on the high-frequency spikes suggested that the underlying roughness was shaped similarly to a cobblestone surface. The first and second harmonics are strong, but the third roughly cancels out. By creating a two-level surface where the troughs were half as long as the tops, we could duplicate the missing third harmonic as shown by the solid red line in the figure.

A sinc function filter was added to reproduce the reduced strength of wavenumbers above 10/m.

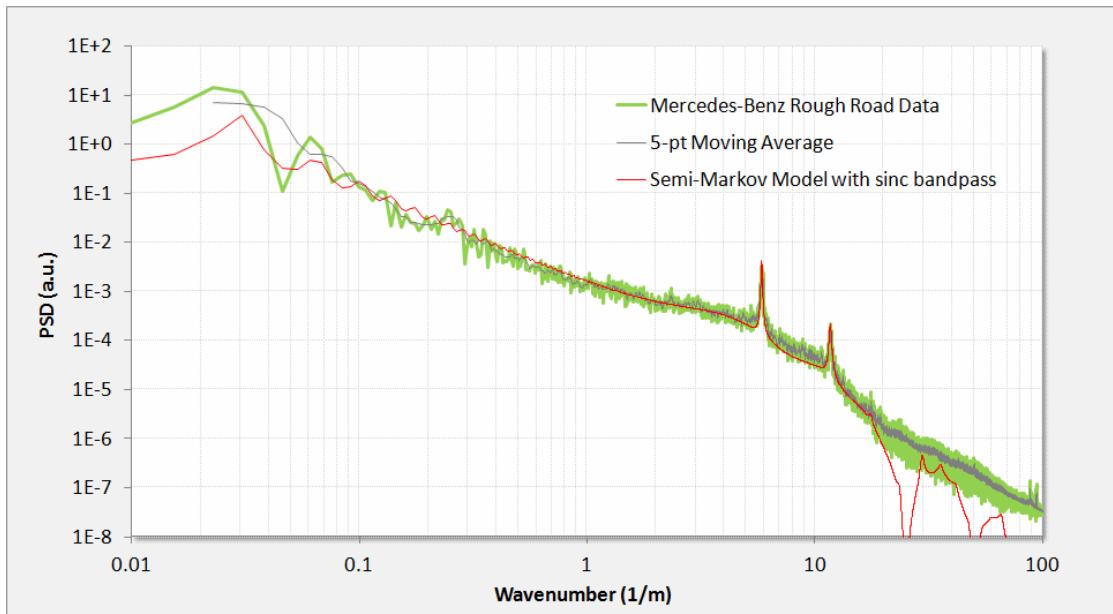


Figure 40: Rough road PSD. From the location and spacing of the harmonics, we can infer a periodic structure very close to a Belgian block structure.

The two-level spectrum is a generalization of the result shown in **Equation 8**.

$$I_0(S_x, S_z) = \mathbb{E} \left\{ \frac{1}{S_x^2 \langle L \rangle} \frac{(1 - P(S_x))(1 - Q(S_x))}{1 - P(S_x)Q(S_x)} \right\} \quad (9)$$

Here each level has its own probability density of lengths, which allows for a surface with wider plateaus and narrower grooves.

$$P(S_x) = \frac{e^{-iS_x L_\alpha}}{1 + iS_x/\alpha}, \quad Q(S_x) = \frac{e^{-iS_x L_\beta}}{1 + iS_x/\beta}$$

A simulated random walk for this terrain relief is shown in Figure 41 alongside an arbitrary path length from the actual terrain. To mimic the sinc filter, a rectangular window moving average was applied to the random walk profile. Note that after calibration of the relative step heights, very good qualitative agreement exists with the actual surface.

The simplified expression from **Equation (9)** is complicated only because of the number of cross terms needed to pair correlate the two levels

$$I = \frac{(\alpha S)^2 + (\beta S)^2 + (\alpha\beta)^2 S^4 - (\beta S)^2 \cos(SL_\alpha) - (\alpha S)^2 \cos(SL_\beta) + \alpha\beta^2 S^3 \sin(SL_\alpha) + \alpha^2\beta S^3 \sin(SL_\beta)}{(1 - \alpha\beta S^2 - \cos(S(L_\alpha + L_\beta)))^2 + (\alpha S + \beta S + \sin(S(L_\alpha + L_\beta)))^2}$$

To generate a synthetic terrain, each level is treated as a sampling distribution drawn from the distributions from each level (see next section).

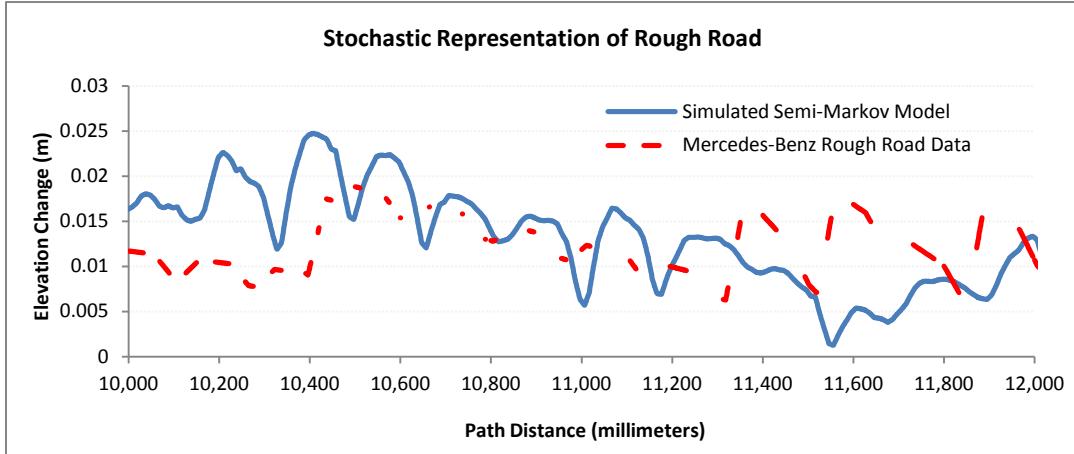


Figure 41: Simulated random walk for the rough road terrain.

As a check to determine whether we can recover the PSD from the simulated semi-Markov random walk, we applied a Fourier transform to 10 sets of simulation traces and squared the amplitude. Note the close alignment with the peaks and with the sinc filter poles. This demonstrates the general utility of switching between a stochastic analytical representation and a Monte Carlo generated simulation.

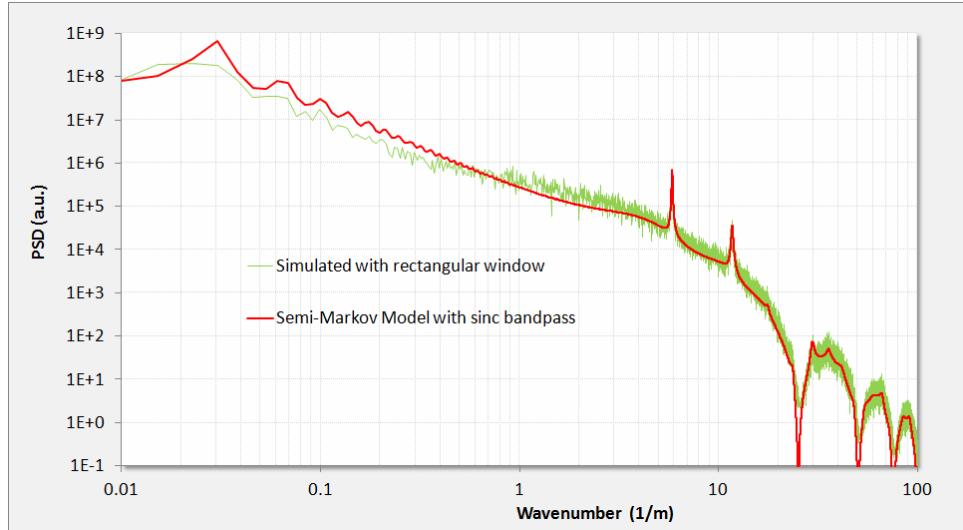


Figure 42: A Monte Carlo simulation of the rough road semi-Markov behavior reveals nearly the same feature amplitudes for the assumed pair-correlation weightings. Contrast this with Figure 40

High Resolution Data

To verify the utility of the semi-Markov approach, we evaluated a fitting procedure against several high-resolution profiles of real vehicle test courses; specifically we chose the Gerotek site in South Africa as this was being measured with advanced profiling tools[13] [32].

To calibrate the data format and determine the limits of resolution, we started with a simulated pothole course. We experimented with the size of the data set so as to clearly reveal features in the PSD, which are shown in Figure 43.

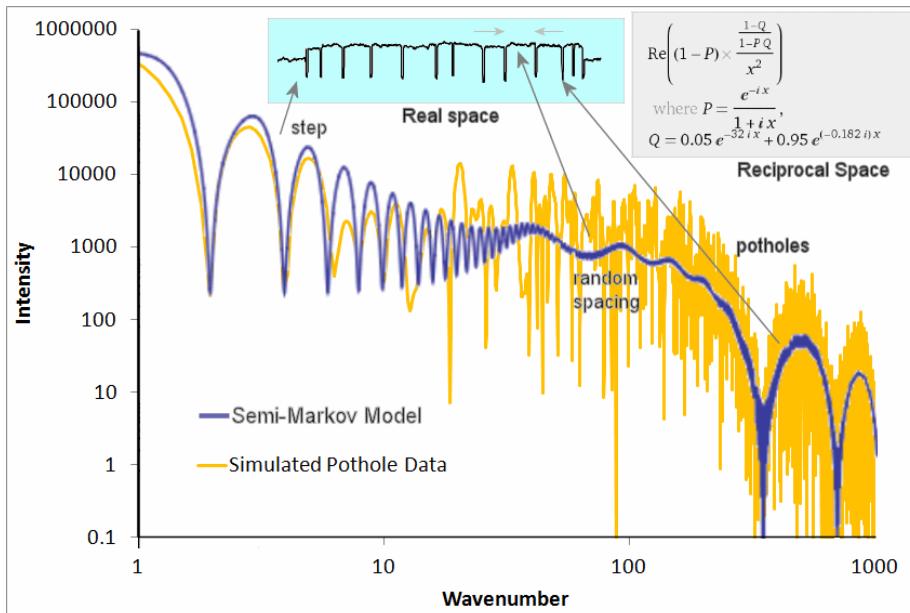


Figure 43: Simulated pothole course PSD. The typical fitting procedure uses knowledge of the underlying terrain profile (top middle inset, showing a pot-hole course with steps) and a

stochastic representation of the PSD (upper right inset) to match the PSD calculated from the data.

For another course at the Gerotek site, we chose the “corrugated” track, consisting of approximately 20 cm high bumps, separated at rather regular intervals. This also provided a very accurate fit to a semi-Markov model (see Figure 44), with the PSD showing a suppressed high harmonic at the spatial frequency predicted by the ratio of the length of the bump to the average spacing.

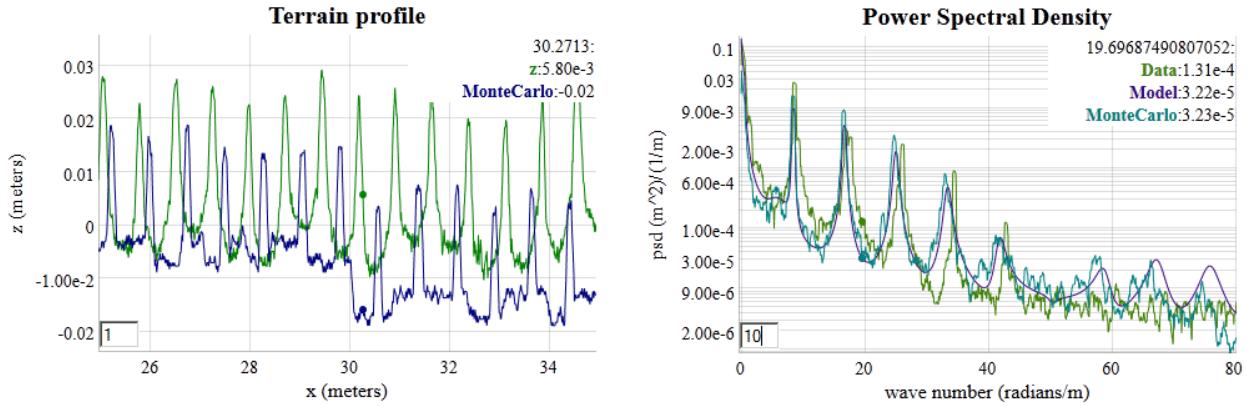


Figure 44: Gerotek Corrugation course. (left) track profile and simulated profile (right) PSD data, simulated PSD, and analytical data

The semi-Markov model with very few parameters is thus able to reproduce a data set containing potentially gigabytes of data.

Generating Synthetic Terrains and Waves and Monte Carlo Sampling

Having a stochastic model of the terrain allows one to generate instances of synthetic terrain that have the same spectral content as the real terrain. This becomes useful for mapping out an *ergodic* representation of possible terrain states suitable for *sampling* (i.e. *Monte Carlo*) simulations.

Several mechanisms exist to generate a random walk, which includes the classical Brownian motion based on a Markov model, the semi-Markov step transition, and the Ornstein-Uhlenbeck random walk process.

Classical Random Walk Model

The simplest *random walk* models will generate a PSD with a $1/S^2$ fall-off, and exemplified by the empirical data in Figure 4. The problem with the simple random walk profile is that it will generate an infinitely high spike for $S=0$, as a random walk is unbounded and will show undulations of infinite length and height. For an example of a typical algorithm for generating a Markov random walk in discrete steps, assume R is a sample drawn as a uniform variate: $R \in [0..1]$.

```
-- classical random walk = rw
rw(Diffusion, Z)
  random(R)
  if R < 0.5 then
    Z = Z + Diffusion
  else
    Z = Z - Diffusion
  end
```

Ornstein-Uhlenbeck Process

The issue with a pure random walk model is that the absolute excursions become unbounded, whereas real data shows bounds in altitude in the terrain or waves. One way around this, which does not impact the spectral fall-off, is to attach a *reversion-to-the-mean* correction term to the random walk algorithm. This procedure is known as the *Ornstein-Uhlenbeck* random walk process[4], and derives from a physical model of an attractor or potential well which “tugs” on the random walker to bring it back to the mean state (see Figure 45 below).

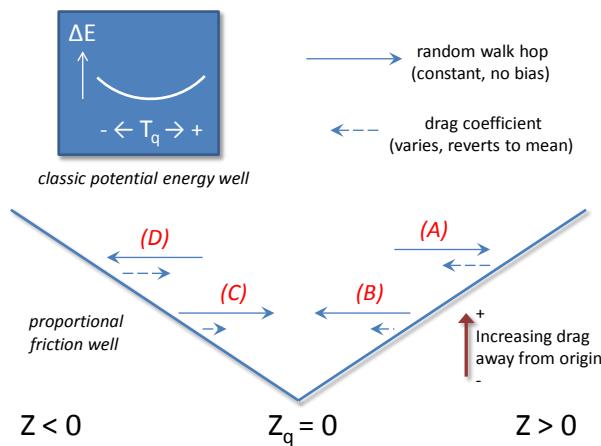


Figure 45: Representation of an Ornstein-Uhlenbeck random walk process for terrain elevation changes. The hopping rate works similarly to a potential well, with a greater resistance to hopping the further excursion ways from a quiescent elevation (Z_q) changes.

The following pseudo-code snippet sets up an Ornstein-Uhlenbeck random walk model with a reversion-to-the-mean term. The *diffusion* term is the classical Markovian random walk transition rate. The *drag* term places an attractor which opposes large excursions in the terrain elevation term, *Z*.

```
-- Ornstein-Uhlenbeck random walk = ou
ou(X1, X2, Drag, Z)
    random(R)
    if R < 0.5 then
        Z = Z*X1 + X2
    else
        Z = Z*X1 - X2
    end

-- This is how it gets parameterized
ou_random_walker (dX, Diffusion, Drag, Z)
    X1 = exp(-2*Drag*dX)
    X2 = sqrt(Diffusion*(1-exp(-2*Drag*dX))/2/Drag)
    ou(X1, X2, Drag, Z)
```

To determine whether an Ornstein-Uhlenbeck process is apparent on a set of data, one can apply a simple multiscale variance (or a multiscale entropy measure [11]) to the result of a *Z* array of length *N*:

```
variance(Z,N) {
    L = N/2
    while(L > 1) {
        Sum = 0.0
        for(i=1; i<N/2; i++) {
            Val = Z[i] - Z[i+L]
            Sum += Val * Val
        }
        print L " " sqrt(Sum/(N/2))
        L = 0.95*L;
    }
}
```

So that for a given random walk simulation, the asymptotic variance will tend to saturate at longer correlation length scales. A typical multiscale variance plot will look like Figure 46.

The autocorrelation of the Ornstein-Uhlenbeck process is, where $\theta=Drag$:

$$R(x) = \frac{D}{\theta} e^{-\theta|x|}$$

Even though this shows a saturation level, the power spectrum still obeys a $1/S^2$ fall-off.[33]

$$I(S_x) = \frac{D}{\pi} \cdot \frac{1}{S_x^2 + \theta^2}$$

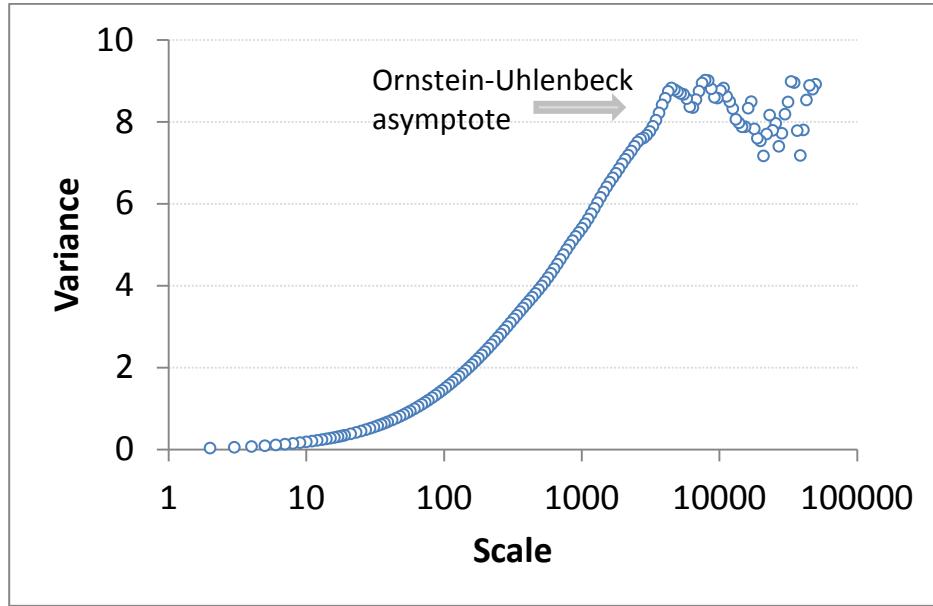


Figure 46: Ornstein-Uhlenbeck process saturates on variance

The Orenstein-Uhlenbeck process is often referred to as *red noise* and the two parameters of Diffusion and Drag can be determined either from the autocorrelation function or from the PSD. For the PSD, on a log-log plot, this involves reading the peak near $S=0$ and then determining the shoulder of the power-law roll-off. Between these two measures, one can infer both parameter values.

Semi-Markov Process

The key to mapping to semi-Markov distributions is to draw from the appropriate PDF. For example, the delayed exponential (see **Equation 6**) is simply a draw from an exponential distribution with a fixed offset (i.e. the delay) added to the variate.

The step deltas and terrace deltas can both draw from a semi-Markov distribution; taking the PSD of the resulting step sequence will generate a noisy **Equation 7**, depending on the amount of samples taken.

The algorithm for generating this semi-Markov model is as follows, where P is a probability of stepping up or down after each length. This has no offset in length. If P is not 0.5 precisely, then the two-dimensional PSD will show an asymmetry reflecting the asymmetry of the terrain profile.

```
BEGIN {
  N = 100000
  h = 0
  srand()
  i = N
  while(i>0) {
    n = - 10.0 * log(rand())
    for(j=0; j<n; j++) {
      i--
      print h
    }
    if(rand()>P) {
      h += +0.0008 * log(rand())
    } else {
      h += -0.0008 * log(rand())
    }
  }
}
```

The “rough road” data model shown previously was generated according to this recipe.

As for the analytical analog semi-Markov model which reproduces the spectral characteristics, consider Figure 47 below and compare it to the Monte Carlo simulation which generates Figure 36. This again is the simplest semi-Markov model, a damped exponential size for both step length and step height.

log plot	$\operatorname{Re} \left(\left(1 - P - H \times \frac{(1-P)^2}{1-HP} \right) z^2 \right)$ <p>where $P = \frac{1}{1+ix}$, $H = \frac{1}{1+iz}$</p>
----------	---

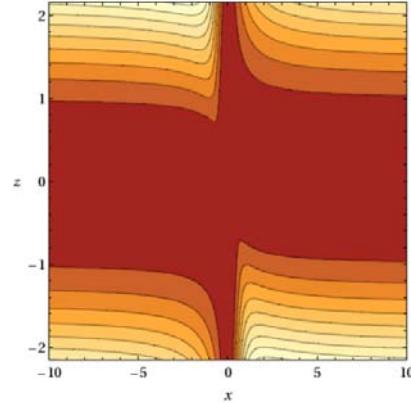


Figure 47 : 2D contour plot of power spectrum for a randomly ascending staircase. To illustrate a 2D power spectrum in terms of the orthogonal wavenumbers S_x and S_z , we multiply by the wavenumber squared and then take the logarithm to give the greatest dynamic range.

The Belgian Block course suggests an asymmetry 0.51 ± 0.01 as shown in Figure 36, indicating sensitivity to incline.

Process of Fitting to Synthetic Terrain

If we have a terrain that indicates some periodicity, these steps will synthesize a semi-Markov profile:

1. Get terrain data $[x, z]$ as distance/displacement pairs
2. Run FFT on data assuming x equally spaced, and take magnitude squared
3. Fit the FFT curve to $w \cdot I(s | \alpha, L)$
 w = scaling
 α = local order
 L = quasi-periodicity
 s = wavenumber

$$I(s | \alpha, L) = \frac{1}{(s - \alpha \cdot \sin(sL))^2 + \alpha^2(1 + \cos(sL))^2}$$

4. Draw a number of Monte Carlo samples from the PDF $\Rightarrow P(x | \alpha, L)$.
 For each sample, use the cookbook inversion of a PDF to generate a step length, x

$$x = L - \frac{1}{\alpha} \cdot \ln(\text{rand}())$$
5. Move the step up and down alternately with amount corresponding to the RMS weighting of w

$$z = c \cdot \sqrt{L}$$

- (if we need to detrend the data against large excursions, apply Ornstein-Uhlenbeck to z)
6. Save the step lengths and step heights as a compact array of $[x, z]$ pairs

Verifying the synthetic PSD

1. Get $[x, z]$ data from last terrain synthesis and discretize so that x is on equal intervals (for the FFT)
2. Run FFT and the magnitude squared
3. Compare the curves. Adjust the step size if magnitude is not to scale

The intensity of a rough or cobbled terrain with differing lengths of troughs and tops is more complicated than that shown here. For the “rough road” model, we need to modify the step lengths to alternate depending on whether they lie on a trough or a top. The approach is straightforward but the pair correlation increases the number of terms in the intensity profile ($I(s)$) quadratically (see Equation 10).

Process Filters

If we wish to add a second-order Markovian feedback, we can filter the data with beyond near-neighbor interactions, i.e. not only $Z[i]$ and $Z[i-1]$ but $Z[i-2]$ and beyond. This adds an additional $1/S^2$ factor as it acts as a low-pass smoothing filter. In the case of an autocorrelation for a second-order step change:

$$c(x) = \int_{y=0}^{\infty} [\alpha^2 y e^{-\alpha y}] \cdot [\alpha^2 (y + x) e^{-\alpha(y+x)}] dy$$

This will generate a power spectrum of order 4:

$$I(S_x) = \frac{1}{\sqrt{2\pi}} \frac{\alpha^4}{(\alpha^2 + S_x^2)^2}$$

Ordinarily, we can apply the class of filters known as autoregressive models - AR(p), where p indicates the interaction order of the model, or in terms of digital processing, finite impulse response (FIR) filters. In general, these will all tend to smooth the simulation step changes, rounding out the edges as a low-pass filter is designed to work. Applying autoregressive models to terrain is an active area of research. [34][35]

Superposition of Waves

A non-stochastic method for generating a synthetic terrain involves extracting the Fourier coefficients from the PSD and superposing sine waves to recreate the original real-space terrain profile. This kind of inversion will generate a repeat sequence with multiples of the longest time period in the spectrum. No phase information is available from the power spectrum, so any long range coherence is artificially introduced, unlike that from a pair correlation. Although potentially useful and practical[36], the superposition approach will definitely not traverse a complete state space and so is not ergodic. It will also not capture skewness and kurtosis in the terrain profile[37], of which the stochastic model can more easily (see Figure 41).

A good application of the spectral superposition approach is for generating aquatic waves for various sea-states in the incoherent regime. Since aquatic waves are already close to sinusoidal, they do not need extra harmonics with phase relationships to match angular shapes (as a terrain might). And the randomness in superposing incoherent waves is visually familiar. This is why most graphics applications use the superposition approach for rendering waves [38].

Calibration of Spectra

The calibration of simulated terrains and their corresponding PSD curves to PSD calculations of actual terrains is aided by the application of Parseval’s theorem [39]:

$$\langle z^2 \rangle = \int I(S) dS$$

This states that the statistical variance in the terrain elevation displacement is equivalent to the integrated power spectrum. In practice, as long as the variance in the real space sampling is stored, then the reciprocal space curve can be shifted by a constant scaling factor to match the variance.

Discussion

Stochastic terrain modeling as described within this study finds practical application for vehicle test and verification applications [40], with the terrain and energy models providing an environmental context for such activities as safety testing. We are essentially trying to map out the *Importance* \times *Likelihood* space for the users of the context models. The users must know the impact or significance of the model effects as well as its likelihood of occurring.

The relevance for a particular topographical model depends on its intended usage. For example, one needs to ask the basic questions:

- What is the importance or impact of the event?
- What is the likelihood of the event?

These occupy largely orthogonal roles as shown in Figure 48 and Table 2 below:



Figure 48: Likelihood versus importance

In the case of terrain, a vehicle operating over a rough profile would experience nuisance effects (vibration and absorbed power) that persistently occur. It could also potentially experience a critical effect (extreme slope) that rarely if ever occurs.

The overall goal is to provide models of environmental stimulus for design verification, using the probabilities inferred from the stochastic models to estimate correctness of the designs.

Note that assessing the impact for a specific design requires composing the context model with a performance model for the design being tested. The context model serves as a stimulus, and the simulation for the design being tested then generates a response. Assessment of a design in terms of probability of failure or probability of satisfying a requirement, involves a likelihood weighted sampling of impact, in general done with Monte Carlo techniques.⁴ Note that as failure conditions will typically be rare, and it is desirable to estimate probably of failure accurately, importance sampling techniques are frequently needed. This is true even though highly parallel computer hardware now allow many thousands of cases to be executed in parallel.

⁴ techniques such as importance sampling[41] may be required

Table 2: Orthogonal spaces

	Description	Examples
High Likelihood	Environmental stimulus that have high likelihood, but only a cumulative impact	Persistent and perpetual nuisance effects <ul style="list-style-type: none"> ▪ Terrain roughness and vibration ▪ Wear and tear ▪ Abrasion ▪ Wind friction and rolling resistance
High Impact	Environmental stimulus that have high impact but often low likelihoods	A critical effect that rarely if ever occurs <ul style="list-style-type: none"> ▪ A large shock ▪ Accidents ▪ 100-year climate events ▪ ... lots of other possible cases ▪ ... or some other pathological or unknown cases

Conclusion

A unified approach for characterizing and modeling topographies has definite advantages for environmental context applications. We have demonstrated how often simple probability density functions (PDF) can characterize the impact/liability factors for various environmental behaviors. A stochastic basis for the models provides an explanation for empirically observed behavior that heuristics often miss. Further, unifying an established set of scientific and ontological criteria helps us to derive and classify the models. This creates an opportunity to establish a collection of context models as a community resource, with supporting technology facilitating its correct utilization for a given purpose. Such a resource can substantially improve the utility of available data for supporting Model Based Engineering, and also create opportunities for improving existing design methods by facilitating the discovery of highly resilient designs.

References

- [1] M. Waechter, F. Riess, H. Kantz, and J. Peinke, "Stochastic analysis of surface roughness -," *EPL (Europhysics Letters)*, vol. 64, no. 5, p. 579, 2003.
- [2] R. H. Stewart, *Introduction to physical oceanography*. A & M University, 2003.
- [3] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Prentice Hall, 1994.
- [4] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis Of Real-World Signals*. A K Peters, Ltd., 2010.
- [5] P. R. Pukite and J. Pukite, "Digital signal processors for computation intensive statistics and simulation," *SIGSIM Simul. Dig.*, vol. 21, no. 2, pp. 20–29, Dec. 1990.
- [6] P. R. Pukite, *The Oil Conundrum: Vol. 1 Decline, Vol. 2 Renewal*, vol. 1,2, 2 vols. Daina, 2011.
- [7] K. Chowdary, "Distinguising and Integrating Aleatoric and Epistemic Variation in UQ," Brown University, 2011.
- [8] E. T. Jaynes and G. L. Bretthorst, *Probability theory: the logic of science*. Cambridge Univ Pr, 2003.
- [9] C. Beck, "Generalized statistical mechanics for superstatistical systems," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 369, no. 1935, pp. 453–465, Dec. 2010.
- [10] P. R. Pukite and C. L. Berman, "Defect cluster analysis for wafer-scale integration," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 3, no. 3, pp. 128–135, 1990.
- [11] P. Pukite and S. Bankes, "Entropic Complexity Measured in Context Switching," in *Applications of Digital Signal Processing*, vol. 17, InTech, 2011.
- [12] P. R. Pukite, C. S. Lent, and P. I. Cohen, "Diffraction from stepped surfaces:: II. Arbitrary terrace distributions," *Surface science*, vol. 161, no. 1, pp. 39–68, 1985.
- [13] C. M. Becker, "Profiling of rough terrain." University of Pretoria, 2008.
- [14] P. H. Cronje and P. S. Els, "Improving off-road vehicle handling using an active anti-roll bar," *Journal of Terramechanics*, vol. 47, no. 3, pp. 179–189, 2010.
- [15] OpenCRG, "OpenCRG Homepage." [Online]. Available: <http://www.opencrg.org/>. [Accessed: 26-Mar-2012].
- [16] F. Behroozi and A. Perkins, "Direct measurement of the dispersion relation of capillary waves by laser interferometry," *American journal of physics*, vol. 74, p. 957, 2006.
- [17] E. Falcon and C. Laroche, "Observation of depth-induced properties in wave turbulence on the surface of a fluid," *EPL (Europhysics Letters)*, vol. 95, p. 34003, 2011.
- [18] Wikipedia, "Cnoidal wave - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Cnoidal_wave. [Accessed: 31-Jan-2013].
- [19] P. R. Pukite, *Reflection High Energy Electron Diffraction Studies of Interface Formation*. University of Minnesota, 1988.
- [20] Patrick Holmes, PhD, "A Course in Coastal Defense Systems I Chapter 5 Coastal Processes: Waves," *CDCM Professional Training Programme, 2001*. [Online]. Available: http://www.oas.org/cdcm_train/courses/course21/chap_05.pdf. [Accessed: 24-Apr-2012].
- [21] FormSys, "Wave Spectra." [Online]. Available: http://www.formsys.com/extras/FDS/webhelp/seakeeper/wave_spectra1.htm. [Accessed: 24-Apr-2012].
- [22] Huang, T.; Alarcon, C.; Bingham, A.; Henderson, M. L.; Kessling, M.; Takagi, A.; Thompson, C. K, "NASA ADS: Data-Driven Oceanographic Web Portal." [Online]. Available: <http://adsabs.harvard.edu/abs/2010AGUFMIN23A1350H>. [Accessed: 12-Apr-2012].
- [23] CDIP, "CDIP Homepage." [Online]. Available: http://cdip.ucsd.edu/?units=metric&tz=UTC&pub=public&map_stati=1,2,3. [Accessed: 24-Apr-2012].
- [24] T. S. Rhee, P. D. Nightingale, D. K. Woolf, G. Caulliez, P. Bowyer, and M. O. Andreae, "Influence of energetic wind and waves on gas transfer in a large wind-wave tunnel facility," *J. Geophys. Res.*, vol. 112, no. C5, p. C05027, May 2007.
- [25] F. Feddersen and F. Veron, "Wind effects on shoaling wave shape," *Journal of physical oceanography*, vol. 35, no. 7, pp. 1223–1228, 2005.
- [26] W. B. Wright, R. Budakian, D. J. Pine, and S. J. Puttermann, "Imaging of Intermittency in Ripple-Wave Turbulence," *Science*, vol. 278, no. 5343, pp. 1609 –1612, Nov. 1997.

- [27] D. Ning, J. Zang, S. Liu, R. Eatock Taylor, B. Teng, and P. Taylor, “Free-surface evolution and wave kinematics for nonlinear uni-directional focused wave groups,” *Ocean Engineering*, vol. 36, no. 15, pp. 1226–1243, 2009.
- [28] BTRC, “BTRC Track Features.” [Online]. Available: http://www.vss.psu.edu/BTRC/btrc_track_features.htm. [Accessed: 24-Apr-2012].
- [29] E. E. Fitch, “Durability Analysis Method for Nontationary Random Vibration,” MIT, 1996.
- [30] Automotive Directorate, “Test Operations Procedure (TOP) 1-1-010 Vehicle Test Course Severity (Surface Roughness),” ATEC, Aberdeen Proving Ground, MD, Final TOP 1-1-010, 2006.
- [31] G. F. Sievert, “Effects of stabilizer bars on road vehicle ride quality,” Rice, 1994.
- [32] C. Becker and P. S. Els, “GEROTEK data from CSIR for C2M2L program.” .
- [33] B. Lindner, “A Brief Introduction to Some Simple Stochastic Processes,” *Stochastic Methods in Neuroscience*, p. 1, 2009.
- [34] L. Li, C. Sandu, and Society of Automotive Engineers, *Modeling and Simulation of 2D ARMA Terrain Models for Vehicle Dynamics Applications*. Society of Automotive Engineers, 2007.
- [35] S. Wagner and J. B. Ferris, “Developing compact autoregressive characterisations of terrain topology profiles,” *International Journal of Vehicle Systems Modelling and Testing*, vol. 7, no. 1, pp. 12–25, 2012.
- [36] J. E. Alexander, “Synthesis of a PSD Compatible Acceleration Time-History,” *Submitted to Shock & Vibration Symposium (November 5-9, 2012)*.
- [37] A. Steinwoff and W. H. Connon III, “Limitations of the Fourier transform for describing test course profiles,” *Sound and Vibration*, vol. 39, no. 2, pp. 12–17, 2005.
- [38] J. Tessendorf, “Simulating ocean water,” *Simulating Nature: Realistic and Interactive Techniques. SIGGRAPH*, 2001.
- [39] L. Romero and W. K. Melville, “Spatial Statistics of the Sea Surface in Fetch-Limited Conditions,” *Journal of Physical Oceanography*, 2011.
- [40] Jonathan Cameron, Abhi Jain, Terry Huntsberger, Garrett Sohl, and Rudranarayan Mukherjee, “Vehicle-Terrain Interaction Modeling and Validation for Planetary Rovers,” NASA JPL, Dartslab, JPL Publication 10-15, 2009.
- [41] M. Denny, “Introduction to importance sampling in rare-event simulations,” *European Journal of Physics*, vol. 22, p. 403, 2001.

Characterizing diffusive growth by uncertainty quantification

Paul Pukite, Steve Bankes, Dan Challou
BAE Systems

Abstract: Environmental models of growth processes contain a great deal of uncertainty. Since the underlying process behavior is rarely well-ordered, any model characteristics will carry along with it a level of aleatory uncertainty governed by the natural disorder. This paper applies novel uncertainty quantification approaches to classes of diffusion problems which illustrate the benefit of assuming natural variability, such as varying rates of oxidation and corrosion.

Introduction

Modeling with uncertainty quantification has application to such phenomena as oxidation, corrosion, thermal response, and particulate growth. These fall into the classes of phenomena governed substantially by diffusional processes. At its most fundamental, diffusion is a model of a random walk. Without a strong convection or advection term to guide the process (e.g. provided by an electric or gravitational field), the kinetic mechanism of a particle generates a random trajectory that is well understood based on statistical physics principles. The standard physics approach is to solve a master diffusion equation under transient conditions. This turns into a kernel solution that we can apply to an arbitrary forcing function, such as provided by an input material flux or thermal impulse.

Yet, the environment that the particle resides in may not be as homogeneous as the ideal diffusion model would imply. Enough uncertainty in the essential diffusion parameters may exist that we need to ask some fundamental questions:

- Why do we assume the diffusion coefficient is a constant?
- How can we know the diffusional interface so precisely?

Within a heterogeneous media, the characteristic diffusion coefficient does not have to remain a fixed value. Varying material composition and amounts of defects can modulate the natural hopping rate and thus smear the diffusion coefficient well beyond the narrow tolerance that is typically assumed. In general, the assumption of a single diffusion coefficient works well for many behaviors — as the observed results are already smeared due to diffusion, while a greater level of uncertainty will not change the mean value of the diffused measures. This mean value approximation works well as a result of applying the central limit theorem of statistics.

Yet, under certain circumstances, the uncertainty in the diffusion coefficient or uncertainty in the experimental geometry will have a subtle yet measurable impact on the diffusional transient behavior. This departure from ideality usually occurs over the short initial growth time, but will also manifest itself via the application of a modified impulse response function.

As an example, if we consider the heat equation, which models the thermal evolution of a transient heat impulse, we immediately note that the standard solution admits an infinite speed of propagation of the initial thermal transient, due to a singularity in the kernel solution expression. That first random walk step is instantaneous according to the math, yet we need to either integrate this transient out or invoke a model such as relativistic diffusion [1][2] to account for the real system behavior. Furthermore, *and just as*

plausibly, the initial transient is likely modified by unaccounted natural uncertainty and disorder in the thermal characteristics and environment, i.e. within the *thermal context*.

This uncertainty also manifests itself in other diffusion models. Deviations from the expected Fickian diffusional growth model lead to the Deal-Grove model[3] for Si oxide growth on a silicon wafer as a practical yet non-ideal heuristic.

In this paper, we apply straight-forward uncertainty quantification. The essential idea is to admit some uncertainty in the diffusion coefficient and in the interface location. By propagating the uncertainty into the diffusion response, certain issues disappear. For example, the infinite speed becomes finite and the kernel solution comes out very clean. This becomes part of the initial condition uncertainty that we know must exist in practice.

In the following work we apply two levels of uncertainty

- Vary the diffusion coefficient to maximum uncertainty with known mean
- Vary the diffusional interface to maximum uncertainty with known mean

As a result we generate a simple diffusion kernel that is much easier to reason with, and can explain several subtle anomalous diffusion behaviors

Example: SiO_2 growth

We first consider the well-known characterization of silicon dioxide, SiO_2 as an example of dispersive growth. Originally characterized by Bruce Deal and Andrew Grove in the 1960's, a careful application of a diffusion-based oxide growth model partially enabled the semiconductor revolution. As illustrated in **Figure 1**, the Deal-Grove model works as a heuristic model in so far as a rigorous first-principles derivation does not exist.

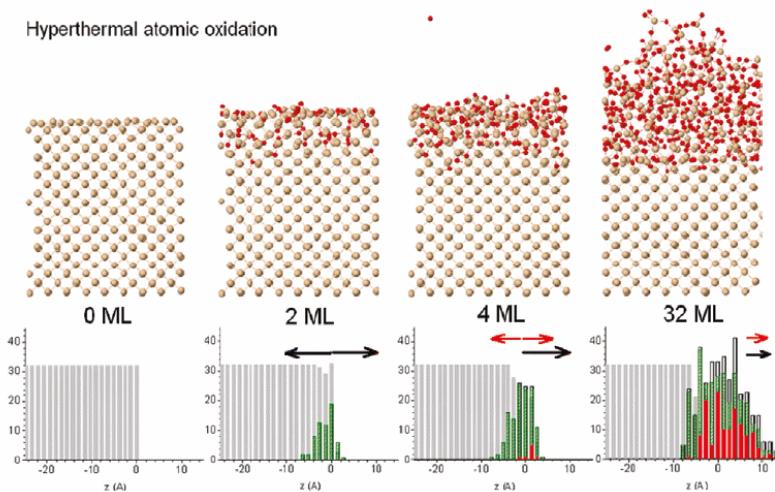


Figure 1: Chemical simulation of oxidation of silicon. A large degree of disorder is evidenced by the mixture of pure Si, SiO_2 , and SiO and as indicated in the histograms by light gray bars, green bars, and red bars, respectively. Black and red arrows indicate the growth direction of the oxidized and silica (SiO) layers, respectively (from [4])

The following derivation improves on the Deal-Grove model by assuming that the diffusion coefficient and location of the growing oxide layer is smeared by a maximum entropy amount; i.e. we can estimate the mean but we leave higher-order moments to vary to maximize the entropy.

Diffusion Solution

The standard approach for solving diffusion problems starts from master diffusion equation

$$\frac{\partial}{\partial t} C(t, x) = D \cdot \frac{\partial^2}{\partial x^2} C(t, x) \quad (1)$$

also known as Fokker-Planck, this is a one-dimensional representation and only lacks the convective or drift term from the general formulation of Fokker-Planck. We also assume that D does not change with respect to spatial coordinates.

We can easily derive the solution of the response function if we think of the diffusion from a planar source outward. The kernel solution gives:

$$n(t, x|D) = \frac{1}{\sqrt{4\pi Dt}} \cdot e^{-x^2/4Dt} \quad (2)$$

We place an impulse of reactants at $x=0$ and want to watch the evolution of the concentration, n , with time. As the concentration drops, we assume that the diffused material from that amount contributes to the growth of the oxide layer.

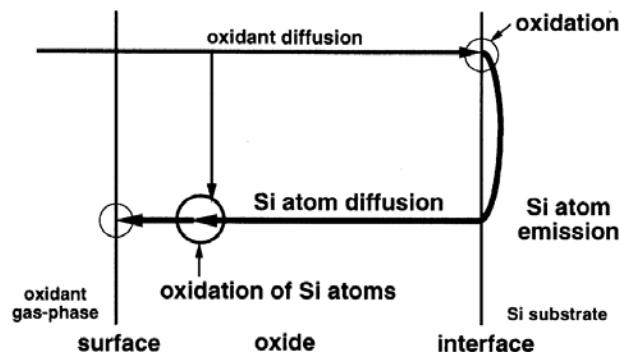


Figure 2: Mechanism for diffusion (from [5])

Consider first that the kernel function represents a one-dimensional concentration profile, in units of number of atomic elements per thickness. The expression $n(t, x)$ describes how quickly the atomic concentration decreases from its initial value. So the accumulated concentration, representing the growth outward from the interface, is the spatial integral of the concentration density. Early on we assume that the region of the interface is spread over a width X .

$$N(t|X, D) = \int_0^X n(t, x|D) dx \quad (3)$$

The average flux of atoms outward from the interface, $J(t|X, D)$, is proportional to the gradient of n , and we apply the diffusion coefficient as the standard proportionality term at position X :

$$J(t|X, D) = D \cdot \frac{\partial}{\partial X} N(t|X, D) = \frac{\sqrt{D}}{\sqrt{4\pi t}} \cdot e^{-X^2/4Dt} \quad (4)$$

Next, suppose we have an idea of a mean value for the diffusion coefficient, D , but don't know how much it varies. Lacking that information, we apply a maximum entropy estimate for the variance assuming a mean value D_0 .

$$p_d(D) = \frac{1}{D_0} \cdot e^{-D/D_0} \quad (5)$$

We can then integrate the concentration across the diffusion probability density function, and the solution reduces to:

$$J(t|X) = \int_0^\infty J(t|X, D)p_d(D)dD = \frac{1}{4\sqrt{t}} \cdot e^{-X/\sqrt{D_0 t}} \left(\sqrt{D_0} + \frac{X}{\sqrt{t}} \right) \quad (6)$$

We also need to consider that for a highly disordered layer, we should place a maximum uncertainty around the value of X .

$$p_x(X) = \frac{1}{x_0} \cdot e^{-X/x_0} \quad (7)$$

Once again we can apply a probability density function, this time to the flux, which marginalizes X according to the following integration:

$$J(t) = \int_0^\infty J(t|X)p_x(X)dX = \frac{D_0}{4} \cdot \left(\frac{1}{x_0 + \sqrt{D_0 t}} + \frac{x_0}{(x_0 + \sqrt{D_0 t})^2} \right) \quad (8)$$

As a last step, we need to integrate the average flux over time to arrive at the growing width, W , of the oxide layer:

$$W(t) = \int_0^t J(\tau) d\tau = \frac{1}{2} \sqrt{D_0 t} \cdot \frac{\sqrt{D_0 t}}{x_0 + \sqrt{D_0 t}} \quad (9)$$

The time integral of this flux is the accumulated concentration of material with a constant inflow of material (i.e. molecular or elemental oxygen) from the surroundings. By integrating the diffusional response, we can demonstrate how the step input transiently supplies reactants to the growing interface. The second factor is the newly realized suppressive effect due to disorder. For small t , this compensates the lead term to provide a linear growth factor, which is the original heuristic in the Deal-Grove growth law.

In summary, we applied a two step maximum entropy estimation process to model the disorder in the growing oxide layer. Without any knowledge about the distribution of D and X , apart from asserting that they must exist, we applied the following series of transforms:

$$n(t, x|D) \xrightarrow{\text{integrate}} N(t|X, D) \xrightarrow{\text{flow}} J(t|X, D) \xrightarrow{\text{average}} J(t) \xrightarrow{\text{accumulate}} W(t) \quad (10)$$

This provides a diffusional response due to a continuously applied step concentration to model a growing thickness. For oxide growth, a step input of oxygen is supplied from one side of the interface, and the substrate supplies silicon atoms, see Figure 2. The figure below provides a model fit to recent data from a set of SiO_2 growth experiments.

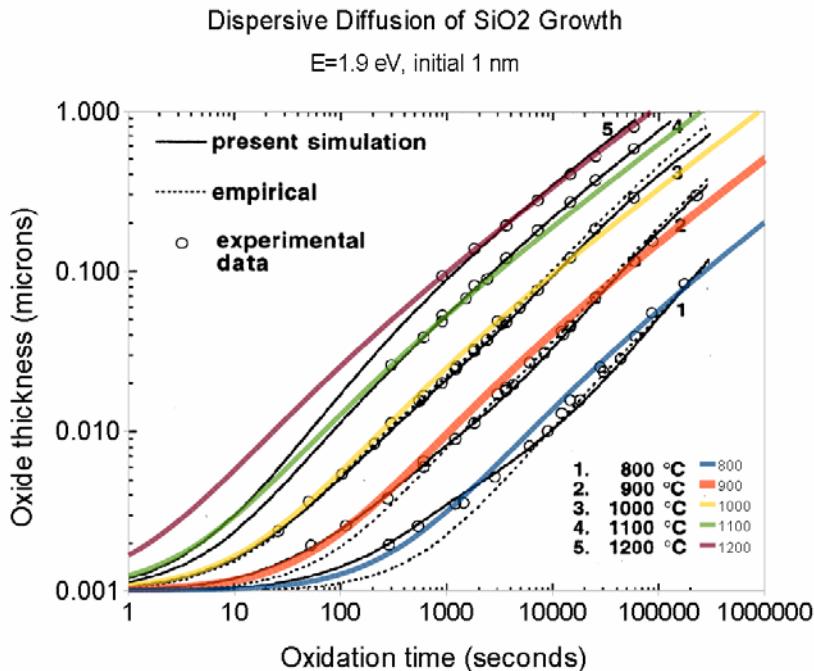


Figure 3: Comparison of dispersive diffusion model against the convention Deal- Grove model (empirical dotted) and a detailed simulation based on reaction kinetics. For silicon, an immediate oxide layer is formed which generates a baseline thickness of about 1 nanometer (data and simulation from [5])

Over time, the response will attain a square root growth law, indicative of the Fick's law regime of what is often referred to as parabolic growth¹. The larger the mean diffusion coefficient or the smaller the uncertainty level x_0 , the more quickly that the response will diverge from the short-term linear growth regime.

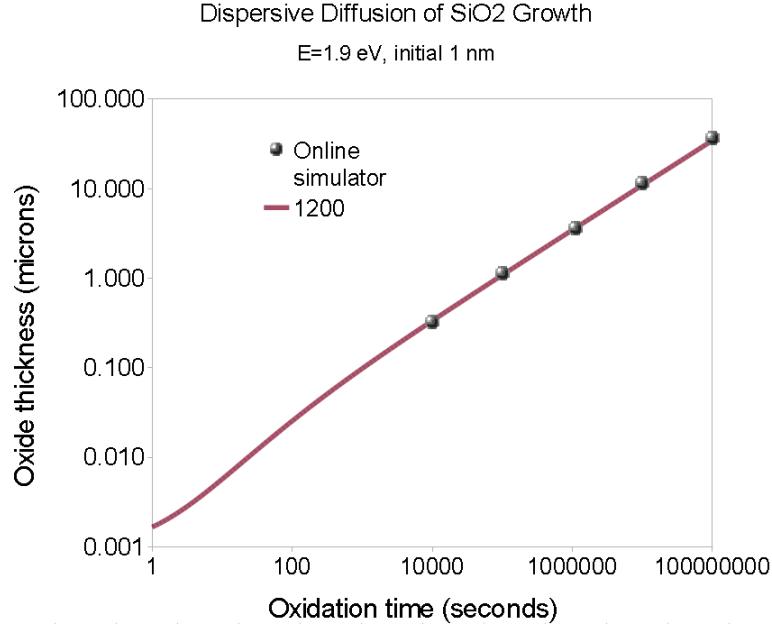


Figure 4: In the Fickian growth regime, the dispersive diffusion formulation follows a square-root time dependence, and can be confirmed with an online SiO_2 growth calculator²

General Applicability

The uncertainty quantification proposed here has more general applicability than just to oxide growth on silicon. Any diffusional process that contains a degree of uncertainty in its parameters is a candidate for this simplification.

The general dispersed response to an impulse at the smeared origin is generated by the marginalization of Eq. 2 with Eq. 5 and Eq. 7 with the result:

$$n(t) = \frac{2}{x_0 + \sqrt{Dt}} \quad (11)$$

This describes the damping of the peak concentration profile with time, cast in terms of a one-dimensional density. Alternatively, by scaling with a time constant, the impulse response can be normalized:

¹ Parabolic growth is somewhat of a misnomer because the actual growth is a square root with time. In other words it is parabolic with respect to the growth distance.

² Go to <http://www.cleanroom.byu.edu/OxideThickCalc.phtml>

$$n(t) = \frac{1}{1 + \sqrt{\frac{t}{\tau}}} \quad (12)$$

Here, τ is a diffusional time constant, indicating how slowly the concentration disperses from the origin. This is also a measure of the persistence of a material to remain localized (see [6]). The scaling implications are important to allow us to generalize the behavior over a range of conditions [7].

In fact, the dispersive formulation can extend to mathematically related behaviors such as corrosion and particulate growth. Corrosion acts very similar to oxidation in that the rate of the corrosive materials has to depend on diffusion of the reacting species with the replenishment of fresh substrate. Further, we can evaluate how well this applies to thermal diffusion, which uses the heat equation in a mathematical formulation very similar to particle diffusion.

Example: Thermal Dispersive Diffusion

The master model for thermal diffusion is referred to as the *heat equation*. This has the same formulation as Eq(1) with the role of material concentration played by temperature (which represents the local thermal excitation). In many practical applications, we may want to know the dissipation of heat from a source. In the case of a planar source of heat such as the surface of a CPU, we can use a one-dimensional model to illustrate the principle. Or we can demonstrate the flow of heat through a rod for different thermal conductivities (see Figure 5).

The ideal non-disordered impulse response should follow

$$\Delta T(t) = \frac{C}{\sqrt{Dt}} e^{\frac{-x^2}{Dt}} \quad (13)$$

The disordered variation of this is

$$\Delta T(t) = \frac{C}{\sqrt{Dt}} e^{\frac{-x}{\sqrt{Dt}}} \left(1 + \frac{x}{\sqrt{Dt}} \right) \quad (14)$$

Since each of the rods consists of a uniform homogeneous metal, one would expect that the response should be near to what theory predicts, and that is the case. [8]

For each of the materials of Figure 5, the idealized impulse response works well (Eq. 13), while the disordered variant (Eq. 14) reveals a larger dispersive spread.

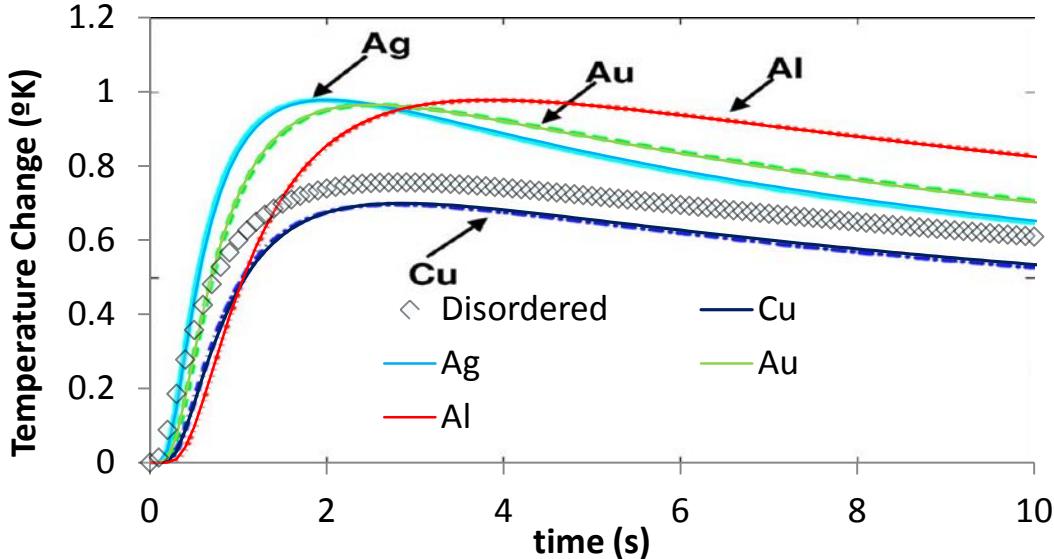


Figure 5: Experimentally measure thermal diffusion profiles for various metals showing excellent agreement with ordered diffusion. The dispersed diffusion profile is shown alongside.

Where we would like to apply the disordered response is to an interface that has a thermal stimulus on one side and an outlet on the other, which turns out to be a realization of a compartmental or box model for thermal dynamics.

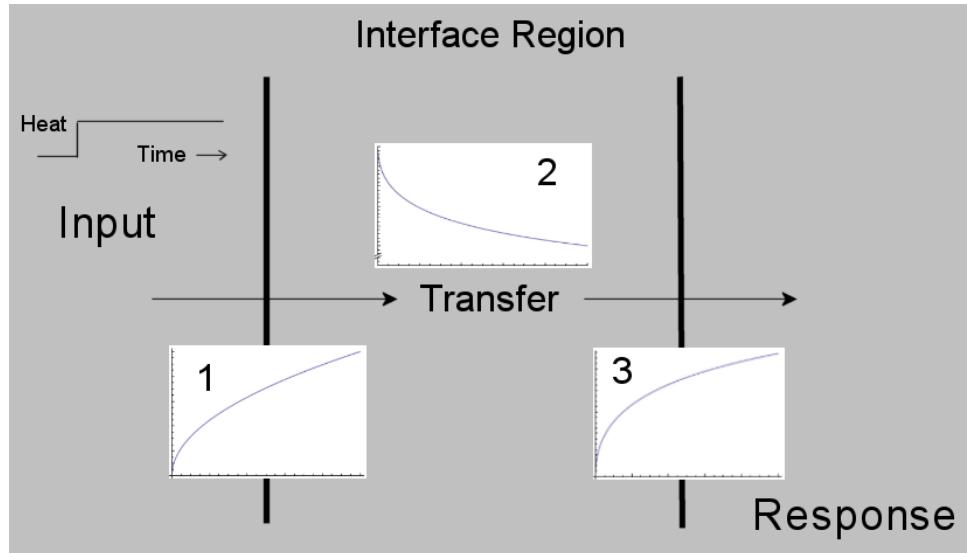


Figure 6: A box model for thermal dynamics assumes a stimulus representing an initial transient (1) convolved against a dissipative transfer function (2) to produce a modulated response (3).

The role of the box model is to model a thermal input along with a response which will allow an alternate path for dissipation of heat. In Figure 6, the input stimulus is a unit step which immediately creates a Fickian square root buildup near the interface (subgraph 1). This gets dissipated by a smeared impulse response function (subgraph 2) and the result is shown as a modulated response (subgraph 3) showing a slower buildup than the square root rise. The modulated reduction results from heat that is dissipated from non-specific paths as described in [8], described by Figure 7.

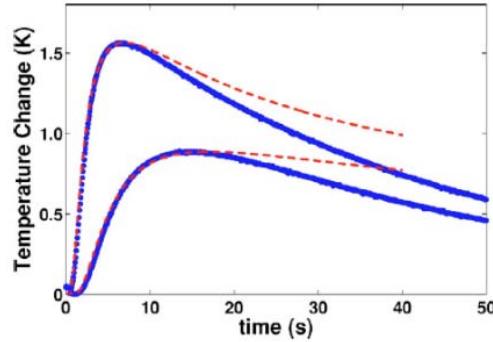


Figure 7: Red lines indicate the theoretical profile under a non-lossy thermal environment.

To analyze the thermal response we use the concept of a convolution to drive the response from a combination of input delta temperature change and the transfer function representing heat dissipation following secondary paths.

$$\text{Response}(t) = \text{Input}(t) \otimes \text{Transfer}(t) \quad (15)$$

At the origin of the thermal impulse, the smeared diffusive response is:

$$\Delta T(t) = \frac{1}{1 + \sqrt{t/\tau}} \quad (16)$$

This agrees with measurements from experiments, as shown in Figure 8.

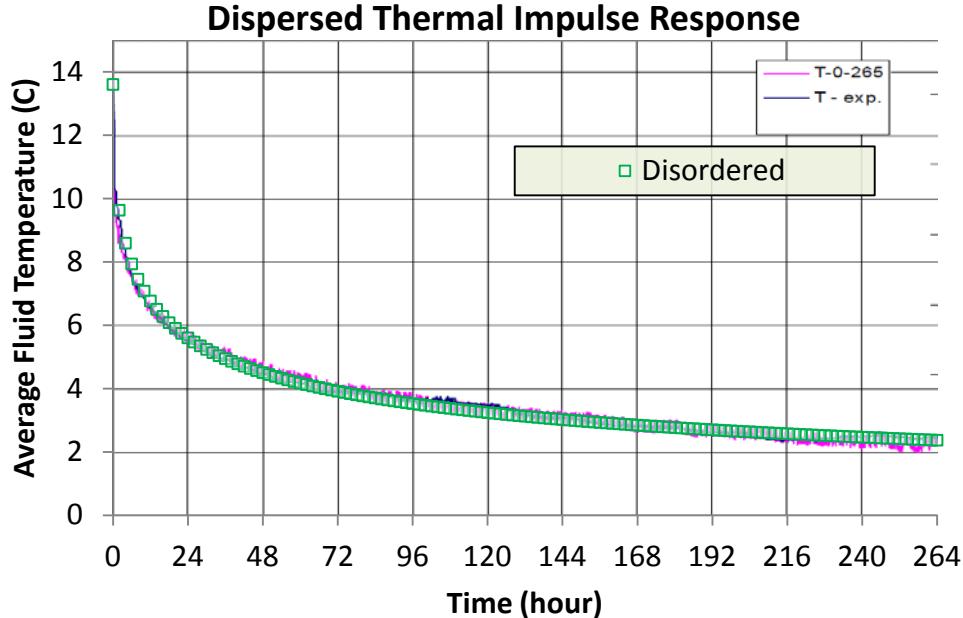


Figure 8: Impulse response from a unit step thermal stimulus applied to an earthen mass (data from [9][10]).

If we take a strong dispersive diffusive decline and convolve with a Fickian growth curve to model the loss:

$$Response(t) = \sqrt{t} - \sqrt{t} \otimes \frac{d}{dt} \left(\frac{1}{1 + \sqrt{t/\tau}} \right) \quad (17)$$

The result is a modulated response:

$$\begin{aligned} t > \tau \\ Response(t) &= \pi - \frac{2 \cosh^{-1} \left(\sqrt{\frac{t}{\tau}} \right)}{\sqrt{\frac{t}{\tau} - 1}} \quad (18) \\ t < \tau \\ Response(t) &= \pi - \frac{2 \cos^{-1} \left(\sqrt{\frac{t}{\tau}} \right)}{\sqrt{1 - \frac{t}{\tau}}} \end{aligned}$$

A contour of the response surface is shown in Figure 9.

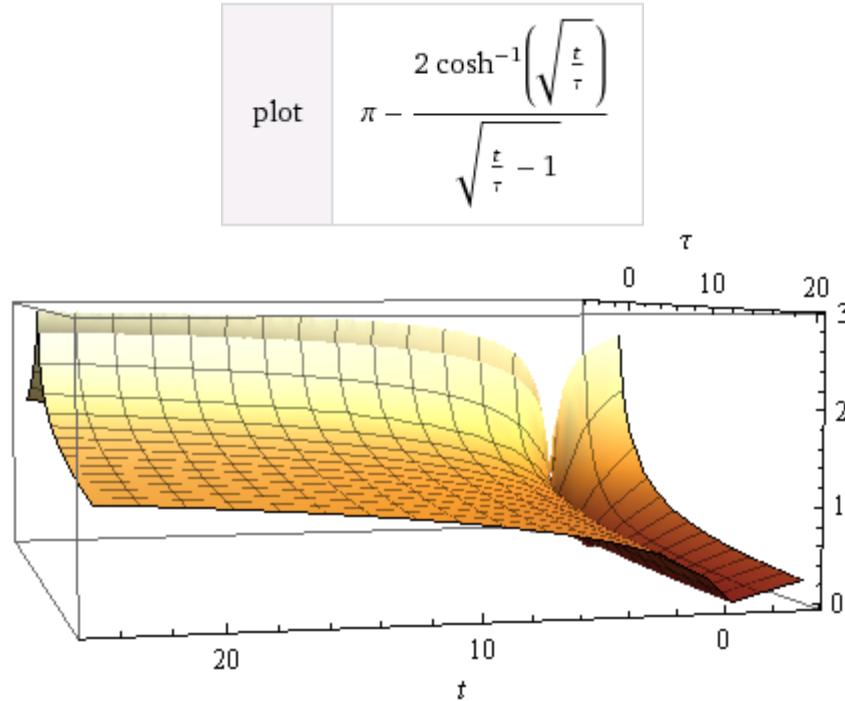


Figure 9: Response surface of the dissipative dispersion function

This formulation accounts for the dispersed heat losses arising from diffusion through dispersed paths.

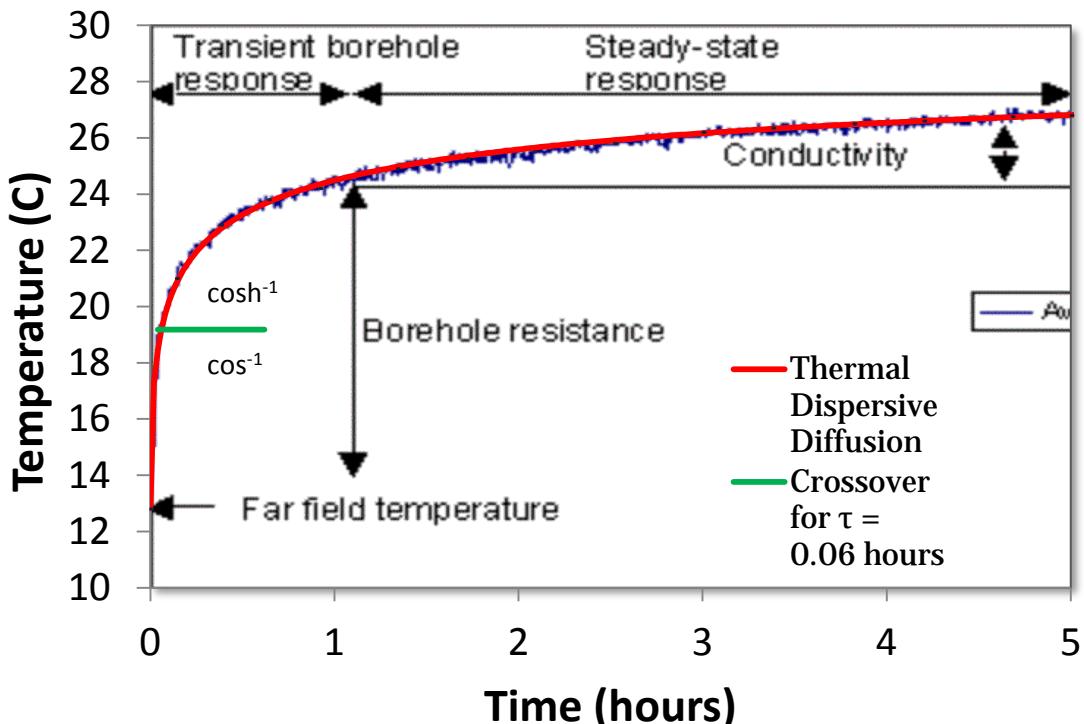


Figure 10: Dispersed impulse response measured away from the stimulus. Note the crossover point separating the $\cosh^{-1}()$ and $\cos^{-1}()$ behavior (from [11]).

By applying a sequence of thermal impulse steps we can demonstrate how to model piecewise transients (see **Figure 11**).

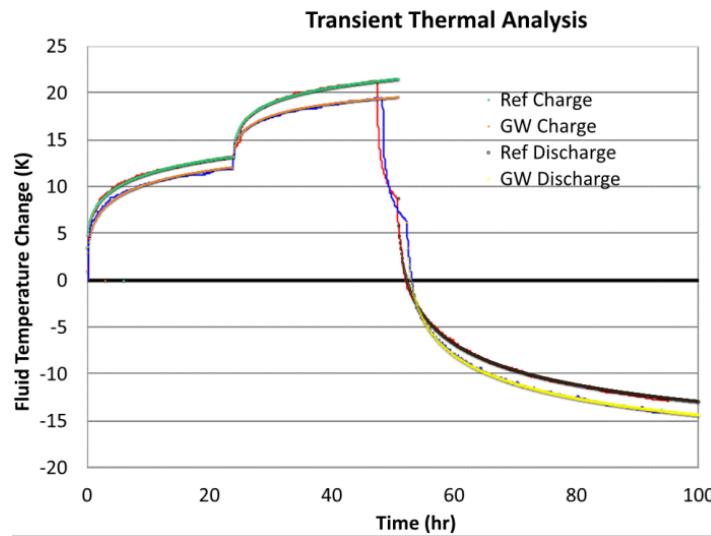


Figure 11: Series of hot and cold unit step impulses applied to an earthen mass measured as a transient response (data from [10] to a groundwater (GW) or reference sink).

In addition to the utility for large scale geothermal transients as just described, we can also apply this dispersive thermal approach to smaller scale contexts, such as a component or subsystem heat sink. Figure 12 shows typical thermal transients observed under various dissipation paths for a powered-up CPU card. Depending on whether the environment contains a well defined heat sink and cased enclosure, the transient will display markedly different apparent time constants. This by itself is not too surprising, but the dispersive diffusion response allows us a simple model to match up with the empirical results. In particular, where we have uncertainty in the environment, this approach excels, as it only requires a mean value estimate for the composite thermal diffusion coefficient or conductivity.

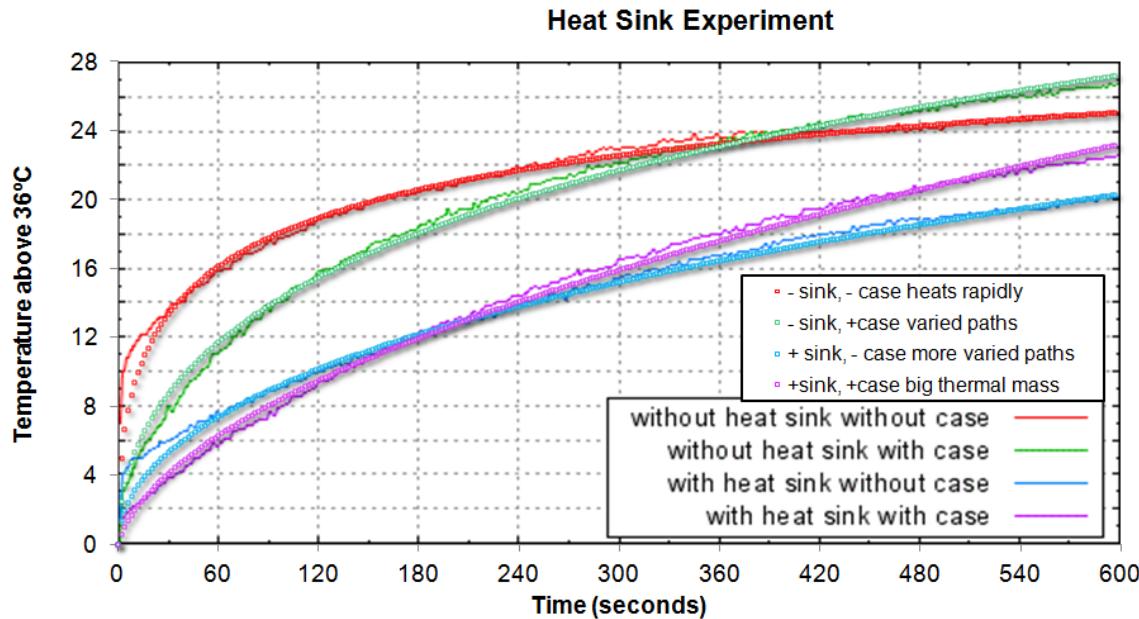


Figure 12: Thermal transients observed within a CPU housing.

Example: Corrosive Growth

The salient principle behind the corrosion of metals is that rust growth mirrors that of the oxide growth of silicon. The number of random environmental factors involved in corrosion suggests that it would make an ideal candidate to apply the dispersive diffusion model. In the oxidation of steel, several different varieties of oxide are involved with varied atmospheric levels of corrosive reagents.

Figure 13 (linear scale) and **Figure 14** (log scale) show an oxide growth model and the results of experimental measurements for a steel composed structure left to weather under different climatic conditions. The behavior specified in Eq(7) was applied directly to the data with assumed mean values for an effective diffusion coefficient.

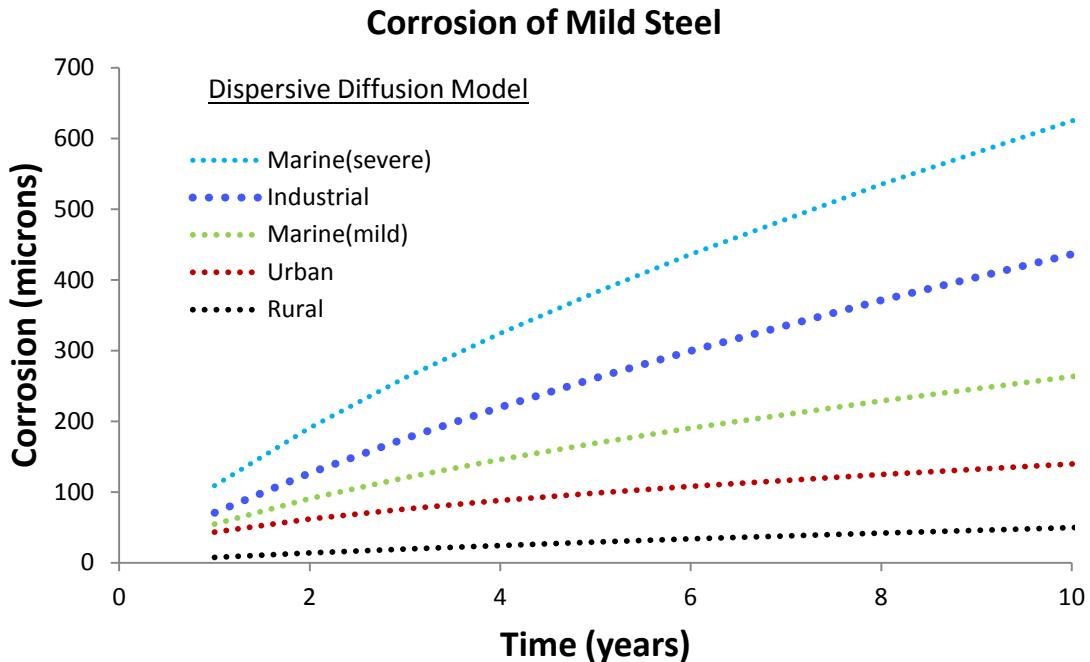


Figure 13: Corrosion growth rates on a linear scale

Uncontrolled rusting also has the propensity for peeling off thick layers, thus exposing fresh layers which will start the oxidation process over again. That tendency allows for the possibility for a growth power law of $\frac{1}{2}$ combining with a linear growth law leading to the rough 0.7 power law observed. This is the same rate observed for the dispersive diffusion model with an uncertainty in the transition zone during the early oxidation process, see Eq(9) rewritten below with x representing the corrosion depth.

$$x(t) = \sqrt{Dt} \frac{\sqrt{\frac{Dt}{x_0}}}{1 + \sqrt{\frac{Dt}{x_0}}} \quad (19)$$

The dynamic range is limited but the rate of growth suggests an initial linear regime, which then bends into the mixed power law growth of Eq(19). The model works well for the highly corrosive regimes of *marine* and *industrial* environments but diverges for the milder environments at longer times.

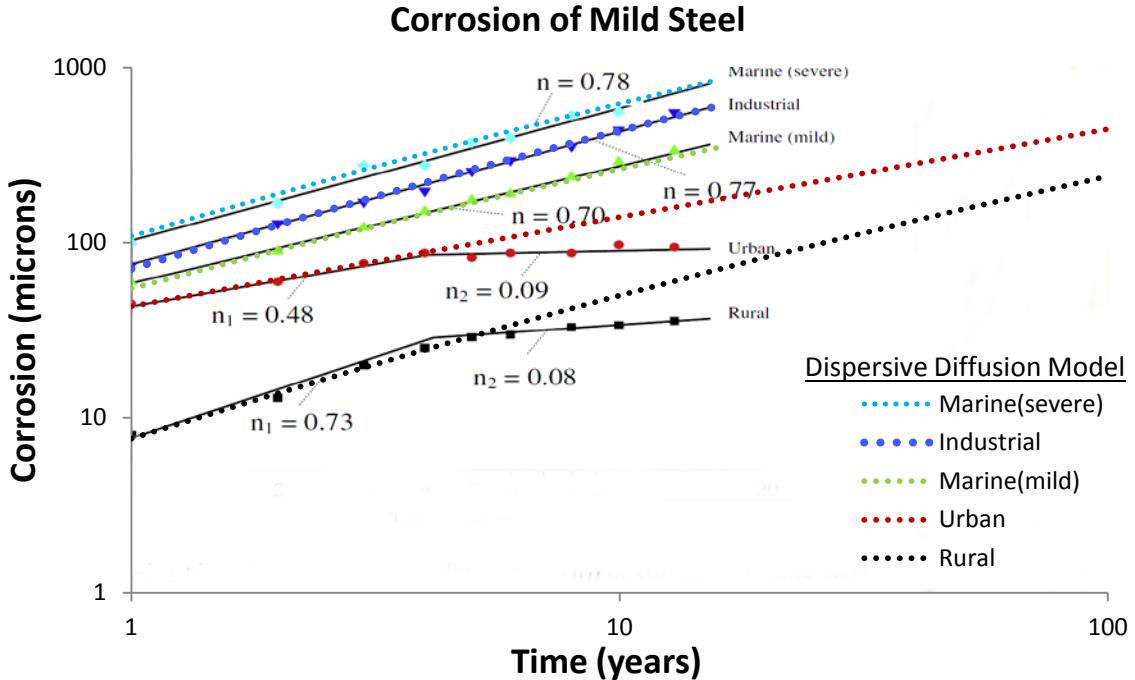


Figure 14: Corrosion growth rates on log scale compared against data from [12].

Ornstein-Uhlenbeck correction: Due to its origins as a random walk process, a pure diffusion model of particles will show unbounded excursions given a long enough time duration. This is characterized by the unbounded Fickian growth law showing a \sqrt{t} dependence for a pure random walk with a single diffusivity.

In practice, the physical environment of a particle may prevent unbounded excursions. It is physically possible that the environment may impose limiting effects on the extent of motion, or that it will place some form of drag on the particle's hopping rate the further it moves away from a mean starting value.

We can use the Ornstein-Uhlenbeck process to model mathematically how this pure random walk becomes bounded. The Ornstein-Uhlenbeck process has its origins in the modeling of Brownian motion with a special “reversion to the mean” property in motion excursions. The following expression shows the stationary marginal probability given a stochastic differential equation $dX = -a X \cdot dt + dW$ which models a drag on an excursion [13]

$$dP(X(t+s) = x | X(s) = 0) = \frac{1}{\sqrt{2\pi\tau}} e^{-\frac{x^2}{2\tau}} dx$$

$$\text{where } \tau = \frac{1 - e^{-2at}}{2a}$$

The rationale for this limiting process to occur in a corrosive environment may arise from a barrier to diffusion beyond a certain critical thickness. As we demonstrated in volume 2 (see Appendix B), the Ornstein-Uhlenbeck process is very common at both gross and granular scales when it comes to describing terrain excursions, and the same process likely occurs at micro scales – perhaps occurring in a

similar fashion to when it was first formulated to describe Brownian motion in the presence of drag on particle velocities.

The O-U correction is straight-forward to apply on our dispersive corrosive growth formulation, we only need apply a non-linear transformation to the time-scale.

$$t \xrightarrow{O-U} \tau$$

and then apply this to the corrosion growth Eq(19):

$$x(\tau(t)) = \sqrt{D\tau(t)} \frac{\sqrt{\frac{D\tau(t)}{x_0}}}{1 + \sqrt{\frac{D\tau(t)}{x_0}}} \quad \text{where} \quad \tau(t) = (1 - e^{-2at})/2a$$

This has the equivalent effect of appearing to slow down time at an exponential rate. This exponential rate turns out to be much faster than the Fickian growth law can sustain, so that an asymptotic limit is achieved in the diffusional or corrosive growth extent.

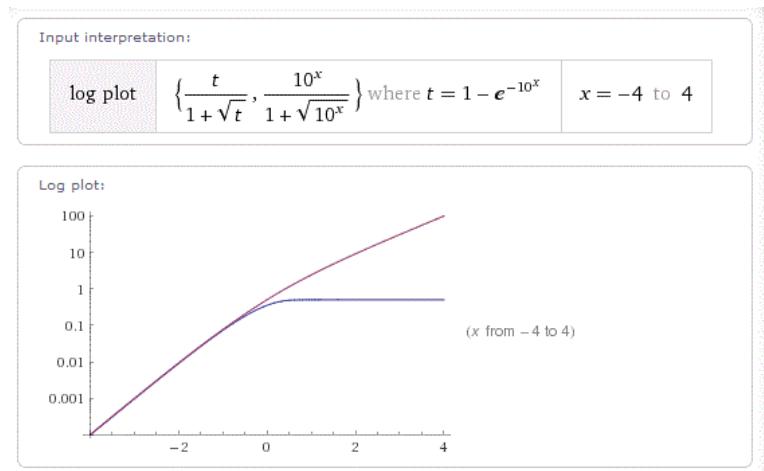


Figure 15: The reversion to the mean process of the Ornstein-Uhlenbeck process will limit the growth of a diffusional process

The caveat on this is that we are only applying this correction based on empirical observations. As an example, if the corrosion appears to flatten out as observed in the *urban* and *rural* rates of corrosion in **Figure 14**, we can model this behavior by assuming an Ornstein-Uhlenbeck reversion-to-the-mean process. In **Figure 16** below we apply the O-U limiting factor to model these two least corrosive environments.

As an explanation for a limiting effect on corrosive growth, it may be that a protective oxide — think in terms of something akin to the self-limiting growth of aluminum oxide Al_2O_3 — or perhaps some anodizing agent which forms after some time duration to limit further growth.

Whatever the rationale, the result of our characterization suggests that a rather simple formulation can be used to model the corrosive growth laws, with enough flexibility to handle the observed growth profiles.

Table 1: Parameterization for the corrosion model with Ornstein-Uhlenbeck reversion-to-the-mean drag, a .

	Marine(severe)	Industrial	Marine(mild)	Urban	Rural
D	90,000	45,000	12,000	6,800	5,300
X ₀	600	400	100	50	500
a	1.00E-08	1.00E-08	1.00E-08	0.4	0.2

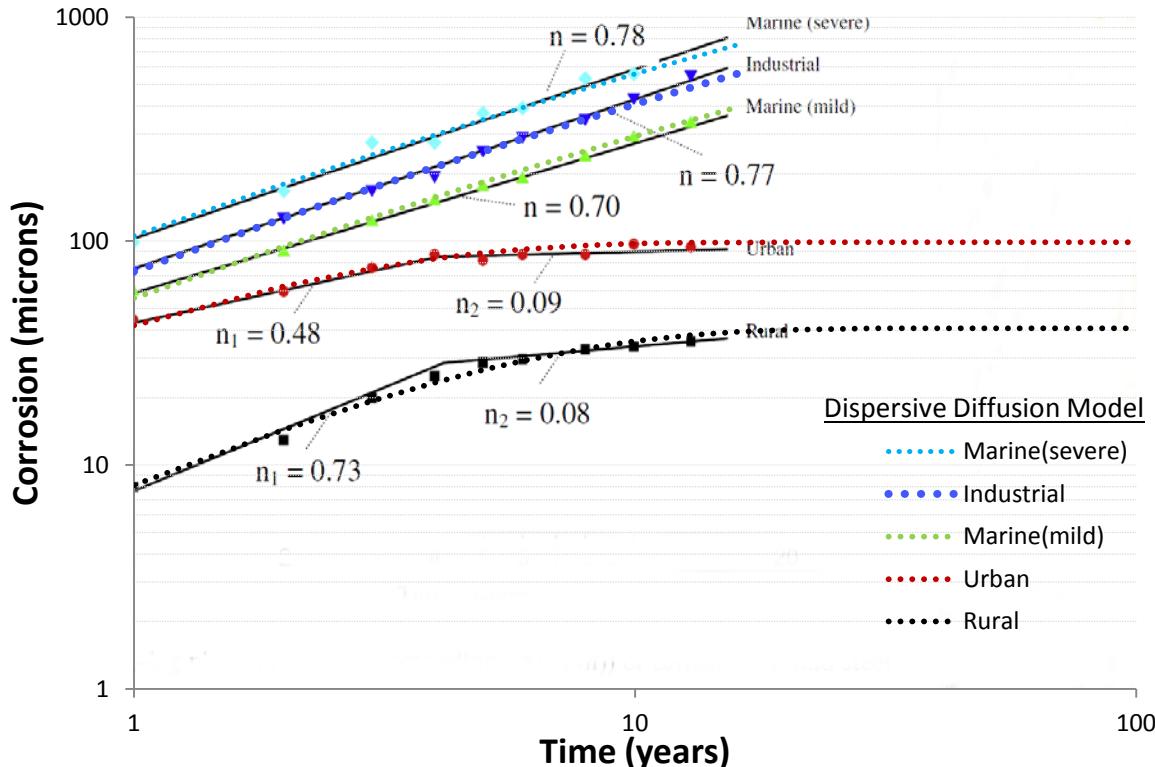


Figure 16: Model fits assuming the Ornstein-Uhlenbeck process. The urban and rural corrosion rates show a stronger asymptotic trend, indicating a reversion to the mean.

Conclusions

We have presented a series of physics-inspired models that approximate assorted diffusion phenomena well enough to be useful for model based engineering, providing significant advantages over both physical testing and reliance solely on data tables drawn from past testing. It should be noted that the question of when an approximate model is sufficiently accurate for a specific use depends upon the requirements that intended use places upon the model.

The statistician George E.P. Box wrote a famous quote concerning the validity of models. Box's oft-quoted statement of "All models are wrong, but some are useful" can be (and at times has been)

misinterpreted. The original quote in full, drawn from the book “Empirical Model-Building and Response Surfaces” by Box and Draper, provides useful context:[14]

“The fact that the polynomial is an approximation does not necessarily detract from its usefulness because all models are approximations. Essentially, all models are wrong but some are useful. However, the approximate nature of the model must always be borne in mind.”

On the rest of that page, Box and Draper present a concise description of the differences between *epistemic* and *aleatoric* uncertainty, which is a crucial distinction for understanding the range of valid use for the models we present here. Epistemic uncertainties are the systematic errors (model uncertainty) that one can introduce in a statistical model, while aleatoric errors are those that are fundamental in the natural behavior itself (parametric uncertainty), be it noise or some other random effect.

The distinction between epistemic and aleatory uncertainty is central to understanding the scope of applicability of the oxide growth and thermal modeling described in this paper, and of the validity of environmental models in general. We can understand the basic mechanisms of oxide growth over many orders of magnitude via the parabolic Fickian diffusion law, but we may miss important details in how we choose to numerically model the fundamental equations. To remedy this situation, we treated the diffusion coefficient and the Si/SiO₂ interface location with the correct amount of aleatory uncertainty. Epistemic uncertainty remains potentially effecting the validity of the model we applied, which is addressed in accurately we can measure against the empirical observations.

The decision on whether a specific model is valid for a given use thus depends on how closely the problem at hand corresponds to data against which the model has been compared. Were the observational results and application identical across available data sets, applying a sophisticated model, beyond a heuristic to match the data, would not be necessary. Repeatability leading to predictability is the key here, and that is why characterization in the semiconductor industry has been historically a critical factor in producing working chips.

If, on the other hand the case at hand is a serious extrapolation from past experience, careful consideration must be made of what effects the model may be neglecting and whether these effects could make the model’s behavior misleading for the extrapolated case. Between the situations where a model is unnecessary and one where it is perhaps inappropriate (due to lack of information, etc), are a range of valid uses where the question at hand is how much error should be expected in model outputs. Our treatment of aleatory uncertainty provides a means to estimate expected error, when the question of model uncertainty has been appropriately addressed.

In that sense we can cast Box’s phrase as essentially cautioning for appropriate care in determining appropriate use of models and for estimating numerical errors in statistical modeling. To interpret Box’s quote as questioning the validity of using models in the first place would clearly be an incorrect argument — look at how far mathematical engineering based on models has gotten us! The Deal-Grove model essentially allowed oxidation processes to become well characterized and predictable, which was central to revolutionizing the integrated circuit manufacturing process. The dispersive diffusion model derived in this paper should be taken in that spirit, a pragmatic model that could become useful in characterizing a fundamental physical process.

References

- [1] J. Dunkel and P. Hänggi, "Relativistic Brownian motion," *Physics Reports*, vol. 471, no. 1, pp. 1–73, 2009.
- [2] B. Baeumer, M. M. Meerschaert, and M. Naber, "Stochastic models for relativistic diffusion," *PHYSICAL REVIEW E Phys Rev E*, vol. 82, p. 011132, 2010.
- [3] B. E. Deal and A. Grove, "General relationship for the thermal oxidation of silicon," *Journal of Applied Physics*, vol. 36, no. 12, pp. 3770–3778, 1965.
- [4] U. Khalilov, E. C. Neyts, G. Pourtois, and A. C. T. van Duin, "Can We Control the Thickness of Ultrathin Silica Layers by Hyperthermal Silicon Oxidation at Room Temperature?," *The Journal of Physical Chemistry C*, 2011.
- [5] M. Uematsu, H. Kageshima, and K. Shiraishi, "Microscopic mechanism of thermal silicon oxide growth," *Computational materials science*, vol. 24, no. 1, pp. 229–234, 2002.
- [6] P. R. Pukite, *The Oil Conundrum: Vol. 1 Decline, Vol. 2 Renewal*, vol. 1,2, 2 vols. Daina, 2011.
- [7] R. D'Almeida, S. Gonçalves, I. Baumvol, and F. Stedile, "Diffusion-reaction in thermal growth of silicon oxide films on Si," *Arxiv preprint cond-mat/9901335*, 1999.
- [8] M. Sullivan, B. Thompson, and A. Williamson, "An experiment on the dynamics of thermal diffusion," *American Journal of Physics*, vol. 76, p. 637, 2008.
- [9] H. J. L. Witte, G. J. van Gelder, and J. D. Spitler, "In Situ Measurement of Ground Thermal Conductivity: The Dutch Perspective." [Online]. Available: <http://www.groenholland.nl/download/Ashrae-108-1.pdf>. [Accessed: 14-Mar-2012].
- [10] H. Witte and A. van Gelder, "Geothermal Response Tests using controlled multipower level heating and cooling pulses (MPL-HCP): Quantifying groundwater effects on heat transport around a borehole heat exchanger," *Proc. Ecostock*, 2006.
- [11] B. Groenholland, "Trial borehole & TRT - site testing & characterisation - Consultancy - Groenholland - Geo Energy Systems." [Online]. Available: http://www.groenholland.com/nl/consultancy/site_testing_and_characterisation/trial_borehole_and_trt.php. [Accessed: 29-Mar-2012].
- [12] D. de la Fuente, I. Díaz, J. Simancas, B. Chico, and M. Morcillo, "Long-term atmospheric corrosion of mild steel," *Corrosion Science*, vol. 53, no. 2, pp. 604–617, Feb. 2011.
- [13] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis Of Real-World Signals*. A K Peters, Ltd., 2010.
- [14] G. E. P. Box and N. R. Draper, *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.

Knowledge-Based Environmental Context Modeling

Paul Pukite, Steve Bankes, Dan Challou: *BAE Systems*
Thomas Huang: *JPL*

Abstract: This paper describes a semantic web architecture based on patterns and logical archetypal building-blocks well suited for comprehensive environmental modeling framework. The patterns span a range of features that cover specific land, atmospheric and aquatic domains intended for terrestrial and amphibious vehicles. The modeling engine contained within the server relied on knowledge-based inferencing capable of supporting formal terminology (through the SWEET ontology and a domain specific language) and levels of abstraction via integrated reasoning modules.

Contents

Introduction	3
Building Blocks	5
Triple Store.....	6
Knowledgebase	7
Entailment	7
Search and Query	8
Data Formats	8
Ontology Graphing.....	8
Code Generation.....	8
Math and Complex Numbers.....	9
Array and List Processing	10
Statistical Integration.....	11
HTML Development	12
Plot Artifacts	13
Random Number Generation.....	14
Dimensional Units Checking.....	14
Diagramming Relationships.....	14
Summary of Building Blocks	14
Pattern Architecture	16
Archetypal Domain Features	17
Fine Terrain Features.....	18
Gross Terrain Features	18
Wave Energy Statistics.....	18
Wind Energy Statistics	18
EMI Clutter Modeling	18
Inland-water Statistics	18
Particle Size Statistics	18
Thermal Dispersion	18
Rainfall Statistics.....	19
Corrosion and Oxidation	19
Information Resources.....	19
Archetypal Usage Facets	19
Dynamic context server knowledgebase and reasoner	20
Example of Environmental Requirement	21
Browsing Interface	22
Reference and Citation Linking.....	23
Workflows.....	23
Search.....	24
Resources	25
Map/Location	28

Knowledgebase and Server Technical Architecture	29
Examples of usage	32
Sea State View	32
Obstacles	34
Fine-relief Terrain View.....	36
Markov and semi-Markov Processes.....	36
Superposition of Sine approach.....	41
Gross-Relief Topographic View.....	42
Temperature	43
Seasonal Model	43
Diurnal precision model	45
Thermal Diffusion	46
Stream Fording.....	47
Clutter Integration	48
Corrosion Example.....	49
Droplet Size.....	49
Wind.....	50
Pressure	51
Buoyancy Example.....	52
Patterns of Usage.....	53
Development Process	53
Summary.....	54
Annexes	55
Annex 1: Browser Narrative.....	55
Annex 2: PDF Models.....	57
Annex 3 : Installation	59
Run-time Packages Required.....	59
Annex 4 : Units	60
Standard Atmosphere properties.....	60
Water properties	60
Solar	60
Physical Constants.....	60
Units suitable for conversion	61
Annex 5 : Table of Acronyms	62
References	64

Introduction

The design of ruggedly complex systems such as advanced ground vehicles requires knowledge of the environmental contexts that occur during operational deployment. For example, the nominal and extreme terrain and weather conditions have an impact on the choices made in the vehicle design. A test vehicle has to endure and even thrive in extreme conditions to earn the title of a ruggedized design.

If you want to know whether a vehicle can travel over hilly terrain, it's a good idea to understand the extent of the hilliness in the region, and characterize that in terms of probability of elevation changes.

If you want to estimate how a vehicle will respond to a rocky road, it makes sense to characterize the bumpiness and fine relief structure and formulate that in terms of a suspension and chassis simulation model.

If you want to predict how a vehicle will respond to windy conditions, it is useful to have estimates and likelihood of the extreme conditions that may arise for that climate.

Figure 1: Examples of rationale for characterization and context modeling. Our concentration is on the invariant aspects of the environment that exist independent of the subject's role. These are considered non-compliant one-way relationships, as a vehicle can't impact certain exogenous properties.

Based on these considerations, we need a comprehensive approach for aggregating model knowledge. The approach that we outline below extends from research describing the elements necessary to create workflow architectures for vehicle design[1]. In particular, the building blocks that originated from design-centric semantic web features together with knowledge-based reasoning were deemed general enough to find applicability to an environmental modeling framework (see **Figure 2**).

Definition: Environmental models for virtual integration and testing of new vehicle designs, at system, subsystem, component levels.

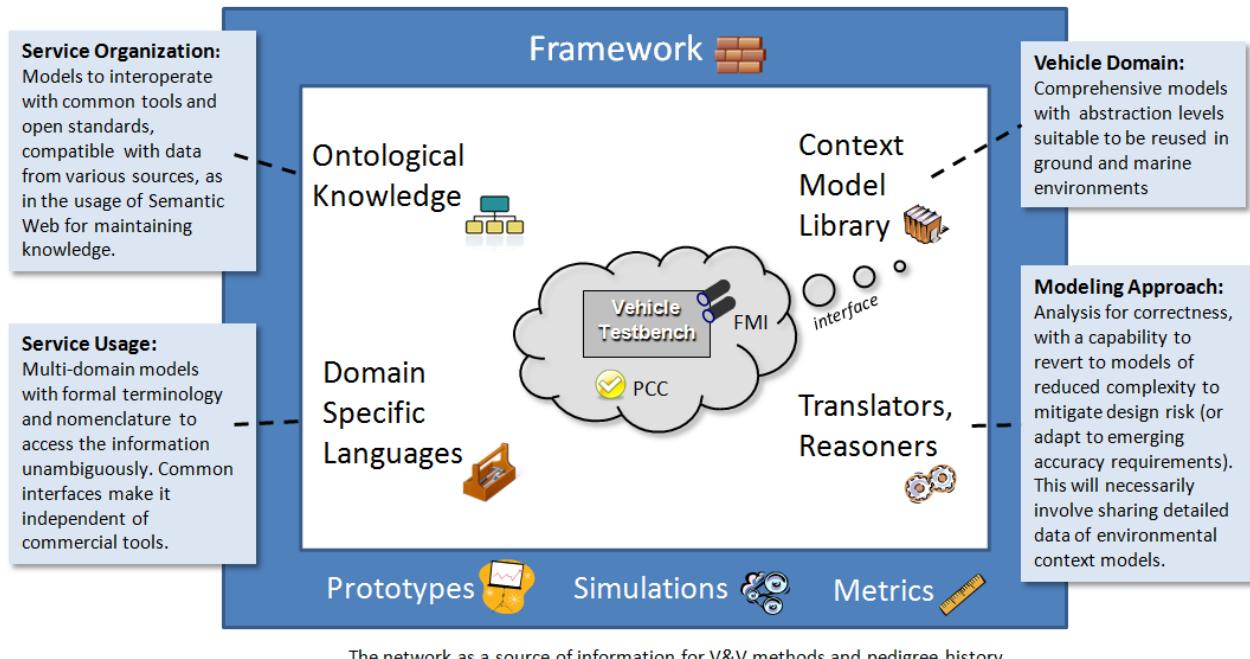


Figure 2: View of context modeling framework with emphasis on vehicle design. For virtual simulation, the innermost vehicle test-bench needs to interface to the environmental context model.

Starting from a general design and workflow approach outlined elsewhere[1], we applied comparable knowledge-based patterns to environmental modeling (see **Figure 3** below). At the modeling level, we

consider abstractions, organization, search, and logical reasoning to improve the convenience and usability across the suite of models.

Since the models are quantitative and often statistical in nature, we employ a concise descriptive language to declaratively specify their behavior. Simulation abstractions allow us to transition between purely data-driven and probabilistic views of models. The convenience of automated searching and reasoning benefited from the semantic-based organization of the underlying knowledge-base.

	Examples	Benefits
Domain Specific Language	Σ Statistical, complex, and matrix math.	Concise, symbolic, declarative, with formal representations
Abstraction Levels	 Local/global spatial levels, probability distributions	Comprehensive, detailed, and aggregated views of environmental contexts
Organization	 Semantic web ontologies such as SWEET	Classification, maintenance, services, library curation
Searching	 Linked data elements, keyword and classification	Models discovered through requirements linkage, etc
Reasoning	 Workflow and parametric modeling	Guided model generation and usage.

Figure 3: Knowledge-based modeling patterns

The objective is to present the environmental view of the system as a dynamic context server (DCS). This provides a flexible framework for handling interactive applications such as guided workflows, artifact viewing, and reusable web services.

The DCS was organized with a breadth-first view to make sure that the essential semantic, ontological, logical, and mathematical capabilities were at least tangentially covered. We start with a set of categorical domain features which cover land, atmosphere, and aquatic environments, and then apply a set of facets to these categories to allow the user various levels of access to the knowledge. The orthogonal axes are illustrated in **Figure 4**.

The basis for the modeling is described in a set of foundation papers [2][3][4] which applied stochastic patterns to the empirical observations, leading to a set of concise formulations. The semantic organization provided a means to manage the growing array of patterns. Much as a scientific library provides organization among the subject domains, the semantic layers afforded a similar discipline to the model hierarchy. In practical terms, we applied the concept of archetypes as patterns of use[1] and organize our knowledgebase at the semantic level to take advantage of the reuse and commonality available by suitable classification.



Figure 4: The general semantic organization corresponds to modeling domain features arranged against user facets. The features provide semantic keying for various levels of search. In terms of the SWEET ontology, the features correspond to environmental categories, while the facets are human-centric.

Building Blocks

The foundation of the context server is built on a knowledgebase based on triple-store technology and managed by a declarative first-order logic language. The language chosen, Prolog[5], bridged the gap between the (1) low-level structures of the underlying triple-store database and the (2) declarative structure of an abstracted user interface[6]. The rules and inferencing capabilities of the language provided sufficient expressivity to allow the context server to function as a flexible semantic web-server[7].

In the following overview, we will cover the following building blocks at least briefly:

- Integration with triple-store (RDF, OWL, SWEET)
- Knowledge base and logic formulation
- Entailment – creating rules that abstractly appear as triple store
- Search and query – Semantic Web servicing – integration of queries as services
- Processing data formats such as XML and JSON
- Ontology graphing
- Generators for code production and data
- Math and complex math
- Array and list processing
- Symbolic and dimensional unit processing
- Integration with statistics framework such as R

- Artifact generation such as plots for PDF and PSD

Triple Store: The triple-store data format is the backbone of the semantic web. A triple consists of a (*Subject, Predicate, Object*) tuple which can be linked to other triples in seemingly infinite combinations, leading to significant flexibility and robustness in a database.

The structure of a triple, cast in a conventional *resource description format* (RDF)[8] or by the XML-free Turtle format, essentially reduces to mapping against what is referred to as a Prolog functor. So the following query maps directly to a RDF triple-store:

```
rdf(Subject, Predicate, Object)
```

From a programmatic level, that is all there is to interfacing Prolog to a triple-store. The semantics turns out to be equivalent to that of a SPARQL query[9], where the binding of the arguments to either variables or ground terms determines what will be returned from the query. Several useful guides are available which describe the essential match between the Prolog binding mechanics and that governing SPARQL queries and description logic written in straight RDF. Further work has been done in mapping to higher level descriptions built on top of RDF, such as in the *web ontology language* known as OWL[8].

We take advantage of a ready-made OWL classification system set up for environmental sciences modeling by using the terminology defined by the OWL-based *Semantic Web for Earth and Environmental Terminology* (SWEET) ontology[10][11]. The key to using ontologies such as SWEET is to create triple-store instance data which subclasses or keys off the classification system defined by the SWEET environmental hierarchies (see **Figure 5**).

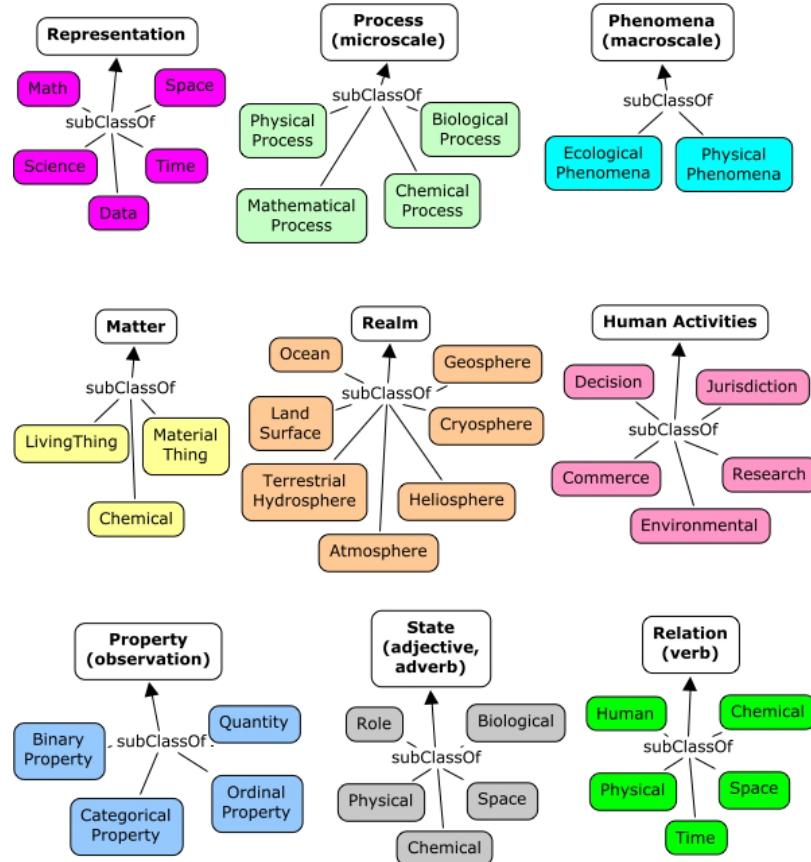


Figure 5: SWEET hierarchies and environmental classifications. The enclosed terminology is sufficient to define the majority of the context modeling knowledge.

We apply an additional abstraction, which is quite common in practical semantic web applications, which is to define a namespace layer between the instance data and that defined by SWEET. In Figure 6 below, we show several DCS-defined sub-classed resources that start with the letter ‘a’, and associate that with the SWEET terminology class that it best fits into.

Context Server Resource	SWEET (<code>rdfs:subClassOf</code>)
<code>ent:atm</code>	<code>reprSciUnits:Unit</code>
<code>ent:atomicUnit</code>	<code>matrParticle:Atom</code>
<code>ent:averageAlbedo</code>	<code>propFraction:Albedo</code>
<code>ent:averageSolarInsolation</code>	<code>propEnergyFlux:Insolation</code>
<code>ent:avogadrosNumber</code>	<code>propQuantity:PhysicalConstant</code>

Figure 6: First several instances in graph `ent = http://entroplet.com/terms#` sorted by label

Suffice to say that the SWEET terminology supplies very complete coverage to the environmental modeling features required in the context server, and we have updated SWEET in the few areas that were deficient in representation power.

Knowledgebase: The context server knowledgebase is a composite of all the information stored in the triple-store database and the rules and facts stored in the Prolog repository. The most elementary form of a rule is to express the logic as some form of constraint, relationship, query, etc.

Rules can be used to define a specific closure or find a specific closure in the underlying triple-store database (see Figure 7)

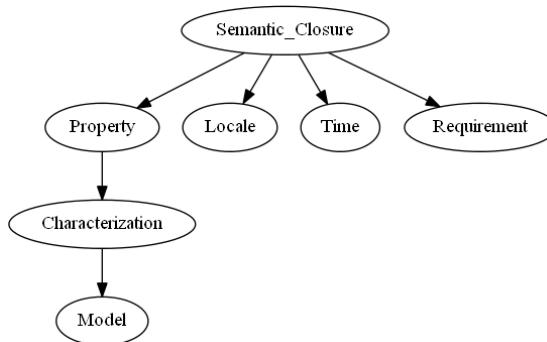


Figure 7: Linked triple-store features from a top-level rule define a semantic closure.

Entailment: A powerful feature of the correspondence between the declarative semantics of Prolog and a triple-store query is that we can create rules that abstractly function as triple stores (or that can extend the triple store). Per the semantic definition, *entailment* is the principle that the truth of one statement ensures the truth of a second statement. For example, we can extend an RDF query[12] that returns a resource with a real-valued component, so that we do not have to perform further parsing to convert the string to a float.

```
rdfR(Subject, Predicate, RealObject)
```

Search and Query: We use a version of Prolog that contains a complete web servicing library (called Cliopatra[6]). This allows searches and queries to be composed as web requests. For example, the following URL request searches for resources that fit within a specific SWEET category (in this case a wind category):

```
http://localhost:3020/context_search/list_cats?name=phenAtmoWind:Wind
```

This works on two levels, the first stage of processing decomposes the URL into an internal format via REST (representational state transfer) semantics, and the second level, which searches the triple-store database (or Prolog knowledgebase) for matches, and then presents the results to the client.

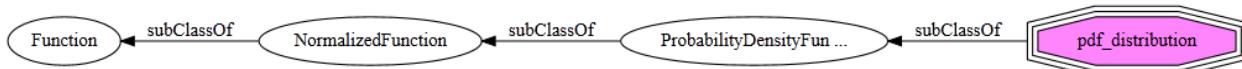
As another example, this query requests a model of ocean clouds (see Annex B)

```
http://localhost:3020/context_model/apply?loglog=false&model=Ocean_clouds
```

Semantic web servicing and specifically the integration of queries as services allow us to build up the capabilities of the context server. In that sense, we can use (1) queries as model services, (2) queries to generate algorithms and code, (3) queries to generate artifacts and metrics, and so on.

Data Formats: Processing data formats such as XML and JSON is important with respect to reading from externally stored knowledge and to providing data to the user. Prolog is essentially a list-processing language and much of the XML and JSON translation is conveniently available through the built-in libraries.

Ontology Graphing: Also built-in as a library is graphical representation for the knowledgebase and its semantic hierarchy. For example, the image below represents the sub-classing of a context modeling class, ent:pdf_distribution, with respect to the SWEET terminology.



Code Generation: Many environmental models are stored as mathematical representations, intended to express a model's behavior as closely to its original abstract form as possible. If this format is unambiguous and thus amenable to straight-forward parsing, the representation can easily be transformed into any software language that can also represent the mathematical expressions.

In certain cases, we have chosen to represent models in a symbolic math format. Here, we can create a suite of translators for conversion from that format to an arbitrary language with the necessary mathematical expressiveness. We have shown below an example for a temperature model, where **T** is the calendar time and **S** returns the symbolic temperature expression.

```
% Generate a symbolic algebraic expression
calculate_symbolic_temperature(Site, T, S) :-
    rdfs(U, ent:name, Site),
    rdfR(U, ent:t0, T0),
    rdfR(U, ent:ty, Ty),
    rdfR(U, ent:dt, DT),
    rdfR(U, ent:td, Td),
    rdfR(U, ent:a, A),
    rdfR(U, ent:b, B),
    rdfR(U, ent:c, C),
    PI is pi,
```

```
S = T0+Ty*sin(2*PI/365*T+A)+(DT*sin(2*PI/365*T+B)+(Td-DT))*sin(2*PI*T+C).
```

The expression **S** can be evaluated directly as below, where the term “**is**” directs the interpreter to evaluate the expression

```
% Generate a numeric from symbolic
calculate_numeric_temperature(Site, T, Temperature) :-
    calculate_symbolic_temperature(Site, T, Symbolic_Temperature),
    Temperature is Symbolic_Temperature.
```

Since **S** is a symbolic representation, it can straightforwardly be translated to C-code, Python, etc. We will demonstrate this later.

Math and Complex Numbers: To minimize the potential complexity of the mathematical representations, we can take advantage of certain idioms that are heavily used and maintain those as domain specific representations. In that sense, we can create a domain specific language to handle the essential and common aspects of math-based models.

Standard Prolog is expressive enough to handle much of the math load as shown with the code production example described previously, yet we find that a mini-language is useful to handle array processing and complex number math.

As an example of the potential conciseness of representation, the complete declarative representation of a recursive Fast Fourier Transform (FFT) is the following (the *product_and_sum* and *evens_and_odds* are helper functions, and *w* is a lookup table).

```
% Fast Fourier Transform
fft(1, Ft, Ft) :- !.
fft(N, F, Ft) :-
    N > 1,
    N2 is N // 2,
    evens_and_odds(F, E, O),
    fft(N2, E, Et),
    fft(N2, O, Ot),
    w(1, W1),
    w(2, W2),
    w(N, Wn),
    product_and_sum(Et, Ot, W2, Wn, Gt, []),
    product_and_sum(Et, Ot, W1, Wn, Ft, Gt).
```

The function *product_and_sum* illustrates how we have abstracted the expression evaluator “**is**” (described in the previous code production example) to “**isx**” which now evaluates expressions involving complex numbers. A complex number is represented as $a + i b$, where i represents the imaginary number $\sqrt{-1}$.

```
product_and_sum([], [], _, _, Ft, Ft).
product_and_sum([E| Et], [O| Ot], Wk, Wn, [F| Ft], Fu) :-
    Temp isx O * Wk,
    F isx E + Temp,
    Wk1 isx Wk * Wn,
    product_and_sum(Et, Ot, Wk1, Wn, Ft, Fu).
```

This complex number formulation is used for calculating terrain profile power spectral densities (PSD) based on input data with array sizes of several hundred thousand elements.

Array and List Processing: Concise list array processing is useful to represent and generate the data elements described in most of our stochastic environmental models. Complex number list processing is used for calculating PSD's as shown in the complex math FFT example, while lists comprised of real numbers are used for calculating probability density functions (PDF).

As a domain specific representation we chose to apply list and array operators with *infix* notation wherever possible. The processing function then reads as *Y results from applying against X*.

Y mapdot X

Map Dot is a dot product of individual elements, retaining the list structure.

For this specific case we take an example from the Matlab world where the operator “`.*`” is defined to multiply a scalar by a vector, so that the following relation holds:

```
[20,40] mapdot 2 .* [10, 20]
```

This expresses the invariant that 20 is the doubling of the first element in the list, and 40 is the doubling of the second element of the list.

For further expressiveness, the term “`~>`” is used to indicate a map apply to a list:

```
[2.0, 4.0] mapdot 2 * sqrt ~> [1.0, 4.0]
```

A number of specialty array and list processing functions follow this infix idiom:

Y range [From, To]

Range of numbers into a list

Y dot A*B

Dot product of two arrays or lists of the same length

Y convolve A*B

Convolve (convolution operator) list with another list.

Y correlate A*B

Pair correlates one list with another list.

Y derivative A/B

Take the derivative of one list with respect to another list.

Y integrate A*B

Take the integral of one list with respect to another delta list.

Y difference A-B

Take the difference of one list with respect to another list.

Y tuple A+B

Create a tuple list from two lists.

Y unbias X

Remove the mean from a list of values, so the sum is zero.

Y normalize X

Normalize a list of values, so the sum is unity.

Y pdf X

Create a PDF from a list of values.

Y zshift A-B

Do a DSP z-shift from a list.

Y window X/A***Y window X*A***

Do a lag-window or centered window on a list.

Y shrink A/B

Shrink a list to match the second.

Y expand A+B

Expand a list to match the second.

Y cat X

Flatten a list.

Y offset X

Offset a list by N elements.

Y ordinal X

Create an ordinal (counting) list from a list.

Y split X

Split pairs of values into two sequential lists.

As an example of the sophistication of use, these four lines generate a number line, a sine and cosine valued list and then a list of (X, Y, Z)-tuples suitable for plotting as a graph:

```
X range [1,50]/0.1,  
Y mapdot sin ~> X,  
Z mapdot cos ~> X,  
Graph tuple X + Y + Z
```

Or consider this case of concisely generating a cumulative density function against an X variate of a given range, and then producing a probability density function from that result

```
X range [0,100]/0.1,  
Y mapdot exp(10) ~> X, % create CDF of exponential with damping factor 10  
Z pdf Y
```

The final objective was to be able to abstractly represent fairly sophisticated list computations in as concise a form as possible. As is the case with many of these domain specific idioms, the result is that a user can scan the code and be able to quickly determine the transformation that is being applied.

Statistical Integration: A language such as Prolog is well-suited for (1) general purpose logic programming, (2) creation of domain specific languages (as shown with the complex and list processing DSLs described above) and (3) application development such as is required for a semantic web server.

Where it falls short is in the area of special-purpose computing such as in statistical and scientific computation, which may require extensive matrix handling and comprehensive math libraries. The open-source statistical computational tool known as **R** is well-suited to corners of the statistical and

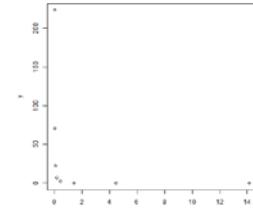
mathematical world that basic Prolog does not have a foothold in¹.

Fortunately, the version of Prolog used on this project has an API to **R** and features a domain specific language abstraction defined specifically for data processing. Similar to what we use for list processing, an equivalent list processing interface is defined to **R**.

In particular, the infix operator “`<-`” is used to generate an input/output transfer from Prolog to **R** (and the reverse). So in the following, we generate a list profile called *Input*, but then request that **R** perform a *BesselK* function lookup on that list and attach it to an **R** state vector called *y*.

```
Input mapdot sqrt ~> 2.*[0.00001,0.0001,0.001,0.01,0.1,1.0,10.0,100.0],
y <- besselK(Input,1),
Y <- y,
Output mapdot Input * Y
```

The rationale for doing this is that Prolog does not have a library of transcendental functions such as the set of modified Bessel functions (which are used in PDF models), yet **R** does contain a comprehensive library, largely due to its statistics underpinnings. The fact that the list processing idioms that we have in place match well with the list I/O of the **R** library makes this a good combination for corner-case modeling needs, and where more sophisticated graphing is needed (see the contour plots in Section X for an example). The combination of Prolog with **R** is comparable to the capabilities of Matlab.



The statistical leverage that **R** provides to the semantic web server is crucial for further enhancements to context modeling.

HTML Development: The semantic web library includes convenient mechanisms for developing HTML code, well suited for making sophisticated interactive web pages. The declarative style of programming used in Prolog allows a very similar but much more concise and powerful representation than does the XSLT (Extensible Stylesheet Language Transformations) language[13] used in conventional web page development.

The library uses what is called definite clause grammars (DCG) to generate valid HTML[14], with the code remaining very readable. A DCG is much like a XSLT fragment, allowing for recursion and pattern matching, but presented in a much more compact and concise style.

```
dispatch(wind) -->
  html([h1('Wind models'),
        \g(search,
          a([href('/context_model/navigate?characteristics=windSpeed'),
             target(target_iframe)], 'Wind PDF models')),
        \g(ref_search('phenAtmoWind:Wind', 'Wind references')),
        \g/browse,
          a([href('/context_browse/navigate?term=atmospheric']),
             'Browse atmospheric characteristics')
        )
    ]).
```

This HTML, when rendered, appears as the following style-sheet-based representation (see **Figure 8**):

¹ See <http://blog.revolutionanalytics.com/2012/07/a-big-list-of-the-things-r-can-do.html> for a list of R application areas.: Basic math and statistics, probability distributions, big data analytics, optimization and mathematical programming, signal processing, simulation and random number generation, statistical modeling, statistical tests, static and dynamic graphics.

Wind models

-  [Wind PDF models](#)
-  [Wind references](#)
-  [Browse atmospheric characteristics](#)

Figure 8 : Rendered HTM from a Prolog DCG rule.

The terms leading with the character ‘\’ call out user-defined DCG functions which will create in-place snippets of HTML code. In practice, by generating the majority of the HTML code using a mix of logic and meta-logic, we employ what amounts to a hybrid of a PHP hypertext preprocessor[15] (executed on the server side) and the declarative nature of XSLT (but without the excessively verbose syntax of the latter).

In general and for most HTML coding we use this idiom, and only for certain applications, such as inform input look-ahead and interactive graphing, do we drop into JavaScript.

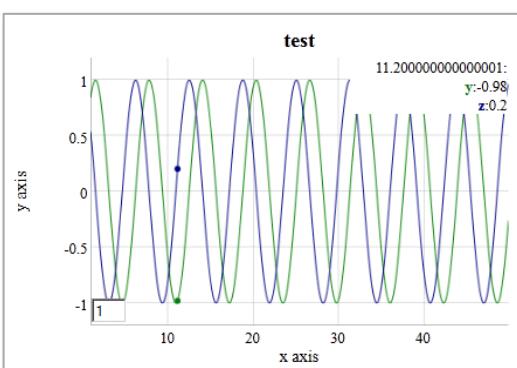
Plot Artifacts: Generating artifacts such as plots of PDF and PSD models is facilitated by the use of open-source JavaScript libraries that feature dynamic interaction.

From either models or data, we use dynamic graphs to construct PDF, CDF, and PSD profiles, along with expected value plots of annual, diurnal, etc measures and various growth curves.

The graphing interactive features include linear or log scales, adjustable noise filtering, the inspection of data points, and panning and scaling.

As an example, we extend the list processing example described earlier to plot a dynamic graph, which consists of two trigonometric functions separated by a phase shift:

```
dygraph_test(_) :-
    X range [1.0,50.0]/0.1,
    Y mapdot sin ~> X,
    Z mapdot cos ~> X,
    Graph tuple X + Y + Z,
    reply_html_page([title('chart'), \(\cong_text:style_cliopatra)], [
        \(\context_graphing:dygraph_native(
            false,
            ['x', 'Y', 'z'],
            'x axis',
            'y axis',
            'test',
            Graph))]).
```



Random Number Generation: A module for random number generation is an important building block for Monte Carlo simulations. A uniform random variate generator can be used in conjunction with invertible PDF profiles to create sampled data for exponential, Bessel, Rayleigh, fat-tail, and several other distributions.

Terrain profile generators also rely on a random number generator to create random walk profiles of Markov and semi-Markov character. For aquatic wave generation, random superposition of sine waves is effective in modeling the empirical characteristics observed.

Dimensional Units Checking: The framework uses an embedded Prolog reasoner to parse and pattern match if dimensional unit conversion is required on some measure. The unit symbology is retained as lexical terms with both proportional scaling or inverse scaling allowed, depending on whether a “*” or “/” appears in the unit dimensionality string. A database of root units and at least one relation to another unit of the same measure is kept in the local knowledgebase so any combination can potentially be parsed and evaluated.

As an example of a relatively complex unit string, we take the Stefan-Boltzmann constant specified in terms of SI units, $5.67 \times 10^{-8} \text{ J/s/m}^2/\text{K}^4$, and can use the unit checking module to convert this value to a hybrid mix of English units:

```
?- context_units:convert(5.67e-8*j/s/m^2/k^4, SB*btu/hr/ft^2/r^4, SB).  
SB = 1.7121822360279162e-9.
```

Where j=joules, s=seconds, m=meters, k=kelvin on the original, and btu=BritishThermalUnits, hr=hour, ft=foot, and r=Rankine on the desired conversion.

This feature is valuable in reducing the amount of translation code that is required, as the number of combinations is nearly unlimited but the parser is able to perform the reduction symbolically.

Diagramming Relationships: Similar to the library for R interoperability, the semantic web infrastructure also integrates to the directed-graph library known as *Graphviz*. The *Graphviz* rendering generates SVG (scalable vector graphics) which is compatible with most HTML browsers.

This capability allows us to visualize closure of rules as a directed graph, and look at hierarchy and cloud diagrams.

Summary of Building Blocks: The portability of the semantic web infrastructure has been tested against Windows, Linux, and a cloud computing environment. The maintenance of the context model library is abetted by automated features such as web-served browsing of source code and triple-store data elements, site roadmaps and documentation, statistical usage, and password authentication for administration of repository and web server options.

In summary, the integration of a logic processor with triple-store semantic data formatted with RDF and Turtle works effectively as a building block foundation. Using the triple-store for input data files and as a knowledge repository interoperates well with the base logical language. The SWEET ontology provides comprehensive classification and we can build an extensible knowledgebase on top of the SWEET terminology

The domain-specific language for creating stochastic context models includes list processing of n-tuples, math operations on lists, constructors of linear and log range ordinals, dot product and mapping on lists, the latter which we can apply functions and scaling. Specialty functions such as convolution, pair

correlation, Z-difference, integral, derivative, histogram, and simple translation of tuple-lists to graphs make it very convenient for data processing applications.

The domain-specific language for complex-number processing has arithmetical infix operators for multiply, addition, division, power, etc which allows for concise semi-Markov terrain profile modeling, and a built-in FFT for PSD calculation.

The following table lists the categorized building blocks so far described:

Table 1: The set of Domain Specific Language building block elements for environmental modeling covers math and logic functionality

Functionality	Terms	Description	Examples
complex math	isx (evaluation) & (constructor)	Manipulating complex numbers used for generating PSD curves.	Num isx 1&2 * 2&1
array math	dot (scalar result) mapdot (array result) ~> (apply function)	High-level array manipulation.	z dot Array1 * Array2, Amp mapdot sqrt ~> PSD * Sx
statistical	<- (translate to R)	Interface to the R statistics package	y <- BesselK(1.0,0.0), Y <- y
geospatial	shape point line	Description of geometric shapes and lines	shape(point(52.3325,4.8673))
markup	html, table, p, b, body, etc.	Definite clause grammar for generating markup	table([border(1)], [tr([th('Mean'),th('Sampled')]), tr([td(Mean_Slope), td(S)])])
semantic	rdf	Library for interfacing to RDF and OWL knowledge	rdf(U, dcterms:'URI', Link)
symbolic logic	Prolog terms (including symbols and	Overall logic programming, dimensional checking,	Distance = 10.0 * meters

	variables)	code generation translation, etc.	
ordinary math	is (evaluation)	Set of math operators and libraries built-in to the language, used for models and Monte Carlo simulations	<code>Y is A + B</code> <code>random(Sample)</code>
	Prolog terms		
chart plot	library calls	Integrated JavaScript for artifact display	<code>context_graphing:plot(...)</code>
directed graph	library calls	Translation of triple- store links to directed graph	<code>graphviz:context_graph</code>

One row from this table was not used but has implications for environmental context modeling. The geospatial functionality is very useful for inferring information on locality. For example, say that temperature statistics are not known for a particular area, but data from nearby locations is available. A geospatial engine can aggregate and select data from weather stations in close proximity and use that to interpolate the statistics. This is built into the semantic web library, yet we deferred applying it until a future need arises.

Pattern Architecture

Based on the modeling described in our foundational research [2][3][4], where we discovered and collected patterns in various environmental contexts, we can start to encode these patterns to build up a comprehensive suite of interactive context models.

We first recognize that many of the context models have similar representation or archetypes, so for example, we can assert that PDF patterns use functional mapping on lists. And we also note that similar representational formats allow reusable artifacts in the form of plots. The basic application of these patterns results in an organization hierarchy shown in **Figure 9**, where we can apply the same development abstractions to each of the contexts that we wish to include. In this case, we consider a terrain context, and realize the same patterns of usage for modeling, language, reasoning, and metrics apply. This may seem overly pedantic, but the patterning approach applies to the development[16] just as it does to the model characterization[17], and it allows the framework to grow and scale. Declarative programming further facilitates pattern-matching, as a set of declarative rules can pattern match automatically on new information.

Organization Pattern

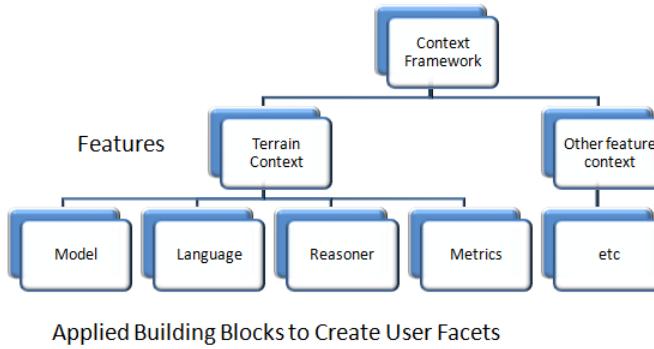


Figure 9: Pattern of associating domain features with building blocks which allow the creation of user facets and web services, maximizing reuse and maintainability.

An example of a basic semantic pattern that we can apply is to parameterize models for different geospatial locations (or locales, for short). One locale may be labeled as “**conus**” (shorthand for the continental USA), and used as a semantic node to attach various properties and characteristics to. **Figure 10** below shows the semantic closure graph for how the information gets attached to a locale node. Note that each graph edge represents a predicate in terms of a (*subject*, *predicate*, *object*) triple-store, so that the labeled arrow is the *predicate*, the source of the arrow is the *subject*, and the destination of the arrow is the *object*. This then creates a directed graph of triple-store links.

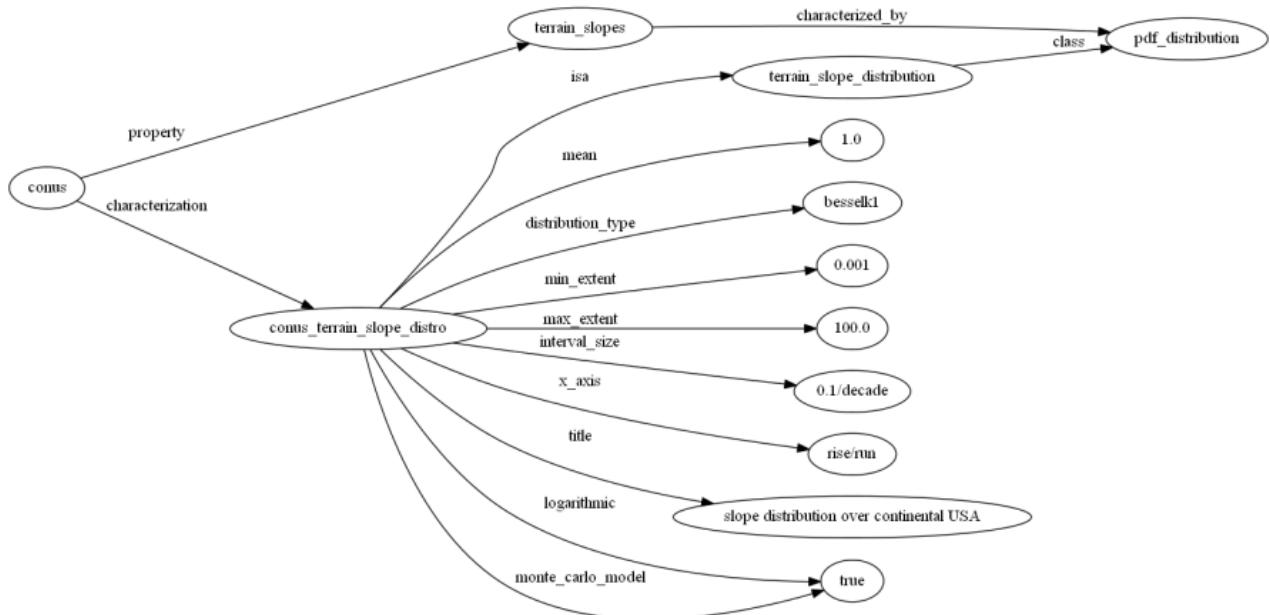


Figure 10 : Semantic closure graph of triple-store relations associated with a locale “conus” and terrain slope properties. Most rules follow a similar scoping closure.

This approach allows us to create a pattern architecture based on semantic links.

Archetypal Domain Features: The *domain features* of the architecture are the environmental contexts that we are interested in modeling. The following iconified list provides the basic categorization levels.



Fine Terrain Features — Models for representing power spectral densities (PSD) based on Markov and semi-Markov representations. Models of obstacle courses and data sets for friction and soil types.



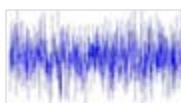
Gross Terrain Features — Simple distribution for sampling likelihood of terrain slopes. Marginal distributions for elevation changes over various geographic locations.



Wave Energy Statistics — Simple approach for modeling PDFs of wave frequencies and wave heights. Models of sea-state wave distributions over various geographic locations. Water density and buoyancy tables.



Wind Energy Statistics — General results for modeling wind energy distribution and autocorrelations for temporal persistence.



EMI Clutter Modeling — Maximum entropy models for electro-magnetic interference (EMI) and noise in the environment. Models of lightning rate.



Inland-water Statistics — Maximum entropy estimation for lake size distributions and river flow rates over various regions.



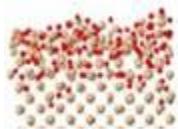
Particle Size Statistics — General method for modeling size distribution of particulates such as ash and ice crystals, and water droplets.



Thermal Dispersion — Simplification of thermal diffusion model to account for disorder and variability in the media. Seasonal and daily temperature models across geospatial regions.



Rainfall Statistics — General model for rainfall distributions within storms using composite process. Cloud size statistics.



Corrosion and Oxidation — Model of oxidation and corrosion growth which improves on the Deal-Grove formulation, using reversion-to-the-mean diffusion kinetics. Table of corrosive categories.

Information Resources: A number of commonly used properties and physical constants span several of these domain features. These can include:

- Physical constants such as water density, gravitational constant, etc.
- Lookup tables such as soil types, sea-state definitions,, etc.
- Nominal environmental and climate information such as standard atmosphere, solar insolation models, etc.
- Dimensional units such as SI and English
- Geospatial definitions and schema – such as UTM and Latitude/Longitude coordinate definitions
- Meta-information: Citations and references for models, algorithms, requirements
- Project specific context data sources
 - Verification & Requirements documents such as TOPS (Test Operating Procedures)[18][19][20]
 - Earth sciences archives such as NASA JPL PO.DAAC [21], US Army Corps of Engineers WIS [22], etc

Archetypal Usage Facets: The usage patterns follow an orthogonal axis that we refer to as usage facets.



Search — The discovery of models needed for a particular purpose is facilitated by various forms of search. A free-form search into the knowledgebase is provided by the search bar in the upper-right corner of the user-interface. This links to knowledge contained within the triple-store knowledgebase, largely independent of semantic context. Other more directed, semantically-driven searches are available from the main search page. Links between specific categories of knowledge and models available within the server are contained here. To accommodate this, specific models are tagged and allocated to specific environmental categories. This is aided by the application of ontologies such as SWEET.



Browse — Environmental and context knowledge follows a natural hierarchical organization. At the top level, we can break out the models into broad categories for Land, Atmosphere, and Aquatic. Below that level, the specific models are allocated to more finely refined categories such as terrain roughness. The basic hierarchy follows that of the SWEET ontology.



Workflows — A process workflow is defined as a software-guided navigation to problem solving. A composable workflow allows for a sequence of problem solving steps depending on the knowledge available. Several workflows to access probability density function (PDF) environmental models and power spectral density (PSD) models are provided. A supplemental system called OSCAR (Ontological System for Context Artifacts and Resources) provides semantic web discovery, registry, and

editing/modification capability. The OSCAR workflow portal will facilitate users to find context models and associated metadata to enable their simulation (see Appendix E).



References — Each model has supporting documentation in the form of references and citations. We also distinguish between specifications, foundational research, requirements, and general reference material. The Zotero citation management system[23] is used to keep track of references and links and we use the SWEET ontology to tag the references with semantic environmental categories. For example, references relating to aquatic wave energy will get tagged with the *phenWave:GravityWave* class resource defined in SWEET.



Resources — Context modeling resources include interactive links to tables and supporting documents, such as environmental regulations, standards, specifications, and nominal or typical operational profiles.



Map View — The scope of environmental models is world-wide. This of course implies that environmental models will depend on the particular geologic and climatic characteristics of specific geospatial locations. Where models have locality links we can search regions of interest to find what is available.



Generic Query — The knowledgebase has SPARQL and native Prolog query support. The intent is that a user can explore the database interactively and reuse resources from the triple-store in other applications.



User Factory — Project-specific requirements provide a means to connect practical applications to information available from an environmental context library. For example, a project requirement that states that a vehicle should be able to operate on terrain of a specific roughness, suggests a link to certain context models available from the server. The links between the requirements and models are accomplished via semantic and ontological organization of the knowledge. In this case, certain keywords and phrases in a requirements document are tagged and allocated to specific environmental categories. This is aided by the application of ontologies such as SWEET.

Dynamic context server knowledgebase and reasoner

The fundamental triple-store data structure generates a substantial fraction of the knowledgebase for the semantic web server. Triple-stores as realized by RDF data sets provide the flexibility to chain and link data at the granularity that we are interested in for context modeling.

The dynamic context server provides interactive content for environmental modeling on demand (i.e. essentially “serving” content). The models are contained within an ontologically organized structure and so can be semantically accessed. The semantic organization allows various search mechanisms, while the information and terminology domains are grouped in data stores according to the bubbles shown in **Figure 11** below.

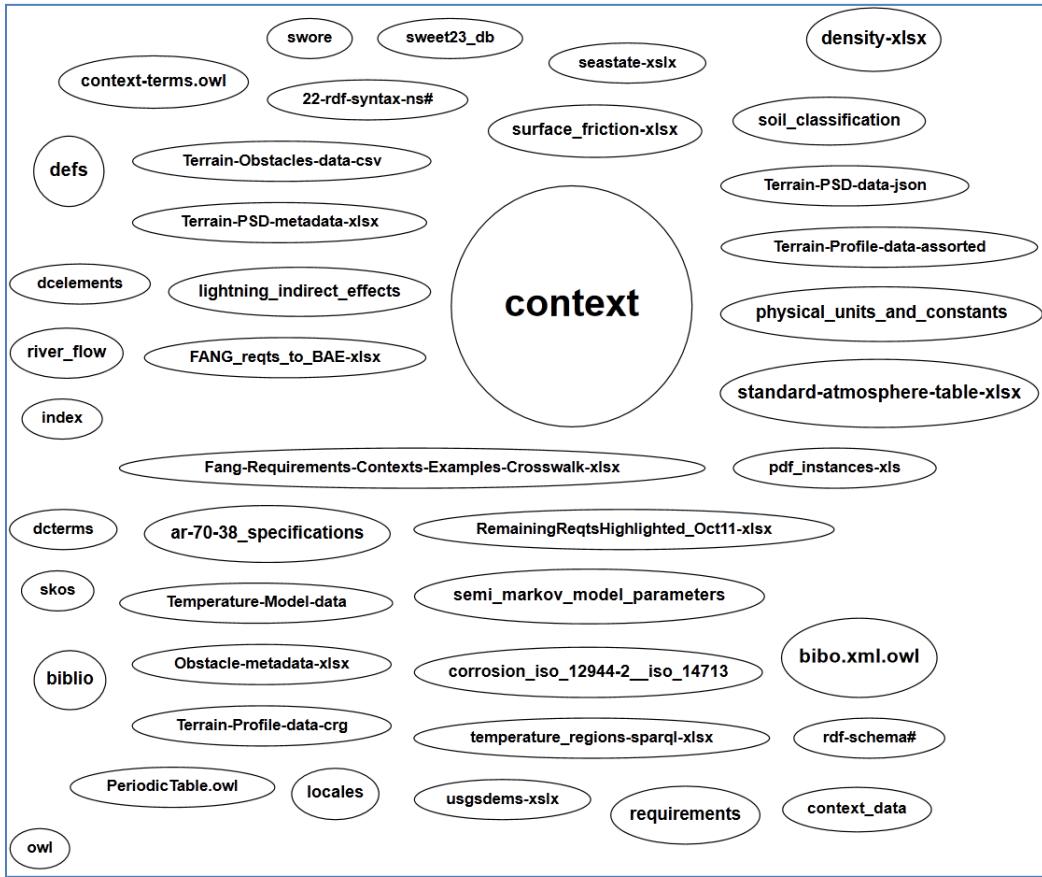


Figure 11: Cloud describing semantic data sets. The large circle labeled “context” contains the majority of the reference and citation data. Each ellipse corresponds to an RDF or Turtle triple-store data set

A context is defined as the surrounding environment for a specific intended use, such as may be applied to evaluating a vehicle design or its performance. The environment itself is wide-ranging so that individual contexts can be independently isolated and treated as domain features. This means that a domain feature such as fine-grained terrain can be isolated from the atmosphere domain or, more subtly, distinguished from the overall gross topography of a region. This has important ramifications in that we can encapsulate certain aspects of the environmental models without leaking abstractions across domain boundaries.

Further, as described in **Figure 1**, many of the environmental models are passive and *non-compliant* in terms of interactions with a subject under test. A *compliant* interface would be, for example, a terrain surface that demonstrates a significant give when interacting with a vehicle [24]. These kinds of compliant interfaces demand a more intimately coupled multi-physics interaction between the environment and subject, and is out of the scope of what a semantic context web server can offer in computational throughput and bandwidth. However, the set of passive interfaces remains quite significant and important in evaluating fitness-for-intended-use and other criteria for designs.

Example of Environmental Requirement

As an example of user requirements, consider the paces that a ground-based vehicle has to go through. A set of use-cases may involve the capability to traverse a cross-country course with a specific root-mean-square (RMS) deviation in the terrain roughness. If requirements phrases such as “cross-country” and

“RMS ride courses” are captured in a semantic knowledgebase, rules can then be added to automatically link to a specific environmental model. This is shown in **Figure 12** below.

Example

Terrain				from ontology graph labelled :	https://babelfish.arc.nasa.gov/confluence/display/AVNPROJ/BAE#					
Req Phrase	Context	Ref	Workflow	Requirement	SWEET	Graph	Attribute	Category	System	Level
level, hard surfaced road	0	doc	path	The vehicle shall attain a speed of no less than XX kph (XX mph) on a level, hard surfaced road	realmBioBiome Terrain	link	Land Speed (road)	Mobility - Land	Full Vehicle	Vehicle
level, hard surfaced road	Asphalt	doc	path	On a level, hard surfaced road , the vehicle, at GVW, shall attain a speed of no less than:	realmBioBiome Terrain	link	Level Road Land Speed	Automotive Performance	Full Vehicle	Vehicle
cross-country	0	doc	path	The vehicle shall be able to attain land speeds of XX kph (XX mph) for cross-country movement.	realmBioBiome Terrain	link	Land Speed (cross country)	Mobility - Land	Full Vehicle	Vehicle
cross-country	Asphalt	doc	path	For cross-country movement, with the vehicle at GVW, shall be able to attain land speeds of:	realmBioBiome Terrain	link	Cross Country Land Speed	Automotive Performance	Full Vehicle	Vehicle
Root Mean Square (RMS) ride courses	0	doc	path	The vehicle at GVW shall attain no more than XX watts average vertical absorbed power, as measured at the base of all occupant seats, while negotiating the following Root Mean Square (RMS) ride courses at speeds listed below XX mph at XX in XX mph at XX in XX mph at XX in	realmBioBiome Terrain	link	Absorbed Power	Mobility - Land	Full Vehicle	Vehicle
Root Mean Square (RMS) ride courses	Asphalt	doc	path	The vehicle at GVW & Combat weight shall attain no more than an average vertical absorbed power of 6 Watts, as measured at the base of all occupant seats, while negotiating the following Root Mean Square (RMS) ride courses at speeds listed:	realmBioBiome Terrain	link	6 Watts Absorbed Power Limit Over RMS Courses	Automotive Performance	Full Vehicle	Vehicle

Requirement phrase extracted from document and then linked to workflow path

Figure 12: Requirements to workflow

If these phrases are captured beforehand and stored as persistent knowledge with links to categories, models and workflows, we can take advantage of “factory-level” automation when a new project requirements document is introduced. Individual requirements stored as triples within a data store (see the ellipses labeled in “Fang-req...” and “RemainingReqs...” in **Figure 11**) can be parsed by the expert system with the phrases highlighted within individual requirements. A tool such as OpenRefine[25] can help with the translation from a spreadsheet to a triple-store format such as RDF[26], N3, or Turtle.

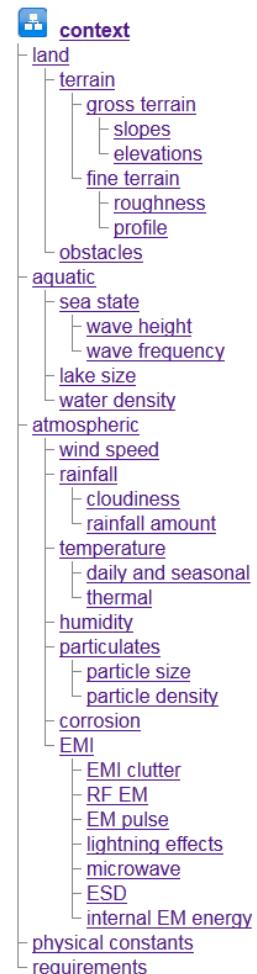
In terms of a semantic web interface, the user is presented with a listing of phrases discovered within a semantic context, and then tabulated against a path to the appropriate workflow.

An aspect to this formulation guards against potential requirements churn. As new requirements come in with extra decomposition levels, new triple-store linkages can be added without upsetting any previous triples that may have previously existed. In other words, a perfect relational schema does not have to be maintained, which is an issue when applying the output from typical requirements databases. Adaptability of knowledge format is the key here, while keeping a balance between free-form textual association and tacit semantic relationships.

Browsing Interface

The server provides a browsing interface that is structured around the natural hierarchy that we employ of *land*, *aquatic*, and *atmospheric* domains.

Internally, we employ triple-store relationships to manage the links in the hierarchy. The hierarchy (see right) is extended to provide space for resources



that may span features, see *physical constants*, and for those that may be user defined, *requirements*.

See the Annex A for a browser-guided narrative that is provided as part of the dynamic context server. This explains the rationale for the categorization chosen for the set of models.

Reference and Citation Linking

The context knowledgebase contains semantic information which allows it to be searched in a structured fashion. The set of references to research articles, specifications, and standards is managed by the [Zotero documentation citation system](#) [27][28][23]. Individual citations are tagged with terms from SWEET within the Zotero browser component (as a Firefox plug-in) to indicate the environmental concepts that apply.

For non-structured information retrieval, the search bar in the upper-right corner finds information within the knowledgebase that is not semantically organized.

A typical semantically-based query is shown in **Figure 13**, shown below for a specific SWEET term signifying the *Temperature* property class. Regular references are distinguished from specifications (grid icon) and model foundational documents (star icon), the latter of which contain detailed information to support the construction of models for the context library.

The screenshot shows a web-based interface for managing references and citations. At the top, there's a search bar with the query "propTemperature:Temperature". Below the search bar, there are two sections: "Query ontology for references and citations" and "propTemperature:Temperature : Semantic Graph".

- Query ontology for references and citations:** Contains two dropdown menus: "Search for a citation relevant to context modeling" (set to "Confluence") and "Search based on SWEET terminology" (set to "propTemperature :: Temperature"). There are "Submit Query" buttons for each.
- propTemperature:Temperature : Semantic Graph:** This section displays a table of results:

Name	Who	Action	Title
propTemperature:Temperature	US Army	spec	Research, Development, Test and Evaluation of Materiel for Extreme Climatic Conditions
propTemperature:Temperature	NOAA	ref	Temperature Records Wilmington
propTemperature:Temperature	Pukite Bankes	model	diffusive growth.docx
propTemperature:Temperature	Challou	spec	NAIO AFP 5
- context graph:** A diagram showing relationships between various SWEET concepts like Descriptive, PeriodicProperty, ThermodynamicProperty, Quantitative, and Temperature.
- apropos terms:** A table listing terms and their properties:

Term	Concept	Category
Temperature	Temperature property (observation)	
BoilingPoint	Temperature property (observation)	
MeltingPoint	Temperature property (observation)	
TriplePoint	Temperature property (observation)	
KineticTemperature	Temperature property (observation)	
StaticTemperature	Temperature property (observation)	
TotalTemperature	Temperature property (observation)	

Figure 13 : View of the reference and citation search interface.

Capturing and integrating reference and citation knowledge via a standard citation management system and a standard ontology such as SWEET allows for future growth, aided by the fact that other users can contribute knowledge to the reference database.

Workflows

Workflow patterns follow the concept of economized development. We rely on patterns of workflow to reduce modeling work, simplify maintenance, and create a record of a project's provenance.

The supplemental workflow used to access statically defined context models for test-bench execution is called OSCAR (*Ontological System for Context Artifacts and Resources*).

OSCAR is a knowledge-based system providing semantic web discovery capability, limited according to the available context library ontology and instance data (see Appendix E). The goal is to provide guided discovery for users to find, register, or edit context models and associated metadata to be used in their simulation. The context models include collections of power spectral density profiles (PSDs) and other

data sets, which provide the generators for model execution. This is considered an industrial strength component as it has a design lineage tracing to oceanography data content service architectures [29].

The other workflows available are intended to provide dynamic content for probabilistic evaluation, such as what is needed to perform PCC (probabilistic certificate of correctness)[1] and for support of formal verification. In these cases, probability density functions provide the constraints and likelihoods for testing extreme values or corner cases, with the distributions mathematically provided to allow Monte Carlo simulations to explore the required state space for test cases.

The workflows can also be used to quickly evaluate model characteristics and collect artifacts for later decision making. For example, the earlier example of requirements search is actually a guided workflow that starts from the captured requirements phrases.

A sampling of the workflows available is shown in Figure 14 below:

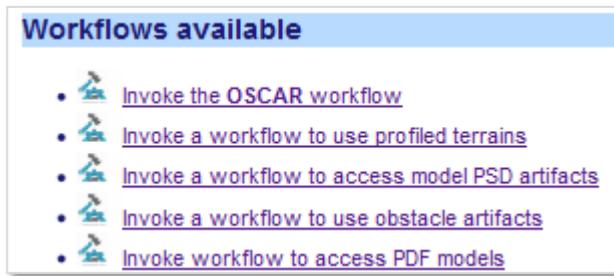


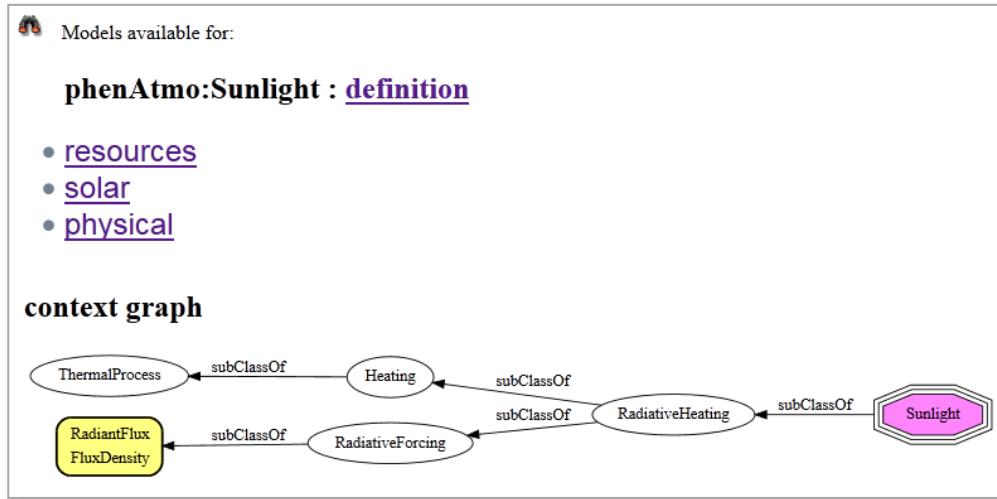
Figure 14 : Workflows available

Search

Many common environmental models share the same underlying stochastic pattern. In general, this means that we can use similar math to capture the characteristics of behavior across a range of domains, even though the measured quantities are not comparable.

The important case for PCC estimates is the probability density function (PDF) of environmental characteristics showing a spread in uncertainty in values.

PDFs and other models can be searched according to feature keywords or SWEET categories based on applicability to the context space as shown in **Figure 15**.

**Figure 15 :** Search results for a SWEET classification term

The general rule we apply for general content search is to base it on ontological terms (if not a free-form textual phrase) and then use that to restrict the resources that match.

The presentation of knowledge from the matching hits could be in HTML with links such as shown in **Figure 15** or in XML, JSON, CSV or other format suitable for external use.

The server also provides a more free-form textual search that evaluates contents of the triple-store, such as `rdfs:comment` instances, and then presents the results as a list of triples (see **Figure 16**):

Search results for token "case(gravel)"		
Subject	Predicate	Object
ent:ATC_Munson_Gravel_10_logspaced	ent:file_path	"/library/data/ATC_Munson_Gravel_10_logspaced.JSON "
ent:ATC_Munson_Gravel_90_logspaced	ent:file_path	"/library/data/ATC_Munson_Gravel_90_logspaced.JSON "
ent:ATC_Munson_Gravel_logspaced	ent:file_path	"/library/data/ATC_Munson_Gravel_logspaced.JSON "
ent:YTC_KOFA_Level_Gravel_logspaced	ent:file_path	"/library/data/YTC_KOFA_Level_Gravel_logspaced.JSON "
ent:YTC_Patton_Hilly_Gravel_logspaced	ent:file_path	"/library/data/YTC_Patton_Hilly_Gravel_logspaced.JSON "
ent:YTC_Patton_Level_Gravel_logspaced	ent:file_path	"/library/data/YTC_Patton_Level_Gravel_logspaced.JSON "
realmLandCoastal_Beach	rdfs:comment	<p>"A beach, or strand, is a geological landform consisting of loose rock particles - such as sand, gravel, shingle, pebbles, cobble - or even shell fragments, along the shoreline of a body of water. Beaches occur along coastal areas, where wave or current action deposits and reworks sediments, or at the margin of land along a lake or river subject to erosion caused by rainfall. Beaches are not necessarily found in conjunction with salt water, such as the ocean, in all instances. A seashore beach is merely one type of beach but it is the most commonly associated with the perception of the word beach."@en</p>

Figure 16: Free-form search partial results for the phrase “gravel”.

Resources

Models in the form of interactive standards are also provided via the context server. These are transcribed into semantic content from various standards documents. The semantic content enables the data to be used to automatically construct tables, graphs, or other structural forms (see **Figure 17**).

-  [Access OSCAR workflow](#)
-  [Access environmental resource artifacts](#)
-  [Access physical constants and units](#)
-  [Access standard solar artifacts](#)
-  [Access standard atmosphere resource artifacts](#)
-  [Access standard atmosphere table](#)
-  [Access periodic table of elements](#)
-  [Access materials properties](#)

Figure 17 : List of resources (tables, charts, and data) taken from environmental standards.

Also units of dimensionality and physical constants are categorized here. For various dynamic artifacts, the applicable units are stored as resources and necessary conversions between units are done automatically.

Examples of tables from triple-store include the Periodic Table of Elements (**Figure 20**), a Coefficient of Friction table (**Figure 19**), and a Soil classification table (**Figure 18**).

category	division	groupSymbol	groupName
coarse grained	gravel	GW	gravel containing well-graded (diversified particle sizes)
coarse grained	gravel	GP	gravel containing poorly graded (uniform particle sizes)
coarse grained	gravel	GC	gravel containing clay-like character
coarse grained	gravel	GM	gravel containing silt-like character
coarse grained	sand	SW	sand containing well-graded (diversified particle sizes)
coarse grained	sand	SP	sand containing poorly graded (uniform particle sizes)
coarse grained	sand	SC	sand containing clay-like character
coarse grained	sand	SM	sand containing silt-like character
highly organic	organic	OH	organic high plasticity mainly clay-like character
highly organic	organic	OL	organic low plasticity mainly silt-like character
fine grained	clay	CL	clay with low plasticity
fine grained	silt	ML	silt with low plasticity
fine grained	clay	CH	clay with high plasticity
fine grained	silt	MH	silt with high plasticity

Figure 18: Soil Classification table featuring USCS two-character code automatically

The following coefficient of friction table **Figure 19** was generated from a table of values captured from engineering data[30] and then converted to RDF triples through the *OpenRefine* tool[25].

surface material	description	dry slow friction	dry fast friction	wet slow friction	wet fast friction	constraint	note
PORLTAND CEMENT	New,Sharp	[.80,.120]	[.70,.100]	[.50,.80]	[.40,.75]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
PORLTAND CEMENT	Traveled	[.60,.80]	[.60,.75]	[.45,.70]	[.45,.65]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
PORLTAND CEMENT	Traffic Polished	[.55,.75]	[.50,.65]	[.45,.65]	[.45,.60]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ASPHALT or TAR	New,Sharp	[.80,.120]	[.65,.100]	[.50,.80]	[.45,.75]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ASPHALT or TAR	Traveled	[.60,.80]	[.55,.70]	[.45,.70]	[.40,.65]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ASPHALT or TAR	Traffic Polished	[.55,.75]	[.45,.65]	[.45,.65]	[.40,.60]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ASPHALT or TAR	Excess Tar	[.50,.60]	[.35,.60]	[.30,.60]	[.25,.55]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
GRAVEL	Packed,Oiled	[.55,.85]	[.50,.80]	[.40,.10]	[.40,.60]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
GRAVEL	Loose	[.40,.70]	[.40,.70]	[.45,.75]	[.45,.75]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
CNDERS	Packed	[.50,.70]	[.50,.70]	[.65,.75]	[.65,.75]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ROCK	Crushed	[.55,.75]	[.55,.75]	[.55,.75]	[.55,.75]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
ICE	Smooth	[.10,.25]	[.07,.20]	[.05,.10]	[.05,.10]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
SNOW	Packed	[.30,.55]	[.35,.55]	[.30,.60]	[.30,.60]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast
SNOW	Loose	[.10,.25]	[.10,.20]	[.30,.60]	[.30,.60]	Not for heavy large trucks	Speeds of 30 MPH separate slow from fast

Figure 19: Coefficient of Friction table automatically constructed from semantic elements

Perhaps the most sophisticated semantic-based generation example is the periodic table of elements shown below, which uses supplemental logic rules for element placement and coloring.

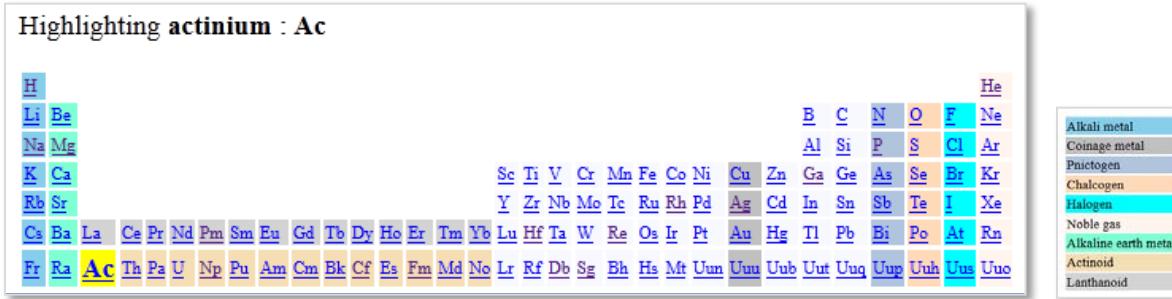


Figure 20 : Periodic Table of the Elements generated from semantic rules

Examples of generating graphs from triple-store include rendering of ideal gas curve and nominal temperature specifications for various climactic conditions (see Figure 21). These are taken from the environmental modeling handbook AR-38 [31].

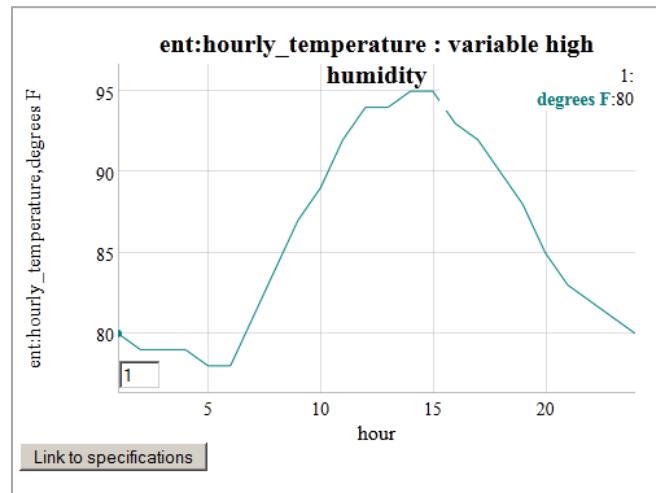


Figure 21 : Generating a plot from an environmental specification of nominal daily humidity swings.

Map/Location

Environmental models can either be general in terms of their suitability (such as water properties or a standard atmosphere), or specific to a geospatial location, which can further include daily or seasonal effects.

The locations are accessible through the semantic drop-down list or through indexed search terms with keystroke look-ahead.



Figure 22: Keystroke look-ahead for domain features linked to geo-spatial locations.

The map button indicates the location of the model or data, and the available button links to the modeling domain(s) available. A Google Map API window is also provided to illustrate the geospatial location and coordinates of the model or data. Both these features are shown in **Figure 23**.

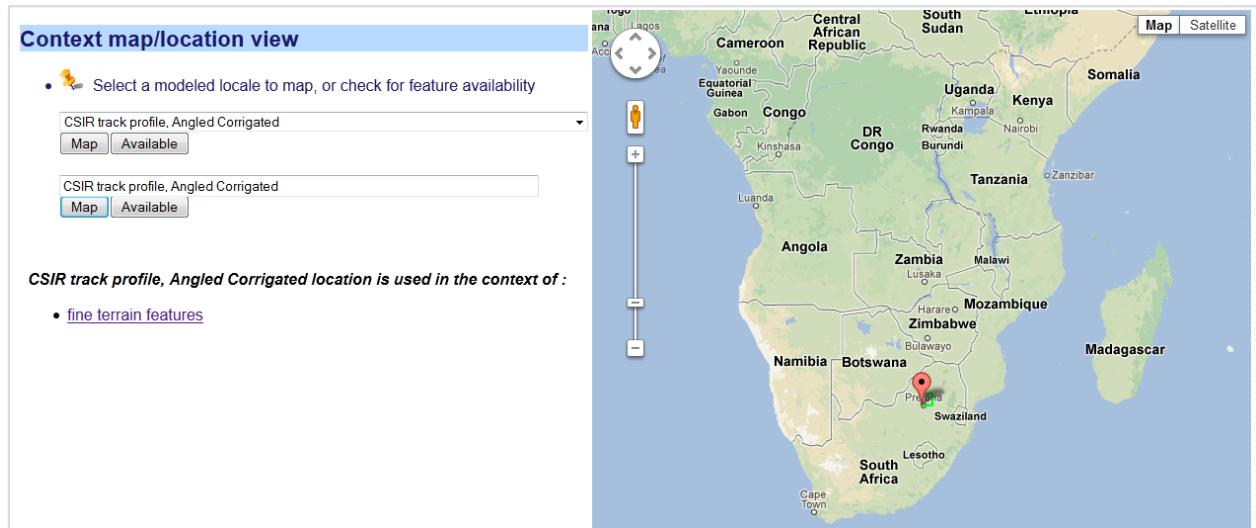


Figure 23: Google Map window for locating geospatial location of queried model, with link to model on left.

Knowledgebase and Server Technical Architecture

The internal structure of the dynamic context server consists of a semantic reasoning engine and its accompanying set of knowledge. The interactivity with this underlying architecture via a web server allows for a user interface and potentially automated modes (see **Figure 24**) to remote data sources or to test benches.

The ontology is cast as a form of *description logic* [1], which consists of a terminology set (the so-called terminology box or *TBox*) and an asserted data store (the assertion box or *ABox*). The terminology and assertions can come from standard ontologies such as SWEET (for the *TBox*) and from qualified data and information sources (for the *ABox*). In other words, the former provide latter provides the classification and the latter provides knowledge and data.

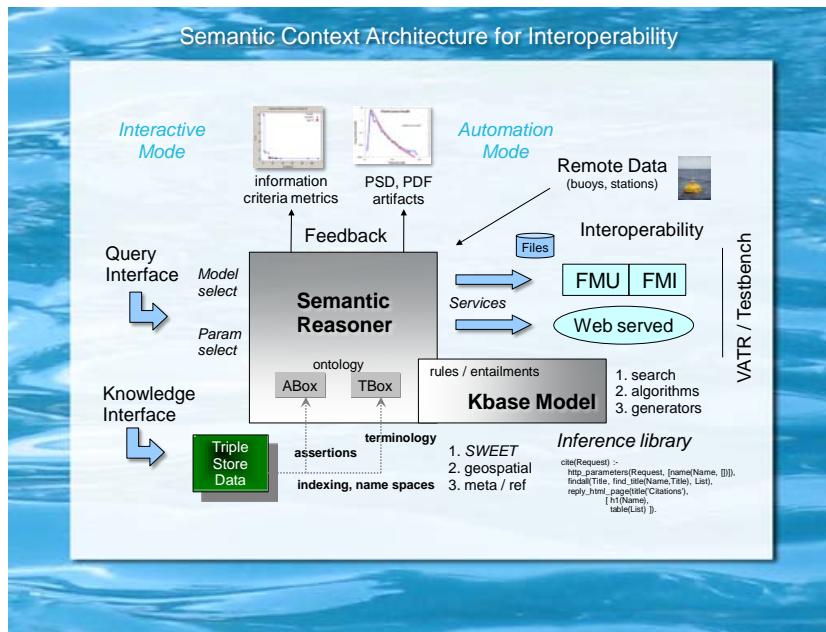


Figure 24: Usage architecture for the semantic web server. The Dynamic Context Server handles artifacts and interactive services, while OSCAR handles static data and executables such as functional mockup units (FMU)

The logical rules work with the terminology and assertions to accomplish searches, execute algorithms, generate code, and otherwise perform general inferencing.

To interface concisely with the triple-store database, a set of *namespace prefixes* is specified and used by the reasoner. A subset of the prefix namespace is shown below in **Figure 25**. As the SWEET ontology provides a comprehensive view of natural terminology, this by necessity has the largest set of prefixes, largely to provide disambiguation for terminology that is borrowed across domains. For example, a *wave* could refer to an aquatic wave or an electromagnetic wave, with the namespace helping to provide a context.

The namespace called *ent*: holds special significance as it provides sub-classing and additional terminology to the SWEET ontology. By providing an additional level of abstraction, we can guard against changes made to the reused standard ontologies.

Known RDF prefixes (namespaces)

The following prefixes are known and may be used without declaration in SPARQL queries to this server.

Prefix	URI
bib	http://purl.org/net/biblio#
bibo	http://purl.org/ontology/bibo/
cite	http://www.zotero.org/namespaces/export#
cntxt	http://on.cs.unibas.ch/owl/1.0/Context.owl#
context	https://babelfish.arc.nasa.gov/confluence/display/AVMPROJ/BAE#
cpack	http://clopatria.swi-prolog.org/schema/cpack#
dbpedia	http://dbpedia.org/resource/
dc	http://purl.org/dc/elements/1.1/
dcterms	http://purl.org/dc/terms/
ent	http://entropplet.com/terms#
eor	http://dublincore.org/2000/03/13/eor#
foaf	http://xmlns.com/foaf/0.1/
graphviz	http://www.graphviz.org/
human	http://sweet.jpl.nasa.gov/2.3/human.owl#

Figure 25 : Configured prefixes for URI name-spaces.

As an example of the sub-classing, the resource **ent:C4** refers to a specific classification of corrosive environment (see **Figure 26**). According to the semantic hierarchy, it sub-classes from both the SWEET **procChemical:Corrosion** class as well as the **procCategorical:StandardIndustrialClassification** class from SWEET. The directed graph also shows how these SWEET classes are sub-classed within that hierarchy. The C4 property is extended to include specific information relevant to this corrosion classification, such as qualitative descriptions of **ent:environmental_conditions** and **ent:exterior_examples**.

Local view for "http://entropplet.com/terms#C4"

Predicate	Value (sorted; default)
rdfs:subClassOf	procChemical:Corrosion ^{2,3} procCategorical:StandardIndustrialClassification ^{2,3}
ent:carbon_steel_mass_loss	"[400,650]" ¹
ent:carbon_steel_thickness_loss	"[50, 80]" ¹
ent:corrosion_scale	"C4" ¹
ent:environmental_conditions	"High" ¹
ent:exterior_examples	"Industrial and coastal areas with moderate salinity." ¹
ent:interior_examples	"Chemical plants. Swimming pools. Coastal shipyards, boatyards and seaside docks." ¹
ent:standard_applies	ent:corrosion ¹
ent:unit	"micron/yr" ¹
ent:zinc_mass_loss	"[15,30]" ¹
ent:zinc_thickness_loss	"[2,1,4,2]" ¹

Named graphs describing this resource:

¹file:///c:/users/pukitepa/git/eval/context/dynamic_context_server/instances/corrosion_iso_12944-2_iso_14713.ttl

²file:///c:/users/pukitepa/git/eval/context/dynamic_context_server/terms/defs.ttl

³<http://entropplet.com/terms>

The resource appears as object in 2 triples

Context graph

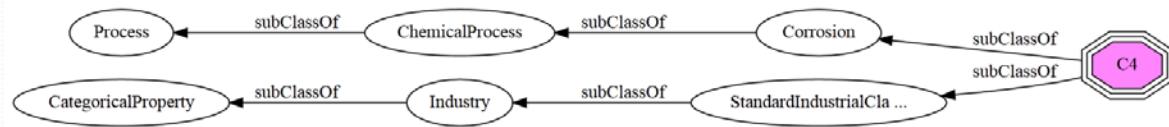


Figure 26: Local view of the **ent:C4** resource represented as an octagon-shaped node

To understand how this resource can be used, refer to the corrosion classification table in **Figure 27** shown below. This is constructed based on information contained in ISO standards [32][33] and then converted to triple-store. The semantic rules then enable the generation of the table.

class	env	exterior	interior	carbon steel	zinc	units
C1	Very low	Inland rural	Interior of buildings, offices, stores, schools, hotels. Indoor spaces with occasional condensation.	[1,3]	[0,0.1]	micron/yr
C2	Low	Very ligh industrial zone or in a city without pollution	Warehouses, sports auditoriums, logistic centers, hangars. Dry indoor spaces.	[3,25]	[0.1,0.7]	micron/yr
C3	Medium	Inland urban, mildly saline. Urban and Industrial atmospheres, moderate SO ₂ pollution, coastal areas with low salinity.	Indoor spaces with high moisture content, not much impurities. Production rooms with high humidity and some air pollution, e.g. food processing plants, breweries, laundries.	[25 ,50]	[0.7,2.1]	micron/yr
C4	High	Industrial and coastal areas with moderate salinity.	Chemical plants. Swimming pools. Coastal shipyards, boatyards and seaside docks.	[50 , 80]	[2.1,4.2]	micron/yr
C5-I	Very high (heavy industrial zone)	Very humid industrial atmosphere. Heavy industrial zone, contact with chemicals .	Industrial plants, warehouses, heavy equipment workshops	[80,200]	[4.2,8.4]	micron/yr
C5-M	Very high (marine atmosphere)	Saline seaside atmosphere. Coastal and off-shore areas with high salinity, up to 1000 m from the coastline	Buildings or areas with almost permanent condensation and high pollution.	[80,200]	[4.2,8.4]	micron/yr

Figure 27: Corrosive classification table based on ISO 12944-2 and ISO 14713 constructed from semantic elements

The reasoner itself uses rules and logic that are embedded in run-time loaded modules. A partial list of logic modules is shown in **Figure 28**.

Server plugin configuration		
Config	Title	Status
010-packs		Local
020-prefixes	Configure prefixes (namespaces)	Installed (modified)
100-used_modules		Local
cache	Configure caching of RDF inputs	Not installed
cloud		Installed (linked)
con_text	HTML utilities	Local
config		Installed (linked)
context	Top-level context module	Local
context_acronyms		Local
context_ar7038	Environmental specifications according to standard	Local
context_atm	Atmosphere specifications	Local
context_autocorr	Topography interface	Local
context_browse	Tree browser for context models	Local
context_clutter	Model of EMI clutter	Local
context_codegen	Code Generation utilities	Local
context_complex	Math operations for complex numbers	Local
context_corrosion	Models of corrosion	Local

Figure 28: The web server Prolog run-time automatically loads all available modules specified by a configuration. These are considered the set of server plug-ins.

The cloud of **Figure 11** uses a manifest of triple-store data-sets. The manifest itself is an RDF list in Turtle format which describes the name and location of each of the triple sets loaded by the server. So the general strategy is to load the logic and data at run-time startup².

² The data-sets are cached and optimized for faster loading on subsequent start-ups.

The consistency of the knowledgebase is automatically tested on load and reload by a unit-test suite, which is equipped to execute whenever module-based changes are made to the knowledgebase.

The final deployment is targeted for cloud computing, and the server can be accessed at initially an Amazon EC2 server located at <http://entroplet.com>. Statistics of usage are accessed from an administration level, with typical output shown in **Figure 29**.

Static workers and statistics:					
Port: 3020 Started: Wed Jan 16 16:49:56 2013 Total CPU usage: 22.98 seconds Heap memory: 0 bytes Requests processed: 1.7K Bytes sent: 2.9M # worker threads: 5					
Statistics by worker					
Thread	CPU		Local	Global	Trail
httpd@3020_1 0.718		In use	904	23.1K	0
		Limit	256M	256M	256M
httpd@3020_2 0.764		In use	6.7K	215K	30.7K
		Limit	256M	256M	256M
httpd@3020_3 0.889		In use	904	23.1K	0
		Limit	256M	256M	256M
httpd@3020_4 1.903		In use	904	23.1K	0
		Limit	256M	256M	256M
httpd@3020_5 0.827		In use	904	23.1K	0
		Limit	256M	256M	256M

Figure 29: Administrative statistical view of server usage.

Examples of usage

The following are examples of usage for the dynamic context server. These are selected to show a cross-section of capabilities available.

Sea State View

A sea-state model is used to provide a standard quantification for the significant wave height expected or measured for various coastal and inland bodies of water. A sea-state rating follows a scaling system starting at 0 for calm water. The objective and subjective description for a specific sea-state depends on the characteristics of the environment as defined by the World Meteorological Organization specification[34] (see **Figure 30**):

value	low	high	units	description
0	0	0	m	Calm (glassy)
1	0	0.1	m	Calm (rippled)
2	0.1	0.5	m	Smooth (wavelets)
3	0.5	1.25	m	Slight
4	1.25	2.5	m	Moderate
5	2.5	4	m	Rough
6	4	6	m	Very rough
7	6	9	m	High
8	9	14	m	Very high
9	14	9999	m	Phenomenal

Figure 30: Semantically generated chart of sea-stat rating system

The wave elevation cumulative distribution function (CDF) maps to the sea-state for a given region, such that lower sea-state values are more common for smaller and shallower bodies of water. The mapping from sea-states to probabilities over a long term is shown in **Figure 31**.

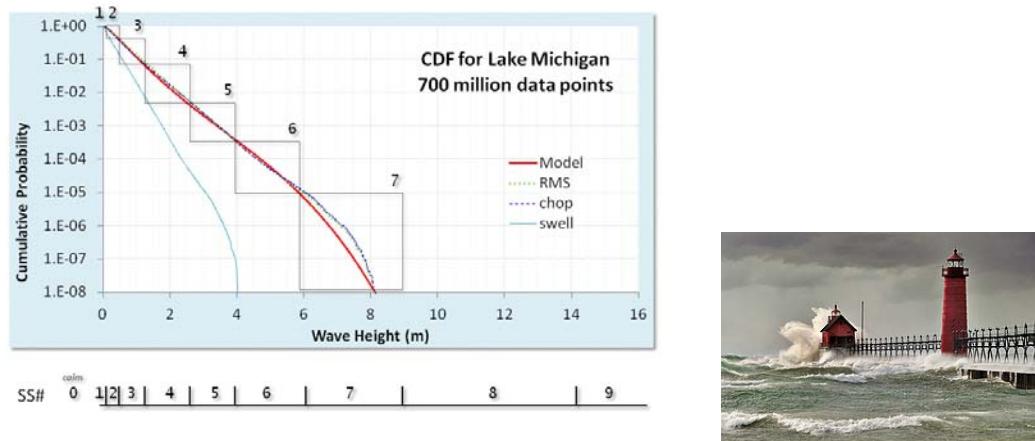


Figure 31: Sea-state rating mapped to a CDF culled from several years of Lake Michigan significant wave height data. Sea-state classification depends on probability distribution of wave-height according to a specific geo-spatial location. Sea-states of higher than index 7 have not been observed for this body of water.

To make the sea-state table practical for probability estimates in an interactive setting, we needed to attach the sea-state alignment to a particular region (see **Figure 32**).

Seastate as Significant Wave Height

Likelihood of specific sea state depends on region

1 2 3 4 5 6 7 8 9 <= sea-state
 meters

Atlantic coast
 Lake Superior
 Lake Michigan

Figure 32: Sea-state selection window

If we select a sea-state and the Atlantic coast region with logarithmic scaling, the following cumulative distribution function (CDF) chart results (see **Figure 33**). The reasoner automatically intersects the low and high end of the significant wave height for the sea-state enumeration selected.

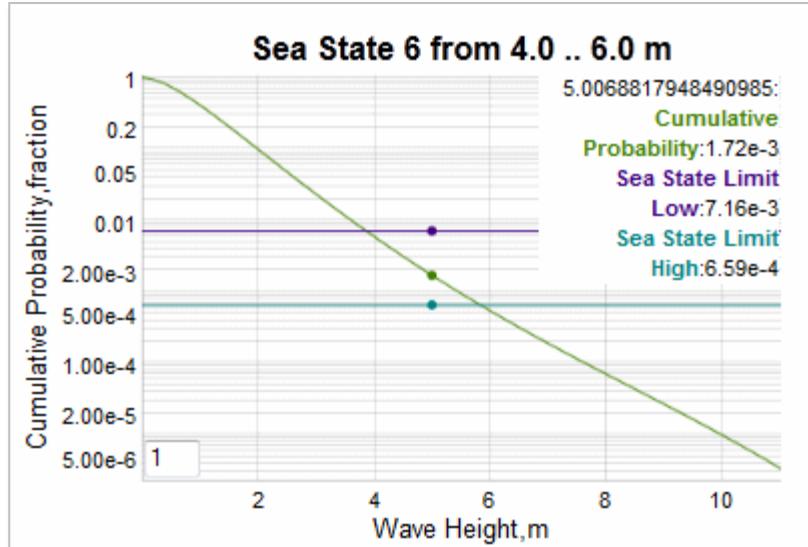


Figure 33: Selected interactive sea-state 6 classification for Atlantic coast

In the example chart above, the cumulative probabilities for a sea-state of index 6 are highlighted at the position of the cursor. The lower cumulative is given by the purple line and the upper (rarer) cumulative given by the high-end sea-state value is the bluish-green line.

The ranges depend on a model fit to the historical data culled from the selected aquatic region[2], so to add another region requires only an additional region and parameterized model to the triple-store database.

Obstacles

Not all terrain features have natural origins. We catalogued man-made terrain profiles as *obstacles* and captured and stored their geometric shape as deterministic segments. Most of the obstacle profiles were culled from specifications derived from vehicle test courses[20]. Since these test courses were created specifically for testing vehicles with a parallel wheel-base, we stored the data as double track profiles with a fixed separation between the two tracks.



Figure 34: Jersey Barrier and two-track deterministic profile shown beside it.

The range of obstacles includes jersey barriers (**Figure 34**), washboard surfaces (**Figure 35**), cross-tie courses (**Figure 36**), half-round bumps, staircases, potholes, and ditches. To get a rough indication of the spatial orientation of a track, a feature for rendering the double tracks as a 3D image is provided. This extrapolates the detailed profile specification as an extruded three-dimensional projection (see **Figure 35**).

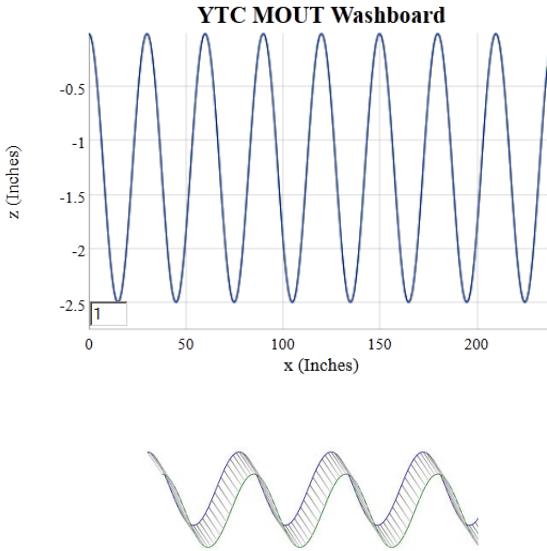


Figure 35 : Washboard track. Detailed view (above) and 3D view (below).

A workflow was incorporated to guide the choice of obstacle profile data sets (**Figure 36**). In this case a cross-tie course is rendered in a schematic projection, along with the extrapolated 3D view. The green trace is the inside track and blue trace is the outside/far track of what could represent remnants of an old railroad track.

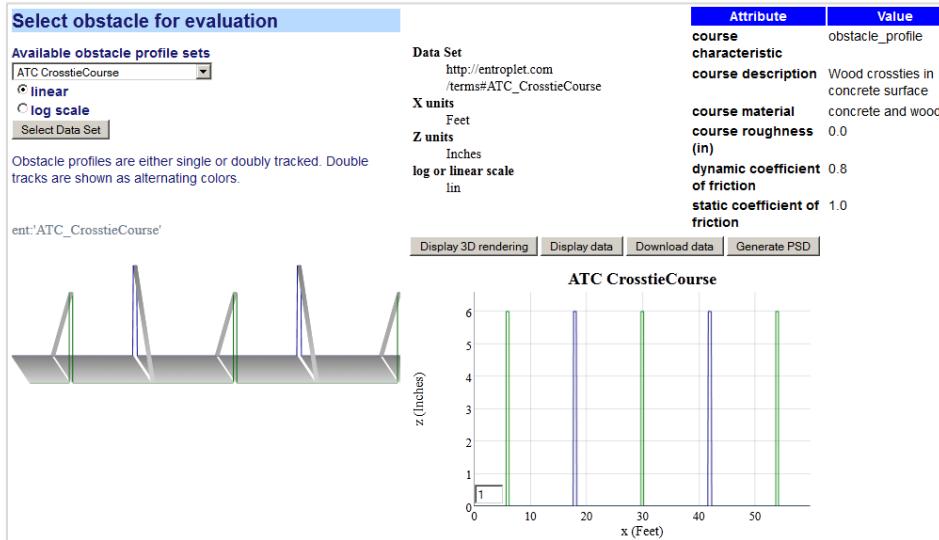


Figure 36: Obstacle evaluation workflow showing selection of a cross-tie track.

The deterministic nature of these tracks is evident when one considers that artificial washboards (or periodic cross-ties) as shown above have a defined periodicity while natural washboards created by vehicular traffic (or decomposing and shifting cross-ties) show a disordered range of periodicities. The latter are more suited to descriptions by a semi-Markov model, which is described next.

Fine-relief Terrain View

We illustrate two complementary approaches to generating terrain profiles based on power spectral densities (PSD).

1. Superposition of randomly phased sine waves to accomplish a pseudo inverse Fourier transform of the PSD. This is essentially a Fourier series reconstruction
2. Fitting a semi-Markov autocorrelation model of random walk to the PSD (via the Weiner-Kinchin theorem [35])

Each approach has benefits and drawbacks

Benefits	<ul style="list-style-type: none"> ▪ The superposition approach is fast and automatic as it works as a rough heuristic ▪ The semi-Markov autocorrelation function approach is based on stochastic properties of the terrain relief changes so can model properties such as phase and skew[3]
Drawbacks	<ul style="list-style-type: none"> ▪ The superposition approach will show a repeat sequence based on the lowest spatial frequency and needs to approximate phase (no asymmetries possible) ▪ The semi-Markov autocorrelation function approach requires a fitting process, and gets the most benefit from prior knowledge in addition to that provided by the PSD. However, once the models are created, new models can easily be composed from the old ones.

The terrain profile context models include both variants.

Markov and semi-Markov Processes

The fine-relief profile of natural terrain features can be characterized and modeled effectively by Markov and semi-Markov processes.

Mathematical Markov formulations such as the Ornstein-Uhlenbeck process[36] can model rough random terrain, while semi-Markov processes capture pseudo-random features with emerging periodicities[3]. We provide a workflow to guide the user in evaluating these kinds of models as shown in **Figure 37**.

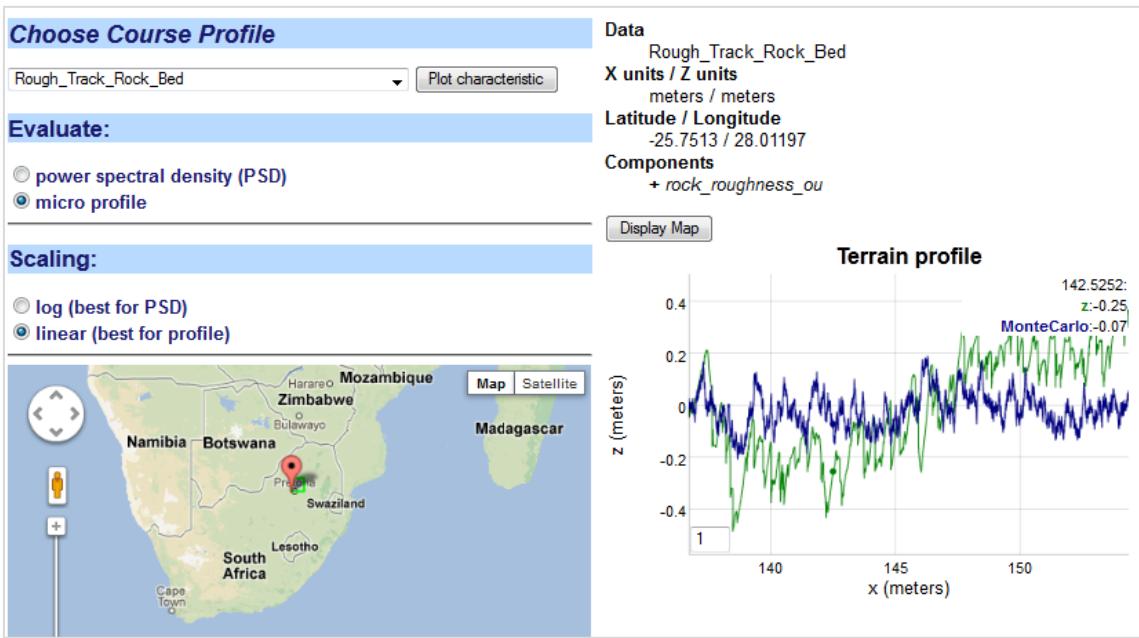


Figure 37: Micro profile simulation of a rough track (labeled z) overlaid with a Markovian Ornstein-Uhlenbeck random walk Monte Carlo simulation

To generate a profile, a Monte Carlo simulation draws a sequence of steps from sampling from the Markov or semi-Markov algorithm. This is compared to a profile from a high-resolution terrain taken from the Gerotek vehicle testing course, in this case a rough track rock bed.

For the PSD, the plot consists of two statistically equivalent representations starting from the Markov formulation, and the FFT of the data it is derived from (see **Figure 38**):

- An analytical expression for the PSD derived stochastic representation
- A windowed FFT of the Monte Carlo profile drawn from the stochastic representation.
- The FFT spectrum of the data.

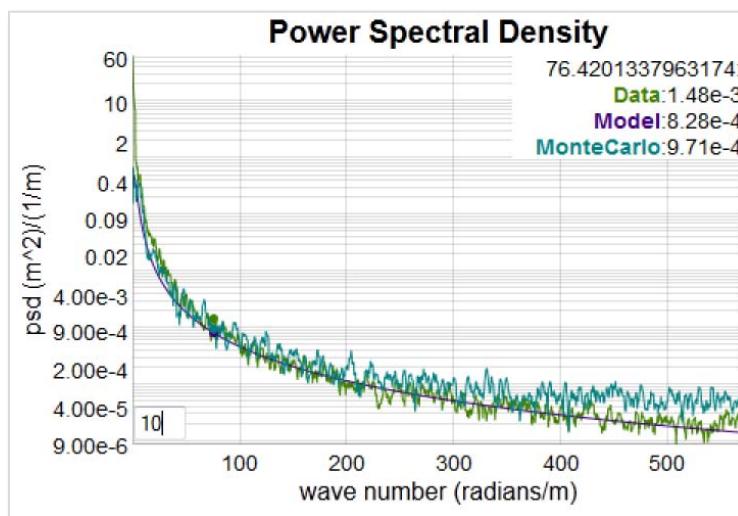


Figure 38: PSD of rough road

High-resolution data from CSIR is stored in the database (see the sets labeled GEROTEK and Rough Track) as well as a few other test sets taken from online archives[37].

The FFT can show substantial noise, so we use a moving average filter to reduce the noisy values so that the underlying spectrum is clearly delineated. The graph can be stretched and then filtered to reveal the signal (see **Figure 39**).

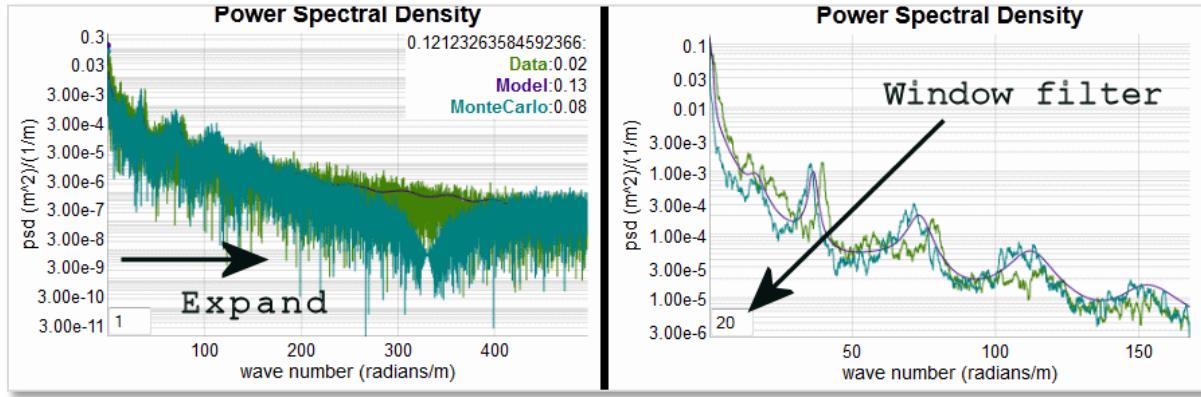


Figure 39: Use of interactive filter and drag selection to narrow in on a portion of the frequency spectrum. When the filter is used the underlying fine structure used for fitting is revealed.

The model contains parameters that are stored as a few triples, while the original data is also a triple but one that can contain several thousand data points. The general fitting strategy is shown in **Figure 40** and described in detail elsewhere[3].

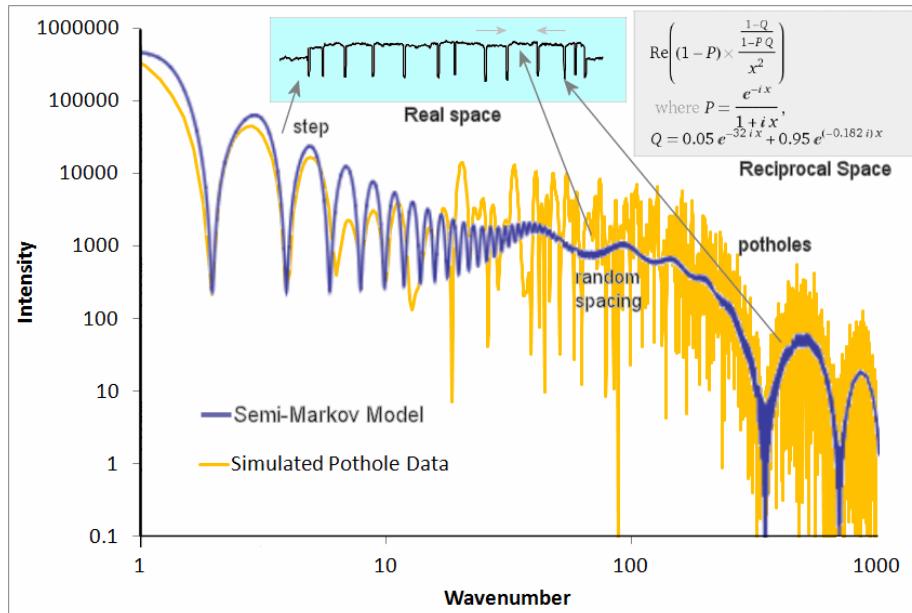


Figure 40 : The typical fitting procedure uses knowledge of the underlying terrain profile (top middle inset, showing a pot-hole course with steps) and a stochastic representation of the PSD (upper right inset) to match the PSD calculated from the data.

We essentially use the PSD to estimate the strength of the periodicity in the underlying terrain features. The sharper and stronger the harmonic peaks are, the stronger the periodicity.

For example, **Figure 41** shows the terrain and model for a corrugated course profile which shows strong periodic features.



Figure 41: Workflow for Gerotek corrugated course. The terrain profile is indexed to a Google Map of that area and we can zoom in and pick up the scale of the highlighted corrugated features.

The following models (**Figure 42**) show a range in periodicities ranging from very strong (corrugated course), to easily detectable but disordered (Belgian block cobblestone course), to weak (random spaced obstacles).

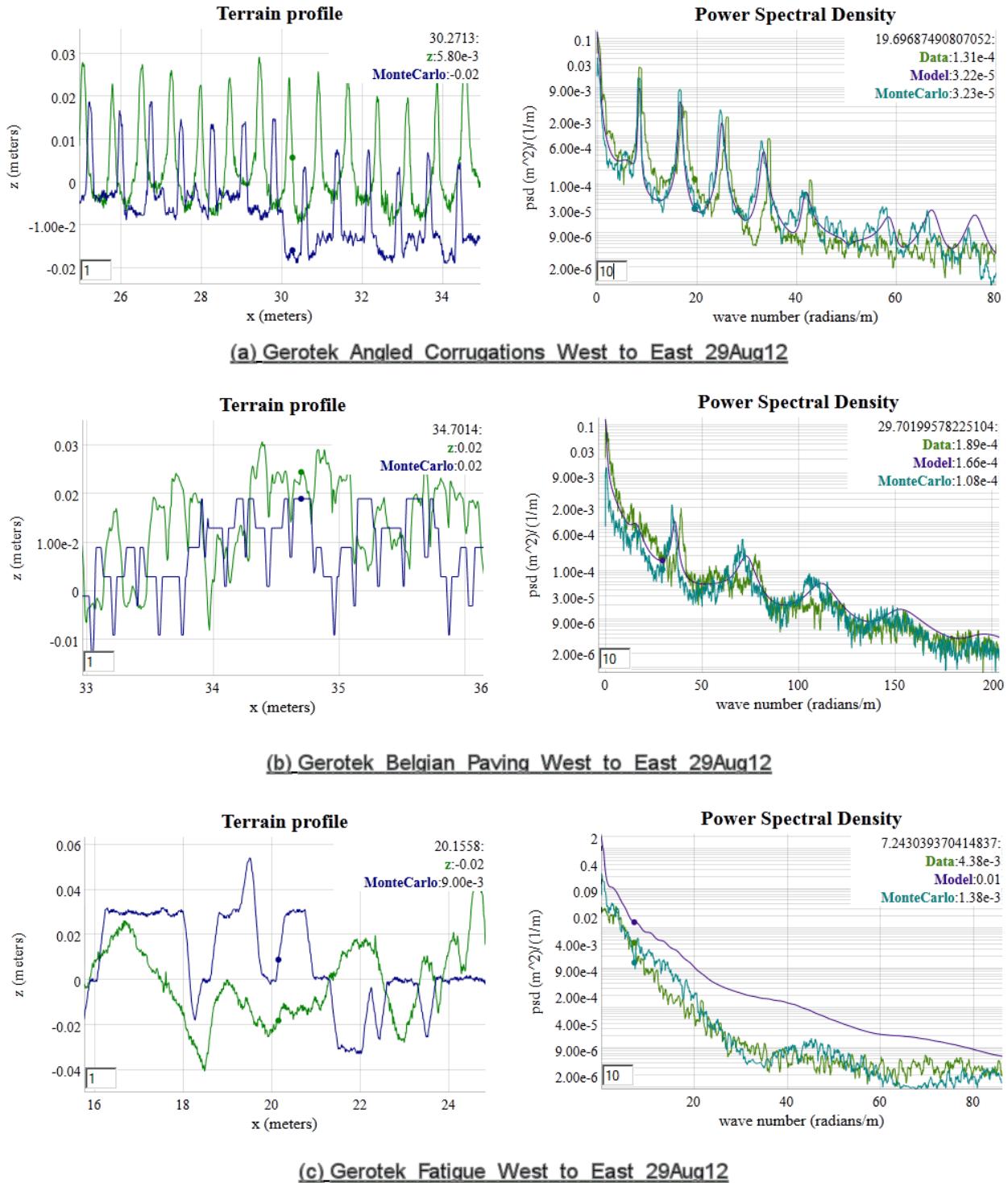


Figure 42: Semi-Markov model fits for high-resolution Gerotek courses. For each terrain, a profile of the actual terrain is shown as the bright green, and the model as the blur. Course (a) is a fairly regular layout of square bumps. The missing harmonic at 50 radians/m is due to the width of the bump. Course (b) is a worn layout of Belgian block cobblestones, with periodicity emerging from the block spacing according to the PSD. Course (c) is a course with random obstacles and very weak periodicity.

Superposition of Sine approach

This is essentially a Fourier series reconstruction approach based on having only empirical PSD data available.

For each course, identifying characteristics are provided along with a PSD plot of the course profile. The plot is usually attached to an otherwise classified course layout (see **Figure 43**).

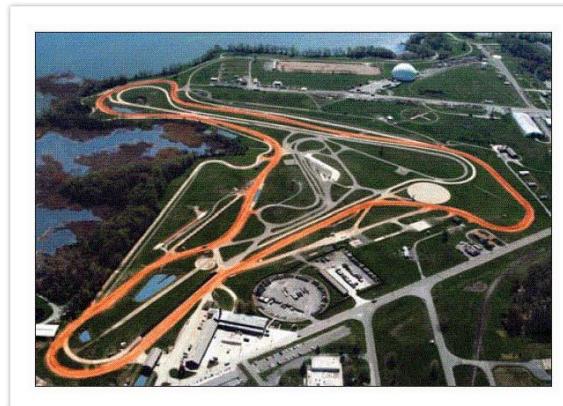


Figure 43: TOPS test course

The figure below shows the workflow for selecting from PSD course data (see **Figure 44**). This indexes against an ontologically categorized set of terrain PSD data cataloged from Test Operating Procedures (TOPS)[20]. The data set contains header information necessary to calibrate the scales.

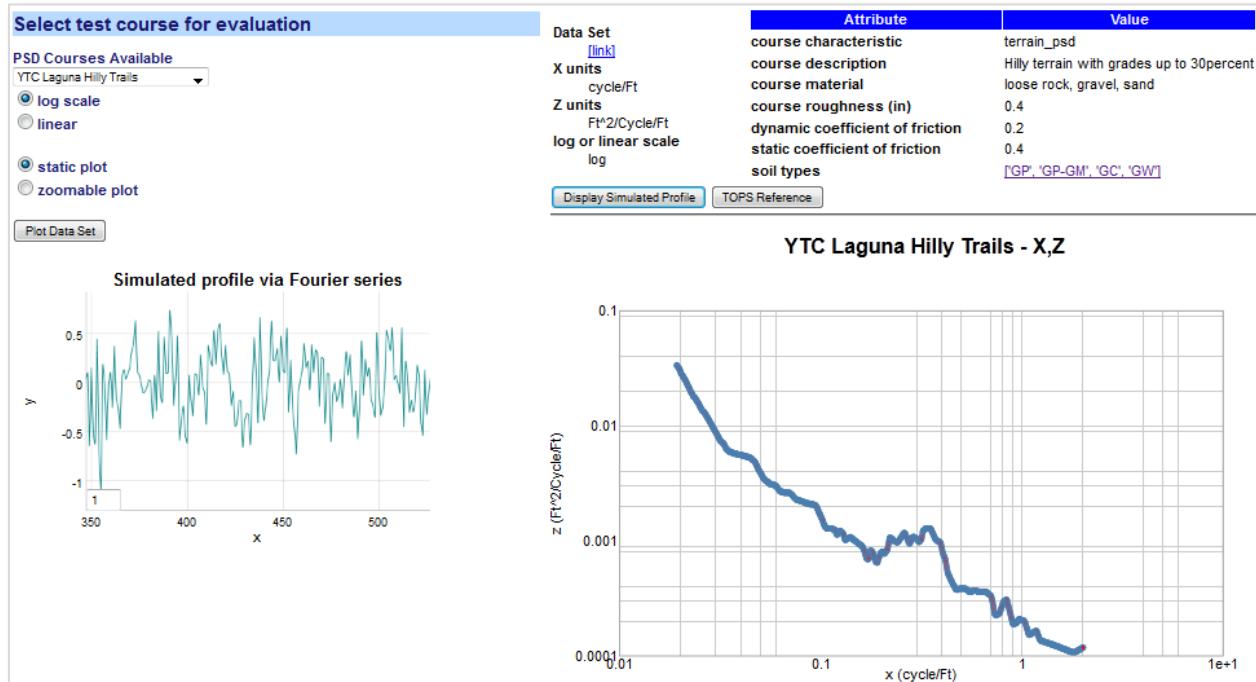


Figure 44: PSD workflow example. The PSD data set is selected from a drop-down list. The PSD is plotted on the lower right and a Fourier series reconstruction is displayed on the lower left.
If soil types are available, they are linked to a soil classification table.

Gross-Relief Topographic View

Terrain Elevation models marginal probabilities shown on a log scale indicating likelihood of an elevation change given a surface translation.

Terrain is characterized for the lower 48 states according to digital elevation model (DEM) [38][39][40] profiles. The surface translation is limited to 40 post-spacing's, which corresponds to a range of less than 4000 meters and the elevation change is limited to 20 meters.

Fits for each 1 degree section are generated via a Gauss-Markov model following an Ornstein-Uhlenbeck (O-U) reversion-to-the-mean random walk process. Maximum entropy (MaxEnt) estimation generates uncertainty in the O-U model, which in general gives a better fit to the terrain data sets. (see further Appendix F for an application). For a selected point on the map (**Figure 45**) or the drop-down list of a named section (**Figure 46**), the enclosing 1° DEM quadrangle is analyzed for elevation correlations.



Figure 45: Map selection mode pick a 1 degree quadrangle, shown by the green outline on the right.

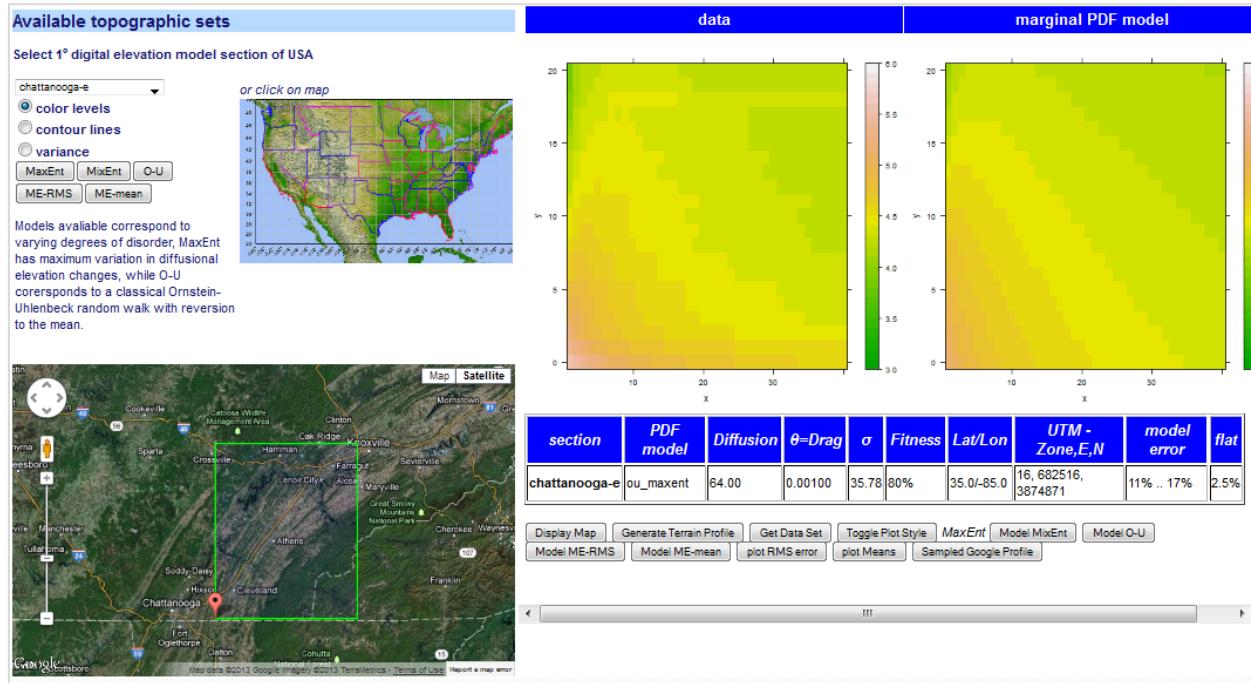


Figure 46: Elevation correlations are shown as color contour maps. The model fit is shown in the far right, which should match in levels to the data contour next to it. This is a correlation for the Chattanooga east DEM, shown in the lower left.

As the color contours may be difficult to delineate, a conventional line contour display is available as shown in **Figure 47**. Both types of contours were rendered via calls to the **R** statistics package.

If the model fit for a specific region matches the contour profile with relatively small error, then the model can be used to simulate a typical terrain elevation profile for that section. So the model essentially takes the place of actual contours from the region; for example **Figure 48** (a) is the model and **Figure 48** (b) is a randomly located sample from a Google Map query in that same section.

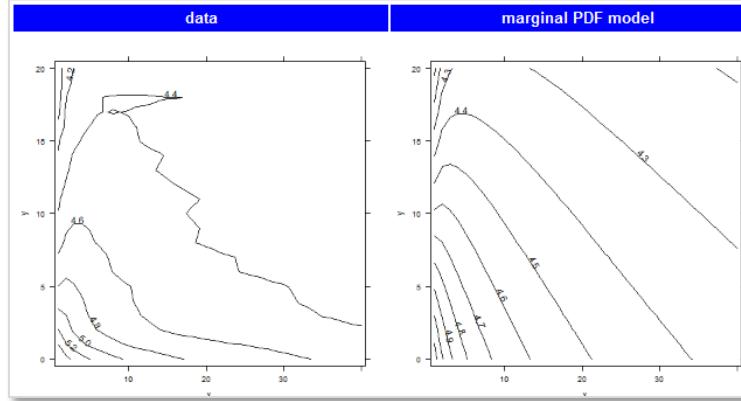


Figure 47: Contour plot variation

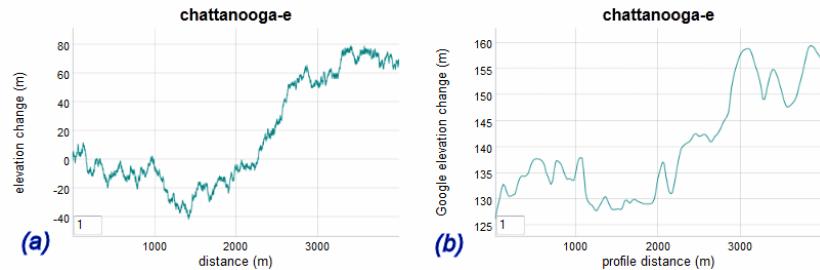


Figure 48: The fit to the DEM section via an Ornstein-Uhlenbeck model is translated to a random walk profile of terrain in (a). A sample profile extracted from a Google Map API query is shown in (b).

Temperature

The characterization of temperature is obviously acutely dependent on geospatial location of the simulation, test, or use case. Although environmental standards handbooks provide nominal characteristics for cold and hot climates [31] (which we cover elsewhere in the ontology), the need often arises for providing temperature models for specific locales.

Seasonal Model

We leveraged linked semantic data from other open sources to demonstrate what is possible with the context server. The site dbpedia.org [41] provides average monthly climate data as part of their instance database, and other sites such as Wikipedia already rely on it. For this example, we demonstrate how it is integrated with the dynamic context server.

The server essentially takes two steps in its workflow reasoning. It first requests all available locations with temperature from **dbpedia**, and then once the user selects an available location, the monthly high and low averages are plotted (see **Figure 49**).

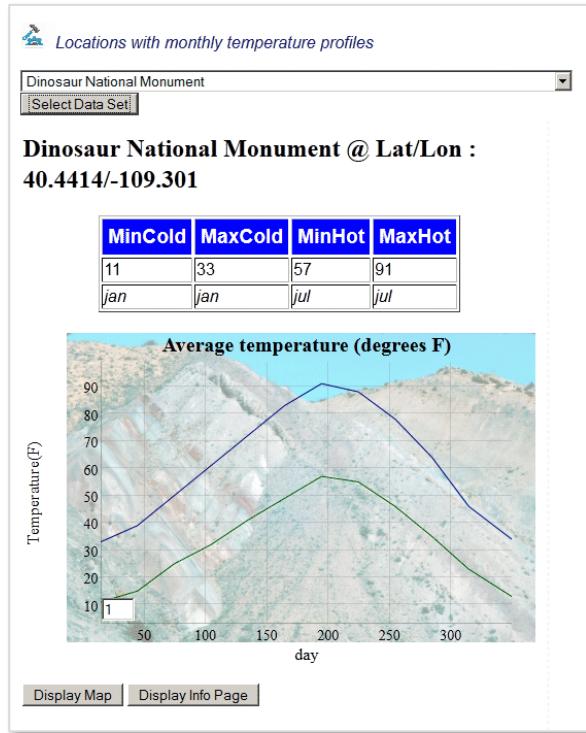


Figure 49: Average monthly high and low temperature extremes captured from dbpedia. The background image is also supplied as a linked data element to the geospatial location.

Other atmospheric measures are also available depending on whether a specific location (and dbpedia) has the data recorded. In **Figure 50** below, we show average precipitation for a semi-arid location in Oregon.

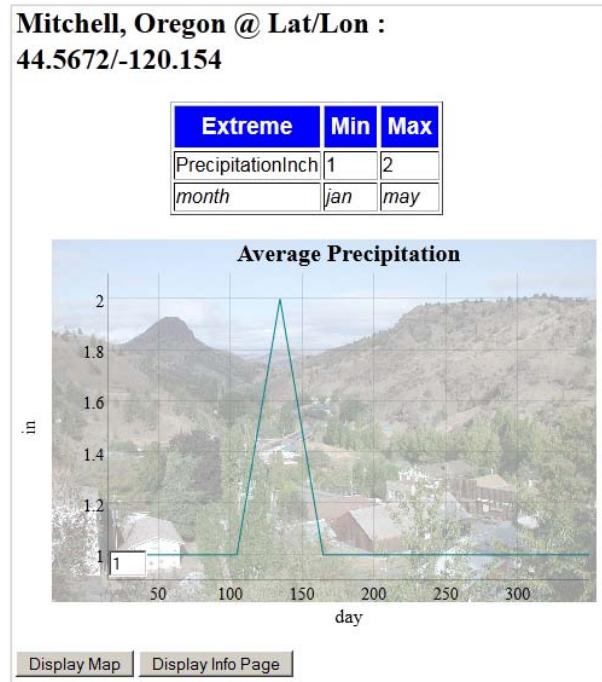


Figure 50: Average monthly precipitation of eastern Oregon acquired from dbpedia

Diurnal precision model

As part of the AVM project, we learned that context model users requested more temporally-detailed temperature records than were available from **dbpedia**. In particular, diurnal records were requested for Camp Lejeune in North Carolina and also for Aberdeen Proving Ground near Baltimore, Md.

For North Carolina, the closest sites we could find in NOAA national weather service records were the Wilmington and Morehead City sites. These are equally close to Camp Lejeune according to the NWS records (<http://www.erh.noaa.gov/>). This situation points to the hit-and-miss nature of raw non-semantically indexed records. The historical data at the Wilmington location was in a PDF file, so we had to extract the temperature data manually. The Morehead City page had a slot but no data. Baltimore had a text file which required custom parsing. This is referred to as “unique local data”, and the culling of this data for semantically-aware context usage is part of the knowledge capture process, and which shows why the structured approach of **dbpedia** is so effective.

After extracting the data from the National Weather Service records for these two locations, expected value models were generated to give the “normal” temperature that a weather forecaster would provide for any calendar date and time-of-day (no statistical moments were generated). **Figure 51** shows the diurnal model over a calendar year. The higher resolution mode of the dynamic graph functions by dragging the cursor across a calendar range and then the daily periodic nature becomes obvious.

Some of the non-uniformity in reporting is being addressed by NOAA, and they have recently deployed the U.S. Climate Reference Network to streamline the reporting: <http://www.ncdc.noaa.gov/crn/>



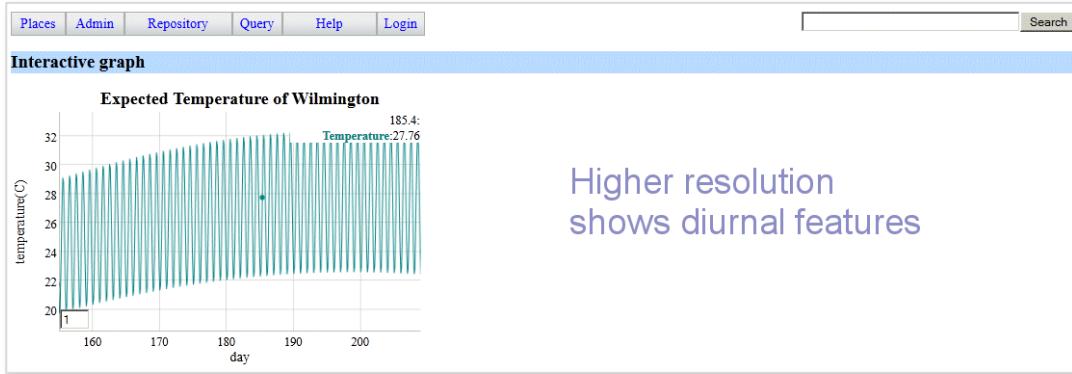


Figure 51: Diurnal temperature model for Wilmington, NC.

As the functional form of the expected diurnal temperature profile is saved in symbolic format, the semantic reasoner can convert that to a source code output. Below is a C-code snippet that uses the triple-store parameterization for a specific location.

```
#include <math.h>

float Get_Temperature (float time) {
    return (16.0+11.5*sin(2*3.141592653589793/365*time+ -1.85)+  

    (1.0*sin(2*3.141592653589793/365*time+4.44)+ (5.0-
    1.0))*sin(2*3.141592653589793*time+ -2.4));
}
```

The input (time) is in days since the first of the year, so noon on January 1st would be input as 0.5. Noon on the last day of the year is 364.5.

The semantic architecture can potentially apply geospatial reasoning[42] to pull in appropriate data. Indeed, geospatial functionality has found widespread use for inferring locality information. For example, say that temperature statistics are not known for a particular area, but data from nearby locations is available. A geospatial engine can aggregate and select data from weather stations in close proximity and use that to interpolate the statistics.

Thermal Diffusion

Models of thermal transients follow the boundary conditions set by temperature constraints[4]. As temperature always seeks equilibrium, the boundary conditions will be compliant if large thermal masses added to the environment. This means that the thermal transient will likely asymptotically approach some intermediate value of temperature factored by the ratios of the thermal masses involved.

The model used here applies a thermal diffusion process with a range in uncertainty of the coefficient of diffusivity or conductance.

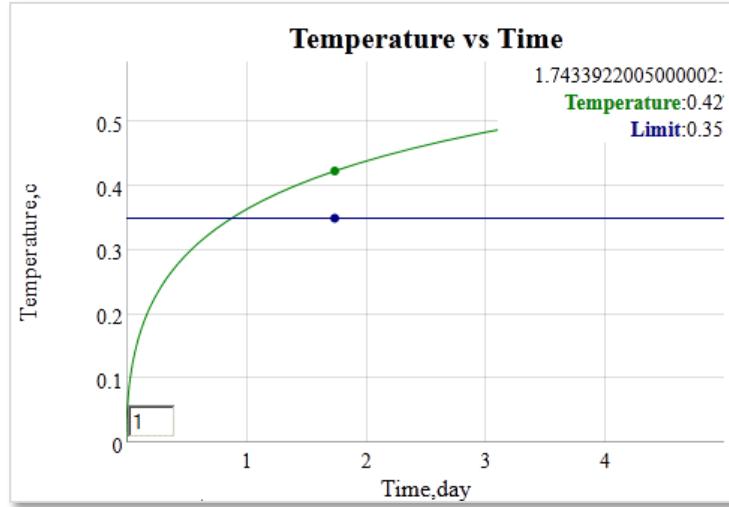


Figure 52: Thermal dispersion model assuming uncertainty in the coefficient of thermal conductivity, and a unit step thermal stimulus.

Stream Fording

Empirical information on stream flow suitable for evaluation of vehicle fording is subject to availability of discharge rates near dam sites. **Figure 53** shows a possible trajectory for an amphibious vehicle to cross a stream. Enough data is available that modeling of stream flow is possible[43].

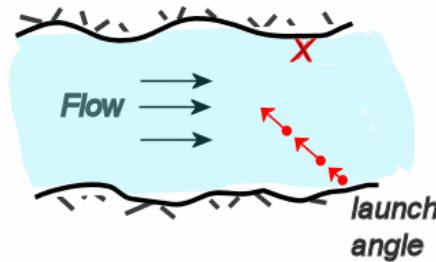


Figure 53 : Planning trajectory for fording a stream. To reach a designated point (X) on the far side of a stream, a launch angle and velocity is selected to compensate for the stream flow rate.

The stream flow is related to discharge rate via the cross-section of the river at the location of the measurement. Some sites have this available and will translate it directly to current or stream flow rate, yet this is not the rule. A typical PDF for stream flow is shown in **Figure 54**.

The other factor affecting stream flow is the seasonal variation caused by local weather patterns, precipitation, and upstream run-off or flood control.

The approach taken here is to provide a pattern which can be extended to allow for parametric analysis for specific locations. The range of variability in currents and discharge rates is great enough that a predictive generic model is not possible for the current statistical information.

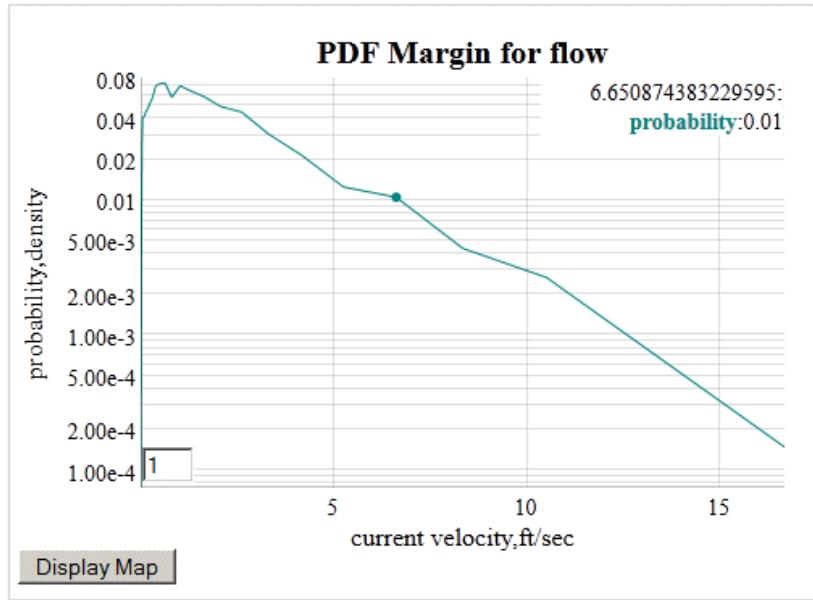


Figure 54: PDF for marginal flow of the Allegheny river. To convert from discharge rate to current velocity, a river cross-section (width \times depth) was assumed.

Clutter Integration

A clutter example demonstrates the time it takes to integrate an electronic signal buried in environmental EMI noise [2].

In this case, a GPS is initiated from a cold-start and the probability that the GPS is locked on a position after a specified time is shown in **Figure 55**.

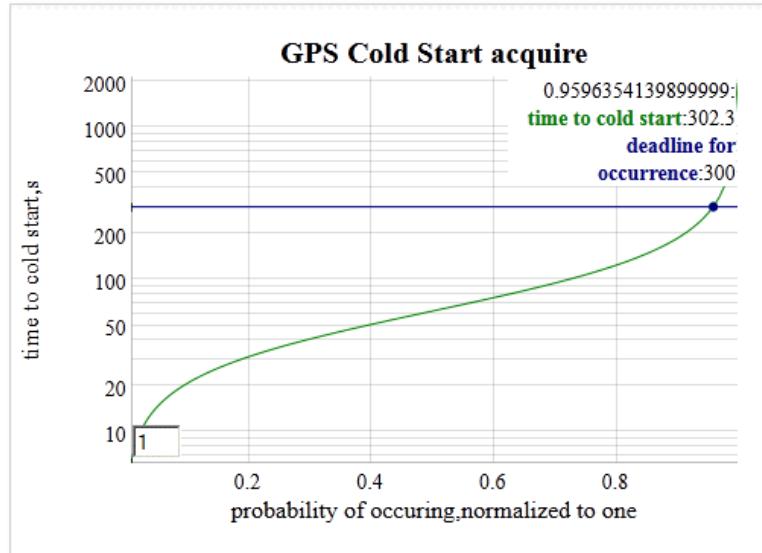


Figure 55: Time it takes to lock into a GPS signal from a “cold-start”. This integrates the statistical EMI noise until a certain level is reached. The result at the deadline is a probability of lock-in.

Corrosion Example

Corrosion of metals is a slow process, accelerated by adverse environmental factors such as high levels of salt and acidity.

	Marine(severe)	Industrial	Marine(mild)	Urban	Rural
D	90,000	45,000	12,000	6,800	5,300
X ₀	600	400	100	50	500
a	1.00E-08	1.00E-08	1.00E-08	0.4	0.2

$$X(t) = \sqrt{\frac{D \cdot (1 - e^{-at})}{a}} \cdot \left[\frac{\sqrt{\frac{D \cdot (1 - e^{-at})}{a X_0}}}{1 + \sqrt{\frac{D \cdot (1 - e^{-at})}{a X_0}}} \right]$$

We apply a diffusional model of corrosion which emulates the temporal profile of the empirical observations while providing an intuitive understanding to the gradual processes involved.[4]

The empirical parameters were taken from a recent study which considered long term corrosive effects in variously categorized environments. **Figure 56**

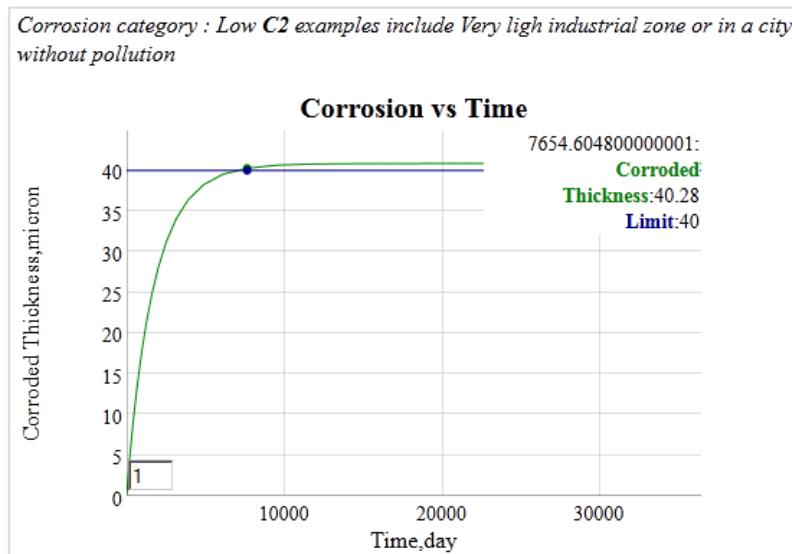


Figure 56: Example of a corrosive growth model parameterized from a low pollution site.

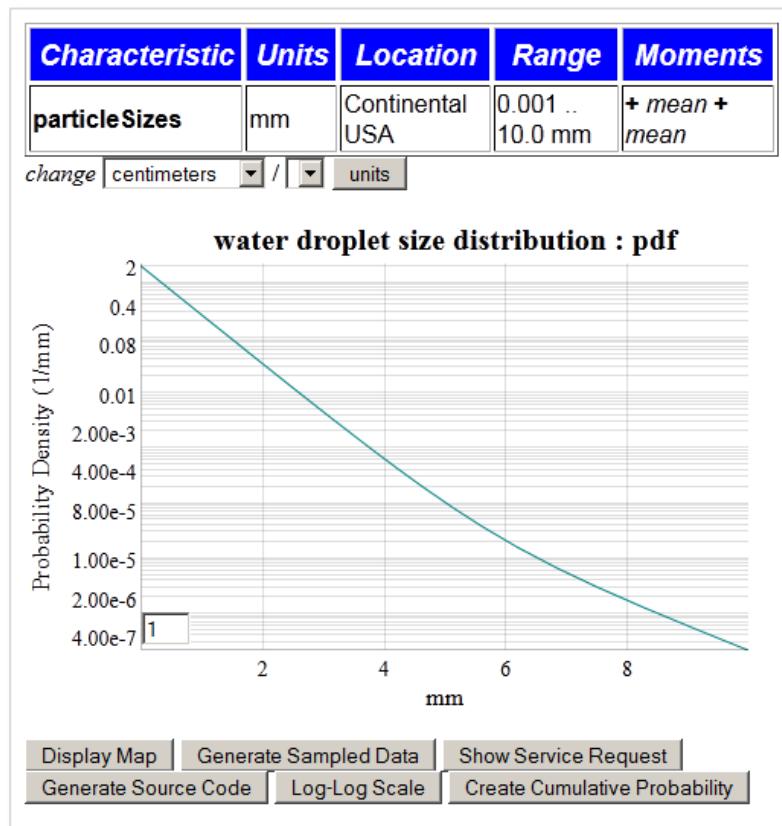
Droplet Size

For particulates, we consider a standard specification of rainfall droplet size from an environmental standard[31].

Table 2: From AR 70-38 standard population density against raindrop size

Drop Diameter Range (mm)						
density	0.5 - 1.4	1.5 - 2.4	2.5 - 3.4	3.5 - 4.4	4.5 - 5.4	5.5 - 6.4
# / m ³	2626	342	45	6	1	<1

The histogram in **Table 2** is an interpolated fit of the drop diameter data to a PDF based on two parameters, a mean droplet size that follows an exponential decline, mixed with a much lower exponential decline probability of a droplet with a higher size. The mix generates the slight knee in the curve shown in **Figure 57**.

**Figure 57:** PDF of water droplet size distribution

This class of PDFs is indexed via the pattern described in Annex 2: PDF Models.

Wind

We captured the wind velocity distributions from several regions of the world as a set of parameterized PDFs. The characterized distributions were fit to maximum entropy models as described elsewhere[2].

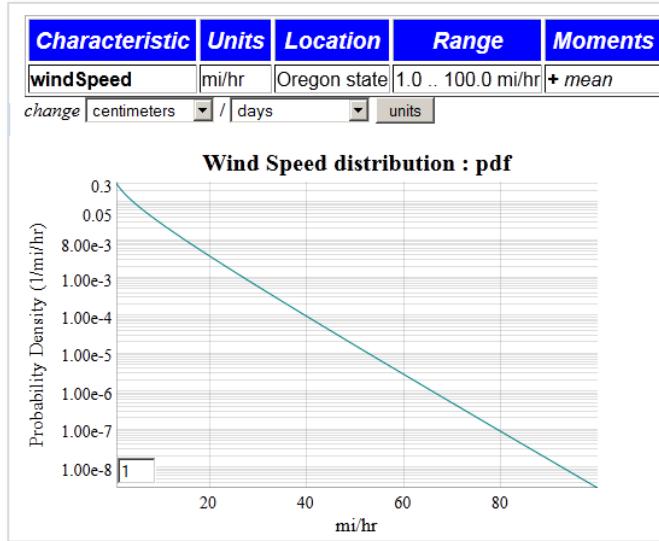


Figure 58: Wind speed probability density function for data from the Bonneville Power Authority in Oregon.

Figure 58 shows the result of choosing a certain region (Oregon's BPA data set) and then plotting the PDF from a BesselK wind speed distribution. The unit dimensionality rule was attached to allow the user to change the wind speed variate to any ratio of distance/time drawn from the triple-store unit definitions.

A sampling query is also available for drawing values from the PDF of the selected model. The BesselK requires two independent samples from an exponential damped distribution[2].

Pressure

A set of standard atmospheric profiles is encoded to allow a user to quickly estimate measures such as pressure at different altitudes. A pre-selector for altitude units and altitude specified is used as input to generate the graph shown in **Figure 59**.

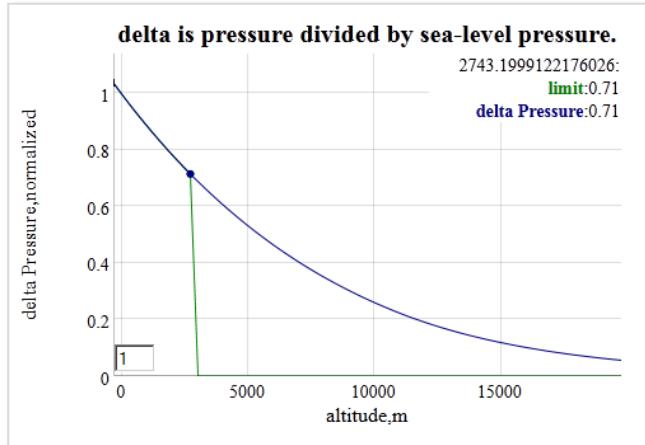


Figure 59: Pressure for standard atmosphere as a function of altitude.

Buoyancy Example

For aquatic environments we demonstrate a buoyancy model that compares density of immersed volumes against water density. We select a mass of an object and the volume that it displaces. To keep it obvious, we select an object that has about the same density of freshwater, 1 g/cm^3 or 1000 kg/m^3 . A rough prototype of the object density selection is shown below in **Figure 60**³.

Figure 60: Prototype for buoyancy input

In the figures below, one can see that the object is critically buoyant near freezing temperature (**Figure 61**) but at higher temperatures (**Figure 62**), it becomes denser than water and therefore will sink.

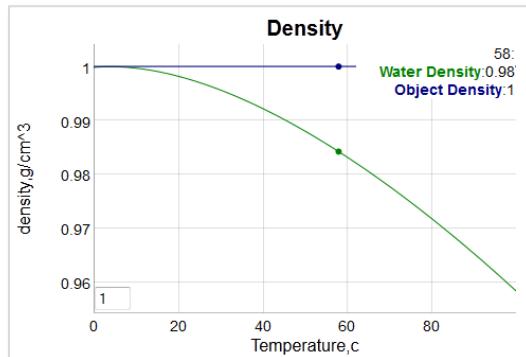


Figure 61: Critical buoyancy calculation for fresh water. If the green water density line is above the blue object density line, the object is buoyant and will float. (note that units in g/cm^3).

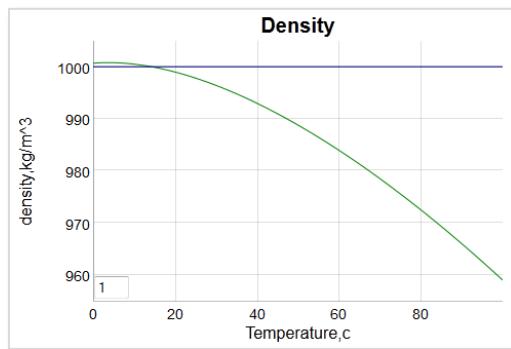


Figure 62: For salt water, note the slightly higher density, so object is more buoyant. (alternative units in kg/m^3).

³ As the buoyancy example is limited, a link to a more detailed spreadsheet model is provided in the example page. This includes instructions for creating a ProE model for the immersed object, which can calculate immersed density and center of gravity for more complex structures. Here we concentrate only on the context.

Patterns of Usage

All the examples were created from rules operating on semantic triple stores. Many of the examples operated on a standard pattern; for example, the wind speed and rain droplet size used the same PDF generator (see Annex X for a complete list).

For general external use, triple stores that are exposed by the dynamic context server can be accessed via a query language such as SPARQL. For example, the SPARQL expression

```
select * where{?s relaSci:hasNumericValue ?o}
```

can be transformed into a web service query by prefixing a “sparql/?query=” to the SPARQL expression. Then the following service call

```
http://localhost:3020/sparql/?query=select * where{?s relaSci:hasNumericValue ?o}
```

will return an XML list of all the triples that contain the predicate :relaSci:hasNumericValue.

So the examples of usage described in this section can get integrated with other semantically-aware languages that can either create web requests as URL expressions or as SPARQL queries. Having some facility with how to read triple-store graphs and construct URLs, a savvy developer can reuse the web services and/or browse through triple-store graphs

```
http://localhost:3020/browse/list_graphs
```

Development Process

The development of the semantic web software was facilitated by several collaboration tools. As this was part of the AVM effort, we took advantage of the collaboration tools supplied to the project teams shown in **Figure 63**. The Confluence Wiki provided the anchor for the semantic web development, as it provided a scratchpad for architectural ideas and for sharing knowledge and data for the context model library.

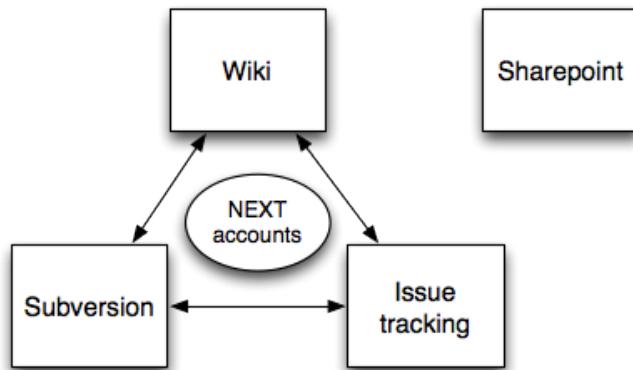


Figure 63: Team collaboration tools used during the development process.

Summary

The Dynamic Context Server provides models and artifacts as part of a comprehensive environmental modeling library. To maintain the library with the necessary level of organization and categorization, we used the Semantic Web for Earth and Environmental Terminology (SWEET) ontology as a semantic and terminology base. The system was built as an interactive web server that consisted of the DCS with an adjunct model delivery service called Ontological System for Context Artifacts and Resources (OSCAR). OSCAR's capabilities do not overlap with the DCS capabilities and in fact the two complement each other, with the two views integrated through a conventional web-service front-end.

The physical domain models represent the land, atmosphere, and aquatic realms. These have deterministic and stochastic representations; as an example from the terrain realm, the stochastic terrains characterized by PSD's and Markov models, and the obstacles as deterministic fixed geometries.

As architected, DCS and OSCAR perform as portals for serving context models and knowledge which can meet the needs of vehicle design and test.

Annexes

Annex 1: Browser Narrative

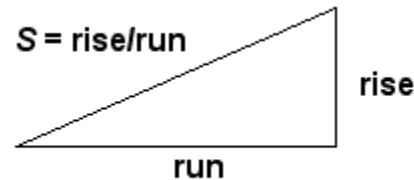
The server supplies an online narrative that describes the library hierarchy of major environmental classifications, **Land, Aquatic, and Atmospheric**. It is reproduced below.

The **Land** classification consists of all terrestrial models, specifically those attached to solid earth. This includes both natural terrain and man-made obstacles, and anything that deals with the terrestrial surface. Natural terrain is largely stochastic, with elements of determinism mixed in via certain man-assisted or physical processes.

Terrain is defined by the vertical and horizontal dimension of land surface. Characteristics of the terrain is usually expressed in terms of the elevation, slope, and orientation of terrain features

The **gross terrain** is also referred to as topography or land relief. The deterministic view of terrain is typically represented by a scaled elevation contour plot of a specific region of land. The stochastic view is represented by models of the topography, which is often cast in terms of a random walk process.

The **slope** of the terrain is characterized by the empirically measured rise/run of the local surface area. It is also known as the grade or pitch. By orientating perpendicular to the maximum slope, a pitch characterization becomes a roll characterization. In general, natural terrains show greater variation in slopes than do man-made features, such as roads or rail-road tracks, which will show signs of grading and switchbacks or other grade limitations.



A characterization of terrain **elevations** is usually correlated against a geospatial surface dimension. Analyzing pair correlations between elevations separated by a lateral surface dimension allow models of regional topography to be made. By using stochastic models of the terrain with well-characterized probability distributions, one can use that as an input or constraining stimulus for navigability and formal verification.

Characterization of **fine terrain** requires a scale much less than the topography or land relief of the local region. To remove this macroscopic effect of large scale elevation and slope changes, data is often detrended to reveal only the fine detail.

The fine terrain **roughness** is characterized by a pair correlation function and the frequency representation known as the power spectral density. The typical terrain roughness may follow a random enough pattern so that the information contained in a correlation function or PSD is enough to reveal the stochastic nature of the ground.

Oftentimes the fine terrain **profile** shows enough regularity that the addition of non-randomness to the model becomes effective in reproducing the spectral properties. Fine terrain with elements of randomness can include washboard surface and cobblestone roads. Individual profiles are often referenced for further evaluation and usually referred to as test tracks or courses when used in the context of vehicle evaluation. A stochastic model of the terrain, if it is characterized fully, can reduce the storage data requirements by orders of magnitude, leaving a few parameters to describe a specific course profile.

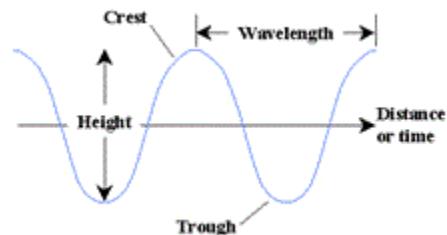
Discrete **obstacles** are always characterized by a geometrically defined spatial profile.

Soil classification schemes simplify the cataloguing of the variety of naturally occurring soils into a handful of types according to a few properties such as granularity.

The **aquatic** classification consists of all ocean and inland water models, and specifically those dealing with the surface characterization of a particular body of water. This can include wave height, lake sizes, and river current speed.

The **sea state** of the water surface at a specific location and time qualifies its general conditions by categorizing according to wave height and other related characteristics such as period and power spectrum. Higher sea-states correspond to rougher conditions, with zero indicating calm waters.

By empirically measuring the historical **wave height** in a specific location and attaching probabilities to the occurrence of various wave heights, we can generate the probability of encountering a particular sea state.



The **wave frequency** is closely related to wave height and wave frequency by a dispersion relationship. Because of the common hydrodynamic properties of water, it is often enough to consider only wave heights when inferring the frequency and wavelength of water.

The distribution of **lake sizes** and **stream currents** has a stochastic explanation, and models of these distributions can fit to empirically collected statistics.

Characterizing **water density** allows for the effects of buoyancy to be determined. Both temperature and salinity can subtly affect the density of water.

The category of **atmospheric** models contains phenomenon that relate insofar as they propagate through the air.

The variation of **wind speed** in specific geographic locations is predominantly a stochastic phenomenon. The general trend is of decreasing likelihood of wind speeds with increasing magnitude.

Precipitation in the form of rain and snow varies according to geographic location and season. Models of rainfall are available, as are environmental standards which give nominal conditions for certain regions and climates.

The statistics of **rainfall amount** has a foundation in extreme value theory, yet the commonly occurring measured rainfalls follow distribution functions that can be modeled.

Based on recently accumulated data, the areal extent of **clouds** has been shown to follow a simple stochastic model.

The modeling of atmospheric **temperature** has a lower spread in relative uncertainty than other environmental contexts. This has much to do with the large thermal inertia of the planet and of the predictability of diurnal and seasonal change. Thus, models of temperature are better suited to description by a mean value and small *relative* (on the Kelvin scale)

The model of **solar** insolation is deterministically predictable, abated only by sporadic cloud cover.

The diurnal or **daily and seasonal** temperature variations can be captured for specific geospatial locations and modeled over the course of a full year, including a moving average for the 24-hour period. In terms of nominal conditions, representative regions can be defined as "hot" to "extreme cold" and typical mean values can be generated to cover the expected values.

Models of transient response to **thermal** steady-state require compliance between the system under study and the environment. In general terms, this means that the system can influence the environment enough to affect the dynamics and the eventual steady-state point (if one exists).

The **humidity** of a particular region is given by nominal conditions. Since humidity only plays a contextual role over a long term, in say promoting corrosion, the nominal values are adequate. Therefore, environmental standards serve as better defining documents than models based on empirical data for a specific location.

Airborne **particulates** can show persistent or transient effects. After significant natural events such as the eruption of volcanoes, the density and size of particulates can increase rapidly. Otherwise, a heterogeneous mix of particles from different origins describes the typical distribution.

Airborne particulates can thus show great diversity in **particle size**. If the particles can grow over time, such as ice crystals, the steady-state distribution in parts reflects where the particle is in with respect to its life cycle. The same process occurs with suspended water-borne particulates.

Airborne particulates show ranges in **particle density** usually depending on the transient event.

Background noise from the environment is referred to as **clutter**

Radio and radar transmitters can present very high levels of energy and electromagnetic field strength to a system and the equipment and subsystems it contains. This type of environment is referred to as the external **Radio Frequency (RF) Electromagnetic Environment (EME)**. RF EME is described in terms of field strength as a function of frequency.

An **electromagnetic pulse** is a rare occurrence of either natural or man-made origin. One variation of this is high power high power microwave which can be a single pulse or transmitted as a repetitive waveform.

Lightning effects can be either direct or indirect. The direct effects refers to the situation were the lightning channel couples to the system, whereas indirect refers to situations where the lightning is observed only remotely through sound, flashes, vibration, or interference.

Static electricity can interact with the environment to create an **electrostatic discharge**

A vehicle can also generate **internal sources of EM energy**, which by way of proximity can interfere with objects in the environment.

Annex 2: PDF Models

The set of PDF models follow a pattern of a variate with parameters that allow us to collect all of the information as triple-stores.

PDF mapdot K*f(A,B,C) ~> X

What this does is apply a lambda function with scaling factor K and fixed parameters A,B,C to a variate list X ranging from Min to Max (with a step interval). The entire list is shown in **Figure 64**.

Characteristic	Model	Title	Xaxis	Location	Lat/Lon	PDF	Moment s	Max	Min	Interva l
cloudArea	Ocean_clouds	cloud diameter distribution	m	Pacific near Hawaii	17.5/-162.5	$1.0 * \text{power_law_2}(0.1, \text{Dist})$	[median]	100000.0	1.0	1.1
clutterPower	Variable_clutter	clutter power distribution Rayleigh	w/m^2	The World	0.0/0.0	$1.0 * \exp(1.0, \text{Dist})$	[mean]	10.0	0.01	1.1
clutterPower	Variable_clutter_maxent	clutter power distribution MaxEnt	w/m^2	The World	0.0/0.0	$1.0 * \text{besselk0_sqrt}(1.0, \text{Dist})$	[mean]	10.0	0.01	1.1
lakeSize	Amazon_lakes	lake size distribution	km^2	Amazon region	-5.0/-60.0	$1.0 * \text{power_law_2}(0.19, \text{Dist})$	[median]	1000.0	0.01	1.05
lakeSize	Canadian_lakes	lake size distribution	km^2	Northern Quebec	54.0/-75.0	$1.0 * \text{power_law_2}(0.1, \text{Dist})$	[median]	1000.0	0.1	1.05
particleSizes	ice_particles	ice particle size distribution	micron	Continental USA	38.0/-100.0	$[0.995 * \text{power_law_2_area}(3.0, \text{Dist}), 0.005 * \exp_{\text{area}}(3000.0, \text{Dist})]$	[[median], [mean]]	10000.0	1.0	1.1
particleSizes	particles	particle size distribution	nm	A Volcanic area	63.0/-16.0	$[0.5 * \exp(3.0, \text{Dist}), 0.5 * \exp(0.1, \text{Dist})]$	[[mean], [mean]]	10.0	0.001	1.1
particleSizes	rain_drops_standard	water droplet size distribution	mm	Continental USA	38.0/-100.0	$[0.9951 * \exp(0.489, \text{Dist}), 0.0049 * \exp(0.996, \text{Dist})]$	[[mean], [mean]]	10.0	0.001	1.1
rainfall	Iowa_rainfall	Rainfall Distribution	mm/hr	Iowa state	41.658/-91.548	$1.0 * \text{besselk0_sqrt}(2.29, \text{Dist})$	[mean]	100.0	0.001	1.05
rateNumber	lightning_rate	lightning rate distribution	num/mi ²	Continental USA	38.0/-100.0	$1.0 * \text{diffusion_accel}(16.5, 5.5, \text{Dist})$	[[diff], [accel]]	100.0	0.1	1.1
slopes	CONUS_slopes	Slope Distribution	rise/run	Continental USA	38.0/-100.0	$1.0 * \text{besselk0_sqrt}(0.037, \text{Dist})$	[mean]	10.0	0.01	1.05
waveFrequency	San_Diego_waves	Wave frequency Distribution	hz	Point Loma South	32.67/-117.2419	$1.0 * \text{pierson_moskowitz}(0.085, \text{Dist})$	[mean]	0.6	0.05	1.01
waveFrequency	San_Diego_waves	Wave frequency Distribution	hz	San Nicolas Island North	32.25/-119.5	$1.0 * \text{pierson_moskowitz}(0.085, \text{Dist})$	[mean]	0.6	0.05	1.01
waveHeight	Atlantic_seaboard_waves	Wave height Distribution	m	Atlantic area	35.0/-75.0	$1.0 * \text{bessel_seastate}(1.0, 81, 26.0, \text{Dist})$	[mean, depth]	16.0	0.001	1.05
waveHeight	Michigan_waves	Wave height Distribution	m	Lake Michigan	43.0/-87.0	$1.0 * \text{bessel_seastate}(0.0, 56, 12.5, \text{Dist})$	[mean, depth]	10.0	0.001	1.05
waveHeight	Superior_waves	Wave height Distribution	m	Lake Superior	49.0/-90.0	$1.0 * \text{bessel_seastate}(0.0, 85, 16.0, \text{Dist})$	[mean, depth]	12.0	0.001	1.05
windSpeed	Germany_wind_energy	Wind Energy distribution	MW-Hr	Germany	49.0/9.0	$1.0 * \exp(1737, \text{Dist})$	[mean]	10000.0	1.0	1.05
windSpeed	Ontario_wind_energy	Wind Energy	MW-Hr	Ontario	52.0/-82.0	$1.0 * \exp(178, \text{Dist})$	[mean]	1000.0	1.0	1.05

windSpeed	Oregon_wind_energy	Wind Speed as Energy distribution	MPH^2	Oregon state	43.0/-121.0	1.0*besselk0_sqrt(144,Dist)	[mean]	10000.0	1.0	1.1
windSpeed	Oregon_wind_speed	Wind Speed distribution	mi/hr	Oregon state	43.0/-121.0	1.0*besselk0(12,Dist)	[mean]	100.0	1.0	1.05

Characteristic	Model	Title	Xaxis	Location	Lat/Lon	PDF	Moments	Max	Min	Interval
cloudArea	Ocean_clouds	Cloud diameter distribution	m	Pacific near Hawaii	17.5/-162.5	1.0*power_law_2(0.1,Dist)	[median]	100000.0	1.0	1.1
clutterPower	Variable_clutter	clutter power distribution Rayleigh	w/m^2	The World	0.0/0.0	1.0*exp(1.0,Dist)	[mean]	10.0	0.01	1.1
clutterPower	Variable_clutter_maxent	clutter power distribution MaxEnt	w/m^2	The World	0.0/0.0	1.0*besselk0_sqrt(1.0,Dist)	[mean]	10.0	0.01	1.1
lakeSize	Amazon_lakes	lake size distribution	km^2	Amazon region	-5.0/-60.0	1.0*power_law_2(0.19,Dist)	[median]	1000.0	0.01	1.05
lakeSize	Canadian_lakes	lake size distribution	km^2	Northern Quebec	54.0/-75.0	1.0*power_law_2(0.1,Dist)	[median]	1000.0	0.1	1.05
particleSizes	Ice_particles	Ice particle size distribution	micron	Continental USA	58.0/-100.0	[0.995*power_law_2_area(3.0,Dist), 0.005*exp_area(3000.0,Dist)]	[median], [mean]	10000.0	1.0	1.1
particleSizes	particles	particle size distribution	nm	A Volcanic area	63.0/-16.0	[0.5*exp(3.0,Dist), 0.5*exp(0.1,Dist)]	[mean], [mean]	10.0	0.001	1.1
particleSizes	rain_drops_standard	water droplet size distribution	mm	Continental USA	58.0/-100.0	[0.9951*exp(0.489,Dist), 0.0049*exp(0.996,Dist)]	[mean], [mean]	10.0	0.001	1.1
rainfall	Iowa_rainfall	Rainfall Distribution	mm/hr	Iowa state	41.658/-91.548	1.0*besselk0_sqrt(2.29,Dist)	[mean]	100.0	0.001	1.05
rateNumber	Lightning_rate	Lightning rate distribution	num/min	Continental USA	38.0/-100.0	1.0*diffusion_accel(16.5,5,Dist)	[diff], [accel]	100.0	0.1	1.1
slopes	CONUS_slopes	Slope Distribution	rise/run	Continental USA	58.0/-100.0	1.0*besselk0_sqrt(0.037,Dist)	[mean]	10.0	0.01	1.05
waveFrequency	San_Diego_waves	Wave frequency Distribution	hz	Point Loma South	32.67/-117.2419	1.0*person_moskovitz(0.085,Dist)	[mean]	0.6	0.05	1.01
waveFrequency	San_Diego_waves	Wave frequency Distribution	hz	San Nicolas Island North	32.25/-119.5	1.0*person_moskovitz(0.085,Dist)	[mean]	0.6	0.05	1.01
waveHeight	Atlantic_seaboard_waves	Wave height Distribution	m	Atlantic area	85.0/-75.0	1.0*bessel_seastate(1.0,81.26,0,Dist)	[mean, depth]	16.0	0.001	1.05
waveHeight	Michigan_waves	Wave height Distribution	m	Lake Michigan	43.0/-87.0	1.0*bessel_seastate(0.0,56.12,5,Dist)	[mean, depth]	10.0	0.001	1.05
waveHeight	Superior_waves	Wave height Distribution	m	Lake Superior	49.0/-90.0	1.0*bessel_seastate(0.0,85.16,0,Dist)	[mean, depth]	12.0	0.001	1.05
windSpeed	German_wind_energy	Wind Energy distribution	MW-Hr	Germany	49.0/0.0	1.0*exp(1737,Dist)	[mean]	10000.0	1.0	1.05
windSpeed	Ontario_wind_energy	Wind Energy distribution	MW-Hr	Ontario	52.0/-82.0	1.0*exp(178,Dist)	[mean]	1000.0	1.0	1.05
windSpeed	Oregon_wind_energy	Wind Speed as Energy distribution	MPH^2	Oregon state	43.0/-121.0	1.0*besselk0_sqrt(144,Dist)	[mean]	10000.0	1.0	1.1
windSpeed	Oregon_wind_speed	Wind Speed distribution	mi/hr	Oregon state	43.0/-121.0	1.0*besselk0(12,Dist)	[mean]	100.0	1.0	1.05

Figure 64: Listing of available PDF models, with compact served HTML table shown below it

Annex 3 : Installation

The context server requires the installation of several run-times in addition to the DCS source.

Run-time Packages Required

1. SWI Prolog for Linux, Mac, or Windows (version w64pl636 for 64-bit Windows)
<http://www.swi-prolog.org/Download.html>
2. R Statistics package (version 2.15.2) <http://www.r-project.org/>
3. AT&T Graphviz graphics package (version 2.28.0) <http://www.graphviz.org/>
4. Firefox preferred, but Google Chrome adequate as a browser. Only recent versions of Internet Explorer work well.

Make sure to install all the add-on packages for SWI if installation is not automatically installed. The Prolog runtime will need to know the path to R and Graphviz (this is automatic with a Windows install).

After extracting the source from the archive, go to **Ontology\dcsl\dynamic_context_server** directory to start the DCS.

- **run.pl** (Windows registered)
- **run.sh** (Linux command line)
- **nohup-run-cloud** (running on a Cloud server as a “no-hang-up” job)

The port is 3020 for local clients and 80 for cloud configured (modify network.pl). If you have the DCS set up on a local machine then go to the following link: <http://localhost:3020>

An authorization login is required to administratively load the ontological context data, via menu item **Repository/Load Context Data**. This can be reset by modifying the names and MD5 hash passwords in the **users.db** file in the main **dynamic_context_server** directory.

Annex 4 : Units

We have collected several standard properties and constants as categorized triple-stores.

Standard Atmosphere properties

Attribute	Value
Name	standardAtmosphere
Comment	standard temperature and pressure (STP)
Description	Specification of standard atmosphere
dryAdiabaticLapseRate	9.8*c/km
dryAdiabaticPressureHead	-0.116*km
moistAdiabaticLapseRateTypical	5.0*c/km
molecularWeight	28.966*au
Pressure	1.0*atm
seaLevel	0.0*km
specificGasConstantDryAir	287*j/kg/k
specificGasConstantWaterVapor	462*j/kg/k
specificHeatRatio	1.4
Temperature	70*f
temperature	273.15*k

Water properties

Attribute	Value
name	water
comment	fresh water properties
deltaHvap	9717.1*cal
deltaSvap	26.04*cal/k
molecular_weight	18.015268*au

Solar

Attribute	Value
name	solar
comment	average sunshine
averageAlbedo	0.31
averageSolarInsolation	1366*w/m^2

Physical Constants

Attribute	Value
name	physicalConstants
comment	commonly used physical constants
atomicUnit	1.66053886e-24*g
avogadrosNumber	6.02214129e+23*n
boltzmannConstant	1.38065e-23*j/k
dielectricConstant	8.854187e-12*fd/m
electricalCharge	1.60218e-19*coulombs
faradaysConstant	96485.3383*coulombs/mol
gasConstant	8.314472*j/k/mol
gasConstant	1.9858775*cal/k/mol
gravity	9.80665*m/s^2
gravity	32.174049*ft/s^2
planckConstant	6.62607e-34*j*s
speedLight	300000000.0*m/s
stefanBoltzmann	5.670373e-8*w/m^2/k^4

Units suitable for conversion

Attribute	Value
All Units	symbolic
area	ft^2
area	m^2
density	g/cm^3
density	kg/dm^3
density	kg/m^3
density	lb/ft^3
density	mg/micron^3
density	oz/in^3
density	ton/km^3
dimensionless	percent
length	cm
length	ft
length	in
length	kft
length	km
length	m
length	micron
length	mi
length	mil
length	mm
length	nm
length	yd
mass	kg
mass	lb
pressure	atm
pressure	bar

pressure	millibar
pressure	mpa
pressure	mtorr
pressure	pa
pressure	psi
pressure	torr
temperature	c
temperature	f
temperature	k
temperature	r
time	day
time	decade
time	hr
time	mics
time	min
time	s
time	yr
volume	cm^3
volume	m^3

Annex 5 : Table of Acronyms

AC	Autocorrelation
ACF	Autocorrelation Function
APG	Aberdeen Proving Grounds
ATC	Aberdeen Test Center
BPA	Bonneville Power Administration
BesselK	Modified Bessel Function of the Second Kind
C2M2L	Component,Context,Manufacturing Model Library
CDF	Cumulative Distribution Function
CONUS	Continental United States region
CRG	Curved Regular Grid format for terrain description from OpenCRG
CSIR	Council for Scientific and Industrial Research (South Africa)
CSV	Comma-Separated Values
DCG	Definite Clause Grammar (what renders this text)
DCS	Dynamic Context Server
DEM	Digital Elevation Model for mapping
DOI	Department of the Interior
DSL	Domain Specific Language
E-M	Electro-Magnetic
EME	Electro-Magnetic Environment
EMI	Electro-Magnetic Interference
EMP	Electro-Magnetic Pulse
FFT	Fast Fourier Transform
FMI	Functional Mockup Interface

FMU Functional Mockup Unit
FP Fokker-Planck
FT Fourier Transform
GIS Geographic Information Systems
GPS Global Positioning System
GW Gravity Wave
HTML hypertext markup language (what you are reading right now!)
JPL Jet Propulsion Laboratory
JSON JavaScript Object Notation
LL log-log scale
MC Monte Carlo random number simulation
ME Maximum Entropy
MaxEnt Maximum Entropy
NASA National Aeronautics and Space Administration
O-U Ornstein-Uhlenbeck random walk process model
OSCAR Ontological System for Context Artifacts and Resources
OWL Web Ontology Language
PCC Probabilistic Certificate of Correctness
PDF Probability Density Function (for documentation see Portable Data Format)
PSD Power Spectral Density or Particle Size Distribution depending on context
RDF Resource Description Format
RF Radio Frequency
RMS Root Mean Square
SWEET Semantic Web for Earth and Environmental Terminology
SWH Significant Wave Height
SoS Superposition of Sine waves
TOPS Test Operating Procedures
TTL Turtle Triple-Store format
URL Universal Resource Locator
USGS US geological Survey
XML Extensible Markup Language
YPG Yuma Proving Grounds
YTC Yuma Test Center

References

- [1] S. Bankes, D. Challou, T. Haynes, H. Holloway, P. Pukite, J. Tierno, and C. Wentland, “META Adaptive, Reflective, Robust Workflow (ARRoW),” BAE Systems, Final Report TR-2742, 2011.
- [2] P. R. Pukite, S. Bankes, and D. Challou, “stochastic_analysis.docx.”
- [3] P. R. Pukite, S. Bankes, and D. Challou, “terrain_characterization.docx.”
- [4] P. R. Pukite, S. Bankes, and D. Challou, “diffusive_growth.docx.”
- [5] R. O’Keefe, *The craft of Prolog*. 1990.
- [6] Wielemaker,J., T. Schrijvers, M. Triska, and T. Lager, “SWI-Prolog,” *Theory and Practice of Logic Programming*, vol. 12, pp. 67–96, 2012.
- [7] J. Wielemaker and Z. Huang, “SWI-Prolog and the web,” *Theory and Practice of Logic Programming*, vol. 8, no. 3, pp. 363–392, 2008.
- [8] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann, 2011.
- [9] G. Macgregor, “E-resource management and the Semantic Web: applications of RDF for e-resource discovery,” *The E-Resources Management Handbook*, vol. 18, 2009.
- [10] R. G. Raskin and M. J. Pan, “Knowledge representation in the semantic web for Earth and environmental terminology (SWEET),” *Computers & Geosciences*, vol. 31, no. 9, pp. 1119–1125, 2005.
- [11] C. Yang, R. Raskin, M. Goodchild, and M. Gahegan, “Geospatial Cyberinfrastructure: Past, present and future,” *Computers, Environment and Urban Systems*, vol. 34, no. 4, pp. 264–277, 2010.
- [12] W3C, “SPARQL 1.1 Entailment Regimes.” [Online]. Available: <http://www.w3.org/TR/sparql11-entailment/>. [Accessed: 25-Jan-2013].
- [13] M. Kay, *XSLT programmer’s reference*. Wrox Press Ltd., 2001.
- [14] J. Wielemaker, *SWI Prolog Reference Manual 6.2. 2*. BoD—Books on Demand, 2012.
- [15] PHP Group, “PHP: hypertext preprocessor,” 2001.
- [16] J. Vlissides, R. Helm, R. Johnson, and E. Gamma, “Design patterns: Elements of reusable object-oriented software,” *Reading: Addison-Wesley*, 1995.
- [17] D. Mumford and A. Desolneux, *Pattern Theory: The Stochastic Analysis Of Real-World Signals*. A K Peters, Ltd., 2010.
- [18] ATEC, “US Army Test and Evaluation Command Test Operations Procedure 1-1-011,” 1981.
- [19] Automotive Directorate, “Test Operations Procedure (TOP) 1-1-010 Vehicle Test Course Severity (Surface Roughness),” ATEC, Aberdeen Proving Ground, MD, Final TOP 1-1-010, 2006.
- [20] Automotive Directorate, “Test Operations Procedure (TOP) 01-1-011A Vehicle Test Facilities at Aberdeen Test Center and Yuma Test Center,” US Army Aberdeen Test Center, Aberdeen Proving Ground, MD, Final TOP 01-1-011A, 2012.
- [21] podaac@podaac.jpl.nasa.gov <podaac@podaac.jpl.nasa.gov>, “PO.DAAC.” [Online]. Available: <http://podaac.jpl.nasa.gov/>. [Accessed: 09-Apr-2012].
- [22] C. of E. US Army, “Wave Information Studies.” [Online]. Available: <http://wis.usace.army.mil/hindcasts.shtml>. [Accessed: 31-May-2012].
- [23] T. E. Vanhecke, “Zotero,” *Journal of the Medical Library Association: JMLA*, vol. 96, no. 3, p. 275, 2008.
- [24] Jonathan Cameron, Abhi Jain, Terry Huntsberger, Garrett Sohl, and Rudranarayan Mukherjee, “Vehicle-Terrain Interaction Modeling and Validation for Planetary Rovers,” NASA JPL, Dartslab, JPL Publication 10-15, 2009.
- [25] GitHub, “OpenRefine (OpenRefine).” [Online]. Available: <https://github.com/OpenRefine>. [Accessed: 25-Jan-2013].
- [26] “GRefine RDF Extension.” [Online]. Available: <http://refine.deri.ie/>. [Accessed: 25-Jan-2013].
- [27] Center for History and New Media, “Zotero Quick Start Guide.” [Online]. Available: http://zotero.org/support/quick_start_guide.
- [28] J. Ritterbush, “Supporting Library Research with LibX and Zotero,” *Journal of Web Librarianship*, vol. 1, no. 3, pp. 111–122, 2007.
- [29] T. Huang, S. Hardman, A. Bingham, A. Takagi, Q. Chau, and M. Gangl, “Building the Next Generation Data Management and Archive System,” presented at the AGU Fall Meeting Abstracts, 2009, vol. 1, p. 1061.
- [30] Atlanta Engineering, “Calculate Vehicle Speed from Skid Mark Distance.” [Online]. Available: http://www.atlantaeng.com/aes_calculator_vehicle_speed_combined.html. [Accessed: 08-Nov-2012].

- [31] Army, “Research, Development, Test and Evaluation of Materiel for Extreme Climatic Conditions,” APD, Headquarters, Dept of the Army, Army Regulation 70-38, Sep. 1979.
- [32] ISO, “ISO 12944-5-2007.” [Online]. Available: <http://www.scribd.com/doc/56997496/ISO-12944-5-2007>. [Accessed: 02-Nov-2012].
- [33] O. LivingSteel, “Corrosion - Living Steel - Living Steel.” [Online]. Available: <http://www.livingsteel.org/corrosion-3>. [Accessed: 02-Nov-2012].
- [34] Wikipedia, “Sea state - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Sea_state. [Accessed: 21-Jun-2012].
- [35] S. Machlup, “Noise in Semiconductors: Spectrum of a Two-Parameter Random Signal,” *Journal of Applied Physics*, vol. 25, no. 3, pp. 341–343, 1954.
- [36] Wikipedia, “Ornstein–Uhlenbeck process - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Ornstein%2080%93Uhlenbeck_process. [Accessed: 19-Oct-2012].
- [37] C. Becker and P. S. Els, “GEROTEK data from CSIR for C2M2L program.” .
- [38] G. Gonçalves and J. Santos, “Propagation of Dem Uncertainty: An Interval Arithmetic Approach.,” in *XXII International Cartographic Conference, Spain*, 2005.
- [39] USGS, “USGS/EROS Find Data/Products and Data Available/DEM.” [Online]. Available: http://eros.usgs.gov/#/Find_Data/Products_and_Data_Available/DEM. [Accessed: 24-Apr-2012].
- [40] VTP, “DEM (digital elevation model).” [Online]. Available: <http://vterrain.org/Elevation/dem.html>. [Accessed: 12-Nov-2012].
- [41] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” *The Semantic Web*, pp. 722–735, 2007.
- [42] P. Yue, J. Gong, and L. Di, “Augmenting geospatial data provenance through metadata tracking in geospatial service chaining,” *Computers & Geosciences*, vol. 36, no. 3, pp. 270–281, 2010.
- [43] M. C. Bowers, “The distributions of seasonal river flows: lognormal or power-law?,” MSc, Purdue.

Ontological System for Context Artifacts and Resources (OSCAR) ¹

Revision A

Dr. Rudranarayan Mukherjee
Thomas Huang
Nga Quach

National Aeronautics and
Space Administration



Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91109-8099
California Institute of Technology

¹ © 2013 Jet Propulsion Laboratory, California Institute of Technology. All rights reserved. Distribution Statement C : Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document shall be referred to the DARPA Technical Information Office

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. JPL RESPONSIBILITIES.....	2
3. METHODOLOGY AND DEVELOPMENT PROCESS	4
3.1 Semantic Web Development Methodology	4
Capturing Complex Relationship and Taxonomy	5
Leverage SWEET Ontologies.....	7
4. SYSTEM ARCHITECTURE	9
The OSCAR Software Stack	11
5. SYSTEM CAPABILITIES	12
6. OSCAR INSTALLATION.....	15
6.1 Prerequisites	15
6.2 Dependencies and Downloads.....	15
6.3 Unpack and Environment Configuration	15
6.4 Install Java	16
6.5 Install Groovy.....	16
6.6 Install Grails	16
6.7 Install Apache Maven.....	16
6.8 Up and Running.....	16
Running Apache Fuseki.....	17
Running Apache Solr	17
Running OSCAR web service.....	17
Firewall Configuration (Optional)	19
7. MANAGEMENT OF THIS DOCUMENT	20

1. INTRODUCTION

JPL is a member of a larger BAE led team that has successfully developed and won a proposal in response to the DARPA-BAA-11-47 Component, Context, and Manufacturing Model Library 1 (C2M2L-1) form by the Tactical Technology Office, DARPA. The following text taken from BAA is provided to the reader for understanding the sponsor's needs:

The DARPA META program is developing a formal metalanguage for the representation of complex cyber-electro-mechanical systems, a set of design tools and metrics for performing trade-space exploration, and a set of verification tools for stochastic formal verification of large system designs...

One of the principal thrusts of this program is on the development of overall environment models (e.g., terrain, atmosphere, water, etc.) affecting drivetrain and mobility subsystems to include amphibious considerations...

Terrain models are expected to represent the surface/fluid that an amphibious infantry fighting vehicle would traverse, ranging from paved road surfaces to rocky, mountainous terrain, slope, discrete obstacles (such as step climbs, v-ditches, etc.), mud, sand, snow, and water fording (both salt water in a ship-to-shore deployment typical of a Marine Air- Ground Task Force, as well as fresh or coastal water for river/lake/etc. fording)...

Particulates models of interest include atmospherically-borne particulate matter such as dust, sand, snow, ice particles, water-borne particulates (when submerged during amphibious transit), and volcanic ash, and their interaction with the mobility and drivetrain subsystems....

The software tools, documentation, specifications, and sample models being produced under the META, iFAB, and vehicleforge.mil efforts are being developed as open-source software.

2. JPL RESPONSIBILITIES

As described in the proposed TERRAIN ONTOLOGY AND GRANULAR MEDIA MODELING FOR MOBILITY CONTEXT task plan, the ontology portion of the task requires the following:

- JPL will adapt its SWEET ontologies to provide a formal ontology to formulate, design, and implement architecture for search/locating, storing, retrieving and maintaining an ontology of the context models.
- JPL will work with the BAE C2M2L-1 team to formulate, design, and implement architecture to search, locate, store, retrieve and maintain a semantic-based digital library for the context models. The context library will store all of the contexts developed over the life of this project. The library will be based on an expansion of the SWEET ontologies.

The final deliverable is a knowledgebase library system for context models. An archival system that is driven by the backend ontology designed for environmental context along with concepts for common archival transactions.

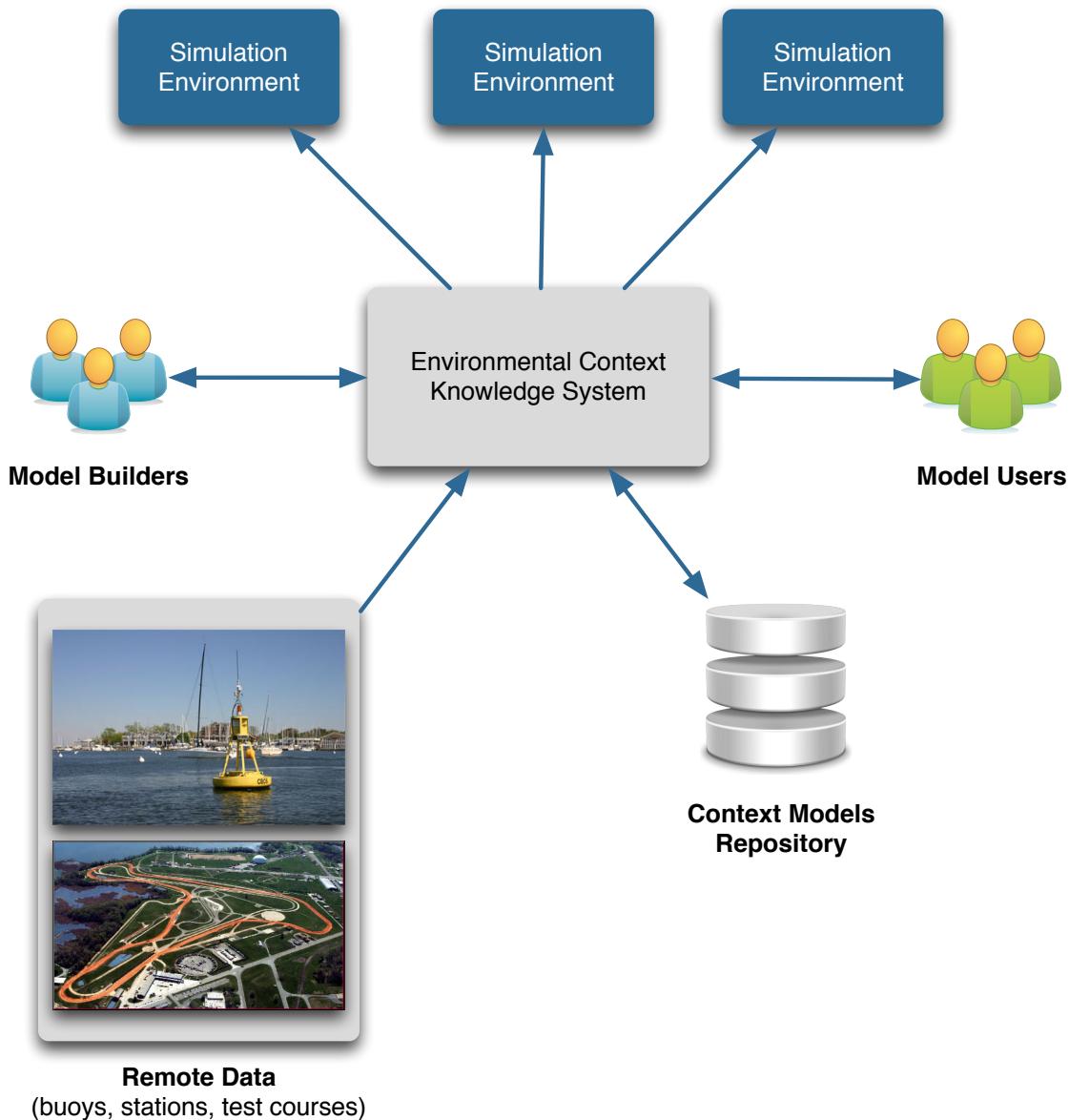


Figure 1 High-Level System Architecture

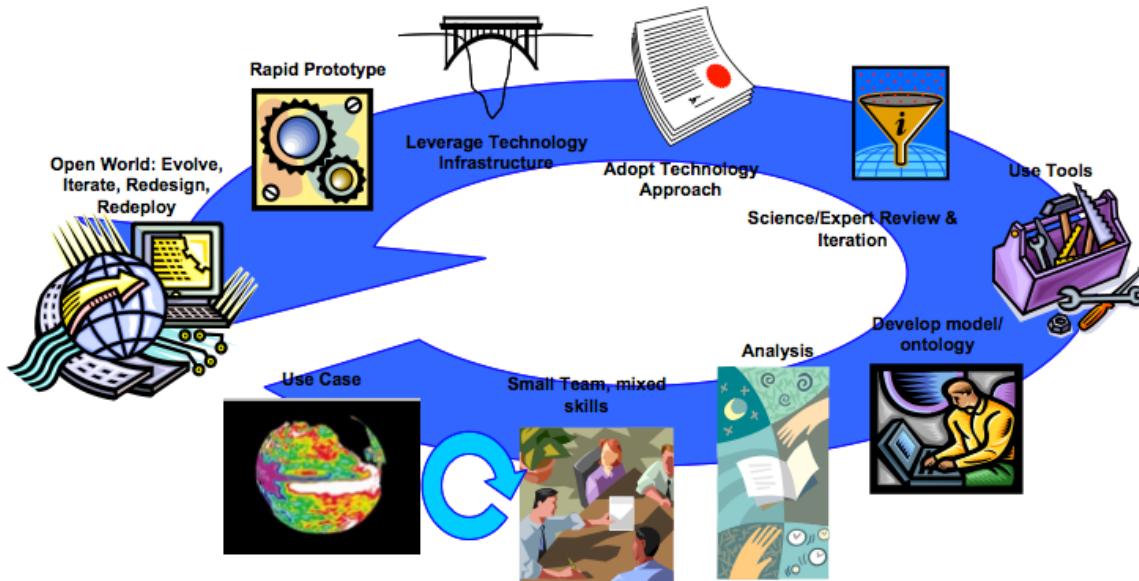
3. METHODOLOGY AND DEVELOPMENT PROCESS

Methodology is a guideline system for solving problem. A process is a series of actions or steps taken to achieve an end. Both of these are essential in development of any data system. This section describes the methodology, the process, and some of the considerations were made in developing this knowledgebase system.

3.1 Semantic Web Development Methodology

Developing an ontological system requires careful planning and promotes teaming. An iterative methodology was adopted from the beginning with the following steps

- Use Case development
- Small Team with maxed skills
- Analysis
- Develop model / ontology
- Use Tools
- Science/Expert Review & Iteration
- Adopt Technology Approach
- Leverage Technology Infrastructure
- Rapid Prototype
- Open World: Evolve, Iterate, Redesign, Redeploy



Capturing Complex Relationship and Taxonomy

Through working closely with the sponsor and building a team with mixed skill sets: project management, ontologist, system and information architect, model builder, testbed developer, and software developer, a set of use cases were rendered. These use cases enable discussion on taxonomy and relationship between artifacts.

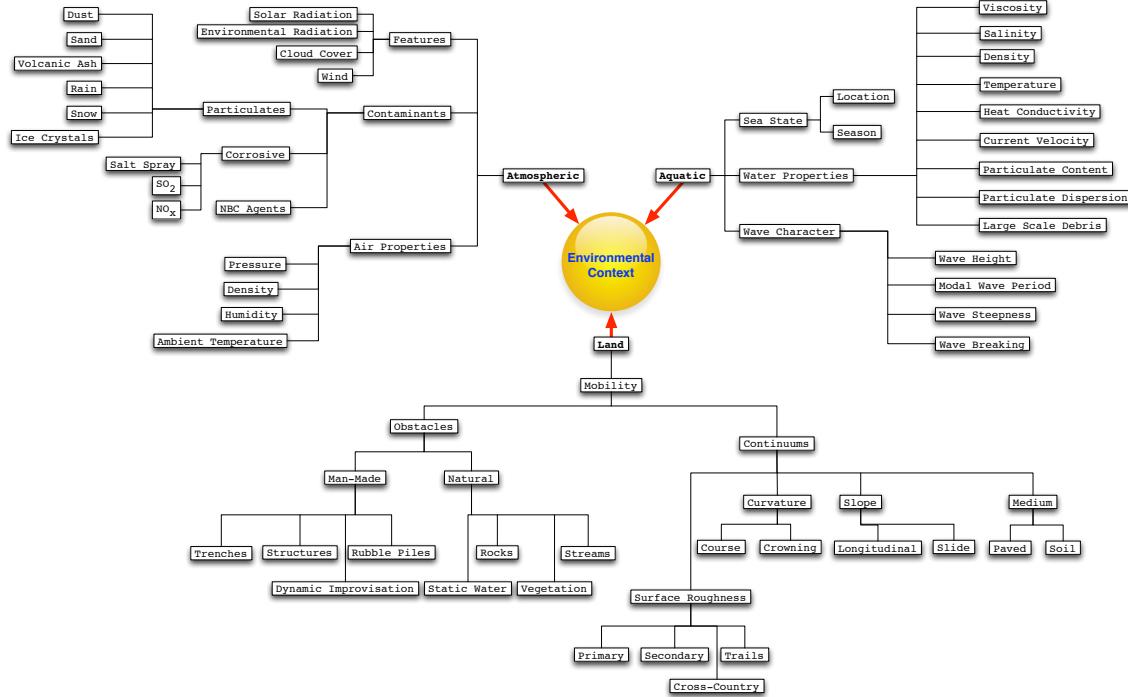


Figure 2 Environmental Taxonomy Organization

The diagram below illustrates using environmental context to link various classes of metadata together.

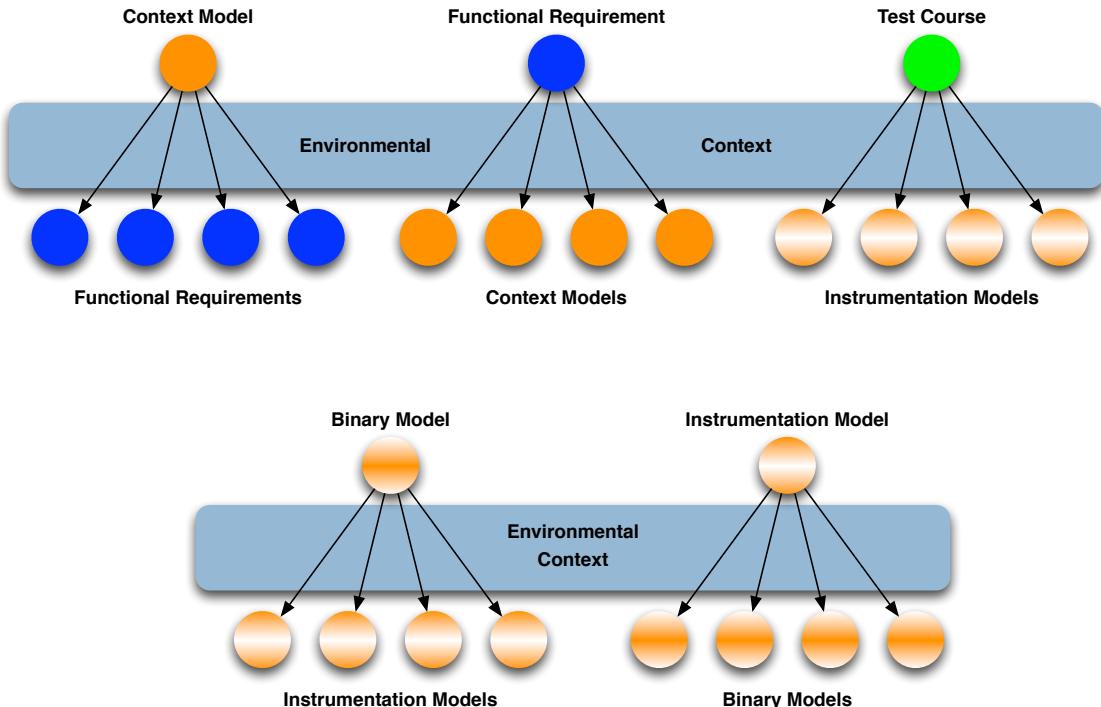


Figure 3 Linking Between Metadata and Artifacts

Leverage SWEET Ontologies

The Semantic Web for Earth and Environmental Terminology (SWEET) ontologies (<http://sweet.jpl.nasa.gov>) is a widely adapted by the Earth Science community as the de facto ontology for modeling earth environment. The ontologies were initially developed to capture the relationships between keywords defined by the Global Change Master Directory (GCMD) (<http://gcmd.jpl.nasa.gov>). The SWEET ontologies enables scalable classification of Earth system concepts and has expanded for support space science in recent years. The current release, SWEET 2.3, is highly modular with over 6000 concepts and 200 separate ontologies.

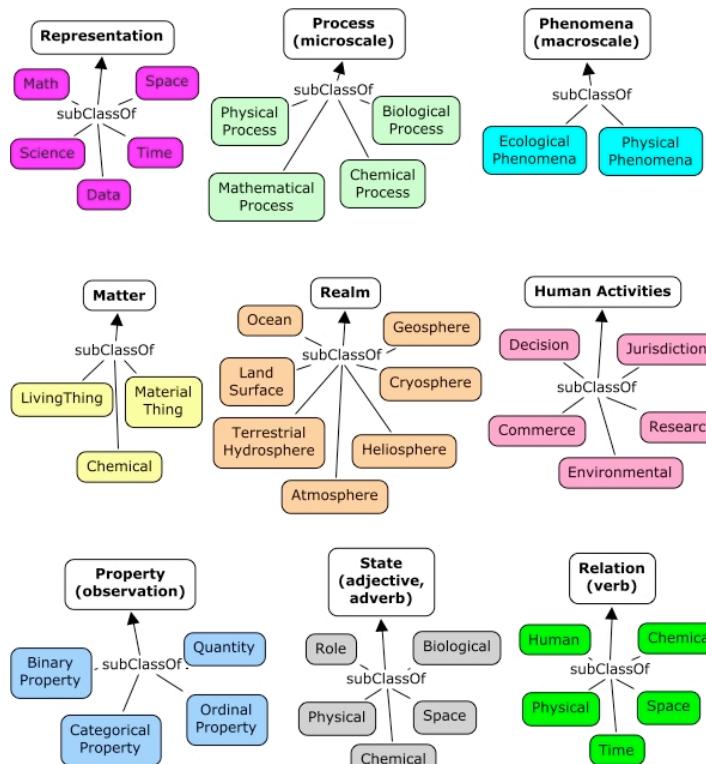


Figure 4 SWEET Ontologies

SWEET ontologies is designed for model natural environment. With terrain modeling also include manmade features and obstacles; the first major milestone of this task is in development of the backend ontology that works with the FANG requirements and captures the expectations of the model and testbed developers. SWEET, as the upper level ontology, promotes the following steps in adapting the richness of SWEET:

- Import – import only the ontologies required
- Expand – introduce new concepts and attributes specific to the application domain

- Specialize – extend and specialize SWEET concepts according to the application domain.

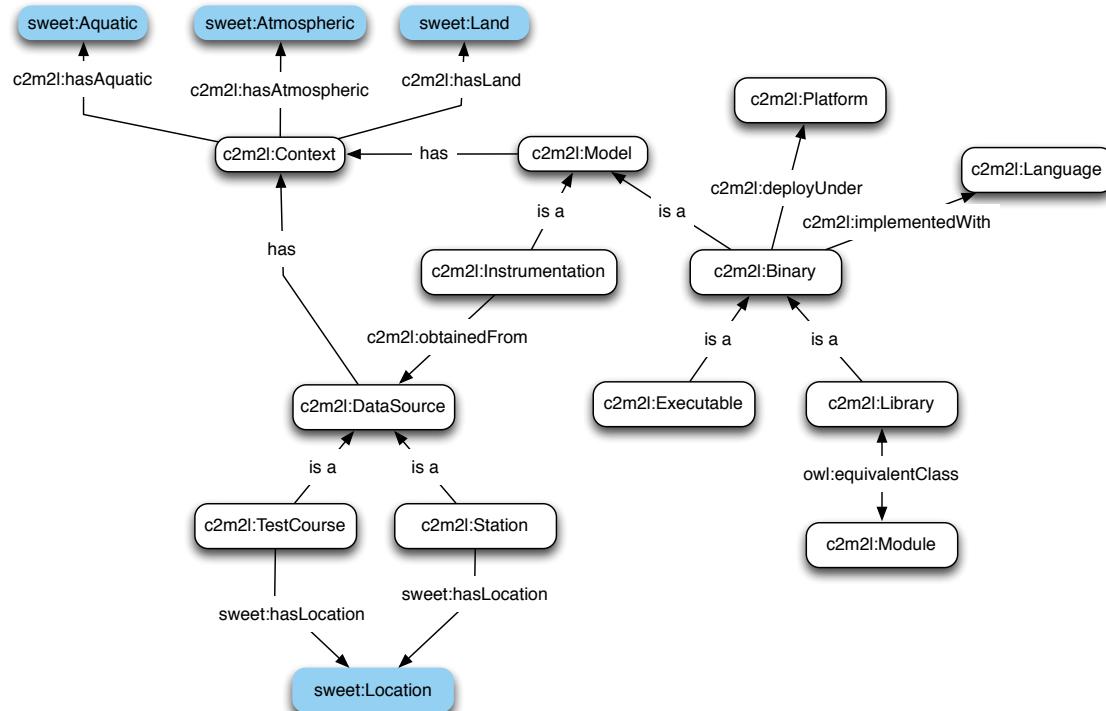


Figure 5 Leverage the SWEET Ontologies

Tools For Ontology Development

A collection of industry-standard designing tools was used to facilitate the design of the OSCAR ontology. These tools are used for documentation and validation of the ontology in development. Below is a list of the tools used

- Protégé
- TopBraid Composer
- SemanticWorks
- OxygenXML

4. SYSTEM ARCHITECTURE

The Ontological System for Context Artifacts and Resources (OSCAR) is designed from the ground up in accordance to the RESTful service architecture. It promotes abstraction and separation of concerns, by considering objects within the system as resources and each object has a set of operations. The key components of OSCAR includes

- Framework for handling RDF/OWL data including reasoning
- Repository for reliable store and retrieval of data
- Fast indexed search
- Security architecture to support authentication, authorization, and communication

Java™ is used for the implementation of OSCAR, which maximize OSCAR's portability. Over the years, Free and Open Source Software (FOSS) has proven itself to be the winning card for developing any software system. Examples of successful FOSS projects include Linux, Apache Software Foundation, Eclipse Foundation, Android Open Source Project, Perl, PHP, Python, etc. OSCAR is designed and developed to leverage from industry-standard open source components, which frees our sponsor from any long term maintaining licensing and vendor lock-in

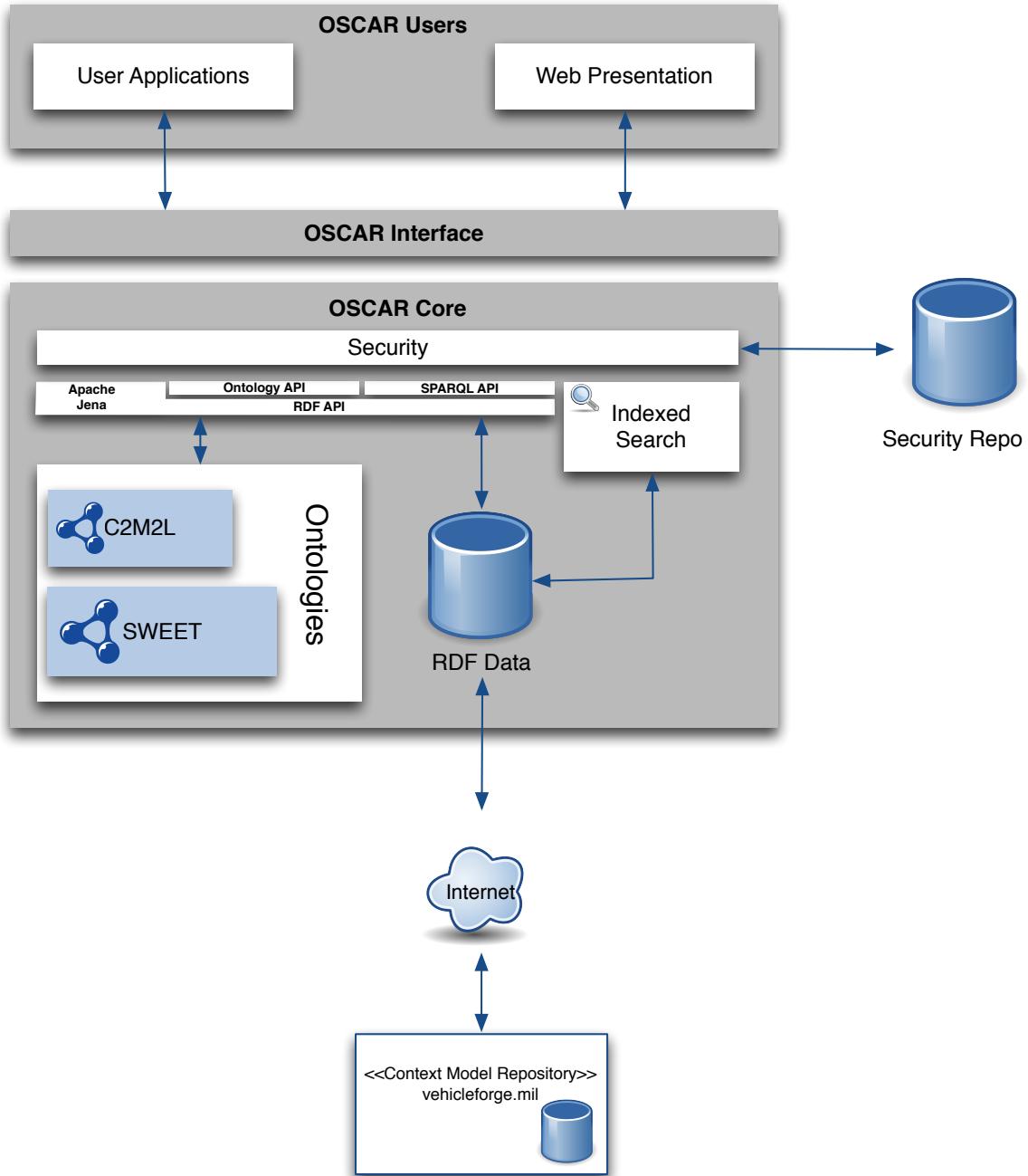


Figure 6 OSCAR Webservice Architecture

OSCAR is designed with the following goals

- A semantic web based system
- Leverages successful JPL data service projects
- RESTful service architecture
- Ontology based on SWEE and adapted to specific mobility contexts for C2M2L

- Archival and distribution - Create, Read, Update, and Delete (CRUD) context model artifacts
- Indexed and guided searches
- Auto linking between artifacts and resources
- Pluggable backend triplestore
- Pluggable security model

The OSCAR Software Stack

- Groovy – The dynamic scripting language built for the Java™ Virtual Machine (JVM)
- Grails – The dynamic web framework built using the Groovy scripting language
- Apache Jena – The RDF/OWL framework for handling RDF data
- Apache Fuseki – The opensource triplestore
- Apache Solr – The indexed search engine
- LDAP (Optional) – For user authentication and authorization

5. SYSTEM CAPABILITIES

In this production release of OSCAR, the system delivers the following capabilities captured in the screenshots below

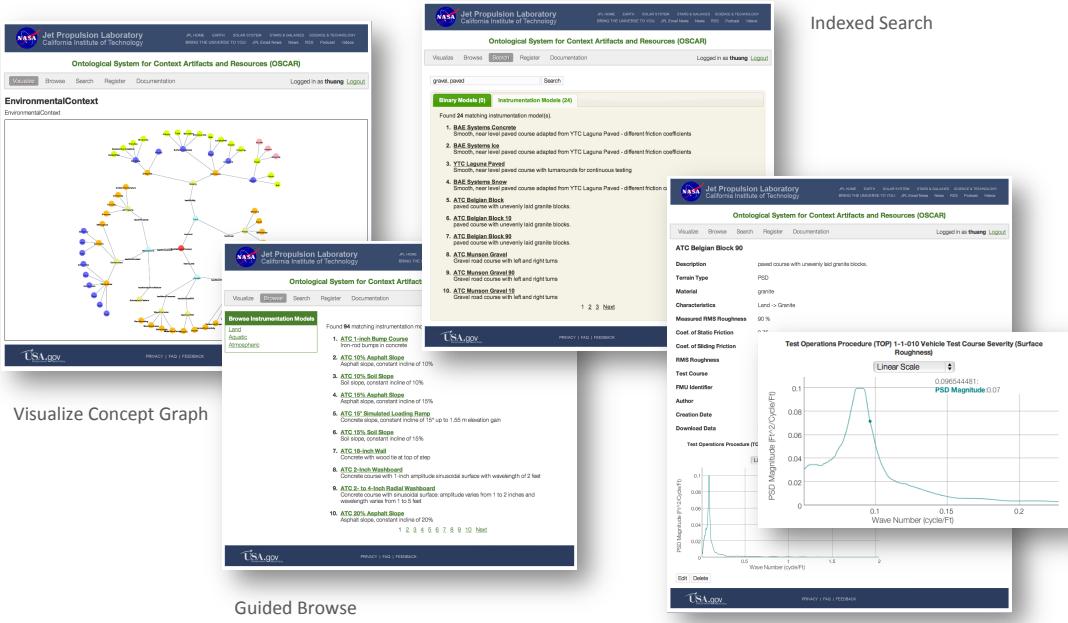


Figure 7 OSCAR Screenshots 1

1. Visualize concept graph. This graph illustrates who information is mapped within OSCAR. The users can zoom in/out of the visual, click on a concept for closer inspection, and freely reposition the visual graph.
2. Guided Browse. All models and requirements are paginated and users are guided by working with the left panel filter
3. Index Search. This offers a horizontal search across all context and artifact descriptions. This enables users to quickly lookup models according to specific context or key phases.
4. Model Detail. Each model has a detail description page. Models are divided into instrumentation and binary models. Instrumentation models are tabular data obtained from data center or produced by model builder. Binary models are either library or executable. They are platform-specific with inputs and outputs. For PSD instrumentation data, the model detail page also provides visual plot of the data.

The user can download the model from this page. For tabular data, OSCAR provides JSON, CSV, and tab-delimited model data. For binary models, OSCAR delivers the model in a compressed TAR file.

Below consists of additional screenshots of OSCAR

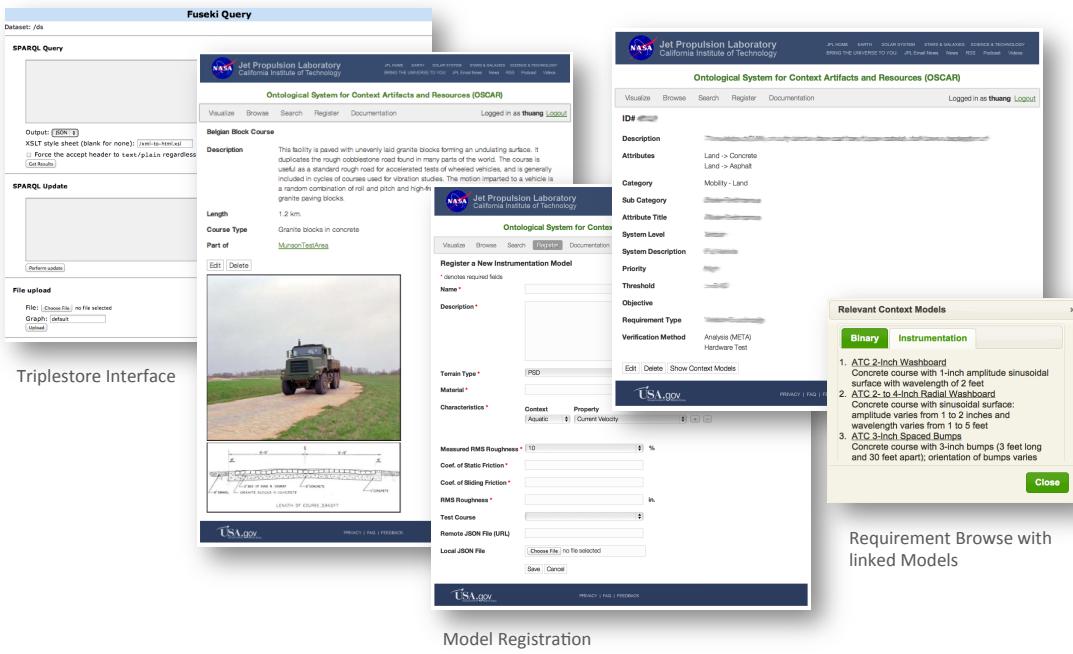


Figure 8 OSCAR Screenshots 2

1. **Triplestore interface.** This is an interface to the Apache Fuseki triplestore. This web interface enables the user/administrator to interact directly with the triplestore via SPARQL, load and export data into the store.
2. **Test Course.** Each instrumentation model obtained from a test course should have an associated test course data sheet, which provides detailed geographic information on the test course.
3. **Model Registration.** This is a form-based interface for model builder to register models. The OSCAR system minds the context and description of the model to enable it to automatically link to test courses and requirements.
4. **Requirement Browse.** This provides detail description on the requirement and the environmental context it associate with. Since this is a knowledgebase system, each requirement also has a list of models it links with. This enables the test bench developer to quickly identify the list of models to use to satisfy a specific requirement.

OSCAR also support range browse for models, which includes ranges like temperature and slope/grade.

© 2013 Jet Propulsion Laboratory, California Institute of Technology. All rights reserved. Distribution Statement C : Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document shall be referred to the DARPA Technical Information Office

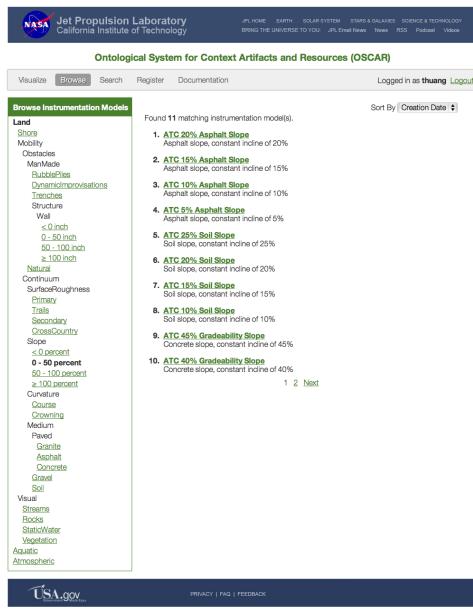


Figure 9 OSCAR Range Selection

To help user to quickly jump the to the requirement they are looking for, OSCAR user interface offers a requirement number search function.

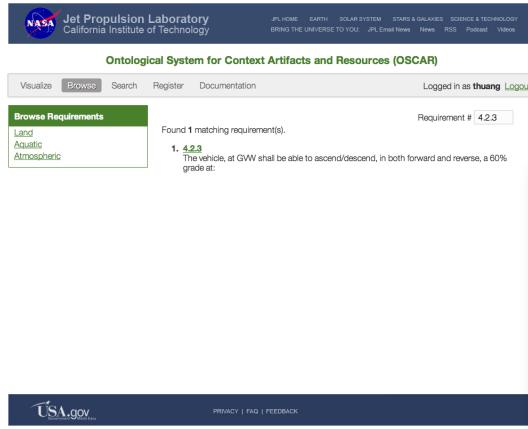


Figure 10 OSCAR Requirement Number Search

6. OSCAR INSTALLATION

6.1 Prerequisites

Prior to running OSCAR, the following must be installed and/or configured. While the entire OSCAR software stack is developed using pure Java™ technology, that is, OSCAR is very portable and can be deployed onto any platform with Java™ Runtime installed, the instructions presented here were captured on a **CentOS 5.8** platform using **T-CSH (tcsh)**, an enhanced UNIX C shell (csh). The installation is performed under the user account (**jdoe**), whose home directory is under (/home/jdoe)

6.2 Dependencies and Downloads

- Java 1.6.0_37
<http://www.oracle.com/technetwork/java/javase/downloads>
- Groovy 1.8.6
<http://dist.groovy.codehaus.org/distributions/>
- Grails 2.0.4
<http://grails.org/Download>
- Apache Fuseki 0.2.4
<http://archive.apache.org/dist/jena/binaries/>
- Apache Maven 3.0.4
<http://maven.apache.org/download.html>
- Apache Solr 3.6.1
<http://www.apache.org/dyn/closer.cgi/lucene/solr/3.6.1>

6.3 Unpack and Environment Configuration

The Java, Groovy, Grails, and Maven are the common tools/runtime environment for OSCAR. It is recommended to have them installed under a global location, such as /usr/local/. For this installation guide, the three packages will be installed under a local directory (/home/jdoe/local)

```
% pwd
/home/jdoe
% mkdir local
% cd local
```

6.4 Install Java

```
% pwd  
/home/jdoe/local  
% chmod ugo+x $HOME/Downloads/jdk-6u37-linux-i586.bin  
% $HOME/Downloads/ jdk-6u37-linux-i586.bin  
% setenv JAVA_HOME $cwd/ jdk1.6.0_37/  
% setenv PATH ${JAVA_HOME}/bin:${PATH}
```

6.5 Install Groovy

```
% pwd  
/home/jdoe/local  
% unzip $HOME/Downloads/groovy-binary-1.8.6.zip  
% setenv GROOVY_HOME $cwd/ groovy-1.8.6  
% setenv PATH ${GROOVY_HOME}/bin:${PATH}
```

6.6 Install Grails

```
% pwd  
/home/jdoe/local  
% setenv GRAILS_HOME $cwd/grails-2.0.4  
% setenv PATH ${GRAILS_HOME}/bin:${PATH}
```

6.7 Install Apache Maven

```
% pwd  
/home/jdoe/local  
% tar -zxvf $HOME/Downloads/apache-maven-3.0.4-bin.tar.gz  
% setenv MAVEN_HOME $cwd/apache-maven-3.0.4  
% setenv PATH ${MAVEN_HOME}/bin:${PATH}
```

6.8 Up and Running

This Alpha release of OSCAR is delivered as semi-packaged distribution to simplify the installation process. A detailed installation procedure will be delivered as part of the OSCAR's final delivery.

This distribution is delivered as a self-contained compressed tarball (`jpl-oscar-1.0.tar.gz`). This installation is performed under `/home/jdoe/dev` directory

```
% pwd  
/home/jdoe  
% mkdir dev
```

© 2013 Jet Propulsion Laboratory, California Institute of Technology. All rights reserved. Distribution Statement 16
C : Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document
shall be referred to the DARPA Technical Information Office

```
% cd dev  
% tar -zvxf jpl-oscar-1.0.tar.gz  
% ls  
C2M2L-1  
% cd C2M2L-1  
% ls  
apache-solr  apache-solr-3.6.1      jena-fuseki-0.2.4  
ontologies   oscar-web
```

Running Apache Fuseki

```
% cd jena-fuseki-0.2.4  
% mkdir tdb  
% sh fuseki-server --update --loc=tdb /ds
```

To check Apache Fuseki is running, point your browser to
<http://localhost:3030>

Running Apache Solr

```
% pwd  
/home/jdoe/dev/C2M2L-1  
% cp -r apache-solr/oscar apache-solr-3.6.1/example/
```

- Edit file apache-solr-3.6.1/example/oscar/conf/data-config.xml
- Locate the 'baseDir' attribute and change its value to <oscar-web>/triples, where <oscar-web> is the fully qualified path to the location of the oscar-web directory.
e.g.
baseDir="/home/jdoe/dev/C2M2L-1/oscar-web/triples"

```
% pwd  
/home/jdoe/dev/C2M2L-1  
% cd apache-solr-3.6.1/example/  
% java -Dsolr.solr.home=oscar -jar start.jar
```

To check Apache Solr is running, point your browser to
<http://localhost:8983/solr/>

Running OSCAR web service

```
% pwd  
/home/jdoe/dev/C2M2L-1  
% cp oscar-rdf/* oscar-web/triples
```

- Edit bootstrap file: `oscar-web/grails-app/conf/BootStrap.groovy`.
- Locate line 37, which should starts with 'rdf: ['
- Update the next 9 lines to reflect the location of the ontologies and the RDF data files. This distribution is preconfigured to assume the root location is `/home/jdoe/dev/C2M2L-1/`

e.g.

```
rdf: [
    "file:///home/jdoe/dev/C2M2L-
1/ontologies/sweet_2.3/sweetAll.owl",
    "file:///home/jdoe/dev/C2M2L-1/ontologies/ContextModel.rdf",
    "file:///home/jdoe/dev/C2M2L-1/ontologies/Course.rdf",
    "file:///home/jdoe/dev/C2M2L-
1/ontologies/EnvironmentalContext.rdf",
    "file:///home/jdoe/dev/C2M2L-1/ontologies/Requirement.rdf",
    "file:///home/jdoe/dev/C2M2L-1/oscar-
web/src/resources/triples/test_courses.rdf",
    "file:///home/jdoe/dev/C2M2L-1/oscar-
web/src/resources/triples/binary_models.rdf",
    "file:///home/jdoe/dev/C2M2L-1/oscar-
web/src/resources/triples/instrumentation_models.rdf",
    "file:///home/jdoe/dev/C2M2L-1/oscar-
web/src/resources/triples/requirements.rdf"
]
```



```
% pwd
/home/jdoe/dev/C2M2L-1
% cd oscar-web
```

Download all dependencies using Apache Maven and install the downloaded libraries for OSCAR to use.

```
% mvn dependency:copy-dependencies
% cp target/dependency/*.jar lib/
```

Now OSCAR webservice is ready to run

```
% grails clean -https
% grails run-app -https
```

The server is ready with the following message

```
| Server running. Browse to http://localhost:8080/oscar or
https://localhost:8443/oscar
```

Point your browser to https://localhost:8443/oscar

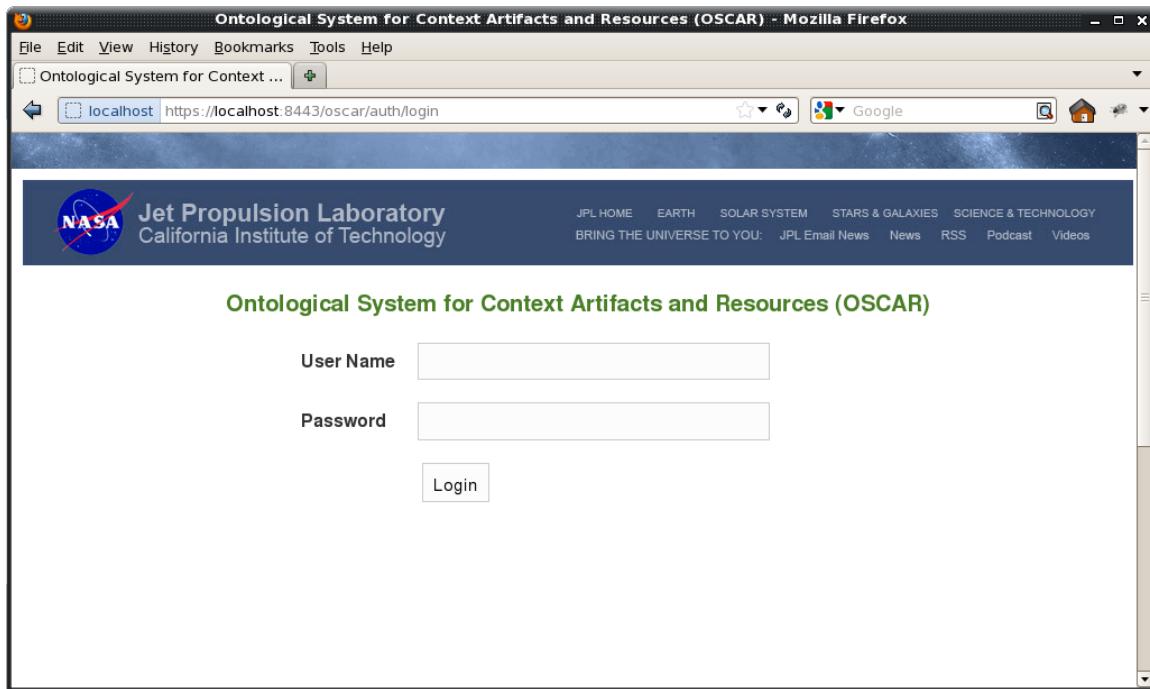


Figure 11 OSCAR landing page

Try logging in using the following account

User Name: thuang
Password: thuang

Note: to add more users, update file `oscar-web/grails-app/conf/BootStrap.groovy` starting line 23.

Firewall Configuration (Optional)

By default, CentOS 5.8 blocks all TCP points. To make OSCAR accessible within local Intranet, a firewall exception must be added for TCP port 8443. To make this change, it requires either root or sudo privilege.

```
% sudo vi /etc/sysconfig/iptables

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --
-dport 8443 -j ACCEPT

% sudo /etc/init.d/iptables restart
```

7. MANAGEMENT OF THIS DOCUMENT

Document Custodian
Rudranarayan M. Mukherjee
Mail Stop 198-235
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Phone: 818-354-2677
Fax: 818-393-3254
Email: *Rudranarayan.M.Mukherjee@jpl.nasa.gov*

January 2013

Technical Report

Probabilistic Models and Model Checking
Context Model Library for C2M2L-1
Component, Context, and Manufacturing Model Library

Prepared for BAE Systems

C2M2L

Subcontract 338724

Technical Point of Contact:

Dr. Darren Cofer

Rockwell Collins, Inc.
7805 Telegraph Rd. #100
Bloomington, MN 55438
Telephone: (319) 263-2571
ddcofer@rockwellcollins.com

Business Point of Contact:

Ms. Kelly M. Scott

Rockwell Collins, Inc.
400 Collins Rd. NE, MS 121-200
Cedar Rapids, IA 52498
Telephone: (319) 295-7323
kmscott@rockwellcollins.com

Distribution Statement C: Distribution Authorized to U.S Government Agencies and their contractors.
Other requests for this document shall be referred to the DARPA Technical Information Office.



Rockwell Collins, Inc.

400 Collins Rd. NE

Cedar Rapids, Iowa 52498

Table of Contents

1	Executive Summary.....	2
2	Drive Train Model.....	3
	2.1 Drive Train Model Description.....	3
	2.1.1 Throttle Driver Module.....	4
	2.1.2 Diesel Engine Module.....	4
	2.1.3 Transmission Module.....	5
	2.1.4 Transmission Controller Module	6
	2.1.5 Vehicle State Module.....	8
	2.1.6 Terrain Module	8
	2.1.7 Vehicle Load and Mileage Computations.....	11
	2.2 Drive Train Model Properties	14
	2.2.1 Vehicle Properties for Moderate Terrain	14
	2.2.2 Vehicle Properties for Moderately Smooth Terrain.....	15
	2.3 Drive Train Model Simulations	16
	2.3.1 Vehicle Start.....	16
	2.3.2 Response to Slope Increase (Level to Uphill).....	17
	2.3.3 Response to Slope Decrease (Uphill to Downhill)	18
	2.3.4 Response to Drastic Slope Increase (Downhill to Steep Uphill)	18
3	PRISM Terrain Modeler.....	19
	3.1 The Terrain Model	19
	3.1.1 Kuchar.....	19
	3.1.2 Max Entropy	19
	3.2 Terrain Specifications	20
	3.3 Running the Tool	23
4	Assume/Guarantee.....	26
5	References	38

1 Executive Summary

Rockwell Collins supported BAE performance on DARPA C2M2L Technical Area 2 (Context Models). This report describes our work on probabilistic context models and associated probabilistic analysis methods. Our work covers three main topics:

1. Developing a probabilistic ground vehicle drive train model to interact with terrain context models
2. Creation of probabilistic terrain context models
3. Developing analysis tools for assume/guarantee-style contracts in probabilistic models.

The first topic, enhancing a ground vehicle drive train model, is described in Section 2 of this report. Our starting point was a drive train model from Vanderbilt University which was subsequently translated by Smart Information Flow Technologies (SIFT) for analysis by the PRISM tool [8]. We have made many extensions and improvements to this model to be able to tie it to context models (such as terrain) and to prove properties such as the expected mileage of the vehicle.

The second topic, creation of terrain context models, is described in Section 3 of this report. The terrain modeler is a tool that automates the generation of probabilistic terrain models for PRISM from terrain specifications provided by the user. The tool supports two different terrain models, the model derived by Kuchar [1] and the Max-Entropy model. Kuchar describes a Markov chain model for expressing probabilities of given changes in elevation. The Max-Entropy model uses the probability distribution that has maximum entropy among those permitted by the observed information. Terrain model specifications take the form of specialized PRISM comments that can be embedded in valid PRISM models. Running the terrain modeling tool on a PRISM model containing a terrain specification will result in a file containing the original PRISM model updated with an instance of the specified terrain model.

The third topic, compositional analysis of probabilistic systems using assume/guarantee reasoning, is discussed in Section 4. Compositional reasoning is a structured approach to verification that enables the analysis of systems whose size would otherwise overwhelm even modern analysis tools [5]. Compositional reasoning enables the analysis of large systems of components by first decomposing the overall problem into several smaller and more tractable problems, solving those problems, and then combining the individual results to formulate a solution for the original problem. While compositional reasoning techniques for formal verification are common and have been studied extensively, they are relatively new in the area of probabilistic reasoning. The most promising research in this area has focused on the use of adversaries to ensure probabilistic bounds on system behaviors. Even this research, however, appears to have been focused primarily on games: systems with some terminal state. We study the use of adversaries in reactive (steady state) systems and a possible approach for verifying assume/guarantee contracts for reactive systems expressed as PRISM modules.

2 Drive Train Model

2.1 Drive Train Model Description

Our starting point was a drive train model provided by Vanderbilt University which was subsequently translated by Smart Information Flow Technologies (SIFT) for analysis by the PRISM tool. We have made many extensions and improvements to this model to be able to tie it to context models (such as terrain) and to prove properties such as the expected mileage of the vehicle. To achieve this goal, we added state to keep track of the vehicle's position within a terrain model. We also added several formulas to more accurately model the load on the vehicle and to compute the instantaneous mileage. These and other aspects are described in the subsections that follow. We have not attempted to make this vehicle model 100% realistic but just close enough to provide a convincing demonstration of property verification that makes use of probabilistic context models. Users with more domain expertise can tune the parameters. Figure 1 depicts the drive train model. This depiction was created with Simulink. It helps visualize the model and keep track of the names of the inputs and outputs to the PRISM modules¹.

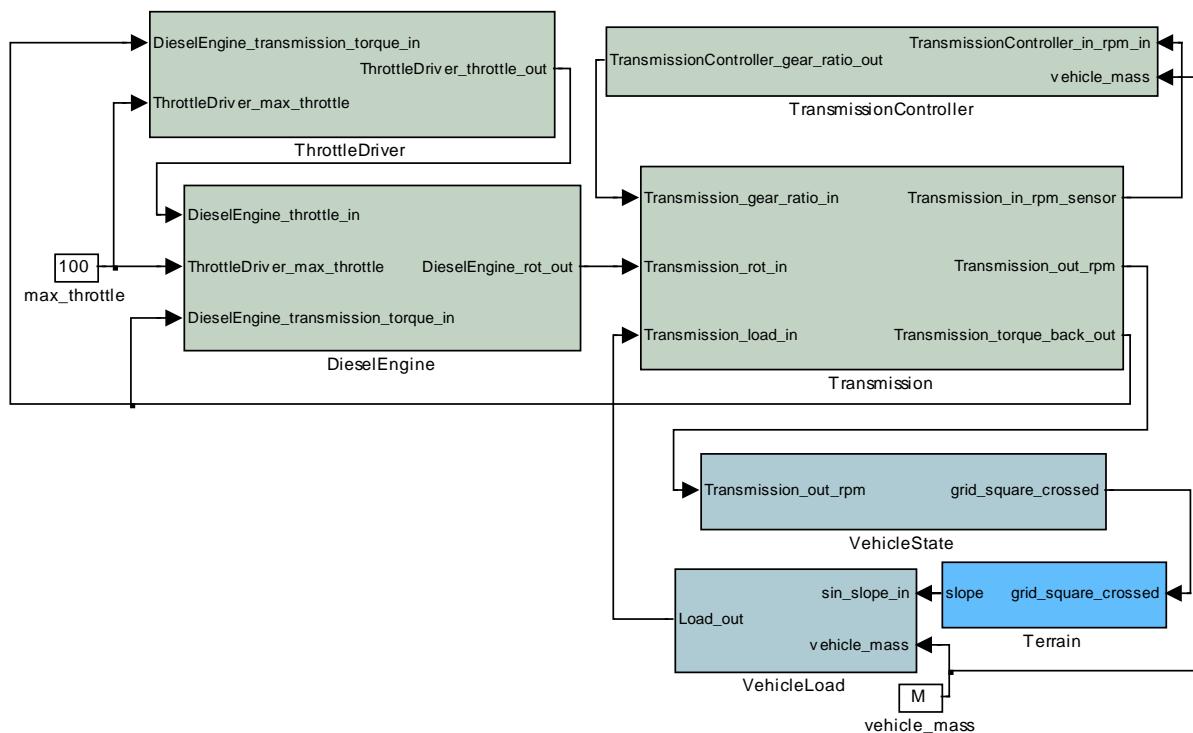


Figure 1 – Drive Train Model Depicted in Simulink®

¹ VehicleLoad is not technically a PRISM module since it does not contain any state variables. It is a set of formulas. The other modules have at least some state (even if it is just to keep track of failure modes).

2.1.1 Throttle Driver Module

The Throttle Driver Module computes the throttle value (ThrottleDriver_throttle_out), which is used by the Diesel Engine Module to compute the engine rpm (DieselEngine_rot_out). The PRISM code is shown in Figure 4. We have experimented with fixed and variable throttles. This version of the vehicle model has a variable throttle with range 0-100%, but only increments of 5% are used (5%, 10%, and so on).

```

const int ThrottleDriver_max_throttle = 100;

formula throttle_goal = max(min(ThrottleDriver_max_throttle,
(DieselEngine_transmission_torque_in + 1500) / DieselEngine_throttle_scale),0);

module ThrottleDriver
    throttle: [0..ThrottleDriver_max_throttle] init 15;
    [tic] (throttle <= throttle_goal - 5) -> (throttle' = throttle + 5);
    [tic] (throttle >= throttle_goal + 5) -> (throttle' = throttle - 5);
    [tic] (throttle < throttle_goal + 5 & throttle > throttle_goal - 5) -> true;
endmodule

formula ThrottleDriver_throttle_out = throttle;

```

Figure 2 – PRISM Code for Throttle Driver Module

2.1.2 Diesel Engine Module

This module computes the engine output in rpm (DieselEngine_rot_out) as the product of the throttle scale (DieselEngine_throttle_scale, equal to 120) and throttle input (DieselEngine_throttle_in) minus the torque back to the engine (DieselEngine_transmission_torque_in). A simplified version (without modes) of the Diesel Engine Module is depicted in Figure 3. The PRISM code is shown in Figure 4.

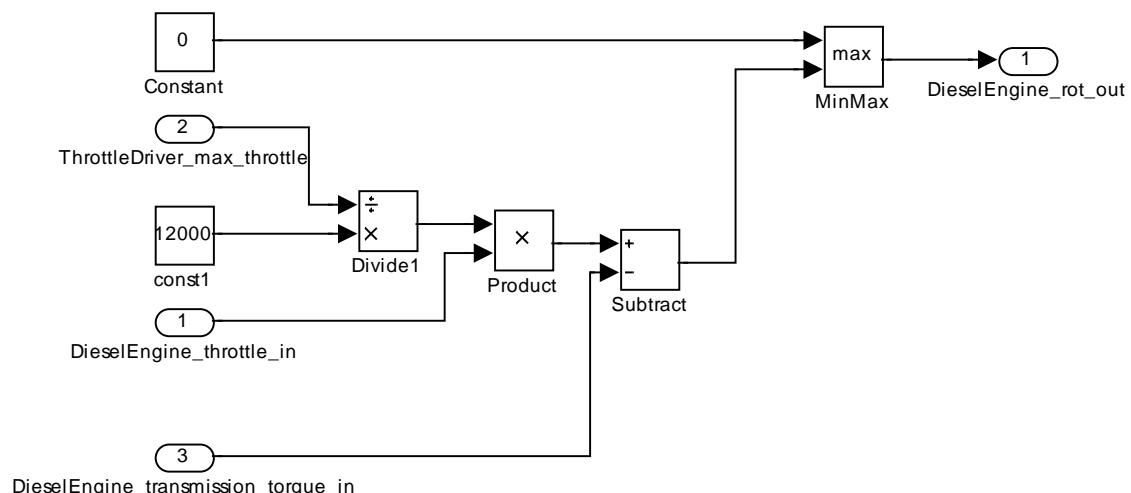


Figure 3 – Simulink® Depiction of Simplified Diesel Engine Module

```

const int DieselEngine_mode_nominal = 0;
const int DieselEngine_mode_failed = 1;
const double DieselEngine_fail_rate = 0;

const double DieselEngine_throttle_scale = 12000.0 / ThrottleDriver_max_throttle;

formula DieselEngine_rot_out =
    (DieselEngine_mode != DieselEngine_mode_nominal) ? 0 :
        max(0, DieselEngine_throttle_scale * DieselEngine_throttle_in -
            DieselEngine_transmission_torque_in);

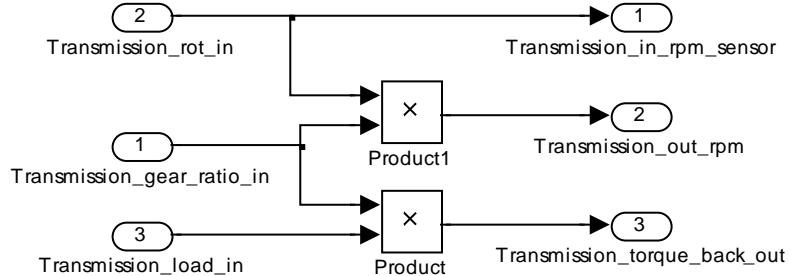
module DieselEngine
    DieselEngine_mode: [0..1] init DieselEngine_mode_nominal;
    [tic] DieselEngine_mode = DieselEngine_mode_nominal ->
        DieselEngine_fail_rate : (DieselEngine_mode' = DieselEngine_mode_failed)
        + 1-DieselEngine_fail_rate: true;
endmodule

```

Figure 4 – PRISM Code for Diesel Engine Module

2.1.3 Transmission Module

The Transmission Module computes the transmission output in rpm (Transmission_out_rpm) as the product of the engine rpm (Transmission_rot_in) and the gear ratio (Transmission_gear_ratio_in) provided by the Transmission Controller Module. It also computes the torque back to the engine (Transmission_torque_back_out) as the product of the gear ratio (Transmission_gear_ratio_in) and the load on the vehicle (Transmission_load_in). A simplified version (without modes) of the Transmission Module is depicted in Figure 5. The PRISM code is shown in Figure 6.

**Figure 5 – Simulink® Depiction of Simplified Transmission Module**

```

const int Transmission_mode_nominal = 0;
const int Transmission_mode_sensor_fail = 1;
const double Transmission_out_rpm_speed_limit = 2000000;
const int Transmission_mode_out_rpm_over_limit = 2;
const double Transmission_sensor_fail_rate = 0.0001;

formula Transmission_out_rpm =
    (Transmission_mode != Transmission_mode_nominal) ? 0 :
    Transmission_rot_in * Transmission_gear_ratio_in;

formula Transmission_out_rpm_sensor =
    (Transmission_mode != Transmission_mode_nominal) ? 0 : Transmission_out_rpm;

formula Transmission_in_rpm_sensor =
    (Transmission_in_sensor_mode != Transmission_mode_nominal) ? 0 :
    Transmission_rot_in;

formula Transmission_torque_back_out = Transmission_gear_ratio_in *
    Transmission_load_in;

module Transmission
    Transmission_mode: [0..2] init Transmission_mode_nominal;
    Transmission_in_sensor_mode: [0..1] init Transmission_mode_nominal;
    Transmission_out_sensor_mode: [0..1] init Transmission_mode_nominal;
    [tic] Transmission_out_sensor_mode != Transmission_mode_nominal -> true;
    [tic] Transmission_out_sensor_mode = Transmission_mode_nominal &
        Transmission_out_rpm <= Transmission_out_rpm_speed_limit ->
        Transmission_sensor_fail_rate : (Transmission_out_sensor_mode' =
            Transmission_mode_sensor_fail) + 1 - Transmission_sensor_fail_rate: true;
    [tic] Transmission_out_sensor_mode = Transmission_mode_nominal &
        Transmission_out_rpm > Transmission_out_rpm_speed_limit ->
        1: (Transmission_mode' = Transmission_mode_out_rpm_over_limit);
endmodule

```

Figure 6 – PRISM Code for Transmission Module

2.1.4 Transmission Controller Module

The Transmission Controller Module computes the gear ratio (TransmissionController_gear_ratio_out), which is used by the Transmission Module. There are six gears (and corresponding gear ratios) defined in the vehicle model. Lower and upper bounds for gear shifting are defined (G1Low, G1High, etc.). The transmission upshifts when the transmission rpm is higher than the upper bound for the current gear. The transmission downshifts when the transmission rpm is lower than the lower bound for the current gear. The Transmission Controller Module includes a counter ('ctr') that restricts the gear to shifting every other step/tic. This prevents the throttle and gear from “co-chatter.” The PRISM code for the Transmission Controller Module is shown in Figure 7.

```

formula level_grade_load = P_inertia + P_tires + P_accessories;

const double G1 = 0.235849057;
const double G2 = 0.327868852;
const double G3 = 0.431034483;
const double G4 = 0.598802395;
const double G5 = 1;
const double G6 = 1.388888889;

const double ShiftRPM = 1800;
const double G1Low = 0;
const double G2Low = ShiftRPM * G1;
const double G1High = ShiftRPM * (G1 + G2)/2 + level_grade_load*(G2-G1);
const double G3Low = ShiftRPM * G2;
const double G2High = ShiftRPM * (G2 + G3)/2 + level_grade_load*(G3-G2);
const double G4Low = ShiftRPM * G3;
const double G3High = ShiftRPM * (G3 + G4)/2 + level_grade_load*(G4-G3);
const double G5Low = ShiftRPM * G4;
const double G4High = ShiftRPM * (G4 + G5)/2 + level_grade_load*(G5-G4);
const double G6Low = ShiftRPM * G5;
const double G5High = ShiftRPM * (G5 + G6)/2 + level_grade_load*(G6-G5);
const double G6High = ShiftRPM * G6;

const int TransmissionController_mode_nominal = 0;
const int TransmissionController_mode_failed = 1;
//const double TransmissionController_fail_rate = 0.0001;
const double TransmissionController_fail_rate = 0;

formula TransmissionController_gear_ratio_out =
    TransmissionController_mode != TransmissionController_mode_nominal ? G1 :
        TransmissionController_gear = 1 ? G1 :
            TransmissionController_gear = 2 ? G2 :
                TransmissionController_gear = 3 ? G3 :
                    TransmissionController_gear = 4 ? G4 :
                        TransmissionController_gear = 5 ? G5 :
                            TransmissionController_gear = 6 ? G6 : 1;

module TransmissionController
    TransmissionController_mode: [0..1] init TransmissionController_mode_nominal;
    TransmissionController_gear: [1..6] init 1;
    ctr: [0..1] init 0;
    [tic] (ctr = 1) & (TransmissionController_mode = TransmissionController_mode_nominal)
    ->
        TransmissionController_fail_rate :
            (ctr' = 0) & (TransmissionController_mode' = TransmissionController_mode_failed) +
            1-TransmissionController_fail_rate :
                (ctr' = 0) & (TransmissionController_gear' =
                    TransmissionController_mode != TransmissionController_mode_nominal ? 1 :
                        (TransmissionController_gear = 1)&(TransmissionController_in_rpm_in > G1High) ? 2:
                        (TransmissionController_gear = 2)&(TransmissionController_in_rpm_in < G2Low ) ? 1:
                        (TransmissionController_gear = 2)&(TransmissionController_in_rpm_in > G2High) ? 3:
                        (TransmissionController_gear = 3)&(TransmissionController_in_rpm_in < G3Low ) ? 2:
                        (TransmissionController_gear = 3)&(TransmissionController_in_rpm_in > G3High) ? 4:
                        (TransmissionController_gear = 4)&(TransmissionController_in_rpm_in < G4Low ) ? 3:
                        (TransmissionController_gear = 4)&(TransmissionController_in_rpm_in > G4High) ? 5:
                        (TransmissionController_gear = 5)&(TransmissionController_in_rpm_in < G5Low ) ? 4:
                        (TransmissionController_gear = 5)&(TransmissionController_in_rpm_in > G5High) ? 6:
                        (TransmissionController_gear = 6)&(TransmissionController_in_rpm_in < G6Low ) ? 5:
                        TransmissionController_gear);
    [tic] (ctr != 1) -> (ctr' = 1);
endmodule

```

Figure 7 – PRISM Code for Transmission Controller Module

2.1.5 Vehicle State Module

The resolution of the terrain models we are using is 100 m. In other words, we have one elevation value for every 100 m x 100 m square in the grid. Since the elevation state (and slope) should only update when a grid square boundary is crossed, we added a Vehicle State Module to keep track of the vehicle's approximate position within the current 100 m x 100 m square of the Terrain Module. The Vehicle State Module sets 'grid_square_crossed' to 1 whenever a grid square boundary is crossed. The elevation state variable (theNextBin) in the terrain module updates if and only if 'grid_square_crossed' is 1. Also, 'delta_position' is used to compute the velocity of the vehicle, which is used in the mileage computation. The PRISM code for the Vehicle State Module is shown in Figure 8.

```

const double Wheel_diameter = 1; //in meters
const double pi = 3.14;
const double time_step = 1; //in seconds
const int grid_size = 100; //in meters

formula Differential_out_rpm = Transmission_out_rpm/10; //Differential
formula delta_position =
floor(Differential_out_rpm*Wheel_diameter*pi*time_step/60);
formula num_crossings = floor((position + delta_position)/grid_size);
formula grid_square_crossed = (num_crossings > 0);

module VehicleState
    position : [0..grid_size] init 0;
    [tic] true -> (position' = position + delta_position -
num_crossings*grid_size);
endmodule

```

Figure 8 – PRISM Code for Vehicle State Module

2.1.6 Terrain Module

The PRISM code for the Terrain Module corresponding to the Moderate Terrain Category is shown in Figure 9. This Terrain Module is defined based on J. K. Kuchar's paper [1] and was created using the PRISM Terrain Modeler we developed (see Section 3). To keep our model manageable, we chose to consider 15 altitude bins in each category. At the upper and lower ends of the altitude range, the transition probabilities are chosen to ensure that the probabilities in each row of the transition matrix sum to 1. Several stand-alone terrain models were delivered in November (some as PRISM files, others as Excel workbooks).

```

const int step = 100;

module terrainModule
// Sigma          342           Terrain Model Parameter
// Beta           1.3e-3        Terrain Model Parameter
// maxElevation   1400          Upper bound on elevation
// minElevation   1100          Lower bound on elevation
// binCount       15            Number of discrete elevations
// tickName       tic           Process synchronization
// currBinName    theNowBin     STATE: slopeModule state
// nextBinName    theNextBin    STATE: terrainModule state
// currElevationName currentElevation OUTPUT: Name of the current elevation signal

```

```

// nextElevationName      nextElevation      OUTPUT: Name of the next elevation signal
// currSlope              slope             OUTPUT: Name of the slope signal
// updateElevationName    grid_square_crossed INPUT: Name of the "step" signal
theNextBin: [0..14];
[tic] ((grid_square_crossed) &
(nextElevation <= 1120.0)) ->
0.2201: (theNextBin' = 1) + // 1130.0
0.0340: (theNextBin' = 2) + // 1150.0
0.0016: (theNextBin' = 3) + // 1170.0
0.7443: (theNextBin' = 0); // 1110.0
[tic] ((grid_square_crossed) &
(nextElevation > 1120.0) &
(nextElevation <= 1140.0)) ->
0.3122: (theNextBin' = 0) + // 1110.0
0.2198: (theNextBin' = 2) + // 1150.0
0.0339: (theNextBin' = 3) + // 1170.0
0.0016: (theNextBin' = 4) + // 1190.0
0.4325: (theNextBin' = 1); // 1130.0
[tic] ((grid_square_crossed) &
(nextElevation > 1140.0) &
(nextElevation <= 1160.0)) ->
0.0510: (theNextBin' = 0) + // 1110.0
0.2617: (theNextBin' = 1) + // 1130.0
0.2194: (theNextBin' = 3) + // 1170.0
0.0338: (theNextBin' = 4) + // 1190.0
0.0016: (theNextBin' = 5) + // 1210.0
0.4325: (theNextBin' = 2); // 1150.0
[tic] ((grid_square_crossed) &
(nextElevation > 1160.0) &
(nextElevation <= 1180.0)) ->
0.0027: (theNextBin' = 0) + // 1110.0
0.0484: (theNextBin' = 1) + // 1130.0
0.2622: (theNextBin' = 2) + // 1150.0
0.2190: (theNextBin' = 4) + // 1190.0
0.0337: (theNextBin' = 5) + // 1210.0
0.0016: (theNextBin' = 6) + // 1230.0
0.4324: (theNextBin' = 3); // 1170.0
[tic] ((grid_square_crossed) &
(nextElevation > 1180.0) &
(nextElevation <= 1200.0)) ->
0.0027: (theNextBin' = 1) + // 1130.0
0.0486: (theNextBin' = 2) + // 1150.0
0.2625: (theNextBin' = 3) + // 1170.0
0.2187: (theNextBin' = 5) + // 1210.0
0.0336: (theNextBin' = 6) + // 1230.0
0.0015: (theNextBin' = 7) + // 1250.0
0.4324: (theNextBin' = 4); // 1190.0
[tic] ((grid_square_crossed) &
(nextElevation > 1200.0) &
(nextElevation <= 1220.0)) ->
0.0027: (theNextBin' = 2) + // 1150.0
0.0487: (theNextBin' = 3) + // 1170.0
0.2629: (theNextBin' = 4) + // 1190.0
0.2183: (theNextBin' = 6) + // 1230.0
0.0335: (theNextBin' = 7) + // 1250.0
0.0015: (theNextBin' = 8) + // 1270.0
0.4324: (theNextBin' = 5); // 1210.0
[tic] ((grid_square_crossed) &
(nextElevation > 1220.0) &
(nextElevation <= 1240.0)) ->
0.0027: (theNextBin' = 3) + // 1170.0
0.0489: (theNextBin' = 4) + // 1190.0
0.2633: (theNextBin' = 5) + // 1210.0
0.2180: (theNextBin' = 7) + // 1250.0
0.0334: (theNextBin' = 8) + // 1270.0
0.0015: (theNextBin' = 9) + // 1290.0
0.4322: (theNextBin' = 6); // 1230.0
[tic] ((grid_square_crossed) &
(nextElevation > 1240.0) &
(nextElevation <= 1260.0)) ->
0.0028: (theNextBin' = 4) + // 1190.0

```

```

0.0489: (theNextBin' = 5) + // 1210.0
0.2637: (theNextBin' = 6) + // 1230.0
0.2176: (theNextBin' = 8) + // 1270.0
0.0333: (theNextBin' = 9) + // 1290.0
0.0015: (theNextBin' = 10) + // 1310.0
0.4322: (theNextBin' = 7); // 1250.0
[tic] ((grid_square_crossed) &
(nextElevation > 1260.0) &
(nextElevation <= 1280.0)) ->
0.0028: (theNextBin' = 5) + // 1210.0
0.0491: (theNextBin' = 6) + // 1230.0
0.2640: (theNextBin' = 7) + // 1250.0
0.2172: (theNextBin' = 9) + // 1290.0
0.0332: (theNextBin' = 10) + // 1310.0
0.0015: (theNextBin' = 11) + // 1330.0
0.4322: (theNextBin' = 8); // 1270.0
[tic] ((grid_square_crossed) &
(nextElevation > 1280.0) &
(nextElevation <= 1300.0)) ->
0.0028: (theNextBin' = 6) + // 1230.0
0.0493: (theNextBin' = 7) + // 1250.0
0.2644: (theNextBin' = 8) + // 1270.0
0.2168: (theNextBin' = 10) + // 1310.0
0.0331: (theNextBin' = 11) + // 1330.0
0.0015: (theNextBin' = 12) + // 1350.0
0.4321: (theNextBin' = 9); // 1290.0
[tic] ((grid_square_crossed) &
(nextElevation > 1300.0) &
(nextElevation <= 1320.0)) ->
0.0028: (theNextBin' = 7) + // 1250.0
0.0494: (theNextBin' = 8) + // 1270.0
0.2648: (theNextBin' = 9) + // 1290.0
0.2166: (theNextBin' = 11) + // 1330.0
0.0329: (theNextBin' = 12) + // 1350.0
0.0015: (theNextBin' = 13) + // 1370.0
0.4320: (theNextBin' = 10); // 1310.0
[tic] ((grid_square_crossed) &
(nextElevation > 1320.0) &
(nextElevation <= 1340.0)) ->
0.0028: (theNextBin' = 8) + // 1270.0
0.0496: (theNextBin' = 9) + // 1290.0
0.2651: (theNextBin' = 10) + // 1310.0
0.2162: (theNextBin' = 12) + // 1350.0
0.0328: (theNextBin' = 13) + // 1370.0
0.0015: (theNextBin' = 14) + // 1390.0
0.4320: (theNextBin' = 11); // 1330.0
[tic] ((grid_square_crossed) &
(nextElevation > 1340.0) &
(nextElevation <= 1360.0)) ->
0.0028: (theNextBin' = 9) + // 1290.0
0.0497: (theNextBin' = 10) + // 1310.0
0.2655: (theNextBin' = 11) + // 1330.0
0.2158: (theNextBin' = 13) + // 1370.0
0.0342: (theNextBin' = 14) + // 1390.0
0.4320: (theNextBin' = 12); // 1350.0
[tic] ((grid_square_crossed) &
(nextElevation > 1360.0) &
(nextElevation <= 1380.0)) ->
0.0028: (theNextBin' = 10) + // 1310.0
0.0499: (theNextBin' = 11) + // 1330.0
0.2659: (theNextBin' = 12) + // 1350.0
0.2495: (theNextBin' = 14) + // 1390.0
0.4319: (theNextBin' = 13); // 1370.0
[tic] ((grid_square_crossed) &
(nextElevation > 1380.0)) ->
0.0028: (theNextBin' = 11) + // 1330.0
0.0501: (theNextBin' = 12) + // 1350.0
0.2662: (theNextBin' = 13) + // 1370.0
0.6809: (theNextBin' = 14); // 1390.0
[tic] (! grid_square_crossed) -> true;
endmodule

```

```

module slopeModule
    theNowBin: [0..14];
    [tic] ( grid_square_crossed ) -> (theNowBin' = theNextBin);
    [tic] (! grid_square_crossed) -> true;
endmodule

/// Generated Terrain Formula
formula currentElevation = 1110.0 + 20.0*theNowBin;
/// Generated Terrain Formula
formula nextElevation = 1110.0 + 20.0*theNextBin;
/// Generated Terrain Formula
formula slope = ((nextElevation - currentElevation)/step);

// We really want sin(slope_angle) for the P_grade formula.
// Note: sin(slope_angle)=rise/(sqrt(rise^2+run^2)).
// Since PRISM can't do a square root, we use a table for possible slope values (not all
used).

formula sin_slope =
    slope = -1 ? -.7071 :
    slope = -.8 ? -.6247 :
    slope = -.6 ? -.5145 :
    slope = -.4 ? -.3714 :
    slope = -.2 ? -.1961 :
    slope = 0 ? 0 :
    slope = .2 ? .1961 :
    slope = .4 ? .3714 :
    slope = .6 ? .5145 :
    slope = .8 ? .6247 :
    slope = 1 ? .7071 : 0;

// Divide by a factor of 4 (assume vehicle will take a longer, smoother path as needed).
// This prevents the vehicle from excessive stalling.
formula sin_slope_in = sin_slope/4;

```

Figure 9 – PRISM Code for (Moderate) Terrain Module (spans multiple pages)

2.1.7 Vehicle Load and Mileage Computations

This portion of the PRISM model computes the instantaneous load on the vehicle as well as the instantaneous mileage and range. The mileage and vehicle range rewards can be used to obtain the expected vehicle mileage and range over the terrain. The primary sources for the formulas and constants used in this module are [2] and [3]. The instantaneous load on the vehicle is

$$P_{\text{inertia}} + P_{\text{grade}} + P_{\text{air_drag}} + P_{\text{tires}} + P_{\text{accessories}},$$

where

- $P_{\text{inertia}} = 0.5 * 1.03 * \text{vehicle_mass} * \text{acceleration}$ is the power to overcome inertia,
- $P_{\text{grade}} = \text{vehicle_mass} * \text{gravity} * \sin(\text{slope angle})$ is the power to overcome the load from grade,
- $P_{\text{air_drag}}$ is the power to overcome air drag,
- P_{tires} is the power to overcome rolling resistance, and
- $P_{\text{accessories}}$ is the power used by the vehicle accessories (such as lights, radio, air conditioning, etc.).

In this first-order vehicle model, we assume the vehicle acceleration is 1 (constant velocity) and that P_accessories is 0.

If the vehicle is moving (nonzero velocity), then the vehicle mileage can be estimated by

$$(\text{energy_per_gallon} * \text{engine_efficiency}) / (\text{instantaneous load}),$$

where energy_per_gallon is the amount of energy in a gallon of diesel (150,150,000 joules) and engine_efficiency is the fraction of the fuel energy converted to work by the engine and delivered to the transmission. We use 18% for the engine efficiency in this model.

Note that no state variables are directly used in the computation of vehicle load or mileage. However, these do rely on velocity, which is computed from the state variable ‘position’ of the Vehicle State Module.

```

const double vehicle_mass = 23991; // kilograms
const double gravity = 9.81;

// Instantaneous load is P_inertia + P_grade + P_air_drag + P_tires + P_accessories
formula P_inertia = 0.5 * 1.03 * vehicle_mass * 1; // Assume constant velocity
formula P_grade = vehicle_mass * gravity * sin_slope_in;

const double drag_coefficient = .6; // Drag coefficient for SUVs is 50-60%.
const double air_density = 1.3; // At sea level, the air density is 1.3 kg/m^3
const double vehicle_frontal_area = 10.8; // Assumes 3.6 meters wide and 3 meters tall const
double rolling_friction_coefficient = 0.01;

formula P_air_drag =
    .5*drag_coefficient*air_density*vehicle_frontal_area*pow(velocity,2);

formula P_tires = rolling_friction_coefficient*vehicle_mass*gravity;

formula P_accessories = 0; //Assume this is negligible

formula Load_out = max(0,P_inertia + P_grade + P_tires + P_accessories);

formula Transmission_load_in = Load_out;

// Mileage Computation
const double energy_per_gallon = 150150000; // joules in a gallon of diesel
const double fuel_capacity = 171; // gallons (used to compute vehicle range)
const double engine_efficiency = .18; // used in BAE Fuel Efficiency example

// Max time a gallon of diesel can last (used for idle scenarios)
const double gallon_max_time = 2*60*60; // seconds. Assumes min burn of 1/2 gal per hr

formula velocity = delta_position/time_step; // meters/second

// If velocity is 0, gas mileage is 0. Otherwise, gas mileage is the minimum of the distance
travelled in 2 hours and the output of the formula based on inertia, grade, drag, rolling
resistance (tires), and accessories.
formula mileage_meters_per_gal = (velocity = 0) ? 0 : min(velocity*gallon_max_time,
    (energy_per_gallon*engine_efficiency)/(Load_out + P_air_drag));

const double meters_per_mile = 1609.344;
formula mileage_miles_per_gal = mileage_meters_per_gal/meters_per_mile;
formula velocity_miles_per_hour = velocity*60*60/meters_per_mile;
formula range_miles_per_tank = mileage_miles_per_gal*fuel_capacity;

rewards "mileage"
    true : mileage_miles_per_gal;
endrewards

rewards "vehicle_range"
    true: range_miles_per_tank;
endrewards

```

Figure 10 – PRISM Code for Vehicle Load and Mileage Computations

2.2 Drive Train Model Properties

2.2.1 Vehicle Properties for Moderate Terrain

We now prove some properties and do some probabilistic computations using PRISM on the drive train model with the Moderate Terrain Module.

2.2.1.1 Ranges of Variables

Ranges of various variables in a PRISM model can be easily computed using a PRISM property of the following form:

```
filter(range, variable, true)
```

Using properties of this form, we were able to compute the following:

- Range of velocity (velocity_miles_per_hour): 0-89 mph
- Range of engine rpm (DieselEngine_rot_out): 0-12000 rpm
- Range of slope (slope): -60% to 60%
- Range of gear (TransmissionController_gear): 1-5

Note that the upper ends for velocity and engine rpm are quite high for a diesel military ground vehicle. Recall that we have not attempted to make this vehicle model 100% realistic but just close enough to provide a convincing demonstration of property verification that makes use of probabilistic context models. We can compute the steady-state probabilities that these variables are less than some threshold. In particular, we find that the value of engine rpm is less than 4000 with a probability of 98.7% and that the vehicle's velocity is less than 5 mpg with a probability of 82.9%. The following properties in PRISM were used to make these computations.

```
S=? [ DieselEngine_rot_out<=4000 ]
S=? [ velocity_miles_per_hour<5 ]
```

2.2.1.2 Stalling

The probability of the vehicle stalling—more precisely, the percentage of all possible states for which the engine rpm (DieselEngine_rot_out) is less than 900 is 2.9%.

This was computed using the following formula and reward structure in PRISM:

```
formula engine_stall = (DieselEngine_rot_out < 900) ? 1 : 0;
rewards "stall"
    true : engine_stall;
endrewards
```

The property in PRISM is written as follows:

```
R{"stall"}=? [ S ]
```

2.2.1.3 Mileage and Range

The instantaneous mileage of the vehicle is computed using the following reward structure in PRISM:

```

rewards "mileage"
    true : mileage_miles_per_gal;
endrewards

```

The expected mileage of the vehicle is 1.4 mpg and is computed with the PRISM property written as follows:

$$R\{ "mileage" \}=? [S]$$

The instantaneous range of the vehicle is similarly computed using the following reward structure in PRISM:

```

rewards "vehicle_range"
    true: range_miles_per_tank;
endrewards

```

The expected range of the vehicle is 238 miles (assuming a 171 gallon tank). This could also be computed directly (external to PRISM) from the expected mileage as $\text{mileage} * 171$, rather than using a separate reward structure.

2.2.2 Vehicle Properties for Moderately Smooth Terrain

Using the Moderately Smooth Terrain Module, we get slightly different results for some of the properties. The PRISM statements are not included in this section since they are the same as in the previous section (Vehicle Properties for Moderate Terrain).

2.2.2.1 Ranges of Variables

- Range of velocity (velocity_miles_per_hour): 0-54 mph
- Range of engine rpm (DieselEngine_rot_out): 0-10200 rpm
- Range of slope (slope): -40% to 40%
- Range of gear (TransmissionController_gear): 1-4

Note that the upper ends for velocity and engine rpm are quite high for a diesel military ground vehicle. We can compute the steady-state probabilities that these variables are less than some threshold. In particular, we find that the value of engine rpm is less than 4000 with a probability of 99.4% and that the vehicle's velocity is less than 5 mpg with a probability of 86.2%.

2.2.2.2 Stalling

The probability of the vehicle stalling—more precisely, the percentage of all possible states for which the engine rpm (DieselEngine_rot_out) is less than 900 is 1.3%.

2.2.2.3 Mileage and Range

The expected mileage of the vehicle is 1.2 mpg. The expected range of the vehicle is 208 miles.

2.3 Drive Train Model Simulations

In this section, we provide some screenshots from simulations of the vehicle model in PRISM (for the Moderate Terrain Category). First we show the full view of PRISM's simulation interface in Figure 11. One can choose the number of steps to simulate on one click by entering that number in the upper left text box. One can also force a transition for the next simulation step by double clicking that transition in the "Manual exploration" section. The results of all state valubles and rewards are shown in the large table at the bottom.

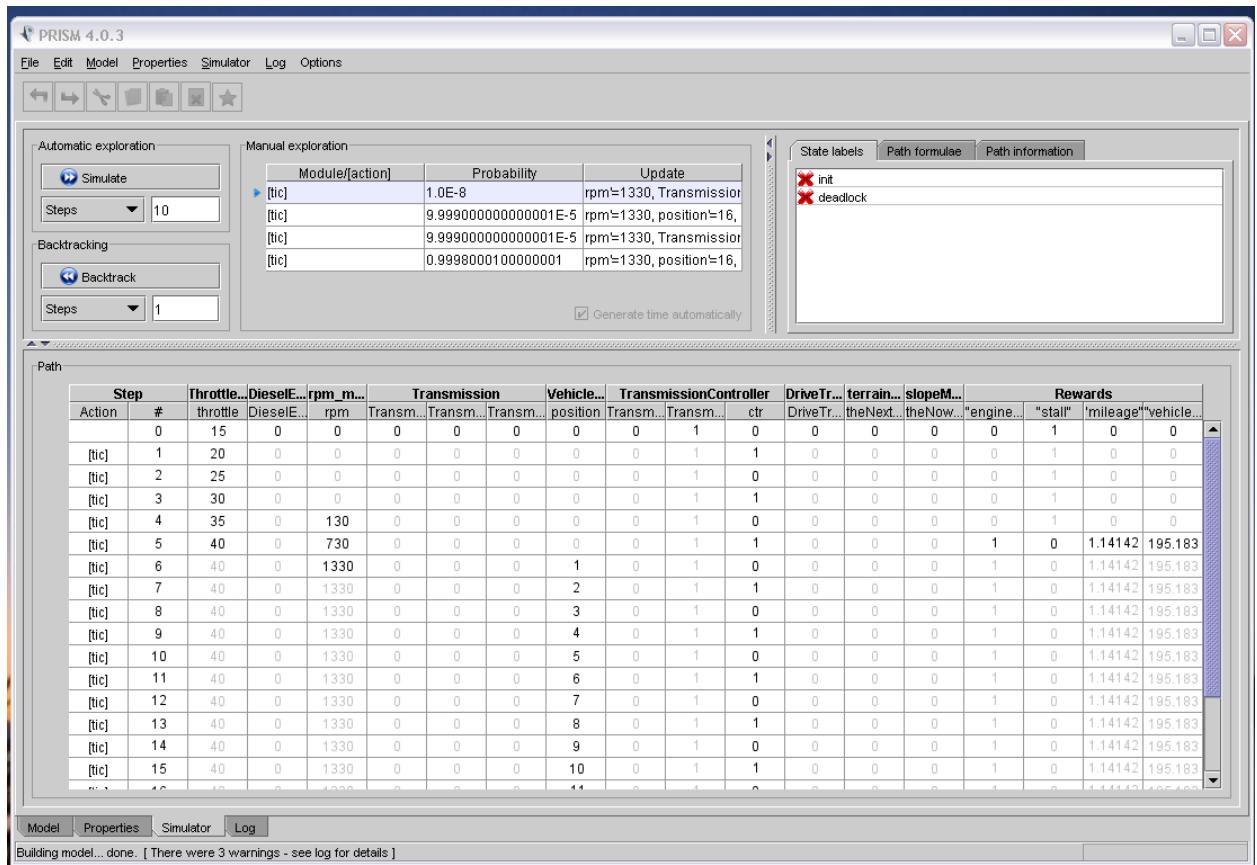


Figure 11 – PRISM Simulation Interface

2.3.1 Vehicle Start

Figure 12 is a zoomed-in version of Figure 11. This shows the first 15 steps of a simulation from the initial state. Notice how the throttle increases from 15% to 40% before rpms are sufficiently high. The rpm_monitor shows the value of engine rpm from the previous step. (There is a one-step delay since this is a state variable updated with the value of a formula.) The position does not begin to change until step 6. Since ‘position’ is a state variable, it must be an integer, so it is the floor of a real number. Until the floor is 1, the vehicle “doesn’t move.” The rewards from left to right, are “engine_sweet_spot” which is 1 whenever engine rpm is between 1100 and 1800. The second reward is “stall” which is 1 whenever the engine rpm is less than 900. The “mileage” reward is the instantaneous mileage of the vehicle in

miles per gallon. The last reward, “vehicle_range” is the instantaneous range of the vehicle in miles per tank, where the fuel capacity of the vehicle is defined to be 171 gallons.

p #	Throttle...DieselE...rpm_m...			Transmission			Vehicle...		TransmissionController		DriveTr...		terrain...		slopeM...		Rewards			
	throttle	DieselE...	rpm	Transm...	Transm...	Transm...	position	Transm...	Transm...	ctr	DriveTr...	theNext...	theNow...	"engine..."	"stall"	"mileage"	"vehicle..."			
0	15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0		
1	20	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0		
2	25	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0		
3	30	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0		
4	35	0	130	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
5	40	0	730	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	
6	40	0	1330	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1.14142	195.183	
7	40	0	1330	0	0	0	2	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	
8	40	0	1330	0	0	0	3	0	1	0	0	0	0	0	0	1	0	1.14142	195.183	
9	40	0	1330	0	0	0	4	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	
10	40	0	1330	0	0	0	5	0	1	0	0	0	0	0	0	1	0	1.14142	195.183	
11	40	0	1330	0	0	0	6	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	
12	40	0	1330	0	0	0	7	0	1	0	0	0	0	0	0	1	0	1.14142	195.183	
13	40	0	1330	0	0	0	8	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	
14	40	0	1330	0	0	0	9	0	1	0	0	0	0	0	0	1	0	1.14142	195.183	
15	40	0	1330	0	0	0	10	0	1	1	0	0	0	0	0	1	0	1.14142	195.183	

Figure 12 – PRISM Simulation for Vehicle Model on Moderate Terrain (Starting out)

```
module rpm_monitor
    rpm: [0 .. 12000];
    [tic] true -> (rpm' = floor(DieselEngine_rot_out));
endmodule
```

2.3.2 Response to Slope Increase (Level to Uphill)

Figure 13 shows the vehicle model’s response to an increase in slope (from level to uphill). Note that the engine rpm drops to 0 when the slope first changes, and the vehicle stalls. Then the throttle increases until engine rpm becomes positive and the vehicle recovers from stall. Note that this is a first-order model of a vehicle and that we don’t actually anticipate the vehicle to stall in the field.

ep #	Throttle...DieselE...rpm_m...			Transmission			Vehicle...		TransmissionController		DriveTr...		terrain...		slopeM...		Rewards			
	throttle	DieselE...	rpm	Transm...	Transm...	Transm...	position	Transm...	Transm...	ctr	DriveTr...	theNext...	theNow...	"engine..."	"stall"	"mileage"	"vehicle..."			
103	40	0	1330	0	0	0	98	0	1	1	0	0	0	0	1	0	1.14142	195.183		
104	40	0	1330	0	0	0	99	0	1	0	0	0	0	0	1	0	1.14142	195.183		
105	40	0	1330	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0		
106	45	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0		
107	50	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0		
108	55	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0		
109	60	0	409	0	0	0	0	0	1	1	0	1	0	0	0	0	0.639734	109.395		
110	60	0	1009	0	0	0	1	0	1	0	0	1	0	0	0	0	0.639734	109.395		
111	60	0	1009	0	0	0	2	0	1	1	0	1	0	0	0	0	0.639734	109.395		
112	60	0	1009	0	0	0	3	0	1	0	0	1	0	0	0	0	0.639734	109.395		
113	60	0	1009	0	0	0	4	0	1	1	0	1	0	0	0	0	0.639734	109.395		
114	60	0	1009	0	0	0	5	0	1	0	0	1	0	0	0	0	0.639734	109.395		

Figure 13 – PRISM Simulation for Vehicle Model Responding to Slope Increase (Level to Uphill)

2.3.3 Response to Slope Decrease (Uphill to Downhill)

Figure 14 shows the vehicle model's response to a decrease in slope (from uphill to downhill). Note that the engine rpm jumps up to 6452 when the slope first changes. Then the throttle decreases and the transmission upshifts (every other step) until engine rpm falls under 1800. Also, note that since the vehicle is travelling downhill, the position is changing by more than 1 meter at each step.

Step [tic]	Throttle... DieselE... rpm_m...			Transmission			Vehicle...		TransmissionController		DriveTr...		terrain...		slopeM...		Rewards			
	Action #	throttle	DieselE...	rpm	Transm...	Transm...	Transm...	position	Transm...	Transm...	ctr	DriveTr...	theNext...	theNow...	"engine..."	"stall"	'mileage'	"vehicle..."		
[tic] 207	60	0	1009	0	0	0	98	0	1	1	0	1	0	0	0	0	0.639734	109.395		
[tic] 208	60	0	1009	0	0	0	99	0	1	0	0	1	0	0	0	0	0.639734	109.395		
[tic] 209	60	0	1009	0	0	0	0	0	1	1	0	0	0	1	0	0	4.97277	850.343		
[tic] 210	55	0	6452	0	0	0	7	0	2	0	0	0	1	0	0	0	4.78192	817.708		
[tic] 211	50	0	5560	0	0	0	16	0	2	1	0	0	0	1	0	0	4.88145	834.727		
[tic] 212	45	0	4960	0	0	0	24	0	3	0	0	0	0	1	0	0	4.78192	817.708		
[tic] 213	40	0	4033	0	0	0	33	0	3	1	0	0	0	1	0	0	4.97277	850.343		
[tic] 214	35	0	3433	0	0	0	40	0	4	0	0	0	1	0	0	0	4.97277	850.343		
[tic] 215	30	0	2301	0	0	0	47	0	4	1	0	0	0	1	1	0	5.12621	876.582		
[tic] 216	30	0	1701	0	0	0	52	0	4	0	0	0	0	1	1	0	5.12621	876.582		
[tic] 217	30	0	1701	0	0	0	57	0	4	1	0	0	0	1	1	0	5.12621	876.582		
[tic] 218	30	0	1701	0	0	0	62	0	4	0	0	0	0	1	1	0	5.12621	876.582		
[tic] 219	30	0	1701	0	0	0	67	0	4	1	0	0	0	1	1	0	5.12621	876.582		

Figure 14 – PRISM Simulation for Vehicle Model Responding to Slope Decrease (Uphill to Downhill)

2.3.4 Response to Drastic Slope Increase (Downhill to Steep Uphill)

Figure 15 shows the vehicle model's response to an increase in slope (from downhill to steep uphill). Note that the engine rpm drops to 0 when the slope first changes, and the vehicle stalls. Then the throttle increases and the transmission gear downshifts (every other step) until engine rpm becomes positive and the vehicle recovers from stall.

Step [tic]	Throttle... DieselE... rpm_m...			Transmission			Vehicle...		TransmissionController		DriveTr...		terrain...		slopeM...		Rewards			
	Action #	throttle	DieselE...	rpm	Transm...	Transm...	Transm...	position	Transm...	Transm...	ctr	DriveTr...	theNext...	theNow...	"engine..."	"stall"	'mileage'	"vehicle..."		
224	30	0	1701	0	0	0	92	0	4	0	0	0	1	1	1	0	5.12621	876.582		
225	30	0	1701	0	0	0	97	0	4	1	0	0	0	1	1	0	5.12621	876.582		
226	30	0	1701	0	0	0	2	0	4	0	0	0	2	0	0	1	0	0		
227	35	0	0	0	0	0	2	0	4	1	0	2	0	0	1	0	0	0		
228	40	0	0	0	0	0	2	0	3	0	0	2	0	0	1	0	0	0		
229	45	0	0	0	0	0	2	0	3	1	0	2	0	0	1	0	0	0		
230	50	0	0	0	0	0	2	0	2	0	0	2	0	0	1	0	0	0		
231	55	0	0	0	0	0	2	0	2	1	0	2	0	0	1	0	0	0		
232	60	0	0	0	0	0	2	0	1	0	0	2	0	0	1	0	0	0		
233	65	0	0	0	0	0	2	0	1	1	0	2	0	0	1	0	0	0		
234	70	0	0	0	0	0	2	0	1	0	0	2	0	0	1	0	0	0		
235	75	0	0	0	0	0	2	0	1	1	0	2	0	0	1	0	0	0		
236	80	0	377	0	0	0	2	0	1	0	0	2	0	0	0	0	0.45928	78.5369		
237	80	0	977	0	0	0	3	0	1	1	0	2	0	0	0	0	0.45928	78.5369		
238	80	0	977	0	0	0	4	0	1	0	0	2	0	0	0	0	0.45928	78.5369		
239	80	0	977	0	0	0	5	0	1	1	0	2	0	0	0	0	0.45928	78.5369		

Figure 15 – PRISM Simulation for Vehicle Model Responding to Slope Increase (Downhill to Steep Uphill)

3 PRISM Terrain Modeler

The terrain modeler is a tool that automates the generation of probabilistic terrain models for PRISM from terrain specifications provided by the user. Terrain model specifications take the form of specialized PRISM comments that can be embedded in valid PRISM models. Running the terrain modeling tool on a PRISM model containing a terrain specification will result in a file containing the original PRISM model updated with an instance of the specified terrain model.

3.1 The Terrain Model

For the purpose of characterizing the terrain context for a ground vehicle, we employ a mathematical model that expresses the likelihood of traversing from one elevation to another. The tool supports two different terrain models, Kuchar and Max-Entropy.

3.1.1 Kuchar

The Kuchar model was derived in [1]. In that paper, Kuchar describes a Markov chain model for expressing probabilities of given changes in elevation. The probability of transitioning, from step n to step $n + 1$, from a starting altitude bin y_n to altitude bin y_{n+1} is given by Kuchar's equation (10):

$$p_{y_{n+1}, y_n}(n) = \int_{y_{n+1}-h/2}^{y_{n+1}+h/2} \frac{1}{\sqrt{2\pi\sigma^2(1-e^{-2\beta})}} \times e^{\frac{-(y-e^{-\beta}y_n)^2}{2\sigma^2(1-e^{-2\beta})}} dy$$

where h is the height of an altitude bin, and σ and β are autocorrelation parameters fitted to a database of actual terrain altitude samples obtained from the U.S. Geological Survey for the Great Plains and Rocky Mountain region between 102 and 112 degrees West longitude and 32 and 49 degrees North latitude. The autocorrelation parameters vary for different terrain types. Kuchar provides a table (Table 2) of values for the parameters for terrain categories of Smooth, Moderately smooth, Moderate, Moderately steep, and Steep.

The integrand in equation (10) is a probability density function (pdf) for a random variable that is normally distributed with mean $e^{-\beta}y_n$ and standard deviation $\sigma\sqrt{1-e^{-2\beta}}$.

3.1.2 Max Entropy

Essentially, the maximum entropy principle states that, when making inferences about a probability distribution based on incomplete information, one should arrive at the probability distribution that has maximum entropy among those permitted by the observed information. Jaynes [4] explains the motivation for this principle by showing that the permissible distributions are strongly concentrated near the one having maximum entropy – in other words, “distributions with appreciably lower entropy than the maximum permitted by our data are atypical of those allowed by the data.” For terrain elevation, it is clear that the absolute value of a change in elevation must be nonnegative; furthermore it is reasonable to assume we can obtain data on average elevation changes for a given region. For a

nonnegative random variable with a given expected value, the maximum entropy distribution is exponential:

$$p(z|F) = \frac{1}{F} e^{-z/F}$$

As before, we can derive the maximum likelihood estimate for F as follows. For ease of calculation we will work with the log likelihood, which, for a set of observations $z_i, i = 1, \dots, n$ is

$$\ln L(F|z) = \sum_{i=1}^n \ln \left(\frac{1}{F} e^{-z_i/F} \right) = \sum_{i=1}^n \left[\ln \left(\frac{1}{F} \right) - \frac{z_i}{F} \right]$$

Differentiating this with respect to F , and setting the derivative equal to zero implies

$$F = \frac{\sum_{i=1}^n z_i}{n}.$$

Given this model, with the relevant range of altitudes divided into bins of height h , the probability of an elevation change from the current bin to one k bins higher is given by

$$\frac{1}{2} \int_{(k-\frac{1}{2})h}^{(k+\frac{1}{2})h} \frac{1}{F} e^{-z/F} dz = \frac{1}{2} \left(e^{-(k-\frac{1}{2})h/F} - e^{-(k+\frac{1}{2})h/F} \right).$$

An analogous expression holds for the probability of moving down by k bins.

If the modelType parameter is set to maxEntropy and no F value is provided, the tool will compute F from the given (or default) Kuchar parameters in the following manner:

$$F = \sqrt{\frac{\sigma^2(1-e^{-2\beta})}{2}}.$$

3.2 Terrain Specifications

Conceptually the generated terrain model has one input (a Boolean value that indicates when the elevation may change) and two outputs (the current elevation and the current slope). Unfortunately, due to the lack of lexical scoping in PRISM, the user may need to provide suitable names for the inputs, outputs and state variables that constitute the terrain model to avoid name conflicts in the rest of their model.

Terrain specifications must be contained within a PRISM module named “terrainModel”:

```
module terrainModel
// .. Terrain Specification ..
endmodule
```

Terrain parameters are specified as whitespace delimited comments of the form:

```
// {keyword} {value}
```

Any text appearing after the {value} and before the next newline is ignored. The valid keywords, their role, types and default values and brief descriptions are provided in the following table.

Keyword	Role	Type	Default Value	Description
modelType	Model Selection	Either Kuchar or maxEntropy	Kuchar	Selects Kuchar or Max Entropy Model
F	Parameter	Double	0.0	Exponent in Max Entropy model
Beta	Parameter	Double	1.3e-3	Beta parameter in Kuchar model
Sigma	Parameter	Double	342.0	Sigma parameter in Kuchar model
maxElevation	Parameter	Double	100.0	Maximum elevation value
minElevation	Parameter	Double	0.0	Minimum elevation value
binCount	Parameter	Integer	10	Number of discrete elevations
tickName	Clock	Symbol	(empty)	Name of synchronizing event
currBinName	Internal State	Symbol (Integer)	currentElevationBin	Current discrete elevation value
nextBinName	Internal State	Symbol (Integer)	nextElevationBin	Next discrete elevation value
currElevationName	Output Formula	Symbol (Double)	currElevation	Current elevation
nextElevationName	Internal Formula	Symbol (Double)	nextElevation	Next elevation
currSlopeName	Output Formula	Symbol (Double)	currSlope	Current slope
updateElevationName	Input	Symbol (Boolean)	updateElevation	Enables transition from current to next elevation

Below is an example terrain specification:

```

module terrainModule
// modelType           Kuchar      Model Type
// F                  3.0        Max Entropy Exponent (unused)
// Beta               1.3e-3    Model parameter
// Sigma              342.0     Model parameter
// maxElevation       200.0     Upper bound on elevation
// minElevation       -200.0    Lower bound on elevation
// binCount            5          How many different elevations
// tickName            tick       Process synchronization
// currBinName         theNowBin slopeModule state
// nextBinName         theNextBin terrainModule state
// currElevationName   currentElevation current elevation signal
// nextElevationName   nextElevation    next elevation signal
// currSlope            currentSlope   slope signal
// updateElevationName stepElevation Name of the "step" signal
endmodule

```

Kuchar provides the following table to aide in choosing Sigma/Beta values based on a characterization of the terrain. Our default terrain model is classified as “moderate”.

Terrain Description	Sigma	Beta
Smooth	79.0	2.2e-3
Moderately Smooth	269.0	6.4e-4
Moderate (default)	342.0	1.3e-3
Moderately Steep	415.0	2.0e-3
Steep	1007.0	6.1e-4

3.3 Running the Tool

The tool is provided as a batch file called Terrain.bat (in Windows) or a shell script Terrain.sh (in Unix). Running the tool (with no arguments) opens a file chooser that allows the user to select the input file and the output destination. It is acceptable to replace the input file with the generated output.

The tool will remove any existing (previously generated) terrain model artifacts and replace them with the new terrain model. The artifacts removed by the tool include any code inside the body of the “terrainModule” (with the exception of the terrain specification comments), the entire “slopeModule”, and any other line outside of said modules preceded by the comment:

```
/// Generated Terrain Formula
```

This purging feature allows users to change the terrain specification, update the model, and analyze the results in an iterative fashion without undue editing. It also allows a user to develop a simple, initial PRISM model that parses without error in the PRISM tool and gets replaced in whole during the model generation process.

The tool generates a new body for the “terrainModule” and “slopeModule” as well as formulas for current elevation, next elevation, and current slope, each named by default or according to the terrain specification provided by the user.

The tool output for the example module described above is as follows:

```

module terrainModule
// modelType           Kuchar          Model Type
// F                  3.0            Max Entropy Exponent (unused)
// Beta               1.3e-3        Model parameter
// Sigma              342.0         Model parameter
// maxElevation       200.0         Upper bound on elevation
// minElevation       -200.0        Lower bound on elevation
// binCount            5              How many different elevations
// tickName            tick            Process synchronization
// currBinName         theNowBin      slopeModule state
// nextBinName         theNextBin     terrainModule state
// currElevationName   currentElevation current elevation signal
// nextElevationName   nextElevation   next elevation signal
// currSlope            currentSlope    slope signal
// updateElevationName stepElevation  Name of the "step" signal
theNextBin: [0..4];
[tick] ((stepElevation) &
        (nextElevation <= -120.0)) ->
        0.0112: (theNextBin' = 1) + // -80.0
        0.9888: (theNextBin' = 0); // -160.0
[tick] ((stepElevation) &
        (nextElevation > -120.0) &
        (nextElevation <= -40.0)) ->
        0.0107: (theNextBin' = 0) + // -160.0
        0.0110: (theNextBin' = 2) + // 0.0
        0.9783: (theNextBin' = 1); // -80.0
[tick] ((stepElevation) &
        (nextElevation > -40.0) &
        (nextElevation <= 40.0)) ->
        0.0109: (theNextBin' = 1) + // -80.0
        0.0109: (theNextBin' = 3) + // 80.0
        0.9782: (theNextBin' = 2); // 0.0
[tick] ((stepElevation) &
        (nextElevation > 40.0) &
        (nextElevation <= 120.0)) ->
        0.0110: (theNextBin' = 2) + // 0.0
        0.0107: (theNextBin' = 4) + // 160.0
        0.9783: (theNextBin' = 3); // 80.0
[tick] ((stepElevation) &
        (nextElevation > 120.0)) ->
        0.0112: (theNextBin' = 3) + // 80.0
        0.9888: (theNextBin' = 4); // 160.0
[tick] (! stepElevation) -> true;
endmodule
module slopeModule
theNowBin: [0..4];
[tick] ( stepElevation) -> (theNowBin' = theNextBin);
[tick] (! stepElevation) -> true;
endmodule
/// Generated Terrain Formula
formula currentElevation = -160.0 + 80.0*theNowBin;
/// Generated Terrain Formula
formula nextElevation = -160.0 + 80.0*theNextBin;
/// Generated Terrain Formula
formula currentSlope = nextElevation - currentElevation;

```

4 Assume/Guarantee Analysis

Modern reasoning techniques and advances in computing technology have enabled tools capable of amazing feats of analysis. However, even the most capable tools are still limited by the size of the systems they can analyze. Compositional reasoning is a structured approach to verification that enables the analysis of systems whose size would otherwise overwhelm even modern analysis tools. Compositional reasoning enables the analysis of large systems of components by first decomposing the overall problem into several smaller and more tractable problems, solving those problems, and then combining the individual results to formulate a solution for the original problem.

Assume/guarantee reasoning is a compositional methodology that works well with a hierarchical design approach. The methodology relies on component contracts, which are assumptions on the component inputs that guarantee certain component behaviors. At each level of component hierarchy, analysis is limited to ensuring that the assumptions on the component inputs and the contracts of each of its sub-components are, together, sufficient to guarantee the desired component behaviors. Such correct by construction techniques minimize reasoning obligations and allow large, analyzable systems to be built from smaller, verified components.

4.1 Contracts

The assume/guarantee methodology requires the use of mathematically sound contracts. Contracts are a style of formal specification that enable one to capture, not only the expected component behavior, but the assumptions necessary in order to guarantee that behavior. To the component, the contract is an obligation. The component must (is obligated to) satisfy the contractual guarantees whenever the assumptions are met. To a system that employs the component, the contract is a guarantee. It can depend upon the component's contract (if the assumptions of that contract are satisfied), and it can, in turn, use the guarantees of that contract to satisfy its own contractual obligations. Contracts can be expressed mathematically as an implication: the assumptions imply the guarantee. Compositional reasoning follows from the compositional property of implication, which is to say, if A implies B and B implies C, then A implies C. The most crucial and easily overlooked step in this chain of reasoning is that that guarantee of the first contract must match (or satisfy) the assumption of the subsequent contract.

4.2 Typical Probabilistic Contracts

The statistical analysis of a component requires that certain assumptions be made about the component inputs and certain measurements be made on the component outputs. These assumptions and measurements (guarantees) constitute the component contract. Recall that the most crucial step in compositional reasoning is that the guarantee of one contract must match the assumption of another contract. Or, in its simplest form: a contract must guarantee the same sorts of things that it assumes.

When using tools such as PRISM, it is easy to express assumptions on inputs as probability distributions or Markov chains and it is natural to express properties as probabilities that certain events take place. The issue with such analysis is that the resulting contract is not compositional. In particular, the conclusion (that some event takes place with some probability) does not guarantee the same sorts of things that it assumes (that the inputs obey some given probability distribution or Markov relation).

Consequently, while typical probabilistic contracts may be quite useful for understanding various statistical behaviors of components, such contracts are of little use as part of an assume/guarantee methodology.

4.3 Compositional Probabilistic Contracts

There is a style of specification in PRISM that does work in a compositional assume/guarantee methodology. It requires the use of what are called Markov Decision Process, or MDP, models. In such models, the behavior of the input is essentially unconstrained. In each step of the model, an input may choose freely from any element of its domain. Assumptions in such models are expressed as simple probabilistic assertions about the behavior of the input (ie: the expected value of the input is 0.4) and properties are expressed as probabilistic assertions about the values of the outputs (i.e., the expected value of the output is 0.7). Such contracts are appealing in that they make the same sorts of assertions about the inputs as they do about the outputs. In other words, they are compositional and thus useful in an assume/guarantee methodology.

In order to verify such properties, PRISM must consider every possible sequence of inputs that satisfies the assumption (every sequence in which the expected value of the input is 0.4) and verify that the guarantee holds for every such sequence. If PRISM finds an input sequence for which the guarantee does not hold, the verification fails. Otherwise, the verification succeeds. Such analysis is called “adversarial analysis” because PRISM is searching for an adversary (a specific sequence of inputs) that causes the property to fail. If it cannot find such an adversary, it must concede that no such adversary exists and, thus, the contract will always hold.

4.4 Prototype Adversarial Assume/Guarantee Reasoning (AAGR) Framework

The PRISM tool supports adversarial analysis of MDP models of games: systems that have well defined end states. We are interested in reactive systems, such as vehicles operating over terrains, which do not have any particular end state. This lack of compositional reasoning support for reactive systems has compelled us to explore how such a reasoning system might be constructed.

The greatest challenge in the adversarial analysis of reactive systems is the size of the decision space. Recall that the adversarial analysis procedure must consider every possible sequence of inputs that satisfies the assumption. This means that the procedure must consider every possible probability assignment to every possible input state for every possible machine state. Considering that the size of the state space is exponential in the number of bits in the machine, it follows that the size of this decision space is doubly

exponential in the number of bits. For systems of any reasonable size, it would be impossible to enumerate all possible decisions before the heat death of the universe.

Previous research into adversarial analysis of games employed linear optimization to find the best adversary for a given system [6]. In other words, rather than considering all possible adversaries, it considered only the best (most malicious) adversaries. If no such adversary violates the contractual guarantee, the system must always satisfy the contract.

We attempt to leverage concepts from this previous work analyzing games and extend it to reasoning about reactive systems. The primary challenge with reactive systems is that properties are expressed in terms of expected values in the steady state, but the steady state of the machine depends upon the current decision. We were able to overcome this challenge by encoding the steady state of the system as an additional obligation to the linear optimization procedure. This solution provides a fast and effective decision procedure for deciding probabilistic assume/guarantee properties about reactive systems. In our experiments to date, even the largest problem we attempted was solved in only a few seconds.

4.5 The AAGR Prototype

In the AAGR prototype, component behavior is expressed in PRISM, properties are expressed in an Eclipse-based domain specific language called PrSL, and the adversarial analysis algorithm is implemented in MATLAB. We assume that the user is already familiar with model creation in the PRISM [8] probabilistic model checker.

4.5.1 Eclipse Workspace

Effective use of the AAGR framework requires that development be performed in an Eclipse workspace. This requirement is compatible with both PRISM and MATLAB and maximizes the utility of the PrSL tool. A “ready to use” example workspace containing four example projects has been provided with the AAGR distribution. When the tool is first started, it will query the user as to which workspace to use. Selecting the ready to use workspace provided with this distribution will allow for a quick evaluation of tool capabilities.

4.5.2 Probability Specification Language (PrSL)

The Probability Specification Language (PrSL) is a domain specific language for formulating properties to be checked by the AAGR tool. At the heart of it is an Eclipse-based editor that provides users with a modern specification environment complete with dynamic checking for syntax and well-formedness. In addition it provides a set of translations that tie behavioral descriptions written in PRISM with specifications written in PrSL and, ultimately, combines them both into assertions expressed in MATLAB.

PrSL specifications are composed of several sections. The interface section provides descriptions of the component’s various PRISM model principles, such as inputs, states and rewards. The equations section allows the user to construct descriptive shorthand notations for common specification expressions. Finally, the properties and refutation sections contain contract assertions that the framework will attempt to either verify or refute.

4.5.3 PrSL Interface Definition

The interface section is used to describe each principal found in the model. The interface section begins with the keyword “Interface” and contains a list of declarations that describe whether the principal is an input, state, constant or reward.

Each item in the interface list must contain an identifier that denotes the name of the principal, the type of principal (input, state, constant, or reward), and the position of this principal in the MATLAB reward vector. The following example shows an example of one of each type of principal.

```
Interface
  in      : input    @ 1;
  pin     : state    @ 2;
  pconst  : const    @ 3;
  reward1 : reward   @ 4;
```

During development, users are likely to add and remove inputs, states and rewards in their PRISM models. Also, the user is unlikely to know (or care) about the location of a given principal in the underlying MATLAB model. Consequently, users are discouraged from writing interface specifications by hand. Rather, PrSL provides a translation path that takes a PRISM model and generates an interface file. For a PRISM file named “foo.pm” the interface file will be named “fooInterface.prsl”. This machine generated interface file can then be imported into a hand-written specification, insulating the specification from changes in the PRISM model. Importing can be accomplished by using an import declaration. The following example illustrates the use of the import declaration.

```
Interface
  import "derivativeInterface.prsl";
```

4.5.4 PrSL Equations, Properties, and Refutations

Three sections are provided for users to specify properties of interest. The format of expressions in each of these sections is the same, but the manner in which expressions are interpreted in each section is different. The Equations section is a place for the user to place macros and other expressions that are frequently used. Any equation defined in this section can be referenced in expressions in the Properties and Refutations section. The Properties section is where the user places properties to be verified. Properties can refer directly to previously defined equations, or contain expressions that describe the property of interest. The Refutations section is where the user places properties expected to be false under some circumstances. Refutations and Properties are specified in the exact same syntax. Refutations, however, will be translated into MATLAB expressions that succeed when a counterexample is identified, rather than expressions that succeed when no such counterexample exists.

Here is an example of a small PrSL specification that employs each of these sections:

```
Interface
  import "derivativeInterface.prsl";

  Equations
```

```

e1 = P pslope > 0;
e2 = P pslope > 0 & P pslope < 0.5 ;

Refutations
r1 = E pslope < 0;

Properties
p1 = E pslope < 3;

```

4.5.5 PrSL Expressions

The user may reason about principals using either an expected value operator (E) or, if the principal is Boolean, the probability operator (P). Principals may appear only in the context of one of these two operators. Note that expected value here means steady state (or long term) expected value. The expected value and probability operators may then be compared to constant bounds using any of the standard relational operations (<, <=, >, >=, ==, !=), resulting in a Boolean expression. The relational operators may also be used in ternary expressions, allowing the user to specify both upper and lower bounds on the expected value of a single principal (ie: $1.0 < E x < 2.0$). Such Boolean expressions may then be combined using standard logical operations to construct properties. The and (&), or (+), not (!), implies (->) and xor (#) logical operations are supported. The example below demonstrates the expression features of the language.

```

Interface
import "derivativeInterface.prsl";

Equations
e1 = P pslope > 0;
e2 = P pslope > 0 & P pslope < 0.5 ;
e3 = P reward1 < 0.85;

Refutations
r1 = e1;
r2 = E in < 2.3 ;
r3 = P in < 0.75 & P in < 0.9;

Properties
p1 = E pslope < 3;
p2 = P in > 0.75 + P in < 0.25 ;
p3 = e1 -> e2 ;
p4 = e1 # e3 ;

```

Validation passes are performed automatically to indicate to the user when an expression is incorrect or unsupported. For example, all relational operations must only compare a principal probability expression to a literal constant. If the user specifies something outside of that format, the tool will yield a graphical error and suggest to the user how to fix it. Shown below is an example of a few expressions that are not allowed.

```

Interface
import "derivativeInterface.prsl";

Equations
e1 = P pslope > P in;

```

```

e2 = pslope < 2 ;
e3 = e1 < e2 ;

Properties
r1 = e1;

```

4.5.6 MATLAB Output

For a given project, the final outputs of the PrSL tool are two MATLAB script files. The first script is the model file. It defines, among other things, the system transition matrix. This file is generated, along with the PrSL interface file, from the PRISM specification. For a PRISM file named “foo.pm” the model script will be named “fooModel.m”. The second script contains property assertions about the model. This script is generated from the PrSL specification. For a PrSL file named “foo.prsl”, the assertion script will be named “foo.m”. Performing an analysis involves first executing the model script and then executing the assertions script. The results of the analysis will then appear in the MATLAB console.

4.6 System Requirements

AAGR has been shown to work with the following software/systems:

- MATLAB 8.0.0.783 (R2012b) with Optimization Toolbox under Linux-64
- MATLAB 7.10.0.499 (R2010a) with Optimization Toolbox under WinXP-32

The PrSL executable was built using:

- Java 7
- Eclipse 4.2.1
- XText 2.3.1

In addition, PRISM 4.0.3 is assumed to be installed in the standard location, C:\Program Files\prism-4.0.3\.

4.7 Analyzing an AAGR Example

The distributed workspace provides four example projects. In this section, we focus on the /correlation project. This project comes with a PRISM module and a PrSL specification. In this section, we will give some background on this example and then walk thru the steps required to analyze it. The steps used to analyze this example are similar to the steps required to analyze any of the distributed examples.

4.8 The correlation model

The correlation model describes a system that detects temporal correlations in a binary input. Following is a PRISM description of its behavior.

```

module Correlation
s:[0..3];
[tic] (s=0) -> (s'=(input ? 1 : 0));
[tic] (s=1) -> (s'=(input ? 0 : 2));
[tic] (s=2) -> (s'=(input ? 3 : 0));

```

```

[tic] (s=3) -> (s' = (input ? 0 : 2));
endmodule

formula alarm = ((s=2)/(s=3));

rewards "Alarm"
alarm: 1;
endrewards

```

This system starts out in state 0, waiting for a true input. When it sees a true input it transitions to state 1. At this point, every time it sees an input that differs from the previous input, it advances (from 1 to 2, 2 to 3 and 3 to 4) until it reaches state 4 at which point it will oscillate between states 4 and 3 as long as the input keeps changing. If the machine ever sees the same input twice in a row, it returns to state 0. Note that we have defined a reward called “Alarm” that detects when the machine is in states 2 or 3. Our objective is to analyze the behavior of this system when the input is true 50% of the time.

First, let’s consider how we might ensure that our input is true 50% of the time. One possibility is to draw the input from a uniform probability distribution. The following module constrains the input in this manner.

```

module UniformInputModule
    input: bool;
    [tick] true -> 0.5:(input' = true) +
                    0.5:(input' = false);
Endmodule

```

Now can we prove something about the expected value of the Alarm under this assumption? Can we prove, for example, that it is less than 0.5? In PRISM we would express that steady state property as follows:

$$R\{ \text{"Alarm"} \} \leq 0.5 \text{ [S]}$$

Running the analysis in PRISM confirms this assertion. In fact, the computed probability of failure for this system is about .25 which is actually much better than .5. But now consider this: does this analysis prove that, any time the input is true with probability .5, the system will satisfy the contract? The answer is no. In particular, the sequence [T F T F T F T F ..] (which is true with probability .5) will fail with probability 1.0. Conversely, the sequence [F T T F F T T ..] fails with probability 0.0.

Unfortunately, the uniform distribution assumption that the input is true with probability 0.5 is not sufficient to guarantee the desired result: that the probability of failure is always less than 0.5. The problem with assumptions like uniform distributions is that they are too uniform: they don’t really account for temporal correlations in the input. In other words, they do not take into consideration worst case scenarios. This is where adversarial analysis comes in.

In AAGR, we employ PRISM's Markov Decision Models (mdp) to allow us to specify arbitrary inputs and then constrain the behavior of the inputs with explicit assumptions. The input specification for the correlation model now looks like this:

```
formula input = Ainput;
module AdverseInputModule
    Ainput: bool;
    [tic] true -> (Ainput' = true);
    [tic] true -> (Ainput' = false);
Endmodule
```

Then start PrSL, point it to the provided workspace, and open up the correlation project. Clock and drag the correlation.pm module into the editor pane. When the Prism model has focus (is the file being currently edited) the user can select “Translate from Prism” from the Translate menu. Figure 16 is a screenshot showing this action being performed for the file “correlation.pm” from the correlation project in the distributed examples.

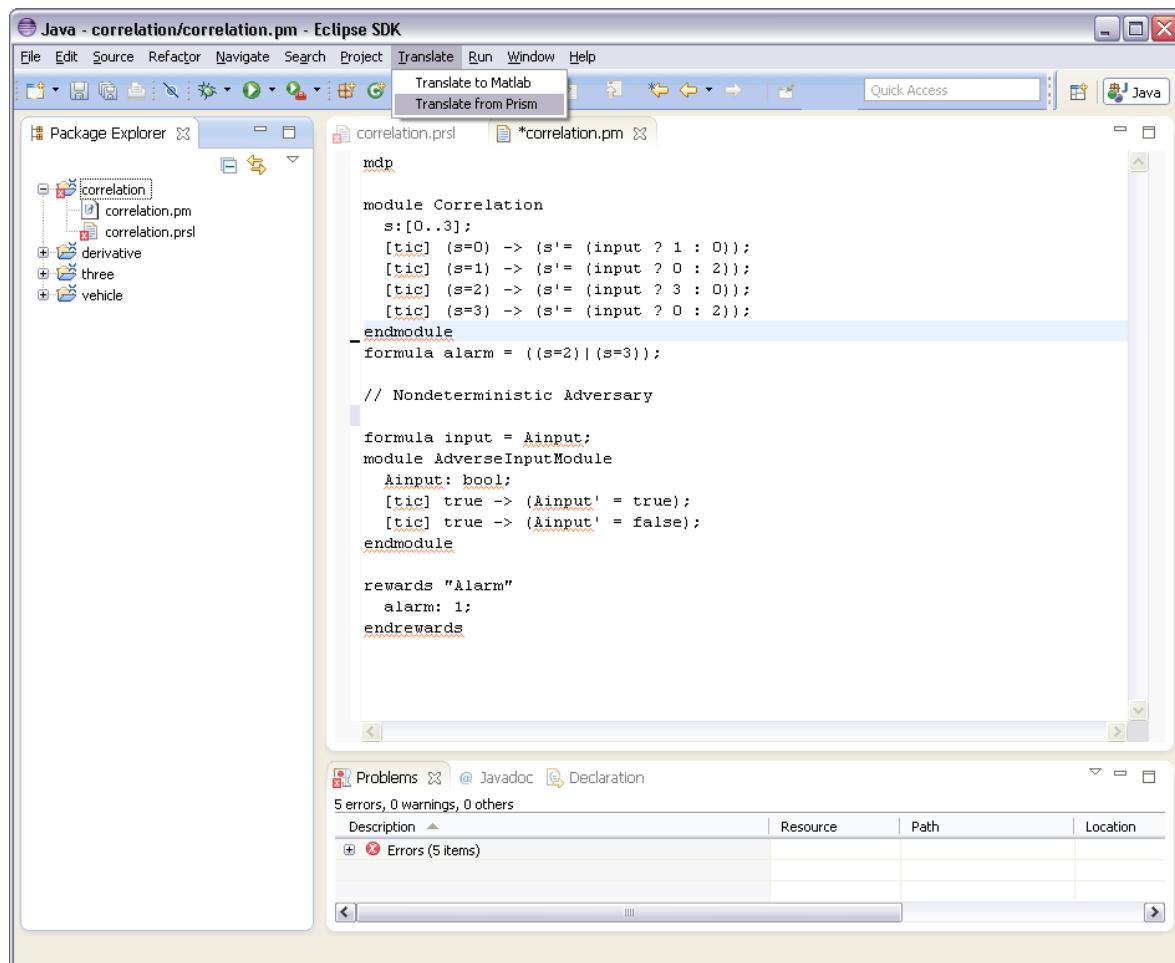


Figure 16 - Translate from PRISM menu

This will result in two new files in the project. The first is the MATLAB script that describes the behavior of the correlation model. The second is an interface file,

“correlationInterface.prsl”, that describes the principals of the correlation model. The interface file has the following contents:

```
Interface
Ainput      : input    @ 2;
s           : state    @ 1;
Alarm       : reward   @ 3;
```

Note that the input (Ainput), state (s) and reward (Alarm) values from the PRISM specification all appear here. They can now be used to express properties about this system. Note, too that the formula (alarm) is not provided here. If one needs to express a property or assumption about a formula, one must tie the formula to a reward.

We can now express our properties of interest using these principals. The provided PrSL file contains the following specifications.

```
Interface
import "correlationInterface.prsl";

Equations

// Property 1
h1 = P Ainput == 0.5;
r1 = P Alarm > 0.0;

// Property 2
h2 = P Ainput == 0.5;
r2 = P Alarm < 1.0;

Refutations

p1 = h1 => r1;
p2 = h2 => r2;
```

We have specified two properties. The first (p1) says that, if the input is true with probability 0.5, the expected value of the alarm will always be greater than 0.0. The second (p2) says that if the input is true with probability 0.5, the expected value of the alarm must be less than 1.0. Both of these properties are expressed as “refutations” because, as it turns out, both are false.

After a specification has been created, the user can generate the Matlab script that will perform the analysis of the user specified properties. This can be done by invoking the “Translate to Matlab” command while editing the PrSL specification of interest. This is shown **Error! Reference source not found.** in Figure 17.

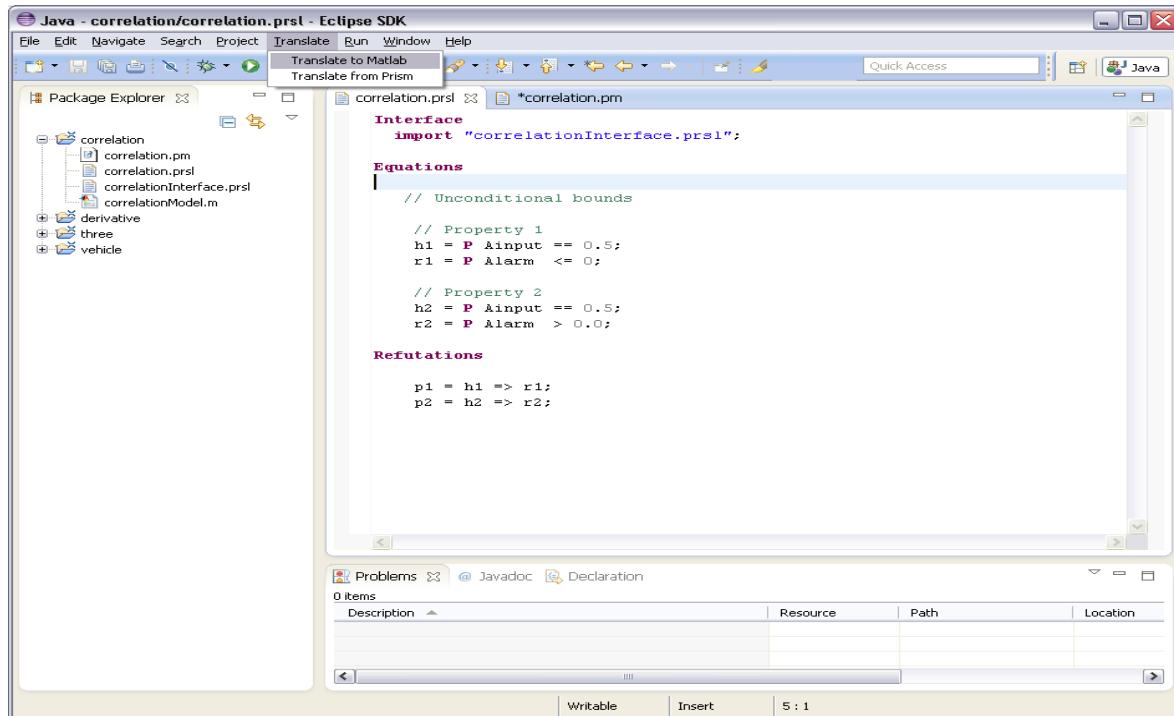


Figure 17 - Translate to MATLAB

This action produced one new file in the workspace, “correlation.m”. This is a MATLAB script that contains the assertions from the PrSL specification encoded as MATLAB commands.

To run the MATLAB scripts, start MATLAB and set the working directory to the correlation project in the distributed workspace. Note that your MATLAB path will need to include the /AAGR-MATLAB-source directory from the AAGR distribution so that MATLAB is able to find the “verify” and “refute” commands. Once the path and working directory have been set, the analysis proceeds by first executing the “correlationModel” script (to load the model into memory) and then executing the “correlation” script (to analyze the properties). Note that in this case, both properties were successfully refuted (Figure 18).

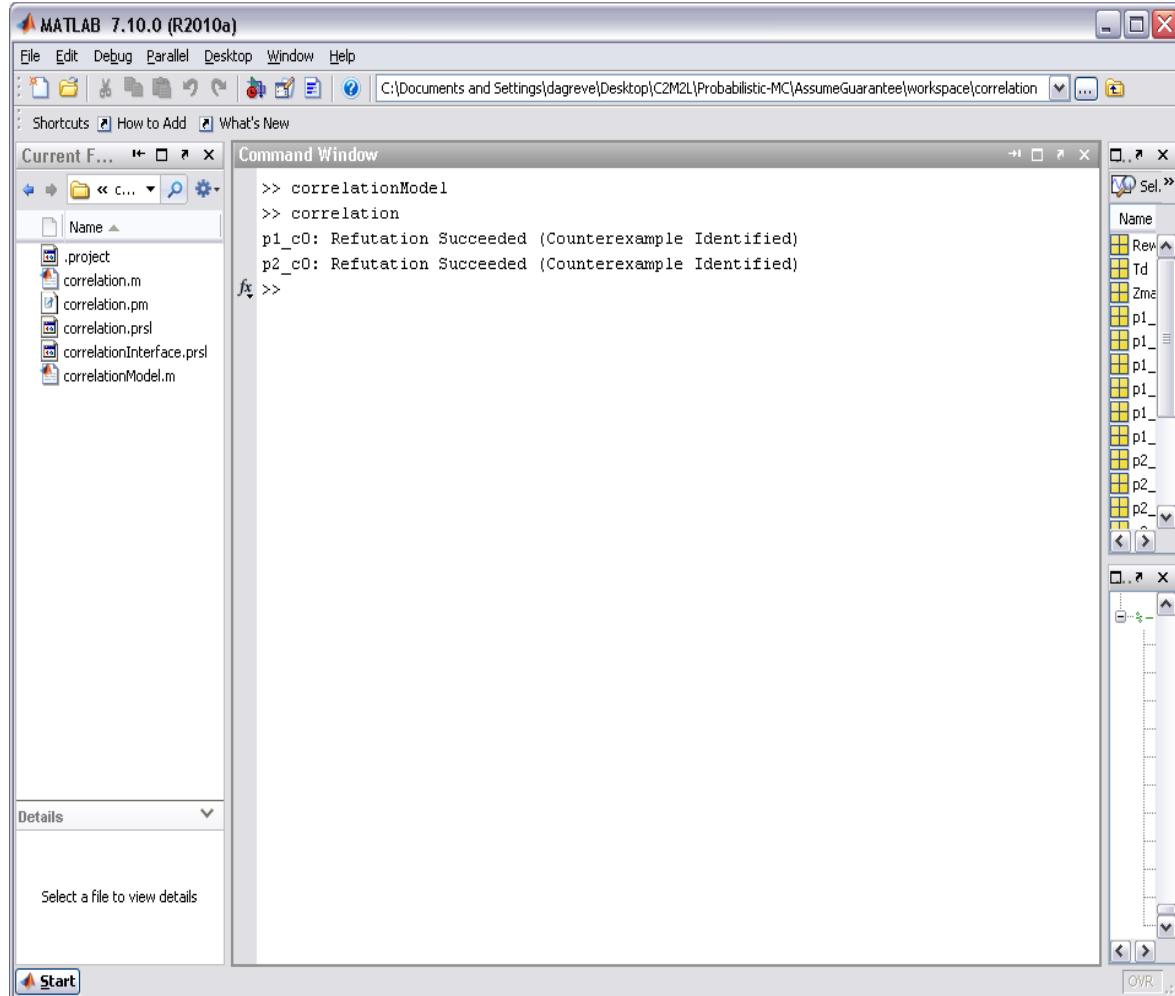


Figure 18 - Analyze example in MATLAB

4.9 Distributed Examples

In the distributed workspace we provide four projects containing different systems that have been used in developing and optimizing our analysis framework.

/correlation

Correlation is a system that detects temporal correlations in a binary input. This example clearly indicates the importance of adversarial analysis in establishing (or refuting) bounds on system behavior.

/three

Three is a somewhat non-linear system with three outputs. This system is intended as a simple regression test for the analysis algorithms. Note that not every clause of the second refutation in this example is falsifiable. Finding a counterexample to any clause of an assertion is enough to falsify the entire assertion. However, because each clause is processed separately, the tool currently reports those refutations as failures.

/derivative

Derivative is a small system that tracks the change in (derivative of) a 4-valued input. This system is intended as a small scale model that mimics some of the behaviors of the slope variable found in the larger vehicle model.

/vehicle

Vehicle is an early snapshot of the vehicle model. This model is particularly useful for identifying scalability issues in the algorithms. In our experience, this model is too big to load into 32-bit MATLAB. We were able to run it in 64-bit mode under Linux.

4.10 Future Work

There are several enhancements that can be made to the tool to make it more user-friendly. The issue of false negatives when processing multiple refutation clauses, for example, should be addressed. There are also some questions about what the most appropriate numerical error bounds should be. The current tool, for example, cannot decide questions with a precision greater than 0.0001. Also, a tighter integration with MATLAB could allow the user to drive the entire process from the Eclipse (PrSL) front end.

The issue of circular composition has vexed mathematicians for some time. Circular composition involves two contracts of the form A implies B and B implies A. The question is: when can these contracts be combined to soundly conclude that A and B are always true. There exist classes of temporal logic properties that admit circular composition [7]. Is there a similar class of probabilistic assertions? While general class of properties would be ideal, it could easily be the case that circular composition will always require an additional set of bootstrapping properties to preserve the soundness of composition.

Finally, an integrated infrastructure is needed to perform compositional verification and leverage assume/guarantee contracts. Such an infrastructure would provide comprehensive system analysis capabilities and enable the rapid evaluation of different design trade-offs and the impact of changes in environmental assumptions.

5 References

- [1] J. K. Kuchar, "Markov Model of Terrain for Evaluation of Ground Proximity Warning System Thresholds," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 3, pp. 428-435, May-June 2001.
- [2] T. Murphy, "100 MPG on Gasoline: Could We Really?," 17 July 2011. [Online]. Available: <http://physics.ucsd.edu/do-the-math/2011/07/100-mpg-on-gasoline/>. [Accessed 14 December 2012].
- [3] M. Ross, "Fuel Efficiency and the Physics of Automobiles," *Contemporary Physics*, vol. 38, no. 6, pp. 381-394, 1997.
- [4] E. T. Jaynes, "On the Rationale of Maximum-Entropy Methods," *Proceedings of the IEEE*, vol. 70, no. 9, p. 939, 1982.
- [5] Henzinger, Thomas A., Marius Minea, and Vinayak Prabhu. "Assume-Guarantee Reasoning." (2001).
- [6] Marta Z. Kwiatkowska, Gethin Norman, David Parker, Hongyang Qu: "Assume-Guarantee Verification for Probabilistic Systems", TACAS 2010: 23-37
- [7] John Rushby, "Formal Verification of McMillan's Compositional Assume-Guarantee Rule", CSL Technical Report, September 2001.
- [8] PRISM, <http://www.prismmodelchecker.org/>

January 28, 2013

Technical Report

Context Model Library for C2M2L-1

Interim Report - EMI Susceptibility via Cables

Prepared for BAE Systems

C2M2L

Subcontract 338724

Technical Point of Contact:

John G. Kraemer PE
Rockwell Collins, Inc.
400 Collins Rd. NE, MS 106-183
Cedar Rapids, IA 52498
Telephone: (319) 295-4681
jgkraeme@rockwellcollins.com

Business Point of Contact:

Ms. Kelly M. Scott
Rockwell Collins, Inc.
400 Collins Rd. NE, MS 121-200
Cedar Rapids, IA 52498
Telephone: (319) 295-7323
kmscott@rockwellcollins.com

Distribution Statement C: Distribution Authorized to U.S Government Agencies and their contractors. Other requests for this document shall be referred to the DARPA Technical Information Office.



Rockwell Collins, Inc.

400 Collins Rd. NE

Cedar Rapids, Iowa 52498

TABLE OF CONTENTS

1. INTRODUCTION.....	7
1.1 SCOPE.....	9
1.2 REFERENCE DOCUMENTS	9
1.3 UNITS OF MEASURE	10
1.4 LIST OF ACRONYMS	10
1.5 DEFINITIONS.....	11
2. THE ELECTROMAGNETIC ENVIRONMENTS.....	14
2.1 SEMANTICS	14
2.2 GROUND VEHICLE SYSTEM ELECTROMAGNETIC ENVIRONMENTS	15
2.2.1 <i>Radiated Radio Frequency (RF) Electromagnetic Environment (RF EME)</i>	15
2.2.2 <i>Electromagnetic Pulse (EMP)</i>	15
2.2.3 <i>Lightning Effects</i>	15
2.2.4 <i>High Power Microwave (HPM)</i>	16
2.2.5 <i>Electrostatic Discharge</i>	16
2.2.6 <i>Internally Generated Electromagnetic Energy</i>	16
2.3 ELECTROMAGNETIC ENVIRONMENT FOR GROUND VEHICLE SUBSYSTEMS AND EQUIPMENT	17
2.3.1 <i>Conducted Susceptibility - RF – CS114 Environment Model</i>	19
2.3.2 <i>Conducted Susceptibility – Bulk Cable Injection, Impulse Excitation – CS115 Environment Model</i>	24
2.3.3 <i>Conducted Susceptibility – Damped Sinusoidal Transients – CS116 Environment Model</i>	28
2.3.4 <i>Radiated Susceptibility- Electric Field - RS103 Environment Model</i>	32
2.3.5 <i>Radiated Susceptibility - Transient Electromagnetic Field - RS105 Environment Model</i>	33
3. CABLE/CONNECTOR ASSEMBLY MODELING METHODOLOGY.....	35
3.1 MODEL ENTITY BLOCKS	35
3.2 GROUNDS AND GROUNDING.....	41
3.2.1 <i>CST Studio Suite Electromagnetic Effects Simulator Overview</i>	41
3.2.2 <i>Grounding within CST Studio Suite</i>	41
3.3 CABLE HARNESS DEFINITION.....	45
3.3.1 <i>Ground Plane</i>	45
3.3.2 <i>Nodes</i>	45
3.3.3 <i>Segments, Traces, and Routes</i>	45
3.3.4 <i>Cables and Cable Groups</i>	46
3.3.5 <i>Cable Bundles</i>	46
3.3.6 <i>Cable Harness</i>	46
3.3.7 <i>Shielded Cable Bundles</i>	46
3.3.8 <i>Signals and Signal Terminals</i>	46
3.3.9 <i>Connectors, Plug-ins, Pins, and Junctions</i>	47
3.3.10 <i>Cable Block I/O</i>	47
3.4 CABLE SHIELD TRANSFER IMPEDANCE	48
4. MULTI-BRANCH CABLE EXAMPLE	54
4.1 FOCUS ON EMI COMPLIANCE.....	54
4.2 EXAMPLE COMPLIANCE TEST SETUP SPECIFICS	57
4.3 DETERMINATION OF SIGNIFICANT FEATURES.....	57
4.3 ENTITY BLOCKS	60
4.4 MODEL INPUT AND ANALYSIS – EXPLANATION BY EXAMPLE.....	66
4.4.1 <i>I/O and Signaling Schematic with Common Ground</i>	66
4.4.2 <i>I/O and Signaling Schematic with Target Grounding System</i>	67
4.4.3 <i>Cable Harness Model Build</i>	68
4.4.4 <i>Combine Cable and Interface Circuit Entity models</i>	69

4.4.5 <i>Addition of EMI Environment Injection</i>	69
4.4.6 <i>Build Models for 3-D Connector, Backshell, and Transition Entity Structures</i>	70
4.4.7 <i>Complete Co-Simulation Integration and Checkout</i>	71
4.4.8 <i>Apply Environment, Probe Victim Circuits and Analyze Results</i>	72
4.4.9. <i>Analysis of the Results</i>	74
4.4.10. <i>Implement Needed Design Changes, Re-Run Simulation</i>	75
5. SUMMARY AND CONCLUSIONS	77
APPENDIX A: STATEMENT OF WORK	122

TABLE OF FIGURES

Figure 2.3-1. Flow-down of System Level Electromagnetic Threats to Equipment, Subsystems and Cables	18
Figure 2.3-2. Origin of Baseline MIL-STD-461F RS103 Requirement Applicable to Ground System Equipment.....	19
Figure 2.3.1-1. CS114 Current Limits	21
Figure 2.3.1-3. MIL-STD-461F Power Line Impedance Stabilization Network (LISN)	22
Figure 2.3.1-4. CS114 Environment Model	23
Figure 2.3.2-1. CS115 Pulse Waveform	25
Figure 2.3.2-2. Compliant Injected CS115 Pulse Waveform into 100 Ohm Loop.....	26
Figure 2.3.2-3. CS115 Environment Model	27
Figure 2.3.3-1. CS116 Waveform.....	30
Figure 2.3.3-2. CS116 Ip Limit	30
Figure 2.3.3-3. CS116 Environment Model	31
Figure 2.3.5-1. RS105 Waveform.....	34
Figure 2.3.5-2. RS105 Setup Example for a Subsystem.....	34
Figure 3.1-1. Model Entity Blocks	36
Figure 3.2.2-1. Grounding Scheme #1 in Co-simulation Environment for Simple 3-Conductor Cable above Ground Plane	43
Figure 3.2.2-2. Grounding Scheme #2 in Co-simulation Environment for Simple 3-Conductor Cable above Ground Plane	44
Figure 3.4-1. Terminology for Braided Shields.....	50
Figure 3.4-2. Screenshot Showing Uses of Kley's Model in Cable Studio	51
Figure 3.4-3. Cable with Combination Foil and Braid Shield	52
Figure 3.4-4. Transfer Impedance of Shields on Various Belden Data Cables Containing Foil	52
Figure 3.4-5. Transfer Impedance of Shields on Popular Coax Cables Utilizing Braid-only Shields	53
Figure 4-1. EMI Test Setup for CS Injection On to Cable 1 at LRU 1	56
Figure 4.2-1. Cable Harness Build Information	58
Figure 4.3-1. I/O Circuits Schematic with Probe Locations	62
Figure 4.3-2. Entity Blocks and Interconnection in Co-simulation Framework.....	63
Figure 4.4.1-1. I/O and Signaling Schematic with Common Ground - CST Design Studio	78
Figure 4.4.1-2. I/O and Signaling Schematic with Common Ground – Results - Gain in Desired Direction	79
Figure 4.4.1-3. I/O and Signaling Schematic with Common Ground – Results - Gain in Reverse Direction	80
Figure 4.4.2-1. I/O and Signaling Schematic with Target System Floating Grounds - CST Design Studio.....	81
Figure 4.4.2-2. I/O and Signaling Schematic with Target System Floating Ground – Results - Gain in Desired Direction.....	82
Figure 4.4.2-3. I/O and Signaling Schematic with Target System Floating Ground – Results - Gain in Reverse Direction	83
Figure 4.4.3-1. Cable Harness Layout and Build – CST Cable Studio	84
Figure 4.4.3-2. Cable Harness Build and Connections – CST Cable Studio.....	85
Figure 4.4.3-4. Cable Harness Signal Transfer Simulation – Desired Path Transfer	87
Figure 4.3-5. Cable Harness Signal Transfer Simulation – Cross-talk.....	88
Figure 4.4.3-6. Cable Harness Signal Transfer Simulation – Coupling From Overshield	89
Figure 4.4.4-1. Cable Harness and Interface Circuits Imported in to Co-Simulation Framework for Transfer Checkout	90
Figure 4.4.4-3. Cable Harness with Interface Circuits Signal Transfer Simulation – Cross-talk	92

Figure 4.4.5-1. Cable Harness, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout.....	93
Figure 4.4.5-2. Cable Harness, Interface Circuits and CS114 Environment Signal Transfer Simulation – Desired Path Transfer.....	94
Figure 4.4.5-3. Cable Harness, Interface Circuits and CS114 Environment Signal Transfer Simulation – Coupling From Overshield.....	95
Figure 4.4.6-1. LRU 1 Connectors and Transition 3D Model - CST Microwave Studio.....	96
Figure 4.4.6-2. LRU 1 Connectors/Transition Block in Co-Simulation Framework for Transfer Checkout.....	97
Figure 4.4.6-3. LRU 1 Connectors/Transition Block – Transfer through Desired Signal Paths	98
Figure 4.4.6-4. LRU 2 Connectors and Transition 3D Model - CST Microwave Studio.....	99
Figure 4.4.6-5. LRU 2 Connectors/Transition Block in Co-Simulation Framework for Transfer Checkout.....	100
Figure 4.4.6-6. LRU 2 Connectors/Transition Block – Transfer through Desired Signal Paths ...	101
Figure 4.4.7-1. Cable Harness, Connectors, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout - left side.....	102
Figure 4.4.7-2. Cable Harness, Connectors, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout – right side.....	103
Figure 4.4.7-3. Cable Harness, Connectors, Interface Circuits and CS114 Environment Signal Transfer Simulation – Desired Path Transfer	104
Figure 4.4.7-4. Cable Harness, Connectors, Interface Circuits and CS114 Environment Signal Transfer Simulation – Environment Transfer through Overshield.....	105
Figure 4.4.8-1. Complete System in Co-Simulation Framework for CS114 Simulation – left side schematic	106
Figure 4.4.8-2. Complete System in Co-Simulation Framework for CS114 Simulation – right side schematic	107
Figure 4.4.8-3. Complete CS114 Simulation – Results: Raw Calibration Current and Shield Currents	108
Figure 4.4.8-4. Complete CS114 Simulation – Results: Raw Interference Probe Voltages.....	109
Figure 4.4.8-5. Complete CS114 Simulation – Results: Computed Scaling Factors Per MIL-STD-461F Methods.....	110
Figure 4.4.8-7. Complete CS114 Simulation – Results: Corrected Victim Net Currents.....	112
Figure 4.4.8-8. Complete CS114 Simulation – Results: Corrected Victim Circuit Interference Voltages	113
Figure 4.4.9-1. CS114 Simulation Results Analysis - Interference Prediction for Digital Video Circuit	114
Figure 4.4.9-2. CS114 Simulation Results Analysis - Interference Prediction for Analog Video Circuit	115
Figure 4.4.9-3. CS114 Simulation Results Analysis - Interference Prediction for Ethernet Circuits	116
Figure 4.4.9-4. CS114 Simulation Results Analysis - Interference Prediction for Power Circuit.	117
Figure 4.1.10-1. Complete System in Co-Simulation Framework for CS114 Simulation w/ EMC Improvements – left side schematic	118
Figure 4.4.10-2. Complete System in Co-Simulation Framework for CS114 Simulation w/ EMC Improvements – right side schematic	119
Figure 4.4.10-3. CS114 Simulation Results Analysis - Simulation w/ EMC Improvements – Analog Video Circuit.....	120
Figure 4.4.10-4. CS114 Simulation Results Analysis - Simulation w/ EMC Improvements – Power Circuit	121

TABLE OF TABLES

Table 1.3-1. Units of Measure	10
Table 3.1-1. Description and Specification of Model Entity Blocks.....	37
Table 3.4-1. Simple Transfer Impedance Model Parameters for Popular Braid Coax Shields	48
Table 4.3-1. Entity Block Description and Specification for Example Model	64

Context Model Library – EMI Susceptibility via Cables

Abstract: *This report presents a set of modeling methods and general context models for predicting susceptibility of electronic equipment to MIL-STD-461F environments with respect to reception of interference via the cable/connector assemblies. The primary focus of the resulting methods and models is to support the DARPA Component, Context, and Manufacturing Model Library (C2M2L) effort to reduce the design and development cycle time of military ground vehicles. The report covers three areas – System Electromagnetic Environment Flowdown to Equipment and Cable/Connector Assemblies, Modeling Method for Cable/Connector Assemblies, and Detailed Example.*

1. INTRODUCTION

Electronic subsystems and equipment in military ground vehicles are subject to Electromagnetic Interference (EMI) from sources internal and external to the vehicle via pick-up and subsequent conduction through the cables and wires attached to the equipment. The general external electromagnetic environment presented to a military system (aircraft, ship, ground vehicle) is defined in MIL-STD-464C, “*Electromagnetic Environment Effects Requirements for Systems*.“ In the end, the system level electromagnetic environment presents itself at the equipment connector pins as a voltage or current with an equivalent source impedance to result in potential upset of the electrical interface and/or damage to electronic components, unless adequate protection is part of the equipment and/or cable-connector assembly design. The interference voltage or current may be a transient (typically described in the time domain), or it may be a somewhat steady-state condition at a particular frequency.

The typical electromagnetic interference environments that are presented to cable-connector assemblies for equipment used in military vehicles are defined in standards such as MIL-STD-461F, “*Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment*.“ MIL-STD-464C Section 5.7 specifically calls out the interference control requirements of MIL-STD-461 for individual subsystems and equipment.

The requirements in MIL-STD-461F represent the electromagnetic environment expected of subsystems and equipment, and the associated cables, with adjustments made for standard laboratory test setups for compliance verification. MIL-STD-461F contains various “limits” for each type of electromagnetic environment based on the type of platform (system) and the location of the subsystem or equipment and its associated cabling within the system. MIL-STD-461F also contains the compliance verification test methods for each electromagnetic environment. A key requirement is to verify compliance with cables that are representative of the intended installation. Section 2 of this report thus addresses the external electromagnetic environments and presents the environment models, as applicable to cable assemblies, in detail.

Typically, the design and analysis for EMI compliance of equipment and subsystems falls into two areas – the equipment items themselves and the cable/connector assemblies.

Design of the equipment for EMI compliance centers around designing circuits, both electrically and mechanically (e.g. printed circuit board layout) for low emissions and a required level of immunity, and

designing chassis shielding to provide electromagnetic attenuation at least as great as the difference between a circuit's emission levels and immunity and the required emissions limit and the external electromagnetic environment to which the equipment must be immune. The equipment design may also utilize filtering, typically low pass networks, at the shield boundary to attenuate interference picked up by external wires and cable, as well attenuate internally generated noise to reduce its radiation via external wires and cables. Modeling and analysis allows filter performance prediction to be made before hardware is built. A filter's performance with respect to the attenuation of the interference must be considered along with its effect on the desired signal going through it.

The second area of design and analysis for EMI control and compliance centers on the cable/connector assembly design. Cable/connector assembly design and analysis for EMC considers conductor size, characteristic impedance, electrical terminations, shielding, shield characteristics, shield termination methods, shield ground references and coupling between conductors within a cable bundle, and the resulting voltages and currents coupled in from the external environment. The cable/connector assembly modeling is the focus of this effort. Modeling and simulation can be used early in the system design to analyze trade-offs between cable and connector weight (shielding), filtering, signaling schemes and electrical interface design.

This report documents a set of generic context environment models and modeling methods to allow prediction of the interference coupled to equipment connector pins and interface circuits resulting from exposure of the vehicle's cables to radiated and conducted (induced) interference.

Stimuli to the model include the CS114, CS115, CS116, RS103 and RS105 environments as applied per MIL-STD-461F.

Section 2 of this report shows how the system level environments translate to the applicable MIL-STD-461F environment. These applicable environments, and hence the stimuli to the simulation models, are described as CS114, CS115, CS116, RS103 and RS105. A model of each of the five environments is provided.

Inputs to the cable model include cable and connector cross-section details, transfer impedance of the cable and connector shielding, and interface circuit design details.

Outputs from the simulations using these models would include the time domain and/or frequency domain interference at the connector pins presented in terms of voltages and currents. The resulting interference levels can then be compared to the levels of the desired signals and component damage threshold levels to determine if interface failure (upset) and/or component damage is expected. Of course, if unacceptable upset and/or damage are predicted, changes to the design of the cable and/or the protection at the interface circuits can be made to improve the probability of electromagnetic compatibility and first time design success. Section 3 of this report presents the modeling and simulation methodology.

As transfer impedance of a cable shield characteristic is often not available from the shield vendor, methods are presented to allow the determination the transfer impedance of cable shields and connectors.

It should be noted that simulation of electromagnetic effects is a complex computational intensive effort. Computer-based modeling tools continue to evolve with ever increasing capability. This project identified a simulation tool that is believed to be the best mix between tool capability, user base, and ease of use with respect to model development and analysis against the MIL-STD-461F susceptibility environments as seen at equipment interface cables typically used in military vehicles. Section 4 of this report presents a detailed analysis example for a multi-branch cable that may be found in a vehicle.

1.1 Scope

This report documents the findings with respect to the scope in Rockwell Collins Environmental Effects Engineering C2M2L Statement of Work (Appendix A). Specifically, the following is presented.

- The system level electromagnetic environment and flowdown to subsystems, equipment and cables
- Interference environment models as present at the cables
- Cable/connector assembly modeling approach
- Definition of required model inputs
- Methods of obtaining model inputs
- Cable Shield Transfer Impedance Determination
- Simulation of the electromagnetic environments
- Model and simulation workflow
- Examples

1.2 Reference Documents

The following documents are referenced in this report and may be referred to for addition information and background.

MIL-STD-464C – *Department of Defense Interface Standard: Electromagnetic Effects Requirements for Systems*, 1 December 2010.

MIL-STD-461F – *Department of Defense Interface Standard: Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment*, 10 Dec 2007.

MIL-HDBK-235-1C – *Department of Defense Handbook: Military Operational Electromagnetic Environment Profiles, General Guidance*, 1 Oct 2010

MIL-HDBK-235-8 (Classified) – *Department of Defense Handbook: External Electromagnetic Environment Levels from High-Power Microwave Systems (U)*.

IEC 61000-2-13 - *Electromagnetic Compatibility – Part 2-13: Environment – High-Power Electromagnetic (HPEM) Environments – Radiated and Conducted*, First edition, March 2005.

IEC 61000-4-35 - *Electromagnetic Compatibility – Part 4-35: Testing and Measurement Techniques – HPEM Simulator Compendium*, Edition 1.0, July 2009.

MIL-STD-2169 (Classified) – *High Altitude Electromagnetic Pulse Environment (U)*.

CST Studio Suite 2012 User Manual Set – *Installation, Modeling, Post-Processing, Workflow & Solver Overviews*. Part of CST Studio Suite 2012 package, Computer Simulation Technology AG, 2012.

IEC 62153-4-4 – *Metallic Communication Cable Test Methods – EMC – Shielded screening attenuation, test methods for measurement of screening attenuation, a_s , up to and above 3 GHz*, First edition. May 2006.

1.3 Units of Measure

Unless specifically noted otherwise, the units of measure for quantities presented herein are assumed to be as listed in Table 1.3-1.

Table 1.3-1. Units of Measure

Quantity	Unit	Symbol
Area	Sq meter	m ²
Capacitance	Farad	F
Charge	Coulomb	C
Conductivity	Siemens per meter	S/m
Current	Ampere	A
Electric Field Strength (E-field Strength)	Volt per meter	V/m
Impedance	Ohm	Ω
Inductance	Henry	H
Length	meter	m
Magnetic Field Strength (H-field strength)	Ampere per meter	A/m
Permeability	Henry per meter	H/m
Permittivity	Farad per meter	F/m
Power	Watt	W
Resistance	Ohm	Ω
Voltage	Volt	V

1.4 List of Acronyms

The following acronyms are used herein.

BCI	Bulk Cable Injection
C2M2L	Component, Context, and Manufacturing Model Library
CAD	Computer Aided Design
CST	Computer Simulation Technology AG
EM	Electromagnetic
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
EME	Electromagnetic Environment
EMI	Electromagnetic Interference
EMP	Electromagnetic Pulse
HEMP	High-altitude Electromagnetic Pulse
HPM	High-Power Microwave
ICD	Interface Control Document
IEC	International Electrotechnical Committee
I/O	Input / Output
LISN	Line Impedance Stabilization Network
LRU	Line Replaceable Unit
MCAD	Mechanical Computer Aided Design
NTSC	National Television Standard Committee

NEMP	Nuclear (generated) Electromagnetic Pulse
PCB	Printed Circuit Board
RF	Radio Frequency
rms	Root Mean Square
TLM	Transmission Line Matrix

1.5 Definitions

Definitions of key terms used within this report are presented below. They are applicable per EMI requirements stated in MIL-STD-464C and MIL-STD-461F.

2.5-D Model – An electromagnetic model that uses Transmission Line Matrix (TLM) or a Partial Element Equivalent Circuit Model (PEEC) representation of a set of objects being modeled based the cross-section dimensions of the set of objects. This type of model is sometimes referred to as a 2-D model, but is often more correctly stated as a 2.5-D model since some consideration with respect to mesh size against the highest frequency of interest and wave propagation time in the 3rd orthogonal direction is part of the modeling process.

Cable - A set of one or more conductors following a common route designed to provide a particular interface, such as power input, Ethernet, analog video output, digital video input. The conductors in a cable are assumed to be electromagnetically coupled to each other, although coupling may be weak between conductors separated by shields.

Cable Bundle - A section of a cable harness containing all the conductors that occupy the same route. A cable bundle will contain one or more cables. Technically, a cable bundle may consist of only one wire or lead. The conductors in a cable bundle, and the conductors within the cables they contain (if applicable) are assumed to be electromagnetically coupled to each other, although coupling may be weak between conductors separated by shields. A cable bundle runs between two connectors, between one connector and a branch node, or, between two branch nodes. From the standpoint of applying the MIL-STD-461F CS114, CS115 and CS116 environments, a cable bundle includes all the conductors associated with a particular equipment connector.

Cable Harness – A set of cable bundles used for interconnection of equipment items and to the boundary of the electromagnetic volume of interest (if applicable) within a simulation problem space.

Electromagnetic coupling between conductors in different cable bundles that make up a cable harness may be present, although it is often insignificant and thus analytically ignored when compared to other couplings of concern. For the analysis described herein, electromagnetic coupling between cable bundles that make up a cable harness will be ignored.

Equipment item – A electric item designed and typically procured as a stand-alone item that is directly attached to the vehicle or equipment mounting structure (e.g. communications equipment rack). Equipment items connect to other equipment items within a system or subsystem via cable assemblies. An item, from an EMI control standpoint, that is applicable to the EMI requirements contained in MIL-STD-461F with minimal tailoring. “Equipment item” and Line Replaceable Item (LRU) are often considered synonymous.

Electronic Module – A function specific subassembly that is part of a higher level equipment item. Modules typically interface with other modules through a backplane or rack module and are often shielded from some of the electromagnetic environments present at the equipment and subsystem levels.

With the exception of areas where a module interfaces to other equipment via platform cables, this effort does not address modules.

Gross over-shielded Bundle – A shield cable bundle containing a relatively high quality bundle shield, with the shield being electrically bonded to a conductive backshells on the bundle's connectors such that, for all practical purposes, the interference present on the shielded conductors from the external environment applied in the vicinity of the over-shield will be indirectly due to environment induced current flow on the over-shield rather than from direct coupling to the external environment.

Gross overbraid shield – A shield on a gross over-shielded bundle that uses a MIL-QQB575R, or equivalent, braid cable bundle shield.

Ground plane - A reference plane for the electromagnetic environment and for physical objects used in an electromagnetic model or simulation. The ground plane, per common convention in EMI/EMC problems, is located at $z = 0$ in the 3-dimensional Cartesian coordinate problem space. The ground plane is a conductor, but it does not have to be a perfect conductor.

Grounded node – A node containing a conductor that connects to the ground plane at the physical planar ($x-y$ plane if $z = 0$ plane is the ground plane) location of the node.

Line Replaceable Unit (LRU) – See Equipment Item.

Node – A point in three dimensional space that can be used as a terminus for a segment and/or a set of conductors.

Platform – A physical structure that hosts system electronic equipment. Examples include aircraft, ground vehicles, fixed facility buildings and soldiers (for a soldier-mounted system)

Shield - A conductor used for one or more of the following purposes:

- (a) A conductor not specifically designed as a signal return, but rather for the attenuation of electromagnetic energy that is being imposed upon a wire, a group of wires, a cable, a cable bundle and/or a cable harness by geometrically surrounding the conductors being shielded. The imposed interference may be from the external environment or from other conductors. A shield over a wire pair that interfaces a remote temperature sensor (thermo-resistor) located in an area of intense electromagnetic radiation to a data acquisition receiver/processor is an example of this type of shield – the shield is used to reduce interference seen across the input port of the receiver.
- (b) A conductor not specifically designed as a signal return, but rather for the reduction of coupling from a particular wire, a group of wires, a cable, a cable bundle and/or a cable harness to the environment by geometrically surrounding the conductors being shielded. The shield of a shielded twisted pair cable used for a transformer coupled Ethernet interface is an example of this type of shield – the shield is used to reduce emissions from the cable.
- (c) A signal or power return conductor having a cross-section that surrounds the signal or power conductor(s) designed with the intention of reducing signal radiation, reducing the coupling of external interference and/or providing a particular characteristic transmission line impedance. The shield on a coax cable used for cable TV is an example of this type of shield, as well as the shield on a transition minimized differential signaling pair used in a Digital Video Interface (DVI) interface - the shield is the signal return.

Route – A physical path, consisting of one or more segments, for a cable and/or cable bundle.

Segment – A physical path between two nodes.

Shielded Cable – A cable (per definition above) containing one or more shields (as defined above).

Shielded Cable Bundle – A cable bundle (as defined above) containing a shield, or multiple layers of shields, that run the length of the bundle and geometrically surrounds all the conductors in the bundle.

System - A composite of equipment, subsystems, skilled personnel, and techniques capable of performing or supporting a defined operational role (MIL-STD-464C definition). With respect to hardware, “system” refers to top level platform (ship, aircraft, fixed station, ground vehicle).

Subsystem – One or more integrated equipment items interfaced within a system or platform to perform a top level function.

2. THE ELECTROMAGNETIC ENVIRONMENTS

The baseline system level electromagnetic environment for military ground vehicles is specified in MIL-STD-464C. This standard contains a “main body” which states a baseline set of requirements and electromagnetic environments to which systems are exposed to. It also contains an appendix which provides rationale, guidance, and lessons learned for each requirement to enable the procuring activity to tailor the baseline requirements for a particular situation. Further guidance on specific tailoring of the environment is provided in MIL-HDBK-235-1C. In general, the system level environments presented in MIL-STD-461C are “worst case”. Often, after consideration of mission, platform locations and the criticality of equipment, subsystem or system survival and operational needs with trade-offs centered around weight, cost and procurement schedule, the procuring agency may present only a subset of the requirements as being applicable. As an example, with respect to the nearby lightning threat (the baseline threat is specified with a 10 meter distance), MIL-STD-464C states: *“Many ground systems can accept some risk that the system operates only after a moderate lightning strike at a reasonable distance. For example, a requirement for equipment in a tactical shelter to survive a 90th percentile lightning strike at 50 m may represent reasonable risk criteria for that shelter. This type of requirement would result in a high level of general lightning protection at a reduced design and test cost.”*

2.1 Semantics

Per MIL-STD-464C, the term “system” refers to a composite of equipment, subsystems, skilled personnel, and techniques capable of performing or supporting a defined operational role. With respect to hardware, “system” refers to top level platform (ship, aircraft, fixed station, ground vehicle). This is how “system” will be used in this endeavor.

MIL-STD-461C defines a “subsystem” as a collection of devices or equipments designed and integrated to function as a single entity. A military vehicle may contain several subsystems, examples including a power distribution subsystem, a satellite radio navigation subsystem, a high frequency communication radio subsystem, an engine control system and a fire control subsystem. The subsystem not only includes equipment, but cables as well. Platform or facility cables and their associated connectors are often referred to as a subsystem by themselves – “cable/connector assemblies” will be used to describe this type of asset.

Subsystems contain integrated equipment items interfaced within a system or platform with specific cabling. An “equipment item” typically refers to a Line Replaceable Unit (LRU) that receives operating power, signal I/O and control functions via platform or facility wiring and is designed perform a specific function or task, and which can typically be procured as a stand-alone item.

Equipment may contain “modules”, which may be line or field replaceable entities. When considering electromagnetic environments, “modules” are notably different than “equipment items” in that they often do not mechanically interface directly with the system platform or facility equipment racks, but rather interface to a host equipment item which may provide to it conditioned power, filtered signal I/O, and/or electromagnetic shielding. Because of this, modules typically cannot easily be tested by themselves in an electromagnetic environment derived from the system level electromagnetic environment. Hence the lowest level EMI compliance verification, and requirements for procurement, typically occur at the equipment item / Line Replaceable Unit level. It is very important to delineate these levels of distinction as this will drive how the system level environment is modeled to various parts of the system, especially when procurement of equipment, cables and modules is considered.

2.2 Ground Vehicle System Electromagnetic Environments

The baseline electromagnetic environment seen by ground vehicle systems is presented in MIL-STD-464C. The environments include those in which the source is from within the system as well as those that are sourced from outside the system. Each is described below, noting the absolute threat levels may be tailored by the procuring activity based on system location, mission profile and other previously mentioned trade-offs.

2.2.1 Radiated Radio Frequency (RF) Electromagnetic Environment (RF EME)

Radio and radar transmitters can present very high levels of energy and electromagnetic field strength to a system and the equipment and subsystems it contains. This type of environment is referred to as the external Radio Frequency (RF) Electromagnetic Environment (EME). RF EME is described in terms of field strength as a function of frequency. Table 4 in MIL-STD-464C presents the RF EME applicable to ground systems. This table represents typical worst-case field strengths that may be present in the area in which a ground system is operating. The sources of these electromagnetic fields are radiation from transmitter antennas of known radios used for communication, navigation and surveillance.

If a ground vehicle were to be operated on a ship deck, the baseline levels in Table 1 of MIL-STD-461C may apply as well.

Typically, the vehicle / platform structure provides some (6 to 20 dB) attenuation of this environment with respect to subsystems and equipment contained within.

2.2.2 Electromagnetic Pulse (EMP)

MIL-STD-464C states: “*The system shall meet its operational performance requirements after being subjected to the EMP environment. This environment is classified and is currently defined in MIL-STD-2169. This requirement is applicable only if invoked by the procuring activity. Compliance shall be verified by system, subsystem, and equipment level tests, analysis, or a combination thereof.*” Note that MIL-STD-2169 is a classified document. Often, if the EMP environment is applicable, subsystem and equipment specifications will contain detailed requirements with respect to required recovery time and output and actions during the recovery period. As an example, a mission display may be required to recover within 40 seconds after the EMP event and must never provide misleading or non-current data. The terms High-altitude Electromagnetic Pulse (HEMP) and Nuclear Electromagnetic Pulse (NEMP) are often synonymous with the term “EMP.”

2.2.3 Lightning Effects

MIL-STD-464C presents aspects of the lightning environment that are relevant for designing protection against the direct effects lightning. “Direct effects” refers to the situation where the lightning channel attaches to the system. Also provided are aspects of the lightning environment associated with a direct strike that are relevant for protecting the platform from indirect effects. Based on the electrical characteristics of the platform or facility, the direct effects environment will induce currents into wires, cables and cable bundles inside the platform or facility as well as produce voltages at equipment connector pins. Thus indirect effects particulars can vary considerably between platforms and facilities and between locations within a platform or facility.

Indirect effects particulars for civil aircraft subsystems and equipment are typically determined by detailed testing, often on scale models or representative production sections of aircraft containing proposed cable assemblies. For subsystems and equipment on ships, ground platforms, ground facilities, and often military aircraft, generic “catch-all” equipment and subsystem standards, such as ANSI C62.14, IEC 61000-4 and -5 and CS115 and CS116 from MIL-STD-461F, are used to generically capture the indirect effects environment as present at cable assemblies and/or connector pins.

2.2.4 High Power Microwave (HPM)

MIL-STD-464C notes the existence of hostile radio frequency environments produced by microwave sources (weapon) capable of emitting high power or high energy densities. The weapon may produce, and direct at the system (e.g. ground vehicle), microwaves in the form of a single pulse, repetitive pulses, pulses with complex modulation, or continuous wave (CW) characteristics.

MIL-STD-464C (Section 5.4) states the following about HPM: *“The system shall meet its operational performance requirements after being subjected to the narrowband and wideband HPM environments. Applicable field levels and HPM pulse characteristics for a particular system shall be determined by the procuring activity based on operational scenarios, tactics, and mission profiles using authenticated threat and source data such as the Capstone Threat Assessment Report. This requirement is applicable only if specifically invoked by the procuring activity. Compliance shall be verified by system, subsystem, and equipment level tests, analysis, or a combination thereof.”*

The details of the HPM sources are generally classified. MIL-HDBK-235-8 (Classified) provides information on HPM threats. Appendix A.5.4 in MIL-STD-464C provides methods to compute resulting threat levels from a given HPM threat at a particular distance from the threat’s victim.

IEC 61000-2-13 defines a set of typical radiated and conducted High Power Electromagnetic (HPEM) environment waveforms that may be encountered from an intentional generator targeting a civilian facility. IEC 61000-4-35 provides information about existent system-level HPEM simulators and their applicability as test facilities and validation tools for HPEM immunity testing. The terms Ultra Wideband (UWB) and High-Power Electromagnetic (HPEM) used in IEC 61000-2-13 are generally synonymous with the term HPM as used in military standards.

2.2.5 Electrostatic Discharge

MIL-STD-464C defines an Electrostatic Discharge (ESD) environment for electrical and electronic subsystems as an 8 kV contact discharge or a 15 kV air discharge. In both cases the discharge capacitance is 150 pF and the discharge resistance is 330 ohm, with the circuit inductance not to exceed 5 microhenry. This environment, as a worst case, is thus assumed to be present at equipment and cable assemblies. Although requirements for compliance testing to these exact requirements is typically not present in subsystem and equipment procurement documents, the intent is absorbed in other subsystem and equipment level EMI control compliance requirements, such as MIL-STD-461F CS115.

2.2.6 Internally Generated Electromagnetic Energy

MIL-STD-464C requires that the electromagnetic environment generated by equipment and subsystem is required to be controlled per MIL-STD-461. MIL-STD-461 places limits on conducted and radiated emissions from equipment and systems. These limits are generally designed to protect system radio

receivers from interference that may enter via the radio's antenna located within the system. Interference present at equipment and associated cables from energy unintentionally radiated and conducted from other nearby equipment is typically orders of magnitude less than what is seen from the environments described in 2.2.1 through 2.2.5 above.

2.3 Electromagnetic Environment for Ground Vehicle Subsystems and Equipment

Figure 2.3-1 is a diagram illustrating how the system level electromagnetic threat environments described in Section 2.2 above flow down to subsystems, equipment and associated cables.

From a procurement standpoint, the vehicle or platform is specified to exhibit electromagnetic compatibility (EMC) per a system level standard, such as MIL-STD-464C. With respect to immunity from the threats described in 2.2 above, compliance is verified typically by limited testing at the system level. Application of the complete threat environments as described in MIL-STD-464C is impractical. However, it is practically possible to set limits and demonstrate electromagnetic control compliance at the equipment and subsystem level. Hence MIL-STD-464C states: "*Individual subsystems and equipment shall meet interference control requirements (such as the conducted emissions, radiated emissions, conducted susceptibility, and radiated susceptibility requirements of MIL-STD-461) so that the overall system complies with all applicable requirements of this standard. Compliance shall be verified by tests that are consistent with the individual requirement (such as testing in accordance with MIL-STD-461).*" Thus equipment and subsystems, with their associated cables, are procured with the requirement to comply, with verification by test and/or analysis, to an equipment/subsystem EMI control standard, such as MIL-STD-461F.

MIL-STD-461F essentially condenses the many complex and extensive collection of external and internally generated electromagnetic threats present at the system or platform level and consolidates them into a practical set of generic requirements that can be economically tested and analyzed at the subsystem and equipment levels. The limits in MIL-STD-461F do consider a small amount of shielding and threat attenuation that may be provided by the host platform. For instance, the generic RF E-field radiated susceptibility requirement (RS103) for army ground equipment and associated cabling is 50 V/m from 30 MHz to 18 GHz, using 1 kHz, 50% duty cycle pulse modulation. This is notably different than the external EME stated for ground systems shown in MIL-STD-464 as illustrated in Figure 2.3-2. As seen in Figure 2.3-2 the MIL-STD-464C EME threat for ground vehicles contains many different frequency ranges, each containing a different type of modulation, and with high peak field strength levels. Note that the 50 V/m MIL-STD-461F requirement also considers the internal threats, such as RF produced by on-board radio antennas, to equipment. It is also important to note that testing systems, and large subsystems and equipment to the MIL-STD-464C levels is costly, as generally any swept frequency radiated E-field environment with peaks and/or averages greater than 600 V/ and 200 V/m, respectively, is considered very expensive to produce for testing items larger than a breadbox.

The following sections describe influencing factors, tailoring considerations, and the interference model for each of the five MIL-STD-461F electromagnetic environments that may be applicable to susceptibility via cables in ground vehicles.

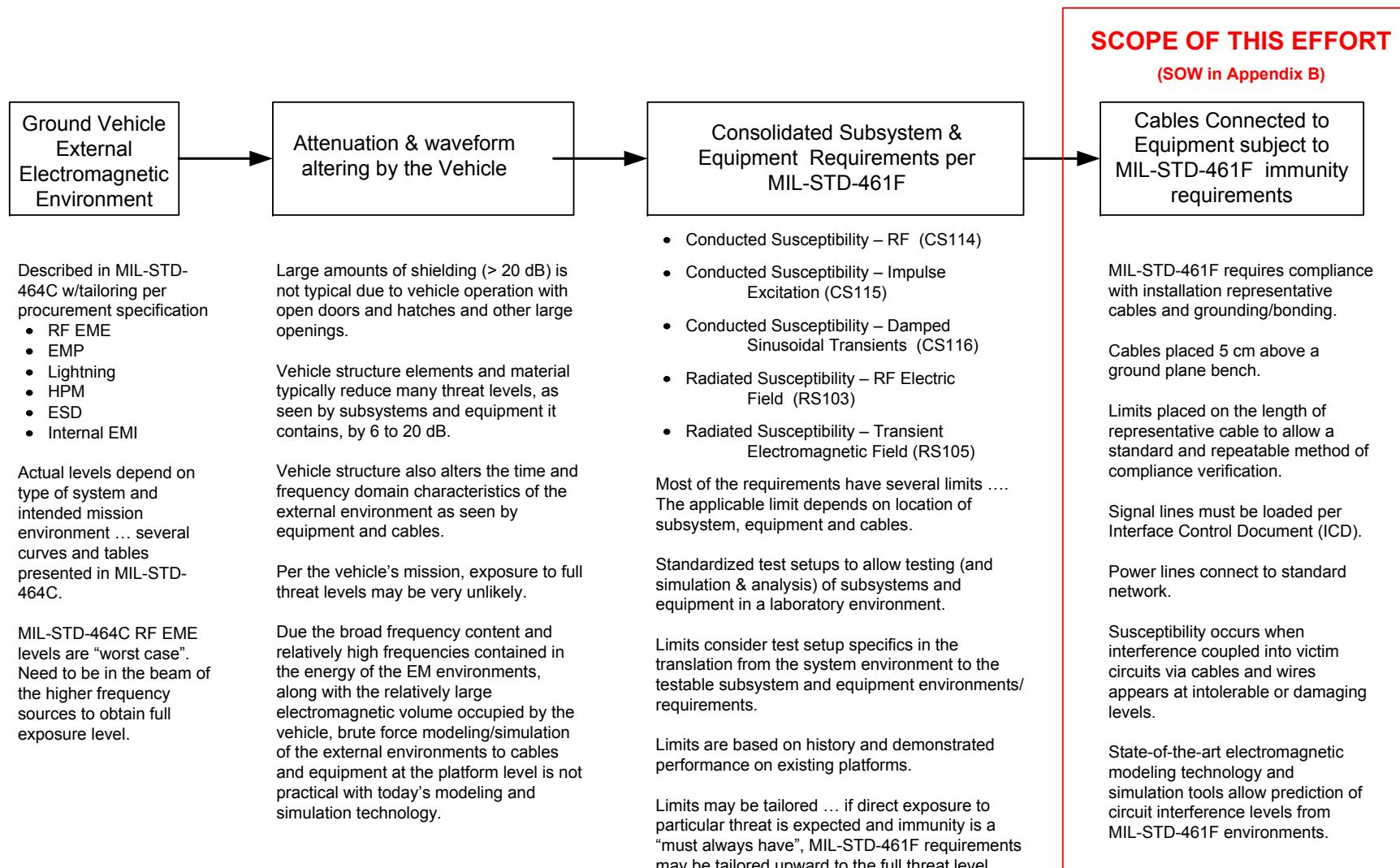


Figure 2.3-1. Flow-down of System Level Electromagnetic Threats to Equipment, Subsystems and Cables

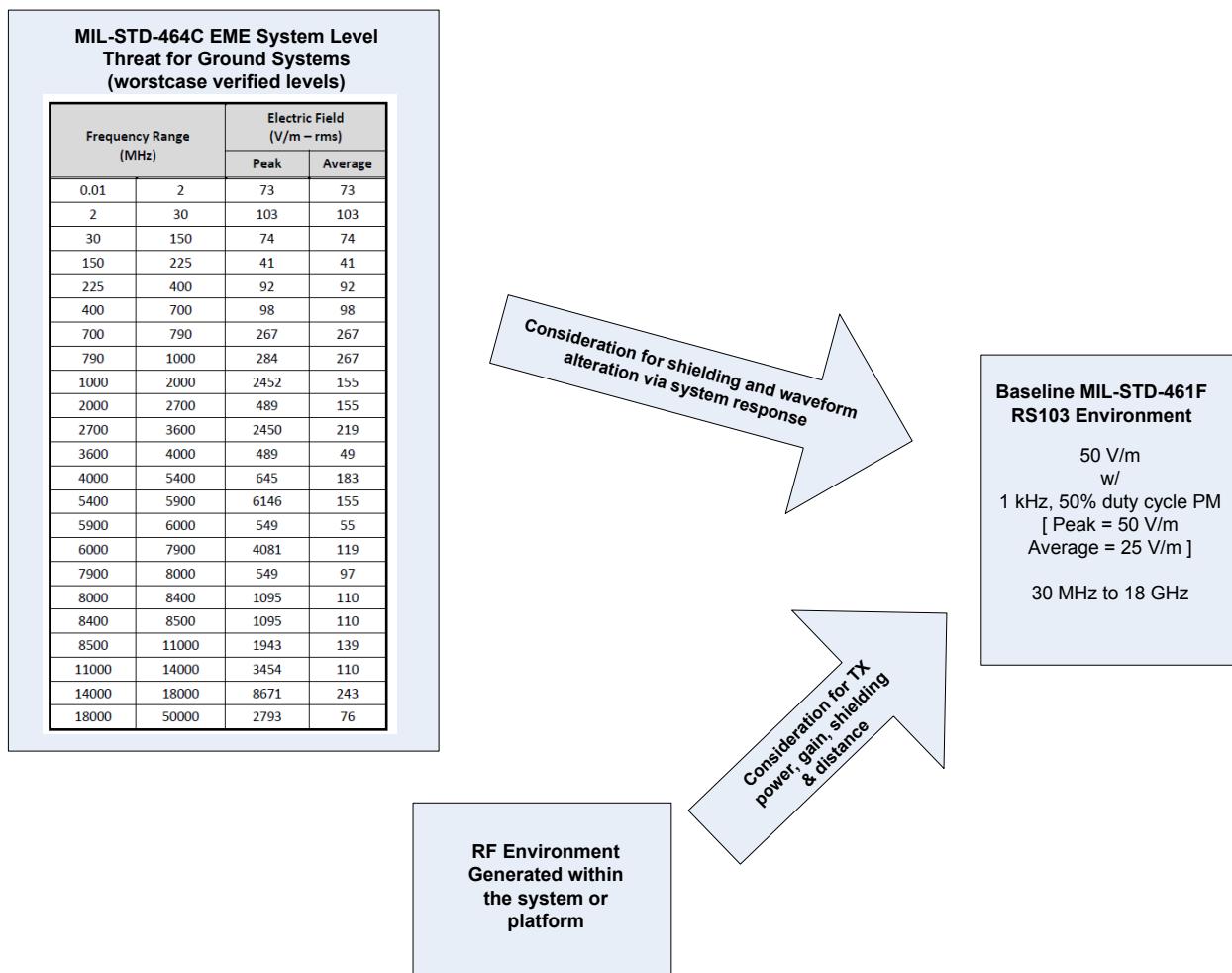


Figure 2.3-2. Origin of Baseline MIL-STD-461F RS103 Requirement Applicable to Ground System Equipment

2.3.1 Conducted Susceptibility - RF – CS114 Environment Model

The CS114 environment involves injection of RF over the frequency range of 10 kHz to 200 MHz on to all cables bundles¹, as well as the power cable without the return and ground conductors, in turn, using a current transformer injection probe. Injection occurs 10 cm from the equipment connector. The system level RF EME environment, and the internal system / platform transmitters and electrical equipment, produce the CS114 threat described in MIL-STD-461F at cables connecting to equipment and subsystems. The CS114 requirement is stated in terms of induced cable current from a source with a specified impedance and a limited power output capability. Figure 2.3.1-1 shows baseline CS114 limits. Figure 2.3.1-2 shows limit applicability as a function of platform type. From Figure 2.3.1-2, we see that for Army ground platforms, curve 3 and curve 4 are applicable for the 10 kHz to 2 MHz and the 2 MHz to 200 MHz frequency ranges, respectively.

¹ Includes power cables/bundles containing all applicable return and ground conductors.

The levels noted in Figure 2.3.1-1 are to be applied to cable bundles that are representative of the intended installation, as documented in the Interface Control Document (ICD), vehicle wiring diagrams, and/or the equipment installation manual. “Representative” equates to identical cable and wire cross-section, identical or electromagnetically equivalent connector assemblies, identical wire and cable shield properties (if shielding is applicable), and identical or equivalent shield and wiring termination, routing, and grounding at the connectors.

The equipment subject to the CS114 environment must be electrically bonded to the copper bench in a manner that represents how it is bonded (or not bonded) to the host platform. Key quantities to consider in equipment bonding include DC resistance, contact locations, bonding conductor material and geometry (if applicable), and the amount of surface area at bond between conductive surfaces.

The MIL-STD-461F CS114 requirements are also normalized to a condition where the cables under test are placed 5 cm above a conductive ground plane bench. Length may be limited to 10 meters if the installation cable is longer than 10 meters. If the length is unknown, a compliant default length of 3 meters is typically used, as this allows compliance to other baseline conditions stated in elsewhere in MIL-STD-461F.

The cables must also be terminated with loads to simulate the electrical properties (impedance, grounding, balance, and so forth) present in the actual installation.

Power leads are to be between 2 and 2.5 meters long and terminated into a line impedance stabilization network (LISN). Figure 2.3.1-3 shows a schematic of the LISN. The purpose of the LISN is to provide a standardized platform-representative power system impedance to the equipment or subsystem being tested or analyzed.

As noted in MIL-STD-461F, the limits are derived from measurements made on platforms that were basically electrically conductive, but not designed to have intentionally shielded volumes. Also, as noted in MIL-STD-461F, the platform can be illuminated with a low level version of the EME threats while monitoring induced levels on cables, scaling the measured levels by the same factor used for the EME threat to determine the expected levels to tailor the baseline requirements.

MIL-STD-461F presents specific requirements for the application of the CS114 environment to cables and leads – proper attention to these details will assure correct translation of the system environment to the cable/connector assemblies. Figure 2.3.1-4 describes the CS114 environment in terms of an electrical model with equations that can be used to simulate the environment as applied to cable bundles, and to power cables with power returns and grounds excluded.

Note that Figure 2.3.1-4 contains three schematics. The first is for the establishment of the reference maximum interference power from the environment source that could be coupled to the cable bundle of interest. The second two schematics represent the interference as applied to the cable bundle or cable of interest. If the cable bundle has a gross over-shield (often implemented with a MIL QQB575R braid shield and thus referred to as a “gross overbraid shield”), then injection of the interference would be viewed as direct coupling to the over-shield conductor only. Characteristics of the over-shield will determine coupling from it to the conductors it shields.

The third schematic in Figure 2.3.1-4 applies to cable bundles and cables that do not contain a gross over-shield. In this case, each conductor, or the shields on the shielded cables within the bundle, would be subjected to the direct injection of the environment. The total bundle injected current would divide amongst the outer conductors based on their terminating impedances and other electrical characteristics.

Per the equations in Figure 2.3.1-4, the environment model essentially applies current to a cable bundle or power cable less the returns and grounded conductors, through an injection probe driven by a source initially set to produce the current indicated in Figure 2.3.1-1 into a 100 Ohm loop, with that source level being reduced during test, if required, to the point where the resulting cable bundle current is limited to twice the Figure 2.3.1-1 current. It should be noted that the aforementioned levels are peak detected, but stated as the rms value of a continuous sinewave having that peak level. The CS114 RF is 1 kHz pulse modulated with a 50% duty cycle. The injection probe model shown in Figure 2.3.1-4 produces an insertion loss that is compliant to the MIL-STD-461F requirements.

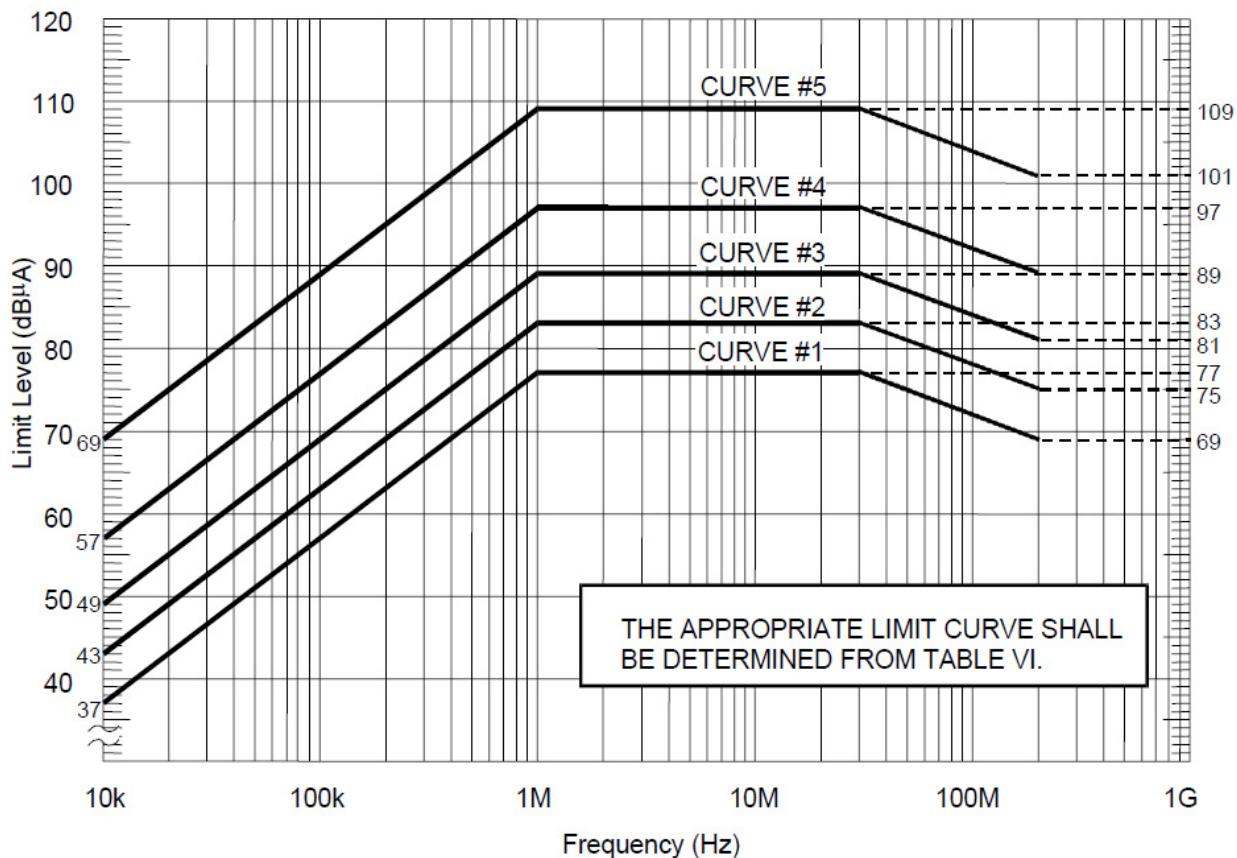


Figure 2.3.1-1. CS114 Current Limits

		LIMIT CURVE NUMBERS SHOWN IN FIGURE CS-114-1 AND LIMITS							
PLATFORM FREQUENCY RANGE		AIRCRAFT (EXTERNAL OR SAFETY CRITICAL)	AIRCRAFT INTERNAL	ALL SHIPS (ABOVE DECKS) AND SUBMARINES (EXTERNAL)*	SHIPS (METALLIC) (BELOW DECKS)	SHIPS (NON- METALLIC) (BELOW DECK) **	SUBMARINE (INTERNAL)	GROUND	SPACE
4 kHz to 1MHz	N	-	-	77 dB μ A	77 dB μ A	77 dB μ A	77 dB μ A	-	-
10 kHz to 2 MHz	A	5	5	2	2	2	1	3	3
	N	5	3	2	2	2	1	2	3
	AF	5	3	-	-	-	-	2	3
2 MHz to 30 MHz	A	5	5	5	2	4	1	4	3
	N	5	5	5	2	4	1	2	3
	AF	5	3	-	-	-	-	2	3
30 MHz to 200 MHz	A	5	5	5	2	2	2	4	3
	N	5	5	5	2	2	2	2	3
	AF	5	3	-	-	-	-	2	3

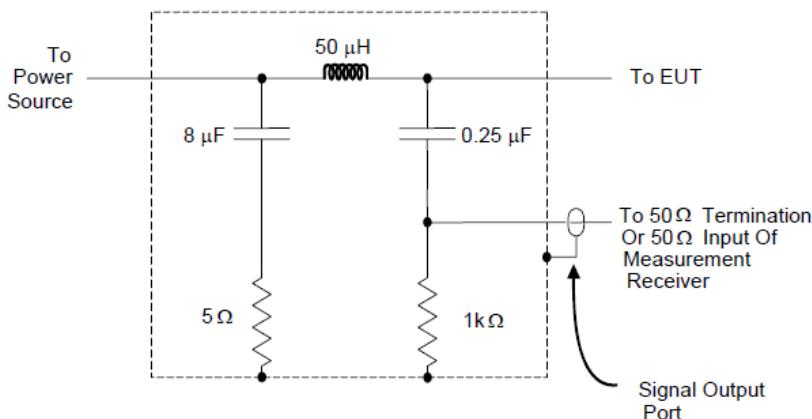
KEY: A = Army

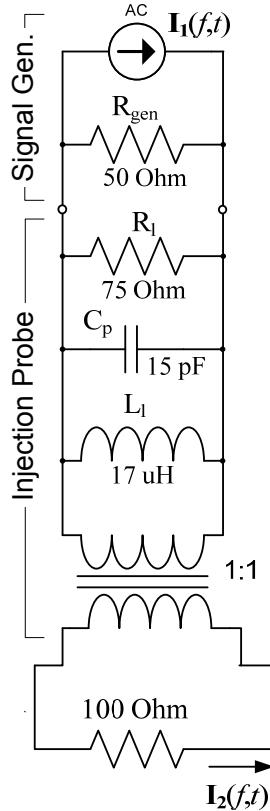
N = Navy

AF = Air Force

* For equipment located external to the pressure hull of a submarine but within the superstructure,
use SHIPS (METALLIC) (BELOW DECKS)

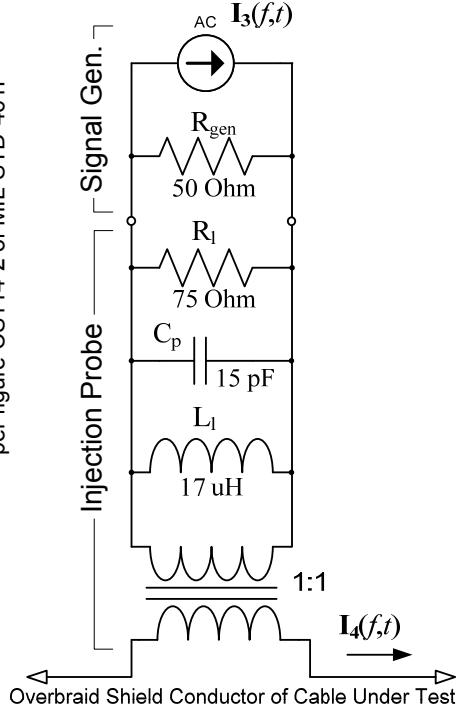
** For equipment located in the hanger deck of Aircraft Carriers

Figure 2.3.1-2. CS114 Limit Curve Applicability**Figure 2.3.1-3. MIL-STD-461F Power Line Impedance Stabilization Network (LISN)**

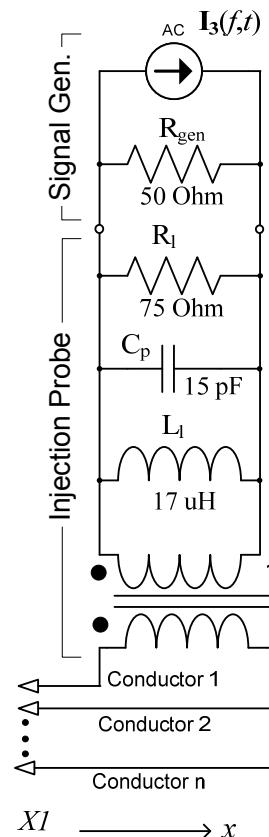


Calibration Setup

Injection Probe
Signal Gen.
Injection Probe meets insertion loss requirements per figure CS114-2 of MIL-STD-461F



Injection to Cable Bundle Presented as a Single Conductor



Injection to Cable Bundle Presented as Multiple Conductors

Conditions with no modulation applied to \mathbf{I} :

$$IL_{inv_base}(f) = \frac{I_1(f)}{I_2(f)}$$

$$I_4(f) = \sum_n I_{cn}(f) \quad \text{if} \quad \text{Outer_conductors} > 1$$

$$Xfer(f) = \frac{I_4(f)}{I_{Limit}(f)} \quad \text{for} \quad I_3(f) = IL_{inv_base} \cdot I_{Limit}(f)$$

$$I_3(f) = \begin{cases} IL_{inv_base} \cdot I_{Limit}(f) & \text{if} \quad Xfer(f) \leq 2 \\ IL_{inv_base} \cdot I_{Limit}(f) \cdot \frac{2}{Xfer(f)} & \text{Otherwise} \end{cases}$$

Condition during susceptibility sweep:

$$I_3(f,t) = I_3(f) \cdot \sqrt{2} \cdot \sin(2 \cdot \pi \cdot f \cdot t) \cdot m(t)$$

$$m(t) = \sum_{n \geq 0} u(t - 0.5 - 1 \cdot n) - u(t - 1 \cdot (n+1))$$

t = time in milliseconds

2.3.2 Conducted Susceptibility – Bulk Cable Injection, Impulse Excitation – CS115 Environment Model

The CS115 environment involves injection of fast transition time pulses on to cable bundles² as well as power cables with power returns and grounds excluded. The CS115 environment presents transients at the platform's cables that may be expected to result from the external lighting environment, certain HPM environments and EMP environments, as well as transients from platform switching operations and equipment/subsystem ESD. The CS115 excitation is a trapezoid pulse at the generator output; the actual waveform on the cable bundle or cable will be dependent on natural resonance conditions associated with cable and characteristics of the interfacing circuits.

Since this requirement is transient in nature, it may be acceptable from a system performance standpoint if desired signals are overcome with the transient, as response smoothing, signal processing and/or data error correction may make the incident transparent to the user. This must be considered before drawing conclusions from a simulation of the application of the CS115 environment to cable bundles, and power cables with returns and grounds excluded.

CS115 is applied to cable bundles and power cables with returns and grounds excluded using an injection probe (current transformer). Figure 2.3.2-1 shows the CS115 current waveform that would be present in a 100 Ohm calibration loop if the injection probe had a flat frequency response. Realistically, the current pulse in the 100 Ohm calibration loop would appear as shown in Figure 2.3.2-2 (from MIL-STD-461F). The calibration pulse is considered compliant if the peak current is equal to 5 Amperes and the rise and fall times are no greater than 2 nanoseconds, as shown in Figure 2.3.2-2.

After the transient generator is adjusted to produce a compliant pulse into a 100 Ohm loop, the injection probe is moved to each cable bundle, and then to each power cable with returns and grounds excluded, in turn, to allow application of the reference transient. The cable bundles must be representative of the intended installation. "Representative" equates to identical cable and wire cross-section, identical or electromagnetically equivalent connector assemblies, identical wire and cable shield properties (if shielding is applicable), and identical, or equivalent, shield and wiring termination, routing, and grounding at the connectors.

The equipment subject to the CS115 environment must be electrically bonded to the copper bench in a manner that represents how it is bonded (or not bonded) to the host platform. Key quantities to consider in equipment bonding include DC resistance, contact locations, bonding conductor material and geometry (if applicable), and the amount of surface area at bond between conductive surfaces.

The MIL-STD-461F CS115 requirements are also normalized to a condition where the cables under test are placed 5 cm above a conductive ground plane bench. Length may be limited to 10 meters if the installation cable is longer than 10 meters. If the length is unknown, a compliant default length of 3 meters is typically used, as this allows compliance to other baseline conditions stated in MIL-STD-461F.

The cables must also be terminated with loads to simulate the electrical properties (impedance, grounding, balance, and so forth) present in the actual installation.

Power leads are to be between 2 and 2.5 meters long and terminated into a line impedance stabilization network (LISN). Figure 2.3.1-3 shows a schematic of the LISN. The purpose of the LISN is to provide a platform representative power system impedance to the equipment or subsystem being tested or analyzed.

² Includes power cables/bundles containing all applicable return and ground conductors.

The CS115 pulses are to be applied at a 30 Hz rate for one minute to ensure that a sufficient number of pulses are applied to provide confidence that the equipment will not be upset.

MIL-STD-461F presents specific requirements for the application of the CS115 environment to cables bundles and power cables without returns and grounds – proper attention to these details will assure correct translation of the system environment to the cable/connector assemblies.

Figure 2.3.2-3 describes the CS115 environment in terms of an electrical model with equations that can be used to simulate the environment as applied to cable bundles, and to power cables with power returns and grounds excluded.

Note that Figure 2.3.2-3 contains three schematics. The first is for verification that the CS115 source can produce the required transient into a 100 Ohm loop. The second two schematics represent the interference as applied to the cable bundle of interest. If the cable bundle has a gross over-shield (often implemented with a MIL QQB575R braid shield and thus referred to as a “gross overbraid shield”), then injection of the interference would be viewed as direct coupling to the over-shield conductor only. Characteristics of the over-shield will determine coupling from it to the conductors it covers.

The third schematic in Figure 2.3.2-3 applies to cable bundles that do not contain a gross over-shield. In this case, each conductor, or the shields on the shielded cables, would be subjected to the direct injection of the environment. The total bundle injected current would divide amongst the outer conductors based on their terminating impedances and other electrical characteristics.

Per the equations in Figure 2.3.2-3, the environment model essentially applies current to a cable bundle or power cable less the return and ground conductors, through an injection probe driven by a source initially set to produce the current pulse shown in Figure 2.3.2-2 into a 100 Ohm loop. The injection probe model shown in Figure 2.3.2-3 is compliant to the MIL-STD-461F requirements as it produces a transient current into a 100 Ohm loop with the specified transition time, pulse width and peak amplitude. The circuit model of the probe represents the Solar EMC 9142-1N injection probe.

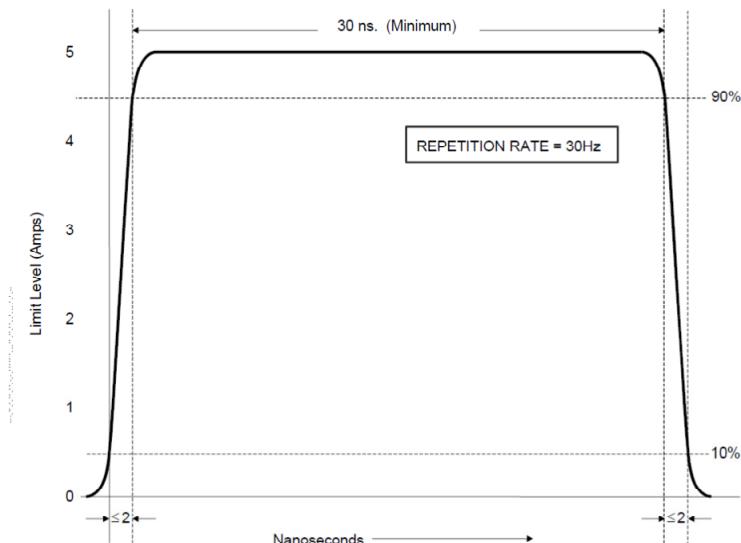


Figure 2.3.2-1. CS115 Pulse Waveform

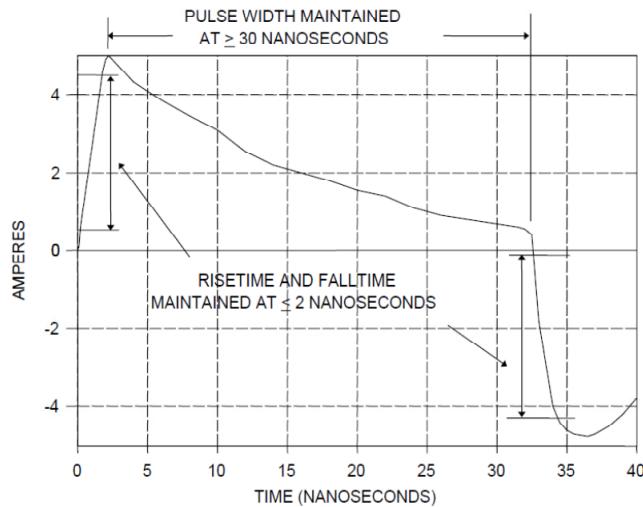
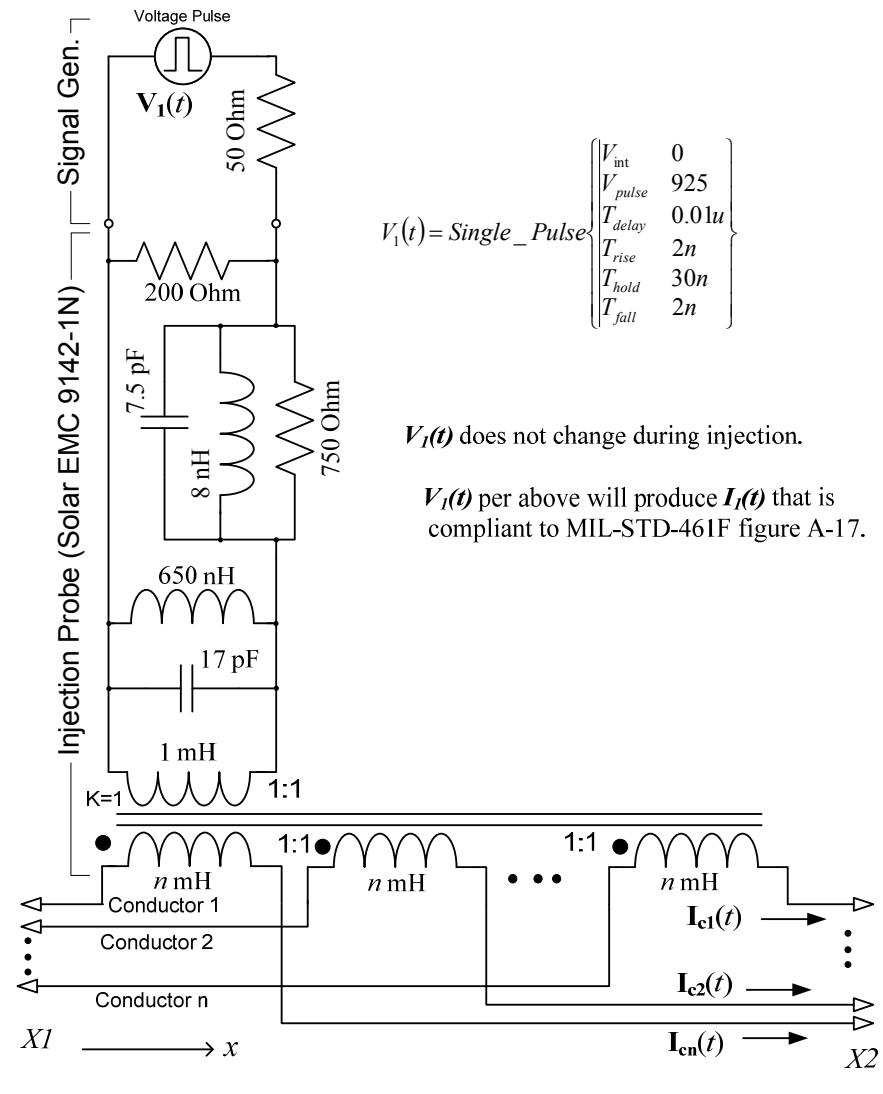
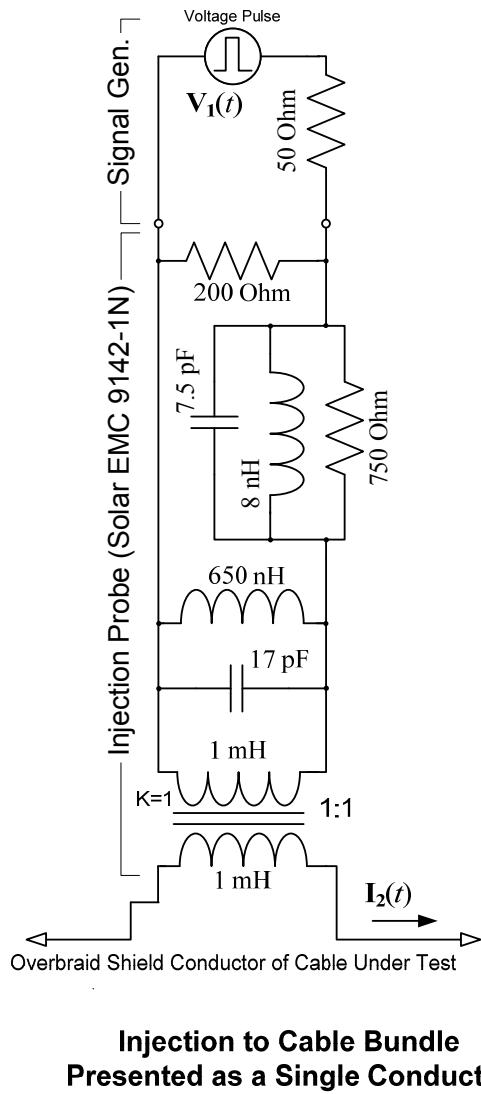
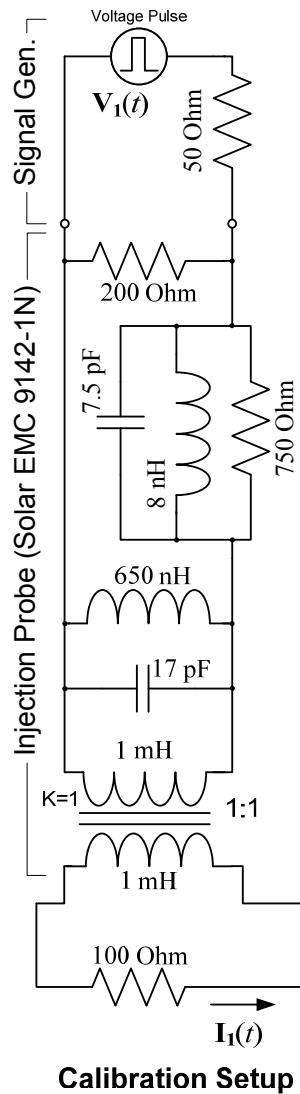


Figure 2.3.2-2. Compliant Injected CS115 Pulse Waveform into 100 Ohm Loop



$$V_1(t) = \text{Single_Pulse} \left\{ \begin{array}{ll} V_{\text{int}} & 0 \\ V_{\text{pulse}} & 925 \\ T_{\text{delay}} & 0.01\mu \\ T_{\text{rise}} & 2n \\ T_{\text{hold}} & 30n \\ T_{\text{fall}} & 2n \end{array} \right\}$$

$V_1(t)$ does not change during injection.

$V_1(t)$ per above will produce $I_1(t)$ that is compliant to MIL-STD-461F figure A-17.

2.3.3 Conducted Susceptibility – Damped Sinusoidal Transients – CS116 Environment Model

The CS116 environment involves injection of damped sinusoidal transients over the frequency range of 10 kHz to 100 MHz onto all cables bundles³, the power cables by themselves, as well as individual high side power leads, in turn, using a current transformer injection probe. The CS116 environment simulates electrical current and voltage waveforms occurring in platforms from excitation of natural resonances. (In contrast, CS115 excites natural resonances.) Damped sinusoidal waveforms are a common occurrence on platforms from both external stimuli such as lightning and EMP, and from platform electrical switching phenomena. Waveforms appearing on cables can be due to the cable itself resonating or due to voltage and current drives resulting from other resonances on the platform. Wide frequency coverage (10 kHz to 100 MHz) is included in the base environment to account for a wide range of possible conditions.

Since this requirement is transient in nature, it may be acceptable from a system performance standpoint if desired signals are overcome with the transient, as response smoothing, signal processing and/or data error correction may make the incident transparent to the user. This must be considered before drawing conclusions from a simulation of the application of the CS116 environment to cable bundles, power cables and individual high side power leads.

Figure 2.3.3-1 shows the required damp sinusoid current waveform as present when injected into a 100 Ohm loop. Figure 2.3.3-2 presents the limit values for I_p . Typically, evaluation by test is performed at 10 kHz, 100 kHz, 1 MHz, 3 MHz, 10 MHz, 30 MHz and 100 MHz.

The damped sinusoid waveforms are to be applied to cable bundles that are representative of the intended installation. “Representative” equates to identical cable and wire cross-section, identical or electromagnetically equivalent connector assemblies, identical wire and cable shield properties (if shielding is applicable), and identical, or equivalent, shield and wiring termination, routing, and grounding at the connectors.

The equipment subject to the CS116 environment must be electrically bonded to the copper bench in a manner that represents how it is bonded (or not bonded) to the host platform. Key quantities to consider in equipment bonding include DC resistance, contact locations, bonding conductor material and geometry (if applicable), and the amount of surface area at bond between conductive surfaces.

The MIL-STD-461F CS116 requirements are also normalized to a condition where the cables under test are placed 5 cm above a conductive ground plane bench. Length may be limited to 10 meters if the installation cable is longer than 10 meters. If the length is unknown, a compliant default length of 3 meters is typically used, as this allows compliance to other baseline conditions stated in MIL-STD-461F.

The cables must also be terminated with loads to simulate the electrical properties (impedance, grounding, balance, and so forth) present in the actual installation.

Power leads are to be between 2 and 2.5 meters long and terminated into a line impedance stabilization network (LISN). Figure 2.3.1-3 shows a schematic of the LISN. The purpose of the LISN is to provide a platform representative power system impedance to the equipment or subsystem being tested or analyzed.

As noted in MIL-STD-461F, the limits are set at levels that cover most induced levels found in platforms during system-level testing to external transient environments.

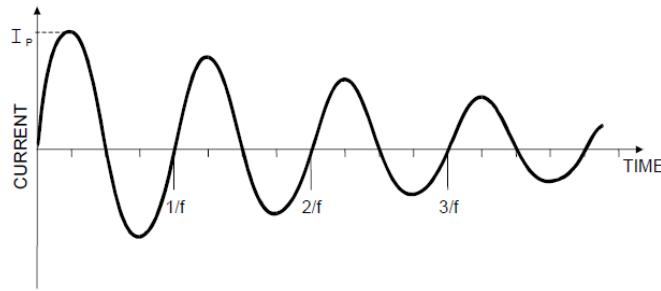
³ Power cables to be include in the bundle if present with other cables using the same connector.

MIL-STD-461F presents specific requirements for the application of the CS116 environment to cable bundles, power cables and high side power leads – proper attention to these details will assure correct translation of the system environment to the cable/connector assemblies. Figure 2.3.3-3 describes the CS116 environment in terms of an electrical model with equations that can be used to simulate the environment as applied to cable bundles, power cables and high side power leads.

Note that Figure 2.3.3-3 contains three schematics. The first is for the establishment of the source characteristics that will produce the required damped sinusoid current pulse into the base 100 Ohm calibration loop. The second two schematics represent the interference as applied to the cable bundle of interest. If the cable bundle has a gross over-shield (often implemented with a MIL QQB575R braid shield and thus referred to as a “gross overbraid shield”), then injection of the interference would be viewed as direct coupling to the over-shield conductor only. Characteristics of the over-shield will determine coupling from it to the conductors it covers. The second schematic in Figure 2.3.3-3 is also applicable to individual high side power leads.

The third schematic in Figure 2.3.3-3 is applicable to a multiple conductor cable bundle that does not contain a gross over-shield. In this case, each conductor, or the shields on the shielded cables that make up the outer conductors of the cable bundle, would be subjected to the direct injection of the environment. The total bundle injected current would divide amongst the outer conductors based on their terminating impedances and other electrical characteristics.

Per the equations in Figure 2.3.3-3, the environment model essentially applies current to a cable bundle, a power cable, or an individual lead, through an injection probe, driven by a source initially set to produce the current waveform described in Figures 2.3.3-1 and 2.3.3-2 into a 100 Ohm loop. That source level is reduced, if required, to the point where the resulting peak cable bundle, cable or lead current is limited to the value shown Figure 2.3.3-2. The injection probe model shown in Figure 2.3.3-3 produces an insertion loss that is compliant to the MIL-STD-461F requirements and is representative of what would be expected to be used in an EMI test lab. The circuit model of the probe represents the ETS Lindgren models 95236-1 and 95242-1 injection probes.



NOTES: 1. Normalized waveform: $e^{-(\pi f t)/Q} \sin(2\pi ft)$

Where:

f = Frequency (Hz)

t = Time (sec)

Q = Damping factor, 15 ± 5

2. Damping factor (Q) shall be determined as follows:

$$Q = \frac{\pi(N - 1)}{\ln(I_p/I_N)}$$

Where:

Q = Damping factor

N = Cycle number (i.e. $N = 2, 3, 4, 5, \dots$)

I_p = Peak current at 1st cycle

I_N = Peak current at cycle closest to 50% decay

\ln = Natural log

3. I_p as specified in Figure CS116-2

Figure 2.3.3-1. CS116 Waveform

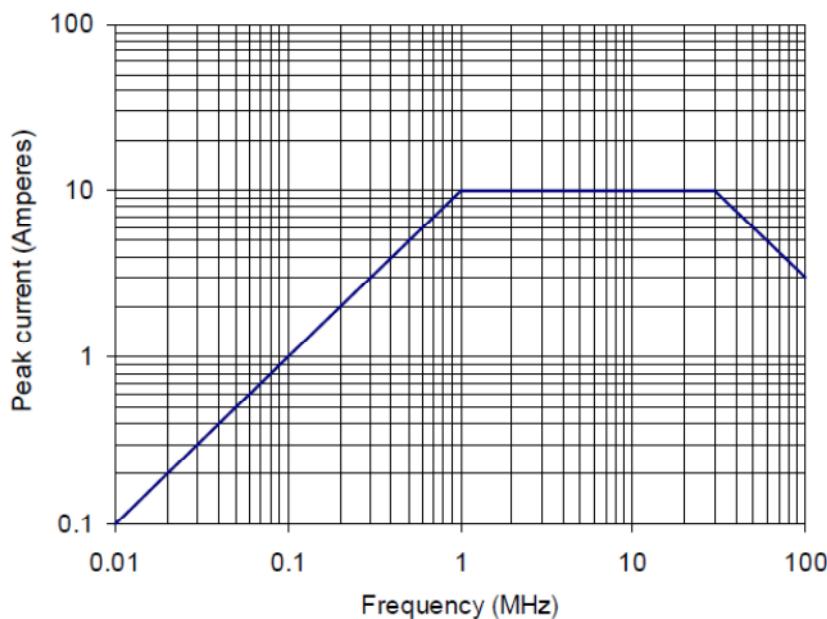


Figure 2.3.3-2. CS116 I_p Limit

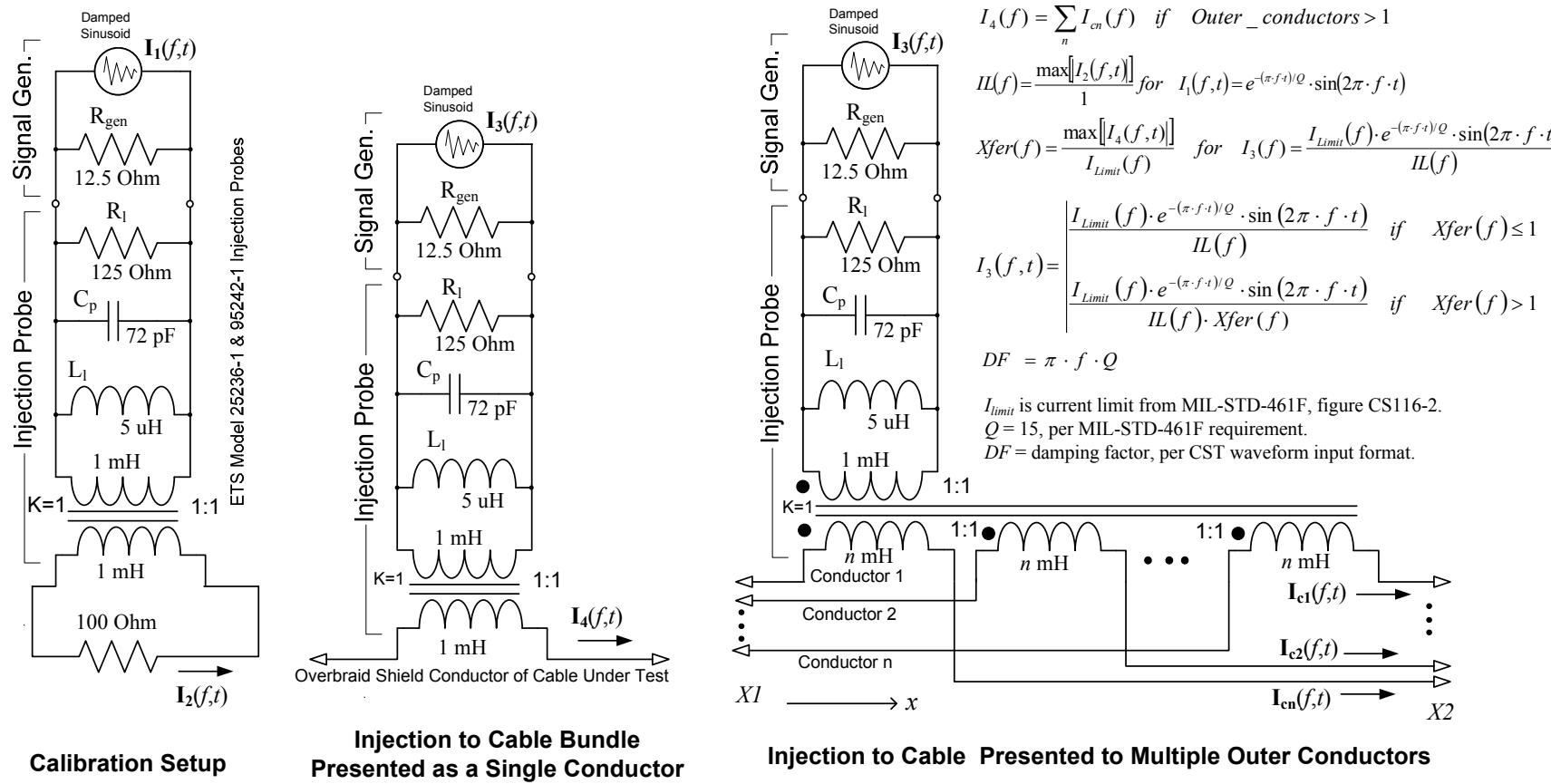


Figure 2.3.3-3. CS116 Environment Model

2.3.4 Radiated Susceptibility- Electric Field - RS103 Environment Model

After consideration of the EME threat and how the fields are attenuated and altered by a typical ground vehicle structure between the threat and the equipment and subsystem, as well as measurements of field strengths present at subsystems and equipment due to radiation from on-board radio transmitter antennas, MIL-STD-461F specifies a baseline E-field environment of 50 V/m from 30 MHz to 18 GHz at equipment and subsystems. This environment is specified under a particular test setup for application of the 50 V/m field.

Although the system level EME environment specified in MIL-STD-464C extends down to 2 MHz, and many ground vehicles do contain on-board HF radio transmitters (2 to 30 MHz), there is no radiated E-field requirement for the 2 to 30 MHz frequency range in MIL-STD-461F. This is because equipment and subsystems, if susceptible to this environment in the 2 to 30 MHz frequency range, would be most likely be susceptible via the field inducing current into the interconnect cables, not via pickup at a circuit card. The CS114 environment, conducted RF susceptibility, considers the MIL-STD-464C EME and on-board transmitter threats in the 2 to 30 MHz frequency range.

As noted in MIL-STD-461F, the base E-field threat levels present at subsystems and equipment may be tailored by the procuring activity based on particular situations. MIL-HDBK-235 is cited for guidance. The MIL-STD-461F RS103 50 V/m environment is pulse modulated with 50% duty cycle and a 1 kHz pulse rate. The MIL-STD-464C radiated environment for ground systems above 1 GHz is also pulse modulated, but with a duty cycle notable shorter than 50%.

For evaluation against the RS103 environment, the cable bundles must be representative of the intended installation. “Representative” equates to identical cable and wire cross-section, identical or electromagnetically equivalent connector assemblies, identical wire and cable shield properties (if shielding is applicable), and identical, or equivalent, shield and wiring termination, routing, and grounding at the connectors.

The equipment subject to the RS103 environment must be electrically bonded to the copper bench in a manner that represents how it is bonded (or not bonded) to the host platform. Key quantities to consider in equipment bonding include DC resistance, contact locations, bonding conductor material and geometry (if applicable), and the amount of surface area at bond between conductive surfaces.

The MIL-STD-461F RS103 environment requirements are also normalized to a condition where the cables under test are placed 5 cm above a conductive ground plane bench. Length may be limited to 10 meters if the installation cable is longer than 10 meters. If the length is unknown, a compliant default length of 3 meters is typically used, as this allows compliance to other baseline conditions stated in MIL-STD-461F.

The cables must also be terminated with loads to simulate the electrical properties (impedance, grounding, balance, and so forth) present in the actual installation.

Power leads are to be between 2 and 2.5 meters long and terminated into a line impedance stabilization network (LISN). Figure 2.3.1-3 shows a schematic of the LISN. The purpose of the LISN is to provide a platform representative power system impedance to the equipment or subsystem being tested or analyzed.

In general, the RS103 environment baseline for exposure of cable bundles is to have at least 2 meters of cable bundle, starting at the equipment connector, exposed to the field with the direction of propagation normal to the run of the cable bundle. Thus simulation models should be set up to provide at least 2 meters of cable bundle running perpendicular to the direction of propagation. Looping the cable back and

forth such that additional lengths run parallel to the direction of field propagation should be avoided. For each direction of propagation (there may be many if a complex cable harness is being analyzed) the field must be applied with both horizontal and vertical polarization, assuming the plane of the cable run and ground plane bench are horizontal. From a simulation standpoint, this may be done by running the simulation twice for each direction of propagation – once with horizontal polarization and once with vertical polarization, or combining polarizations for each propagation direction by using an E-field vector 45 degrees from horizontal, setting the E-field magnitude at $\sqrt{2}$ x required base level. However, this is often argued as a worst case test or analysis since, in practice, the system cables will only see one polarization being dominant at the baseline level – the component in the other polarization being much less. However, the maximum amount of pessimism would be limited to $20\log(\sqrt{2}) = 3$ dB.

2.3.5 Radiated Susceptibility - Transient Electromagnetic Field - RS105 Environment Model

MIL-STD-464C states: “*The system shall meet its operational performance requirements after being subjected to the EMP environment. This environment is classified and is currently defined in MIL-STD-2169. This requirement (environment) is applicable only if invoked by the procuring activity. Compliance shall be verified by system, subsystem, and equipment level tests, analysis, or a combination thereof.*” Note that MIL-STD-2169 is a classified document. Often, if the EMP environment is applicable, subsystem and equipment specifications will contain detailed requirements with respect to required recovery time and actions and outputs during the recovery period. As an example, a mission display may be required to recover within 40 seconds after the EMP event and must never provide misleading or non-current data. The terms High-altitude Electromagnetic Pulse (HEMP) and Nuclear Electromagnetic Pulse (NEMP) are often synonymous with the term “EMP.”

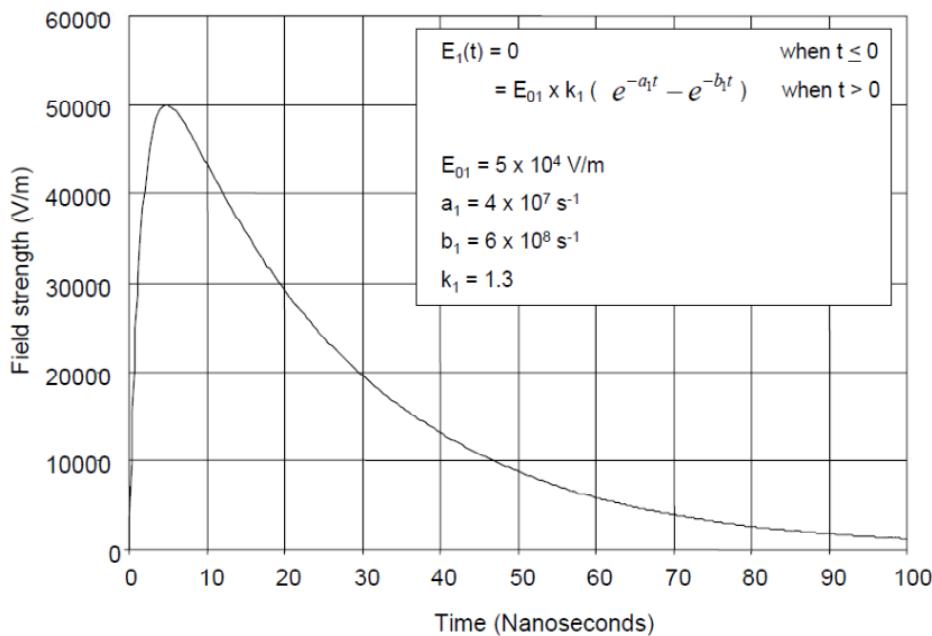
The RS105 requirement in MIL-STD-461F is an attempt to provide a representative unclassified compilation of various EMP environments that may be seen by subsystems and equipment in unshielded systems and platforms.

The RS105 requirement is technically applicable only to equipment enclosures, as the result of EMP onto cable bundles and power leads is covered under the generic CS116 environment. However, there may be instances where the installation cables are relatively short (< 10 meters) and it may be more practical, in lieu of applying CS116, to subject the cable harness to the RS105 environment along with the equipment enclosure. Examples include soldier mounted systems and small vehicles. In these cases test or modeling with respect to the RS105 environment may be more appropriate. Hence a model of the RS105 environment for cables is included herein.

The RS105 environment is a radiated planewave with the characteristics shown in Figure 2.3.5-1. Figure 2.3.5-2 is a photograph a RS105 test of a small subsystem. Clearly, this setup will expose cables to the RS105 environment.

With respect to short cables, the RS105 environment is typically modeled with the cables 5 cm above a ground plane bench, as with the other MIL-STD-461F setups, with the plane wave propagation in the direction of the cable run and the E-field set normal to the ground plane. For complex cable harnesses, the model may include application of the field with multiple directions of propagation (each applied as a separate case), always with the E-field normal to the ground plane.

Often when the RS105 environment is to be applied to equipment or subsystem cables, the cable and equipment layout on the ground plane used for RS103 is directly used for application of the RS105 environment – a parallel plate transmission line plate is placed above the ground plane over the equipment and cables.

**Figure 2.3.5-1. RS105 Waveform****Figure 2.3.5-2. RS105 Setup Example for a Subsystem**

3. CABLE/CONNECTOR ASSEMBLY MODELING METHODOLOGY

The object of the cable/connector assembly modeling and simulation process described herein is to determine voltages and currents at key circuit interfaces resulting from the cable/connector assembly's exposure to the MIL-STD-461F environments described in Section 2. These voltages and currents can then be compared to the characteristics of the expected desired signal to allow determination of expected susceptibility or immunity to the environment.

With respect to the use of commercially available state-of-art electromagnetic effects simulation tools, the recommended modeling and simulation approach is to break up the model problem into entity blocks. The models for these entity blocks can be built as stand-alone items and merged together in a top level co-simulation framework using supported formats to allow interface between the entity blocks.

Entity blocks include cable harnesses modeled as a chain of transmission line matrices, connectors and mechanical aspects of shield and wire terminations modeled as a S-parameter blocks as a result of full-wave 3-dimension (3-D) EM simulations, and I/O circuit blocks consisting of discrete components, SPICE models and S-parameter blocks. "Probes," which allow measurement of voltages and currents, can be placed within I/O circuit blocks and at connection points between entity blocks.

Of course each of the electromagnetic environments, as described in Section 2 above, would be considered an entity block as well. However, only one at a time would typically be used.

The entity block approach allows model reuse and allows construction of models for various parts of the problem simultaneously and independently to provide greater throughput than previously realized before the availability of co-simulation features in electromagnetic effects analysis tools.

3.1 Model Entity Blocks

Figure 3.1-1 presents the various top-level entities, or sub-model blocks, that may be used in a co-simulation setup to determine the effects of the of MIL-STD-461F susceptibility environments applicable to cable/connector assemblies. Each entity block can be viewed as a separate stand-alone model. The entity models (blocks) interface via electrical connections where the block edges touch each other. Interconnection of "grounds" between blocks is implemented as electrical connections in the co-simulation setup as well. The interconnection of block "grounds" in the co-simulation setup may occur through networks built from resistors, inductors and/or capacitors to model the "real" aspect of grounding and bonding.

Table 3.1-1 provides a description and specification/build methods for each type of entity block that can be used in a co-simulation setup for the prediction of susceptibility with respect to the MIL-STD-461F environments as applied to cable/connector assemblies. The entity block descriptions, modeling methods and interfaces presented are compatible with the CST Studio Suite⁴ simulation tool set. Other vendors offer tools with similar features and capabilities, although some of the semantics and problem solving specifics may vary slightly. CST Studio Suite was chosen for this effort due to its current in-house presence.

⁴ CST Studio Suite is a collection of simulation tools that includes Design Studio, Cable Studio, PCB Studio, Microwave Studio and various time domain and frequency domain solver modules, all capable of interface with each other in a co-simulation environment.

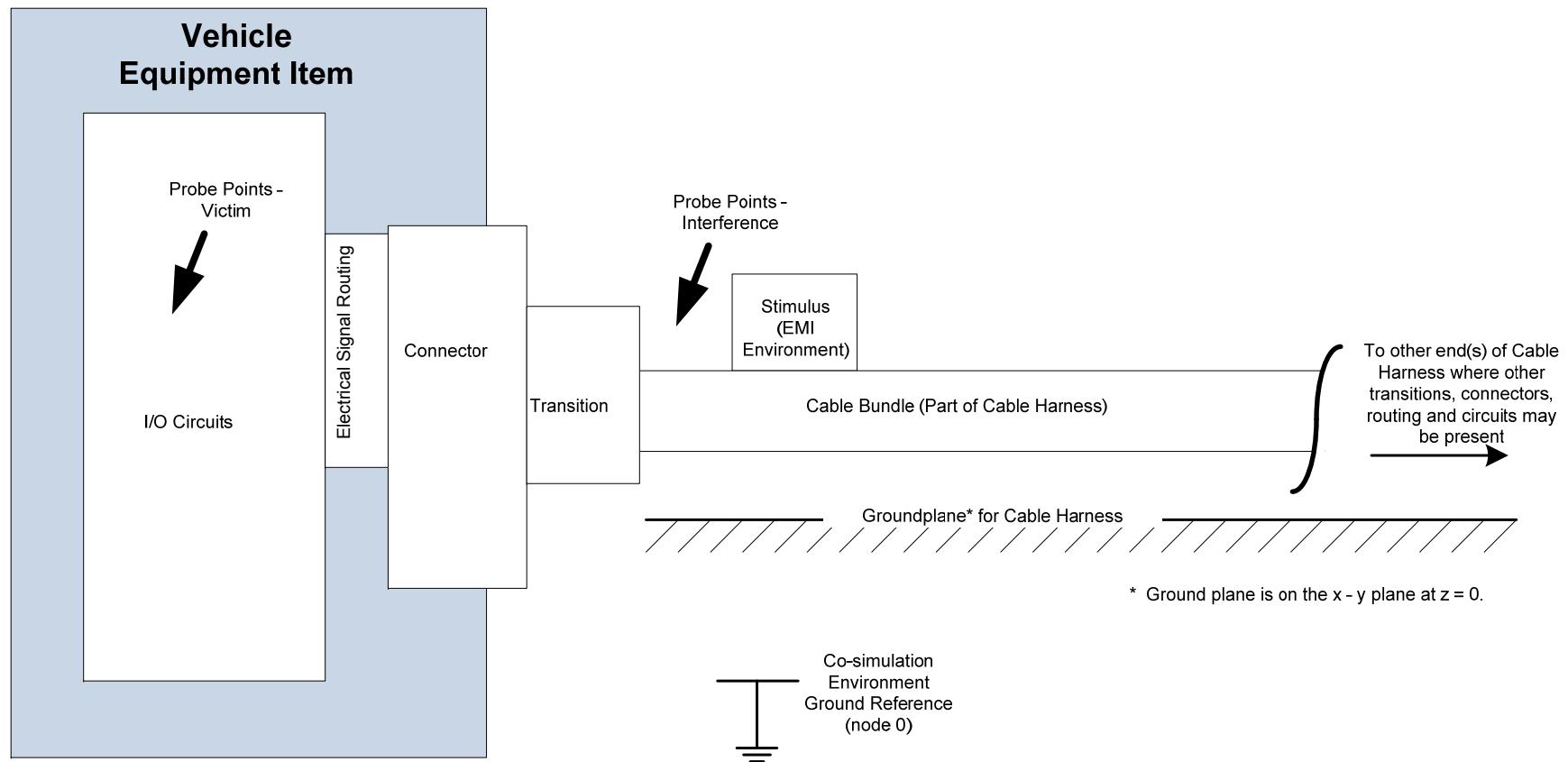
**Figure 3.1-1. Model Entity Blocks**

Table 3.1-1. Description and Specification of Model Entity Blocks

Entity Block	Description	Method of Specification	Notes
Cable Harness	The collection of wiring, to include wiring shields, that will be exposed to the electromagnetic environment, either directly or indirectly, as bounded and geometrically laid out above a ground plane per the conditions of MIL-STD-461F, or tailoring thereof.	<p>Geometry entered into a 2.5-D cable EM simulator (e.g. CST Cable Studio) by specifying the following:</p> <p>Nodes: Points in 3-dimensional space that form a cable route path.</p> <p>Routes: Connected segments that form a path for a cable bundle to be placed on. Routes have two end nodes.</p> <p>Cables and Cable Groups: One or more conductors with a particular cross-section, which may include shields. Shields can be assigned a transfer impedance.</p> <p>Cable Bundle: Collection of one or more cables that share the same route and are electromagnetically coupled by default.</p> <p>Signal: A unique name assigned to each conductor in a cable bundle.</p> <p>Ground Plane: Conductive surface typically at z = 0.</p> <p>Additional Optional Items:</p> <ul style="list-style-type: none"> Connector (ideal): Assignment of an ideal connector at ends of cable bundles to allow connection of signals. Junction: Ideal connection of pins from more than one connector. 	<p>After 2.5-D simulation, the cable harness will be presented in the co-simulation framework as a block with the terminals of the cable signals that do not connect to another cable signal within the cable model space.</p> <p>As with other conductors in a cable bundle, the shields are also assigned a signal name.</p> <p>A signal has two terminals: each terminal will connect to either one or more signal terminals within the cable simulation space, or, if unconnected, will be available as a terminal for external connection outside the cable model space.</p> <p>The Cable Studio tool supports the modeling of shielded cables. It does not have the capability to apply a shield over a cable bundle. Thus shielded cable bundles will be limited to one cable group – a cable group that is shielded.</p> <p>Capability exists for automatic computation of shield transfer impedance if shield details are entered.</p>
Connector	Model of a real connector assembly that typically includes the part attached to the equipment item, the part attached to the cable bundle and the backshell/strain relief.	MCAD file import of geometry and/or manual entry of geometry followed by material property assignment in 3-D EM modeling tool (e.g. CST Microwave Studio). Result after 3-D simulation will be an S-parameter block.	The radiated environment is not presented to the connector.

Entity Block	Description	Method of Specification	Notes
Transition	Model of the interface between the constant cross-section cable bundle and the connector pins.	Model built from primitives in a 3-D EM modeling tool (e.g. CST Microwave Studio). A port is assigned to each signal and shield conductor (if applicable) at each end of the transition. Result after 3-D simulation will be an S-parameter block.	The radiated environment is not presented to the Transition.
Electrical Signal Routing	The routing inside an equipment item between the connector, transition or cable bundle interface, as applicable, to the I/O circuits of interest.	<p>S-parameter block resulting from one of the following build and simulation activities. Note that a port is assigned to each signal and shield conductor at each end of the routing structure. Multiple methods of specification:</p> <ol style="list-style-type: none"> 1) Model built using a cable harness a 2.5-D cable EM simulator tool (e.g. CST Cable Studio) 2) Model built using a 2-D or 3-D PCB EM simulator tool (e.g. CST PCB Studio) 3) Model built from primitives in a 3-D EM modeling tool (e.g. CST Microwave Studio). 	The radiated environment is not presented to the Electrical Signal Routing.
I/O Circuits	Loads and terminations for conductors contained in a cable bundle which are not directly terminated to the connector or to the connector at the transition structure. Also includes networks that connect circuit grounds to 3-D cable harness problem space ground, as well as networks that connect to the co-simulation framework ground reference (node 0).	<p>Electrical schematic items consisting of one or more the following entered in the co-simulation framework (e.g. CST Design Studio):</p> <ol style="list-style-type: none"> 1) Passive components 2) Active components⁵ 3) SPICE model blocks (active and passive devices)⁵ 4) Passive S-parameter model blocks 	<p>SPICE model block files will have to be modified if it is not desired to have “node 0” in the SPICE model be coincident with the ground reference (node 0) of the co-simulation framework.</p> <p>The reference of the S-parameter block will be the ground reference of the co-simulation framework.</p>
Stimulus - CS	Application of MIL-STD-461F CS114, CS115 or CS116 environment to cable harness.	Electrical schematic items, including probes, as described in Sections 2.3.1, 2.3.2 or 2.3.3, inserted into the co-simulation framework (e.g. CST Design Studio) between a cable bundle’s end node and transition block.	The stimulus is typically applied to various combinations of cable conductors at each equipment item connector-cable interface node, in turn, per Sections 2.3.1 through 2.3.3.

⁵ Some simulators, such as CST Design Studio, cannot perform successfully perform transient analysis in a co-simulation environment unless particular passivity criteria is met. In these cases, active components may need to be represented as passive linearized network.

Entity Block	Description	Method of Specification	Notes
Stimulus - RS	Application of MIL-STD-461F RS103 or RS105 environment to cables within the cable simulation space.	Planewave inserted into the cable simulation space (e.g. CST Cable Studio) per the conditions stated in Sections 2.3.4 and 2.3.5.	The RS103 stimulus is applied twice – y-direction propagation, hor. & vert. polarization. The RS105 stimulus is applied twice – propagation in the x and y directions, vertical polarization.

It is important to realize that not all of the entity model blocks listed in Table 3.1-1 are required to model a particular cable/connector harness. For instance, if the pins of a connector are present on the same circuit card as the I/O circuits of interest, and the routing is, from a practical sense, electrically insignificant, the Electrical Signal Routing block would not be needed.

It should also be noted that multiple entity blocks may be combined into one block if they are connected and the modeling format for both are the same. For example, at the end of complex cable bundle, conductors of twisted shielded pairs may become untwisted as they transition to the connector pins, and shields may be terminated to the connector backshell via wire pigtails. This transition may be modeled as sections of cable with varying cross-sections, or it may be modeled as a set of conductors and insulators in 3-dimensional space in the same manner as the connector assembly. If the later method will be used, it may be easier to build the connector and transition into one combined 3-dimensional entity block to represent the whole structure.

Note that Figure 3.1-1 only shows one end of a cable bundle. Typically, a cable harness applicable to the MIL-STD-461F environments defined in Section 2 will have at least one other equipment item or bulkhead panel, complete with a connector, I/O circuits, etc., connected to the other end.

3.2 Grounds and Grounding

A co-simulation problem may contain several “grounds.” The details of the electrical bonding between these grounds can greatly influence the ability of a cable assembly to provide immunity to the electromagnetic environment. Hence it is extremely important to capture the grounding and bonding attributes of the assemblies being modeled. Ideally, every item (objects, wires, cables) would be modeled in a 3-D electromagnetic space. However, this is simply not practical. But, with some insight from experienced EMC practitioners, reasonable modeling approximations can be made. (There will be some illustrative examples included in the following sections that show this.)

3.2.1 CST Studio Suite Electromagnetic Effects Simulator Overview

The most important item to understand is how the grounding scheme is set up in the co-simulation framework. The setup of entity block ground in the CST Studio Suite co-simulation framework will be presented. Other co-simulation tools will utilize similar grounding setups.

CST Studio Suite is a set of embedded simulation modules that are integrated for use with a common graphical user interface, or “front end,” that have the capability to operate together in a co-simulation setup. CST, from a sales standpoint, labels the modules as “studios,” each focused on a particular type of simulation problem. The various “studio” modules, each which can be purchased separately, include Microwave Studio, Design Studio, PCB Studio, Particle Studio, Mechanical Physics Studio and Cable Studio. Additionally, some of the modules provide the capability to use multiple solvers, of course each being a separately purchased item. For example, Microwave Studio can support Time Domain, Frequency Domain, Integral Equation and Eigenmode solvers.

Simulation problems involving the application of MIL-STD-461F susceptibility environments to cable/connector assemblies require the use (purchase) of the following CST Studio Suite components:

- (1) Frontend
- (2) Design Studio
- (3) Cable Studio
- (4) Microwave Studio
- (4a) Time Domain Solver.

3.2.2 Grounding within CST Studio Suite

Cable Studio is designed to be used as an embedded module within Design Studio. Design Studio is basically a schematic entry, results presentation and circuit simulation tool which provides the framework for a co-simulation environment. Voltage and current probes are inserted into Design Studio to allow monitoring of voltages and currents. Voltage and current probes cannot be inserted into Cable Studio or Microwave Studio.

Cable Studio allows the user to define complex cable harnesses with objects, such as a ground plane, which may be present in the cable’s environment. Per MIL-STD-461F, it will be assumed herein that the cable harness is routed above ground plane. During the modeling process, Cable Studio generates transmission line matrix equivalent circuits from the geometry of the cable and surrounding objects. The result is a “cable” block in Design Studio that provides terminals for connection to the cable harness’s terminal conductors. Terminals for connection to the ground plane or other non-cable conductors in the cable harness’s problem space at a particular location may also be made available. These locations are known as “connect to 3-D” nodes.

In Design Studio, cable block terminals can be connected to circuit elements, SPICE model blocks, S-parameter (Touchstone) blocks and IBIS blocks to complete the co-simulation model. It should be noted that the connector and transition entity blocks are created by modeling these structures in Microwave Studio and importing them into Design Studio as an S-parameter block.

The circuit ground in CST Design Studio is the reference ground for the co-simulation environment and is hence referred to herein as “node 0.” It is also the “node 0” that is inherent in SPICE model files. S-parameter blocks in Design Studio, whether imported from electronic component S-parameter files or from a Microwave Studio model, are also referenced to the Design Studio node 0.

It should also be noted that the metal structures, such as ground plane under the cable harness, in the Cable Studio problem space are also virtually referenced to the Design Studio node 0. Of course grounded nodes (connect to 3-D) allow connection to the metal structure at the node’s location either directly, or indirectly through electronic components, to the Design Studio node 0 reference ground. Hence careful planning is required if it is desired to simulate “floating” interface circuits or returning cable shield currents through the cable’s reference ground plane.

Figure 3.2.2-1 is a notional diagram showing how various blocks and circuit elements for a simple 3-conductor unshielded cable harness routed over a ground plane may appear in Design Studio. For simplicity, the connectors on the cable were deemed insignificant and thus not included. Also shown in this figure are the contents and grounding within the SPICE and S-parameter blocks that interface with the cable harness. The co-simulation in Design Studio allows determination of the voltages and currents at P1, P2 and P3 as a function of the current at P1.

In this particular example, the model of the three microstrips above a ground plane was set up in Microwave Studio with the S-parameter ports connected between strip ends and the ground plane below them to result in a 6-port block. The other S-parameter block, which is from an RC network built on top of a ground plane, was a Touchstone file output from a vector network analyzer measurement. The SPICE block is typical of what is provided by a parts vendor. By default, node 0 (circuit ground reference) within the SPICE model is not passed directly as a block terminal, but is automatically connected to the Design Studio ground node 0 reference. The same is true for the reference ground used for the ports’ references in the S-parameter blocks – that reference ground is automatically attached to the Design Studio reference ground. In this example, injected interference current at P2 will not return via the cable environment’s ground plane. This may cause incorrect results if this is not representative of the compliance test setup.

Figure 3.2.2-2 is a notional diagram of the same cable and “equipment interface” model as shown in Figure 3.2.2-1, except that the interface on the left side of the cable is referenced to the cable environment’s ground plane at the left end of the cable via a grounded node. Hence the return current resulting from current P2 will flow through the cable environment’s ground plane. Note that the addition of this feature does add complexity to the simulation in that a reference ground plane and two additional ports need to be added to this electrically long 3-D microstrip model. In the case of the microstrip structure, the added ground plane must be far enough away from the three strips such that influence on coupling and characteristic impedance is minimal, but not too far away to make the model space unbearably large, which may result in very long simulation times.

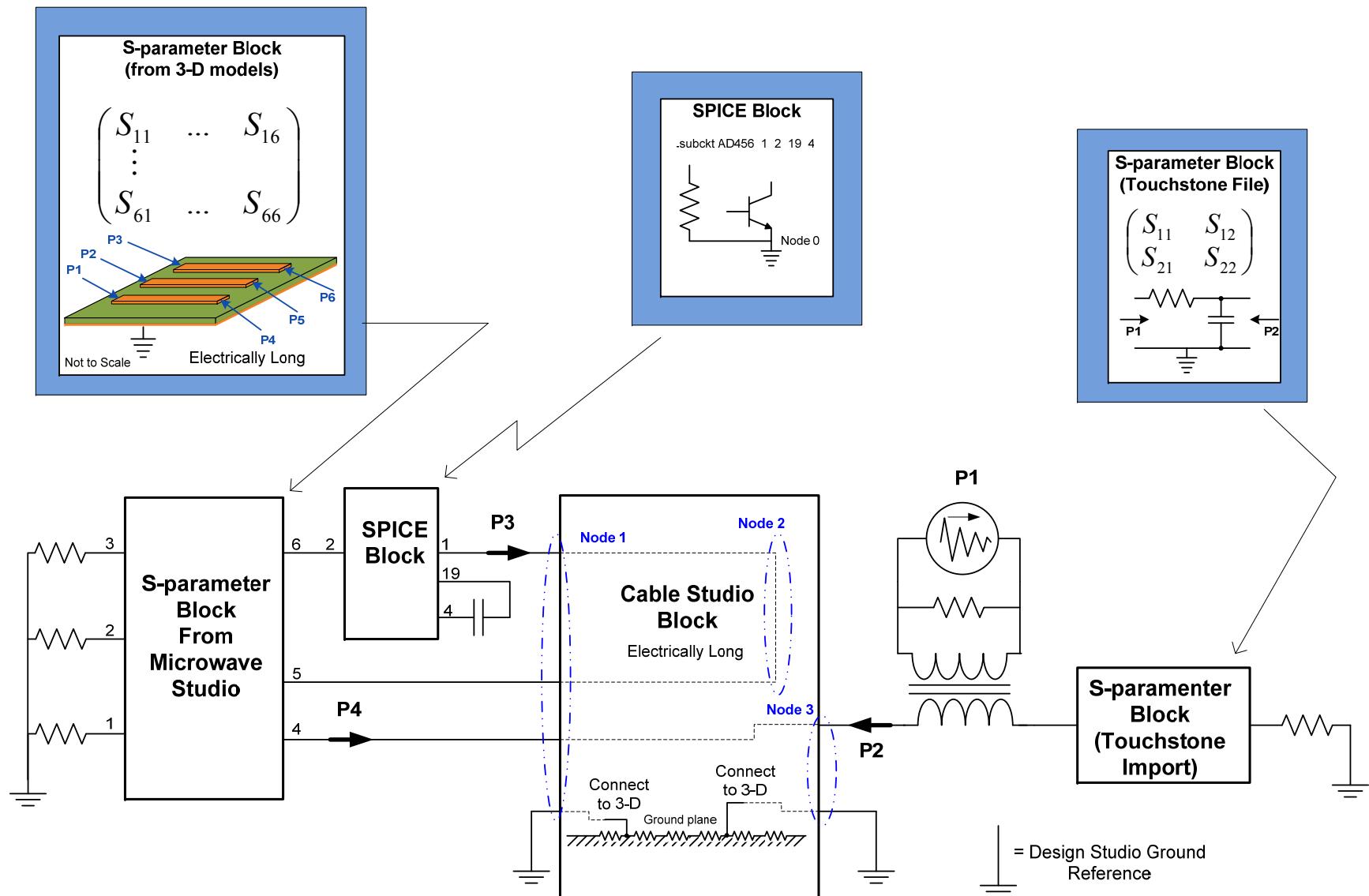


Figure 3.2.2-1. Grounding Scheme #1 in Co-simulation Environment for Simple 3-Conductor Cable above Ground Plane

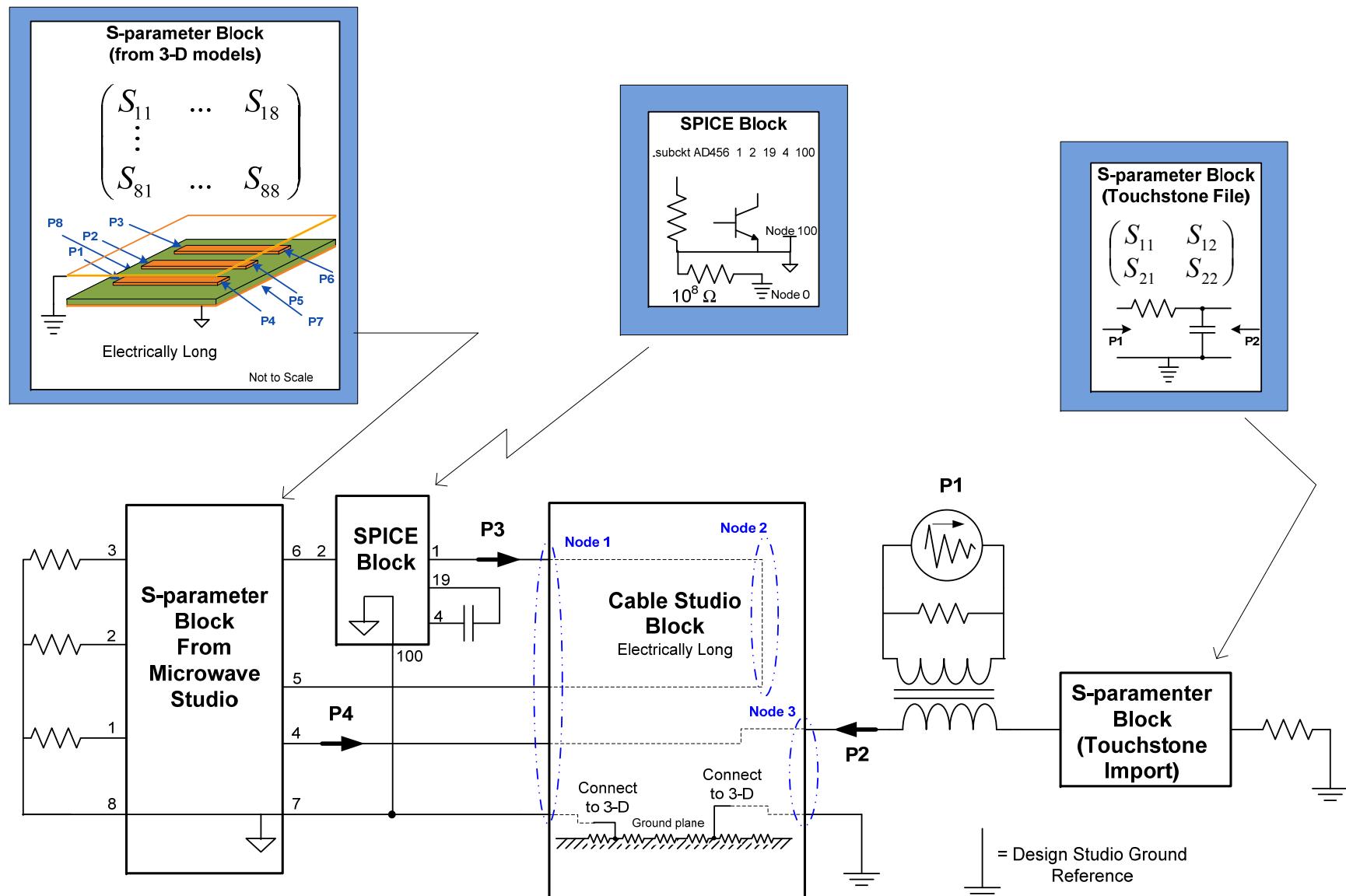


Figure 3.2.2-2. Grounding Scheme #2 in Co-simulation Environment for Simple 3-Conductor Cable above Ground Plane

The SPICE Block file will also have to be modified to allow access to the circuit's reference. In this example, this is done by replacing the node "0" node connections with connections to node "100," or some other unused node number, and then adding a net consisting of a high impedance between node 100 and node 0. When compared to the SPICE block in Figure 3.2.2-1, we note that the new block has five terminals free for connection instead of four.

As a final note to the example shown in Figure 3.2.2-2, we do observe that there is a hard connection to the Design Studio co-simulation reference ground on the right side of the cable. This is highly acceptable when the components involved are electrically short, as one would expect to be the case for the injection transformer and RC termination at the right end of the cable. Although not explicitly documented in CST Studio user manuals, experience has shown that the Cable Studio block should have a hard connection to the co-simulation framework's reference ground to ensure a greater chance of simulation stability.

3.3 Cable Harness Definition

A vehicle will typically contain many complex cable harnesses and equipment items. The vehicle, with the complete collection of cable harnesses and equipment items is applicable to the MIL-STD-464C electromagnetic environments. However, from an equipment and cable/connector assembly design and procurement standpoint, the MIL-STD-461F environments are applicable at the equipment and subsystem level, with the cables connected to this equipment being applicable to the same MIL-STD-461F environment as the equipment items. MIL-STD-461F contains three environments where the environment is applied to the cables at the equipment connectors. These environments, CS114, CS115 and CS116 are discussed in Section 2. Additionally, the MIL-STD-461F RS103 and RS105 radiated environments may be applied to the cables harness as well. These environments are also discussed in Section 2.

It is the MIL-STD-461F environments applicable to the cable/connector assembly that is the focus of this effort. Hence a whole vehicle cable harnesses will not be included into one model problem. Rather, only the cables connected to a procurable line replaceable equipment item or subsystem, which typically consists of one to six equipment items, will be included in a particular model problem. Furthermore, the cable lengths and positioning above a ground plane will be per the MIL-STD-461F requirements. With this in mind, the Cable Harness entity block should be built to emulate a MIL-STD-461F test setup.

3.3.1 *Ground Plane*

Cable harness definition starts with defining the ground plane. Typically, the ground plane will be defined as being normal to the z-axis at $z = 0$.

3.3.2 *Nodes*

"Nodes" are points in 3-D space that represent the ends of a cable bundle and the points where the cable bundle run changes directions. The z-coordinate of nodes is typically 50 mm per the "5 cm above the ground plane" requirement stated in MIL-STD-461F. However, since cable cross-sections are centered on nodes, the z-coordinate may have to be increased by the cable's cross-section radius if it is greater than 5 mm to allow tolerable compliance with MIL-STD-461F cable height conditions.

3.3.3 *Segments, Traces, and Routes*

A "segment" is defined as a *path* between two nodes. A "trace" is a **path** that connects three or more nodes. A "route" is a *path* defined as a set of contiguous segments or traces.

3.3.4 Cables and Cable Groups

A “cable” is a finite length of a set of one or more conductors with insulators and/or shields, as applicable, possessing a particular cross-section. CST Cable Studio contains templates for four types of cable cross-sections: single wire, coax, ribbon and twisted pair. Conductor and insulator characteristics and dimensions can be defined for each type to define a particular cable. For coax cables, CST Cable Studio allows the definition of circular and wrapped shields (screens) of the braided and solid varieties. For braided shields, CST Cable Studio allows use of a simplified model where the user specifies transfer inductance, resistance and capacitance, entry of Kley’s model information, or entry of measured results for the determination of transfer impedance.

Basic “cables” can be combined, with layers of insulation and shielding added to build “cable groups.” It is at the cable group level where gross overshields are added.

3.3.5 Cable Bundles

A “cable bundle” is a set of one or more cables or cable groups which share the same route and thus electromagnetically coupled by default. “Cables” and “cable groups” are laid on to “routes.” CST Cable Studio has the capability to automatically bundle cables and cable groups that share a route.

From a MIL-STD-461F cable environment simulation standpoint, a “cable bundle” will typically be associated with a physical equipment connector at one or both ends. Thus an equipment item containing three connectors will require a cable harness that possesses at least three cable bundles for interface. The MIL-STD-461F CS114, CS115, and CS116 environments are typically applied on an equipment connector basis.

3.3.6 Cable Harness

A “cable harness” is the collection of cable bundles within a Cable Studio cable block. It is the complete cable harness that will be exposed to MIL-STD-461F RS103 and RS105 planewave source environments.

Electromagnetic coupling between cable bundles within a cable harness can also be considered in CST Cable Studio when the Transmission Line Matrix (TLM) meshing parameters are set up for specification of a “maximum search distance.” In general, for the MIL-STD-461F focused cable analysis and simulation, coupling between cable bundles is typically of little concern from a practical standpoint.

3.3.7 Shielded Cable Bundles

CST Cable Studio does not offer the capability of adding an overshell to a cable bundle as it is laid on to a route. Thus if it is desired to have a bundle with an overshell, that bundle will consist of only one cable group; it is that cable group that will contain the overshell.

3.3.8 Signals and Signal Terminals

When cables and/or cable groups are laid on to routes, a unique “signal” name is automatically formulated for each conductor using that route. By default the signal name is a concatenation of the following items, in sequence, separated by an underscore:

- route
- cable or cable group
- instance of the cable or cable group if a particular cross-section is used more than once on a route
- conductor number within the cable or cable group.

Unfortunately, the default signal names can be long and unwieldy. However, they can be renamed by the user. And, nodes, routes, cables and cable groups can be manually named at the time of creation to provide optimal description.

Signals have two “terminals.” By default, terminals have names that start with the terminating node name followed by the signal name.

Note: Cable and cable group shields (screens) conductors are considered “signal conductors” and will thus have signal names, as well as possessing a terminal at each end.

3.3.9 Connectors, Plug-ins, Pins, and Junctions

Within the CST Cable Studio, terminals for signals contained in cable bundles that make up a cable harness are joined via “connectors.” These “connectors” are actually electrically ideal connections as opposed to an electromagnetic representation of the physical connector that may be present in a real cable harness. In most situations with respect to a frequency range of interest limited to 400 MHz, this is acceptable for a typical junction within a cable harness. However, if desired, there are various methods to insert a 3-D electromagnetic model of a physical connector into the simulation – most commonly it is built and modeled with ports that are connected to cable signal terminals in the co-simulation framework. It can be also inserted as a 3-D structure model in the Cable Studio environment as well, but, the former is usually easier. Of course the later will be required if the junction is electrically significant in the sense that it realistically should be exposed to the radiated interference environments.

The CST Cable Studio “connectors” can be subdivided into multiple “plug-ins” to allow more straightforward grouping of signals.

The use of CST Cable Studio “connectors” starts with placing one or more “connectors” at the node where signal terminals at the ends of cable bundles need to connect. Assignment of multiple “plug-ins” can be made within this “connector” if desired. (Per the CST Cable Studio convention, there will always be at least one plug-in per connector.) Signal “terminals” present at the connector’s node are then **linked** to connector plug-in “pins.” Multiple signal terminals may be linked to a connector plug-in pin, as this may be a simple way to connect signal terminals from different cable bundles that come together at a node.

There are two methods to connect signal terminals of interfacing cable bundles at nodes within a cable block. As alluded to above, the first method is to simply define one connector at the connection node and link terminals that need to be linked to a connector plug-in pin. The second method would be to define a “connector” at end of **each** cable bundle and link the bundle’s signal terminals to the connector plug-in pins. A “junction” would then be defined to *link* connector plug-in pins. In summary, “junctions” link connector plug-in pins. Connector plug-ins pins are linked to signals.

3.3.10 Cable Block I/O

A signal terminal that does not connect to another signal terminal via connection to a common connector plug-in pin or via a “junction” will be made available as a cable block I/O when it appears in the Design Studio co-simulation framework. It will be presented with the same terminal name as used in Cable Studio.

3.3.10.1 Connection to 3-D

In a practical sense, connection of circuit elements, cable shields and cable signal conductors to the ground plane are often made where the cable will interface to physical connectors and equipment. Hence CST Cable Studio provides the opportunity for connection to the ground plane at a particular node (physical location) through the use of a “grounded node” designation option when the node is defined. A “grounded node” will present a cable block I/O terminal that connects to the ground plane at the node location when presented in the Design Studio co-simulation environment. As an example, nodes 1 and 3 in Figures 3.2.2-1 and -2 are “grounded nodes” which allow connection to the ground plane at their location.

3.4 Cable Shield Transfer Impedance

Computational methods to determine the coupling of an electromagnetic environment through shielded cables make use of the shield’s transfer impedance quantity.

For frequencies below 1 GHz, the quality of a cable shield is typically defined by its transfer impedance. The transfer impedance is defined as the ratio of the open circuit voltage developed between the shielded conductor and the shield to the current flowing on the shield. The units for transfer impedance is ohms per meter. At low frequencies (typically below a few 100 kHz) the transfer impedance is a resistive quantity. As frequency increases, the transfer impedance is dominated by an inductive component. Thus transfer impedance a complex quantity that varies with frequency.

Below 1 GHz, the transfer impedance of single layer braid cable shields can be simply modeled as a series resistance and inductance. Most electromagnetic effects analysis tools, including CST Cable Studio, allow the entry of a per unit resistance and per unit inductance to describe transfer impedance. This simple model is often adequate for single layer braid shields for analysis over the 10 kHz to 400 MHz frequency range. It should be noted from a practical standpoint that resistance itself is often a function of frequency due to the skin effect, which may be of concern if high accuracy is needed for analysis below 1 MHz. Table 3.4-1 lists transfer impedance in terms of resistance and inductance for a few popular coax cables with braid shields.

Table 3.4-1. Simple Transfer Impedance Model Parameters for Popular Braid Coax Shields

Cable / Shield	Xfer Resistance	Xfer Inductance	Optical Coverage
RG-6 Inner Shield	6.6 mΩ/m	0.42 nH/m	95.6
RG-6 Outer Shield	7.5 mΩ/m	0.36 nH/m	96.2
RG-11	4.0 mΩ/m	0.25 nH/m	96.0
RG-58	14.2 mΩ/m	1.0 nH/m	93.6
RG-213	4.0 mΩ/m	0.25 nH/m	96.0
RG-174	36.5 mΩ/m	2.3 nH/m	86.3
RG-218	1.2 mΩ/m	0.07 nH/m	98.3
RG-316	26.8 mΩ/m	0.88 nH/m	92.3

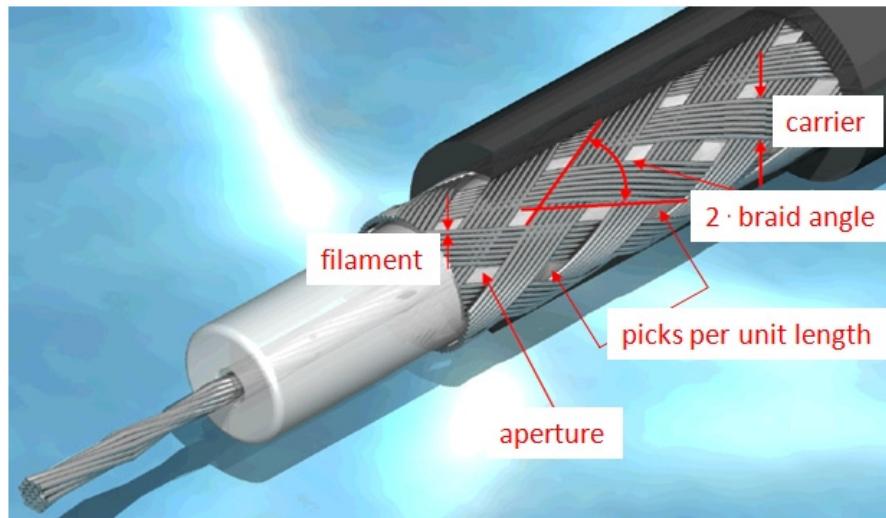
Transfer impedance of single layer braided shields can more accurately be calculated using Kley’s model / method⁶. Many EMC focused cable analysis simulation tools, including CST Cable Studio, offer the option of computing transfer impedance using Kley’s model. Kley’s model allows computation of shield transfer impedance a function of the braid’s physical and geometrically features, such as the shield

⁶ Details on Kley’s model and method for transfer impedance computation can be found in *EMC Analysis Methods and Computational Models* by M. Tesche, et al., published by Wiley Interscience, New York. 1997.

diameter, braid angle, filament wire size, number of filaments per carrier and the number of carriers. Alternatively, some of these quantities can be related to optical coverage, which may be entered instead to allow computation of transfer impedance. Figure 3.4-1 illustrates braid shield terminology as it relates to Kley's model. Figure 3.4-2 shows a screenshot from Cable Studio for computation of transfer impedance via entry of Kley's model parameters for the RG-316 shield, noting the result could be closely modeled using the series resistance and inductance values given in Table 3.4-1 for RG-318.

The transfer impedance of shields containing foil, foil-braid combinations (Figure 3.4-3), or multiple layers of braid can only practically be determined by measurement. Figure 3.4-4 shows measurement data for shields on some popular data cables containing foil based shields. Figure 3.4-5 shows transfer impedance for some popular coax cable shields which contain one or two layers of braid shielding based on measurement. Note the downward slope in impedance for the RG-11U and RG-223U cables in the 1 MHz frequency region. This is due to the skin effect.

Transfer impedance can be measured using methods contained in IEC 62153-4-4, and other similar standards, which basically call for the injection of current at particular frequencies on to an electrically short section of cable shield while measuring the resulting voltage on the shielded conductor using a vector network analyzer with a special test fixture. Most computational electromagnetic simulation and analysis tools accommodate the use of measurement data by allowing the entry of shield transfer impedance as a set of complex impedance versus frequency data points, or a DC resistance and transfer inductance quantity.



The relationship between the three parameters can be expressed in algebraic formulas:

D: Diameter under the braid (defined by the isolator below)
d: Diameter of a single filament
N: Number of filaments in each carrier
C: Number of carriers
F: Fill factor
α : Braid angle
P: Picks per unit length
O: Optical coverage

$$F = N \cdot P \cdot D / \sin \alpha;$$

$$O = 2 \cdot F - F^2;$$

$$\tan \alpha = 2\pi \cdot (D+2d) \cdot P/C;$$

Figure 3.4-1. Terminology for Braided Shields

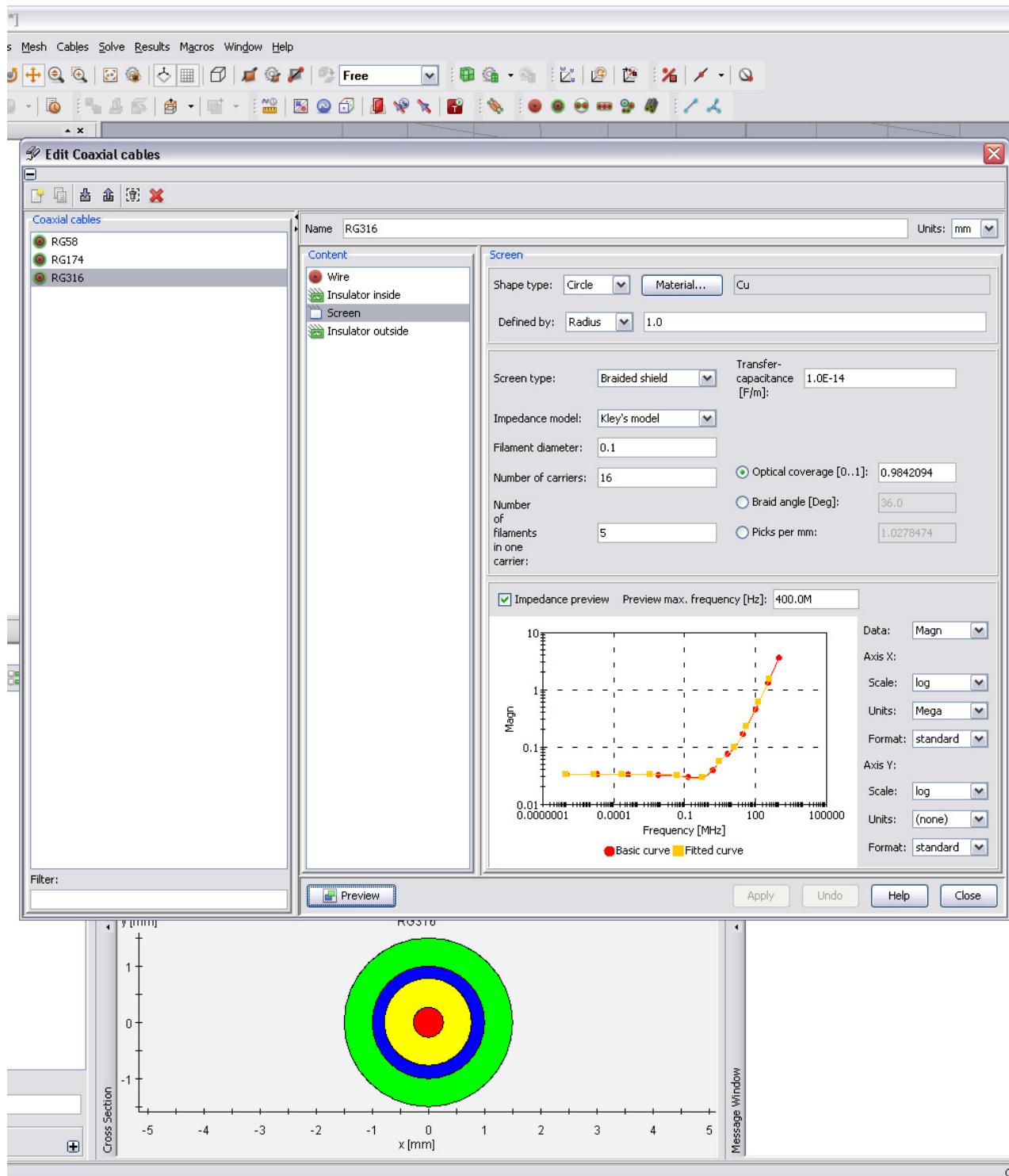
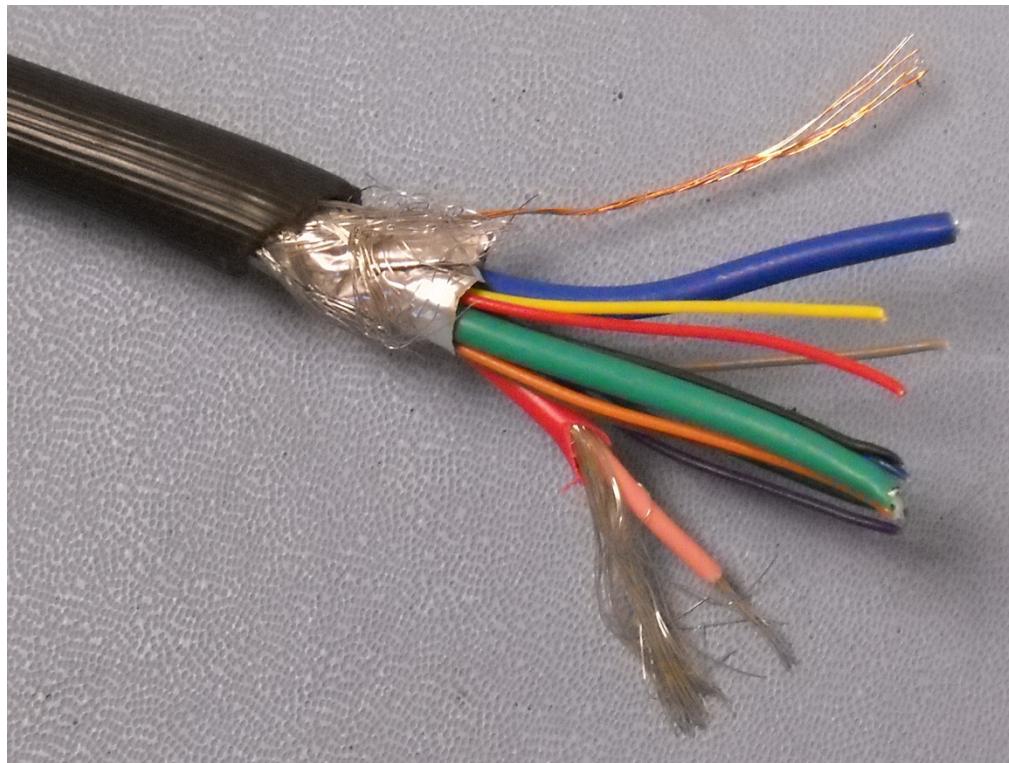
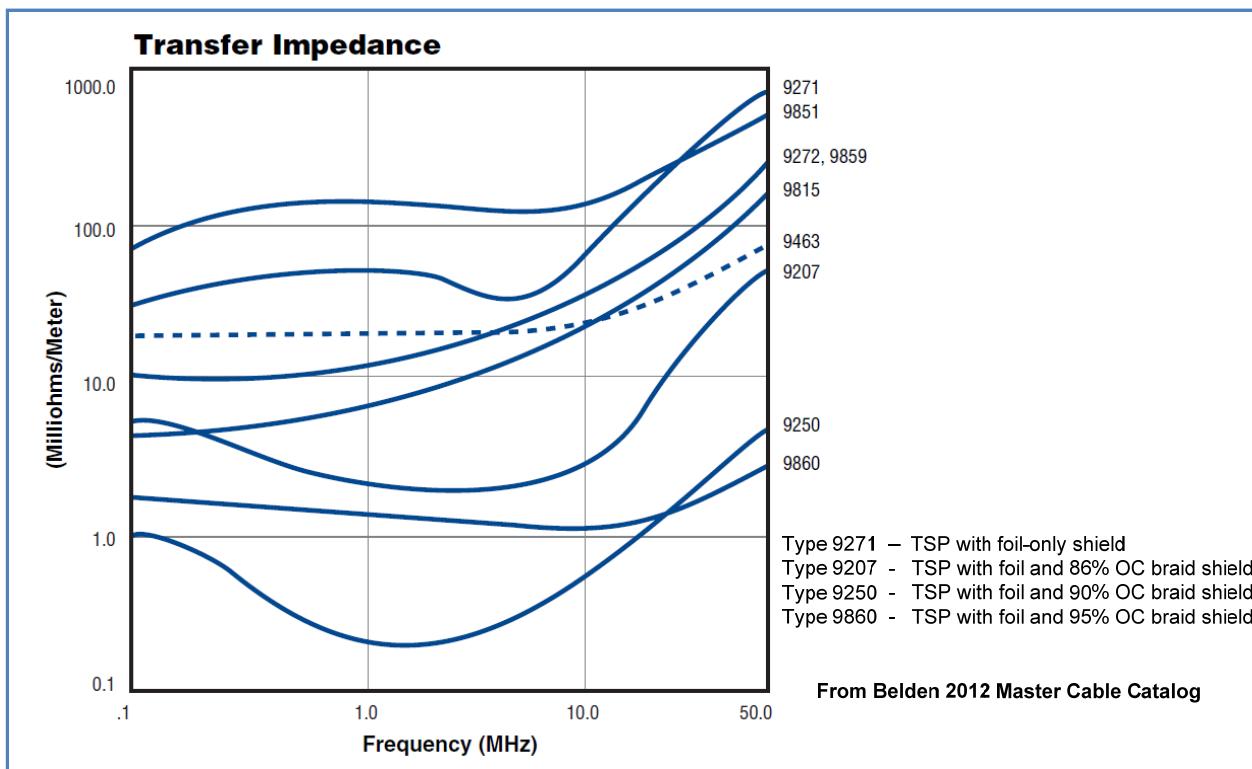


Figure 3.4-2. Screenshot Showing Uses of Kley's Model in Cable Studio

**Figure 3.4-3. Cable with Combination Foil and Braid Shield****Figure 3.4-4. Transfer Impedance of Shields on Various Belden Data Cables Containing Foil**

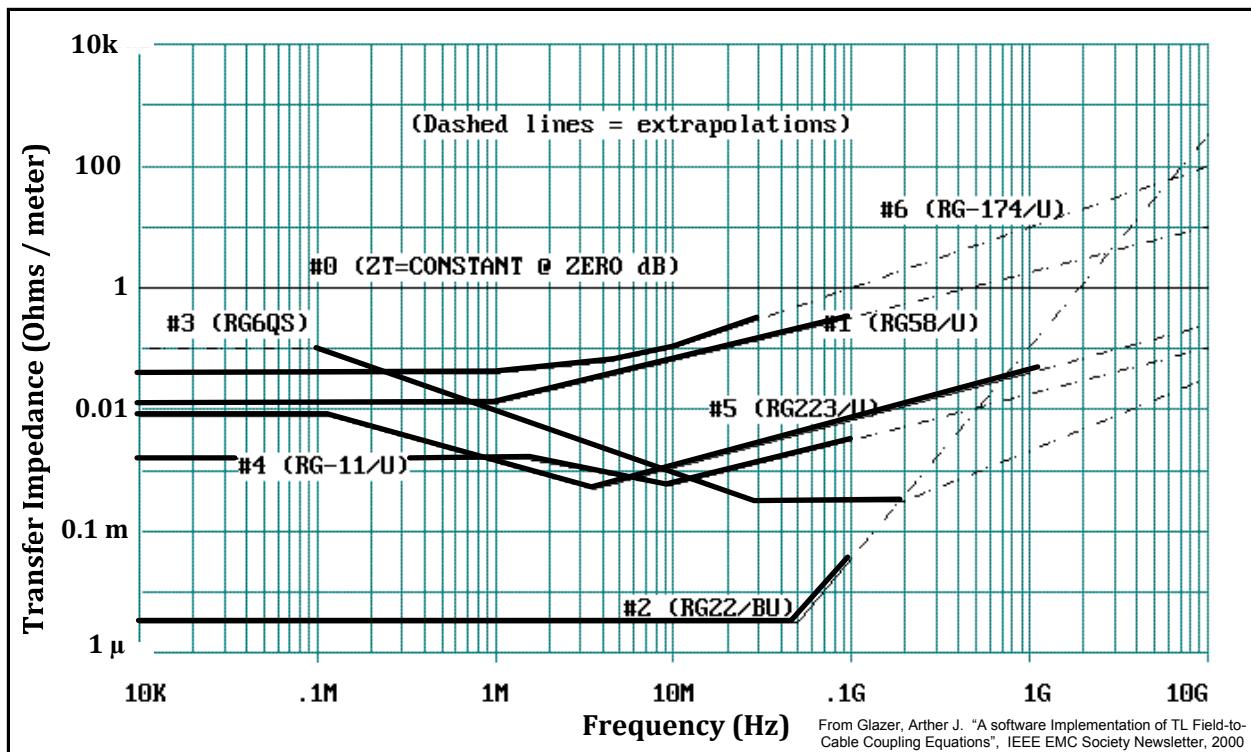


Figure 3.4-5. Transfer Impedance of Shields on Popular Coax Cables Utilizing Braid-only Shields

4. MULTI-BRANCH CABLE EXAMPLE

To illustrate the concepts of breaking an EMC simulation model into entity blocks and the subsequent construction of these blocks for the application of MIL-STD-461F cable related susceptibility environments described in Section 2, an example centered on a multi-branch subsystem cable harness is presented.

Figure 4 -1 shows a compliance test setup notional drawing that may appear in an EMI compliance test procedure for a subsystem applicable to MIL-STD-461F EMI susceptibility requirements. This particular subsystem contains three Line Replaceable Units (LRU). LRU 1 interfaces with the vehicle platform to obtain prime power. It processes video and supplies conditioned power at the appropriate voltage to LRU 2 and LRU 3. LRU 2 is a remote video display which obtains video from LRU 1 via a balanced digital video interface and video overlay data via an Ethernet interface. LRU 3 is a remote video camera that provides unbalanced composite NTSC analog video to LRU 1, where it is subsequently digitized.

It should be noted that the test setup shown in Figure 4-1 specifically shows application of the CS114, CS115 and CS116 environments to the LRU 1 signal cable. This setup, with the current probes removed, is also applicable to the application of the RS105 environment to the three LRUs (if the planewave generating structure is added above the ground plane), as well as RS103 applied to LRU 1 and LRU 2. To be applicable to RS103 for LRU 3, the positions of LRU 2 and LRU 3, and associated cables, would have to be swapped.

For brevity, the complete simulation and analysis will be only shown for the CS114 environment as applied to LRU 1. Application of the CS115 and CS116 environments will be similar, except the CS114 injection setup is replaced with the CS115 or CS116 injection models as shown in Section 2, and, the simulation is performed in the “Transient Simulation” mode, as opposed to the “AC Simulation” mode as for CS114. It should be noted though, that it may also be possible, or even preferred, to perform the CS115 and CS116 simulation and analysis using “AC Simulation” with Fourier Transform post-processing. “AC Simulation” effectively applies the stimulus and presents the results in the frequency domain. “Transient Simulation” effectively applies the stimulus as its time domain representation and presents the results in the time domain as well.

4.1 Focus on EMI Compliance

In this particular situation, the subsystem consisting of three LRUs is subject to MIL-STD-461F environment compliance testing per the five environments presented in Section 2. The diagram in Figure 4-1 specifically illustrates the set up for injection of the CS114, CS115 and CS116 environments on to Cable 1 at LRU 1. For this example, it assumed that the power input is sufficiently isolated from the LRU 1 cabled Input / Output. This is often a valid assumption for well-shielded equipment items that contain a separate power input connector and that are bonded to a ground plane.

All simulation modeling needs to start with an understanding of the environments and how they are applied and measured (see Section 2) and the specific compliance test setup. In the end, we design to “pass the test.” Anything short of passing the test is non-compliant and may result costly redesign, as well as delivery delays. Passing the test with great margins may present an overdesign at the expense of increased system cost, weight and reliability. Thus the best way to most efficiently pass the test is to know what is on the test – understanding the requirements, equipment arrangement, grounding and bonding particulars, cable details, etc. Just as important as understanding these aforementioned items, the criteria for pass/fail with respect to the susceptibility environments must be fully defined.

Pass/fail criteria for the EMI susceptibility environments is often first defined at the equipment or subsystem level, and may vary between the applicable EMI environments to be applied. Equipment and system level pass/fail criteria typically includes statements such as: “performance shall not degrade beyond specified performance parameters,” “the information presented on the display shall always be correct and readable,” “display blanking is allowed during the application of the transient, but the correct image must automatically reappear after blanking within 3 seconds after the transient” and, “the subsystem shall not present hazardous or misleading information.” These criteria are all centered on the end user and are not very easy to quantify or present in a simulation environment, as the simulation environment will present outputs in the form of electrical quantities. Thus for simulation with respect to interference applied to cable assemblies, electrical thresholds of susceptibility must be defined, typically at the connector pins or at the output of the EMI filter between the connector pins (the filter input) and I/O circuit (the filter output). The thresholds of susceptibility will be functions of the desired signals’ characteristics, acceptable signal-to-noise ratios, and of course, the interference level presented on top of the desired signal by the environment. The end result will also consider time coincidence. For example, if the interference is only transient in nature, a loss of a small amount of data or signal become unnoticeable if data buffering and/or error detection and correction are part of the interface design.

Electrical thresholds of susceptibility may equate to a particular common mode or differential mode voltage and current expressed as a function of frequency in the case of the CS114 and RS103 environments. In the case of the CS115, CS116 and RS105 environments, the electrical thresholds of susceptibility may equate to a peak common mode or differential current or voltage, or a maximum allowable time for the presence of a particular common mode or differential mode transient voltage or current. Some basic circuit analysis is typically required to determine these thresholds. Of course the results must be analyzed with respect to thresholds for component damage as well, especially for the CS116 and RS105 environments.

When the interference presented to victim circuits is greater than the susceptibility threshold, failure due to EMI can be expected. Typically, for design confidence, a 10 dB to 20 dB margin (factor of 3x or 10x for voltage and current) should be added for simulations of the type described herein.

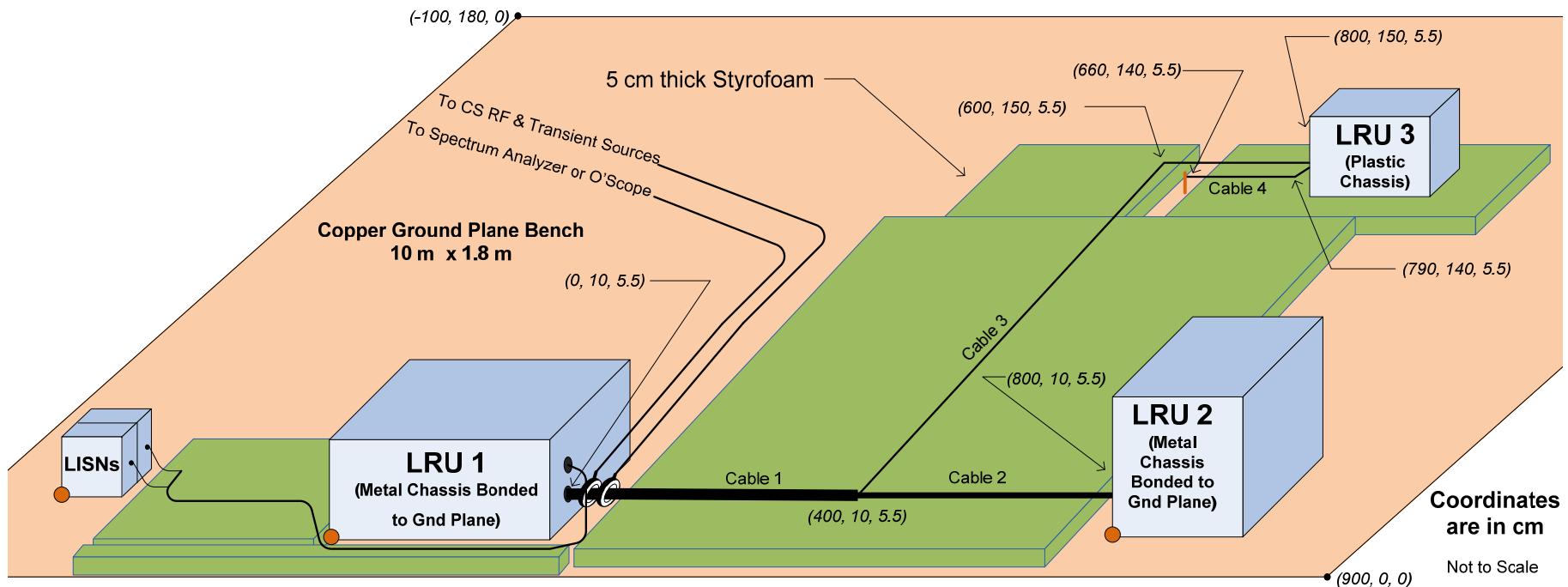


Figure 4-1. EMI Test Setup for CS Injection On to Cable 1 at LRU 1

4.2 Example Compliance Test Setup Specifics

With respect to the MIL-STD-461F example presented in Figure 4-1, the following is noted:

- The subsystem is mounted on top of a 10 meter by 1.8 meter copper bench in a manner to allow compliant testing of cable bundles 1 and 2. For application of the RS103 environment with respect to cable bundles 1 and 2 being applicable, the plane wave is applied in the y -direction. For application of the RS105 environment, the plane wave will propagate in the x -direction. The setup would have to be rearranged for testing of cable bundles 3 and 4.
- LRU 1 and LRU 2 are equipment items with metallic chassis that are normally bonded to the host vehicle frame. Hence in the EMI test setup, they are bonded to the copper ground plane bench.
- LRU 3 has a non-conductive plastic chassis. In the vehicle installation, it is attached to a mount that places it several cm away from the vehicle structure. Thus in the MIL-STD-461F EMI test setup, LRU 3 is placed on top of 5 cm Styrofoam.

LRU 3 has a safety ground conductor that is 1.5 meters \pm 10 cm per installation drawings. This grounding conductor (cable bundle 4) is routed 5 cm above the ground plane and bonded to the ground plane at the end of the 1.5 meter run.

- Per MIL-STD-461F test conditions, cable bundles are placed 5 cm above the ground plane bench. The coordinates shown in Figure 4-1 represent the approximate center of the cable cross-section.
- With the exception of the gross overshield that covers cable bundles 1, 2 and 3, all conductors in cable bundle 1 branch at location (400, 10, 5.5) continue to either cable bundle 2 or cable bundle 3 through a metallic Y tube. The overshields on cable bundles 1, 2 and 3 become continuous with this tube.

Figure 4.2-1 shows a schematic of the cable harness, as well as photographs of its most significant components.

4.3 Determination of Significant Features

The real skill of electromagnetic effects modeling and simulation rests with identifying which features of the real hardware need to be included in the model and simulation. Selecting features that are not significant makes the model complex to build and simulation very time consuming, not to mention the increased likelihood of introducing computational instabilities. Omitting features that may be significant may lead to gross errors and inaccurate conclusions.

To determine which features may be significant, the analyst must first identify the highest frequency of interest. For susceptibility of defense/aerospace electronic equipment and subsystems that utilize military style (e.g. MIL-C-38999) connectors and shielding, notable reception of MIL-STD-461F environments via cable assemblies is typically limited to frequencies below 400 MHz, unless there are gross underlying design flaws that would typically be identified during a design review by a qualified, experienced EMC engineer.

Also, as noted in Section 2, the MIL-STD-461F conducted susceptibility requirements present threats with the highest frequency components of 200 MHz (CS114), 320 MHz (CS115 w/ 1 nS transition time), and 100 MHz (CS116). The RS105 threat carries significant content that quickly falls off above 80 MHz.

Once the highest frequency of interest has been determined (typically 400 MHz), a wavelength, λ , can be associated with it. For example, at 400 MHz, the wavelength is 75 cm in air, and 37.5 cm in a dielectric with a permittivity of 4.0.

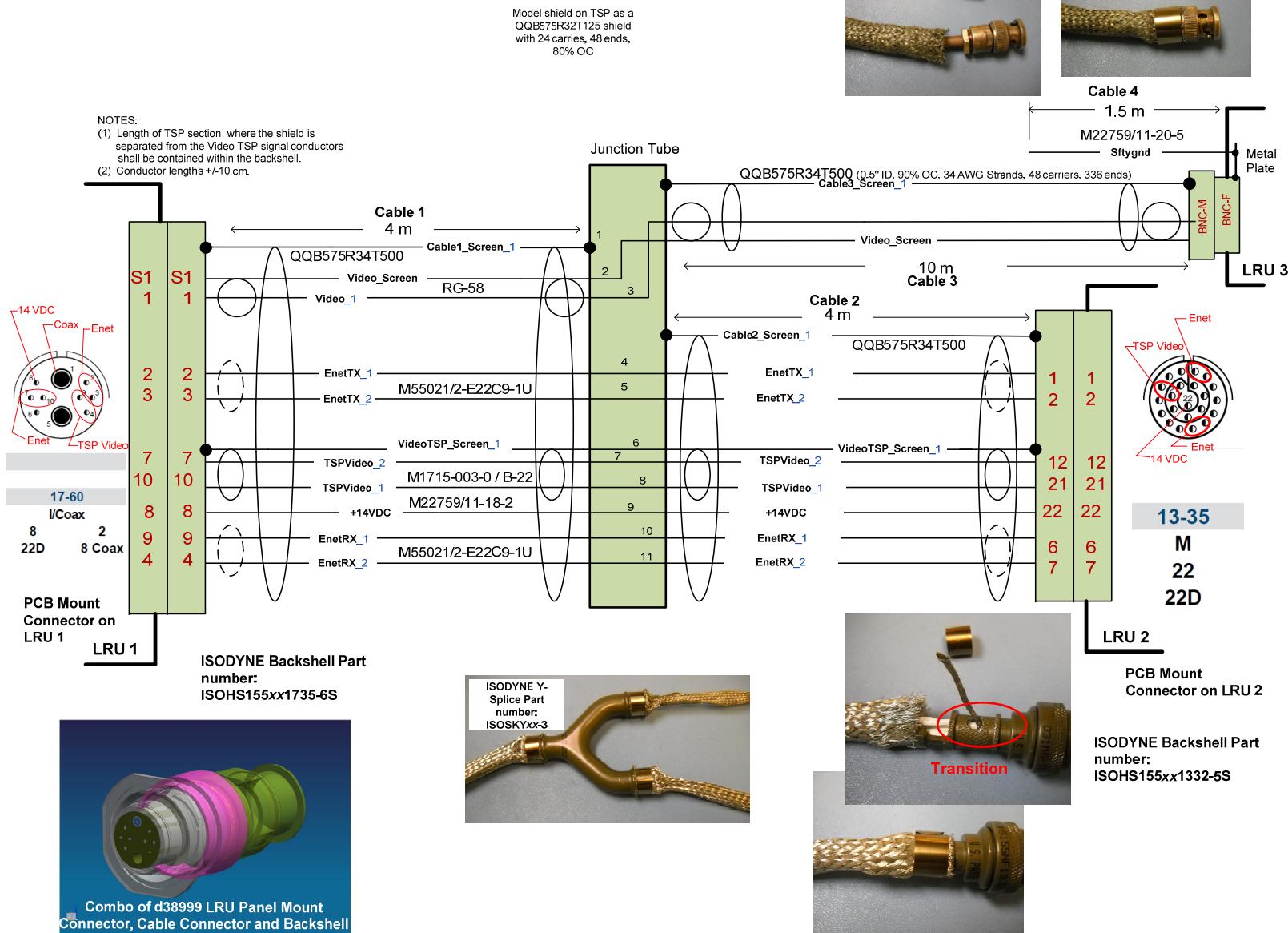


Figure 4.2-1. Cable Harness Build Information

Mechanical features that have dimensions greater than $\lambda/20$ are almost always electromagnetically significant. When a large amount (> 60 dB) of shielding is desired, features as small as $\lambda/100$ could be important. For example, a 3.75 cm wire ($\lambda/20$ at 400 MHz in air) used to terminate a cable shield will most likely degrade the effectiveness of that shield to somewhere in the 10 to 16 dB range at 400 MHz, whereas a continuous circumferential shield termination will typically yield more than 60 dB of shielding if the shield itself can provide that much shielding.

From a practical standpoint, the “electrically significant” threshold falls somewhere between $\lambda/20$ in air and $\lambda/20$ in a dielectric with a relative permittivity of 4.0 (FR-4 in circuit boards, cable dielectrics). This range is thus 1.875 cm (0.74") and 3.75 cm (1.5") at 400 MHz. Some common conclusions that come out of this are as follows:

- Model the backshell-connector-connector assembly if the total length is greater than 0.75". Otherwise ignored in most situations.
- Model the wiring that breaks from a constant cable cross-section if the length is longer than 0.75".
- Model shield termination pigtails longer than 0.75".
- If two connectors are within 1.5" of each other, it is typically acceptable to model both as having a common location (same node).

A few other generalizations that have been derived from experience for frequencies no greater than 400 MHz include:

- Properly terminated coaxial connectors can be ignored. Rather, model as an extension of the coax cable itself.
- Circumferentially terminated cable shields can be modeled as perfect connections.
- Coupling between cable runs separated by 2 cm or more can be ignored.
- A 20 cm change in length or node position of a typical 2 to 10 meter cable in a MIL-STD-461F test setup or its model will typically not change the overall pass/fail conclusion by no more than a dB or so, unless there are gross violations of best industry-standard EMI design practices.
- Features that cause DC bonding resistance to be greater than 2.5 milliohms should be included in the electromagnetic model.
- Electrically short paths (e.g. length less than 1.5") for terminating circuit elements and assemblies to a ground can generally be modeled as an ideal connection, if the DC bonding resistance is less than 2.5 milliohms and the path has a width of at least 1/3 the length.

4.3 Entity Blocks

Build details of the cable harness for EMI test is presented in Figure 4.2-1. It indicates the part numbers of the connectors, backshells, shields and signal conductors. Signal names and connector pins associations are also indicated on the drawing.

Figure 4.3-1 presents the schematic for the I/O associated with Cables 1 through 4. The schematic is limited to I/O components that are deemed instrumental in influencing the interference voltages and currents at critical circuit locations. Not only does this schematic contain discrete components, it contains SPICE model and Touchstone format S-parameters blocks as well. The schematic also indicates points for probing voltage and current (red diamonds). With respect to grounding and bonding, the schematic indicates connection to 3-D ground per Figure 4-1 and establishes the location where the 3-D cable harness environment ground would connect to co-simulation environment ground reference for the case where interference is to be applied at LRU 1.

Based on the highest frequency of interest being 400 MHz and the details presented in Figures 4-1 (the EMI compliance test setup), 4.2-1 (cable build drawing), and 4.3-1 (I/O representative schematic), we would conclude that the following entity blocks would have to be included in our simulation model:

- Cable harness made up of constant cross-section cable bundles 1, 2, 3, and 4.
 - The Y-tube / junction for cable bundles 1, 2, and 3 can be modeled as an ideal connection at the center of the Y-tube.
 - The cable harness will include grounded nodes at (0, 10), (800, 10), and (660, 140) cm.
 - Since the schematic does not show direct or closely coupled connection between the power input conductors and the cable bundle 1 signals, and the prime power input does not share a connector with the signal I/O of interest, the power cable will not be included.
- Connectors-backshell combination at LRU 1
- Transition at LRU 1 backshell
- Connectors-backshell combination at LRU 2
- Transition at LRU 2 backshell (because of the small size of the LRU 2 backshell, the bond between the video twisted shielded pair's shield and the backshell can be modeled as an ideal bond)
- (The connectors and transition at LRU 3 can be ignored due to high-quality shield terminations.)
- One environment stimulus (Figure 4-1 shows setup for CS114, CS115 and CS116 on cable bundle 1.)
- LRU 1 I/O circuits
- LRU 2 I/O circuits
- LRU 3 I/O circuits

Figure 4.3-2 shows the example problem split into entity blocks. The interfaces are shown for each block. Node labels and coordinates denoting the physical location of the Cable Harness entity block's interfaces are also shown. Also shown is the combination of the LRU 1 Connector Set and Transition entities combined into one block, as this appears to be an easy and convenient method to simplify the model build process. The same is done for the LRU 2 Connector Set and Transition entities. In summary, with the exception of the RS Stimulus entity block (which would be embedded in the Cable Harness model space)

entity), the blocks and interconnections shown in Figure 4.3-2 should also be present in the co-simulation tool's presentation of the model. Table 4.3-1 describes and summarizes the applicable entity blocks.

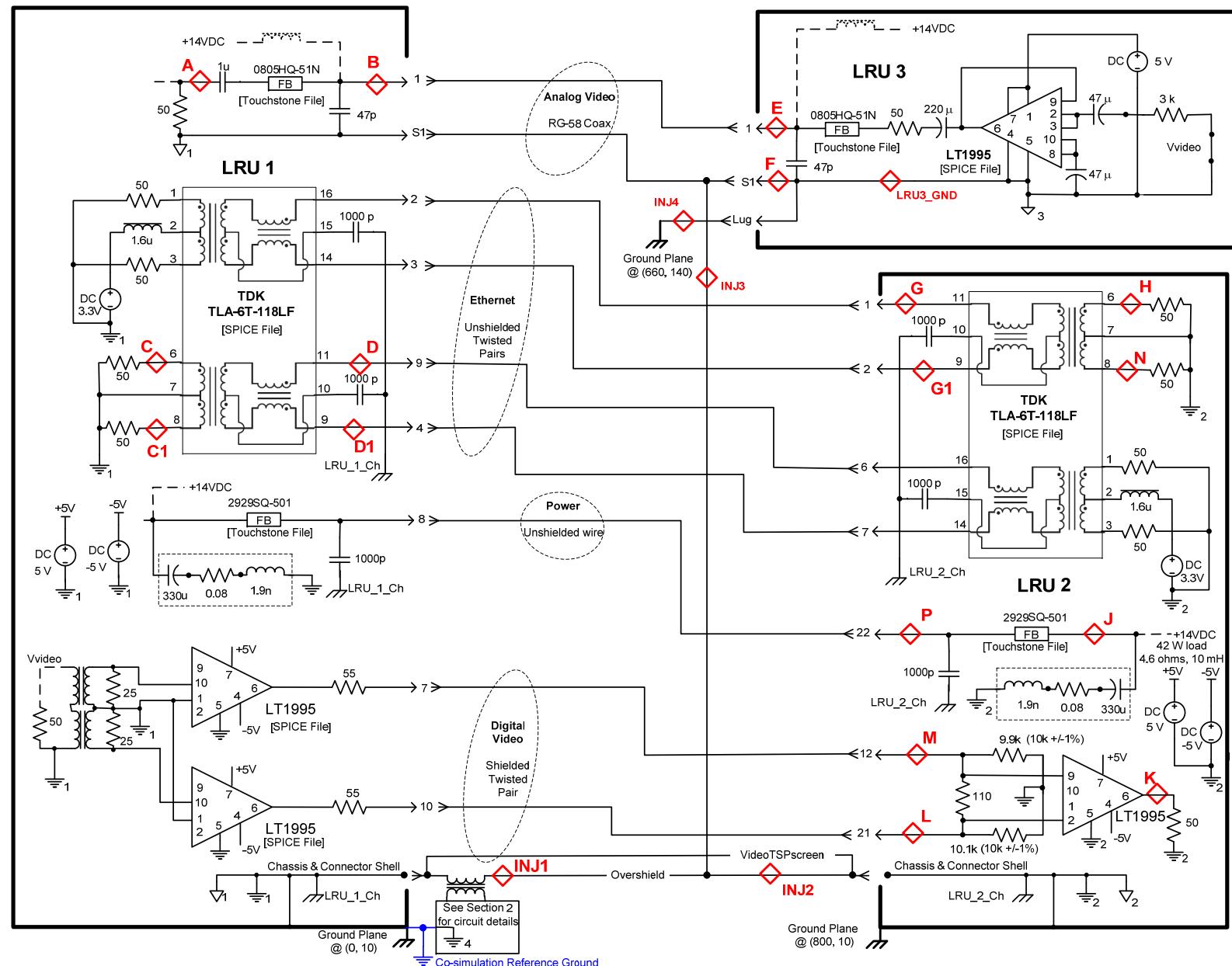


Figure 4.3-1. I/O Circuits Schematic with Probe Locations

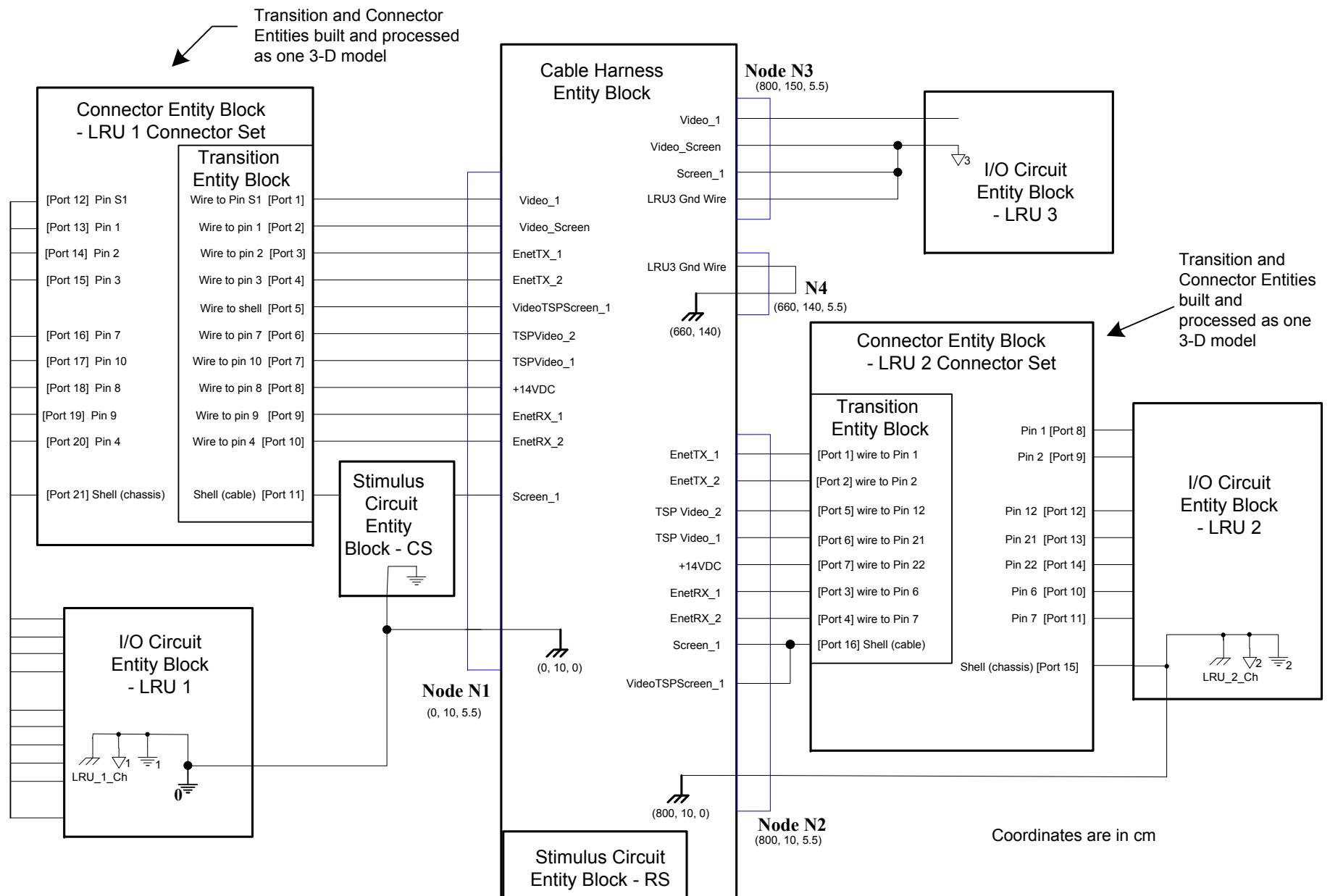


Figure 4.3-2. Entity Blocks and Interconnection in Co-simulation Framework

Table 4.3-1. Entity Block Description and Specification for Example Model

Entity Block Type & Name	Description	Method of Specification	Notes
Cable Harness – Cable Harness	Multi-branch cable as geometrically shown in Figure 4-1, with conductors as noted in Figure 4.2-1. Cable is routed 5 cm above a copper ground plane (centered at 5.5 cm above the ground plane). Y-connector assumed to be ideal.	3-D model (2.5-D TLM format) block built in Cable Studio. Resulting model is represented electrically as an S-parameter block.	This entity contains terminals (which can be connected to other blocks) for electrical connection to the ground plane at three specific locations within the 3-dimensional space occupied by the cable harness and its ground plane.
Connector & Transition Combined – LRU 1 Connector Set	Includes the connector and backshell that is attached to the cable, the connector attached to LRU 1 or LRU 2 and the wiring that transitions from the constant cross-section cable bundle to connector pins.	3-D model built in Microwave Studio. Connector parts model data imported from mechanical models. Wires added in Microwave Studio. Resulting model is represented electrically as an S-parameter block.	LRU 1 Connector Set w/transition is a 21-port S-parameter block.
Connector & Transition Combined – LRU 2 Connector Set			LRU 2 Connector Set w/transition is 16-port S-parameter block.
I/O Circuit – LRU 1	Driver and receiver circuits with EMI filters that represent what the connector pins will electrically see. Includes probe points on circuit nets and nodes of interest for determining interference levels. Circuit elements representing impedance in connections between grounds also included.	Collection of discrete circuit elements, SPICE models, and S-parameter file/data blocks.	The cable harness's reference ground plane at location (0, 0) and the co-simulator tool node 0 are connected in this block to represent a connection of circuit ground (D1) to the ground plane at the physical location of LRU 1 (node N1).
I/O Circuit – LRU 2			Circuit grounds, through various impedances, connect to the cable harness's reference ground plane at a specific location (node N6).
I/O Circuit – LRU 3	Output voltages from probes will be with respect to co-simulation framework node 0 ground.		Circuit ground in the LRU 3 block is "floating" from the reference ground plane. It does connect to the cable harness's reference ground plane at node N9 through about 1.5 meters of AGG 22 wire.
Connector & Transition Combined – LRU 3 Connector Set	Determined to be electrically insignificant and is thus omitted.		Connection of cable overshield, coax ground and connector shell ground wire is done as an ideal connection at the interface of cable block node N8 and the LRU 3 Circuit block.

Entity Block Type & Name	Description	Method of Specification	Notes
Stimulus – CS (for LRU 1 cable 1 evaluation)	Electrical model of CS114, CS115 or CS116 sources as well as monitoring probes needed to “calibrate” the source levels.	Discrete electrical components with a voltage or current source. See details in Section 2 for models of CS114, CS115 and CS116 sources.	<p>Circuit is referenced to the cable harness ground plane at the point of application (node N1).</p> <p>The use of this block will require an AC Simulation task in the co-simulation framework if the CS114 environment is applicable.</p> <p>The use of this block will require a Transient Simulation task in the co-simulation framework if the CS115 or CS116 environment is applicable.</p> <p>Should not be used at the same time as Stimulus – RS block.</p>
Stimulus – RS (for LRU 1 & LRU 2 evaluation)	<p>Planewave at 50 V/m, 30 to 500 MHz, with the following orientations [E, H, k]:</p> <ul style="list-style-type: none"> (a) [z, x, y] (b) [x, -z, y] <p>Also, planewave with the E-field waveform per Figure 2.5-1 with the following orientations [E, H, k]:</p> <ul style="list-style-type: none"> (a) [z, -y, x] (b) [z, x, y] 	Planewave excitation expressed as a time domain Gaussian waveform (for the 50 V/m planewave), or a double exponential pulse. This block is inserted into the Cable Harness block.	<p>Only objects in the Cable Harness block are exposed to this stimulus.</p> <p>The use of this block will require a Transient Simulation task in the co-simulation framework.</p> <p>Should not be used as the same time as the Stimulus – CS block.</p>

4.4 Model Input and Analysis – Explanation by Example

THE FIGURES REFERENCED IN THIS SECTION ARE PRESENTED AFTER SECTION 5.

Experience has shown that co-simulation models should be built and tested in phases or steps, with a check of the data for reasonableness⁷ at the end of each step. This will allow capture of potential instabilities and errors that may go unnoticed until after the final complete run in the co-simulation framework. For the CST Studio Suite tool set with the co-simulation hosted in Design Studio, the following sequence model build and checks is recommended. The application of the CS114 environment to the example cable harness is presented to illustrate the modeling process. The results from each of the modeling steps are presented in figures at the end of this section.

4.4.1 I/O and Signaling Schematic with Common Ground

The first step of EMC modeling and analysis for a complex cable harness is to identify the significant portions of the I/O interface circuits, along with the EMI filter components. Thus the I/O circuit entity blocks are built and checked as follows, first with a common grounding scheme, in the co-simulation framework's schematic editor.

- a. Connect inputs and output per cable bundle signal wiring (Figure 4.4.1-1).
- b. Do not include cable shields that are not used as signal returns.
- c. Do not include probes.
- d. Terminate the circuit inputs and outputs with 50-ohm S-parameter ports.
- e. Run a S-parameter simulation task over the desired target co-simulation frequency range (e.g. 1 kHz to 400 MHz). It may be beneficial to extend the lowest frequency down to 1 Hz to allow a more complete observation of circuit behavior.
- f. Evaluate S-parameter results for reasonableness. Specifically, ensure that the magnitude of reflection coefficients are no greater than 1 and that the gains (losses) with respect to frequency are as expected.

Note that the S-parameter simulation will provide 50 ohm terminations. This may, or may not, be representative of the actual circuit.

Of particular importance, ensure that the gains for each circuit are flat for the needed frequency range. For example, per Figure 4.4.1-2, we do see that the gain of the analog video circuit is flat from 2 kHz to 10 MHz to provide an adequate NTSC baseband video link, the gain of the 100BaseT Ethernet circuits is flat over the required 1 to 100 MHz range, and the gain of the digital video circuit is flat from below a few Hz to 10 MHz to support digital NTSC video with I-frame compression – 40 Mbps total rate (30 Mbps serial data rate plus 10 Mbps overhead) sent as 8-ary ASK. Additionally, we see the gain through the 14 VDC power circuit falling off above 10 Hz, which is expected of a good power EMI filter. The reverse direction gains shown in Figure 4.4.1-3 verify operability of the SPICE models, noting the high reverse isolation provided by the operational amplifiers used in the video circuits.

⁷ Throughout this section, it is required that the data be evaluated or checked for “reasonableness.” From a practical stand-point this activity typically requires an EMC engineer with many years of practical test, measurement, troubleshooting, and analysis experience.

4.4.2 I/O and Signaling Schematic with Target Grounding System

From a practical standpoint, a common ground is typically not applicable for most systems and equipment. Grounds be intentionally be isolated or they may be have some isolation due to impedance and electromagnetic transmission line characteristics between them if distances are electrically long. This must be accounted for if accurate results are expected. To accommodate ground isolation with the three LRUs that make up the system under test, the following process is followed.

- a. Modify SPICE model files which contain a “node 0” and are used in a situations where the model’s reference ground (node 0) will be connected to the co-simulation framework’s reference ground through an impedance and/or electrically long path. This is done by editing the SPICE netlist files as follows.
 - i. Add **GND** to the list of .SUBCKT terminals
 - ii. Add net **RGND GND 0 1e8**
 - iii. Replace all node zeros with node **GND**

Save the file with **_float** added to the file name

For this example, the operational amplifiers used in LRU 2 and LRU 3 require this modification if we let LRU 1’s ground be equal to the co-simulation framework ground as planned in Figures 4.3-1 and 4.3-2. Note that the modified SPICE blocks will now contain an extra terminal called GND.

- b. Modify S-parameter files that are used in a situations where the model’s reference ground (node 0) will be connected to the co-simulation environment through an impedance and/or electrically long path.
 - iv. In CST Design Studio, the “Differential” selection can be made for imported Touchstone S-parameter files to accomplish this.

For this example, the Touchstone models for the ferrite inductors in LRU 2 and LRU 3 require this modification. The result will be twice as many ports on the affected model blocks.

- c. Build the I/O circuit models with the target grounding scheme. Add connections between the floating grounds and the co-simulation environment’s reference ground using an impedance that may be representative of what is expected to occur via the cable model and other 3-D structures in the paths. (Figure 4.4.2-1).
- d. Connect input and output circuits per cable bundle signal wiring.
- e. Do not include probes.
- f. Terminate the circuit inputs and outputs with S-parameter ports. Use differential S-parameter ports for those ports with a reference other than the co-simulation environment’s ground reference. For this example, this will be needed at LRU 2 and LRU 3.
- g. Run a S-parameter simulation task over the desired target co-simulation frequency range (e.g. 1 kHz to 400 MHz) as done in step 4.4.1.e. It may be beneficial to extend the lowest frequency down to 1 Hz to allow a more complete observation of circuit behavior.
- h. Evaluate S-parameter results for reasonableness. Specifically, ensure that the magnitude of the reflection coefficients are no greater than 1 and that the gains (losses) with respect to frequency are as expected.

Unless there are notable ground impedance issues in the signal paths, the results (Figures 4.4.2-2 and 4.4.2-3) should look almost identical to those in step 1, with possible exception of values that are below -100 dB due to “simulation precision.” If the results do look unsatisfactory, a system EMC design flaw may be present – a flaw that should be corrected before performance in an interference environment is considered.

4.4.3 Cable Harness Model Build

The cable harness model should be built per the instructions below. Prior to build of the cable, the geometrically and physical details of the cable cross-section’s conductors, shields and insulators must be known. This may require some research, especially with respect to the details of shields, which determine transfer impedance.

- a. Using the cable modeling tool (e.g. CST Cable Studio), build the cable model to include the ground plane.⁸ (Figures 4.4.3-1 and 4.4.3-2)
- b. Simulate the cable model with the cable LRU interface ports terminated in to S-parameter blocks. Use differential ports on pairs that would normally utilize differential signaling. Include S-parameter ports on cable shields terminals as well. (Figure 4.4.3-3)
- c. Evaluate S-parameter results for reasonableness. Specifically, ensure that the magnitude of reflection coefficients are no greater than 1 and that the gains (losses) with respect to frequency are as expected. A typical vehicle cable will exhibit no loss at very low frequencies (<10 kHz) and losses increasing to about a 1 to 2 dB nominally at 400 MHz per 2 meters of conductor. The losses will most likely exhibit peaks and valleys starting at the frequency where the signal transmission path is one-quarter wavelength if impedance mismatch is present. For example, a 8 meter transmission path with unmatched terminations will exhibit its first resonant peak or valley at 9.3 MHz if the dielectric is air, a little lower if there is insulation on signal conductors (7.2 MHz if effective permittivity = 1.6). Additionally, other resonant peaks and valleys can be expected where parasitic capacitance and inductance caused by the presence of other conductors and cable features interact with the conductor of interest.

The desired path transfers (Figure 4.4.3-4), cross-talk between conductors in the cable bundle (Figure 4.4.3-5), and coupling between the overshield and shielded conductors (Figure 4.4.3-6) were plotted and evaluated. The latter is instrumental in providing the analyst a view of the expected amount of shielding that can be provided by the cable/connector assembly. As seen in Figure 4.4.3-6, very little shielding (at several frequencies) is provided to the unshielded power conductor in the bundle above a few MHz, whereas more than 80 dB of shielding is achieved with the shielded twisted pair used for digital video. It is also noted that the shielding provided by the coax cable is less than one may expect; often

⁸ For conducted susceptibility modeling with CST Studio, simulation time can be reduced by setting up the cable model using the “none” option for coupling to 3D field solver. Use “uni-directional” coupling for the radiated environments. Other Cable Studio setting should be as follows: Allow modal models, Ohmic losses, Dielectric loss, High accuracy, Hexahedral TLM mesh, -80 dB Solver accuracy, Use Broadband phase shift, Double precision solver, 400 MHz frequency range with 1001 samples, linear sweep.

this results from the system grounding and signal termination scheme, especially as it is noted that the unshielded twisted pair performs better. It is important to observe, when considering the results that the Ethernet and digital video twisted pairs terminate into balanced loads, the analog video coax circuit termination is unbalanced. Finally, note the 50-ohm terminations on the shield terminals will also degrade shielding results.

4.4.4 Combine Cable and Interface Circuit Entity models

The cable can be inserted in to the circuit model developed in Section 4.4.2 to replace the ideal electrical connections between the LRU grouped circuits. The following workflow is recommended.

- a. Using the I/O circuit entity blocks developed in Section 4.4.2, which contains modified SPICE and differential S-parameter blocks to support the floating ground, import the cable model entity block into the co-simulation framework. (Figure 4.4.4-1)
- b. Use the same S-parameter ports at the I/O circuits as used in Section 4.4.2 to determine the signal transfer via the desired signal paths (Figure 4.4.4-2). The results should typically look similar to what were seen in step 2, except for the typical additional loss expected at frequencies starting between 500 kHz and 3 MHz, depending on the type of cable, length of cable, and impedance match between the I/O circuit and the cables. Peaks and valleys may be present due to cable resonance and resonances caused by parasitic capacitances and inductances.
- c. Evaluate cross-talk between key signals for reasonableness (Figure 4.4.4-3). Note that cross-talk is a function of terminating impedances, and, with that, expect differences in end-to-end cross-talk when the intended terminating circuits are included in the model.
- d. Use the results of this step as a decision point for cable design adequacy with respect to signal integrity. If loss and/or cross-talk is excessive, the next action may be to redesign the cable with a focus on signal integrity – poor signal integrity may prevent system level success in a benign environment!

4.4.5 Addition of EMI Environment Injection

The EMI environment injection model is next added in the co-simulation environment. Then the S-parameter analysis is performed to ensure reasonable transfer from the environment to the I/O circuits via the cable. Cable design problems may be identified as early as this step. The process to add the CS114 environment is as follows:

- a. Add CS114 injection model to cable under test at the LRU under test in the co-simulation framework (e.g. CST Design Studio). Drive the source with an S-parameter port. Terminate signal ends with S-parameter ports. (Figure 4.4.5-1)
- b. Recheck coupling through desired signal paths (Figure 4.4.5-2).
- c. Evaluate the coupling between interference drive source and victim circuits for reasonableness. (Figure 4.4.5-3)

Compare the results obtained in Section 4.4.3 (Figure 4.4.3-6) for the coupling

between the cable conductors upon which interference will be applied to with what was obtained in this step. This comparison shows the effects of the filter circuits. With the exception of an offset for insertion loss of the interference entity block, the results at the very low end of the spectrum (< 10 kHz) should look about the same.⁹ As frequency increases, additional attenuation would ideally be expected in this model due to filtering provided by the I/O circuits, at least up through the frequency range where the filter parts are effective. However, interference coupling may increase in balanced circuits due to mode conversion via the interface parts. Again, the importance of this step is to evaluate the performance of the interface filters and circuits with respect to interference rejection: the circuits, not the cable, may turn out to be the limiting factor with respect to interference rejection.

4.4.6 Build Models for 3-D Connector, Backshell, and Transition Entity Structures

As with the cable harness model, the model for each connector, backshell, transition, or combo entity are build as separate stand-alone blocks that can be imported into the co-simulation framework. The process below is performed separately for each connector, transition, and connector/transition combo entity block. This particular example has two connector/transition combos (LRU 1 and LRU 2) that are electrically significant and hence require modeling.

- a. Obtain MCAD data for connectors, backshells, connector pins, and associated assembly parts.
- b. Using a MCAD tool, such as Siemens NX CAD, that can support manipulation of the assembly parts and produce CAD file in PTC/Pro/Engineer .prt format, produce a model that includes the geometry of shell components, dielectric fill, grounding structures, and pin structures. This step is typically done by a mechanical designer, not an EMC engineer.
- c. Convert (translate) .prt format drawing to a Standard ACIS¹⁰ Text (.sat) format file using a CAD conversion tool such as TransMagic, a product of TransMagic, Inc.
- d. Import .sat format 3-D model into CST Microwave studio. Add transition entity block features to connector model if applicable, using the MCAD features of this tool. Assign material properties. Add a 3-D ground plane structure around the connector for use as a reference for ports. Add ports connecting to the connector pins at the LRU end and to pins and/or transition wires at the cable end to the 3-D ground plane. Note that the length of the port will be presented as a wire. Also, add ports between the backshell and the reference ground plane at each end if the backshell is metallic. (Figure 4.4.6-1 shows the LRU 1 connectors and transition, Figure 4.4.6-4 shows the LRU 2 connectors and transition)
- e. Using the 3-D electromagnetic simulator tool (e.g. Microwave Studio), mesh the 3-D model and perform a full-field 3-D simulation to produce a S-parameter block or an 3-D electromagnetic block that can be imported in to the co-simulation framework, representing each applicable connector, transition and/or

⁹ If results are lower than -100 dB, do not expect consistency between minor changes to the circuit, as anything below -100 dB may be approaching the limit of the simulator's dynamic range.

¹⁰ ACIS is an acronym for the Alan, Charles, Ian's System 3-D modeling engine which is now owned by Spatial Corp.

connector/transition entity block. (The full 3-D simulation for a connector with several pins, such as the LRU 1 connector assembly and transition, which is represented as a 21-port structure, may take a couple days to run.)

- f. Link or import the S-parameter block into the co-simulation framework (e.g. Design Studio). Add S-parameter ports to represent signal flow through each structure per the overall cable/connector assembly model build (Figure 4.4.6-1 for LRU 1 and Figure 4.4.6-5 for LRU 2). Run a S-parameter simulation task.
- g. Evaluate S-parameter data to ensure that the transfers through the desired paths are what is expected (Figure 4.4.6-3 for LRU 1 and Figure 4.4.6-6 for LRU 2).

Note that the transmission of a “signal” from one port to another will show the “signal” received another port to be an integer fraction of the transmitted “signal” if connectors occur in the connector and/or transition. This is the case for the shield of the twisted shield pair in the LRU 1 connector assembly – it is connected to the backshell. Thus a signal applied at the LRU end of the connector assembly will be divided between the connection to the shield of the twisted shielded pair and the cable end of the backshell.

- h. Export the connector/backshell and transition structures as Touchstone files for use in the co-simulation framework. Alternatively, the structure can be exported as a 3-D electromagnetic model block that can be imported into the co-simulation framework.

4.4.7 Complete Co-Simulation Integration and Checkout

Electromagnetically significant connector assembly block models are next inserted between the ends of the cable and the interface circuits in the co-simulation framework. The environment interface is also included. At this point, as noted in the process below, S-parameter simulations can be used to ensure correct connectivity.

- a. Import the connector/backshell and transition structure Touchstone files into the co-simulation framework for inclusion with the I/O circuit entities, the cable harness entity and an interference source injecting onto the cable overshell (Figures 4.4.7-1 and -2). Alternatively, the connector blocks can be imported into the cable modeling tool as a 3-D electromagnetic model block.
- b. Use the same S-parameter ports at the I/O circuits as used in Section 4.4.5 to determine the signal transfer via the desired signal paths (Figure 4.4.7-3). The results should typically look similar to what was seen in Section 4.4.5, except for the typical additional loss at frequencies starting above 3 MHz, depending on the type of cable, length of cable and impedance match between the I/O circuit and the cable. Peaks and valleys may be present due to cable resonance and resonances caused by parasitic capacitances and inductances. Losses in signals that are transported via shielded cables may become even more notable at the higher frequencies if the shields do not ideally terminate to connector pins and/or the connector pins do not present an ideal impedance match.
- c. Evaluate cross-talk between key signals for reasonableness. Cross-talk for signals that are transported via shielded cables may become even more notable at the higher frequencies if the shields do not ideally terminate to connector pins and/or the connector pins do not present an ideal impedance match.

- d. Evaluate the transfer between the environment and the circuit interfaces for reasonableness (Figure 4.4.7-4).
- e. Use the results of this step as a decision point for cable design adequacy with respect to signal integrity. If loss and/or cross-talk is excessive, the next action may be to redesign the cable with a focus on signal integrity – poor signal integrity may prevent system level success in a benign environment.

4.4.8 Apply Environment, Probe Victim Circuits and Analyze Results

MIL-STD-461F CS115 Curve 5 limit is used for application of interference at the LRU 1 signal I/O cable for this example. The following process is used.

- a. Set up the co-simulation framework for an AC simulation task over the 10 kHz to 400 MHz frequency range with a logarithmic frequency sweep.
- b. Remove all S-parameter ports, except for the port driving the CS114 injection source. Replace the S-parameter ports with representative circuit termination impedances (Figures 4.4.8-1 and -2).
- c. Add the CS114 calibration setup to the model in the co-simulation framework. Drive it and the injection port as a voltage source. The calibration setup and method is detailed in Section 2.
- d. In the AC simulation task setup menu, set the two aforementioned injection ports to voltage sources with an amplitude of 30 volts. They are shown as ports 11 and 12 in Figures 4.4.8-1 and -2.
- e. Add the probes on the nets of interest as shown in Figure 4.3-1 . Note: The probes will measure the current flowing in a circuit net and the net-to-node_0 voltage, where node_0 is the reference ground in the co-simulation framework.
- f. Run the CS114 simulation.
- g. Review the raw probe voltage and current plots (Figures 4.4.8-3 and -4) for reasonableness, in particular the currents on the cable bundle shields. Note that the simulation tool may require probe voltages and currents to be plotted using a “Results” window prior to export to a text file – the export is done from the “Results” plot.
- h. Determine how the amplitude of the exported and plotted data from the simulation tool is presented. Typically, most simulation tools will present the “magnitude” of magnitude-phase representation of the signal, or as more commonly referred to, the “zero-to-peak” value. On the other hand, the voltage, current and field strength limits in MIL-STD-461F for swept frequency environments are stated as RMS levels. Thus when comparing simulator results to the MIL-STD-461F limit, the simulator results must be converted to RMS quantities, which is done by dividing the magnitude values by $\sqrt{2}$, if presented as zero-to-peak.
- i. Using the post-processing function of the co-simulation framework, or Microsoft Excel, to compute the injection drive level adjustment factor (DRIVE_SF) as a function of frequency based on the applicable CS114 current limit curve for the calibration setup. If this drive correction factor is more than 10x in either direction, consider re-running the simulation (to improve dynamic range) with a different drive level for the calibration and injection circuits, noting that the drive level for each circuit should be the same. The drive scale factor is determined as follows:

$$\text{DRIVE_SF} = I_{\text{limit}} \times \sqrt{2} / I_{\text{CAL}}$$

where:

I_{limit} = Applicable MIL-STD-461F CS114 limit, Amperes

I_{CAL} = Result (0-to-Pk) from simulation of calibration circuit, Amperes.

The Drive Scale Factor is plotted in Figure 4.4.8-5.

- j. Using the post-processing function of the co-simulation framework, or Microsoft Excel, to compute the results scaling factor based on the drive level scaling factor, which provides for the maximum possible drive to the cable under test adjustment per MIL-STD-461F calibration, and the cap on the amount of interference current on the cable per MIL-STD-461F procedures. The equations for this computation are as follows.

$$I_{\text{INJraw}} = I_{\text{INJ1}} \times \text{DRIVE_SF}$$

$$I_{\text{INJactual}} = \begin{cases} I_{\text{INJraw}} & \text{if } I_{\text{INJraw}} \leq 2 \times I_{\text{limit}} \\ 2 \times I_{\text{limit}} & \text{Otherwise} \end{cases}$$

$$SF = I_{\text{INJactual}} / I_{\text{INJ1}}$$

$$= \begin{cases} I_{\text{INJraw}} / I_{\text{INJ1}} & \text{if } I_{\text{INJraw}} \leq 2 \times I_{\text{limit}} \\ 2 \times I_{\text{limit}} / I_{\text{INJ1}} & \text{Otherwise} \end{cases}$$

$$= \begin{cases} I_{\text{limit}} \times \sqrt{2} / I_{\text{CAL}} & \text{if } I_{\text{INJ1}} \times \sqrt{2} / I_{\text{CAL}} \leq 2 \\ 2 \times I_{\text{limit}} / I_{\text{INJ1}} & \text{Otherwise} \end{cases}$$

where I_{INJ1} = Result from simulation for total injected current, Amperes; noting that the calibration circuit and the injection circuit each have the same source driving voltage and impedance.

This scaling factor will be smaller than the drive scaling factor (DRIVE_SF) if the cable harness presents a low impedance to the injection source. This lower factor is representative of reducing the interference source power during an actual laboratory test, as the current limit becomes applicable (6 dB above the CS114 limit per MIL-STD-461F). At frequencies where the presented impedance to the interference source is relative high (50 ohms or greater), the overall scaling factor (SF) will be equal to the drive scaling factor (DRIVE_SF). This represents the interference source power being set to the full calibrated level during a laboratory test. The overall scaling factor is also plotted on Figure 4.4.8-5.

For this example, which has a well shielded cable harness, we see from the scaling factors that over most of the test frequency range the “drive level” will need to be reduced as the cable current limit will be reached before the forward power limit. Where the cable starts to become one-quarter wavelength long, we see resonance effects – the result is the presentation of a relatively high impedance at the injection point.

- k. Adjust the simulation probe currents and voltages by the scaling factor determined in j above. This may be best done by exporting the data to a spreadsheet for this post processing. Figures 4.4.8-6 through 4.4.8-8 present the

corrected results.

It is very important that the modeling and analysis process focus closely on numerical precision, as the dynamic range associated with EMC situations is typically very large. For example, in Figures 4.4.8-7 we see over 160 dB (a ratio of 1 to 10^8) differences in current and voltage amplitudes.

4.4.9. Analysis of the Results

Prior to analyzing the results, the threshold voltages and currents at the key interface circuit nodes and nets must be known. This often involves consulting the interface standards and component data with quality objective. For example, IEEE 802.3 is consulted to determine the allowable noise (interference) at the Ethernet receiver input for the 100Mbps rate. The analysis process for the subject system proceeds as follows:

- a. Identify key signal-to-interference points (probe points) of concern and quantity allowable interference levels. Plot and compare the interference levels (noise) predicted by the simulation with the allowable interference levels. For this example, there are four nodes:

1) TSP Digital Video presented to receiver in LRU 2 at probe K

The format for the video presented to LRU 2 is NTSC DV¹¹ with I-frame compression – 40 Mbps total rate (30 Mbps serial data rate plus 10 Mbps overhead) with a 10 MHz channel bandwidth using 8-ary Amplitude Shift Keying (ASK).¹² A 42 dB S/N (36 dB plus 6 dB margin) is needed to prevent pixilation caused by stationary noise.¹³ The desired signal voltage (S) at node K will be 4 volts peak-to-peak per the link design standard. Maximum interference allowed is thus 11 mV_{rms} for frequencies up to 10 MHz. Above 10 MHz an increase of noise is allowed at 40 dB/decade, up to VCC (= 5 Volts).

Figure 4.4.9-1 presents a plot of predicted interference at the output of the digital video receiver at LRU 2 (probe K in Figure 4.3-1). The allowable noise is also plotted. Of course plot in Figure 4.4.7-3 shows that the channel bandwidth is acceptable. Thus in this example, we can expect acceptable performance from the Digital Video link in the MIL-STD-461F CS114 Curve 5 environment.

2) Analog Video (Coax) at probe A

The analog baseband standard definition NTSC color video originating from LRU 3 and received and processed (digitized) by LRU 1 with a 6 MHz bandwidth. Resolution of A-to-D converter in LRU 1 is 4 mV. Thus max allowed noise up to 6 MHz is 2.8 mV_{rms}. Typically it is acceptable if this increases at a rate of 40 dB/decade above the signals bandwidth (6 MHz) up to 1 volt.

¹¹ National Television Standard Committee (NTS) Digital Video (DV) 525 lines, 480i effective display.

¹² Theoretical channel capacity = $2 \times \text{Bandwidth} \times \log_2(M)$, where M = number of signaling levels.

¹³ Minimum theoretical S/N = $M^2 - 1$ for stationary noise.

Figure 4.4.9-2 is a plot of predicted interference that would be present at the input to A/D converter in LRU 1. The allowable noise is also plotted. As we see from this plot, unsatisfactory system performance can be expected unbalanced analog video link in the harsh CS114 environment.

3) 100BaseTX Ethernet between probes H & N and probes C & C1

For 100BaseTX Ethernet, IEEE 802.3 clause 25.4.6.3.1 limits differential noise at the receiver to 40 mV p-p for frequencies up to 100 MHz under signal loss conditions of clause 25.4.6.2.1 ($\text{Loss} = 2.1 F^{0.529} + 0.4/F \text{ dB}$ between 1 and 100 MHz). Due to the relatively short nature of the cable in this example, the loss is only 4 dB (see Figure 4.4.7-3). Thus the allowable noise can be increased by $2.1 F^{0.529} + 0.4/F - 4 \text{ dB}$.

Figure 4.4.9-3 is a plot of predicted interference that would be present at the input of the Ethernet receivers in LRU 2 and LRU 1 after the transformer-CM choke. The allowable noise per predicted signal attenuation is also plotted along with allowable noise under maximum allowed signal loss. As we see from this plot, satisfactory system performance in the harsh CS114 environment is expected.

4) DC Prime Power to LRU at probe J

If the power conversion and decoupling capacitors at individual integrated devices in LRU 2 are assumed to provide at least 20 dB of noise attenuation between the prime power filter output (probe J) and the power input to the device containing the most susceptible circuit (analog video at probe K), then the allowable noise at probe J would be 20 dB above what is allowed at probe K. Applying the criteria from 4.4.9.a.1) above, this would equate to $11 \text{ mV}_{\text{rms}} + 20 \text{ dB} = 110 \text{ mV}_{\text{rms}}$.

Figure 4.4.9-4 is a plot of predicted interference that would be present at the input to the power converter inside LRU 2 (probe point J). Also plotted is the allowable noise, adjusted for expected filtering provided by the power conversion and distribution system inside LRU 2 (20 dB) with respect to the most sensitive circuit. As we see from this plot, there is some performance concern between 5 and 10 MHz.

4.4.10. Implement Needed Design Changes, Re-Run Simulation

A modeling and simulation effort may identify design deficiencies with respect to the intended environment. Before replacing the environment entity block (e.g. CS114) with a different environment entity block (e.g. CS115) to complete the modeling and analysis effort, it may be prudent to stop and investigate design changes needed to allow compliant operation under the current environment of interest. Design modification may include electrical changes to I/O circuits, cable shield grounding, circuit grounding, cable shield type and quality, as well interface format.

Clearly in this example there are design deficiencies associated with the analog unbalanced analog video link and the 14 VDC power provided to LRU 2.

Modifications suggested for the unbalanced analog video link include inserting a video isolation transformer at LRU 1 and LRU 3 to reduce the flow of interference current on the coax shield. Also, the modification included changing the coax cable from a type RG-58 to a type RG-212, which offers better shielding.

Modifications suggested for the 14 VDC power circuit include the addition of a high frequency filter capacitor ($1 \mu\text{F}$ w/ $\text{ESR} = 0.05 \text{ ohms}$ and $F_{\text{self_resonance}} = 12 \text{ MHz}$) at probe point J to improve the power input filter at LRU 2.

The following processes are used to analyze design improvement.

- a. Make changes to circuit design and/or cable to address potential susceptibilities identified above (Figures 4.4.10-1 and -2). Note that the S-parameter simulations for the desired signal paths should be repeated if circuit and/or cable changes are expected to change the channel bandwidth. The bandwidth must be adequate for the desired signal.
- b. Re-run simulations with EMC changes included.
- c. Reprocess the results, while retaining the original results for comparison purposes.
- d. Add the curves for the predictions of interference with the modified design to the existing plots of performance with the original design (Figures 4.4.10-3 and 4.4.10-4).

We observe from Figure 4.4.10-3 that notable improvement is made to the analog video link. However, the predicted interference is still somewhat excessive above 4 MHz. A coax cable with better shielding than RG-212 would require a connector changes as well as changes to the overbraid on cable between LRU 1 and LRU 3 due to a notable diameter increase. Since 480i NTSC video can be transmitted in a 4 MHz bandwidth, it may be possible to modify the sending and receiving circuit to utilize only 4 MHz instead of 6 MHz to reduce interference risk. Also, given the harsh EMI environment, the modeling and simulation may support the change of the video interface from unbalanced to balanced.

It is also observed that the filter capacitor added to the LRU 2 power circuit was successful in reducing the interference to acceptable levels.

5. SUMMARY AND CONCLUSIONS

It was shown how the system level electromagnetic environment that a military vehicle may be exposed to can be translated to a set of environments applicable to equipment, subsystems and cable assemblies. These environments are quantified with physical set up conditions to allow compliance verification by test and/or analysis at the equipment or subsystem level with representation of the applicable vehicle cable/connector assemblies.

Electrical and mathematical models were developed and presented for the susceptibility environments applicable to cable/connector assemblies. The specific environments for military ground vehicles were MIL-STD-461F CS114, CS115, CS116, RS103 and RS105. They present radio frequency environments in the frequency domain and the time domain, as well as transients directly coupled to cable harnesses.

It was shown how an “entity block” approach to modeling and simulation can be used to allow model reuse and construction of the models for various parts of the problem space simultaneously and independently to provide greater throughput than previously realized before the availability of co-simulation features in electromagnetic effects analysis tools. Details of the entity block approach were presented. Methods of specifying and modeling each type of entity block with respect to state-of-the-art electromagnetic compatibility focused simulation tools were developed.

Since immunity to electromagnetic effects as applied to cable/connector assemblies is greatly influenced by the design of the cable’s shielding, the topic of transfer impedance was discussed. Methods of measurement, specification and input to simulation tools were provided.

Finally, a very detailed example involving the application of a harsh conducted RF susceptibility environment to a multi-branch cable assembly was presented to illustrate the entity block approach to electromagnetic simulation for complex systems. The example included not only the application of the environment model, but also construction of the cable harness, connector assemblies, shield terminations, EMI filters and active I/O device models for integration into a co-simulation environment. The realistic subsystem modeled included power, analog video, digital video, and digital high-speed data interfaces using cable types and shielding typical of what may be found in military vehicles. The results of the modeling and simulation illustrated some very important and basic system EMC design principles, most notable being the benefits of balanced interfaces, digitization of analog signals at the source, grounding methods to control the interference current flow through cables, and the importance of high-quality filter components.

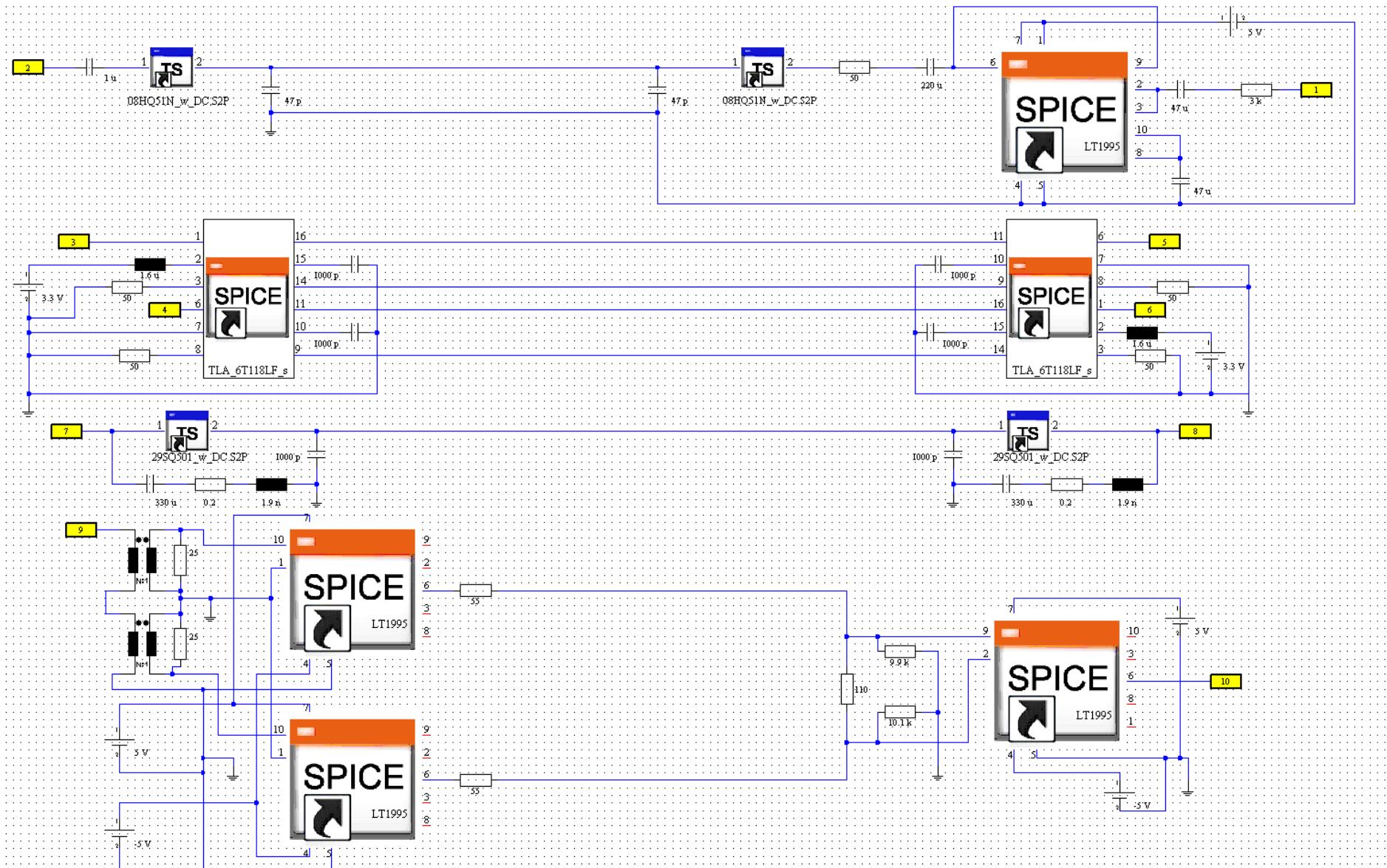


Figure 4.4.1-1. I/O and Signaling Schematic with Common Ground - CST Design Studio

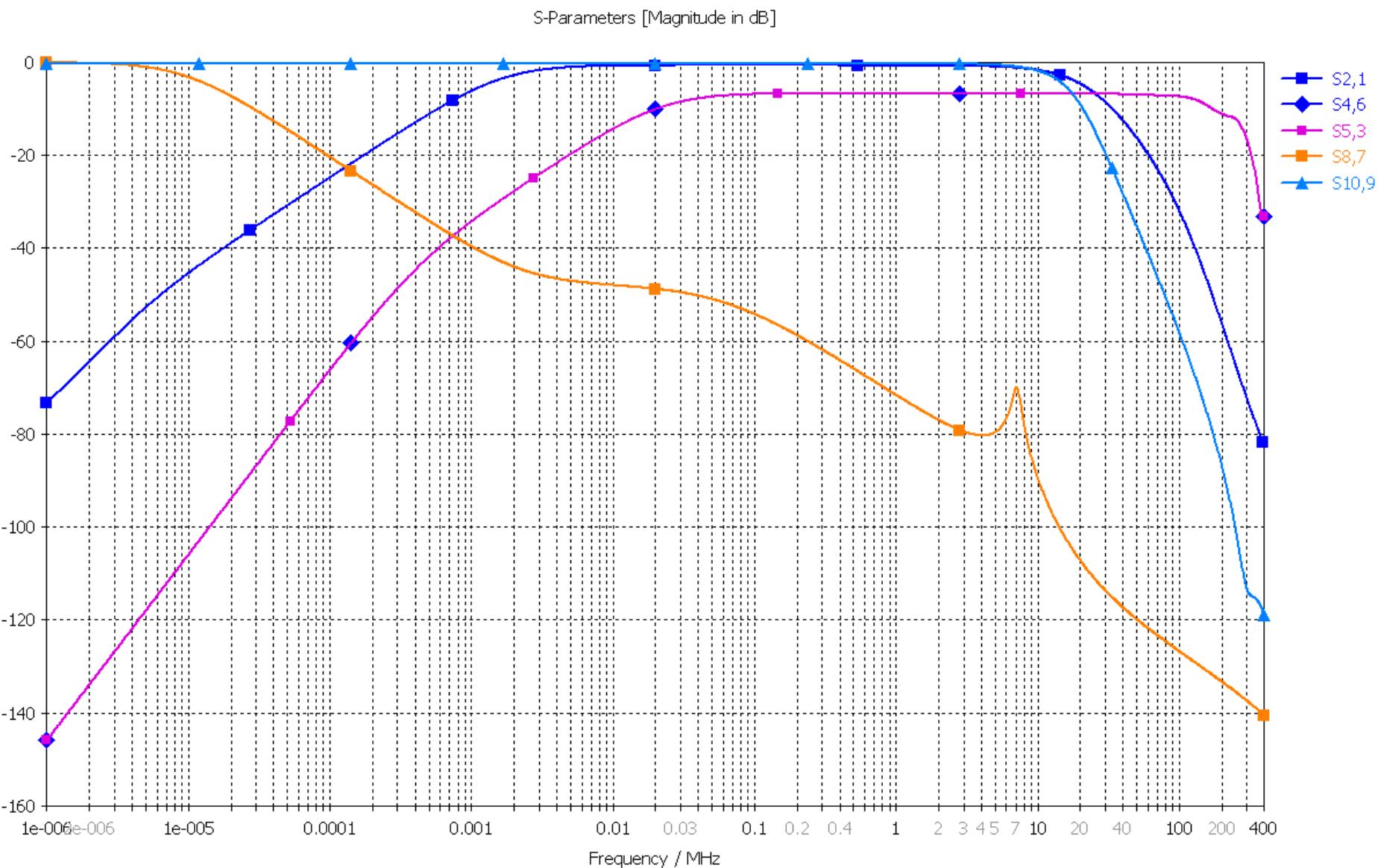


Figure 4.4.1-2. I/O and Signaling Schematic with Common Ground – Results - Gain in Desired Direction

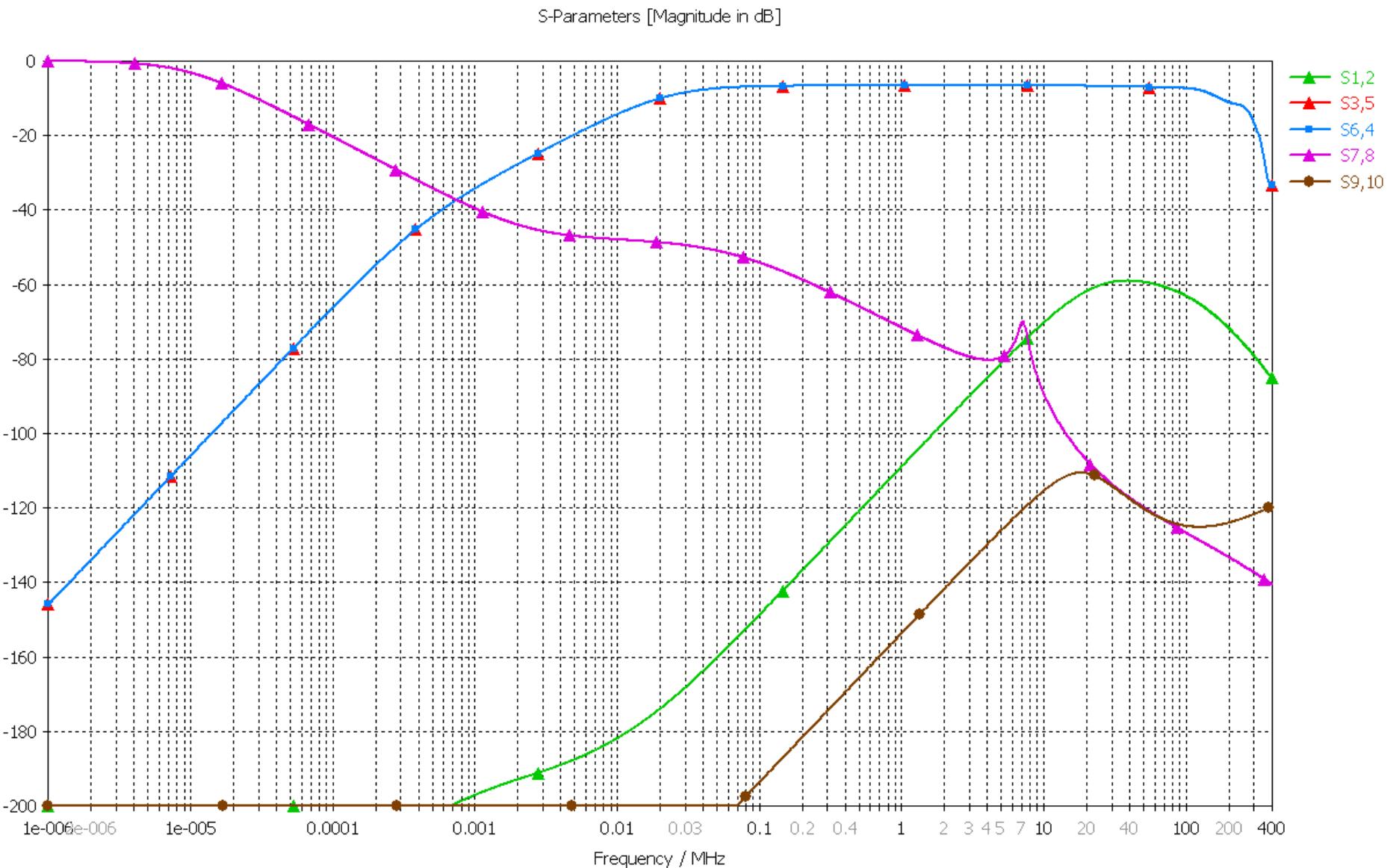


Figure 4.4.1-3. I/O and Signaling Schematic with Common Ground – Results - Gain in Reverse Direction

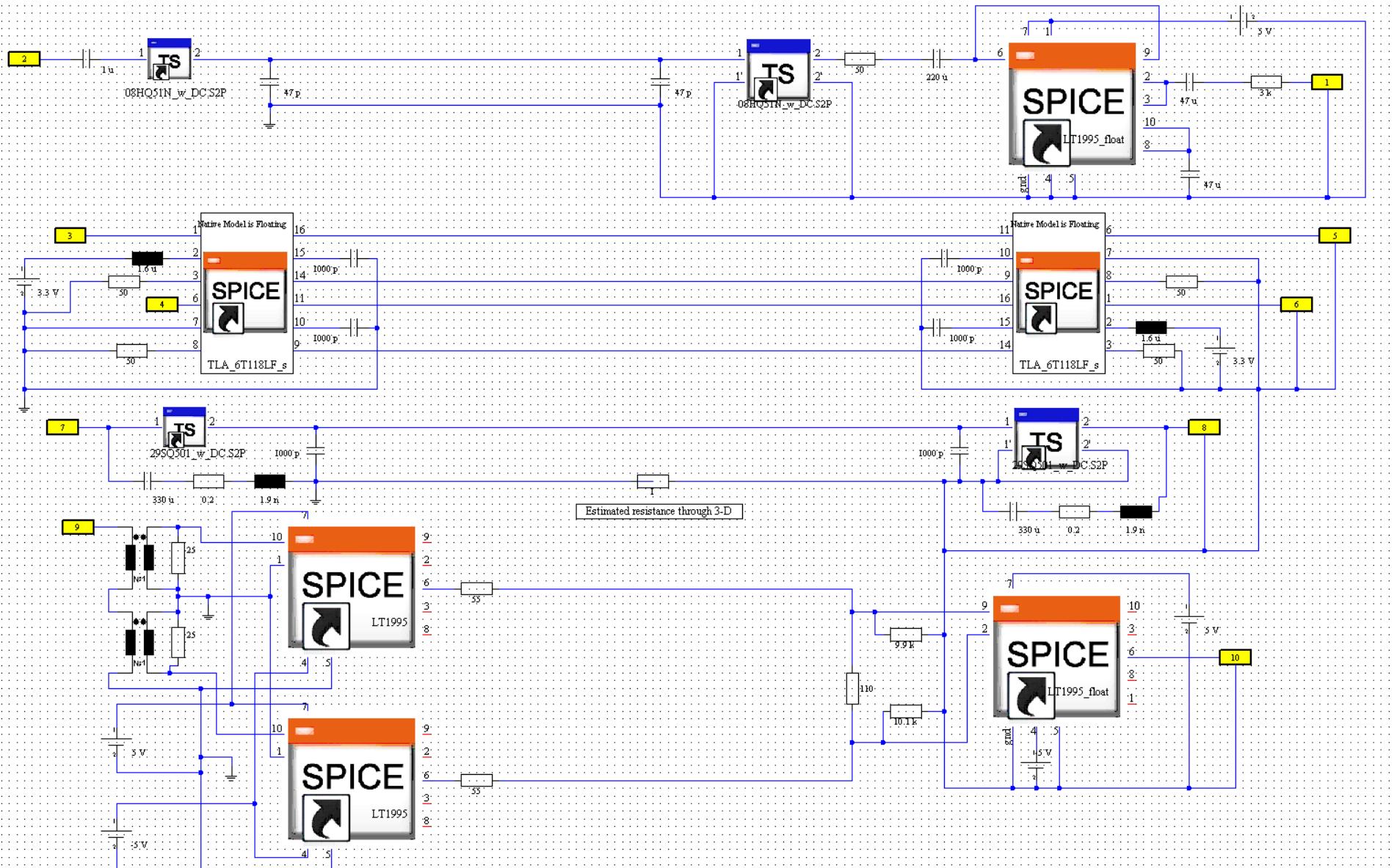


Figure 4.4.2-1. I/O and Signaling Schematic with Target System Floating Grounds - CST Design Studio

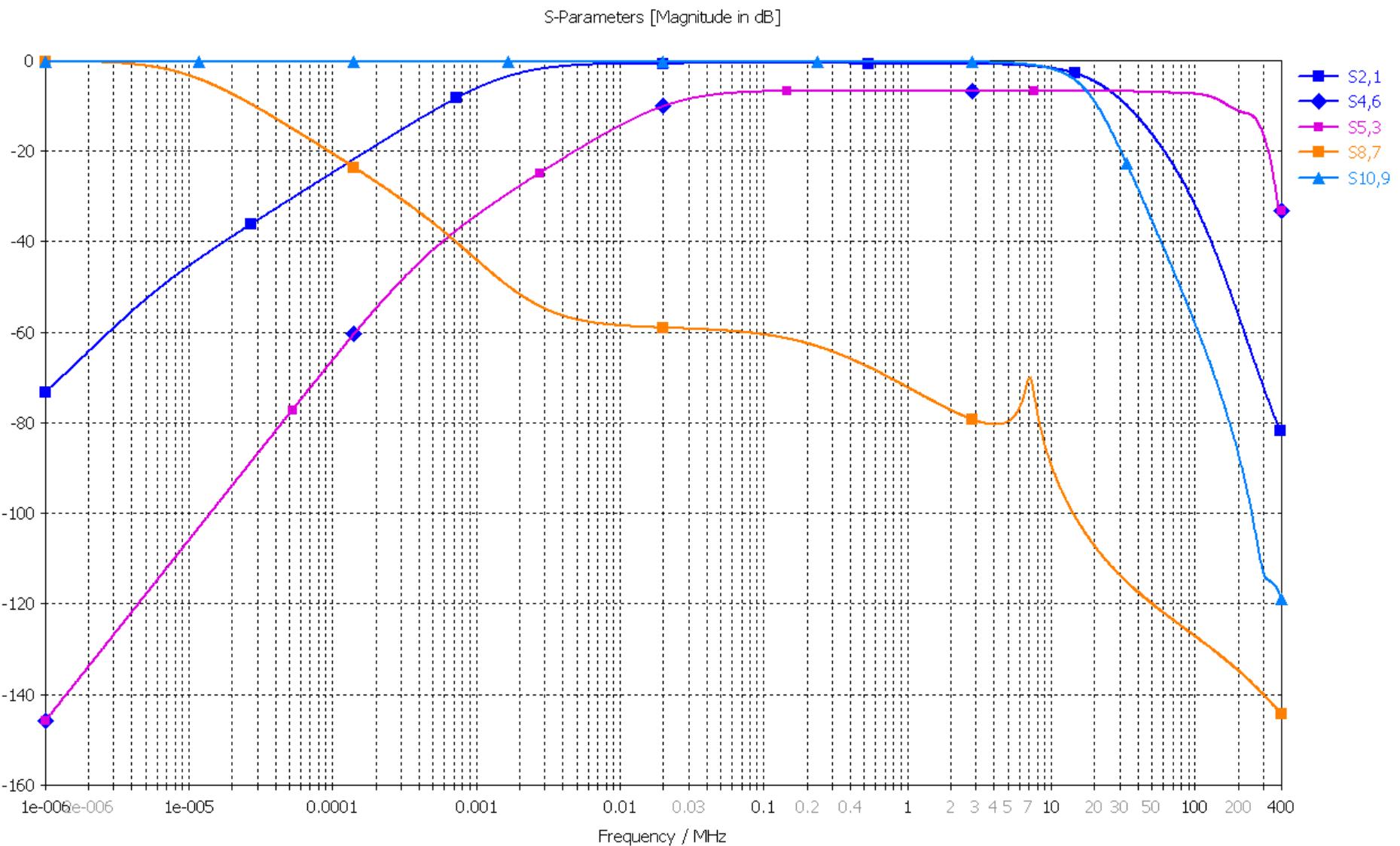


Figure 4.4.2-2. I/O and Signaling Schematic with Target System Floating Ground – Results - Gain in Desired Direction

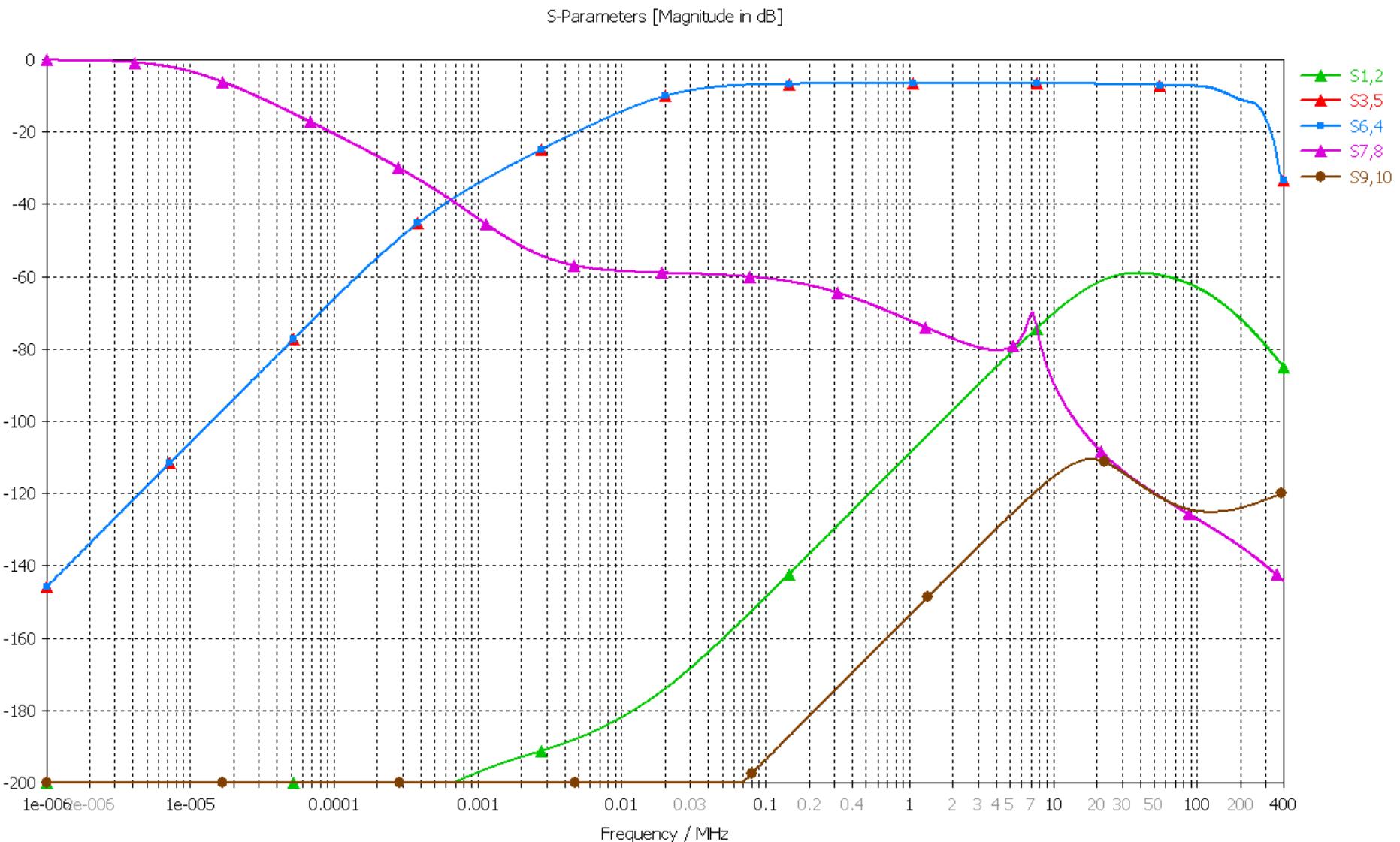


Figure 4.4.2-3. I/O and Signaling Schematic with Target System Floating Ground – Results - Gain in Reverse Direction

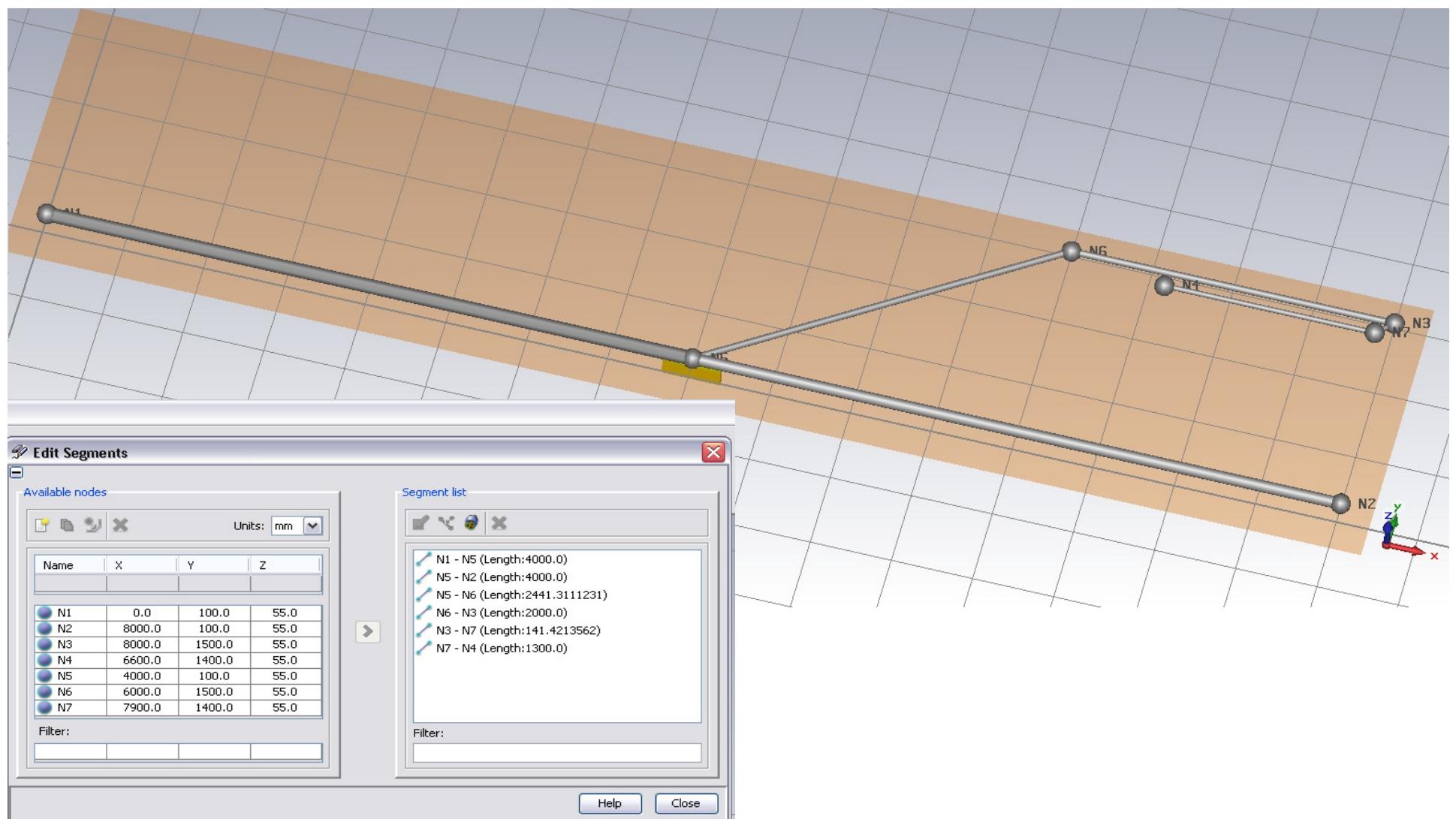


Figure 4.4.3-1. Cable Harness Layout and Build – CST Cable Studio

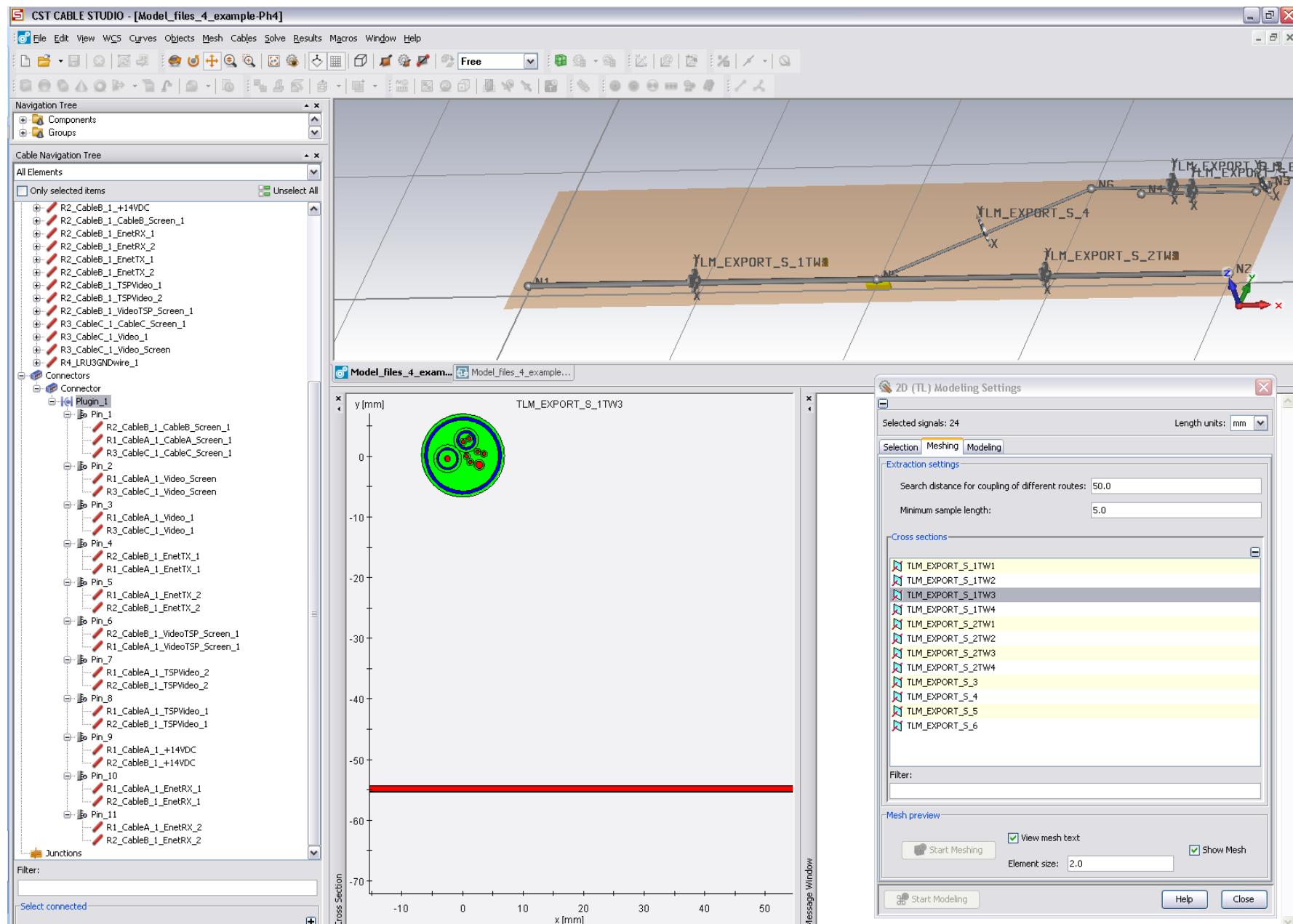


Figure 4.4.3-2. Cable Harness Build and Connections – CST Cable Studio

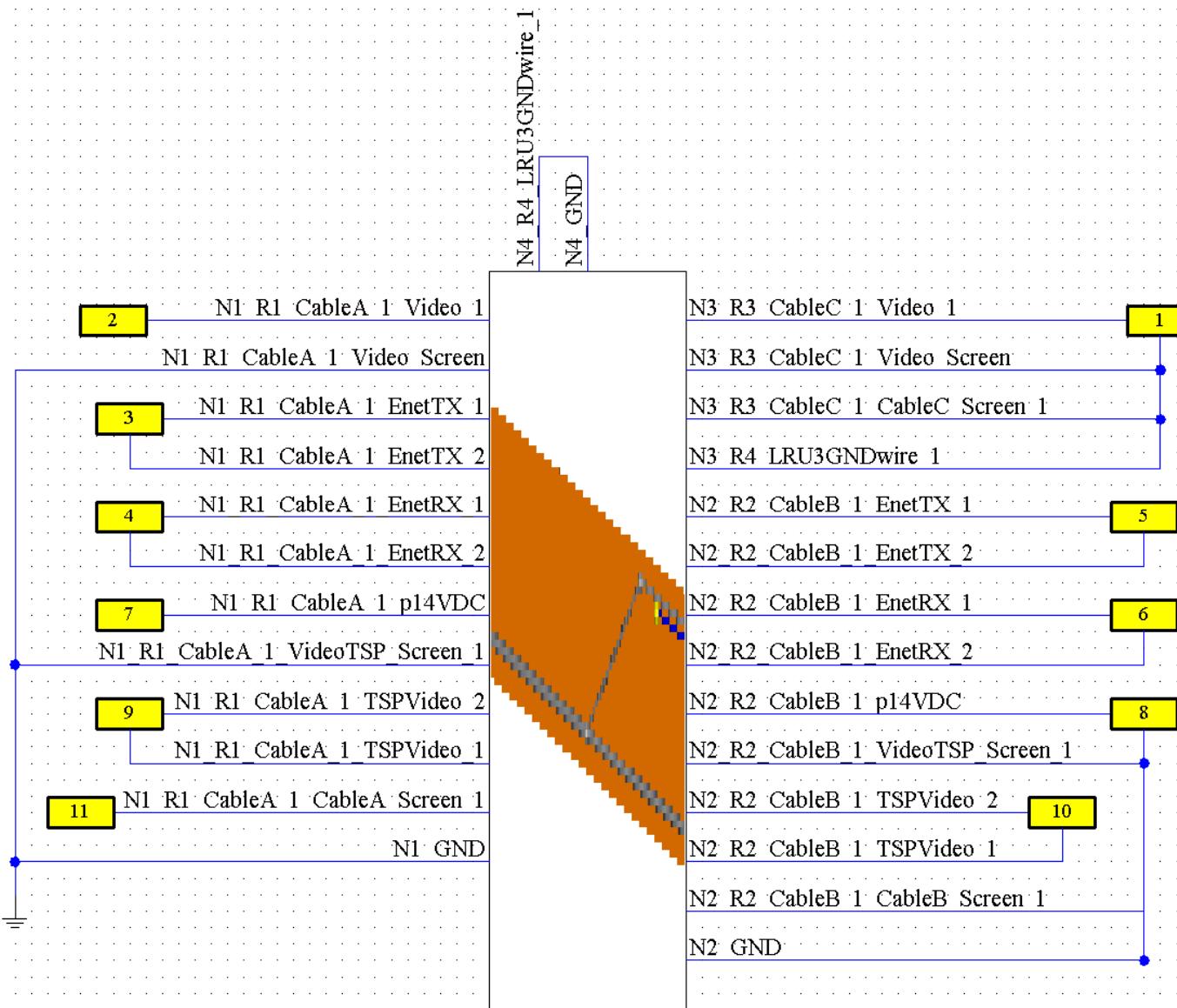


Figure 4.4.3-3. Cable Harness Imported in to Co-Simulation Framework for Transfer Checkout

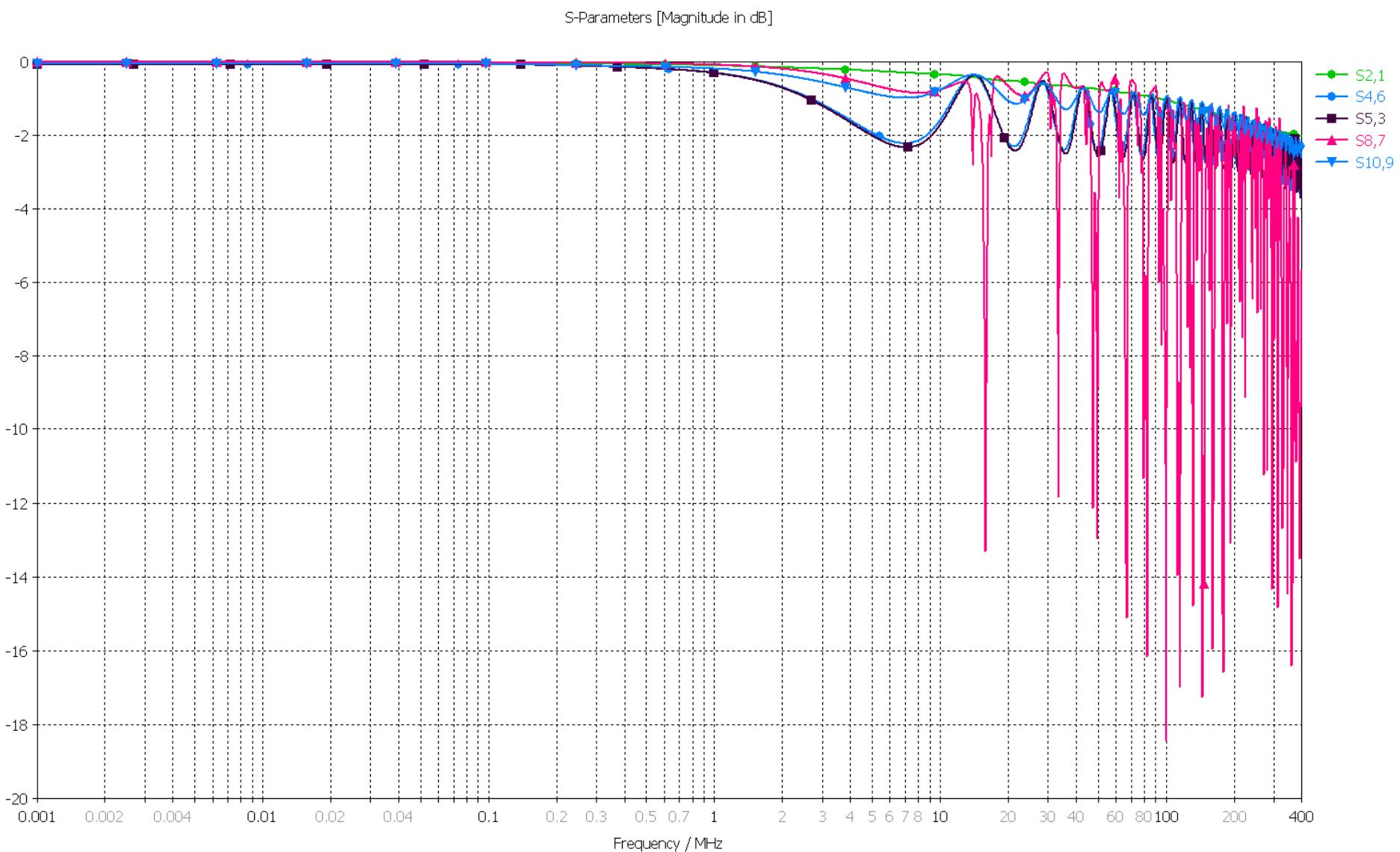


Figure 4.4.3-4. Cable Harness Signal Transfer Simulation – Desired Path Transfer

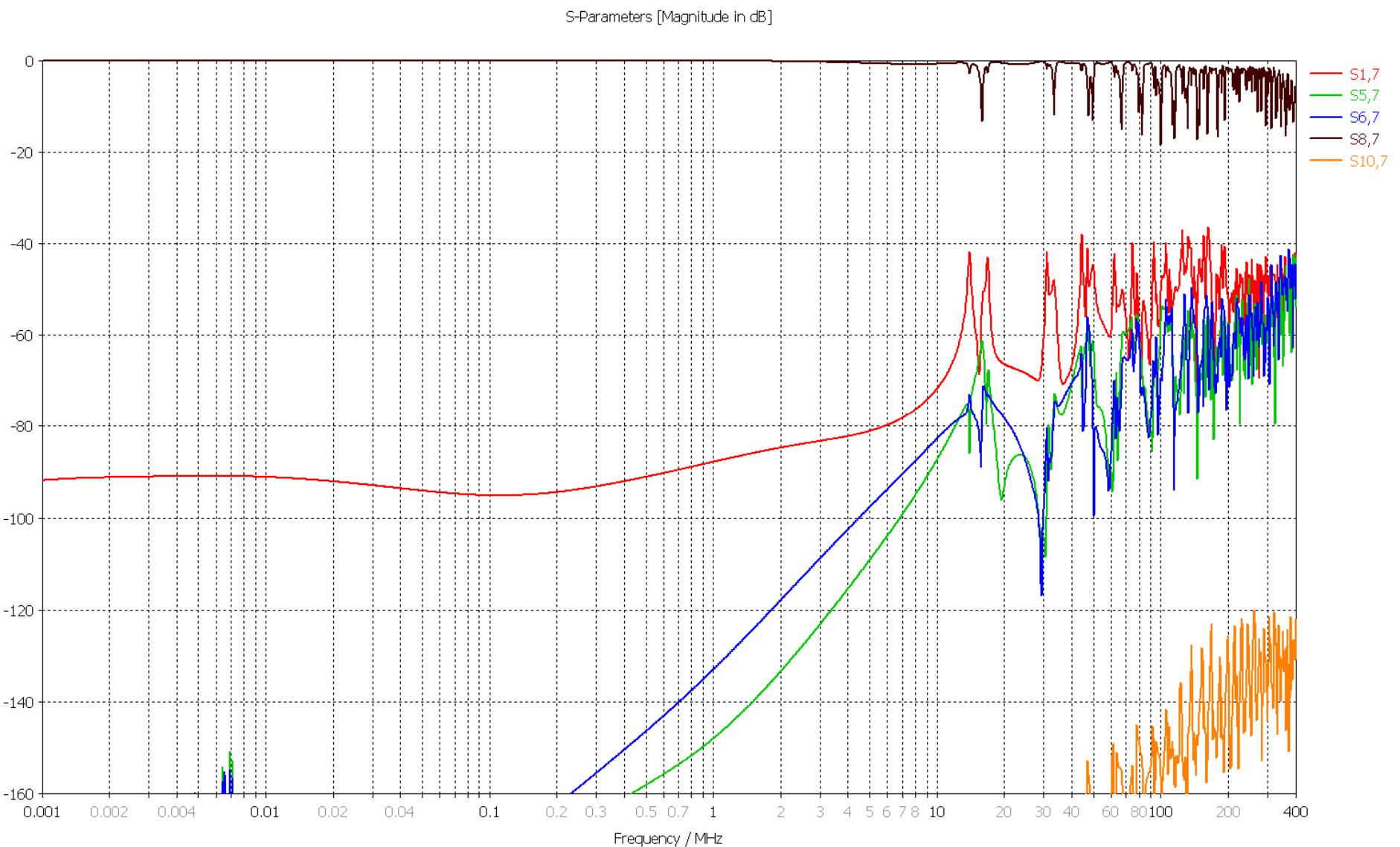


Figure 4.3-5. Cable Harness Signal Transfer Simulation – Cross-talk

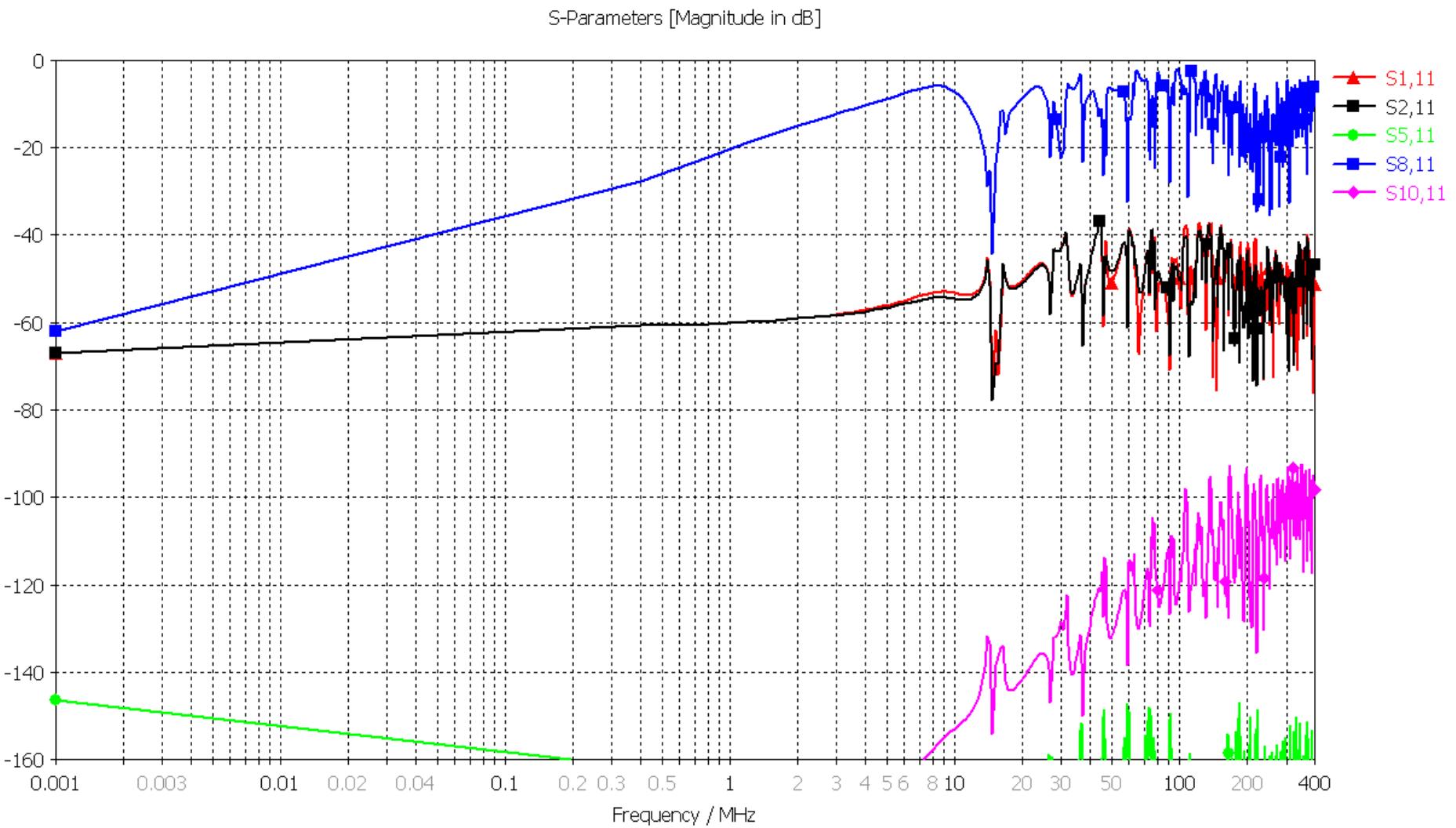


Figure 4.4.3-6. Cable Harness Signal Transfer Simulation – Coupling From Overshield

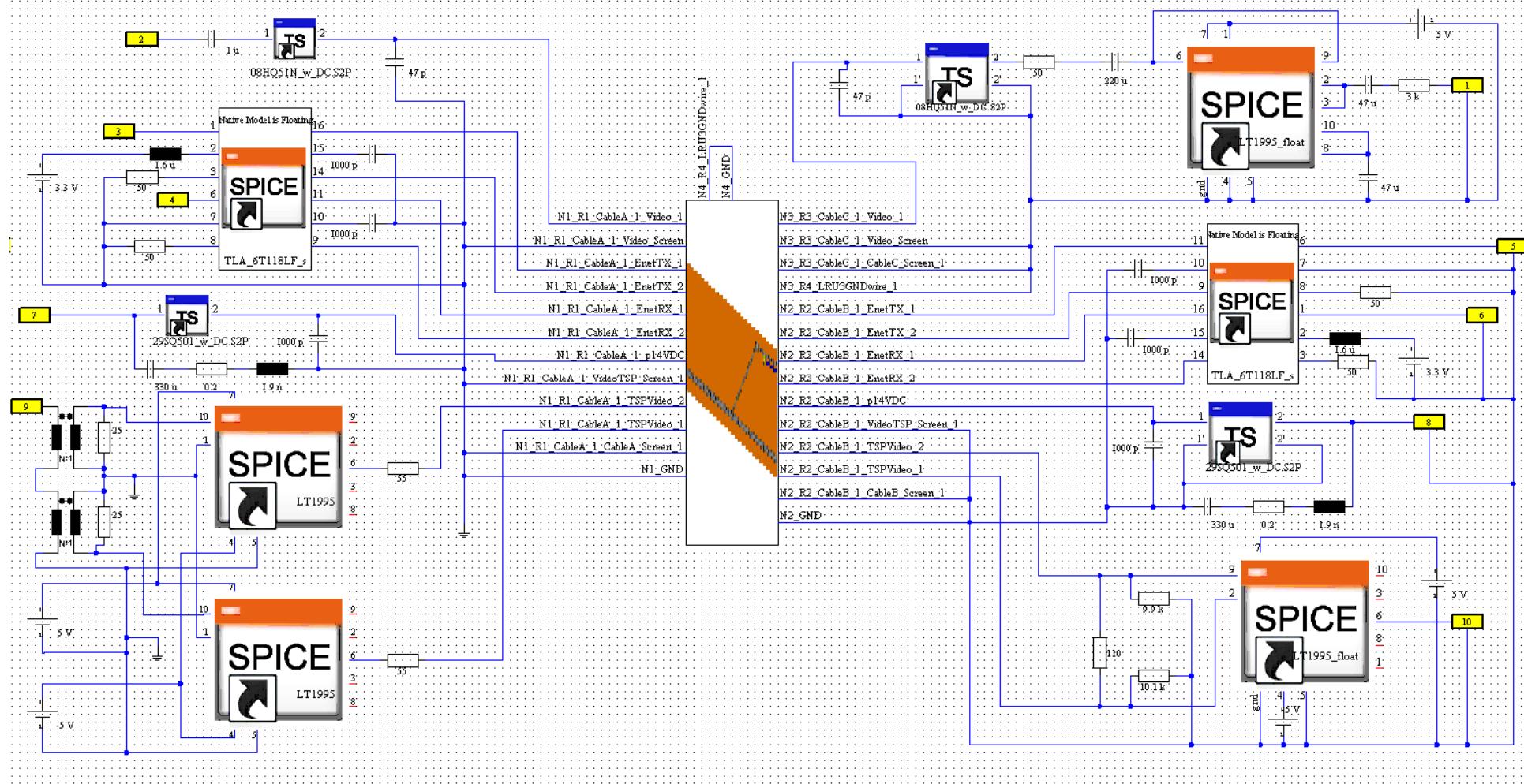


Figure 4.4.4-1. Cable Harness and Interface Circuits Imported in to Co-Simulation Framework for Transfer Checkout

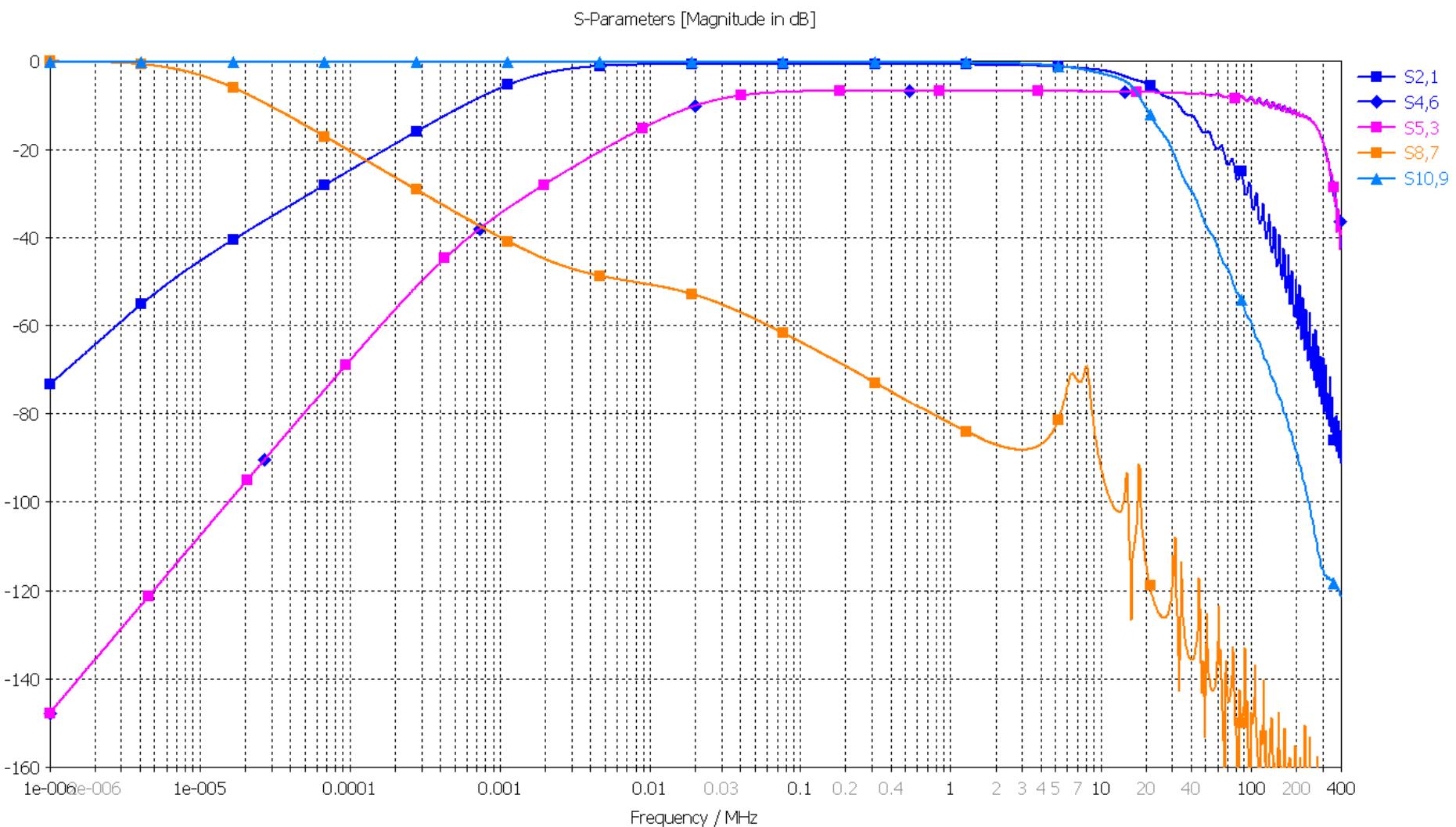


Figure 4.4.4-2. Cable Harness with Interface Circuits Signal Transfer Simulation – Desired Path Transfer

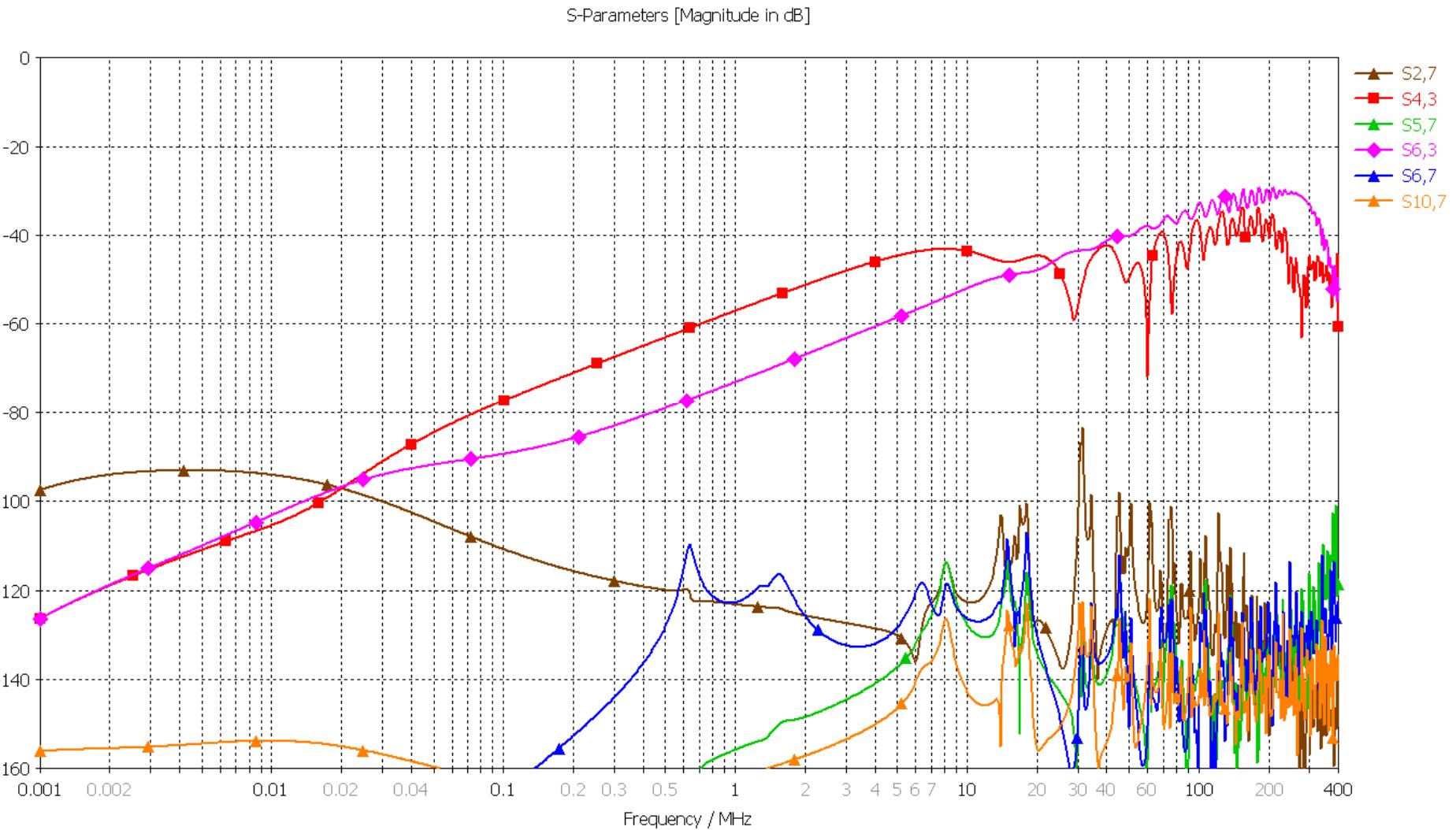


Figure 4.4.4-3. Cable Harness with Interface Circuits Signal Transfer Simulation – Cross-talk

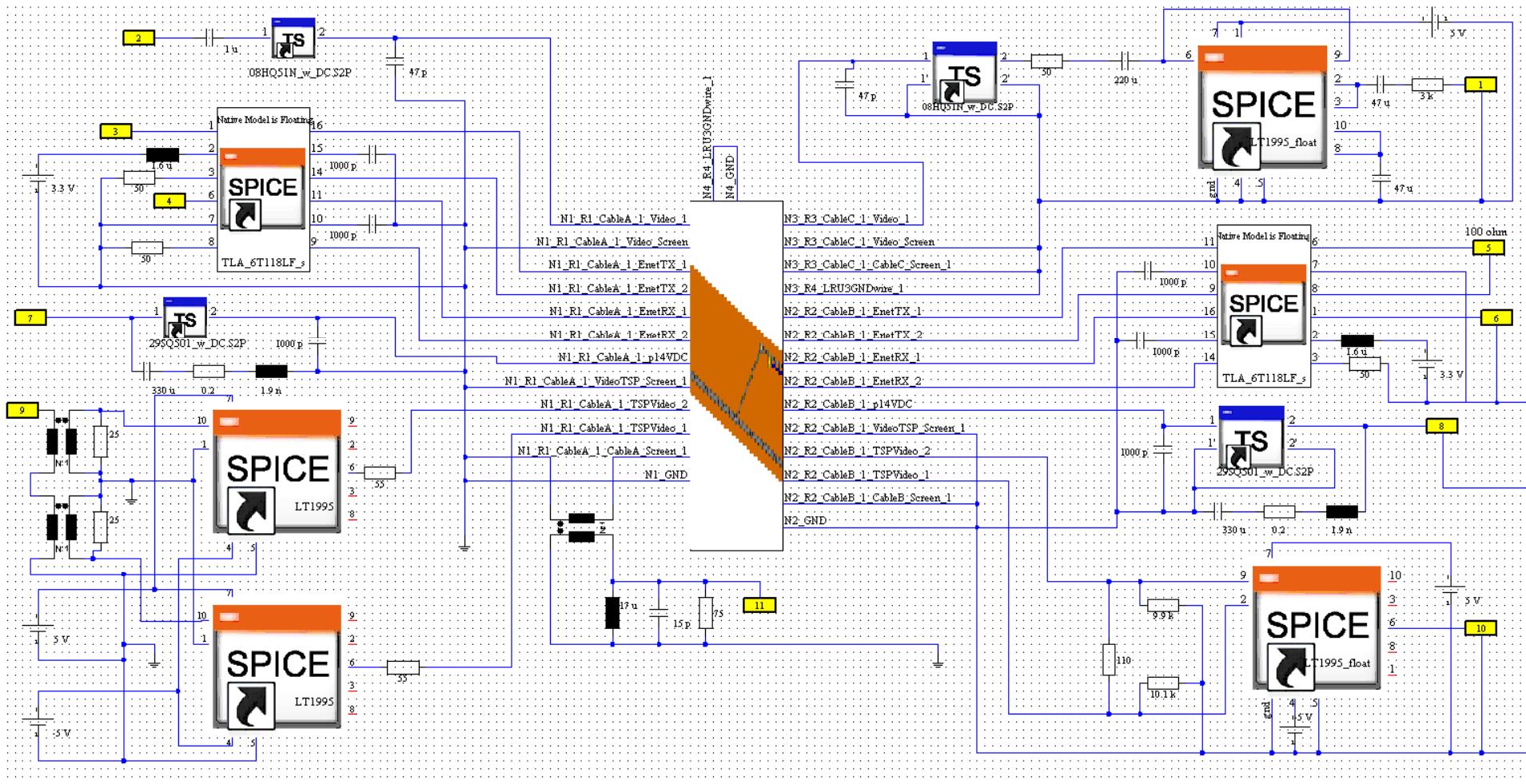


Figure 4.4.5-1. Cable Harness, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout

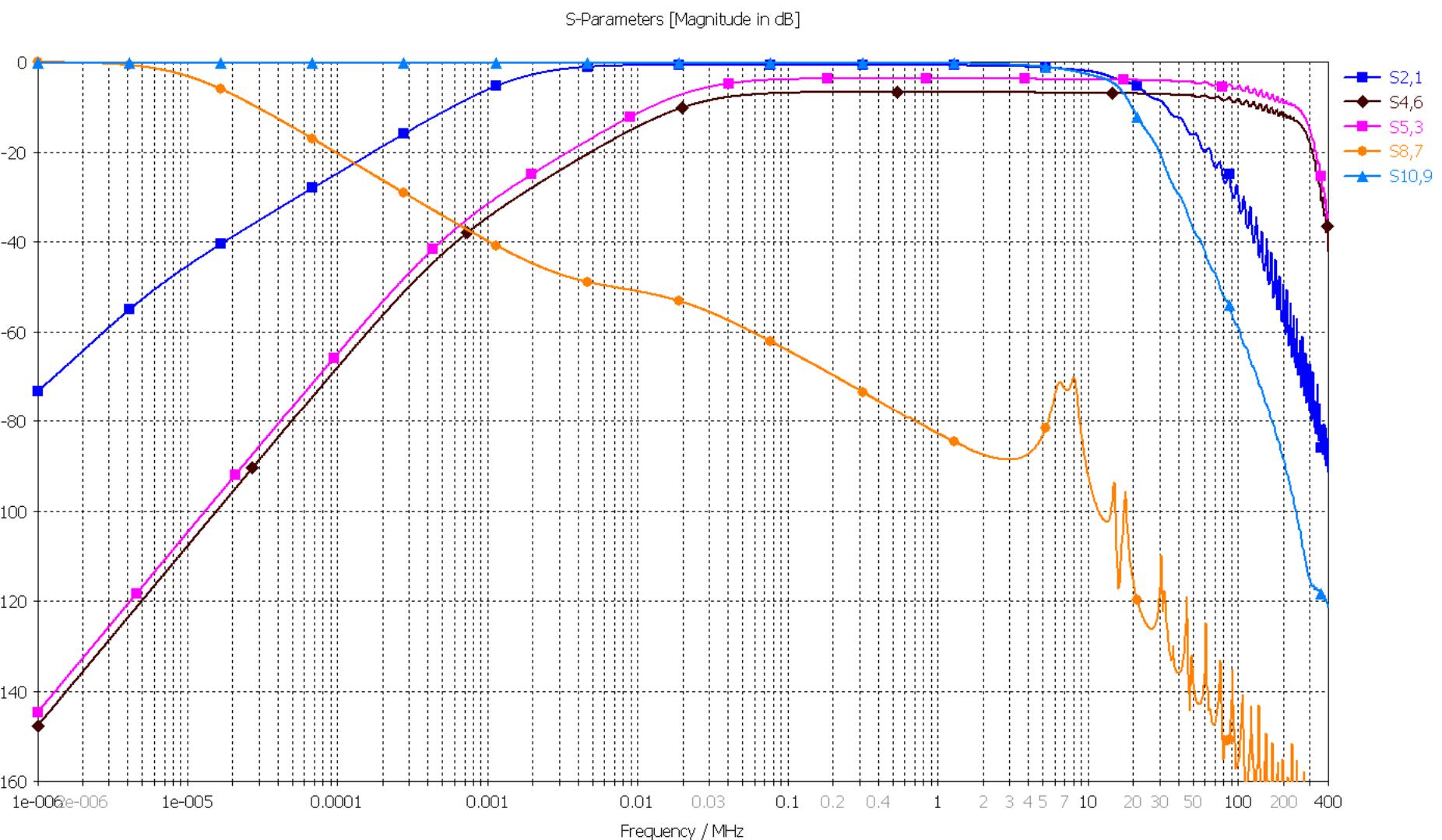


Figure 4.4.5-2. Cable Harness, Interface Circuits and CS114 Environment Signal Transfer Simulation – Desired Path Transfer

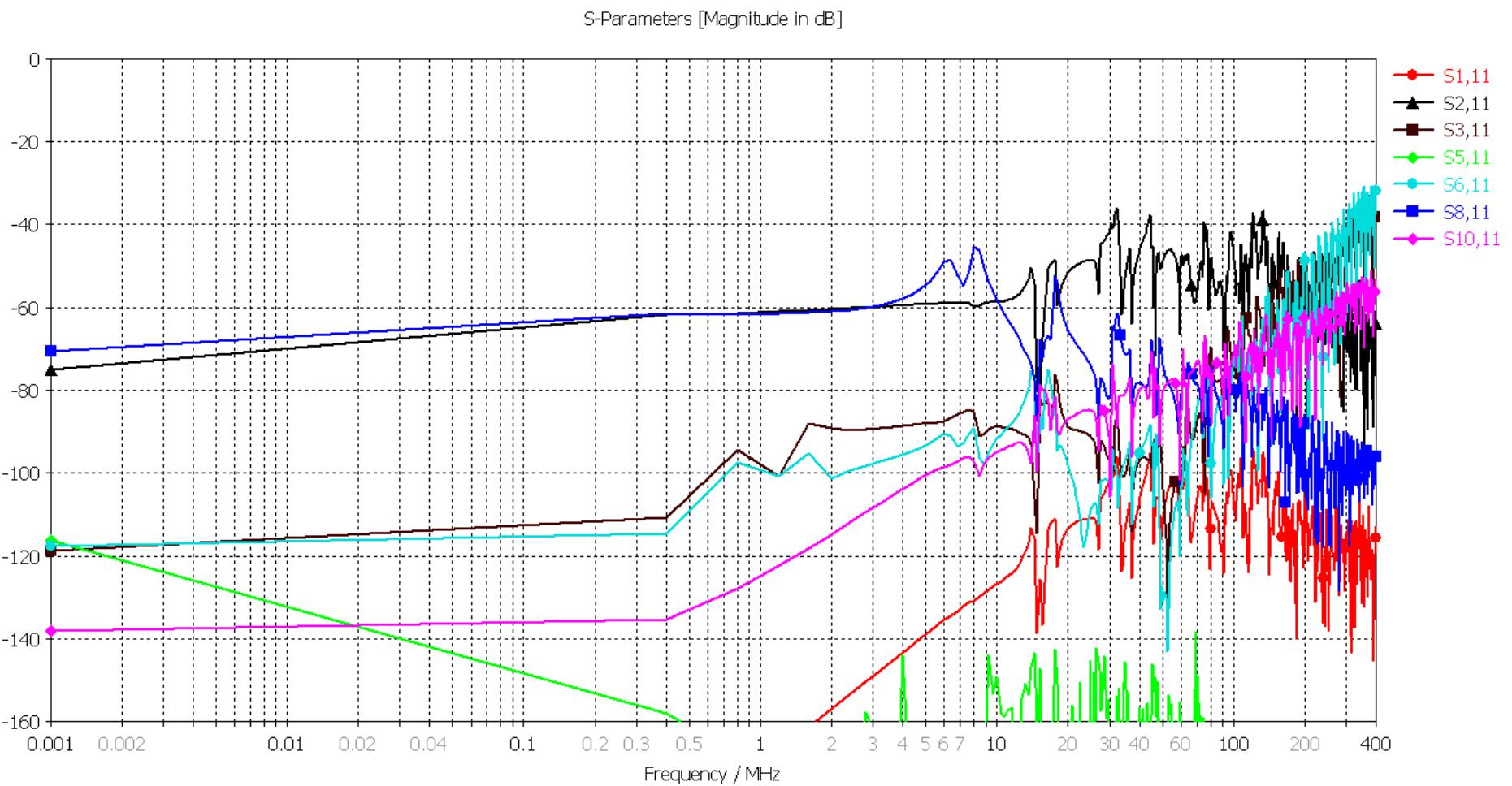


Figure 4.4.5-3. Cable Harness, Interface Circuits and CS114 Environment Signal Transfer Simulation – Coupling From Overshield

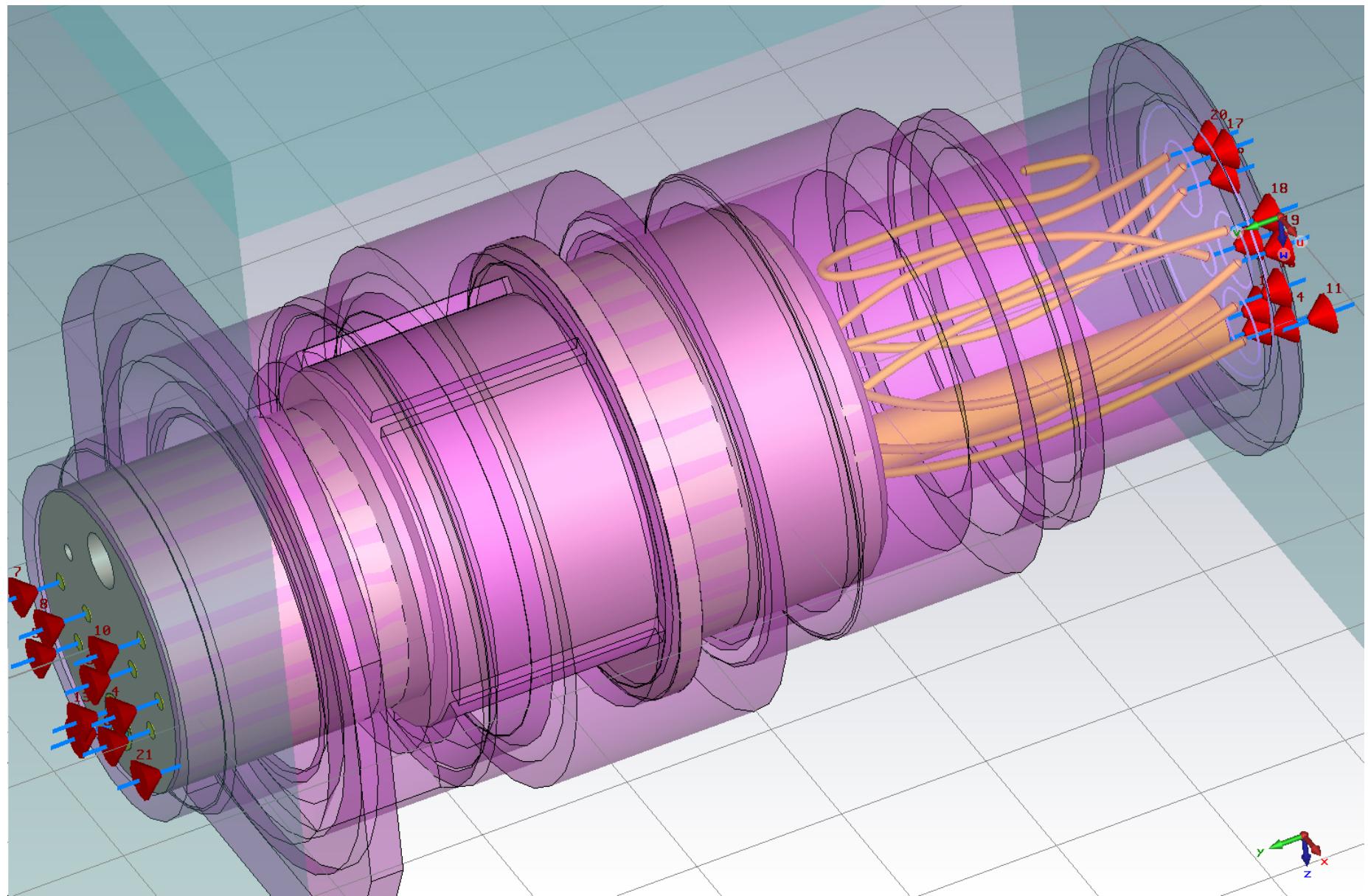


Figure 4.4.6-1. LRU 1 Connectors and Transition 3D Model - CST Microwave Studio

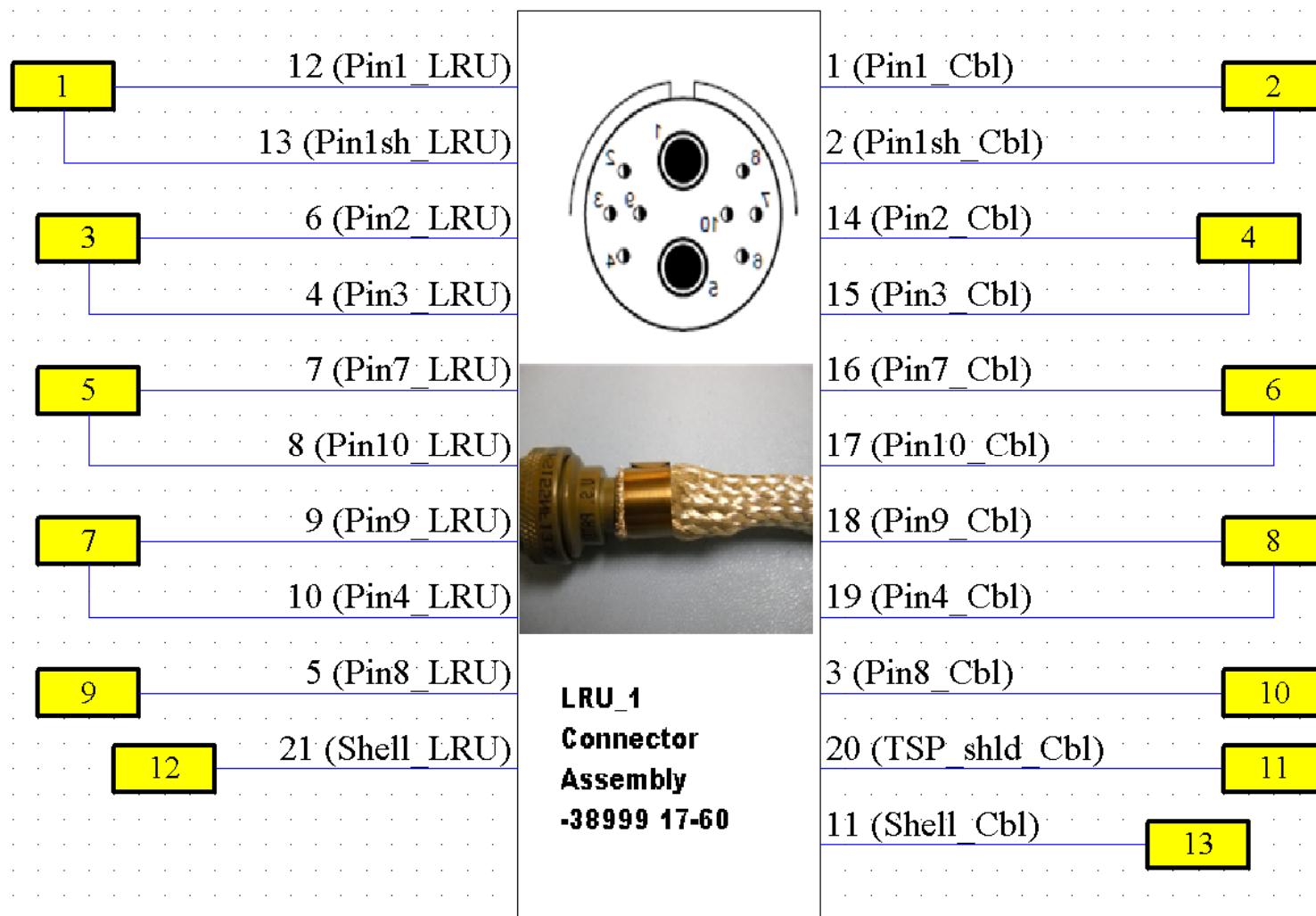


Figure 4.4.6-2. LRU 1 Connectors/Transition Block in Co-Simulation Framework for Transfer Checkout

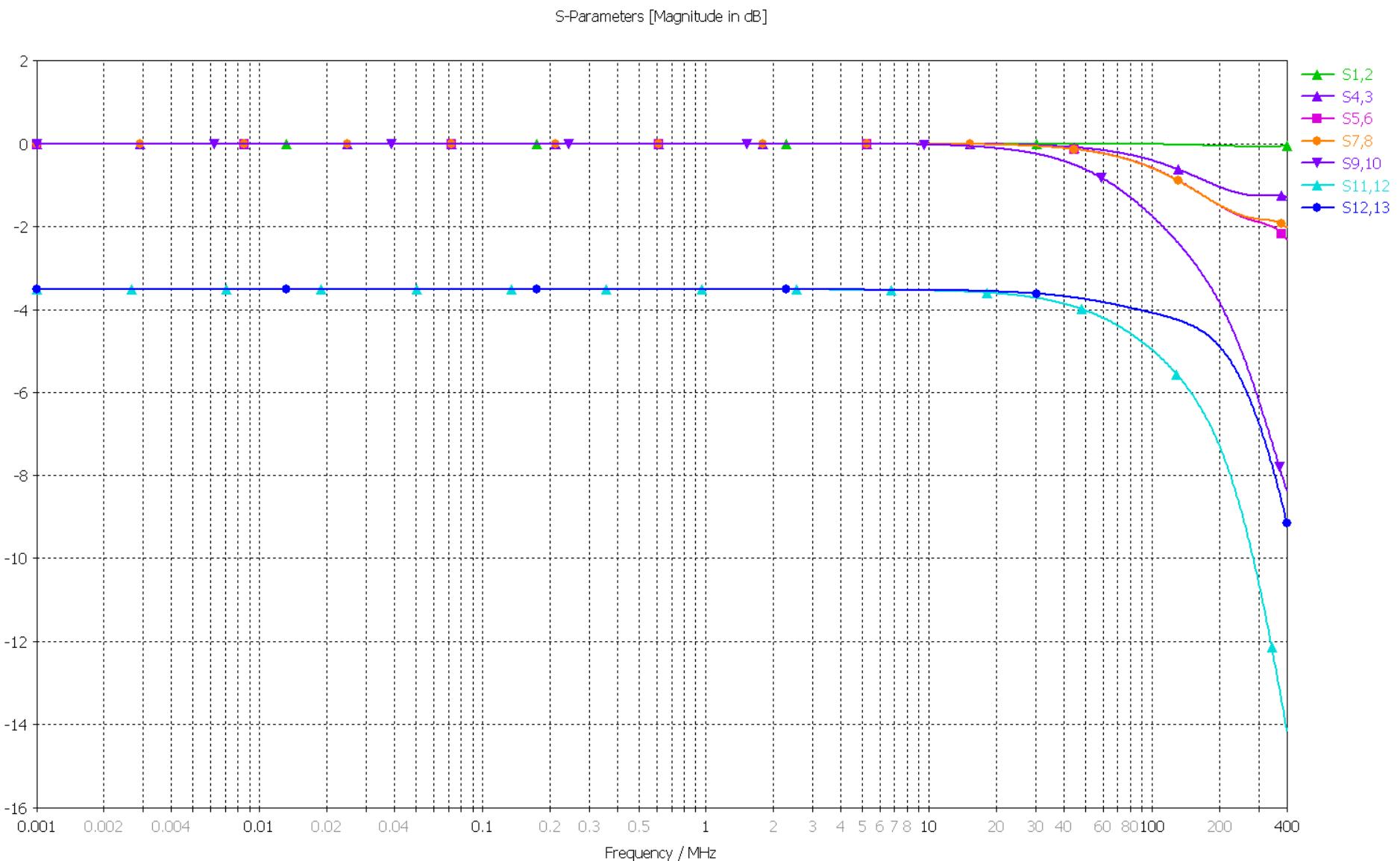


Figure 4.4.6-3. LRU 1 Connectors/Transition Block – Transfer through Desired Signal Paths

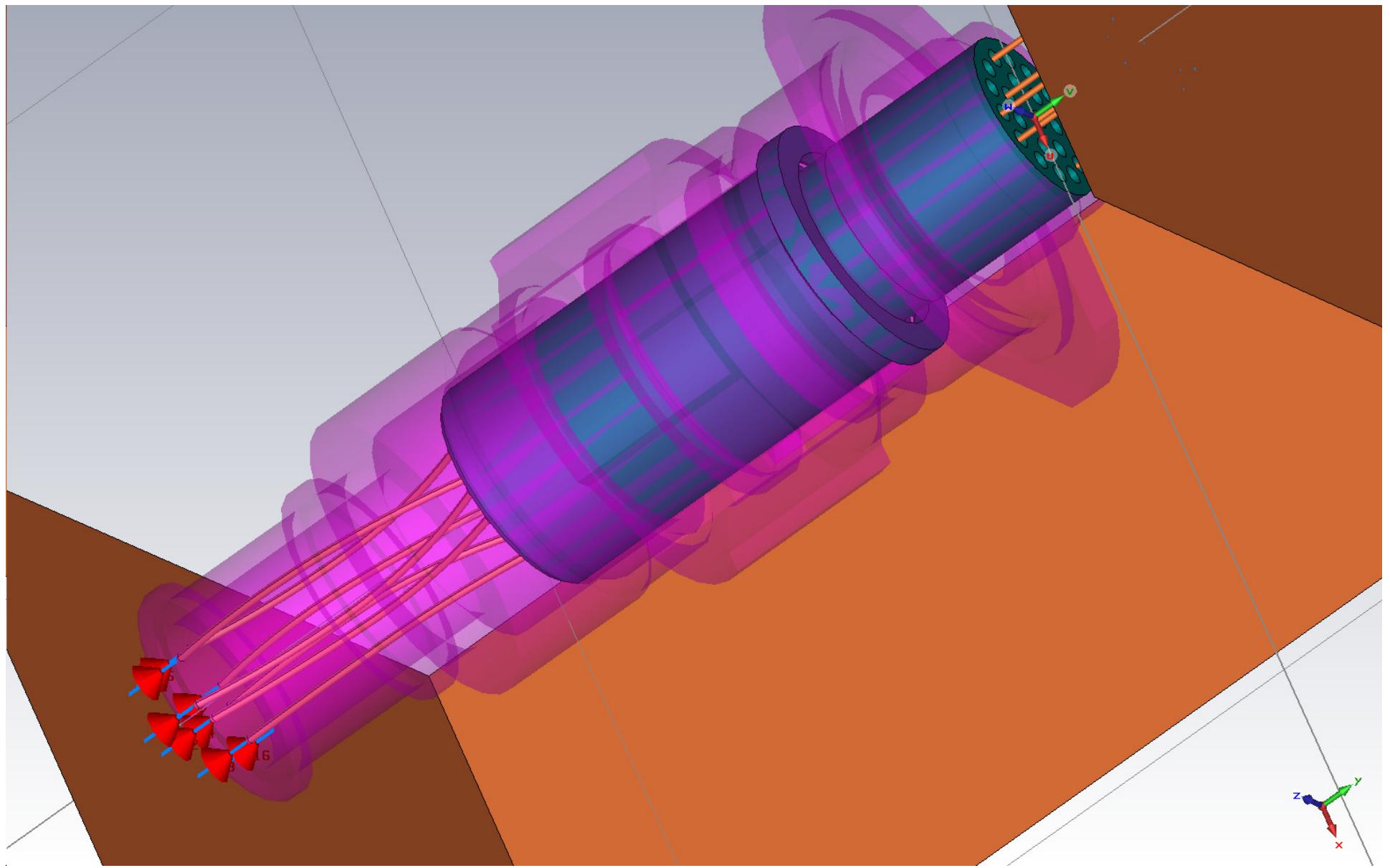


Figure 4.4.6-4. LRU 2 Connectors and Transition 3D Model - CST Microwave Studio

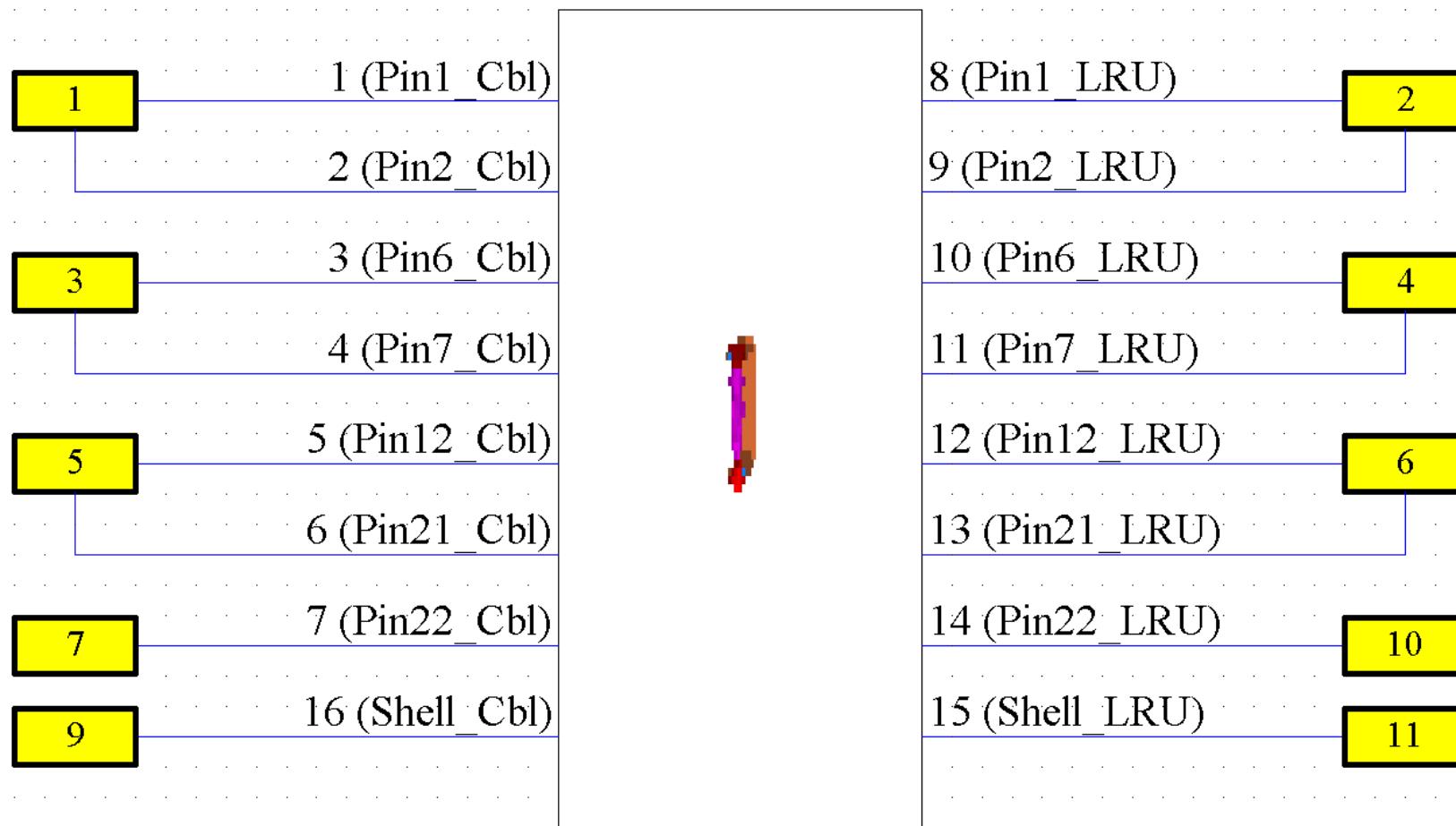


Figure 4.4.6-5. LRU 2 Connectors/Transition Block in Co-Simulation Framework for Transfer Checkout

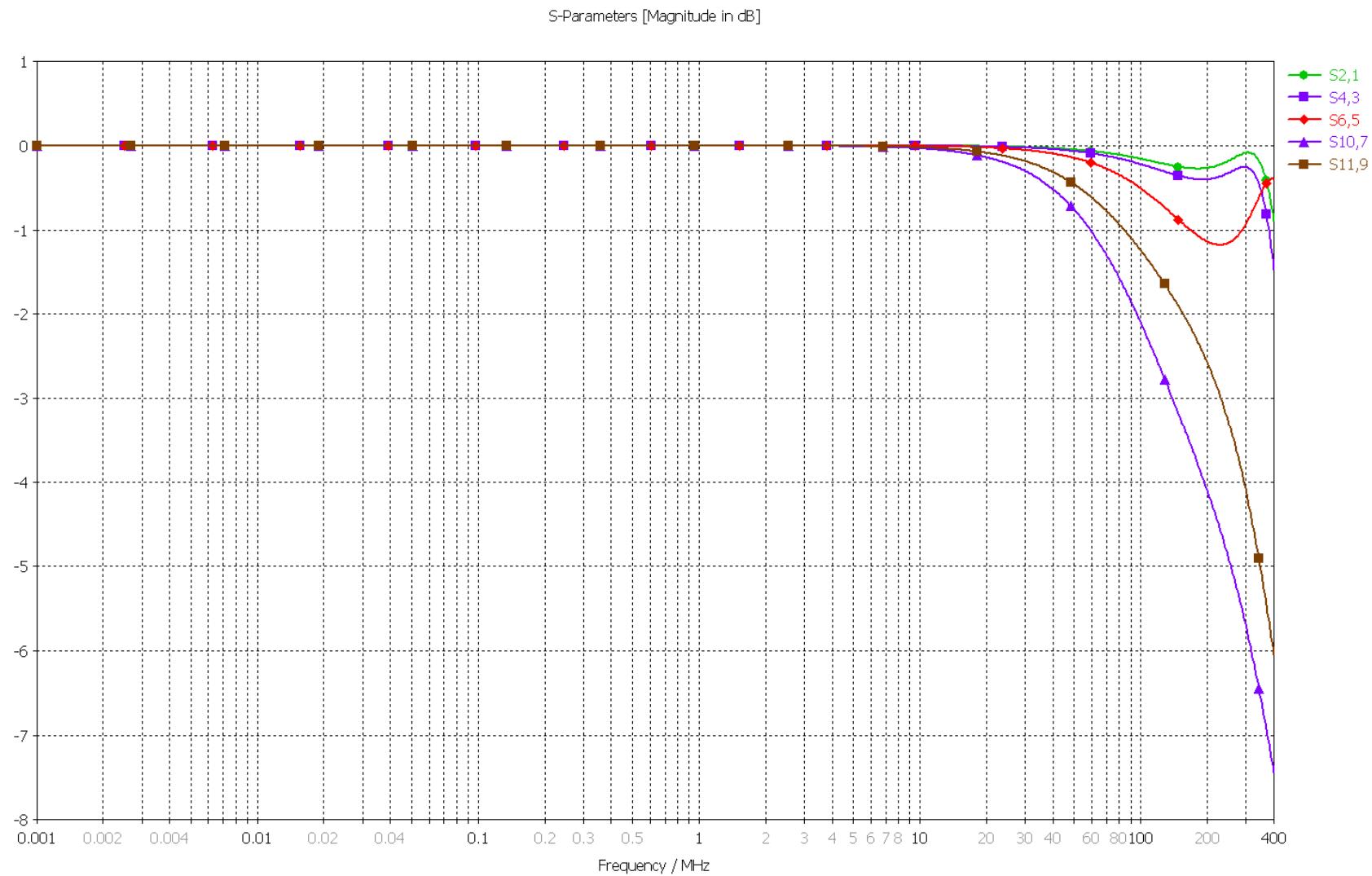


Figure 4.4.6-6. LRU 2 Connectors/Transition Block – Transfer through Desired Signal Paths

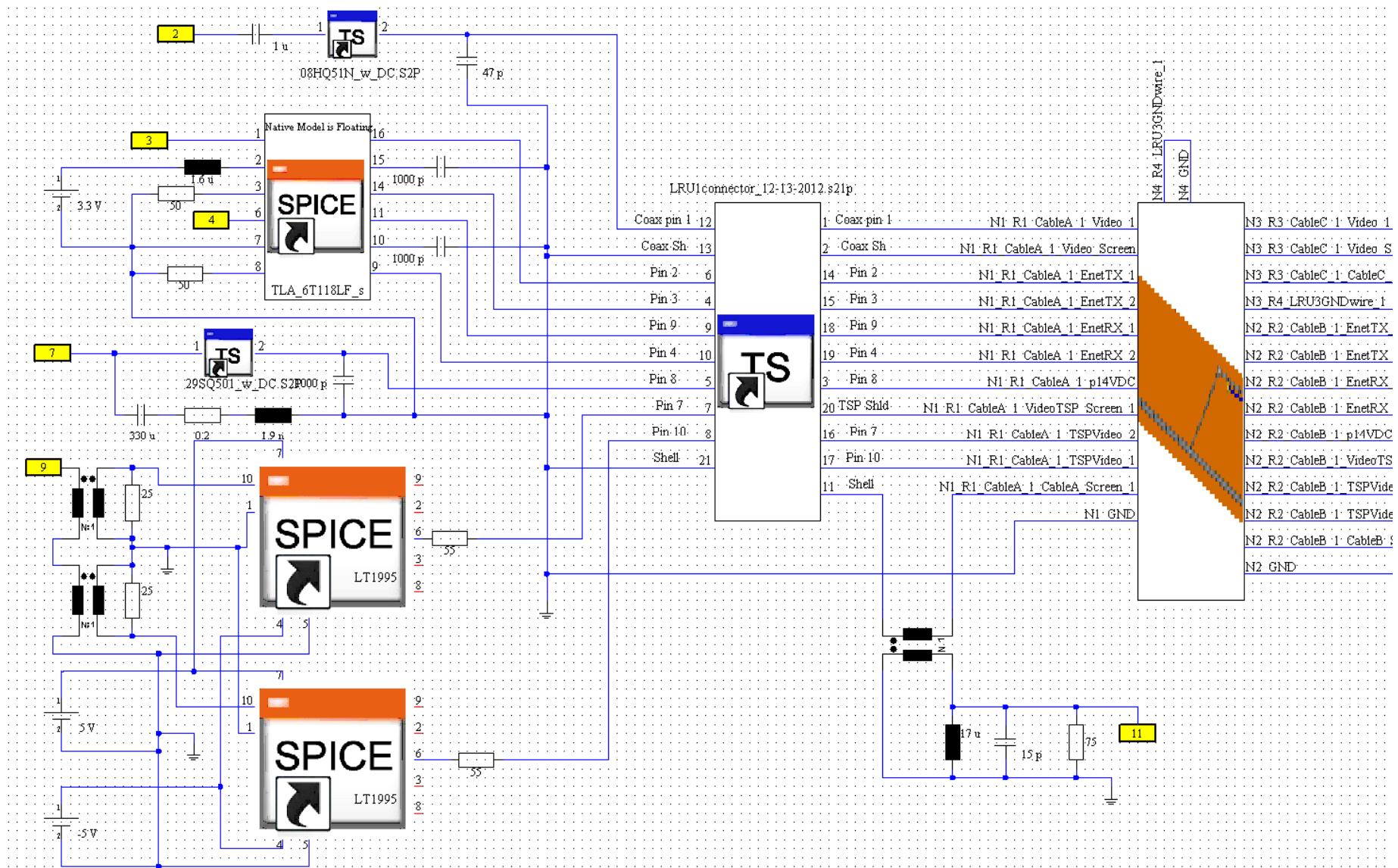


Figure 4.4.7-1. Cable Harness, Connectors, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout - left side

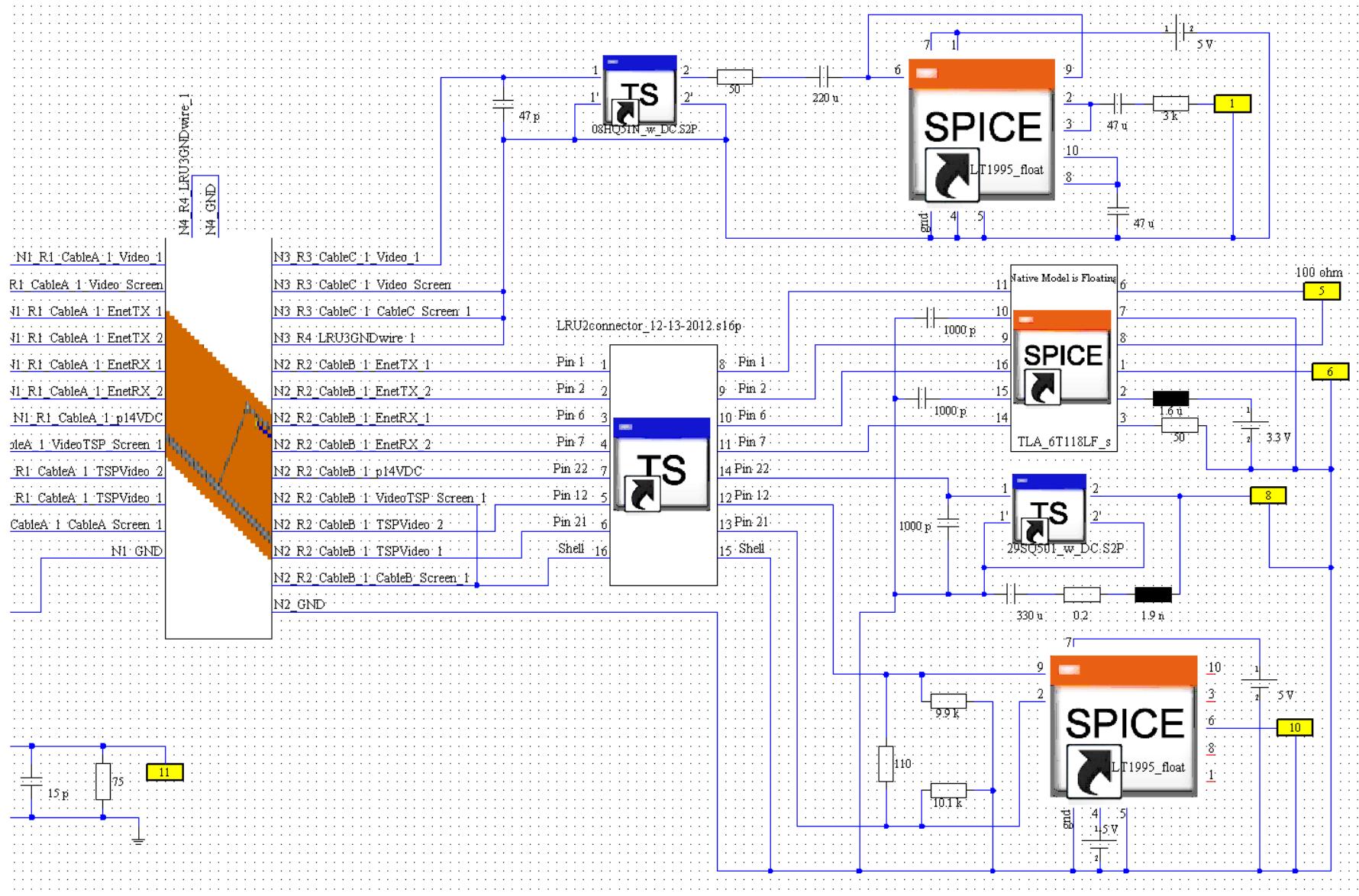


Figure 4.4.7-2. Cable Harness, Connectors, Interface Circuits and CS114 Environment Imported in to Co-Simulation Framework for Transfer Checkout – right side

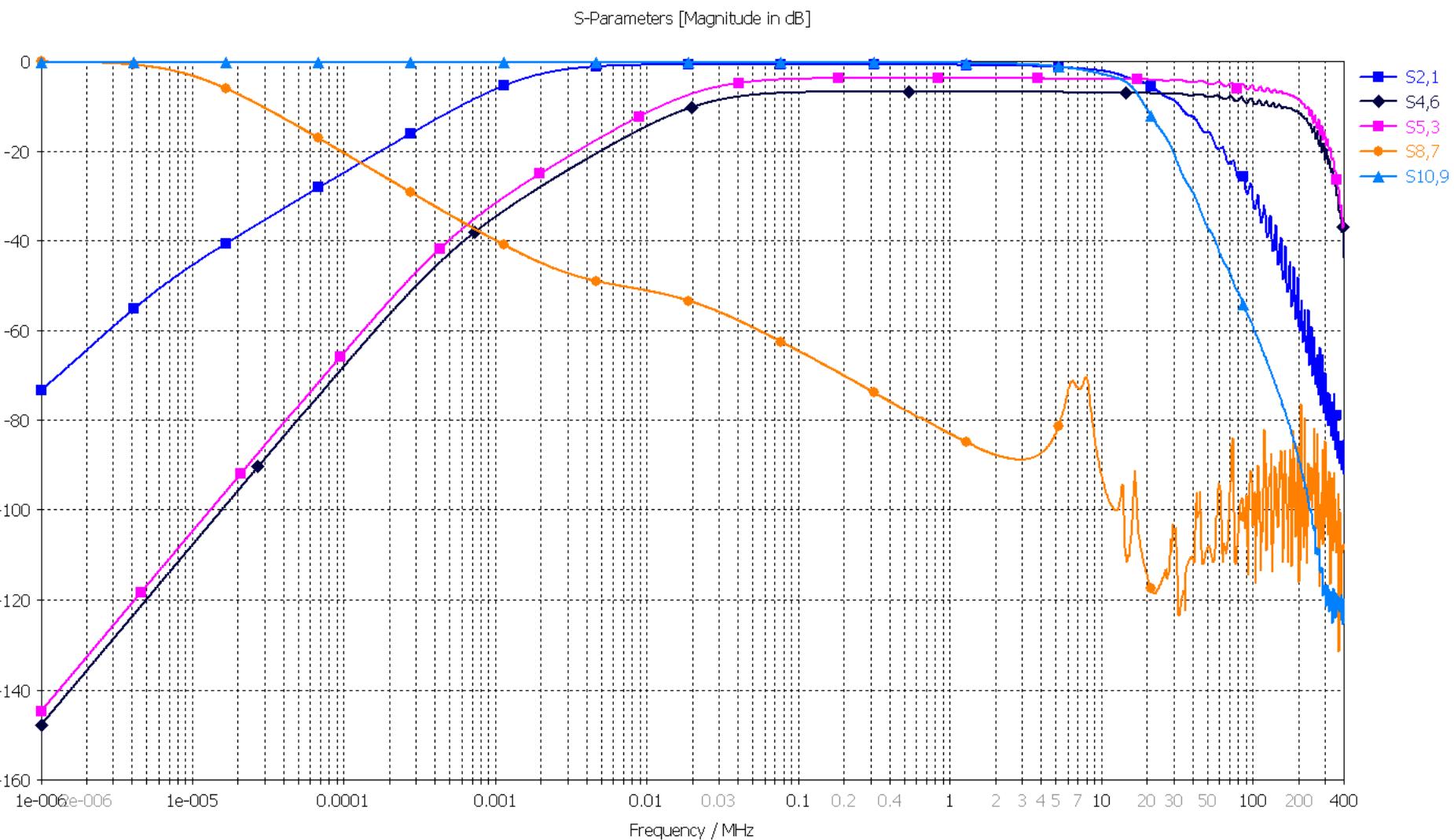


Figure 4.4.7-3. Cable Harness, Connectors, Interface Circuits and CS114 Environment Signal Transfer Simulation – Desired Path Transfer

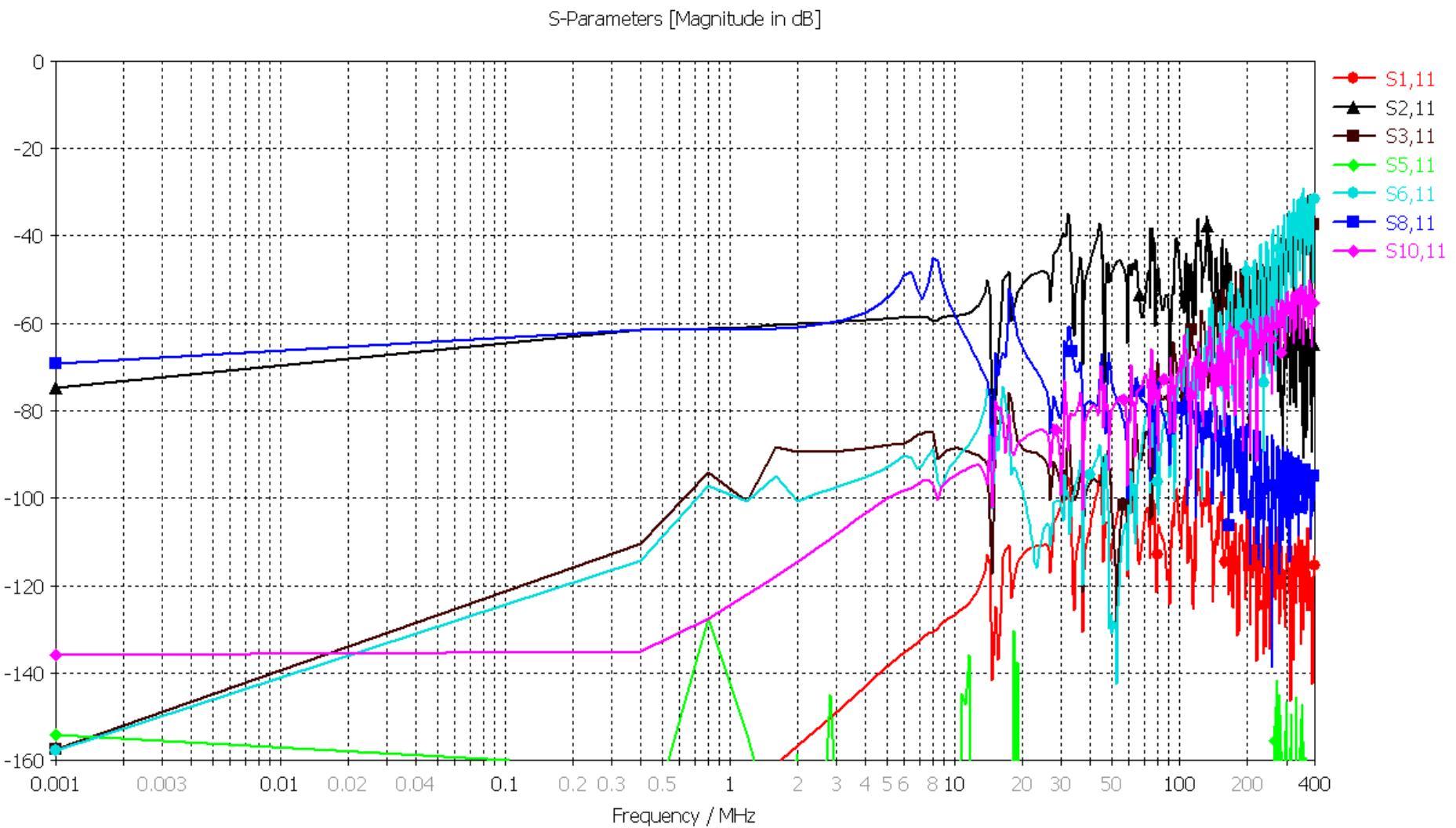


Figure 4.4.7-4. Cable Harness, Connectors, Interface Circuits and CS114 Environment Signal Transfer Simulation – Environment Transfer through Overshield

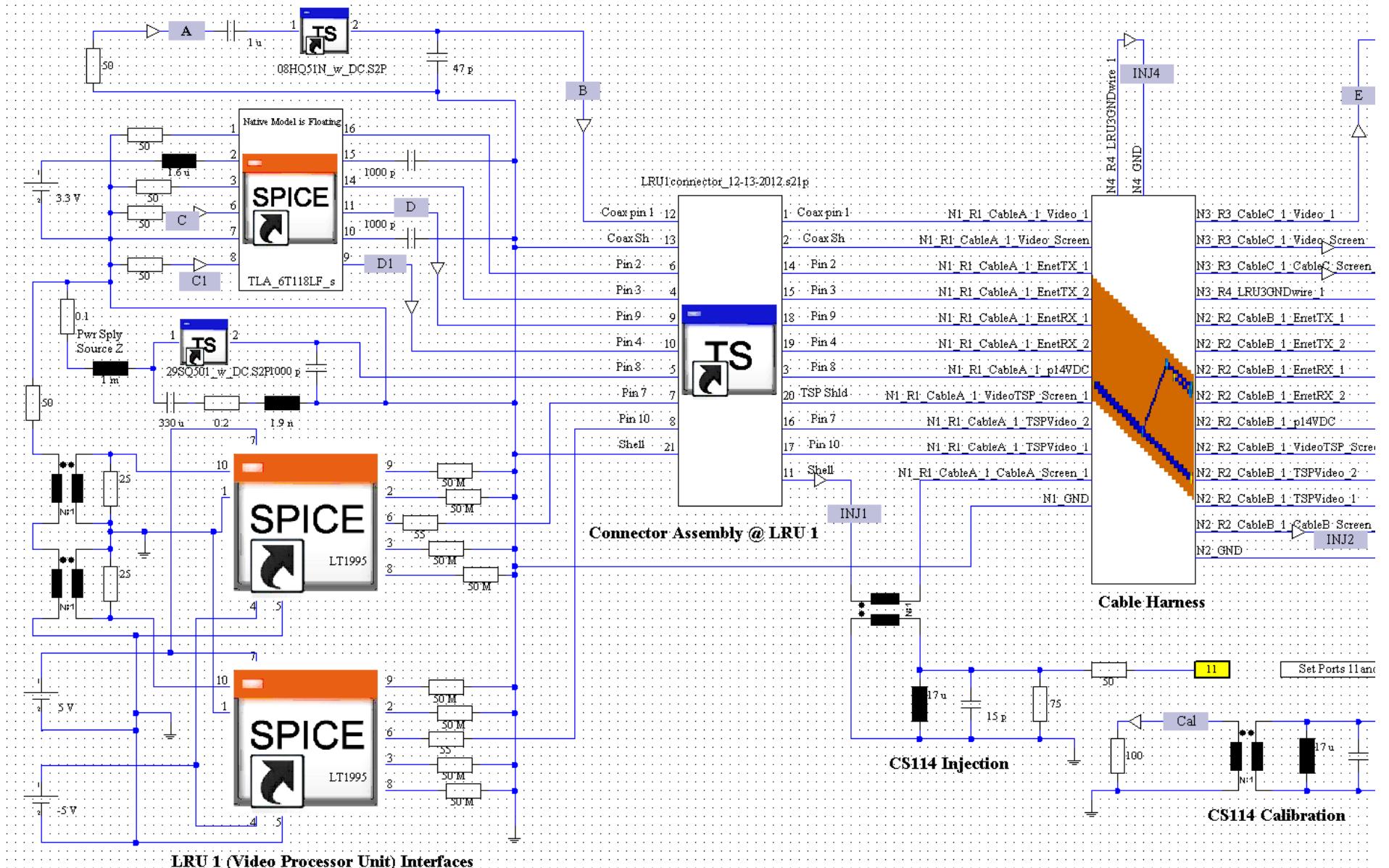


Figure 4.4.8-1. Complete System in Co-Simulation Framework for CS114 Simulation – left side schematic

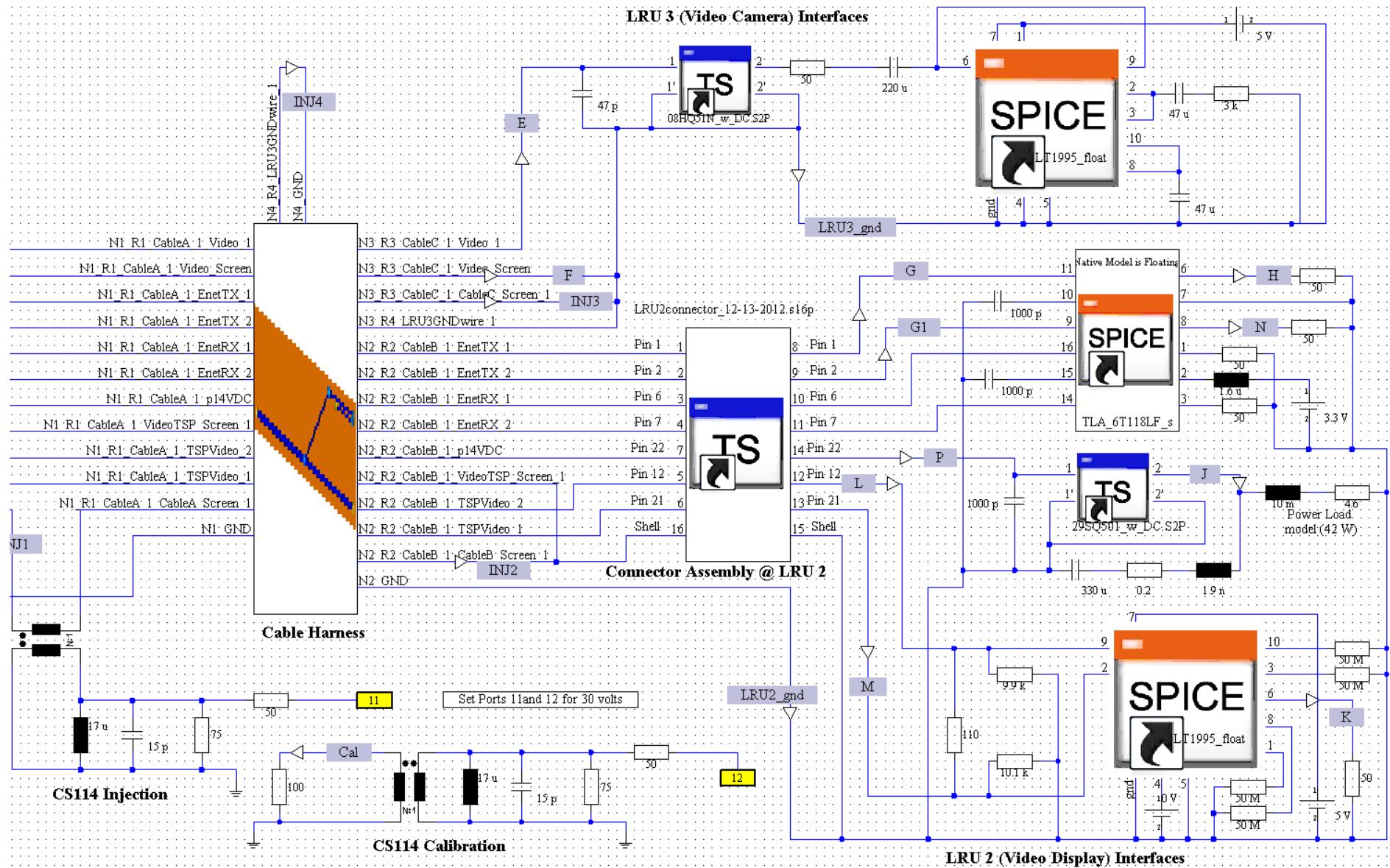


Figure 4.4.8-2. Complete System in Co-Simulation Framework for CS114 Simulation – right side schematic

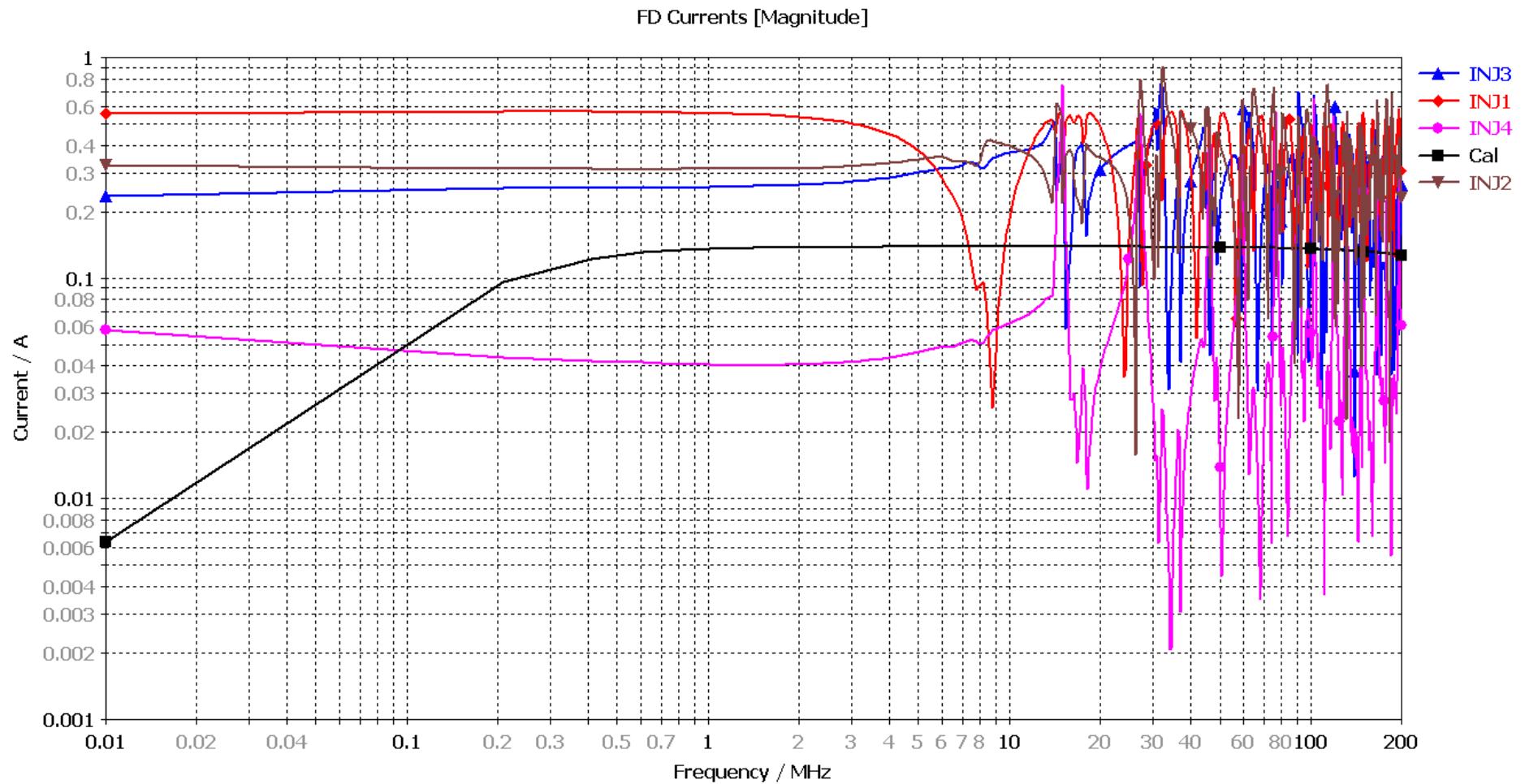


Figure 4.4.8-3. Complete CS114 Simulation – Results: Raw Calibration Current and Shield Currents

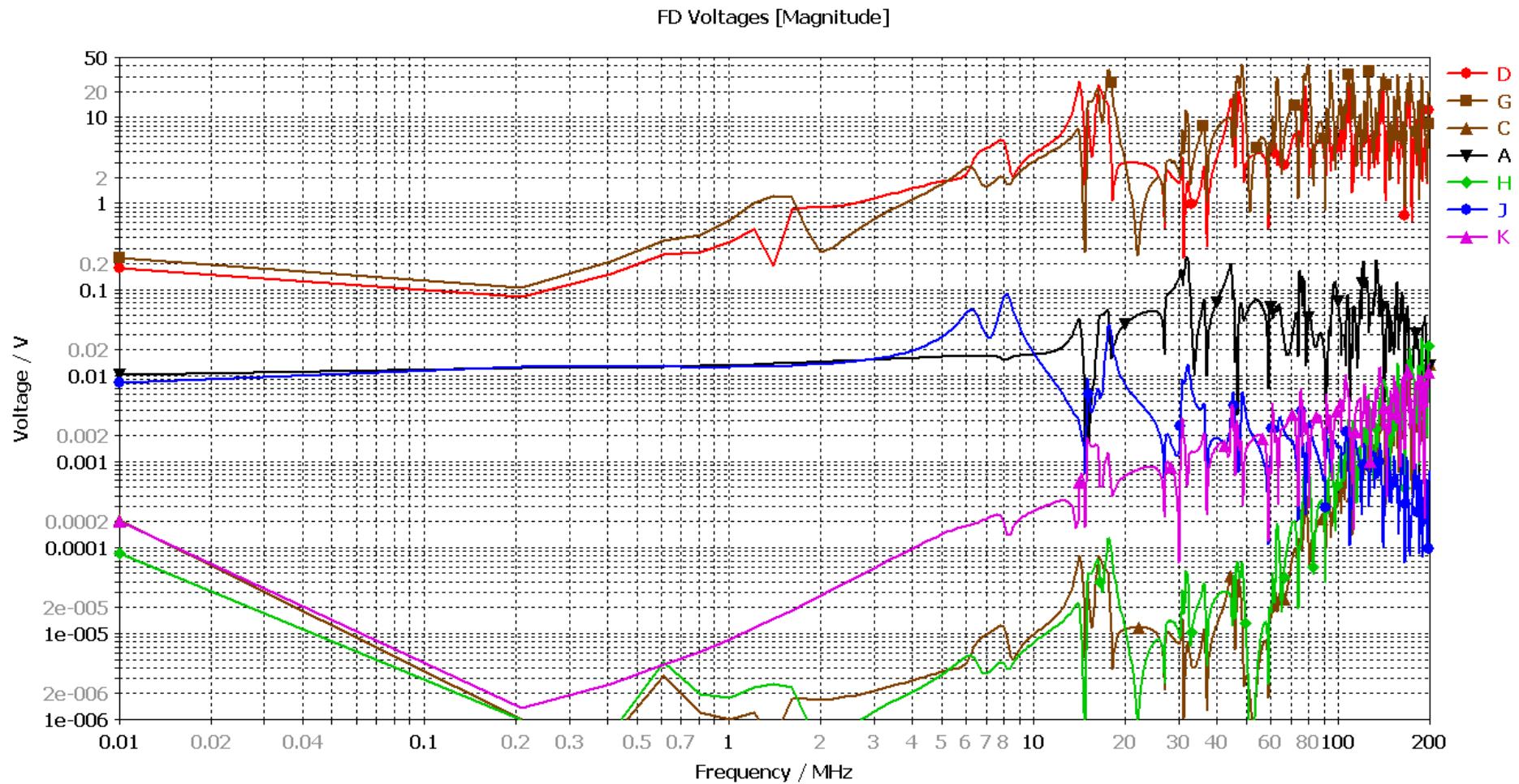


Figure 4.4.8-4. Complete CS114 Simulation – Results: Raw Interference Probe Voltages

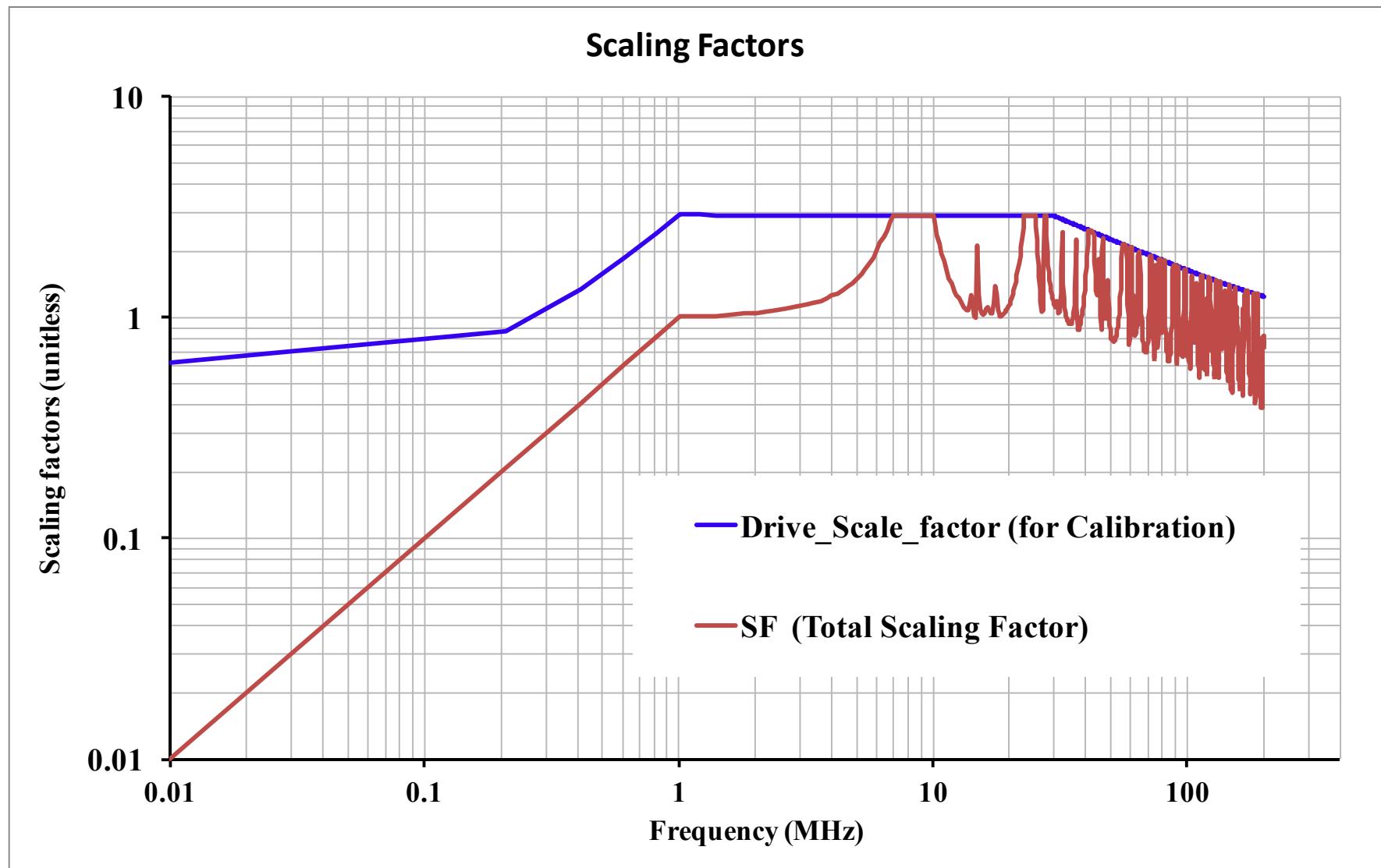


Figure 4.4.8-5. Complete CS114 Simulation – Results: Computed Scaling Factors Per MIL-STD-461F Methods

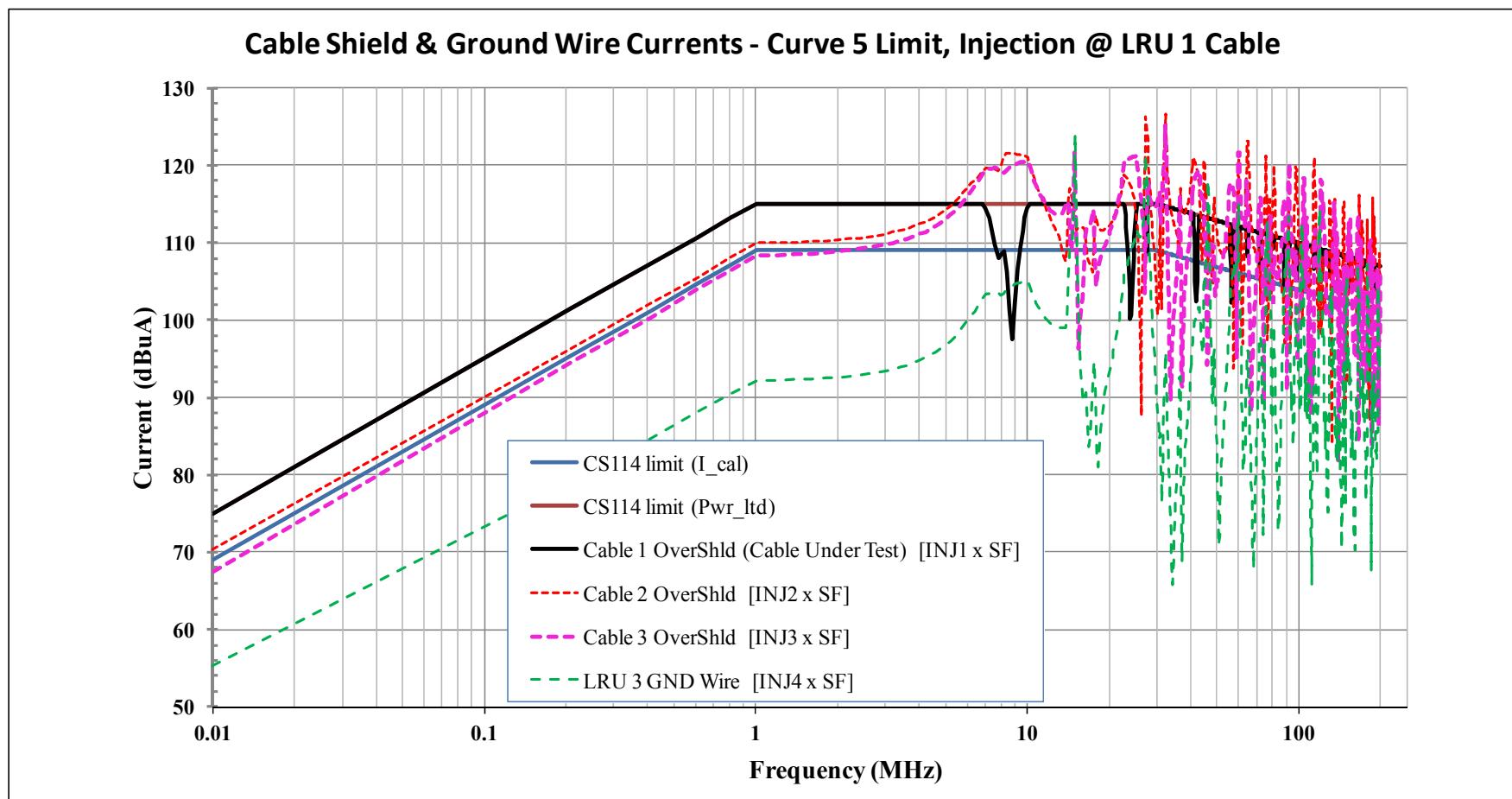


Figure 4.4.8-6. Complete CS114 Simulation – Results: Corrected Shield Currents

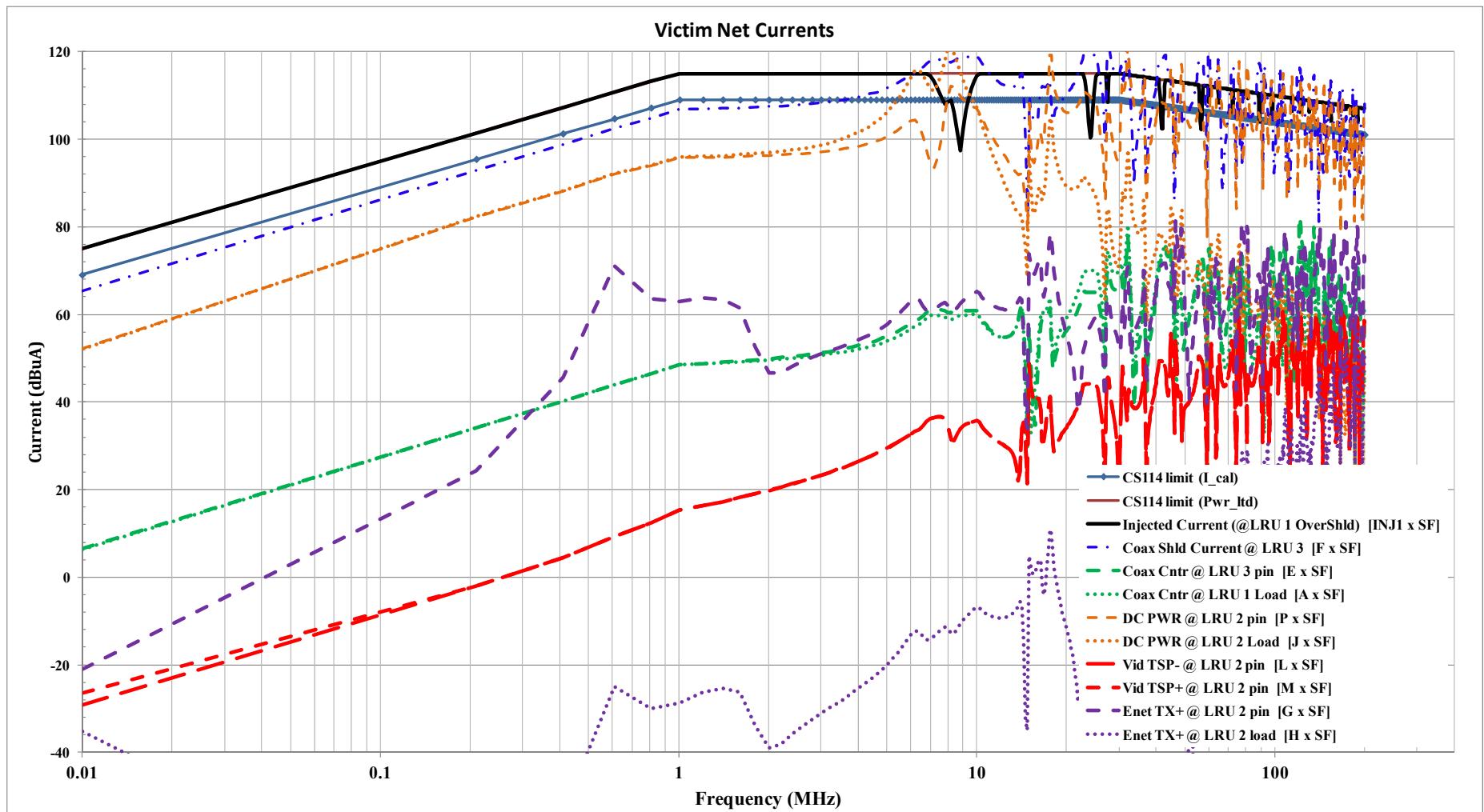


Figure 4.4.8-7. Complete CS114 Simulation – Results: Corrected Victim Net Currents

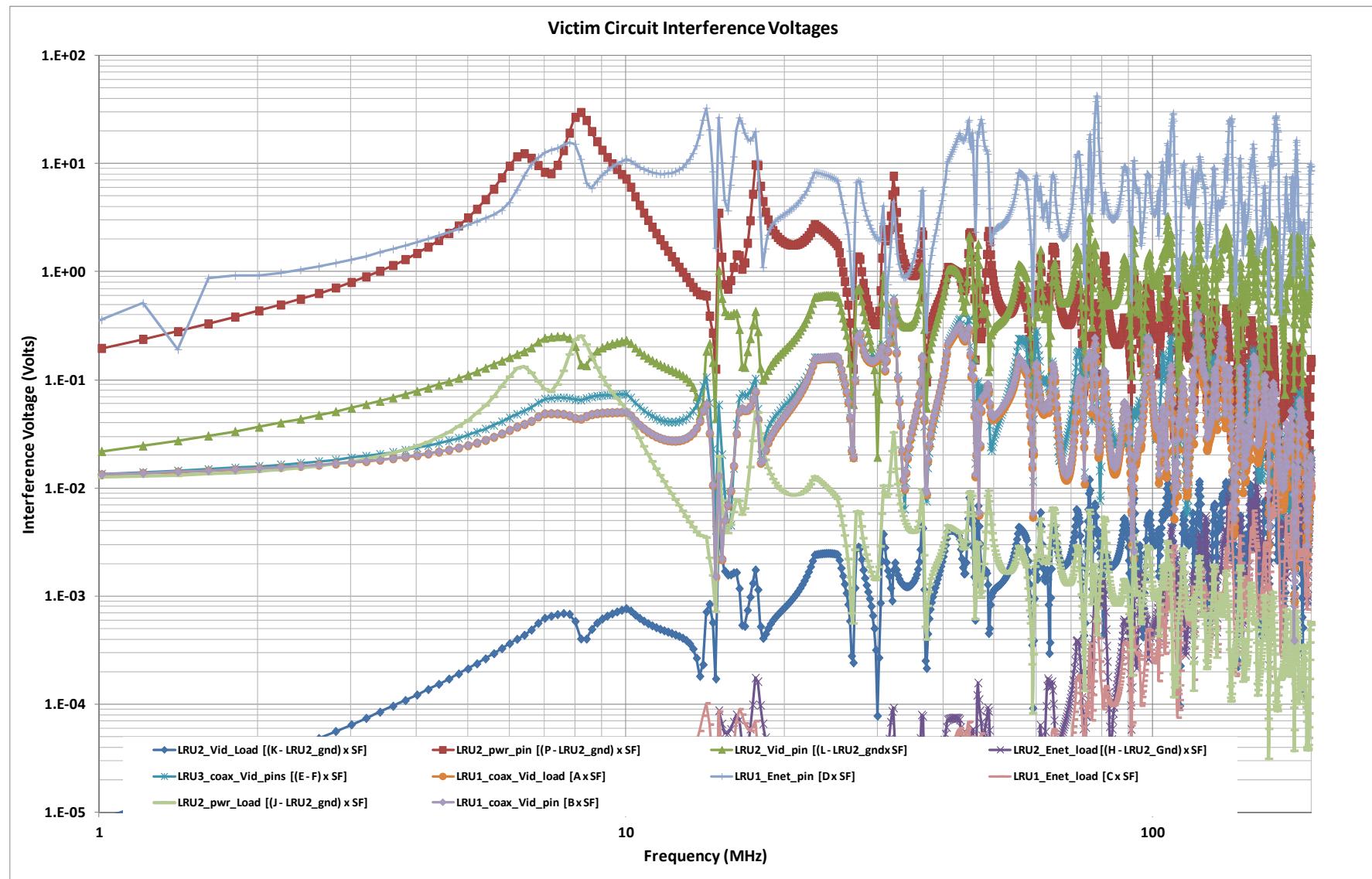
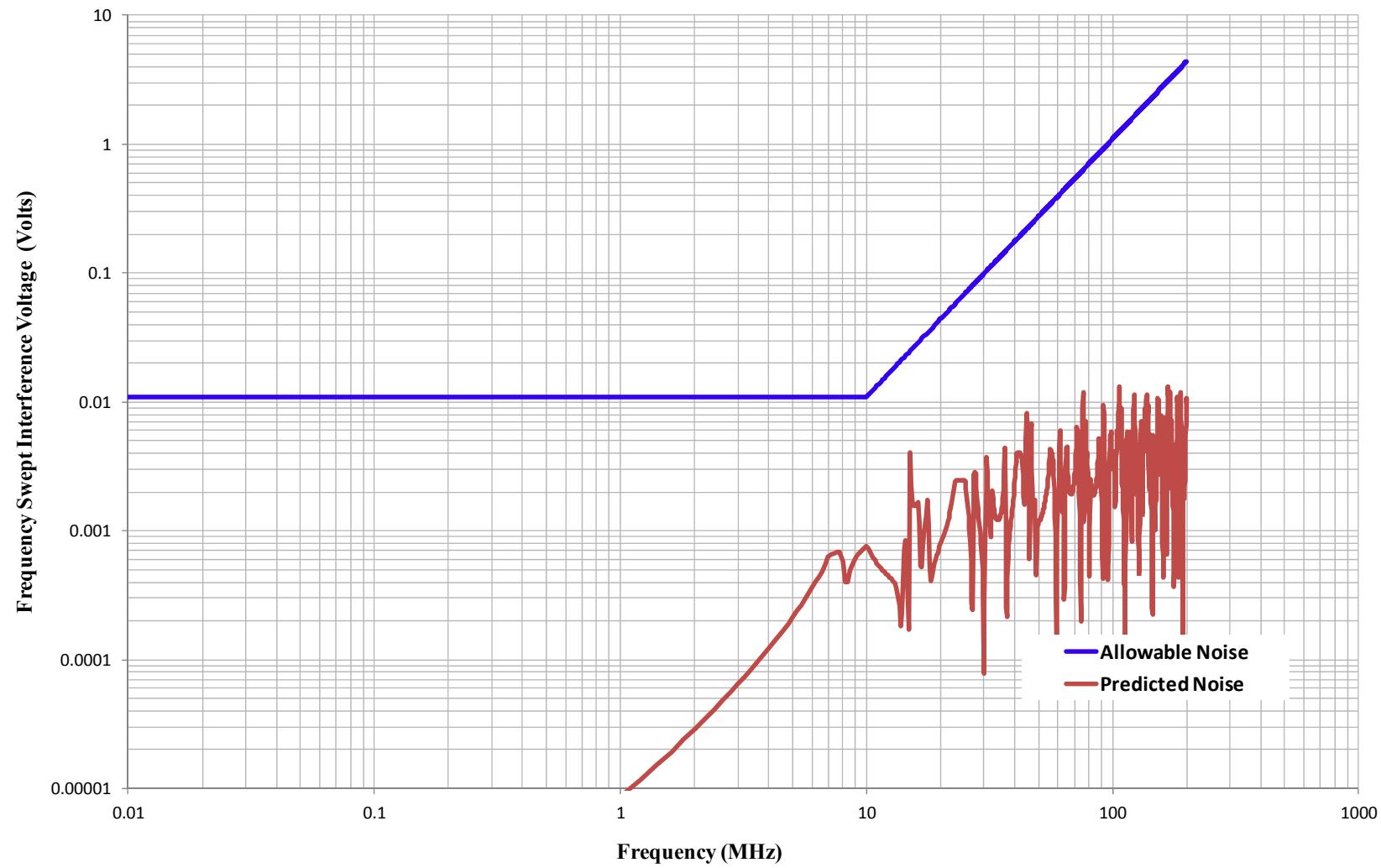


Figure 4.4.8-8. Complete CS114 Simulation – Results: Corrected Victim Circuit Interference Voltages

NTSC I-Frame Compressed Digital Video - Twisted Shielded Pair Serial Link - Interference Analysis**Figure 4.4.9-1. CS114 Simulation Results Analysis - Interference Prediction for Digital Video Circuit**

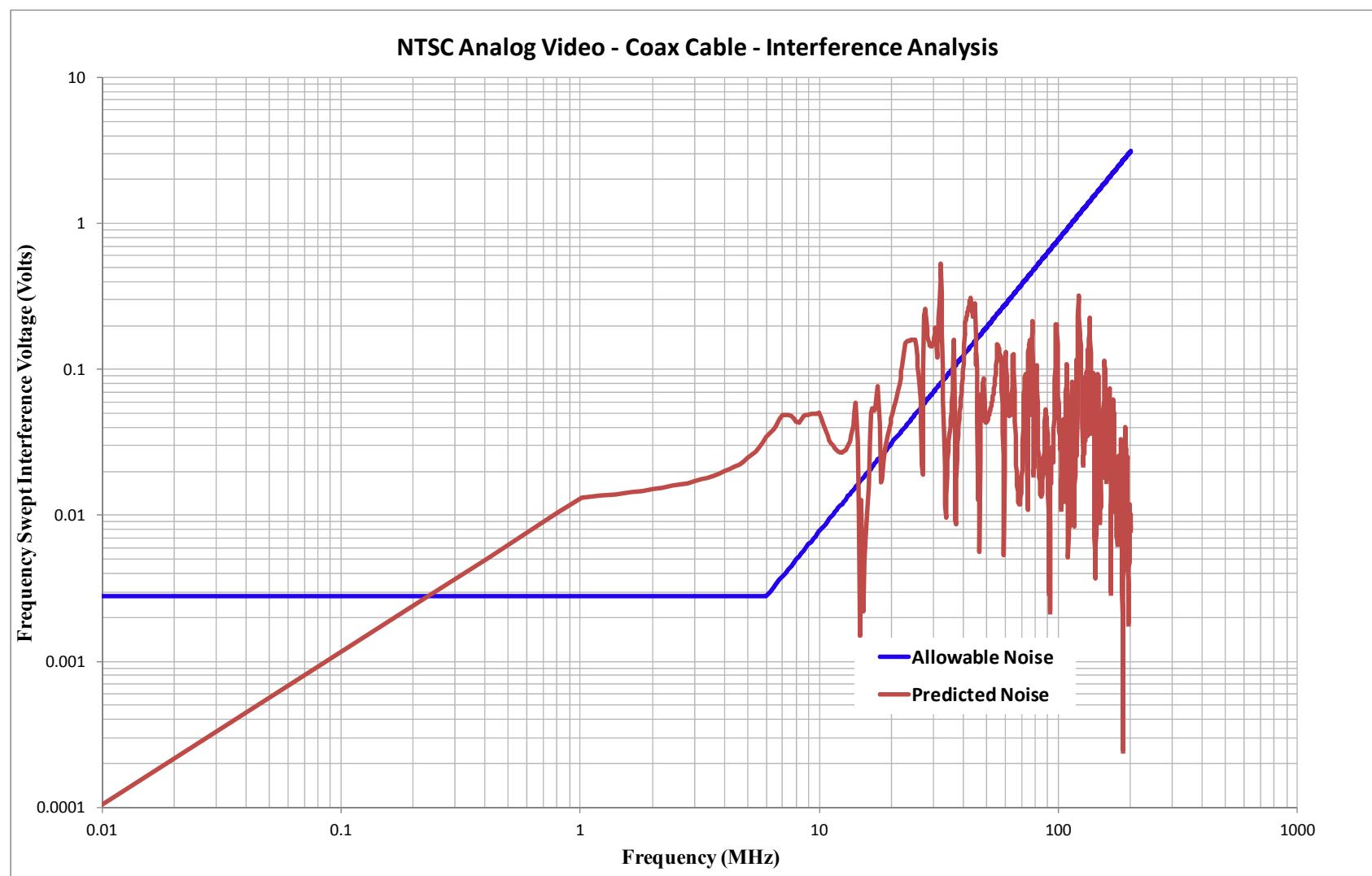


Figure 4.4.9-2. CS114 Simulation Results Analysis - Interference Prediction for Analog Video Circuit

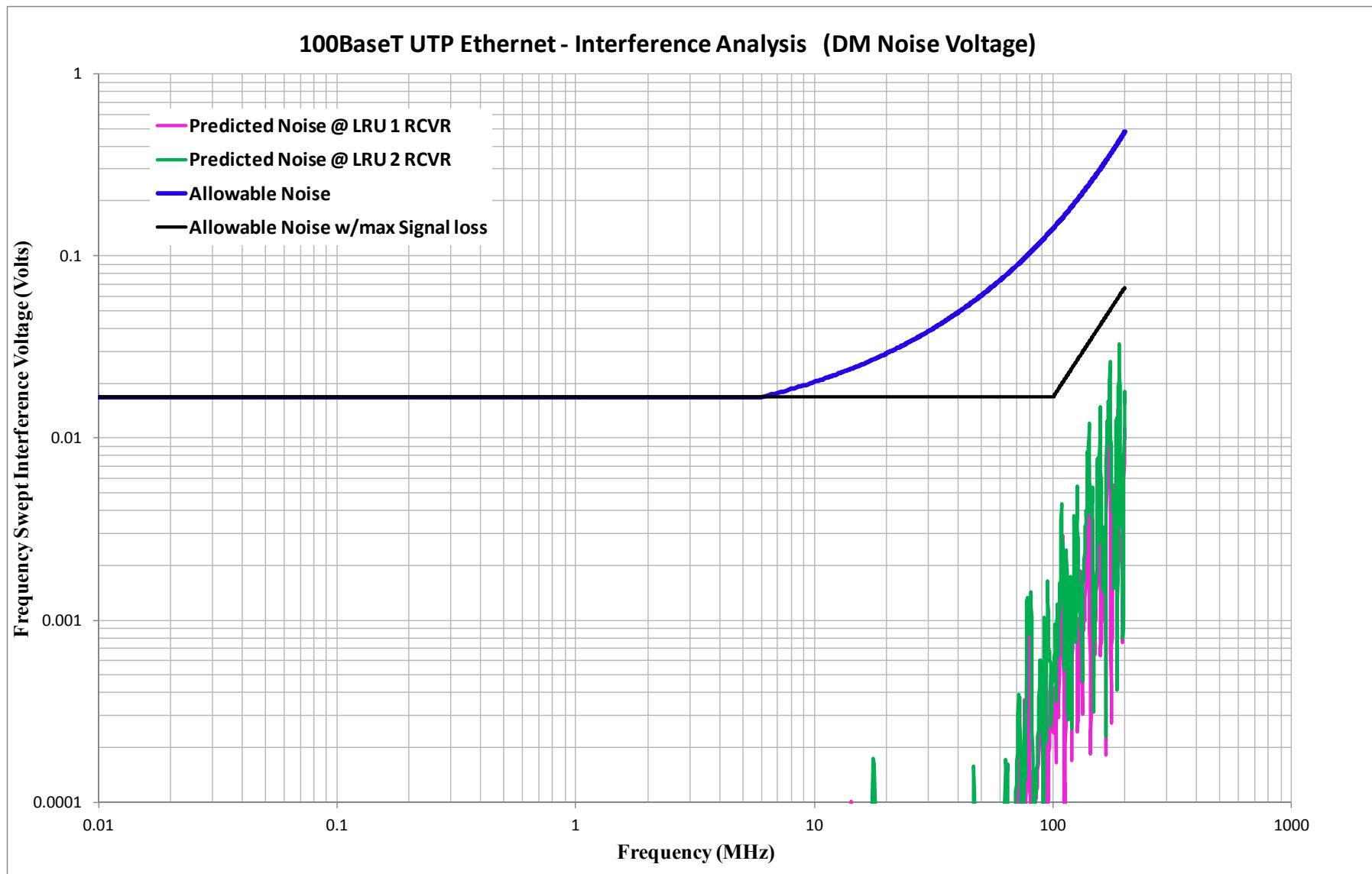


Figure 4.4.9-3. CS114 Simulation Results Analysis - Interference Prediction for Ethernet Circuits

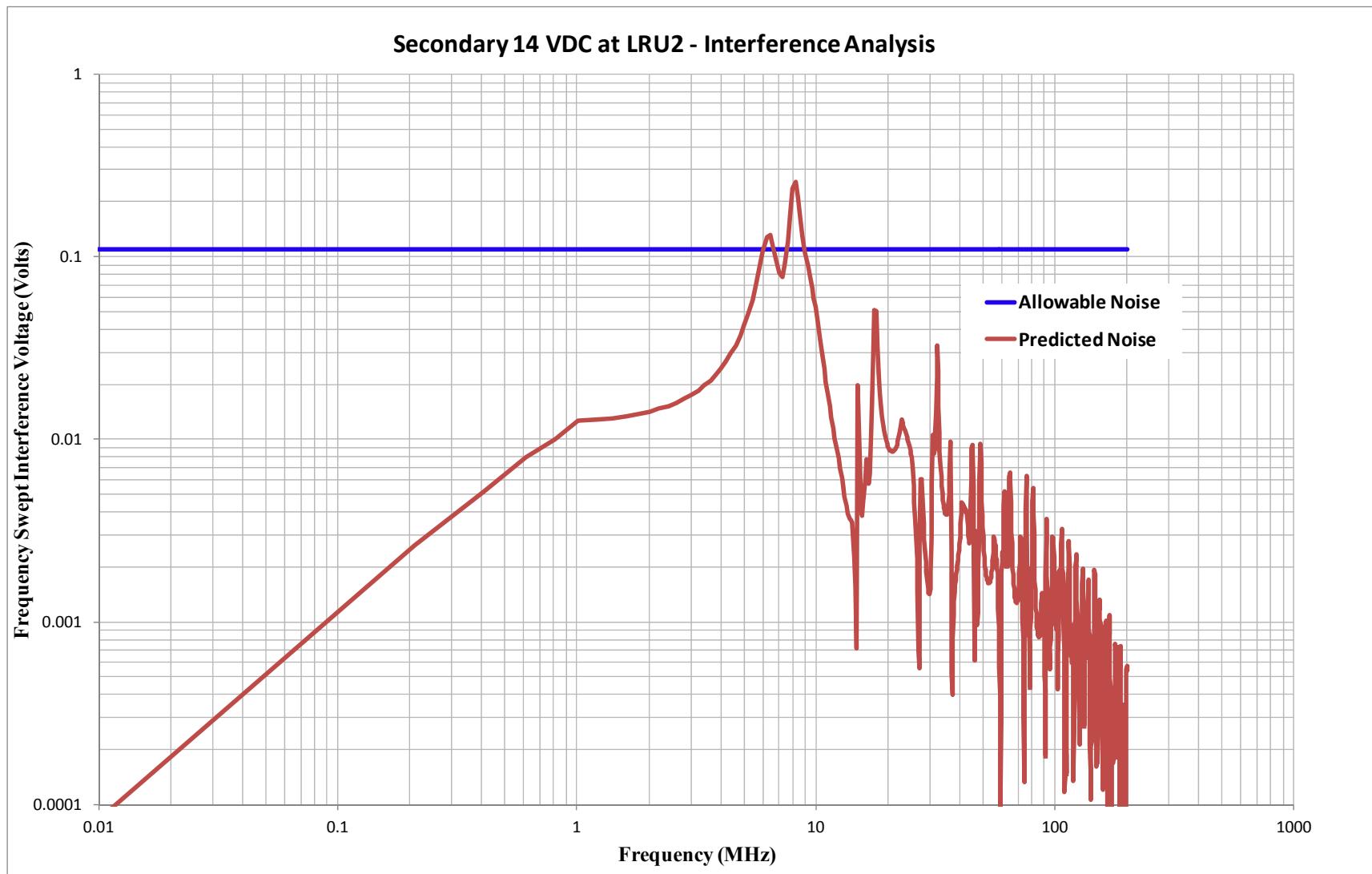


Figure 4.4.9-4. CS114 Simulation Results Analysis - Interference Prediction for Power Circuit

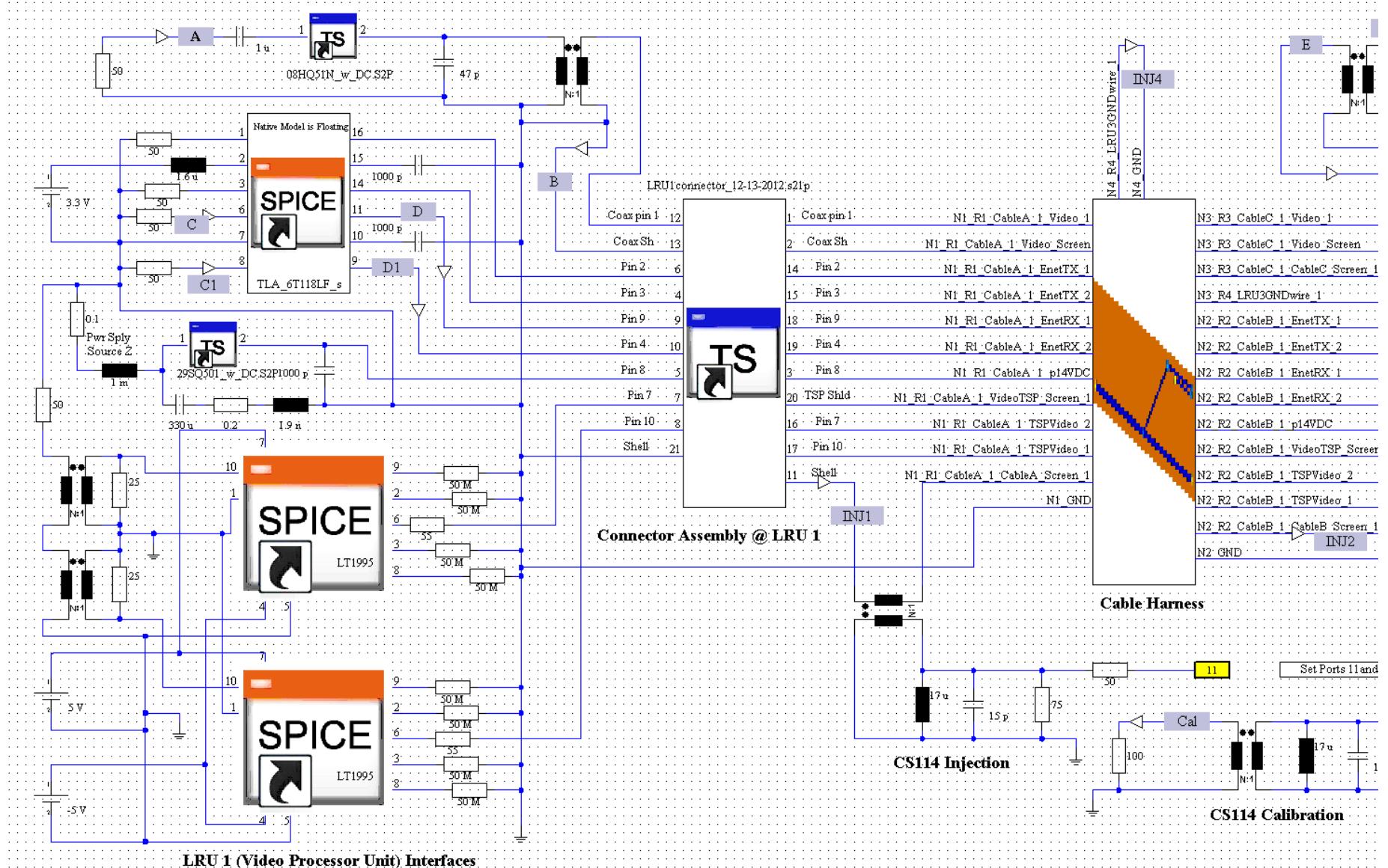


Figure 4.1.10-1. Complete System in Co-Simulation Framework for CS114 Simulation w/ EMC Improvements – left side schematic

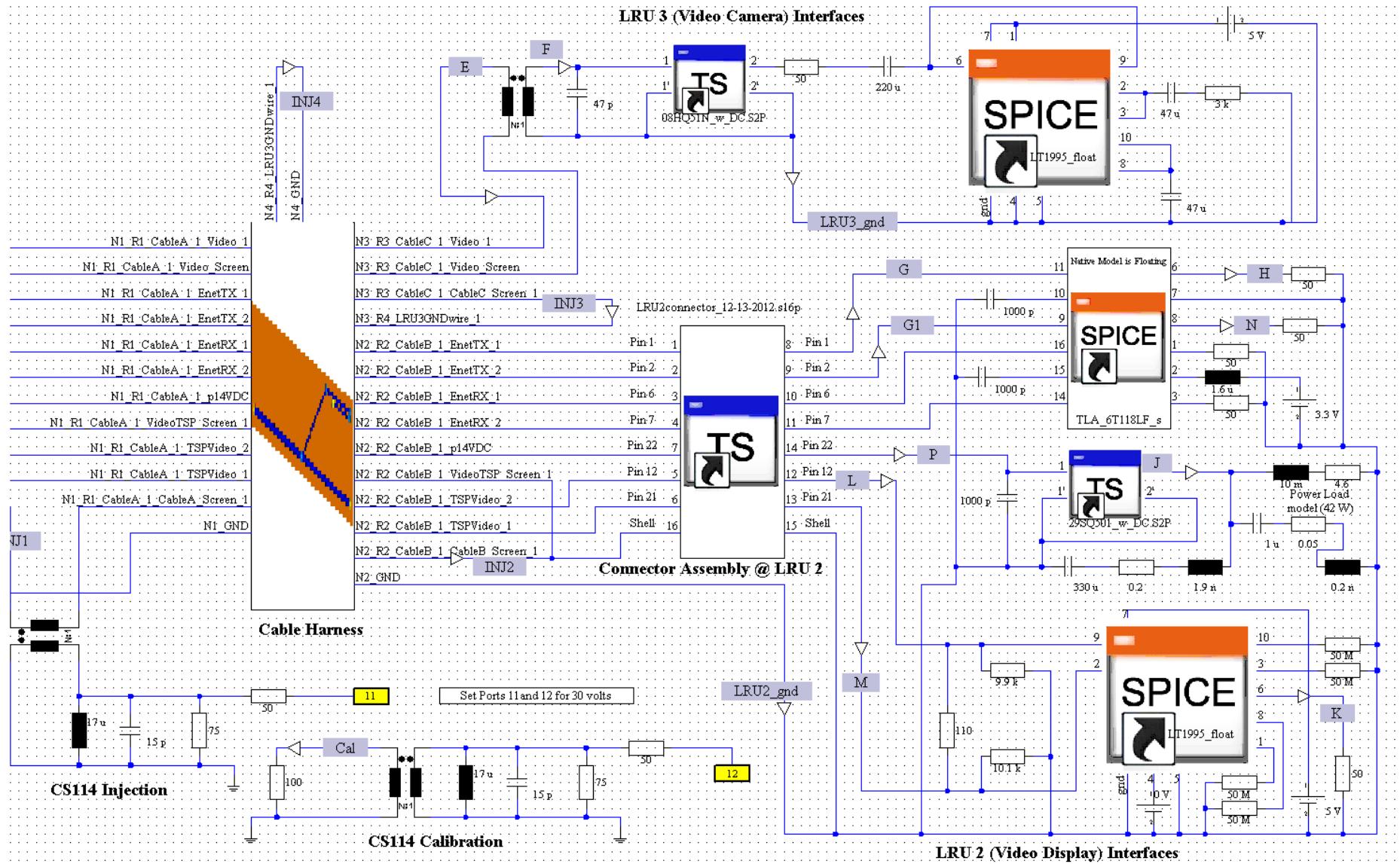


Figure 4.4.10-2. Complete System in Co-Simulation Framework for CS114 Simulation w/ EMC Improvements – right side schematic

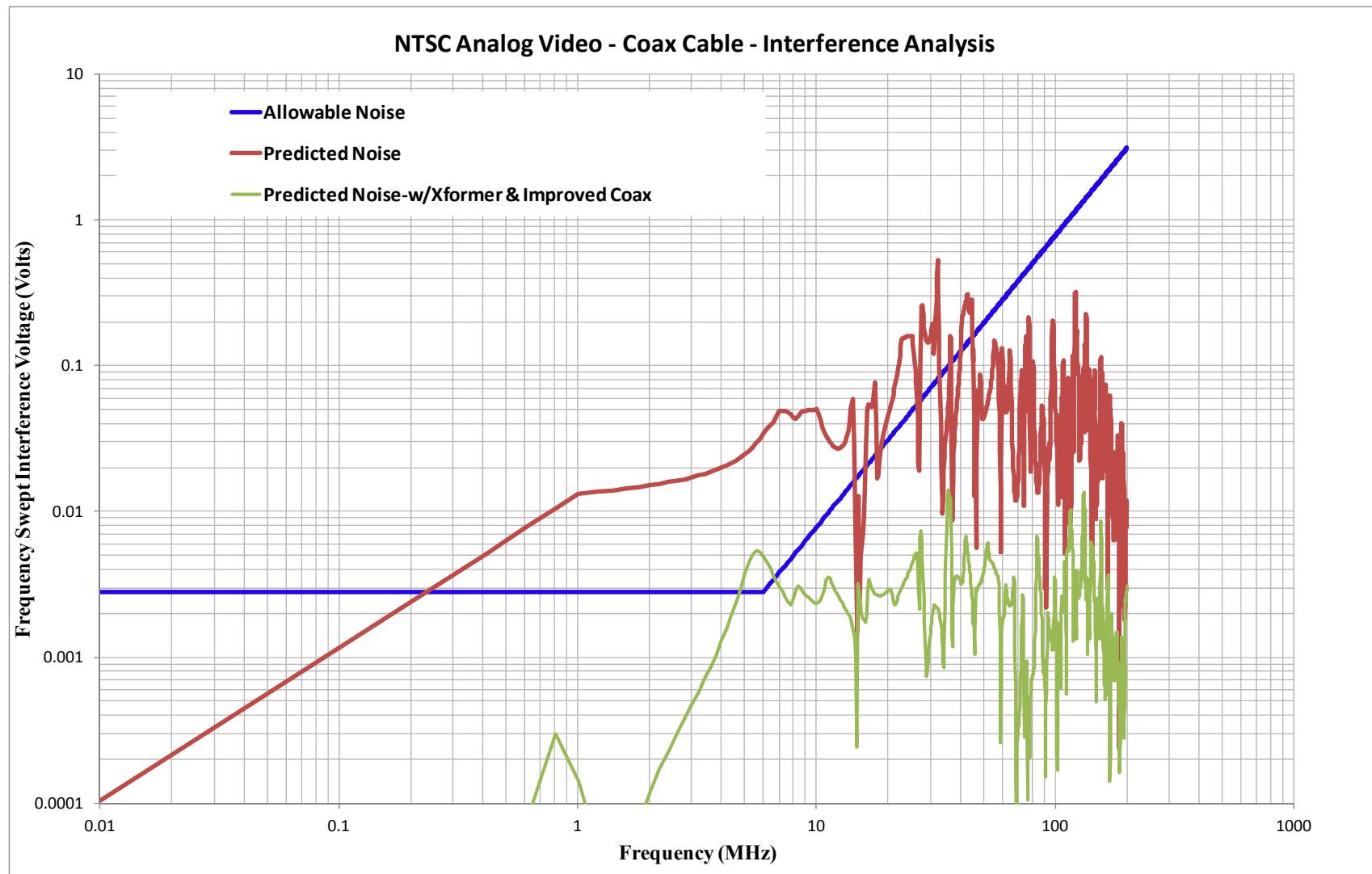


Figure 4.4.10-3. CS114 Simulation Results Analysis - Simulation w/ EMC Improvements – Analog Video Circuit

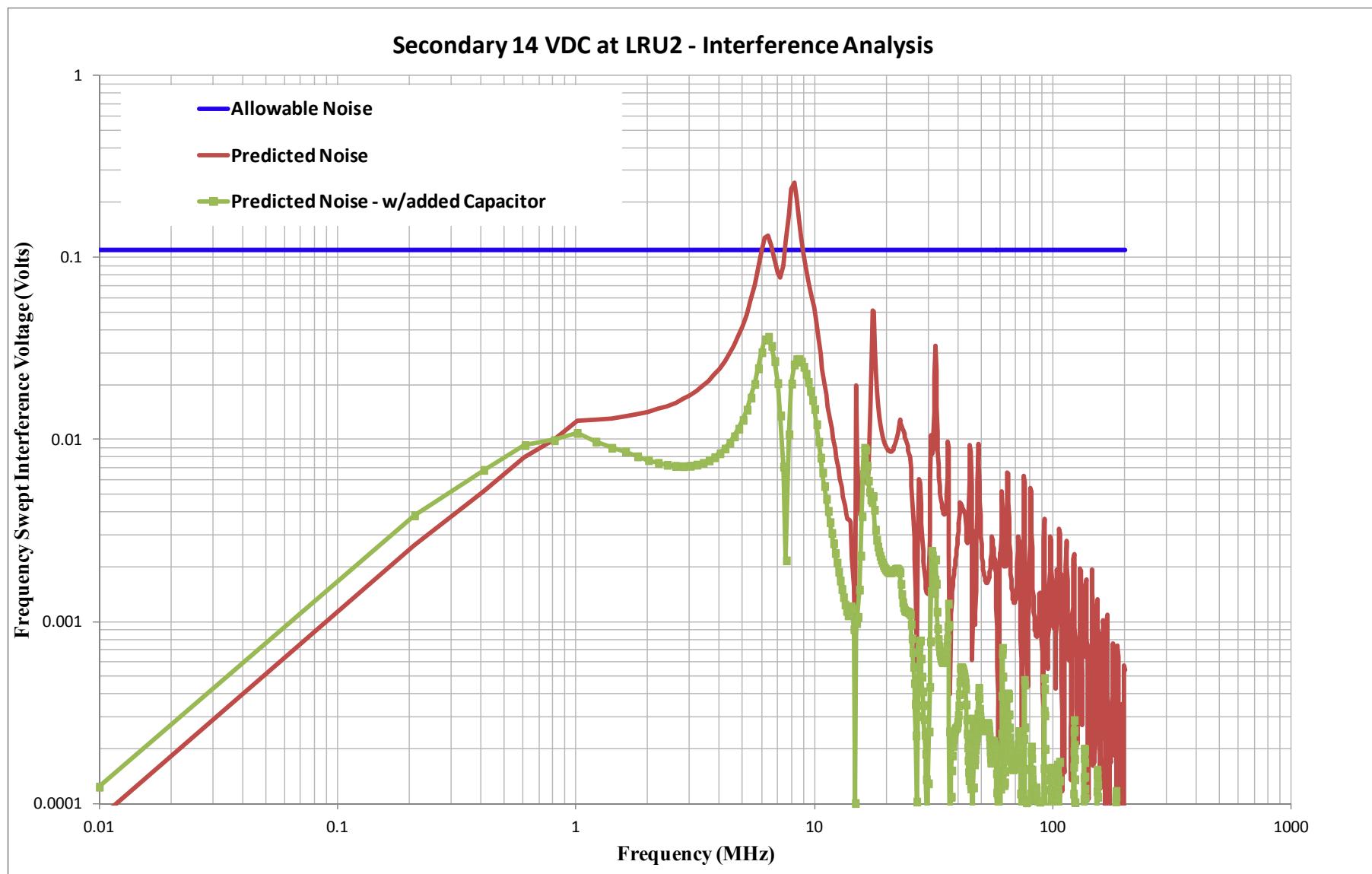


Figure 4.4.10-4. CS114 Simulation Results Analysis - Simulation w/ EMC Improvements – Power Circuit

APPENDIX A: STATEMENT OF WORK

Context Model Library – EMI Susceptibility

Electronic equipment in vehicles is subject to electromagnetic interference from sources internal and external to the vehicle via pick-up and subsequent conduction through the cables and wires attached to the equipment. This interference presents itself at the equipment connector pins as a voltage with an equivalent source impedance to result in potential upset of the interface and/or damage to electronic components if adequate protection is part of the equipment and/or cable-connector system design. The interference voltage may be a transient (typically described in the time domain), or it may be a somewhat steady-state condition at a particular frequency. The typical electromagnetic interference environments that are presented to cable-connector systems for equipment in military vehicles are defined in standards such as MIL-STD-461Fⁱ.

This project involves the development of a set of modeling methods to allow prediction of the interference at equipment connector pins resulting from exposure of the vehicle's cable/connector system to radiated and conducted (induced) interference. Inputs to the model will be cable and connector cross-section design details, transfer impedance of the cable and connector shielding, and interface circuit design details (i.e. schematic and component characteristics or an impedance versus frequency plot). Stimuli to the model include the CS114, CS115, CS116, RS103 and RS105 environments as applied per MIL-STD-461F. Outputs from the model would include the time domain and/or frequency domain interference presented at the connector pins in terms of a voltage source and equivalent impedance for each of the input MIL-STD-461F input environments. The resulting interference levels can then be compared to the levels of the desired signals and component damage threshold levels to determine if interface failure (upset) and/or component damage is expected. Of course if unacceptable upset and/or damage are predicted, changes to the cable-connector systems design and/or the protection at the equipment interface circuits can be made to improve the probability of electromagnetic compatibility.

This effort will also include presentation of methods to determine the transfer impedance of cable shields and connectors.

The final result of this effort will be a technical report containing model topologies, methods and guidance for quantifying expected equipment immunity to the MIL-STD-461F environment with respect to interference coupling via the cable/connector system. Model template and parameters will be provided for common interfaces used on defense ground vehicles.

ⁱ MIL-STD-461F – Department of Defense Interface Standard: Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment, 10 Dec 2007.

Synthesis of a PSD Compatible Acceleration Time-History

J. Edward Alexander
Technical Fellow
BAE Systems, Land and Armaments
4800 East River Road
Minneapolis, MN 55421
ed.alexander@baesystems.com

Abstract

System survival to a random vibration environment is frequently defined in terms of the acceptable response to a prescribed Power Spectral Density (*PSD*). Commercial electro-dynamic shakers have control systems that will subject a test article to an input acceleration that is compatible with a required *PSD*. In some cases, it is desirable to have an acceleration time-history that is compatible with a prescribed *PSD* for pre-test and/or post-test random vibration analysis of the system. Study of the temporal response of a nonlinear system is one example where a spectrum compatible acceleration time-history could be beneficial.

An approach is presented herein to synthesize a base acceleration time-history that is compatible with a prescribed *PSD*. The synthesized acceleration is developed using the summation of sinusoids with frequencies that cover the frequency range of the *PSD*. Each sinusoid has a unique frequency, unique amplitude and random phase angle. The amplitude of each sinusoid is sized by equating the maximum power of the sinusoid with the area under the *PSD* curve corresponding to the frequency of the sinusoid. An envelope function $E(t)$ is used to control the overall shape and duration of the synthesized acceleration. The peak amplitude of the envelope is sized to match the total power of the prescribed *PSD* to that of the *PSD* that results from the synthesized base acceleration. This is done efficiently by Newton Raphson iteration. Three examples are presented to illustrate the process.

I. Introduction

Shock and vibration engineers are typically familiar with the process of converting acceleration in the time domain to spectral representations such as shock response spectra (*SRS*) and power spectral densities (*PSD*). However, occasionally the inverse

process is needed; namely synthesizing an acceleration time-history from a spectral representation. Manufacturers of electro-dynamic test machines create waveforms from prescribed spectral representations to drive a shaker head with an acceleration time-histories compatible with specified *SRS* or *PSD*. These algorithms are proprietary to the manufacturers of these systems. A number of papers have been published, largely by seismic engineers (Alexander, Gasparini, Levy, Lin, Preumont, Rizzo, Salginik, Scanlan) with methods to synthesize an acceleration time-history to be compatible with a prescribed *SRS*. However, the process of synthesizing an acceleration time-history that is compatible with a prescribed *PSD* is not as well published. Seismic engineers (Hou, Housner, Jennings, Kanai, Tajimi) have addressed the synthesis of wave-forms for seismic simulations based on *PSD* spectral representations, motivated by the random nature, relative long durations and the multi-frequency content of a single earthquake. However, there is a paucity of research for *PSD* spectra inverse transform to transient wave forms in the mechanical shock and vibration community. This may be due to the differences of the physical phenomena of mechanical shock and random vibrations. The shock response spectrum, as the name implies, is the spectral representation of a relatively short, high intensity transient shock event with a defined rise, peak and decay. The power spectral density on the other hand, at least for mechanical systems, is used largely to characterize random vibrations over relatively long durations.

The author, recently faced with a task to synthesize an acceleration wave form to meet a prescribed *PSD*, developed a simple procedure described herein. The procedure synthesizes an acceleration time-history, the *PSD* of which has the same total power as that of the prescribed *PSD*. The overall shape and duration of the synthesized wave form is controlled by an envelope function that can be tailored to be consistent with the temporal characteristics of the physical event being considered.

II. Approach

The objective of this study was to synthesize a base acceleration to match a prescribed Power Spectral Density (*PSD*). There are a number of ways that this could be achieved. The one examined herein is the summation of sinusoids of different frequencies with amplitudes A_n and random phase angles ϕ_n for each frequency n . The overall shape and duration of the synthesized wave form is controlled by an

envelope function $E(t)$. Judicious choices for the amplitudes A_n will tailor the synthesized base acceleration to yield a PSD that matches a target PSD within a certain error. The form of the synthesized base acceleration (Alexander) is given by,

$$a_b(t) = E(t) \sum_{n=1}^N A_n \sin(\omega_n t + \phi_n) \quad (1)$$

To determine the amplitude coefficients A_n , the total power of the PSD is matched to the power obtained from the summations of the sinusoids (Gasparini). The total power of a signal $a(t)$ of duration T is defined as,

$$Power = \frac{1}{T} \int_0^T a^2(t) dt$$

$$Power = \frac{A_{ave}^2 T}{T} = A_{ave}^2$$

For a sine wave of peak amplitude A_n , the average amplitude is given by $A_{ave} = \sqrt{1/2} * A_n$ resulting in,

$$Power_n = (0.707 A_n)^2 = \frac{1}{2} A_n^2$$

The total power of the synthesized base acceleration can then be represented as the summation of the power contributions of each sinusoid from (1).

$$Power = \sum_{n=1}^N \left(\frac{A_n}{2} \right)^2 \quad (2)$$

The total power of a PSD also corresponds to the area of the PSD . Figure 1 shows a graph of a notional PSD . For equal discrete frequency intervals Δf the area under the curve, and hence power, over all frequencies n is approximated as,

$$Power = \sum_{n=1}^N PSD_n \Delta f \quad (3)$$

For each frequency n , equating the power represented by the area ($PSD_n \Delta f$) with the power corresponding to the amplitude of the n^{th} sinusoid of a synthesized base

acceleration (2) results in a relationship (4) between the amplitude A_n and the PSD_n for the n^{th} frequency. This relationship is used to determine A_n at each frequency from the target PSD to yield an initial synthesized base acceleration.

$$\frac{A_n^2}{2} = PSD_n \Delta f$$

$$A_n = \sqrt{2PSD_n \Delta f} \quad (4)$$

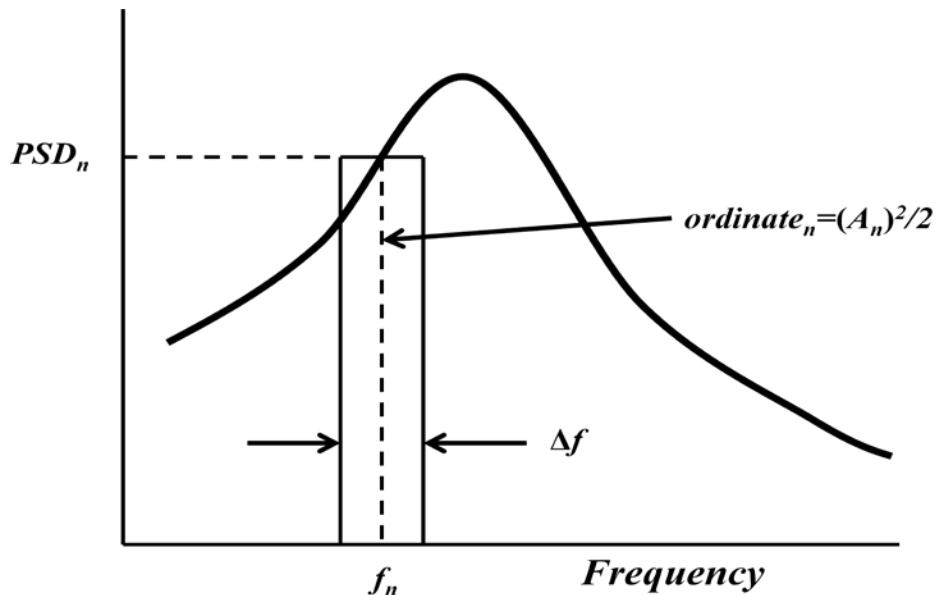


Figure 1 – Relationship of Power Spectral Density Function to Sinusoid Amplitude

Synthesis Process

An initial synthesized acceleration time-history (1) can be readily determined from the A_n coefficients from (4) with the amplitude of the envelope function, A_e shown in Figure 2, set to unity. A PSD of the initial synthesized acceleration-time history can then be determined and compared with the target PSD . However, to get a match of the power calculated from (3) from the target PSD to that of the synthesized acceleration-time history, iteration is required to fine tune A_e . An efficient way to do this is to use Newton Raphson iteration. If it is assumed that the error in the power difference between the

target PSD and the PSD from the synthesized base acceleration can be characterized as a function of A_e , then a Newton Raphson iteration process can be defined by the relationships below. Figure 3 illustrates graphical how the iteration process works for improving successive choices of A_e for each iteration i .

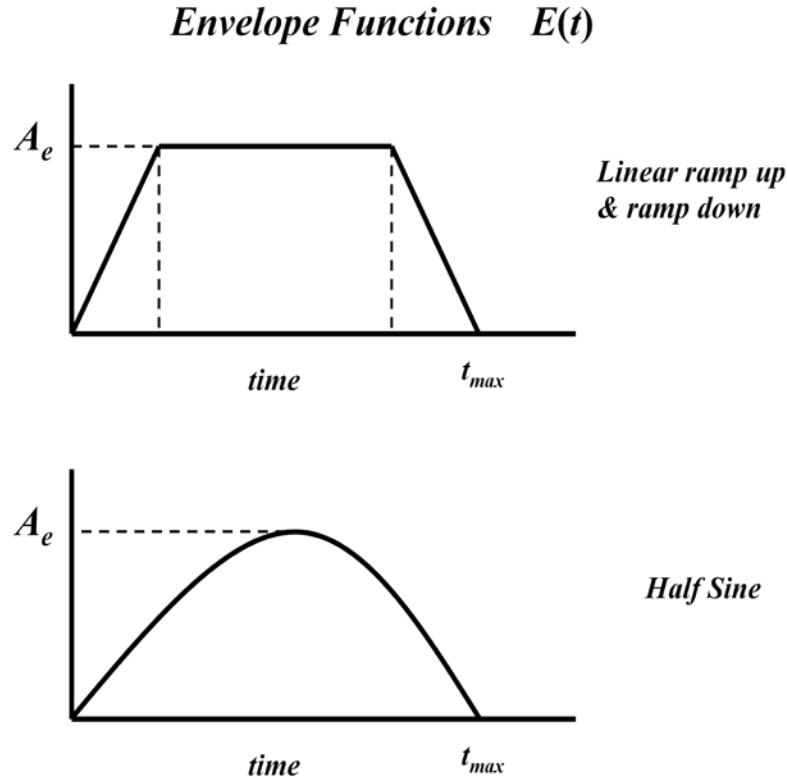


Figure 2 – Typical Envelope Functions $E(t)$

Newton-Raphson Iteration Procedure:

$$\% \text{ error power } (PSD_{synthesized} : PSD_{target}) \equiv f(A_e)$$

$$(A_e)_{i+1} = (A_e)_i - \frac{f[(A_e)_i]}{\frac{df}{dA_e}[(A_e)_i]}$$

where

$$\frac{df}{dA_e}[(A_e)_i] \approx \frac{(\% \text{ error})_i - (\% \text{ error})_{i-1}}{(A_e)_i - (A_e)_{i-1}}$$

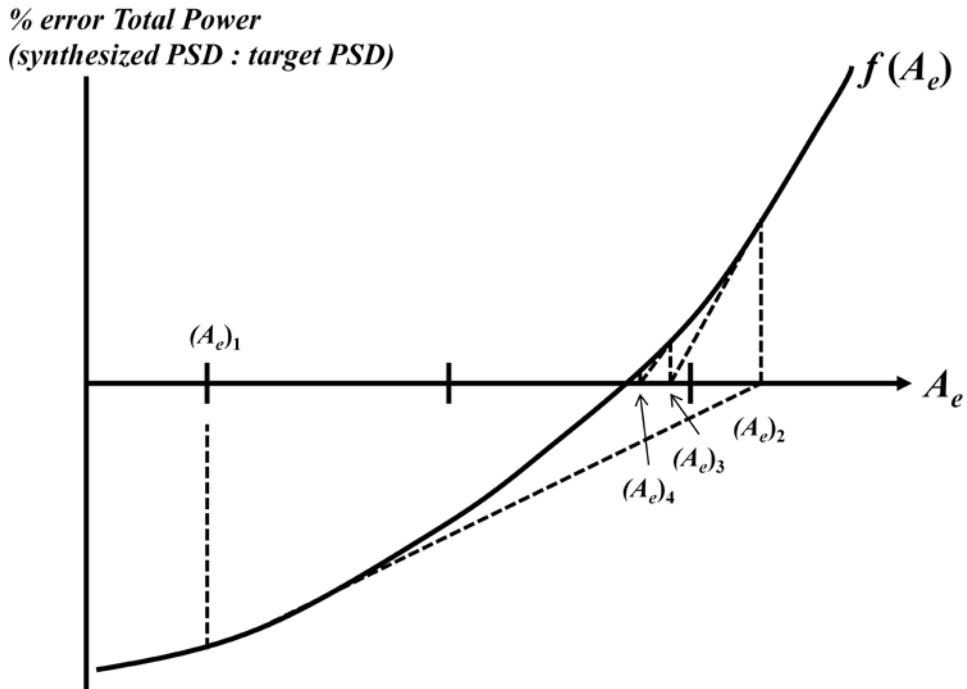


Figure 3: Newton Raphson Iteration to Determine A_e

III. Results

Example #1

A Matlab function was written to synthesize an initial base acceleration from (1) given a prescribed *PSD* using coefficients determined from (4). The process was checked using a known acceleration time-history data record from which a target *PSD* was calculated with Matlab's *PWELCH* function. Two envelope functions $E(t)$ shown on Figure 2, were developed for the synthesis process. Others can be readily developed depending on the specifics of the physical phenomena being simulated. As an initial guess, the $E(t)$ function is set with $A_e=1.0$. The acceleration-time history data used to test the Matlab function and the corresponding *PSD* are shown on Figures 4 and 5, respectively.

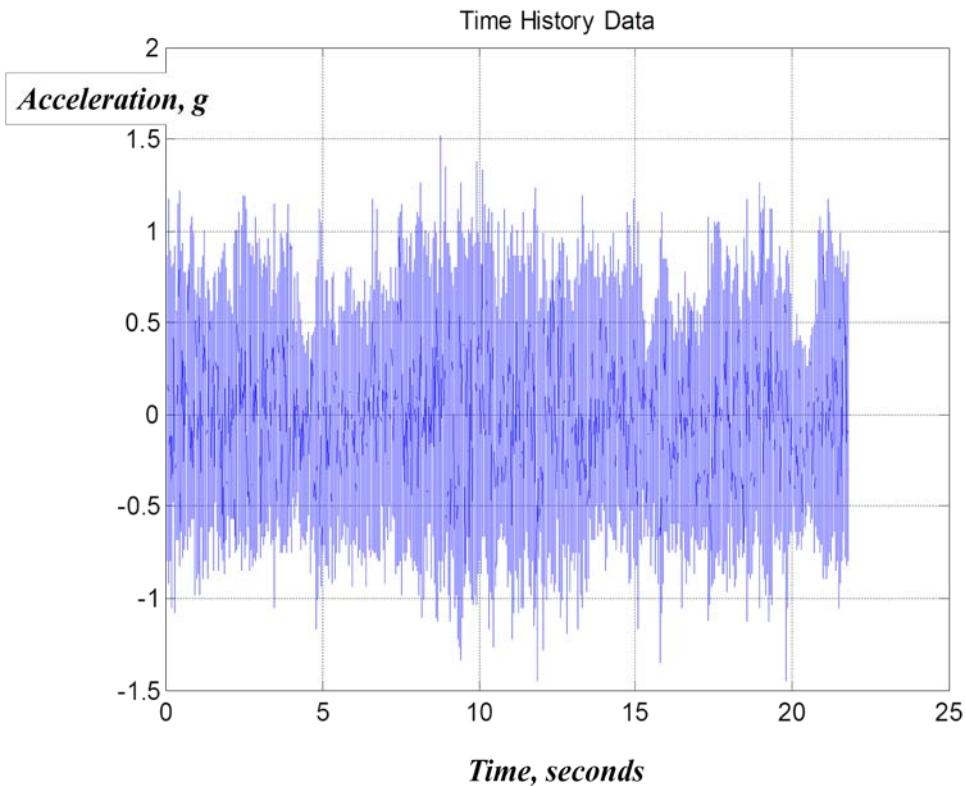


Figure 4 - Test Data Acceleration Time-History

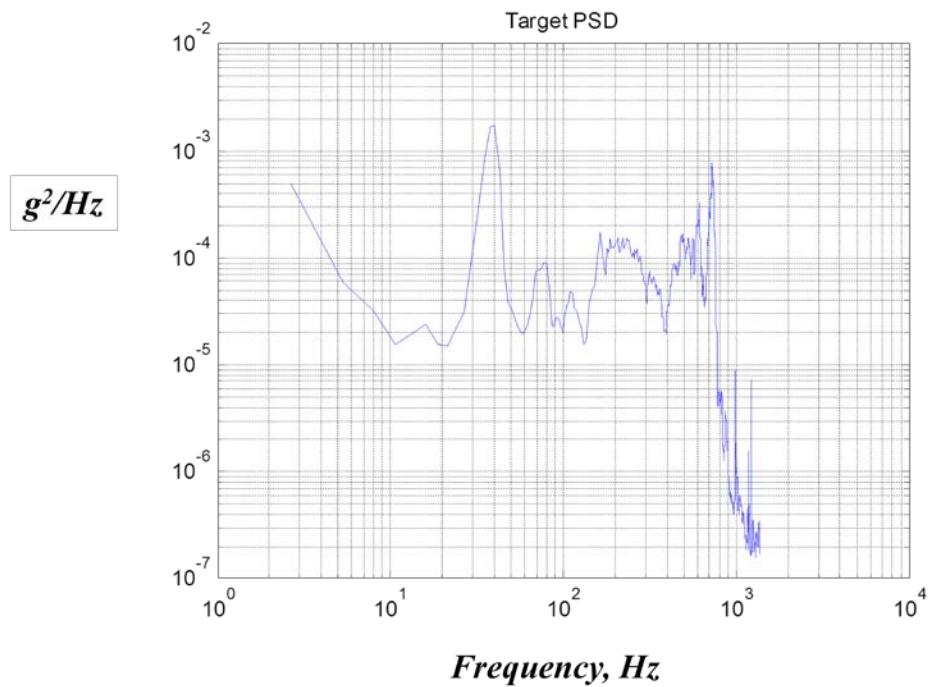


Figure 5 – Target PSD Determined from Time-History Data

Using the relationship (4) and the target *PSD* of Figure 5, the amplitudes A_n for the sinusoids were determined for each frequency, plotted in Figure 6. The first calculation of the synthesized acceleration time-history was developed using these A_n amplitudes and the ramp-up, ramp-down envelope function in Figure 2 with A_e initially set to unity. The ramp-up, ramp-down durations were each set to 10% of the total duration of the data pulse. This resulted in a synthesized base acceleration with a *PSD* error of 13.3% (averaged over all frequencies) and a power error of -14.7% based on the total area under each *PSD* curve. These errors are prior to iterations to adjust A_e . After the iteration process characterized in Figure 3, the final value of A_e converged to 1.083 after five iterations. With this value, the power error was reduced to $1.5 \times 10^{-5}\%$. The final synthesized base acceleration is plotted on Figure 7. The final *PSD* error was 6.7%. The *PSD* of the original time-history data and the *PSD* of the synthesized base acceleration are plotted on Figure 8.

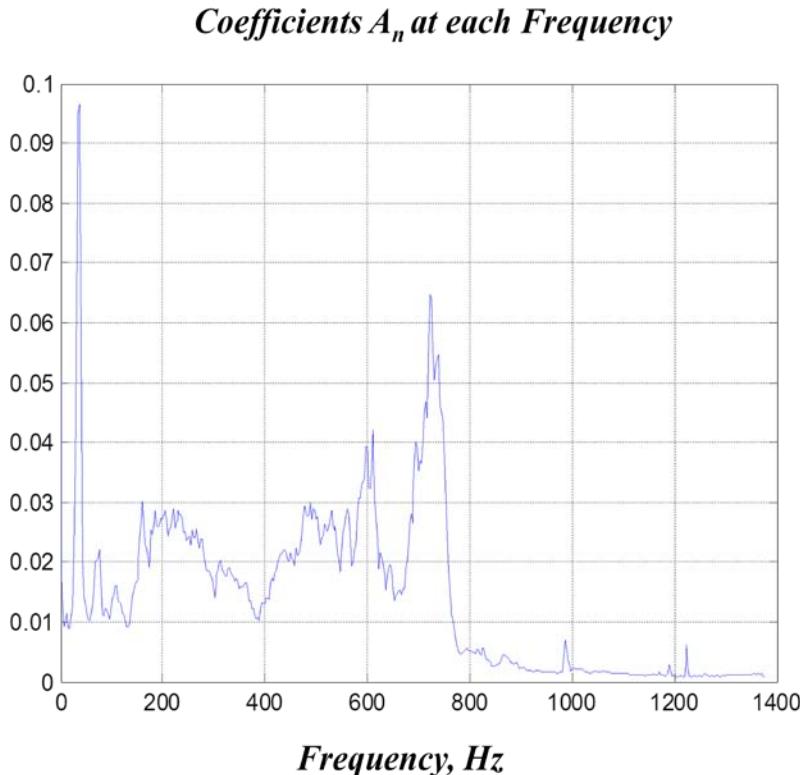


Figure 6 – Sinusoid Amplitudes A_n Determined from Target PSD

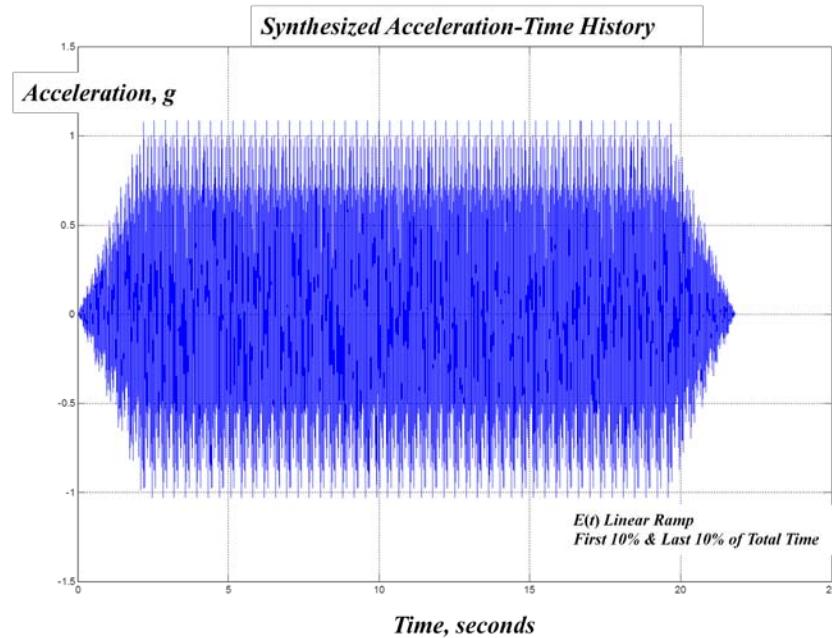


Figure 7 – Final Synthesized Base Acceleration

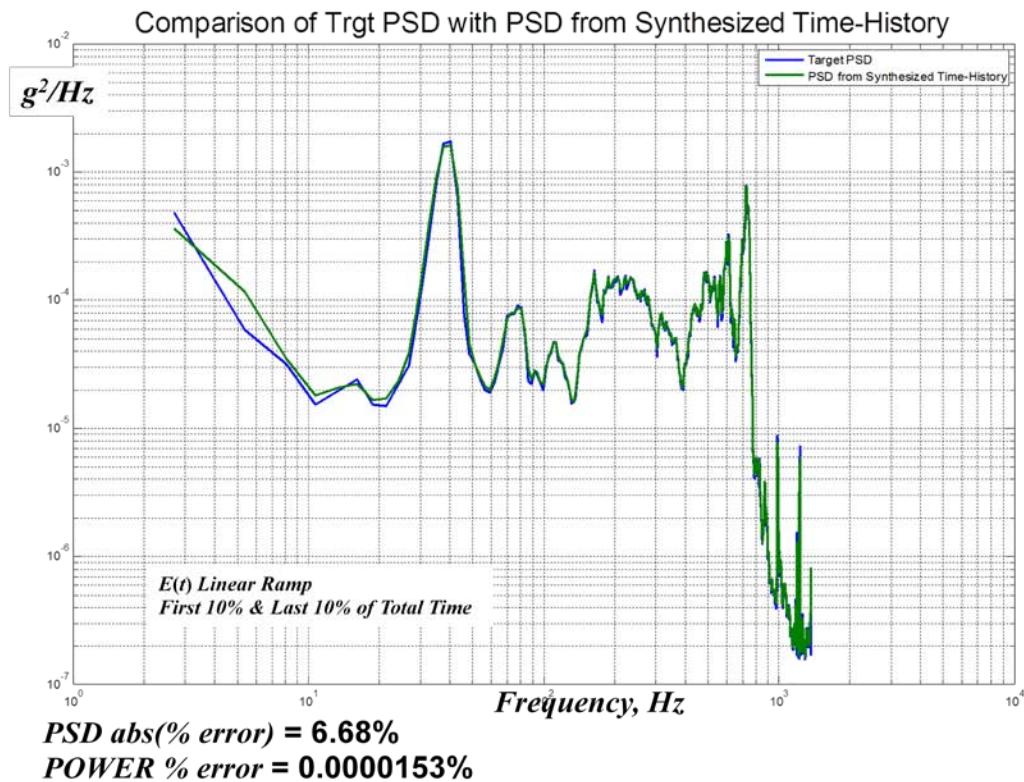


Figure 8 –Comparison of Target PSD with PSD from the Final Synthesized Acceleration-Time History (ramp up/down)

Example #2

A second example was examined using the same test data as Example #1 (Figure 4) with the half sine envelope function of Figure 2. As before, the amplitude of the half sine envelope A_e was initially set to unity prior to iteration. Since the same time history data from Example #1 was used, the target PSD remained the same. Hence, the A_n coefficients from (4) are unchanged from those plotted in Figure 5. Prior to iteration to size A_e , the PSD error was 48.6% and a power error was -50.8%. After the iteration process shown in Figure 3, the final value of A_e converged to 1.4261 after five iterations. With this value, the power error was reduced to $2.7 \times 10^{-4}\%$. The final synthesized base acceleration is plotted on Figure 9. The final PSD error was 6.3%. The PSD of the original time-history data and the PSD of the synthesized base acceleration are plotted on Figure 10.

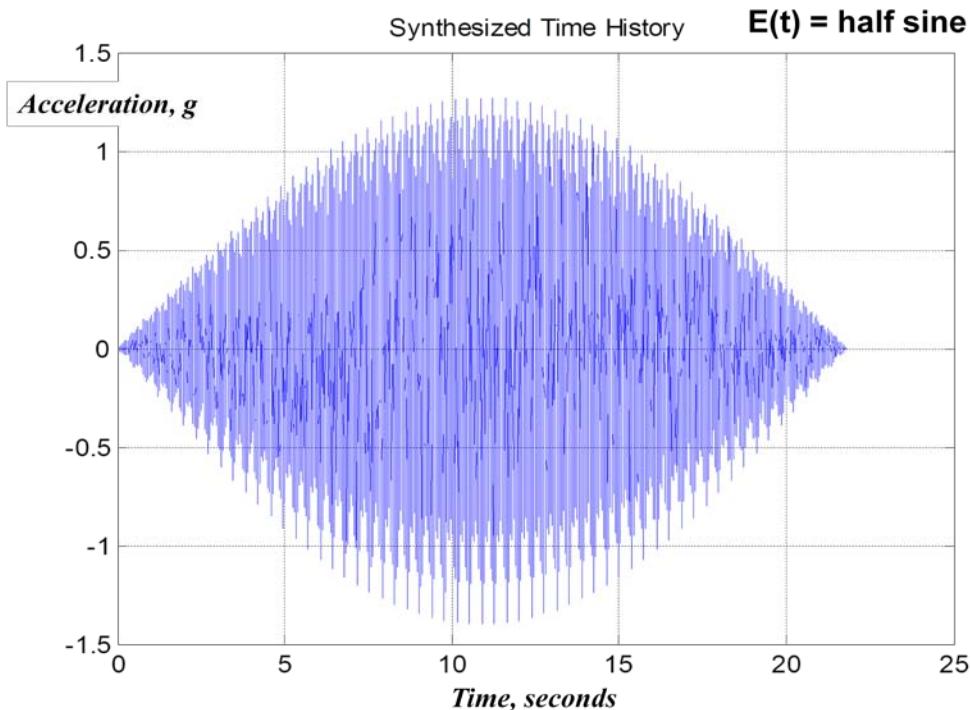


Figure 7 – Final Synthesized Base Acceleration

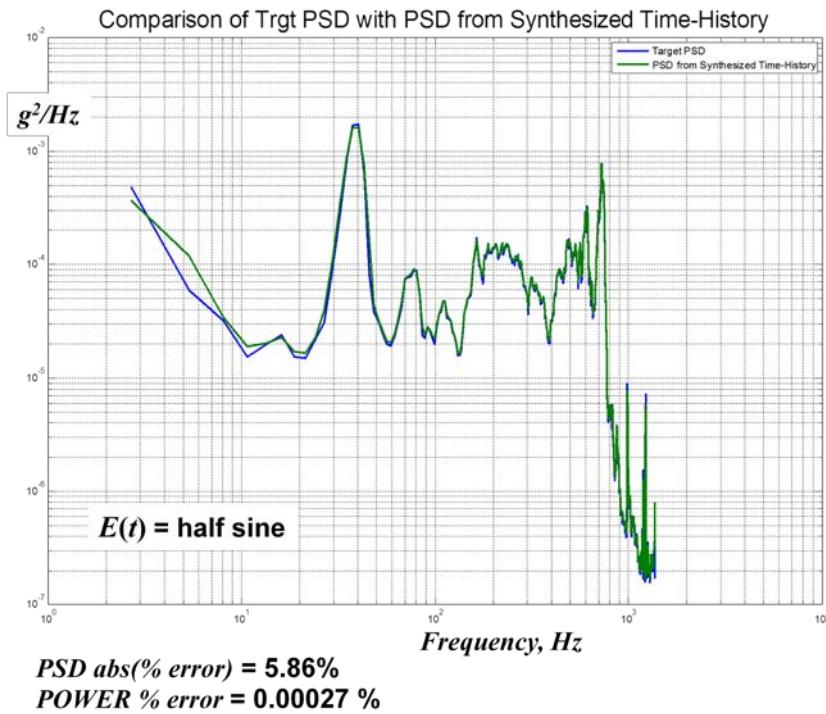


Figure 10 – Comparison of Target PSD with PSD from the Final Synthesized Acceleration-Time History (Half Sine)

Example #3

While for shock and vibration applications the *PSD* is typically a representation of an acceleration spectral density as a function of frequency, there are other applications where the *PSD* is used to represent a physical parameter in the frequency domain. An example of this is the characterization of the surface roughness of ground terrain for combat vehicle mobility testing. The US Army Aberdeen Test Center (ATC) publication TOP 1-1-010 (references list) includes *PSD* roughness plots of numerous ATC vehicle test courses (Belgian Block, Perryman, Munson, Churchville, and others). For this application, the *PSD* magnitude is characterized in units of $\text{ft}^2/(\text{cycles}/\text{ft})$ as a function of Wave Number (frequency) with units of *cycles/ft*. The procedure presented herein is equally applicable to a spectral representation. Using these measured test course *PSD*'s, a spectrum compatible terrain profile can be synthesized. In this case the resulting synthesized profile is the height of the surface (displacement) as a function of distance.

The Belgian Block course was used to demonstrate the utility of the synthesis process for this example. Figure 11 is a plot of the Belgian Block *PSD* which was extracted from a graphical plot in TOP 1-1-101. While the synthesis process works identically as the prior two examples, (1) is rewritten as (5) with symbols that traditionally correspond to distances rather than acceleration and time. In this case the *height(x)* corresponds to the height of the ground as a function of distance *x*.

$$\text{height}(x) = E(x) \sum_{j=1}^{j_{\max}} A_j \sin(\omega_j t + \phi_j) \quad (5)$$

The coefficients A_j of the sinusoids were determined from (4) using the target spectrum and are plotted in Figure 12. The synthesized terrain profile is plotted in Figure 13. For this example, the linear ramp-up, ram-down profile of Figure 2 was used where the duration of each ramp was 10% of the total duration. As a check on the procedure, Figure 14 compares a plot of a 160 foot segment of the synthesized terrain compared with Belgian Block course test data of the same distance. The peak amplitudes of the synthesized terrain profile exceed two inches and are less than three inches, which is also the case with the test data. The frequency content is also generally comparable. By visually counting the peaks, the test data has approximately 13 cycles in 160 feet and the synthesized terrain also has approximately 13 for the same distance. This corresponds to 0.081 cycles/foot which generally agrees with the *PSD* peak which occurs between 0.088 and 0.094 cycles/foot. Figure 15 is a plot of the Belgian Block target *PSD* with that of the synthesized terrain. Newton Raphson iteration sized the envelope peak amplitude, A_e , to match the power of the synthesized terrain *PSD* to that of the target *PSD*. The resulting value for A_e was 1.0335. The error of the synthesized *PSD* was 7.67% averaged over all frequencies when compared to the target *PSD*.

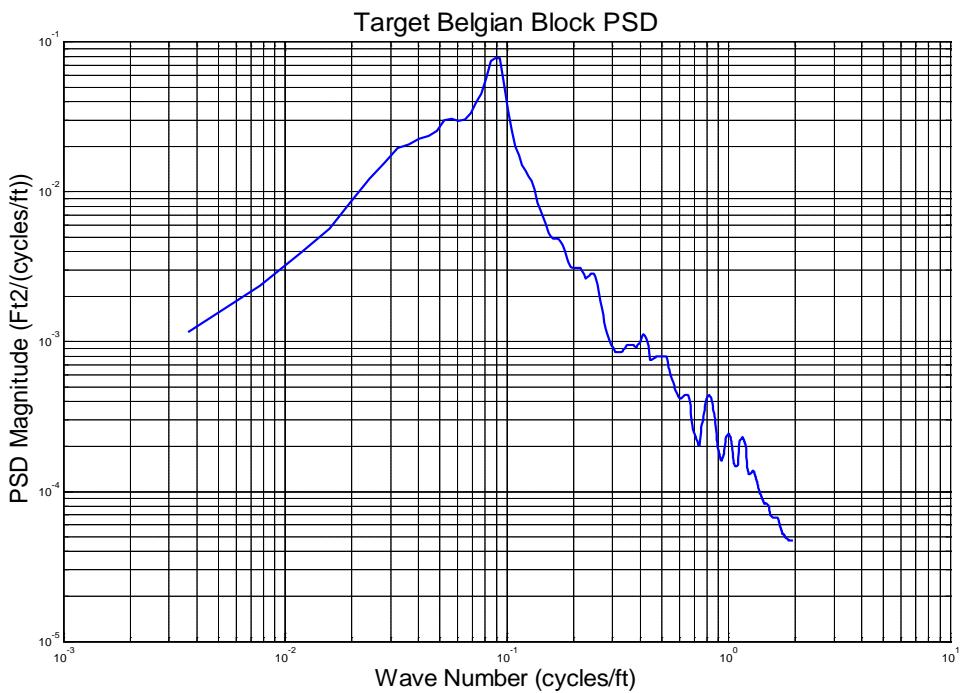


Figure 11 –Target Belgian Block PSD

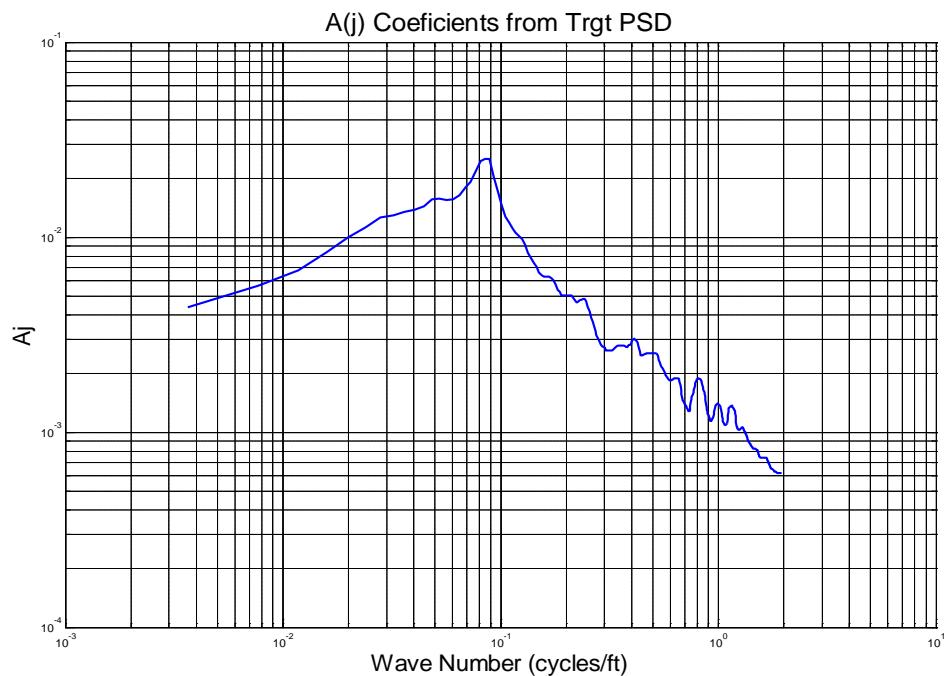
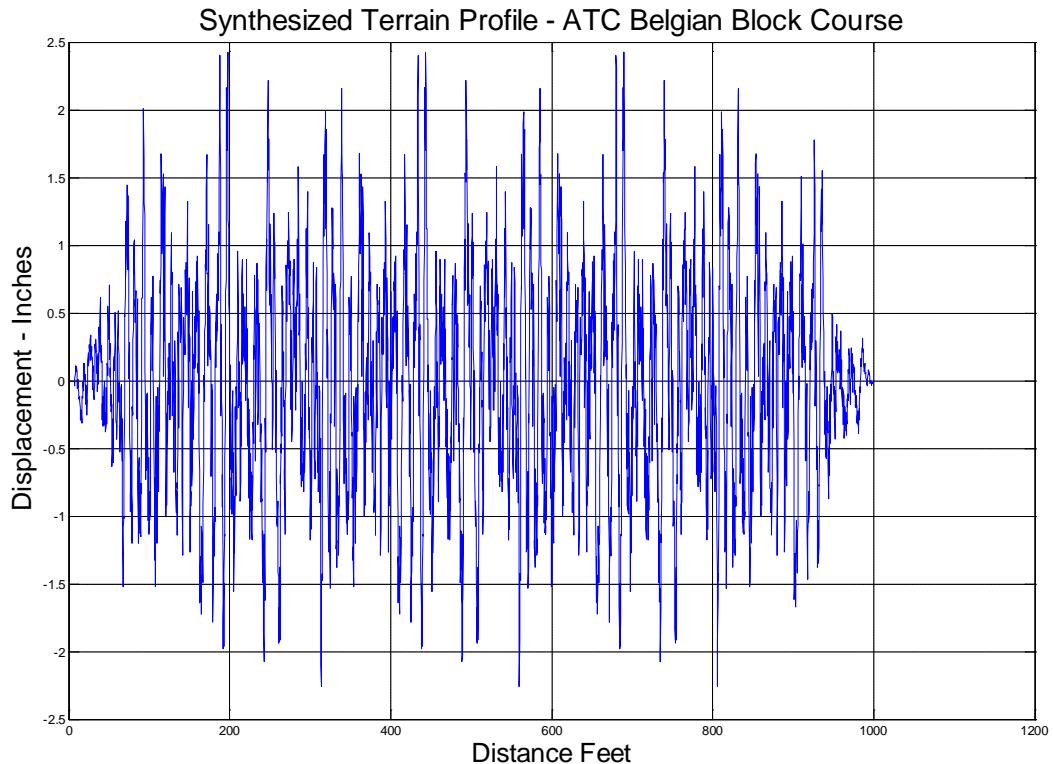


Figure 12 –Coefficients of the Sinusoids of Equation (5)



**Figure 13 –Synthesized Terrain Profile Compatible
with Belgian Block PSD**

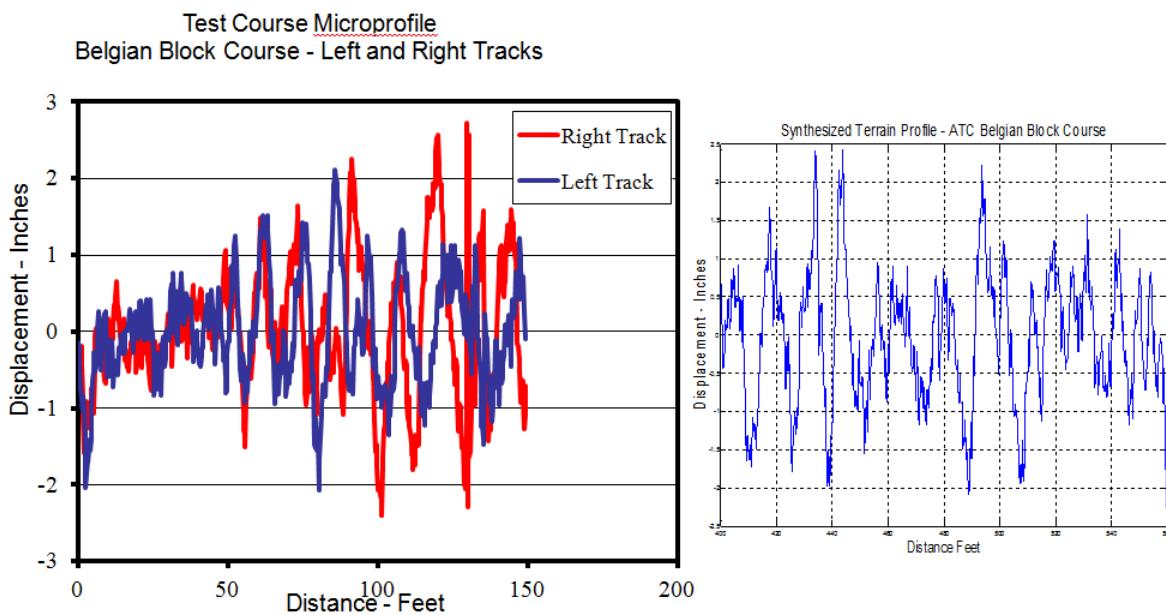
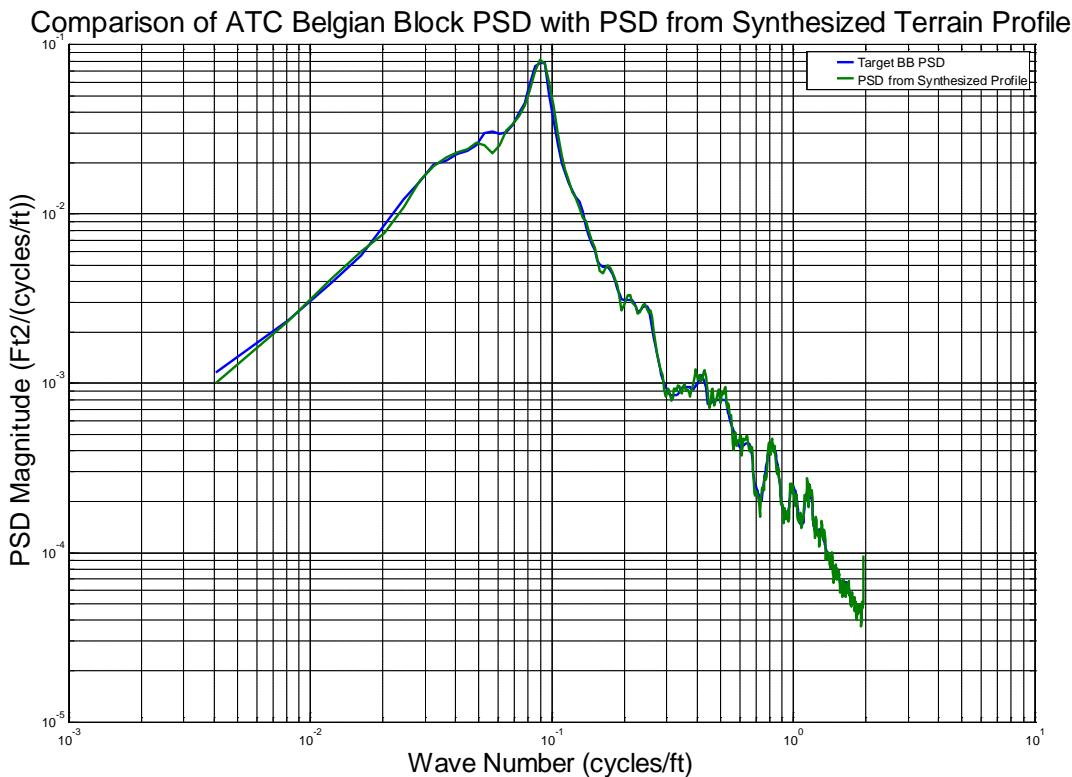


Figure 14 – Belgian Block Test Course & Synthesized Terrain Profiles**Figure 15 – Belgian Block Test Course PSD & Synthesized Terrain PSD**

IV. Conclusions

A simple procedure has been developed to synthesize a base acceleration time-history to be compatible with a prescribed *PSD* using a summation of sinusoids with random phase angles over frequencies that span the frequency range of the *PSD*. Each sinusoid amplitude is sized to match the power of the *PSD* corresponding to that frequency. An envelope function $E(t)$, used to control the overall shape and duration of the synthesized acceleration, is constructed to tailor the synthesized acceleration time-history to be compatible with the physical event. The peak amplitude of the envelope is determined by Newton Raphson iteration such that the total power of the *PSD* that results from the synthesized base acceleration matches that of the prescribed *PSD*.

Three examples were presented. The first two examples are based on the synthesis of an acceleration time-history to be compatible with a conventional *PSD* with units of g^2/Hz . A third example demonstrated the flexibility of the procedure for the synthesis of a ground terrain profile to be compatible with the US Army Aberdeen Test Center's Belgian Block test course *PSD* having units of $\text{Ft}^2/(\text{cycle/Ft})$. Comparison of a segment of the Belgian Block test terrain profile to that of the synthesized terrain were in good agreement in terms of terrain height and frequency content.

V. References:

Alexander, J. Edward, "Structural Analysis of a Nonlinear System Given a Shock Response Spectrum," Proceedings of the 69th Shock & Vibration Symposium, St. Paul, MN, October 12-16, 1998.

Gasparini, D. and Vanmarcke, E. H., "Simulated Earthquake Motions Compatible with Prescribed Response Spectra, Evaluation of Seismic Safety of Buildings," Report No. 2, Publication No. R76-4, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass 02139, January 1976.

Hou, S., "Earthquake Simulation Models and Their Applications," Publication No. R68-17, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass 02139, May 15, 1968.

Housner, G., "Characteristics of Strong Ground Motion Earthquakes," Bull. Seism. Soc. Am., 45 1957, pp. 197-218

Housner, G. and Jennings, P., "Generation of Artificial Earthquakes," Proc. ASCE, 90, No. EM1, 1964, pp. 113-150

Kanai, K, "An empirical formula for the Spectrum of Strong Earthquake Motions," Bull. Eq. Res. Inst., 39, Tokyo, (1961) pp. 85-95

Levy, S. and Wilkinson, J., "Generation of Artificial Time-Histories, Rich in all Frequencies from Given Response Spectra," Nuclear Eng and Design, 38 (1976), pp. 241-251

Lin, C. and Hung, H., "Seismic Accelerogram Compatible with Design Response Spectrum," Inst. for Nuclear Engineering Research, Report INER-0573, DE86 901036, Republic of China, (Aug 1984)

Preumont, A., "The Generation of Spectrum Compatible Accelerograms for the Design of Nuclear Power Plants," Nuclear Eng and Design, 59 (1980), pp.357-368

Rizzo, P., Shaw, D. and Jarecki, S., "Development of Real/Synthetic Time Histories to Match Smooth Design Spectra," Intl. Conf. on Structural Mechanics in Reactor Technology, Berlin, (1973)

Salganik, M., "Constructing Synthetic Accelerograms with Stipulated Response Spectrum," Investiyea, Earth Physics, Vol. 26, No. 8, USSR, (1990), pp. 641-645

Scanlan, R. and Sachs, K., "Earthquake Time Histories and Response Spectra," Earthquake Eng. And Structural Dynamics, Vol. 7 (Aug 1974) pp. 635-655

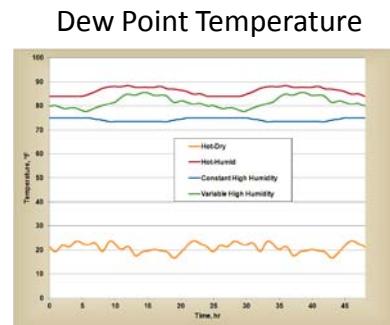
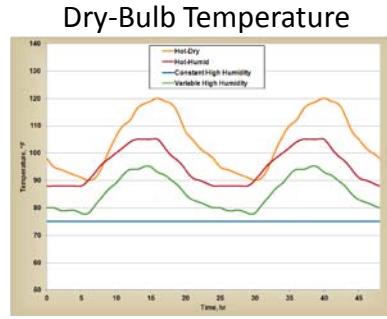
Tajimi, H., "A Statistical Method of Determining the Maximum Response of a Building Structure during an Earthquake," Proc. Second World Conf. on Eq. Eng., II-781, Japan, (1960)

Test Operational Procedure (TOP) 1-1-101, *Vehicle Test Course Severity (Surface Roughness)*, US Army Aberdeen Test Center, Automotive Directorate (CSTE-DTC-AT-AD), 31 October 2006

Context Model C++ Class Library

1. Summary

This appendix points to a C++ class library of nominal and extreme climate-based environmental models drawn from the AR 70-38 and ASHRAE specifications.



Environmental Inputs

AR-70-38 Climates – hot subset

- Hot-Dry – high temperature, low humidity
- Hot-Humid – high temperature and humidity
- Constant High Humidity – sustained high humidity near 100% relative humidity
- Variable High Humidity – varying high humidity near 100% relative humidity

Solar Radiation

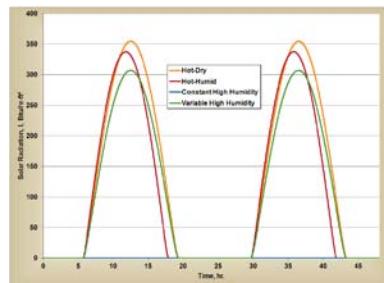


Figure 1: Nominal climate conditions specified in AR-70-38

1.1 Technical Challenges

Synthesize the contents of the specifications to useful library functions. For example the necessary nominal conditions for a thermal model include: temperature, solar radiation, wind speed, and humidity. Optional nominal conditions include: pressure or altitude, contaminants and breakdown of solar radiation into terrain reflection, direct solar (specular, directional), ambient (specular, diffuse), and effective sky temperatures.

1.2 General Methodology

From the specification documents, information in the form of physical constants, tabular data, and algorithms were organized into a C++ class library. Examples and test cases were developed to support usage scenarios. For example, the representation of a thermally loaded vehicle interior which resides within the context of an exterior climate of nominal conditions is shown below:

Simulation Diagram

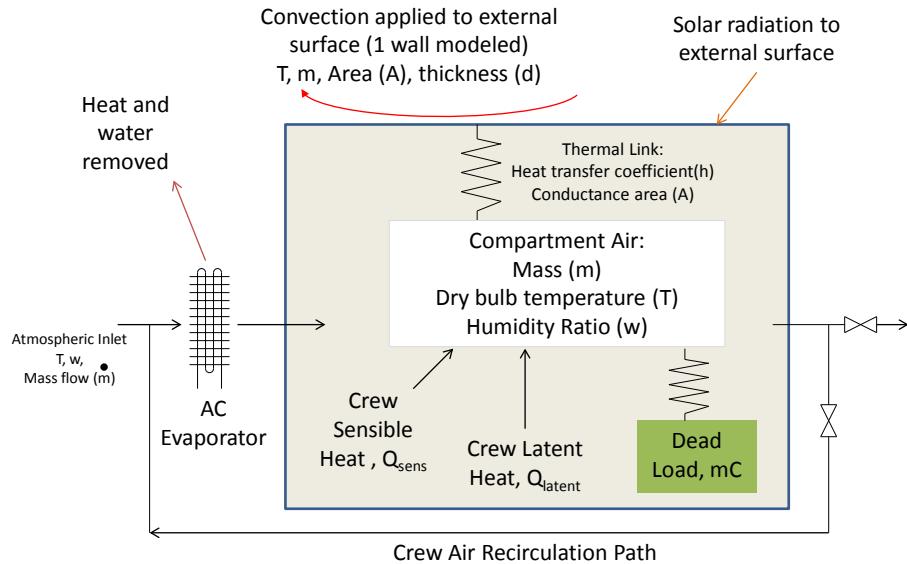


Figure 2: A simulation infrastructure using external contexts

Invoked as stimulus to a thermal solver, the nominal conditions result is shown below

Example Results

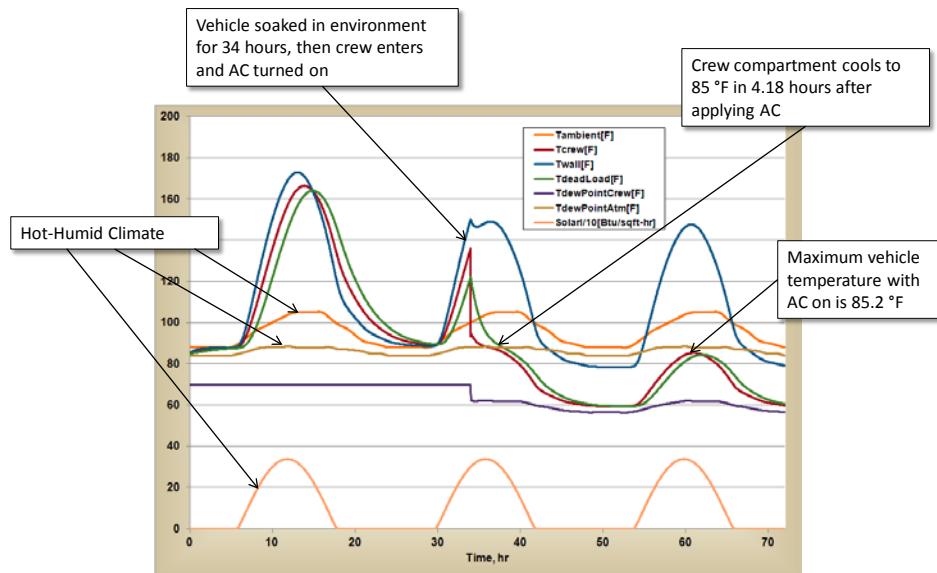


Figure 3: Typical results from applying a nominal diurnal context model.

1.3 Technical Results

The following is a list of C++ classes we have provided:

Climate and Environment Classes

- Class SeaWaterTPProps accesses the thermophysical property tables for salt water as a function of temperature and salinity. See class_SeaWaterTPProps.docx
- Class SolarLoad_ASHRAE encapsulates the ASHRAE Clear Day Model for solar loads. See class_solarLoad_ASHRAE.doc
- Class SolarLoad_ASHRAE2009 is similar to SolarLoad_ASHRAE (see class_solarLoad_ASHRAE.doc) except that it uses the 2009 ASHRAE
- Class ThermalNet models the behavior of a network of thermally connected thermal masses. (see class_ThermalNet.doc)
- Class Climate_AR7038 encapsulates the diurnal cycles of the extreme climatic conditions outlined in Army Regulation 70-38 (AR 70-38). AR 70-38 enumerates 8 climate design types. (see class_Climate_AR7038.doc)
- Class MoistAirMixture approximates the thermodynamic state of a moist air mixture by assuming a binary non-reacting mixture of dry air and water vapor (see class_moistAirMixture.doc)
- Namespace HVAC0 contains auxiliary HVAC functions (see namespace_HVAC.doc)
- Namespace AR7038Environment captures the environment data expressed by Army Regulation AR 70-38 that is not specifically associated with one of the 8 climate design types enumerated in the document and is not covered by class Climate_AR7038
- Class ExplosiveCharge gives the general characteristics of an unconfined hemispherical surface burst of an explosive charge at sea level (see class_ExploriveCharge.doc).
- Class UnderwaterBlast gives the peak blast pressure, characteristic time, impulse and energy flux density for a spherical explosive charge detonated under water (see underwaterBlast.doc).

Environment Example

Namespace AR7038Environment represents the portions of the AR-70-38 standard that are not categorized in the 8 climate design types. Within namespace AR7038Environment are nested namespaces which group data that address sand and dust particle sizes. An overview of the nesting is given by the table below.

SandDust – particle sizes	Aircraft – particle size, density near aircraft, eg helicopter downwash
	SurfaceVehicle – particle size, density near surface vehicle

	Natural – particle size, density in natural surroundings	FineParticleTesting – typical particle size for testing equipment that is sensitive to fine particles
		LargeParticleTesting – typical particle size for testing equipment that is sensitive to large particles

Use of this namespace allows access to amount and size information about particles for each of these environments. The information is available for use programmatically as illustrated in the following example:

Code Example :

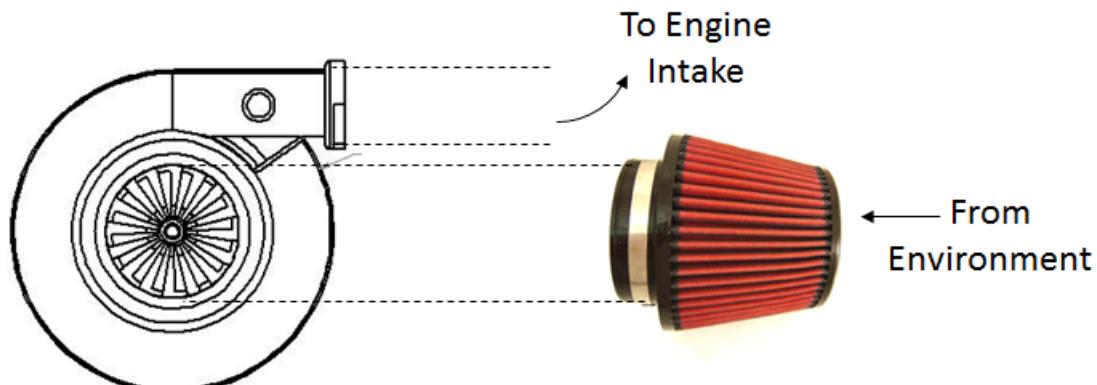
Write the particle density and typical minimum and maximum diameters for environments near operating surface vehicles.

```
#include <iostream>
#include "AR7038environ.h"

int main() {
    std::cout << AR7038Environment::SandDust::SurfaceVehicle::rho << '\t'
        << AR7038Environment::SandDust::SurfaceVehicle::diameter_min_typical << '\t';
    << AR7038Environment::SandDust::SurfaceVehicle::diameter_max_typical << '\t';
    return 0;
}
```

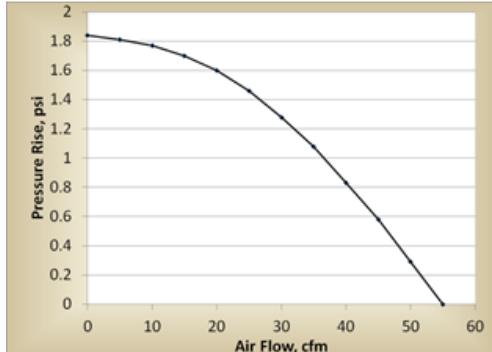
Example of Usage for Vehicle Design and Evaluation :

Determine the increasing pressure drop of a filter due to the buildup of particles. In this example, an intake air filter for a vehicle engine is considered. The notional system, illustrated in the figure below, consists of an intake blower such as a supercharger or turbocharger and an inlet filter. The system duct work, fittings, and inlets and outlets are modeled as having zero resistance.

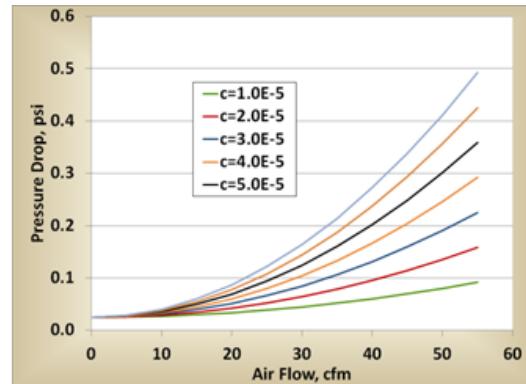


The blower is characterized by a pressure output vs. flow curve, which gives the relationship of decreasing pressure for a rising air flow. The filter is characterized by a series of pressure drop vs. flow curves for different ambient air dust concentrations. These curves shift upwards as

particulates build up in the media. Not shown or modeled in this example is the filter performance relative to the particle size. The blower curve and filter pressure drop curves are illustrated in the figures below.

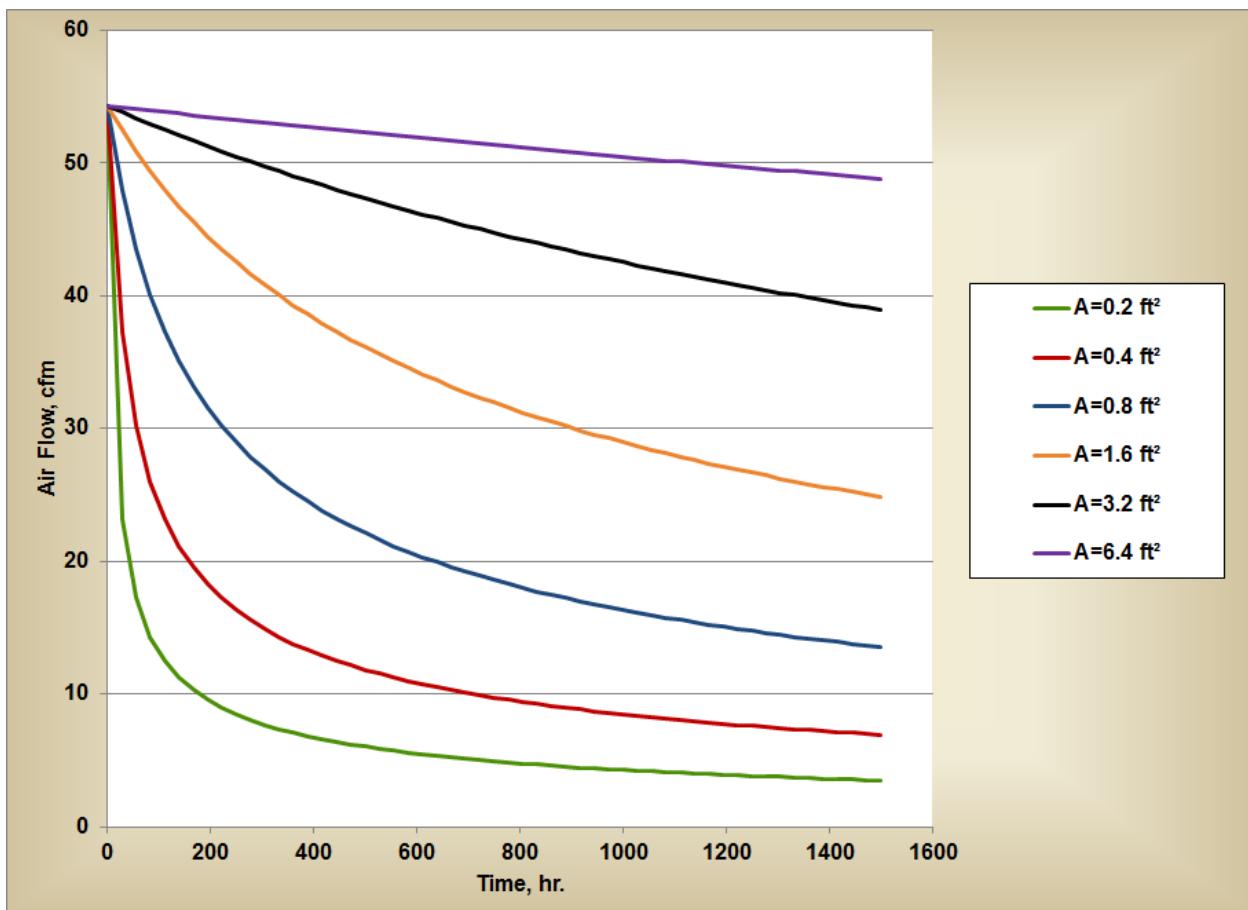


Blower Curve



Filter pressure drop for varying dust concentration

The (quasi) steady state operating point of the system will be the intersection of the blower curve and the filter pressure drop curves. Over time as the filter media accumulates a dust cake, the filter pressure drop curve will shift upwards and the operating point will equilibrate at a higher pressure drop and lower flow rate. The results of this simulation are shown in the figure below for a variety of filter areas.



The C++ code for this example can be found in the non-ITAR svn repository at
`\svn\Models\Composite\FluidThermal\test\IntakeFilter`

The C++ library code containing the AR-70-38 for dust and other environmental factors is located in the non-ITAR svn repository at
`\svn\Models\Composite\FluidThermal\src`

1.4 Important Findings and Conclusions

Used as libraries to run examples.

1.5 Implications for Further Research

**Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2013
August 4-7, 2013, Portland, USA**

DETC2013-13318

MASSIVELY PARALLEL DISCRETE ELEMENT MODELING OF WHEELED MOBILITY ON GRANULAR TERRAIN *

Ryan Houlihan

Summer Intern at Mobility and Robotics Systems Section
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena CA 91109
Email: ryan.houlihan90@gmail.com

Rudranarayan Mukherjee

Mobility and Robotics Systems Section
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena CA 91109
Email: Rudranarayan.M.Mukherjee@jpl.nasa.gov

ABSTRACT

Quantitatively understanding wheeled mobility on granular terrain such as sand or gravel is critical for design and operations of ground vehicles for terrestrial or extra-terrestrial applications. While the Bekker-Wong theory of wheeled mobility and its derivatives have been used in many applications, the static nature of these formulations are limiting in understanding mobility in deformable terrain under dynamic mobility conditions. Single wheel hardware experiments in laboratory settings and detailed modeling of wheel-terrain interactions are two avenues currently being actively pursued to develop quantitative understanding of wheeled mobility. In this paper, we present findings of massively parallel discrete element modeling of wheeled mobility on granular media such as sand. We present a brief overview of the underlying methodology and then focus on the results of the simulation. In these simulations, we model the inter-granular interactions and interactions between the wheel and the granules with an objective of using high fidelity first-principles approach to capture emergent behavior in these complex and highly dynamic phenomena. These simulations typically model millions of granules and use highly scalable software and parallel computing resources to overcome the severe complexity of the problem. We present results of parametric studies with varying levels of both wheel penetration and mobility conditions. These have been modeled to present a quantitative perspective of the diverse behaviors encountered in wheeled mobility on granular terrain.

We have retained the full complexity of the problem by simulating granules of the size encountered in real terrain to overcome the fidelity limited issues of other comparable methods that use much larger granules.

INTRODUCTION

Quantifying mobility of unmanned ground vehicles in natural terrain is essential for their robust design and dynamic operations. Many of the natural terrain encountered in terrestrial applications can be classified as terrain made of compacted granular terrain. For extra terrestrial applications, Martian, Lunar and small body terrains are primarily granular in nature. Unlike on-road mobility, natural terrain poses interesting challenges in quantifying or predicting mobility performance as the terrain is subject to deformations and has artifacts such as various levels of compaction, slopes, moisture content and heterogeneity in composition. Consequently, robotic mobility on granular terrain has become an active area of research.

Experimentally, there are two options often pursued. One is full vehicle testing where the robotic vehicle is driven over natural terrain and its ability to traverse the terrain is measured. This avenue provides binary go / no-go as well as interesting measurable quantities such as vehicle attitude, rut depths, and slip among others. While often pursued, this method results in full coupling between the terrain mechanics and vehicle dynamics and is often difficult to use as a means of segregating the ef-

fects of different parameters on mobility. Consequently, results obtained through this type of testing are difficult to extrapolate parametrically and hence raise significant uncertainties on their predictive basis. However, this is a robust means of quantifying mobility if the types of terrains in which the vehicle is expected to traverse are limited in variability or tests are conducted on all types of terrains to be encountered.

Another avenue of mobility testing that has historically been most useful in understanding parametric sensitivity of mobility is the single wheel experiment. In this, a well instrumented wheel is made to traverse over qualified man-made representations of natural terrain in a laboratory setting. For such experiments, the granular terrain simulant for the laboratory experiment is well qualified in terms of composition, compaction, moisture content etc. Similarly, the instrumentation on the wheel enables precise documentation of forward and rotational motion as well as loads arising from interactions. As this type of experiments are well-controlled, the results from these experiments are highly conducive for understanding parametric sensitivities and hence are amenable to extrapolations. New methods have emerged that use methods from optical flow such as particle image velocimetry to quantify the deformation field in the granular terrain simulant with a high degree of spatial resolution and precision. However the scope of effort, time and resources required to generate these results is considerable and have been a limiting factor.

In modeling terrain interactions, two types of methods have been traditionally pursued. One is based on using finite element based methods with a continuum approximation of the terrain. The second is based on semi-empirical (and sometimes ad-hoc) spring-damper type models. Variations of the second type range from a single point interaction to a volumetric penetration model. However, in all cases, the coefficients required for these spring-damper type force interactions are at best obtained from curve fitting experimental data or at worse generated through trial and error or heuristics that have no foundational basis. Recently, a third approach has emerged where the terrain is modeled using discrete elements and the interactions between the elements and the wheel are modeled using Hertzian type contact models or as inelastic frictional contacts. Unfortunately, a majority of these methods (if not all) attempts at using these models are typically associated with gross approximations where the terrain elements are significantly larger (sometimes several orders) than the granules found in natural granular terrain. Similarly, the terrain is modeled as a collection of uniform sized spheres. Further, due to the scale of approximations, the force interaction models between the elements, and the elements and the wheel are often treated as tunable quantities akin to the spring-damper type model. Despite the large size of elements, the uniformity of the size of the elements and the tunable force fields, good results have typically been reported. However, due to the approximations, in their current form these types of models also lack a firm predictive basis.

In this paper, we report our attempt at using discrete element method to model wheeled mobility on granular terrain. We have attempted to retain as much fidelity as possible in the simulations and have moved away from the tuning paradigm of making simulation results match experimental data. Towards this, we have modeled the terrain as a polydispersed heterogeneous medium made of elements that have the same size distribution as the granules found in natural terrain. We have also used a force interaction model where the coefficients are measurable macroscopic entities and have set these to values reported for most natural granular terrains. The results presented here are not obtained iteratively through tuning of the simulations, but from single runs to test their predictive basis. We have modeled a single wheel experimental set up and results from a parametric set of simulations are presented. Due to our effort in retaining the complexity of the physical system in the numerical simulation, we are reporting results from very large number of granules (order of several millions on an average) simulated using massively parallel DEM software and parallel computing systems.

FORCE MODEL

In our simulations granular media is modeled as a polydispersed set of spherical bodies. While in reality granules are not spherical entities the polydispersed nature of our simulation domains, with granules of various diameters, provides a good approximation. In the simulations we use three types of interaction models; (i) interactions between granules, (ii) interactions between the robotic system and the granules and (iii) interactions between simulation box walls and the granules. The simulation box walls model the effect of retaining walls in the containers used in the corresponding single wheeled experiments and CAD files represent the mechanical systems; these are both modeled as a triangular meshes. The interactions of the granules with both themselves and the triangular mesh are described in [1]. Contact is represented as described in [2]. The wall-granule, robot-granule, and granule-granule interaction forces are modeled by the Hertzian Contact Force Model as shown in equation (1).

$$\vec{F} = \sqrt{\delta} R_{\text{eff}} [(k_n \vec{\delta}_{nij} - m_{\text{eff}} \gamma_n \vec{v}_n) - (k_t \vec{\delta}_{tij} + m_{\text{eff}} \gamma_t \vec{v}_t)] \quad (1)$$

In the contact force model, δ is the scalar overlap between two particles (i and j), $\vec{\delta}_{nij}$ and $\vec{\delta}_{tij}$ are the vector components of the overlap along the normal and tangential directions. The tangential overlap is truncated to ensure that the ratio of tangential to normal force is less than or equal to the value of the coefficient of friction μ_s defined for our material. The terms, m_{eff} and R_{eff} , are the effective mass and radius of the two interacting granules where $m_{\text{eff}} = \sqrt{\frac{m_i m_j}{m_i + m_j}}$ and $R_{\text{eff}} = \sqrt{\frac{R_i R_j}{R_i + R_j}}$. All other

terms are derived from macroscopic properties of the materials themselves. Using this force model we do not allow the ad hoc specification of parameters, moving us a step away from tunable simulations where parameters can be changed to match experimental results. Instead the user can only define physically measurable parameters including; Young's modulus Y , shear modulus G , Poisson's ratio ν , and coefficient of restitution e . Using these parameters and the following equations we are able to derive all necessary variables in the contact force model:

$$k_n = \frac{4}{3} Y_{\text{eff}} \sqrt{R_{\text{eff}} \delta_n} \quad (2)$$

$$\gamma_n = -2 \frac{5}{6} \beta \sqrt{S_n m_{\text{eff}}} \quad (3)$$

$$k_t = 8 G_{\text{eff}} \sqrt{R_{\text{eff}} \delta_n} \quad (4)$$

$$\gamma_t = -2 \frac{5}{6} \beta \sqrt{S_t m_{\text{eff}}} \quad (5)$$

$$S_n = 2 Y_{\text{eff}} \sqrt{R_{\text{eff}} \delta_n} \quad (6)$$

$$S_t = 8 G_{\text{eff}} \sqrt{R_{\text{eff}} \delta_n} \quad (7)$$

$$\beta = \frac{\ln(e)}{\sqrt{\ln^2(e) + \pi^2}} \quad (8)$$

$$\frac{1}{Y_{\text{eff}}} = \frac{(1 - v_i^2)}{Y_i} + \frac{(1 - v_j^2)}{Y_j} \quad (9)$$

$$\frac{1}{G} = \frac{2(2 + \nu_i)(1 - \nu_i)}{Y_i} + \frac{2(2 + \nu_j)(1 - \nu_j)}{Y_j} \quad (10)$$

The coefficient of friction adds a upper limit to the tangential force through $F_t = \mu_s F_n$ where F_t is the total tangential force and F_n is the total normal force. In the Hookean case tangential force between two particles grows according to a tangential spring and dashpot model unit $F_t/F_n = \mu_s$ and is then held at $F_t = F_n \mu_s$ until the particles loose contact. In our case, the Hertzian model, our forces are similar except the spring is no longer linear [3].

We can also define a coefficient of rolling friction μ_r and a cohesion energy density k . For the coefficient of rolling friction we add an additional torque contribution equal to

$$\tau_{rf} = \mu_r k_n \delta_n \frac{w_{r,\text{shear}}}{\|w_{r,\text{shear}}\|} R \quad (11)$$

where $w_{r,\text{shear}}$ is a projection of w_r into the shear plane where

$$w_r = \frac{r_i w_i + r_j w_j}{r_i + r_j} \quad (12)$$

and the contact radius $r_c = r_p - 0.5 \delta_n$ where r_p is the particle radius.

The cohesion energy density is defined for use in the linear cohesion model. If two particles are in contact it will add an additional normal force $F = kA$ tending to maintain the contact where A is the particle contact area and k is the cohesion energy density J/m^3 .

SIMULATION INFRASTRUCTURE

The simulations reported in this paper were run using a software developed in house based on open source software [3,4] that is highly parallelized and scalable. This software makes use of spatial decomposition of the simulation domain as well as high efficiency nearest neighbor sorting algorithms [5–8] to reduce computational expense. When using parallel machines spatial decomposition decomposes the simulation domain into multiple 3-dimensional sub domains which are distributed across the parallel processors. As with any method that includes deformation based force fields catastrophic energy spikes and simulation failures are possible if the appropriate temporal integration time step is not chosen. The only limitation in terms of the potential size of the simulation domain is available computational power and memory. Post process visualization is done using Paraview [9]. While results presented here were generated using polydispersed spherical granules, the software is currently capable of modeling aspherical granules that have abstract convex geometric shapes (not just ellipses or boxes). All simulations were run on one of the following two clusters. Cluster 1 is running with RedHat Linux Enterprise Edition 4.4 on the 32-bit Dell Xeon architecture. Each of the 512 compute nodes of cluster 1 runs 2 Pentium 4 Xeon processor with a clock speed of 3.2 GHz, 1 MB of cache, and 2 GB of RAM per CPU. Cluster 2 is running with SuSE 11.1 on a 64-bit Dell Xeon architecture. Each of the 152 compute nodes of cluster 2 runs 12 Intel Xeon X5650 processors with a clock speed of 2.67 GHz, 12 MB of cache, and 2 GB of RAM per CPU.

REGION DEVELOPMENT METHODOLOGIES

Creating a bed of granular media closely resembling the aggregate properties of true-to-use granular terrains is a critical step in effectively simulating wheeled mobility across a granular terrain. In creating our polydispersed granular media we resembled the statistical distribution of granule size found in natural terrain. Also macroscopic material properties used in our force model, such as Young's modulus, were set to match available material data for natural terrain. While granular size distributions and material properties are straight forward to specify in our simulation, it is not possible to define the bulk density and consequently the porosity of our granular regions as an input parameter. This is because porosity or bulk density is obtained in natural media through a gravity assisted settling process and inter-granular interaction dynamics. Towards creating a realistic bed the sim-

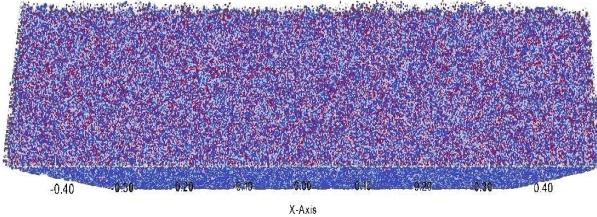


FIGURE 1. Poly-dispersed granular region after settling under a gravity of 1 g. The coloration shows the different radii of the granules that compose the region. In total there are 1,655,064 granules in the region where the distribution of radii is as follows: 15% 2.75 mm, 10% 2.25 mm, 25% 1.75 mm, 50% 1.25 mm. When fully settled, the bulk density of the region when fully settled is 0.769 g/cm³ with a porosity of 44%.

ulation domain was initially discretized into a uniform grid of ghost spheres where the diameter of the ghost spheres was equal to that of the largest physical granule diameter. This is the equivalent of creating a lattice. We introduced physical granules of different sizes based on the statistical distribution of the terrain material into the ghost spheres in the lattice. While it may at first be tempting to introduce these granules in their bounding ghost spheres concentrically, this unfortunately creates an artificial regularity in the granular bed. Instead it is advantageous to randomly offset the centers of the physical granules with centers of the ghost spheres. The offset vector components can be chosen using a random number generator. This irregularity is better able to capture the heterogeneity found in natural terrain. The insertion of these granules into the ghost spheres can easily be performed concurrently and hence is amenable to highly scalable parallelization.

Initially this process will create a region of loosely packed granules with a low bulk density. By allowing the granules in the simulation domain to settle under gravity, the volume is reduced by several orders of magnitude. Once the granules have come to rest we shake the entirety of the simulation box to model compaction of the bulk material. After successive steps of mild shaking we converge to regions with bulk densities comparable to those in the true hardware experiments or natural terrain. Although this process is computational intensive for regions composed of multiple million granules, a single region of compacted granules can be reused for parametric simulations.

WHEEL ROLL SIMULATIONS

For our experiments we used a wheel represented as a cylinder. The wheel was tested at a variety of different speeds, penetration depths, static friction (μ_s) values and rolling friction (μ_r) values.

Before discussing the results of parametric simulations, con-

TABLE 1. Material properties for wheel simulations. The defined properties of the wheel were also the same properties defined for the walls surrounding the region.

Property	Granular Media	Wheel
Youngs Modulus (Pa)	5.00E+006	8.00E+011
Poissons Ratio	0.45	0.3
Coefficient of Restitution	0.75	0.75
Coefficient of Friction	0.5	0.35
Coefficient of Rolling Friction	0.4	0.3
Cohesion Energy Density (Pa)	100000	100000

sider the qualitative validation of our results against experimental data. Note that this is qualitative simulation and there are some parametric difference between the numerical and hardware set ups such as wheel diameter. Figure (2) is made of two rows. The bottom row shows results obtained from particle image velocimetry of a wheel moving on granular terrain [10]. The displacement field of the granular media is shown in color coding with red being high and blue is low. The top row shows results obtained from our numerical simulations. Here too the displacement field of the granular media is shown in the same color coding. We see parabolic curve of high velocity granules extending from the front of the wheel to the back where left is the direction of motion for both images. The bottom image also shows this very distinct parabolic curve extending from the back to the end of the wheel. Similarly, both images show a cusp or lip of the parabola as well as lower speed granules observed through the layer of green. Both images show transitions from blue to green to yellow to red in similar regions. There are some minor observable differences. These arise from our simulation not modeling the very small granules and hence the image has a fair bit of granularity as compared to the hardware experiment.

For the parametric simulations, we first simulated a wheel rolling across a rectangular granular bed at various speeds and depths. For the bed we created a polydispersed region containing a total of 1,655,064 granules. The granules had a density of 2060 kg/m³ and the distribution of granule sizes by diameter was as follows: 15% at 5.5 mm, 10% at 4.5 mm, 25% at 4.5 mm and 50% at 2.5 mm. The bulk density of the bed was 0.8792 g/cm³ with a porosity of 57.32%. Both the wheel diameter and width is 25 cm and all simulations were run with 0% slip. Gravity was set at 9.81 m/s². The material properties used were the same for all experiments except for variations in our value of rolling friction and can be seen in *Table 1*. These simulations were run on approximately 180 processors for a total run time of 4-5 hours.

In the first set of simulations we investigated how changing velocities and wheel penetration depths affect the force in the horizontal (x) direction and the vertical (z) direction. The results of these simulations are shown in *Fig. 6*. We used the standard values listed in *Table 1* for all these simulations. The different

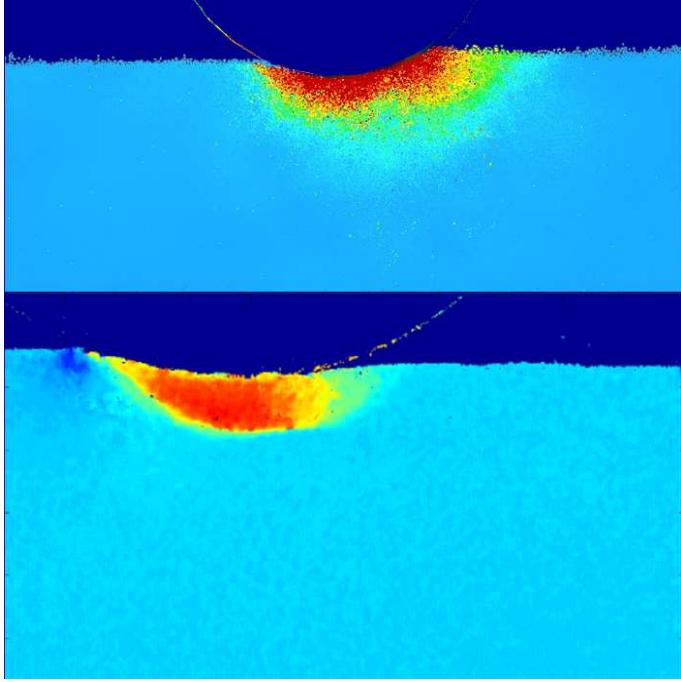


FIGURE 2. **Top:** View of the velocity wave forms forming under the wheel during a simulation run. **Bottom:** View of the velocity wave forms forming under a wheel as it rolls forward. Images is from experimental tests done at CMU [10]

speeds used are 10 mph, 15 mph, 20 mph, and 25 mph. For each of these speeds we used three different depths of penetration into the soil. These values were 1 cm, 1.75 cm, and 2.5 cm. Figure 3 shows a montage of the results obtained from these parametric simulations.

In the second set of simulations we investigated how changing values of the coefficient of friction, μ_s , affects the forces in the x and z directions. We ran these simulations for the 20 mph and 15 mph speeds at a depth of 1.75 cm. For both these speeds we varied μ_s by ± 0.2 . The results of these simulations are listed in (Fig. 7).

In the third set of simulations we investigated how changing values of the coefficient of rolling friction, μ_r , affects the forces in the x and z directions. We ran these simulations for the 20 mph and 15 mph speeds at a depth of 1.75 cm. For both these speeds we varied μ_r by ± 0.2 . The results of these simulations are listed in (Fig. 8).

As depth and speed increased (Fig. 6) so did the \hat{x} and \hat{z} components of the force vector. We also noticed a relationship between F_x/F_z depended on depth the wheel was placed into the material rather then on the speed the wheel was moving. Other interesting results occurred when adjusting the coefficient of friction, μ_s , as shown in (Fig. 7). As we increased μ_s the force opposing our motion decreased and as μ_s decreased the force

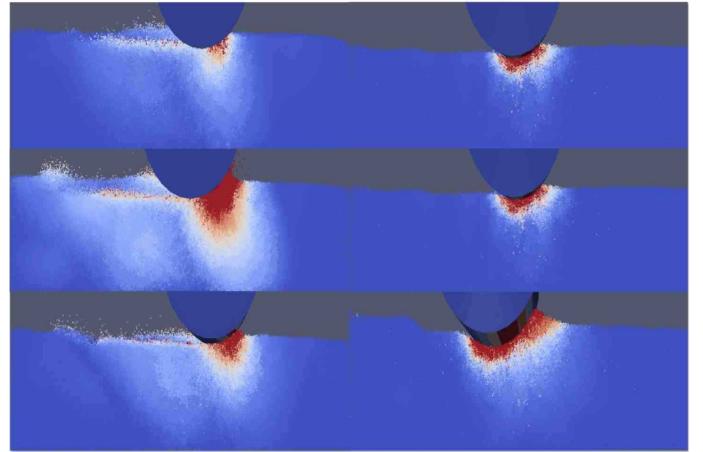


FIGURE 3. Parametric simulations of wheeled mobility on granular terrains. The first column shows the higher speed results with varying penetration while the second column shows the slower speed results for different penetrations. The deformation field of the granular media is shown in color coding with red being high and blue low. The difference in the deformation fields are clearly noticeable

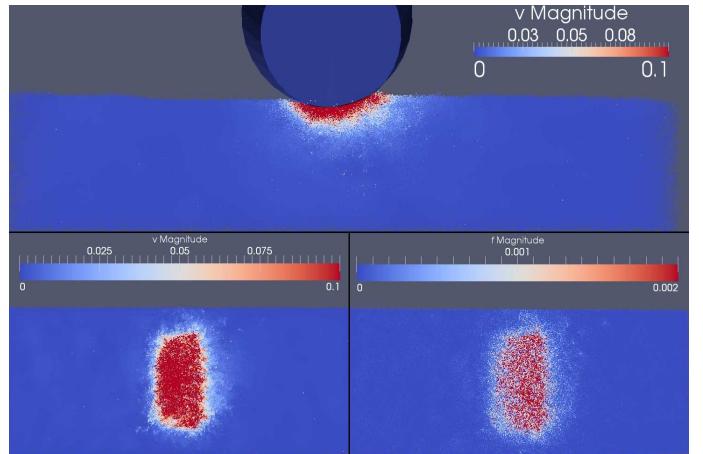


FIGURE 4. Wheel rolling with 0% slip at 25 cm/s (0.55923 mph) to the right at a depth of 1.5 cm into the bed of granules. A parabolic curve of high velocity granules extends from the front of the wheel to the rear with a large accumulation underneath the moving cylinder.

opposing our motion increased. These results make sense if we remember we have kept velocity constant throughout the simulation. Note that this is inter-granular coefficient of friction and not a macroscopic aggregate friction between wheel and terrain. As the coefficient of friction increases, the relative motion between the granules is reduced resulting in a firmer terrain. As the terrain firms up, the displacement field in the granular media reduces and consequently, the resistance to motion arising from terrain deformation reduces. A simplistic interpretation can be that with

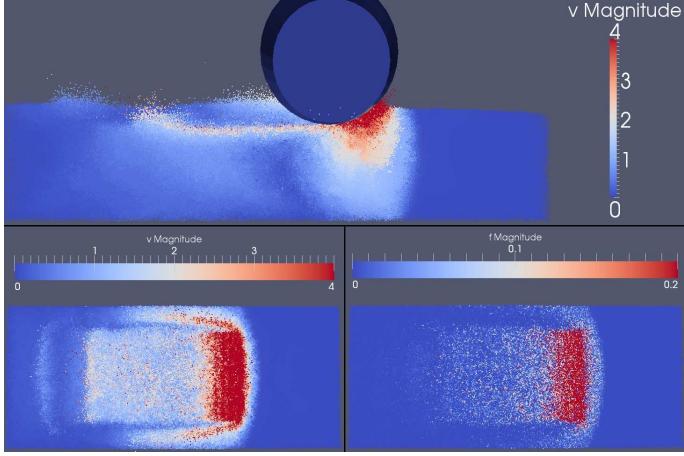


FIGURE 5. Wheel rolling with 0% slip at 11.176 m/s (25 mph) to the right at a depth of 2.5 cm into the bed of granules. A parabolic curve of high velocity granules is visible extending from the front of the wheel to midway back on the wheel. High velocity granules also extend deep into the bed. Behind the wheel the surface which previously interacted with the cylinder is left disturbed with medium velocity granules.

increase in the coefficient of friction, the terrain deformation decreases and the wheel approaches near rolling condition. Hence, the reaction force reduces. This type of emergent dynamic behavior needs to be validated through experimentation.

In (Fig. 4) the wheel is rotating with 0% slip at 0.55923 MPH toward the right. In this image we also see a clear and distinct parabolic curve. From a top view with the wheel made invisible, the force and velocity fields are also in line with what we would expect to see with the bulk of the effects taking place beneath the cylinder. In (Fig. 5) the wheel is also rolling with 0% slip and to the right, but unlike the previous image, it is moving forward at 25 MPH. In this image we also see a distinct parabolic curve. Due to the high speeds, however, this curve is centered more to the front of the wheel and only extends about halfway back on the wheel. We also see this curve of high velocity granules being extended much further into the simulation bed. The surface of the bed is also covered in high velocity granules which have been kicked behind by the rolling wheel. Viewing the force and velocity fields from above shows a parabolic waveform forming in front of the wheel dropping off to the sides of the wheel. Overall the effects we saw with the cylinders interacting with the granular regions was exactly as expected. Our simulation forces were also in line with expectations.

CONCLUSION

We presented the results of parametric simulation of wheeled mobility on granular terrain using massively parallel discrete element method. We reported simulations run on ter-

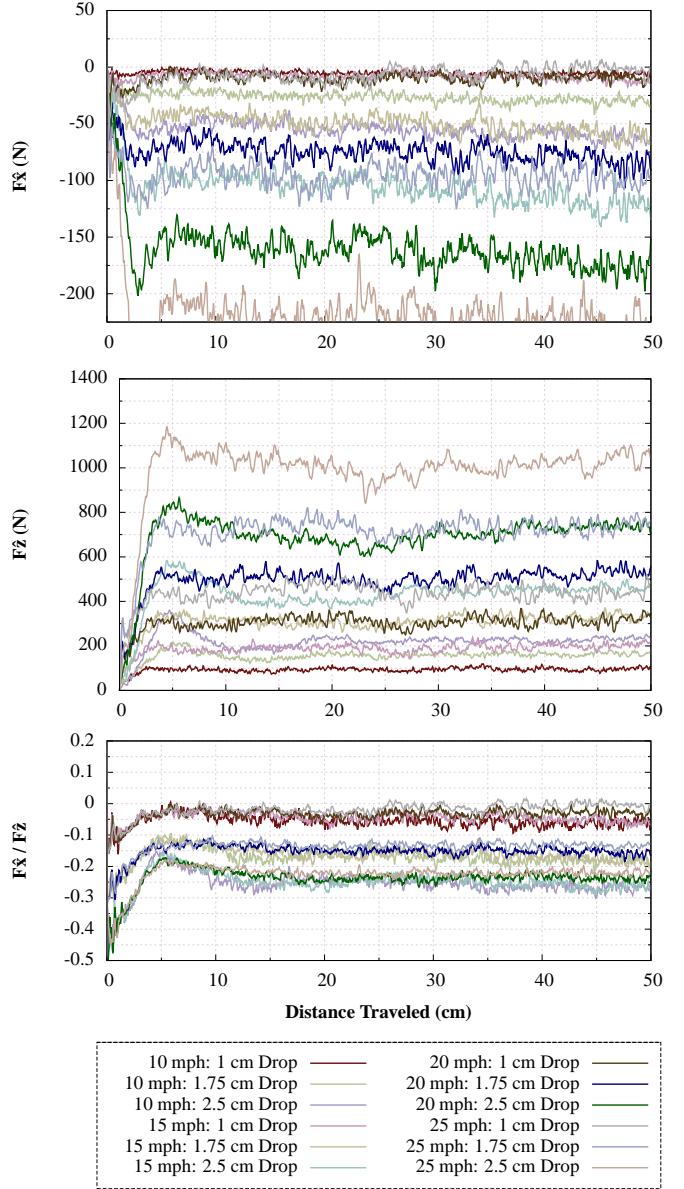


FIGURE 6. Set of simulations performed with varying wheel depths and speeds. We notice that as the depth and speed increases F_x and F_z both increase as well. Also as can be seen in the bottom graph the ratio of F_x to F_z only depends on the distance the wheel is dropped into the bed. The material properties for this simulation are listed in (Table 1)

rains that retain the complexity of natural terrain. Hence these are results of large simulations with millions of granules and the complexity reported here exceeds those of comparable methods reported in literature. Emphasis has been placed on developing single-shot experiments that have a firm predictive basis rather than iteratively turning simulations to match experimental data. The parametric simulations showed the sensitivities of the results

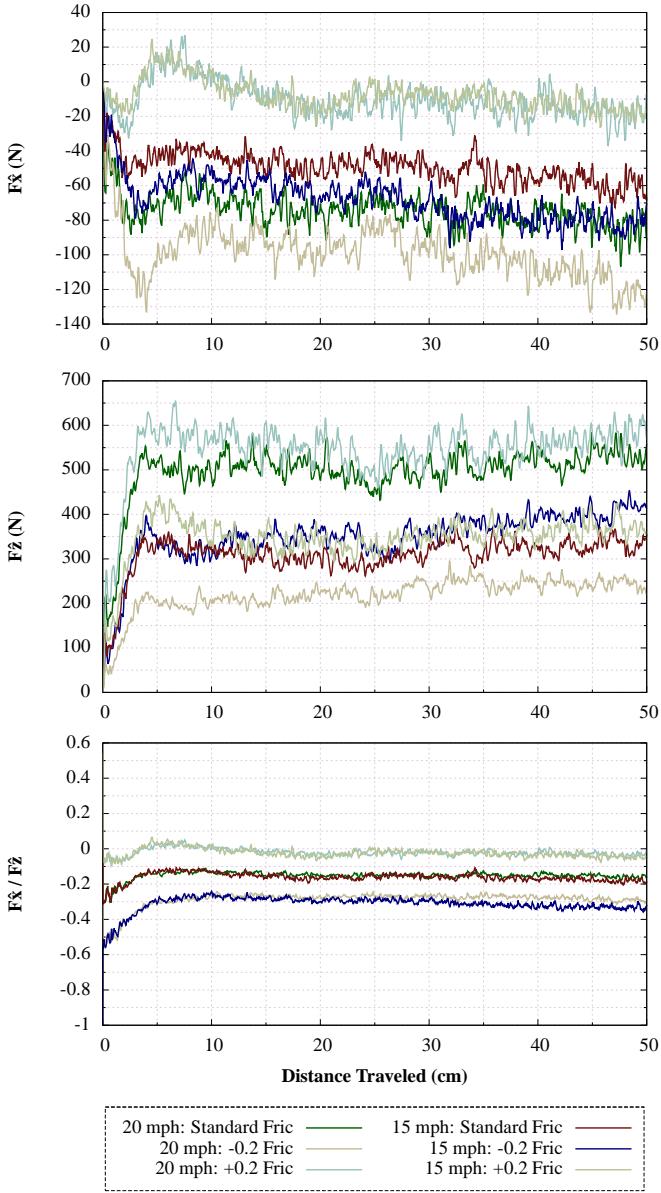


FIGURE 7. Set of simulations performed with varying values for coefficient of friction. As the frictional force increase the F_x force decreases and the F_z force increase. We also notice that the ratio of F_x to F_z only depends on the distance the wheel is dropped into the granular bed. The base material properties for this simulation are listed in (Table 1)

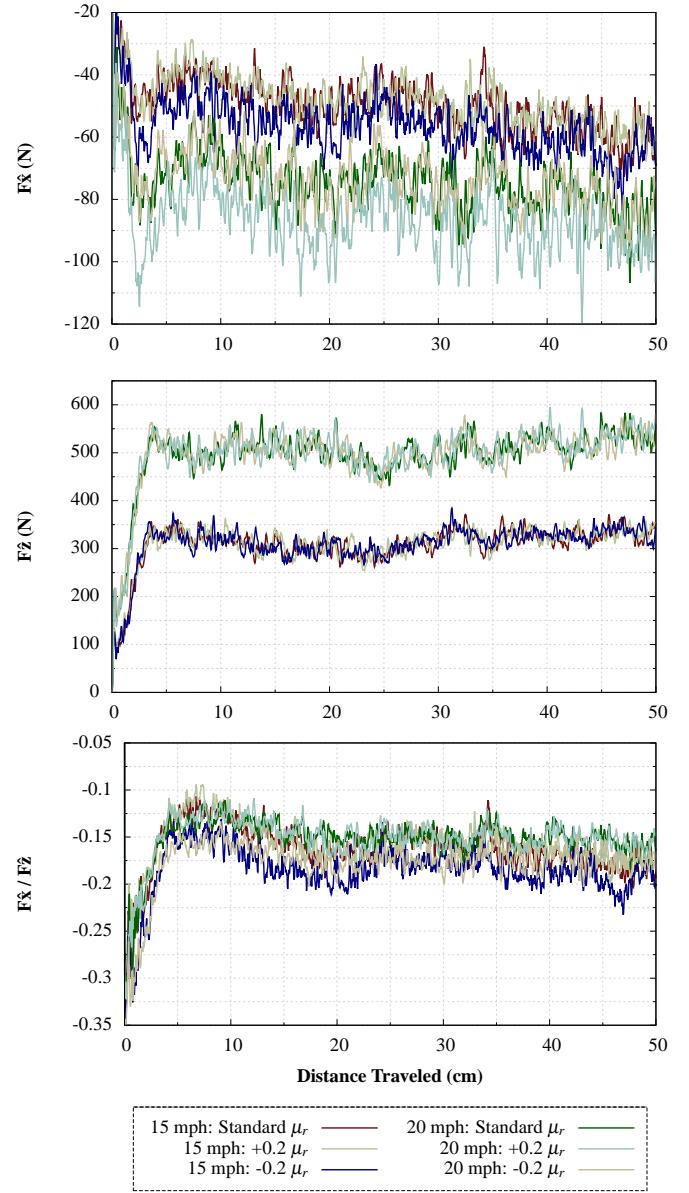


FIGURE 8. Set of simulations performed with varying values of the coefficient of rolling friction. We notice that, as expected, the value of rolling friction has a negligible effect on any of the forces which tend to remain constant. This is due to the implications of Eq. 11 which says that rolling friction only adds an additional torque contribution. The base material properties for this simulation are listed in (Table 1)

ACKNOWLEDGMENT

The research described in this paper was partially performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors thank Scott Moreland, Krzysztof Skonieczny and David Wettergreen from Carnegie

to parametric perturbations. We found that with DEM modeling we were able to reproduce emergent macroscopic behavior. Qualitative validation is demonstrated by comparing our results against experimentally observed data.

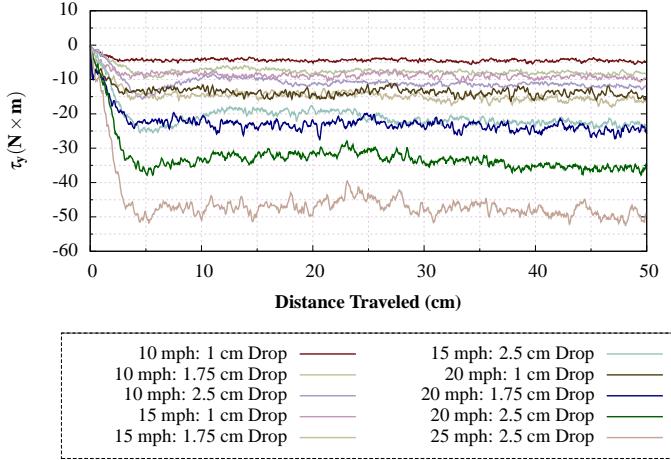


FIGURE 9. Set of simulations performed with varying values of speed and distance. We notice that, as expected, the torque is in the $-\hat{y}$ and it increases with depth and speed. The base material properties for this simulation are listed in (Table 1)

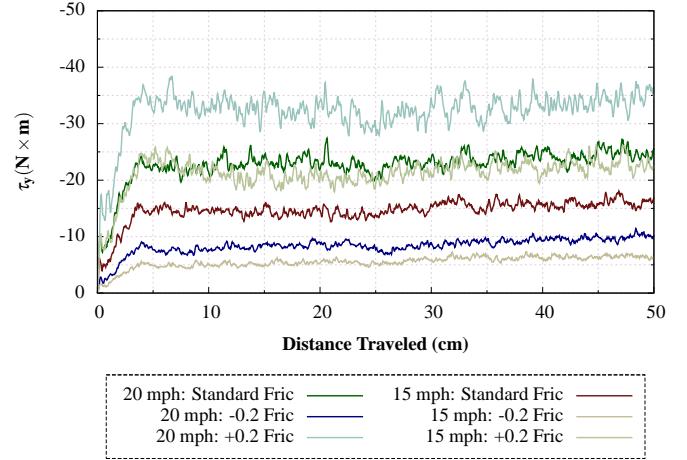


FIGURE 11. Set of simulations performed with varying values of speed and distance and friction. We notice that, as expected, the torque is in the $-\hat{y}$ and it increases increasing friction. The base material properties for this simulation are listed in (Table 1)

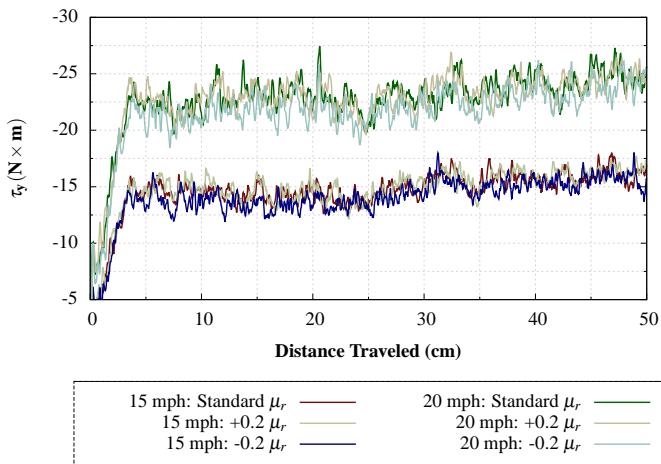


FIGURE 10. Set of simulations performed with varying values of rotational friction. We notice that, as expected, the torque is in the $-\hat{y}$ and it increases with increasing friction. The base material properties for this simulation are listed in (Table 1)

Melon University for providing the results of their particle imaging studies of wheeled mobility on granular terrain. The authors thank the DARPA for funding this work through the Adaptive Vehicle Make program and BAE systems Minneapolis for leading the primary proposal effort. The authors also thank Daniel Challou, Chris Wentland and Michael McCullough from BAE systems for their technical collaboration.

REFERENCES

- [1] Kremmer, M., J. F., 2001. “A computational method for representing boundaries in discrete element modeling”. *International Journal for Numerical Methods in Engineering*, **51 Issue 12**, p. 1407–1421.
- [2] Morris, J., and Johnson, S., 2007. “Discrete element modeling”. *American Society of Civil Engineers: Journal of Geotechnical and Geoenviormental Engineering*, **UCRL-JRNL-237027**.
- [3] Kloss, C., Goniva, C., Hager, A., Amberger, S., and Pirker, S., 2012. “Models, algorithms and validation for open-source dem and cfd-dem”. *An Int. J.*, **12(2/3)**, pp. 140–152.
- [4] Plimpton, S. J., 1995. “Fast parallel algorithms for short-range molecular dynamics”. *J Comp Phys*, **117**, pp. 1–19.
- [5] Munjiza, J., Bicanic, N., and Owen, D., 1993. “Bsd contact detection algorithm for discrete elements in 2d”. In Second International Conference on Discrete Element Methods (DEM).
- [6] Perkins, E., and Williams, J., 2001. “A fast contact detection algorithm insensitive to object sizes”. *Engineering Computations*, **18**.
- [7] Perkins, E., and Williams, J., 2002. “Generalized spatial binning of bodies of different sizes”. In ASCE Discrete Element Methods: Numerical Modeling of Discontinua.
- [8] Williams, J., and O’Connor, R., 1995. “A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries”. *International Journal for numerical and Analytical Methods in Geomechanics*, **12**.
- [9] Henderson, A., 2007. Paraview guide: A parallel visualization applications. Tech. rep., Kitware Inc.

- [10] Moreland, S., Skonieczny, K., Wettergreen, D., Creager, C., and Asnani, V., September, 2011. “Soil motion analysis system for examining wheel-soil shearing”. In International Conference of the International Society for Terrain-Vehicle Systems.

ADDENDUM

On Modeling Tracked Wheels using Massively Parallel Discrete Element Method

Dr. Rudranarayan Mukherjee
Mobility and Robotic Systems Section
Jet Propulsion Laboratory, California Institute of Technology
Pasadena CA 91109
Rudranarayan.M.Mukherjee@jpl.nasa.gov

Distribution: Refer to cover page for Distribution Statement

Introduction

In the body of the main document, we have presented an overview of the underlying methodology of our Discrete Element Method based massively parallel simulations of interactions between compliant granular terrain and wheeled mobility systems. We presented results of parametric simulations that were conducted to understand the effects of speed, penetration and material properties of the terrain on mobility of a cylindrical wheel. This document presents additional results to augment the findings reported in the main document. We aim to transition this addendum to a technical paper in the near future.

Effect of Wheel Profile

Having studied cylindrical wheel motion, we wanted to evaluate the mobility characteristic of a “tracked wheel” that had a more involved geometric profile and was recommended as a track surrogate by our collaborators at BAE Systems. Figure 1 shows a 3D view of the tracked wheel.

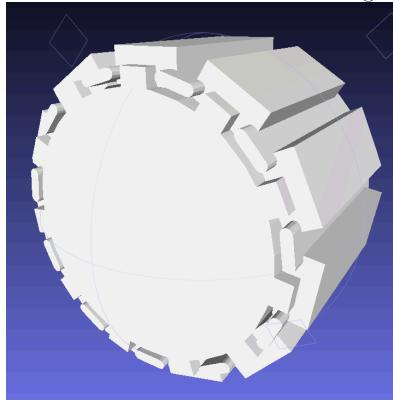
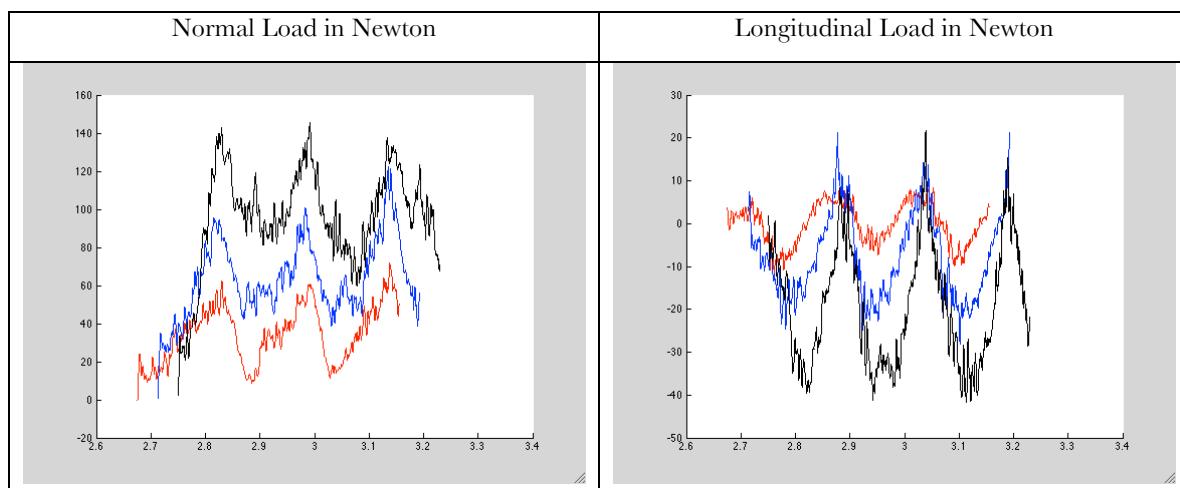


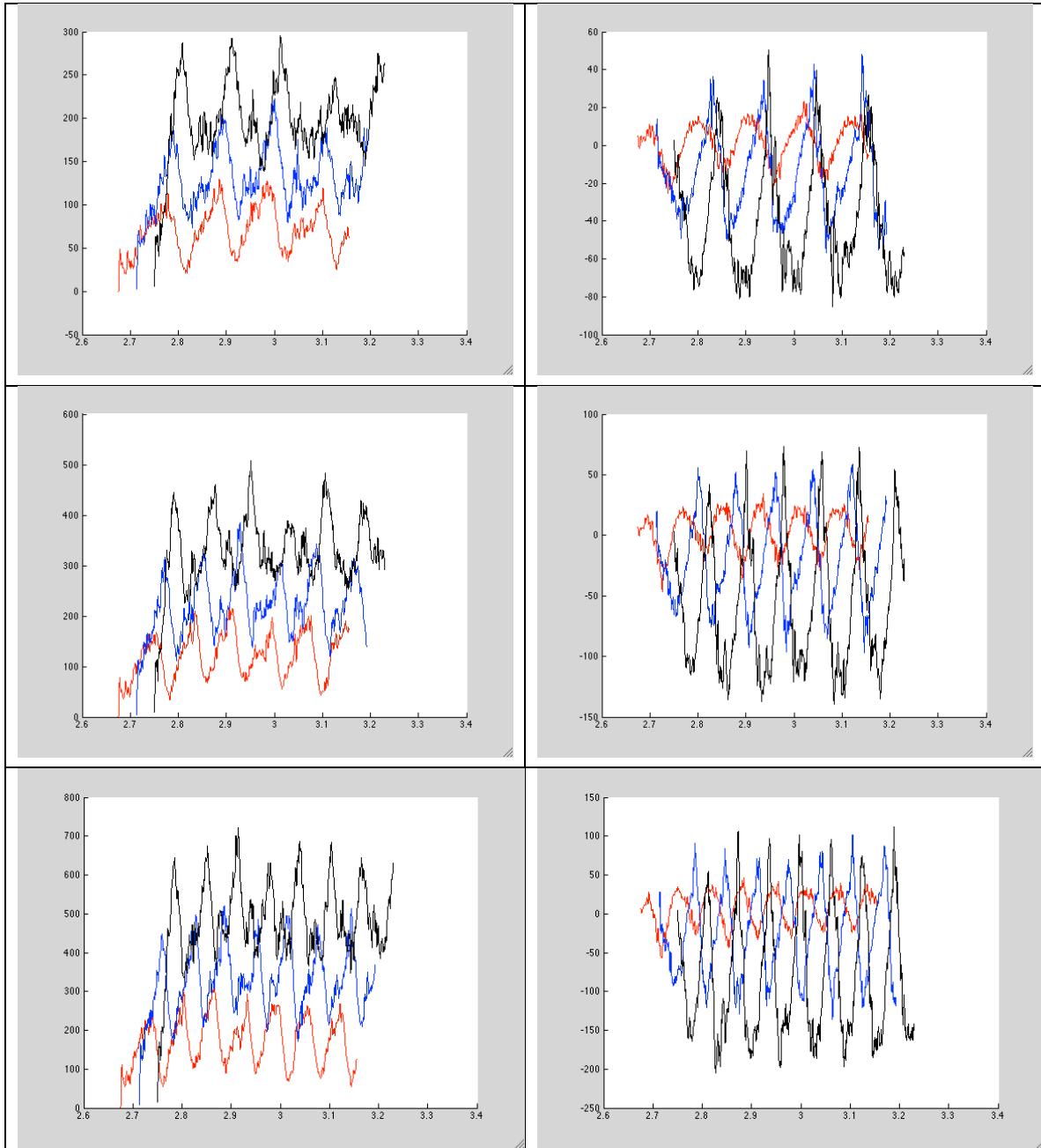
Figure 1 shows a 3D view of the tracked wheel. As observable, this wheel has track pads along its circumference and annular regions between the tracks. We conducted a parametric set of simulations with this wheel similar to the cylindrical wheel i.e. we varied the translational speed of the wheel in zero slip condition to be 10mph, 15mph, 20mph and 25mph. We also varied the penetration or sinkage of the wheel to be 1.0, 1.75 and 2.5 cm into the terrain. We measured the net normal load as well as horizontal load arising on the wheel as it traversed a bed of compacted polydisperse granular media. The material properties on the granular media were set to resemble dry sand. The granular bed was compacted to attain bulk density similar to sand. The simulation consisted of about 2 million granules with a distribution of their bounding spheres ranging from 4mm to 0.5mm.

The simulations were run on the JPL cluster with an average of 200 processors requiring about 4-5 hours of wall clock time per simulation.

Results:

In the following figures, the normal load and longitudinal loads obtained from the simulations are plotted as a function of time. The vertical axis is load in Newton while the horizontal axis is time in seconds. The figures are arranged in the form of a 4x2 table where the left column corresponds to normal load while the right column corresponds to horizontal load. The rows are ordered in terms of increasing speed i.e. the first row corresponds to 10mph while the final row corresponds to 25mph. Each plot shows three sets of results shown in red, blue and black corresponding to 1cm, 1.75cm and 2.5cm of sinkage.

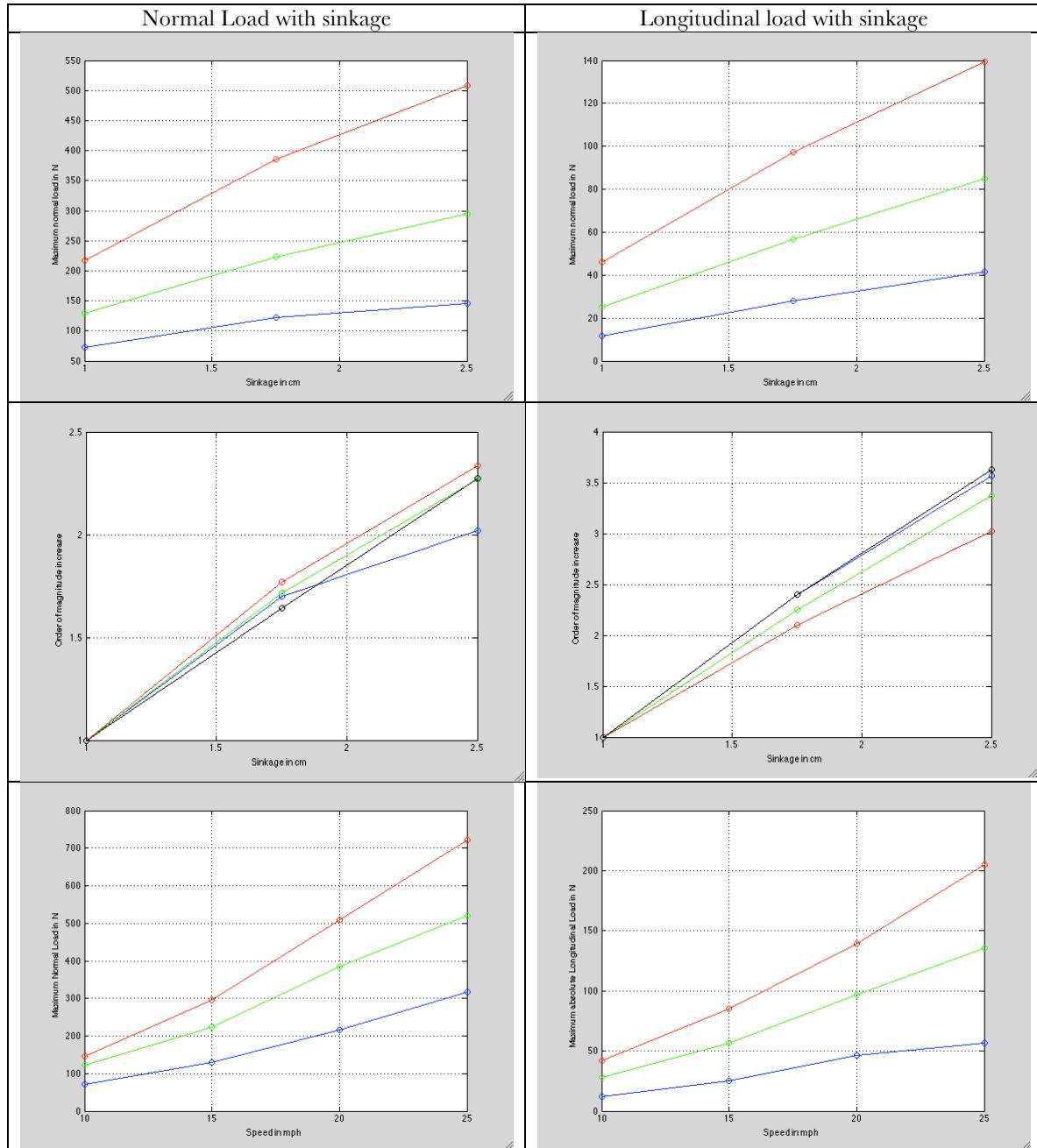




As observed in the figures, the loads vary with the wheel profile. As the wheel rotates, alternately the tracked and annular regions of the wheel come in contact with the terrain. As the track part of the wheel comes in contact with the granular terrain, the extent of compression of the terrain is more compared with the annular region of the wheel. The normal load shows an oscillatory behavior with the same periodicity as that of the pitch of the tracks on the wheel. The additional compression of the terrain also results in an oscillatory longitudinal tractive force on the wheel. As the speed increases, the number of oscillations observed in the figures also increases.

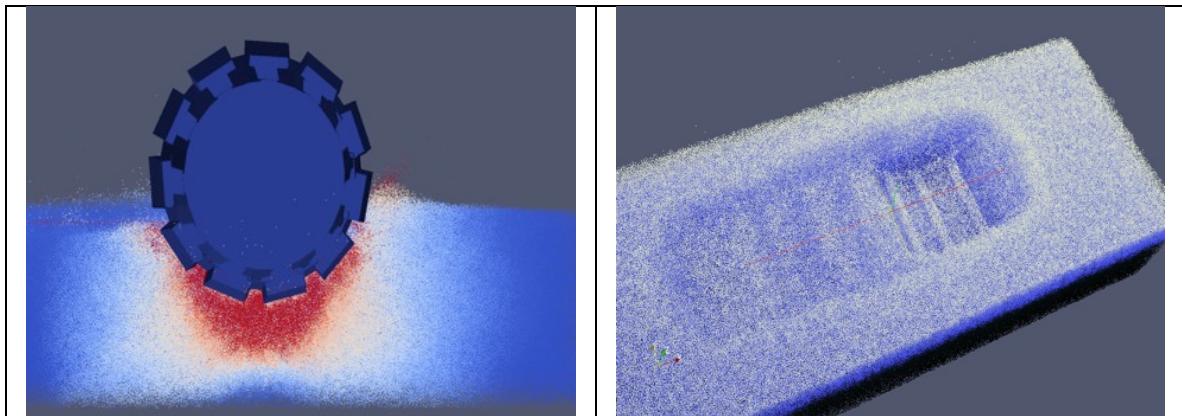
As observed for the cylindrical wheel, higher values of sinkage result in higher normal and longitudinal forces. However the increase in normal force is not linear. In the table below, we plot the factor by which the maximum normal load (left column) and maximum longitudinal load (right column) increase as a function of increase in sinkage value (third row) for the four different speeds (red=10mph, green=15mph,

blue=20mph, black=25mph) as well as the values of the maximum loads (second row). Interestingly, the values are clustered. The 1cm penetration is treated as a base value. Interestingly, the factors of increase in maximum normal load seem to cluster for the different values of speeds. This may be interpreted to mean that the factor of increase in normal load is independent of the speed. However, for the longitudinal loads, this clustering is much less obvious and points to a dependency on the speed. The fourth row of the table shows the maximum normal and longitudinal loads as a function of different speeds for constant penetration.



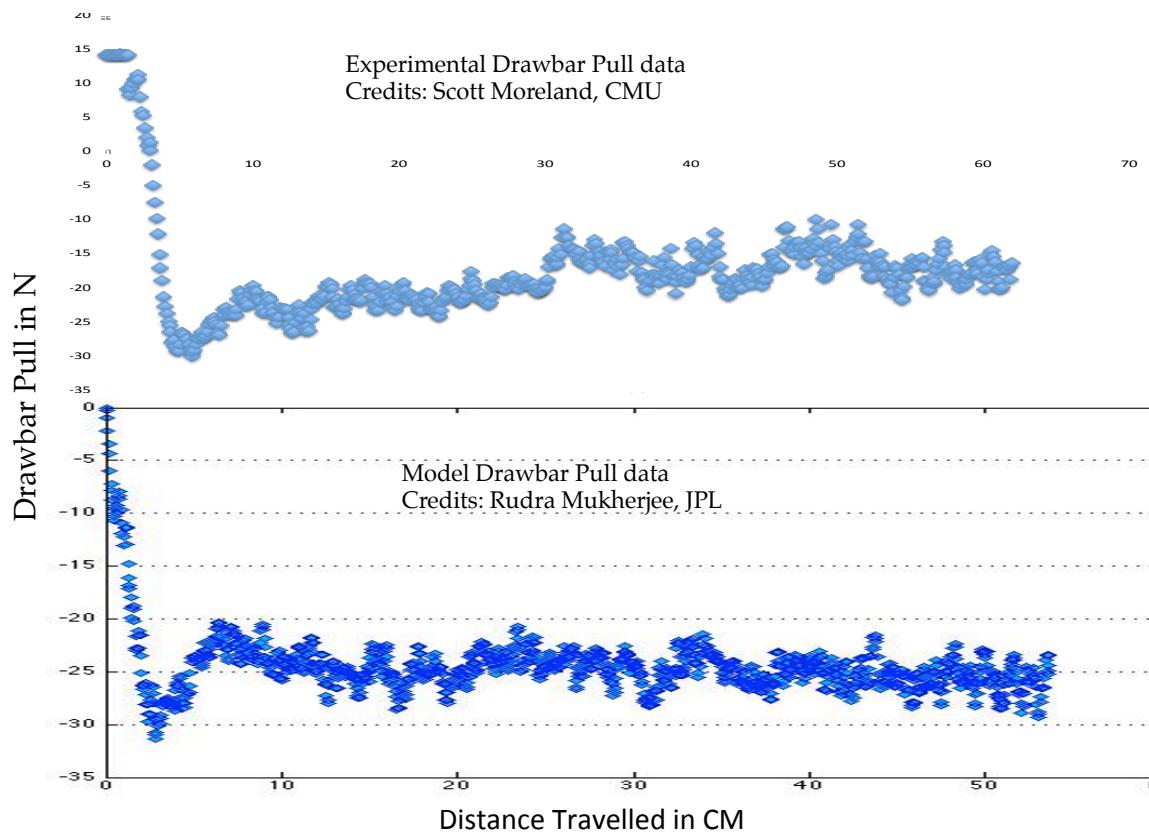
Along with the above quantitative results, we generated full 3D visualizations of the simulation results. The two figures in the following figure show snapshots from these simulations. The figure on the left shows the granules color-coded by their speed. The deformation field is significantly different from the cylindrical

wheel case as the granules conform to the shape of the wheel. In the figure to the right, the wheel is made invisible and the imprint of the wheel on the granular bed is clearly visible.



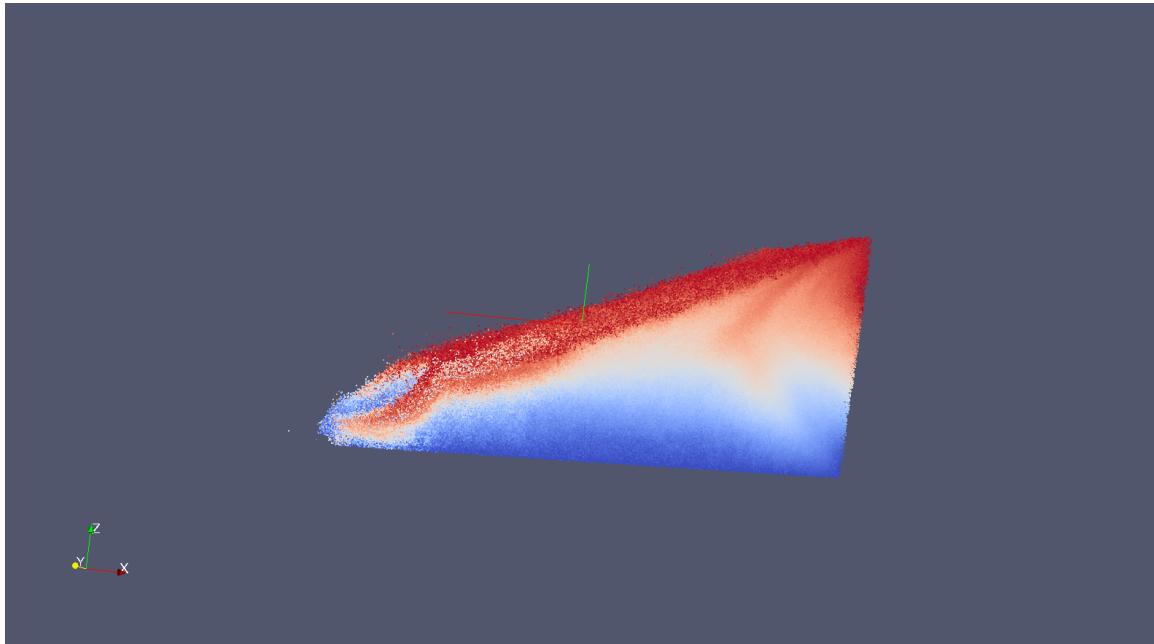
Quantitative Validation of Cylindrical Wheel Results

In the main paper, we only presented qualitative agreement between the images generated by Particle Image Velocimetry and our simulations. In the figure below, we show quantitative agreement between the results. The figure shows the comparison of draw bar pull generated from experimental data at CMU (obtained thru informal collaboration) and that generated by our simulation. As can be seen, the two plots have an identical profile with a peak at -30N and a stable value around -25 and -20N. This demonstrates excellent agreement with the experimental data and validates the simulation. Note that in the experimental data, the sinkage changed in the region between 30-60cm of travel. Hence there is a minor variation between the model and experimental data.



Sloped Terrain

In an effort to quantify mobility on slopes, we have developed granular terrain models for slopes of 10, 15 and 20 degrees. This has been a difficult effort as it has been difficult to get a good compacted slope model. Our simulation results with the slope terrains have generated data that indicate more effort needs to be invested in developing good sloped terrain models. This is currently under continued effort and we aim to resolve this issue shortly.



3D visualization of a sloped terrain. The color coding of the granules indicates granule number, blue being low and red being high. It can be observed that there is no clear boundary and the granules have mixed due the compacting process

Acknowledgement

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors thank Scott Moreland, Krzysztof Skonieczny and David Wettergreen from Carnegie Mellon University for providing the results of their particle imaging studies of wheeled mobility on granular terrain. The authors thank the DARPA for funding this work through the Adaptive Vehicle Make program and BAE systems Minneapolis for leading the primary proposal effort. The authors also thank Daniel Challou, Chris Wentland and Michael McCullough from BAE systems for their technical collaboration.

Simulation of tracked wheels on soft soil substrates

1. Summary

Multiple analysis strategies were applied to the problem of simulating a tracked wheel on soft soils, with the ultimate goal of generating results that could be compiled into Context Models (C2M2L) for the DARPA Advanced Vehicle Make project. These strategies included:

- o Finite Element Analysis
- o FEA using eroding solid elements
- o SPH (Smoothed Particle Hydrodynamics) technology
- o SPH technology utilizing adaptive solid elements

None of the strategies worked perfectly. All of these approaches seem to work reasonably well with firm substrates but until very recently all have encountered significant issues with contact instabilities during the analyses of soft, highly deformable soils. These issues led us to concentrate our efforts on the FEA and SPH strategies while prematurely terminating efforts to implement strategies with eroding solid elements and adaptive solid-SPH elements. Very recently (6-Dec-2012), we began evaluating a fix from LSTC support personnel (new parameter values) that shows great promise in solid element problems but has not corrected the issue in SPH analyses. There, the instabilities affect many (but not all) of the analyses, usually late in the computations. The results from analyses that have completed are encouraging.

Out of the above approaches, SPH still shows the most promise for being able to address the case of soft soils in spite of the contact instabilities. The results from simulations of a simple tracked wheel on several different soil types demonstrated the qualitative differences one would expect between soft and hard materials and demonstrated the qualitative differences one would expect when the analysis was performed on sloped domains and with higher acceleration rates. A substantial increase in SPH mesh density may also be required (with corresponding increase in CPU resources) for the quantitative results necessary to distinguish between different track designs.

A variant of the SPH strategy, the Adaptive SPH technology of LS-DYNA is also, in principle, an ideal way to approach the problem but only preliminary efforts were made to incorporate it before being discouraged by the contact instabilities and turning our attention to other methods. There will still be a need for close interaction/training with LSTC to make effective use of it and tailor the relevant method parameters correctly.

FEA with erosion, with the contact issue under control, appears to be a tractable method but the results will be dependent upon the proper selection of the element failure criteria. The result of using low failure thresholds, while accurate in principle, may result in excessive element removal from the model effectively removing the substrate from the model altogether, compromising the approach significantly.

1.1 Technical Challenges

A significant and persistent issue with contact instabilities during the analyses of soft, highly deformable substrates has affected our ability to complete the evaluation of any of the above solution strategies. The instability manifested itself as the sudden appearance of a very strong, anomalous force in the system, dramatically distorting elements and usually terminating the model at that time. The instability would typically not occur until well into an analysis making the debugging and determination of what was really happening time consuming and tedious. More importantly, as the issue first arose in problems involving erosion (eroding solid and adaptive solid-SPH analyses) it was believed that the problem rose from those methods and not, as later determined by LSTC, from generic contact instabilities. This resulted in a redirection of efforts away from these strategies until very recently.

During the course of the project we have had numerous interactions with LSTC support on the appropriate contact settings to use for both solid-solid contact and SPH-solid contact definitions. Usually the issues were minor and correctable. The suggested fixes have all had the effect of slowing model execution due to reduced time step and more robust contact search algorithms but none of the fixes were able to overcome the instability issue. We have attempted to validate the latest fix from LSTC support personnel (new parameter values) and while the new values give an improvement in our ability to analyze solid element problems it has not corrected the issue in SPH analyses. In fact, they seem to have made things worse in those analyses. Still, we are optimistic that this can be resolved and allow more extensive SPH models to be evaluated in future work.

The choices of appropriate material models and data sets were unspecified at the start of the project. While there is no one model or data set that can represent the wide range of substrates that the tracked wheels could conceivably encounter we believe that we have accumulated an appropriate selection of soil types to test the model rigorously.

The use of eroding solid elements appears to be a viable but risky strategy for analyzing soft soils. One of the keys to a successful erosion model is the proper selection of erosion criteria and parameter values. These parameters and criteria for erosion are not necessarily available for most material models and the selection of parameter values is often *ad hoc*, reducing confidence in the quantitative accuracy of the models (Schwer, 2012). . Also extreme caution needs to be used to ensure that the technique is not excessively removing substrate from the model.

CPU requirements for the relatively coarse mesh models analyzed here have been moderate (generally 1-2 days for SPH models, $\frac{1}{2}$ day for FEA models). However, to increase the SPH mesh density to the level needed to quantitatively resolve different track pad geometries we need to be able to handle models with roughly 100-1000X more nodes. Executing models of this size for extensive work such as Design Of Experiments investigations will require a significant amount of CPU resources.

1.2 General Methodology

Multiple analysis strategies were applied to the problem of simulating a tracked wheel on soft soils. These strategies included:

- o Finite Element Analysis
- o FEA using eroding solid elements
- o SPH (Smoothed Particle Hydrodynamics) technology
- o SPH technology utilizing adaptive solid elements

In considering the interaction of a tracked wheel on soils our main concern was having a capability for dealing with extreme deformation or destruction of the soil with large amounts of the substrate likely thrown clear of the immediate vicinity in such a way that any remeshing approach would likely find itself overwhelmed. For a Lagrangian approach such as Finite Element Analysis this would quickly lead to either termination of the model due to severe mesh distortion or the ignoring of substantial changes in the geometrical domain due to substrate dispersal.

There are analysis methodologies for dealing with situations that encounter the problem of severe mesh distortion that fall into a class referred to as “mesh-free” techniques. One of these techniques, Smoothed Particle Hydrodynamics, (SPH) has been implemented in some mainstream FEA codes (Hallquist, 2006) and has been successfully used in a number of problems involving soils (see, for example, (Kulak & Schwer, 2012) or (Lescoe, 2010)) and should be applicable to our situation. Many of the relevant journal or conference articles using SPH with solid materials are utilizing it in relatively high-speed events where mass dominates over other physical considerations which also would seem to be appropriate for our situation. LS-DYNA, a general-purpose explicit structural analysis finite element code from LSTC was used for all calculations as it has the comprehensive capabilities to address all anticipated analysis techniques (FEA, SPH (Guo, 2010)) and a diverse group of constitutive relations for different soil types. A further advantage of using LS-DYNA is that, being an explicit time-stepping code, the memory requirements are significantly lower than those of an implicit code.

While not all constitutive relations in LS-DYNA are available for SPH materials many of the relevant soil models are. Performing an SPH analysis requires a modest amount of additional information and setup detail, including the specification of an Equation of State to address the pressure-volume relationship in compression.

About half way through the project we became aware of a feature available in more recent versions of LS-DYNA (V971, Rev 5.1.1 or later) that at first glance looks like an even more convenient implementation of SPH technology. In this feature, the substrate areas where interaction with the tracked wheel is anticipated can be meshed with solid elements which are also flagged. When these flagged elements “fail” they are converted to SPH nodes which are then, depending upon the choice of the analyst, free to “couple” or not with the remaining solid elements. Until that time, they behave as regular solid continuum elements. In other words, we should be able to perform a regular Lagrangian analysis when the substrates are strong and firm (presumably more accurate than using SPH elements) but the material will convert to SPH elements under high stress/deformation and allow the analysis to adapt to that situation and carry on. This was also one of the approaches used.

While remeshing of the Lagrangian domain was not an approach implemented in this work we did investigate the use of LS-DYNA's element erosion capability to see if that could be a useful technique. In that approach, elements are deemed to have "failed" when a specified parameter reaches a user-specified threshold. At that time the element is removed from the calculation, creating either voids or new free surfaces. This approach is described in more detail in the following section.

Lastly, primarily as a control, simulations were also performed using a straightforward FE analysis to see just how severe the element distortion problem would be for the different substrate types.

For each of these approaches the goal and initial strategy was the same. A simple validation model was constructed with a simple, tracked wheel on top of a substrate under a gravitational load. After a short time period (0.5-2 sec) for stabilization the wheel would be rotationally accelerated to a modest velocity (\sim 10 MPH center-of-mass) with the torque necessary and actual wheel center-of-mass translational velocity calculated in the program as output quantities. In particular, we wanted to see how far the wheel would compress the soil during the stabilization period. Would the wheel dig into the substrate and displace any material from the model and how that varied for the different soil data sets available were other questions to be answered by the model. As we gained confidence in the models, complexity would be added to the analysis. First, additional soil models and data sets would be analyzed and later, actual track designs would be used in the simulation as well as more demanding types of analyses (sloped domains, braking tests). Then it would be possible to start extracting overall performance metrics from the models and to start to build a database of performance characteristics for the C2M2L Context models.

1.2.1 Details of the Calculations

A wide variety of constitutive models in LS-DYNA have been used in SPH simulations of soils. One of the simpler models, Material Type 5 (Soil and Foam) has been used for the vast majority of the calculations here. In addition to its simplicity this material type has multiple data sets available for appropriate substrates. The data sets available from all constitutive models are summarized in Section 1.7 as an Annex but the majority of the work done here was done using Material Type 5, data sets 4-6 and 8 which are soils or sands ranging from soft to firm. The most recent sets used are from a study (Chitty, 2011) of sand and soil types at the Kennedy Space Center that were explicitly characterized with LS-DYNA material Type 5 in mind. In the results, specific mention is made as to which material model and data set were used.

A further advantage of the Type 5 material model is that there is an associated model in LS-DYNA (Type 14, Soil and Foam Failure) which uses the same data as Type 5 but allows LS-DYNA to make use of the Pressure Cutoff value (point at which the material can no longer hold tension) as a failure criterion for erosion. This was used as part of the work in implementing element erosion as was the use of the *MAT_ADD_EROSION command, where LS-DYNA allows one to define their own erosion criteria. This latter approach was used with a trigger value of -3.4475 KPa for MNPRES (minimum pressure) which corresponds to the PC (pressure cutoff) parameter in Material Type 5 for most data sets.

An equation of state is required in SPH analyses to address the pressure-volume relationship in compression. For the Type 5 or 14 materials this is addressed by the P vs. ln (vol. strain) data. For other constitutive models a specific EOS must be chosen from the

number of models available in LS-DYNA. Most of the EOS models in LS-DYNA are designed to be compatible with shock waves. Due to the relatively low-speed nature of track/substrate interactions (very low Mach number) the simplest form of EOS is all that is required. For those constitutive models that required an EOS we used the simplest representation (Linear Polynomial) with only non-zero coefficient $C_2=K$ (bulk modulus).

Friction coefficients of 0.3 were used for both static and dynamic values, representing a typical value between soil and metal (Tekeste, Raper, Tollner, & Way, 2005).

The units used in all of the calculations were all in the mm-mN-kg-sec system. A simple tracked wheel design was created (Figure 1) for use in this study with properties (steel, rubber) taken from the I-DEAS library:

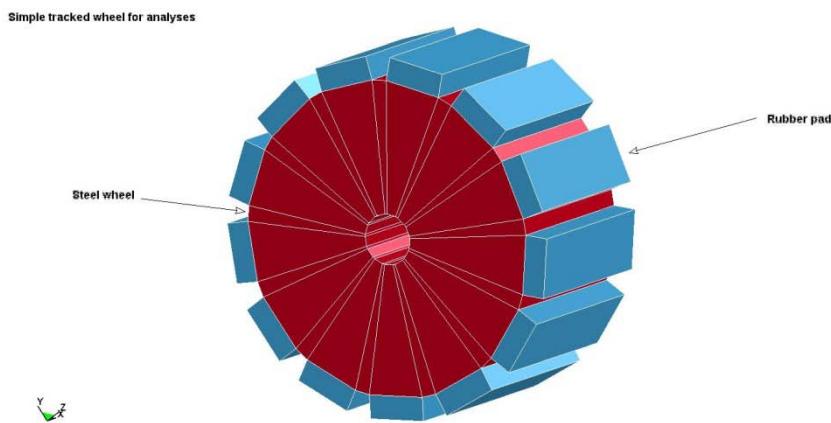


Figure 1 – Tracked steel wheel with 13 equally spaced rectangular rubber pads

The wheel has an overall diameter of roughly 726 mm with the pads measuring 338x105x75 mm. A 101.6 mm section at the center of the wheel was removed. In LS-DYNA, this volume is replaced by a rigid cylindrical body (also of steel) which is used to drive the wheel via a nodal constraint (rotational velocity and load curve). Nodes at the boundary between the rigid body and the wheel deliver the total X-moment needed to accelerate and drive the wheel.

While not an actual tracked wheel design this initial geometry would still be capable of addressing the relevant physics of the interaction between soft soils and hard geometries that can bite into the substrate. A very crude mesh was used to represent the wheel and the pads as these entities were expected to act as semi-rigid bodies with respect to the substrates.

A number of different mesh densities and geometrical designs for the substrate domain were used in the course of the project. As an example of one of the later constructions, Figure 2 shows the wheel in proximity to the substrate which has the upper 500 mm of its 2x2x1 m domain comprised of SPH elements. Note that the size of the SPH elements is only for visualization purposes and that the intended direction of travel for the wheel is in the negative-Y direction, with gravity acting in the negative-Z direction. In all models, the wheel has been placed as close as possible to top surface of substrate (<1 mm).

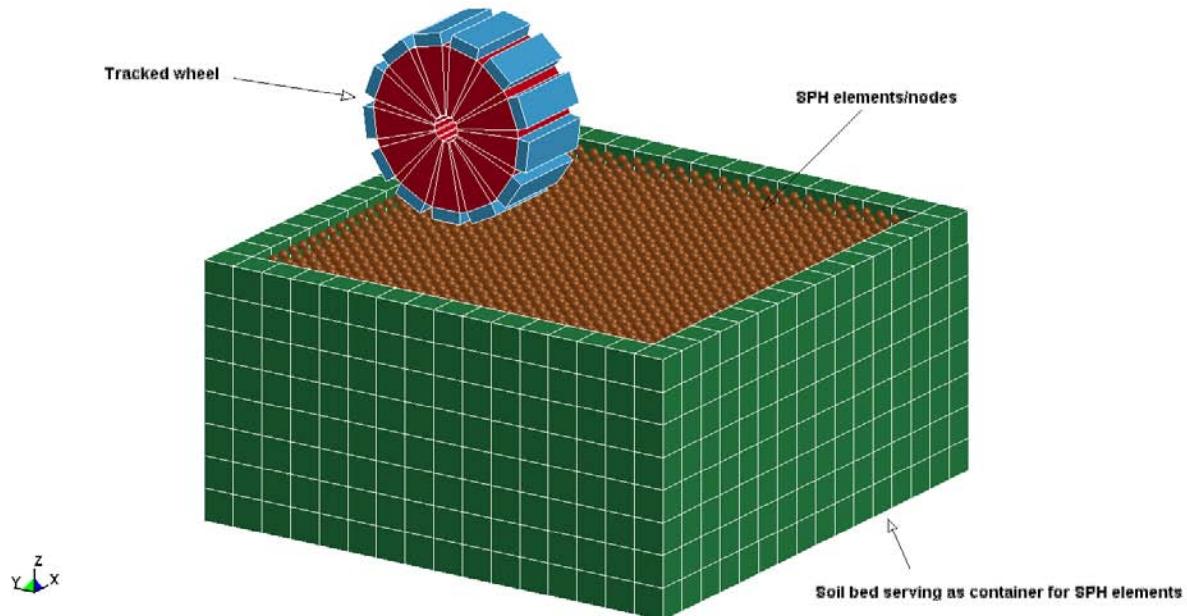


Figure 2 – Tracked wheel with on SPH substrate

One feature of the mesh is that it is relatively coarse in the SPH region (despite having over 6K nodes in the above model) as shown below in Figure 3:

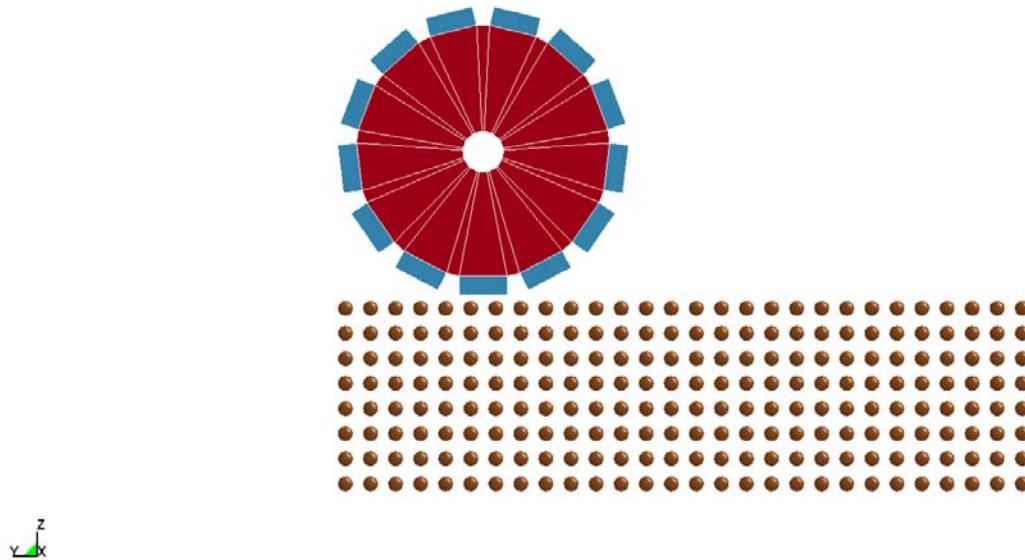


Figure 3 – Tracked wheel with on SPH substrate

With an SPH nodal spacing somewhat greater than the spacing between the pads we did not expect that this mesh could quantitatively distinguish between track pad designs without significant refinement. However, for our initial purposes we believe that the mesh density in all areas is adequate while still allowing for reasonable model computation times (~24-48 hrs for an SPH analysis; less for solid element FEA) although a proper mesh convergence study on these analyses has not yet been performed.

Boundary conditions are straightforward. The nodes on the bottom of the substrate (lowest Z-values) were clamped in all 3 principal directions. The sides were constrained to prevent any normal motion and gravity was applied from the beginning in the negative Z-direction. The wheel is constrained from moving in the X-direction but is allowed to move freely in the other two directions.

The analyses begin with a stabilization period (0.5-2 sec) to allow the system to adjust to the gravitational load and, particularly for soft soil models, to settle as needed. After this period the rotational velocity of the wheel was linearly accelerated (using the rigid body) so that the desired wheel translational velocity would be reached in the desired time. These accelerations corresponded to:

Final angular velocity	Center of mass velocity	Time for acceleration
12.3 rad/sec	4470 mm/sec (10 MPH)	1.6 sec
30.785 rad/sec	11176 mm/sec (25 MPH)	4.0 sec

Given the small geometrical size of the model all the analyses save one were done using the lower velocity and even there it was assumed that the wheel would rotate past the end of the domain before reaching top speed. This was considered acceptable as in these first calculations we were most interested in the qualitative differences between models in the initial phase of the acceleration and in keeping the size of the domain (and CPU execution times) manageable.

1.3 Technical Results

Finite element analyses were performed on several different substrates to see how a mainstream technique would fare with these soils. For the firmer materials, the results were reasonable and show qualitative differences in behavior between the materials. Figure 4 shows the permanent deformation and plastic strain left by the wheel travelling across the surface of sand substrate (Kennedy Space Center dry sand, material type 5/data set 8):

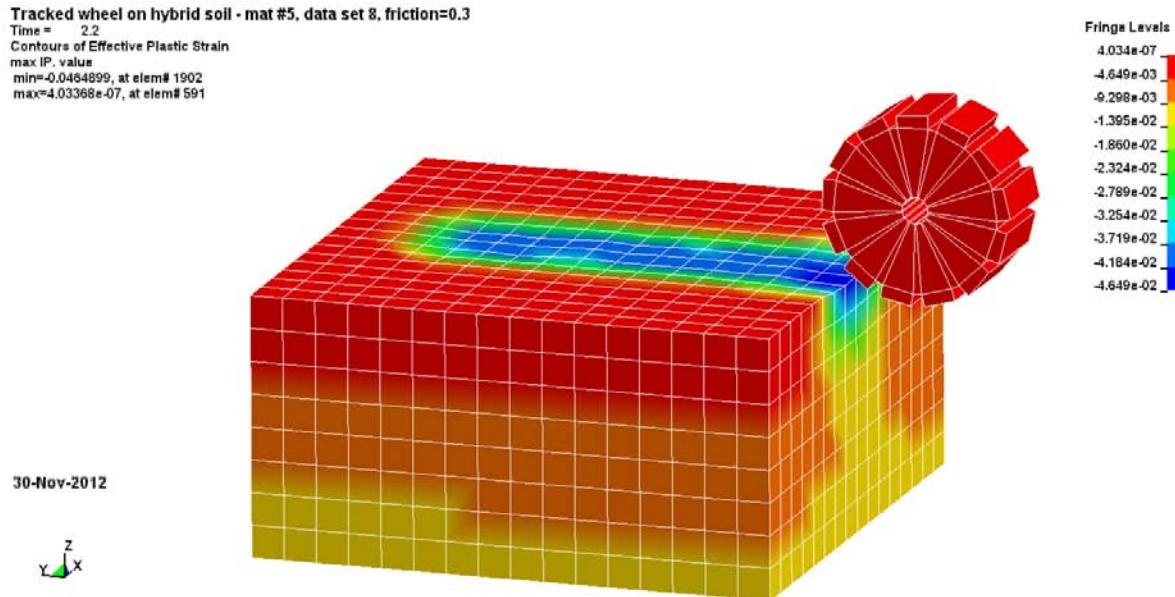


Figure 4 – Plastic strain in KSC dry sand after tracked wheel passage

There is minimal disturbance of the substrate. The next two images (Figures 5a, 5b) show the X-moment (torque) on the wheel and the translational velocity of the wheel center as functions of time:

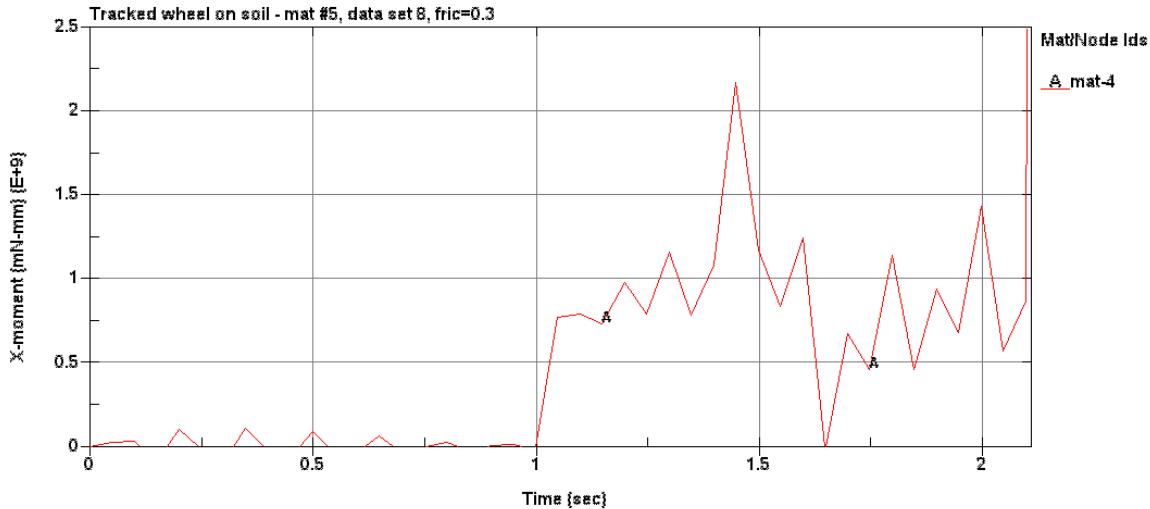


Figure 5a – X-moment on wheel hub

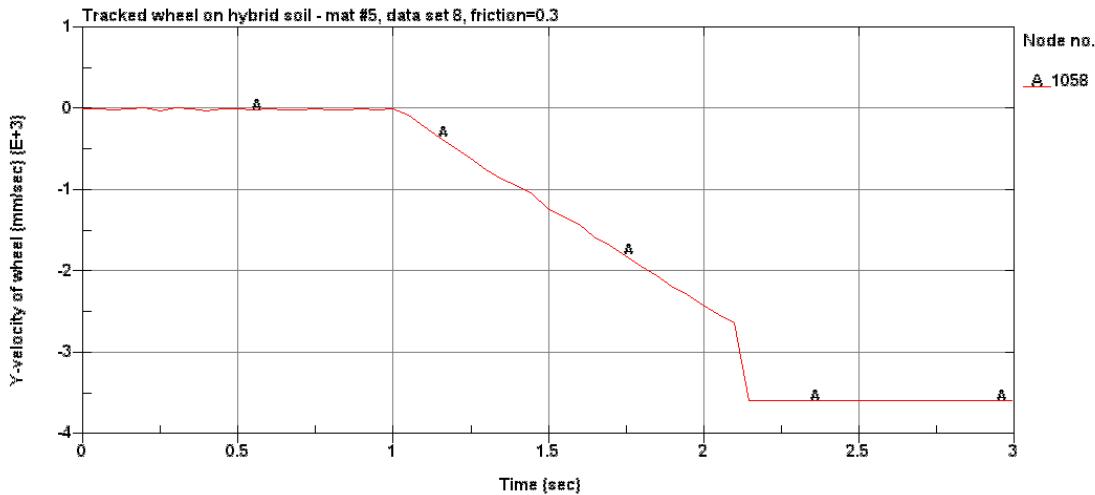


Figure 5b – Y-velocity of wheel

The results for the X-moment are quite noisy, possibly as a result of the track pads impacting the surface. The overall level of torque (roughly 1.E9 mN-mm) is about a factor of 3 greater than the torque required to accelerate the wheel in the absence of any wheel-substrate interaction ($3.32\text{E}8$ mN-mm) which seems reasonable. The velocity graph shows a steady acceleration towards the target of 4470 mm/sec until the wheel hits the edge of the domain. It is apparent that the wheel would not have achieved that target value by the end of the acceleration period (1-2.6 sec). This is due to the friction coefficients being set to 0.3, reducing the effective translational acceleration. The same model using a moist, dense sand (Mason sand, 15% moisture, Material Type 5 data set 9) as the substrate shows a very similar pattern for plastic strain in Figure 6:

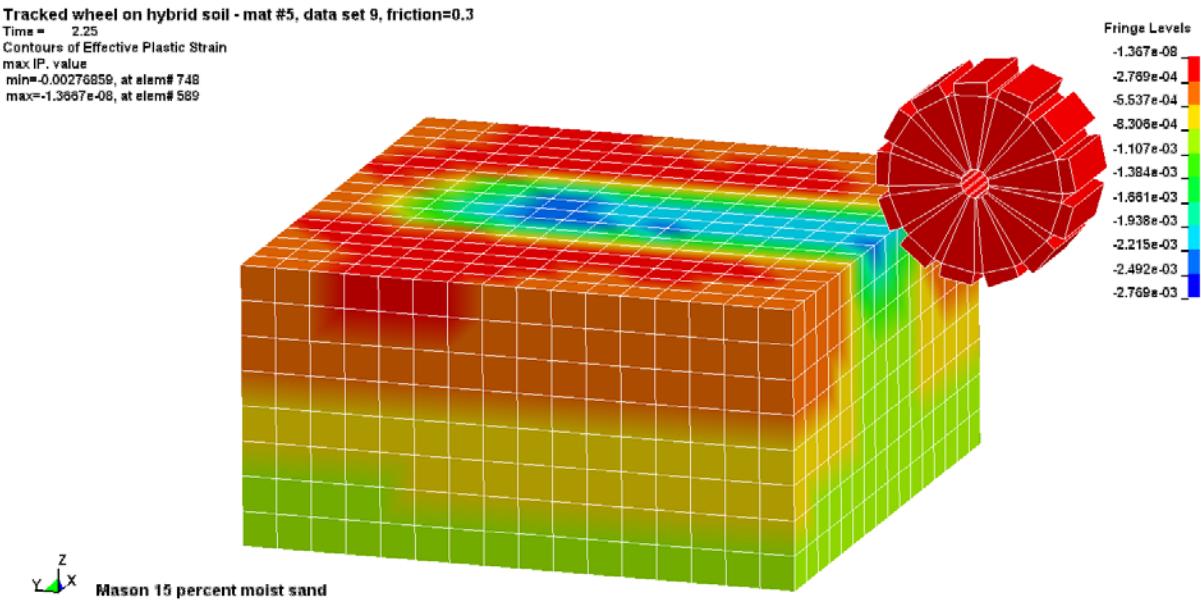


Figure 6 – Plastic strain in Mason 15% moist sand

There is a tiny amount of plastic strain in the sand; the highest amount where one of the pads impacted the surface during rotation. Plots for X-moment and Y-velocity are similar to the previous results:

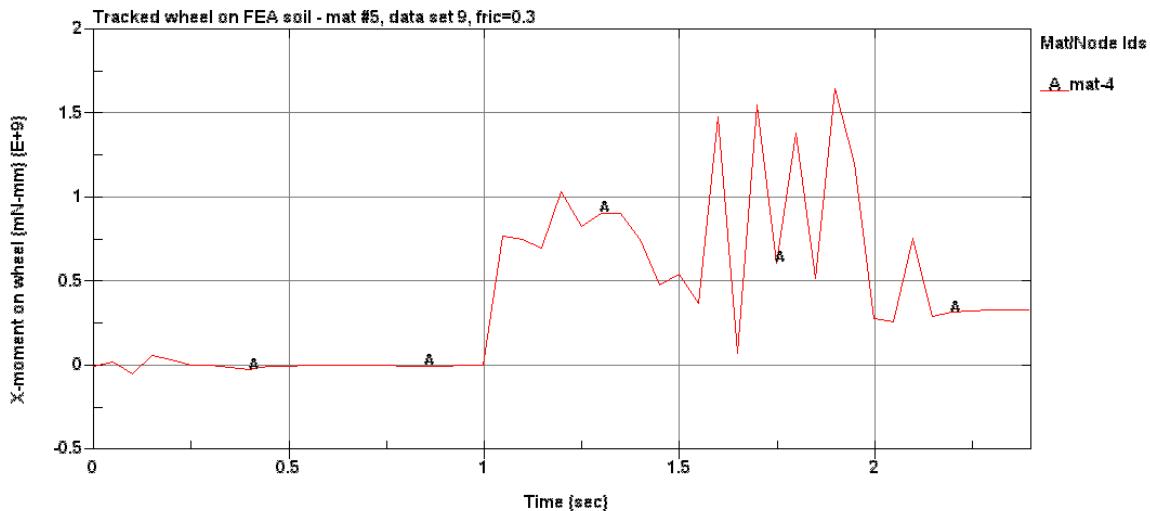


Figure 7a – X-moment on wheel hub

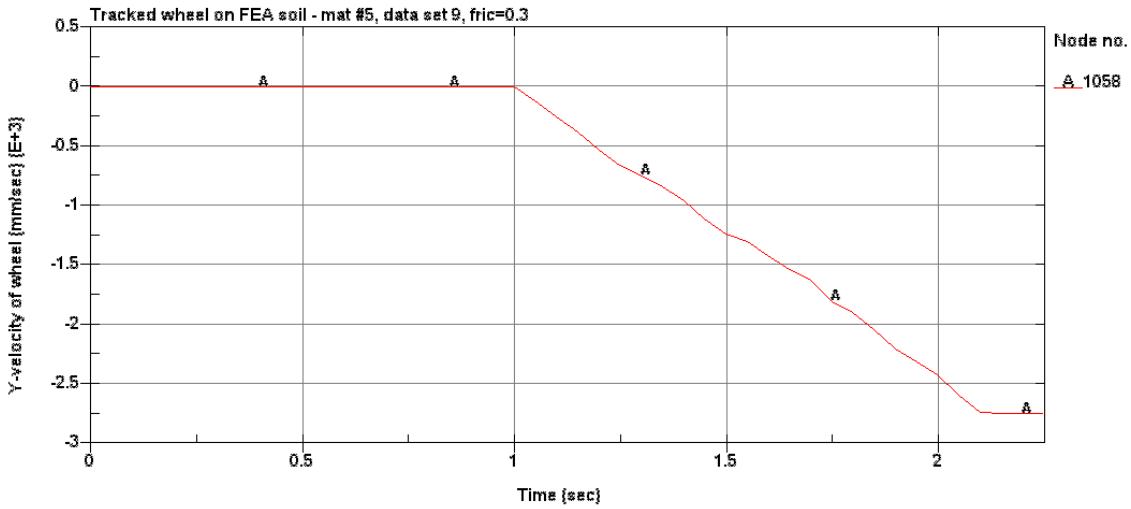


Figure 7b – Y-velocity of wheel

The quality of the results changes when soft substrates are analyzed. Figure 7c shows the deformed geometry from an analysis of a soft soil (“Fasanella soft soil”):

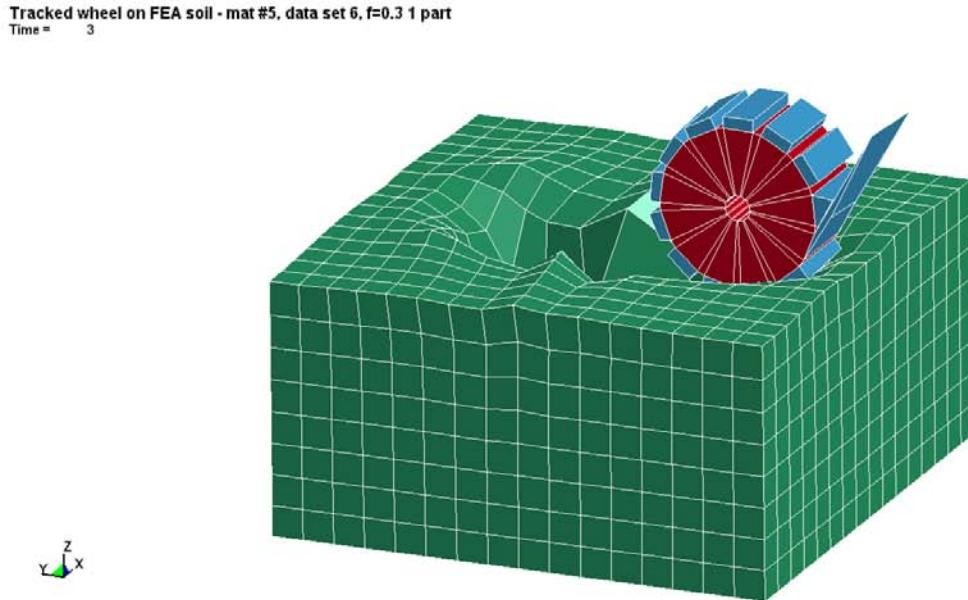


Figure 7c – Deformed geometry during analysis of Material Type 5, data set 6 (soft soil)

Here, an anomalously large force suddenly manifested itself in the system with severe deformation of some soil elements as well as one of the track pads. This was determined by LSTC support to be a contact instability. Utilizing the new Contact parameter set the final deformed geometry is much more reasonable:

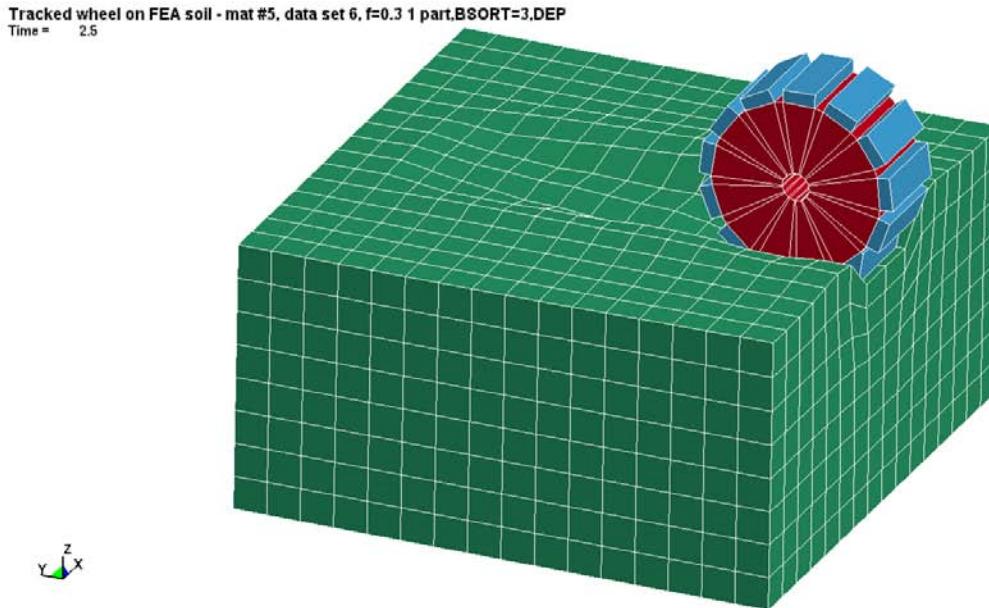


Figure 7d – Deformed geometry during analysis of Material Type 5, set 6 with new parameters
In general, the substrate deformations observed are not severe for the above mesh density, constitutive models and data sets but we anticipate some situations where there will be much more deformation than that shown above. Also, the technique and material models used above do not allow for element failure. In the meantime, using a straightforward FEA analysis we appear to have a viable method for analyzing the interactions between tracked wheels and firm substrates.

As an enhancement to the capabilities of a straightforward FEA model we made use of the eroding capability in LS-DYNA, via the *MAT_ADD_EROSION command. This command allows for the specification of multiple failure criteria for a designated material model. There are some 7 choices; at the suggestion of LSTC support personnel we eventually selected the MNPRES (minimum pressure at failure) parameter as being the closest in physical nature to the PC parameter in the Type 5 (Soil and Foam) material model.

Significant difficulties were encountered in trying to implement this approach. The most significant issue was the contact instability issue with the sudden appearance of anomalous forces in the model which caused tremendous deformation of the mesh and often caused the heavy steel wheel to be propelled backwards and upwards, as was seen during some of the non-eroding FEA analyses. Despite working with LSTC support to try and correct this behavior we were never able to eliminate it until very recently and that affected how we perceived this method in our search for a robust analysis technique. An example of the phenomena is shown in Figures 8a-d where the model is shown over a sequence of output steps. After digging a trench in the material due to erosion (using the MNPRES variable and a setting of -3.4475 KPa, the PC value from the data set) the wheel is stopped in its tracks and one of the rubber track pads is massively distorted:

Tracked wheel on hybrid soil - mat #5, set 4 (erodible), friction=0.3
Time = 1.48

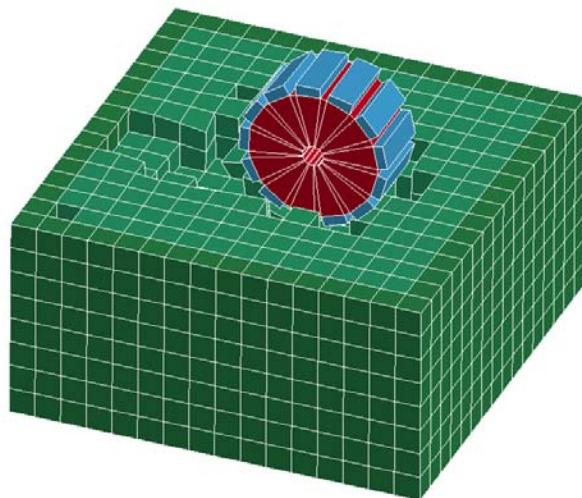


Figure 8a

Tracked wheel on hybrid soil - mat #5, set 4 (erodible), friction=0.3
Time = 1.48

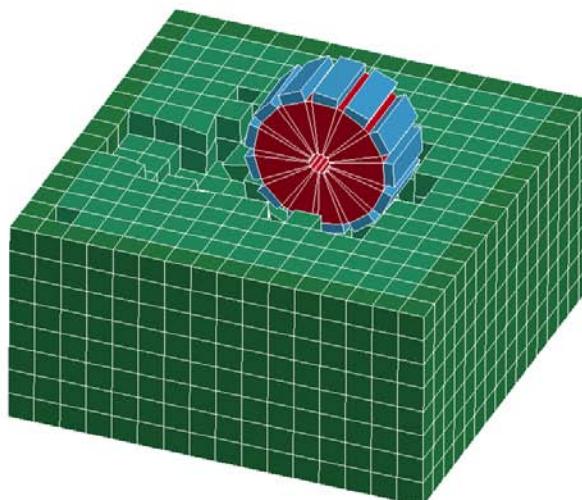


Figure 8b

Tracked wheel on hybrid soil - mat #5, set 4 (erodible), friction=0.3
Time = 1.5

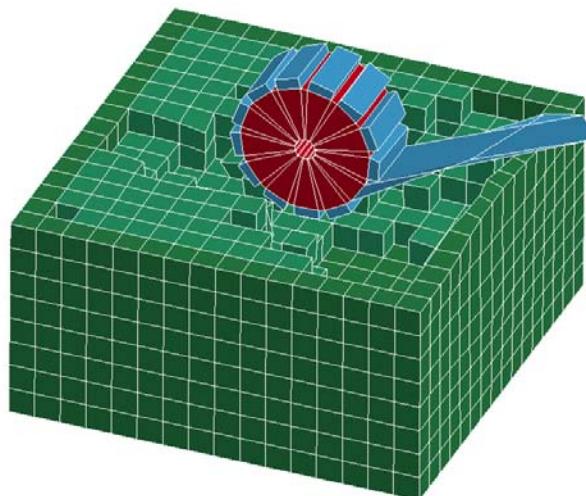


Figure 8c

Tracked wheel on hybrid soil - mat #5, set 4 (erodible), friction=0.3
Time = 1.52

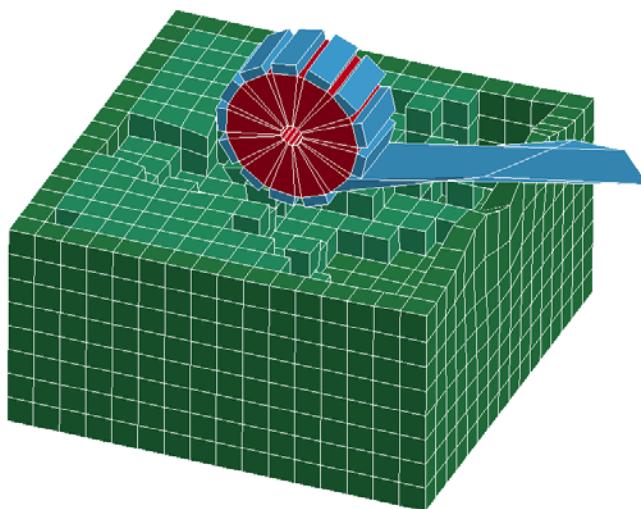


Figure 8d – Eroding contact model using Material Type 5, data set 4 (loose clay sand)

This occurred with the majority of eroding analyses where any erosion occurs and resulted in our not pursuing this technique aggressively.

After receiving the new Contact parameters mentioned previously we re-ran the above model and the results (Figure 8e) looks much more reasonable:

Tracked wheel on erodible soil-mat #5, set 4, fric=0.3, SOFT=2,BSORT3,DEP
Time = 2

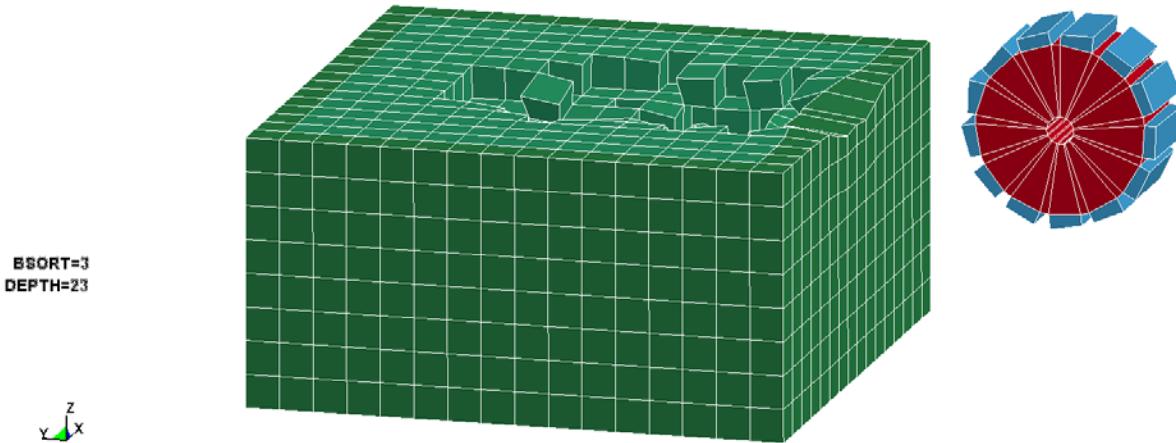


Figure 8e - Eroding contact model using Material Type 5, data set 4 (loose clay sand) and new contact parameters

The instabilities appear to have been avoided in this model and we are optimistic that this approach can now be pursued more vigorously.

While the contact instability issue affected many of the models run with soft substrates, we were occasionally lucky enough to avoid the effect and obtained interesting results. Figure 9 below shows the aftermath of running a simulation of Material Type 5, data set 8 (KSC dry sand) where the wheel has run passed through the entire domain leaving many eroded elements and substantial mesh deformation in its wake:

Tracked wheel on erodible soil - mat #5, data set 8, friction=0.3
Time = 3

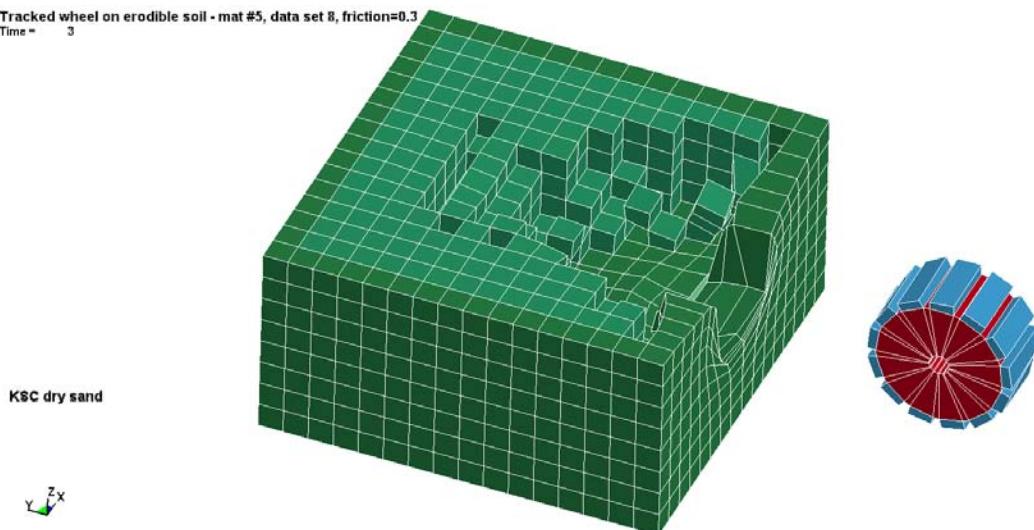


Figure 9 – Deformation after erodible contact model using KSC dry sand

This model was also re-run with the new parameter values and the result looks very much like the above:

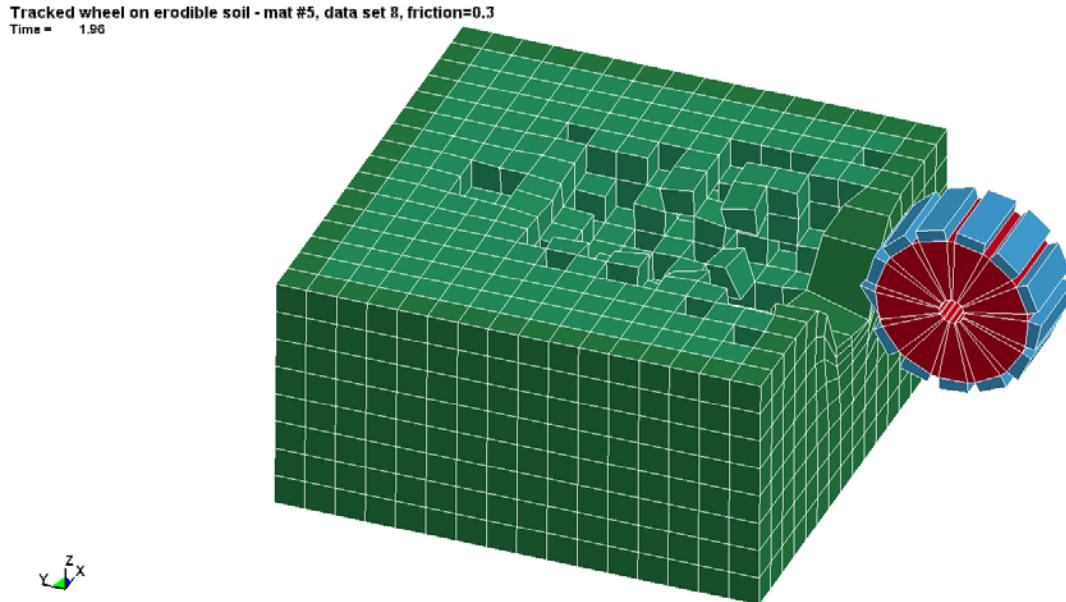
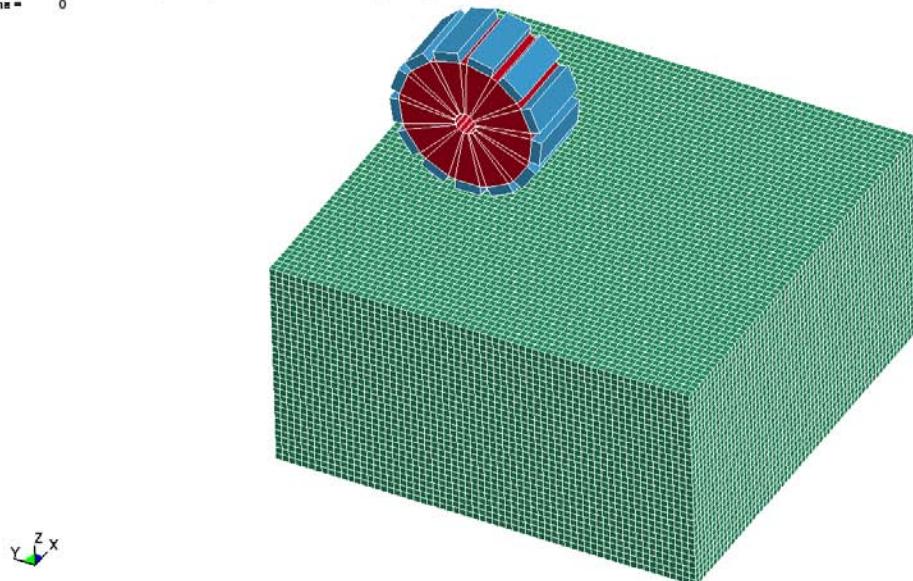


Figure 9a – Deformation after erodible contact model using KSC dry sand and new Contact parameters

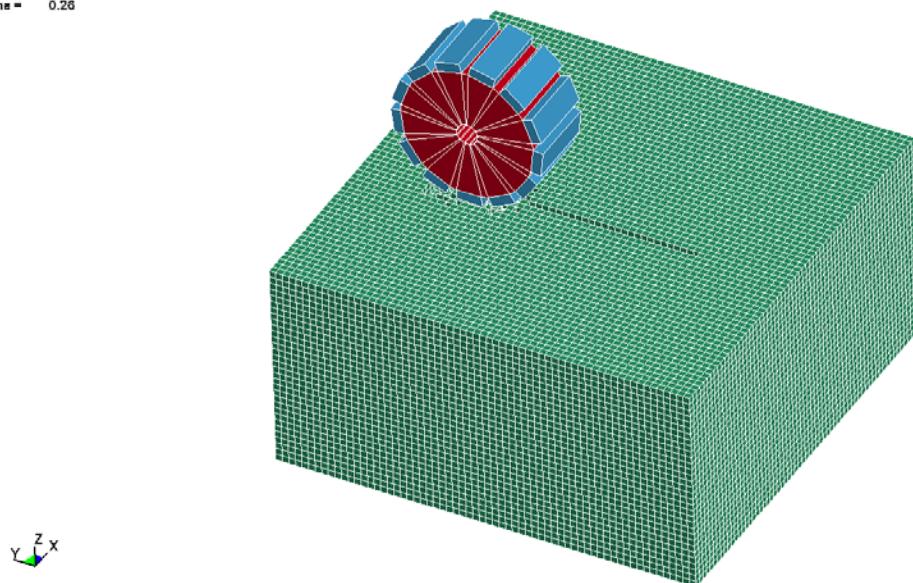
Poor quality results, like those in Figures 8a-d, were obtained with other material models as well. As time was growing short on the project, this approach was put on hold. One suggestion the LSTC support personnel had made that was only just implemented was to significantly refine the erodible mesh. This suggestion made sense as each element that erodes in the above mesh has a significant effect upon the geometry, although a significantly finer mesh would have a correspondingly greater demand for CPU resources. The above model was re-run using a substantially refined mesh (4X increase in mesh density in each direction). Below, a sequence of images showing the movement of the wheel and erosion of elements highlights one of the shortcomings of the erosion technique; as the elements fail and are removed from the mesh, their locations become holes in the model and no longer resist the motion of the wheel in any way. In reality, even a totally fractured solid would still have mass that would resist the impact of the wheel. With this effect missing, the wheel simply sinks out of sight into and through the substrate as all the elements ahead of it have been determined to have failed and have been removed from the model.

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,B3J
Time = 0



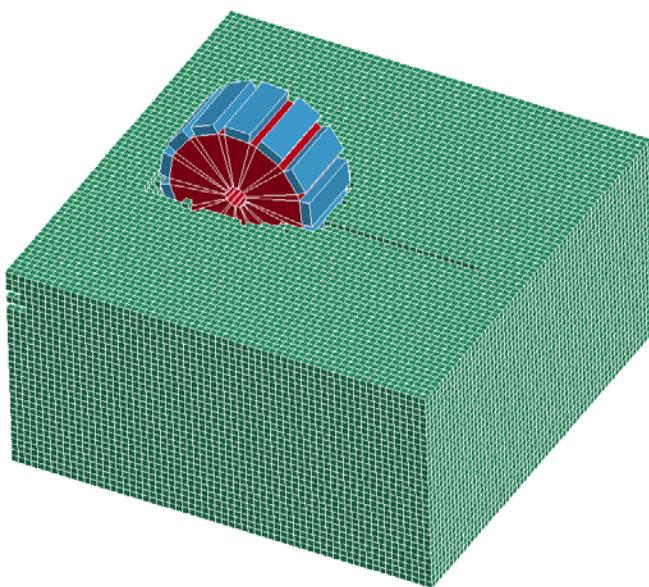
Figures 9b – Wheel on eroding soil (Material type 5, data set 8) at beginning of model

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,B3J
Time = 0.26



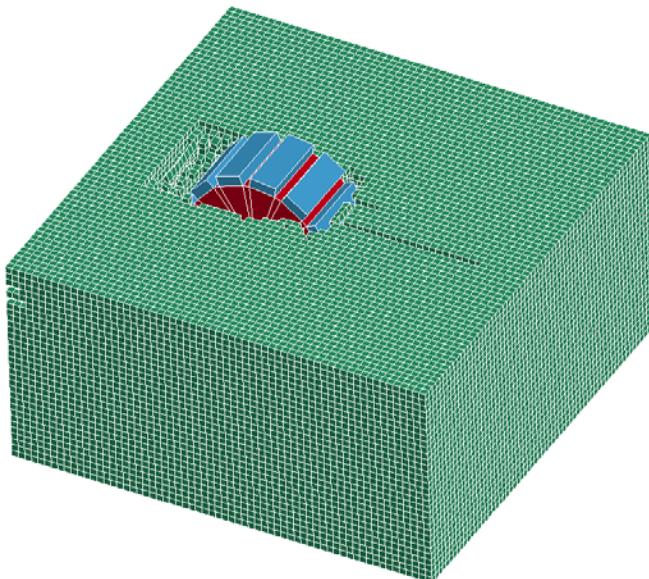
Figures 9c – Wheel on eroding soil (Material type 5, data set 8) at end of stabilization period

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,B\$3
Time = 0.7



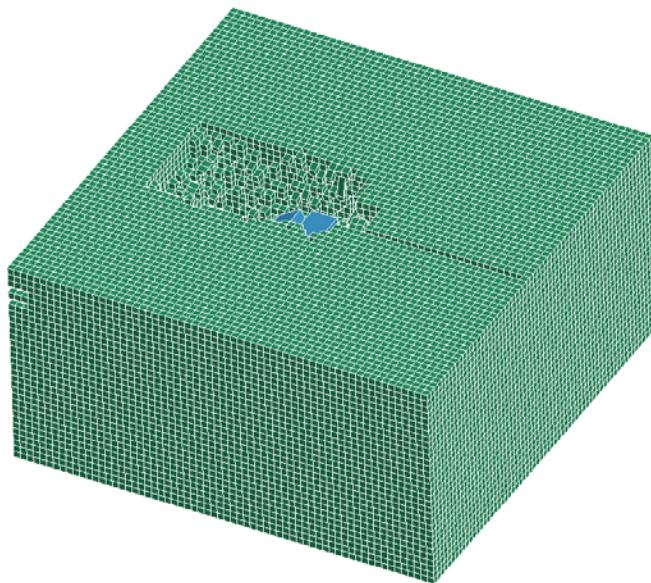
Figures 9d – Wheel on eroding soil (Material type 5, data set 8) begins rotating and digging into soil

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,B\$3
Time = 0.84



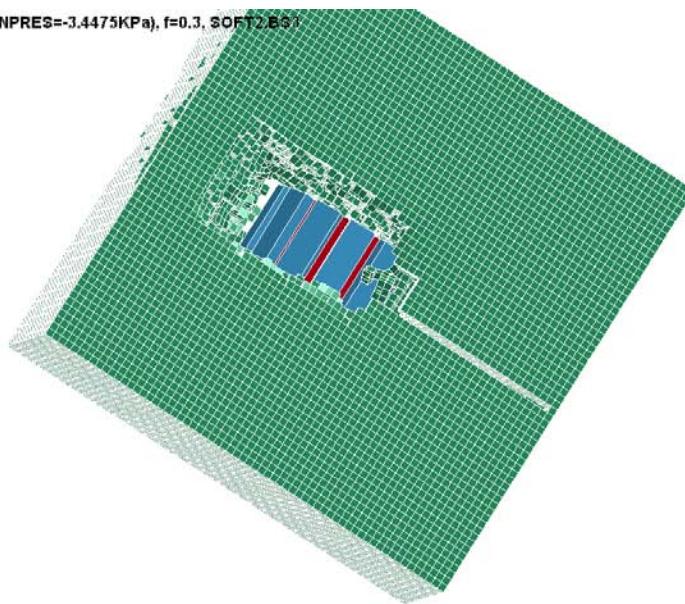
Figures 9e – Wheel on eroding soil (Material type 5, data set 8) digs further into soil

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,BG3
Time = 1



Figures 9f – Wheel on eroding soil (Material type 5, data set 8) sinks out of sight into soil

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,BG3
Time = 1.08

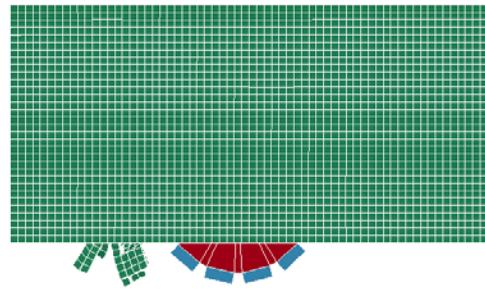


Figures 9g – Wheel on eroding soil (Material type 5, data set 8) sinks through past bottom of model

All erodible soil-mat #5, set 8 (MNPRES=-3.4475KPa), f=0.3, SOFT2,B3

Time = 1.08





Figures 9h – Wheel on eroding soil (Material type 5, data set 8) sinks through past bottom of model

While this effect may not happen with all soil models it is definitely a shortcoming of the approach that needs to be considered for models with low failure thresholds.

Our attempts to make use of the Adaptive-solid/SPH capability in LS-DYNA also did not yield reasonable results and suffered from the same contact instabilities as the eroding contact approach. A key concept in this technique is that, as in erodible solids, the conversion of solid elements to SPH elements takes place after an element has “failed”, with the failure criteria to be decided by the user. Thus the problem is set up in much the same way as the eroding contact approach, but here the elements get converted to SPH nodes rather than disappear from the analysis and so should be more physically realistic.

Unfortunately, our issues with contact instabilities left us unable to make the approach function as we think it should. In Figure 10 the deformations of a model using a Type 5 material (data set 6, Fasanella “Soft Soil”) shows how the SPH and solid meshes have been grossly distorted near the end of the 2 sec stabilization period. Elements had started to erode at about 1.80 sec and by 2.0 seconds an anomalous, massive force appeared in the model:

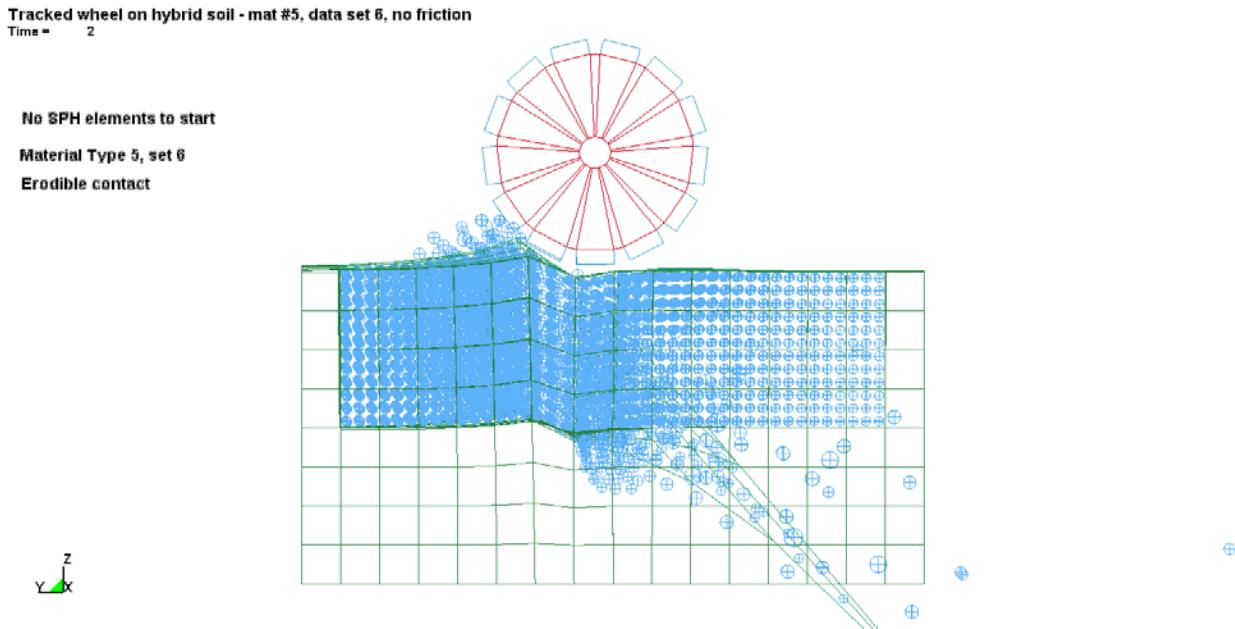


Figure 10 – Adaptive Solid to SPH model using Material Type 5, data set 6

The new LSTC Contact parameters may help correct this behavior but we have not had a chance to evaluate them in this method. Due to our inability to find a solution in a timely fashion, this approach was abandoned in favor of pursuing the SPH approach full time.

One characteristic of the SPH models is that there is almost always a little more deformation and settling due to gravity and the weight of the steel wheel than seen in a solid-element FEA mesh. Given the hydrodynamic character of the analysis this is perhaps to be expected. With an appropriate selection of the SST contact parameter, though, the issue appears to be a negligible one.

Figures 11a-c show the wheel and SPH mesh at the beginning, middle and end of an analysis using a dense, moist sand (Material Type 14, data set 9):

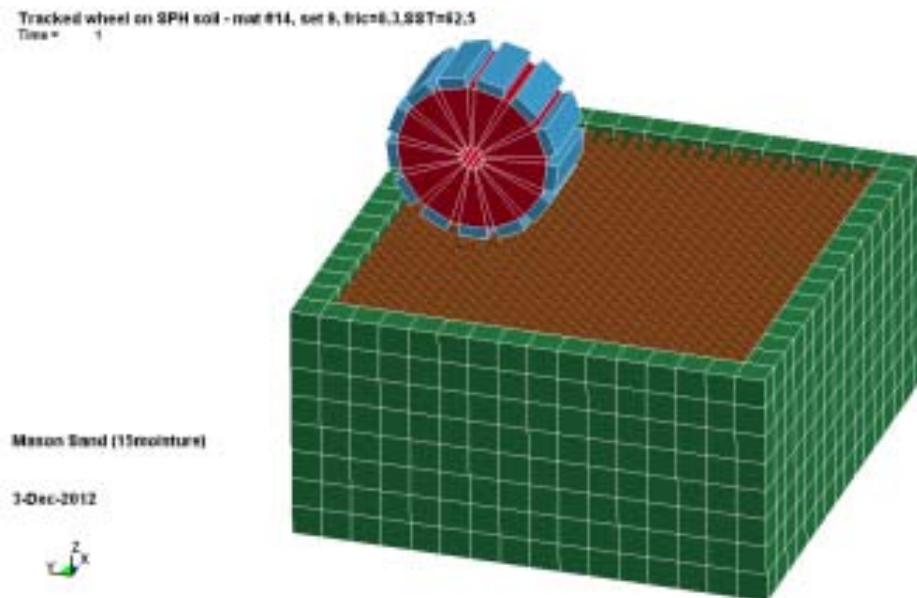


Figure 11a - Wheel at beginning of SPH analysis of Mason sand (15% moisture)

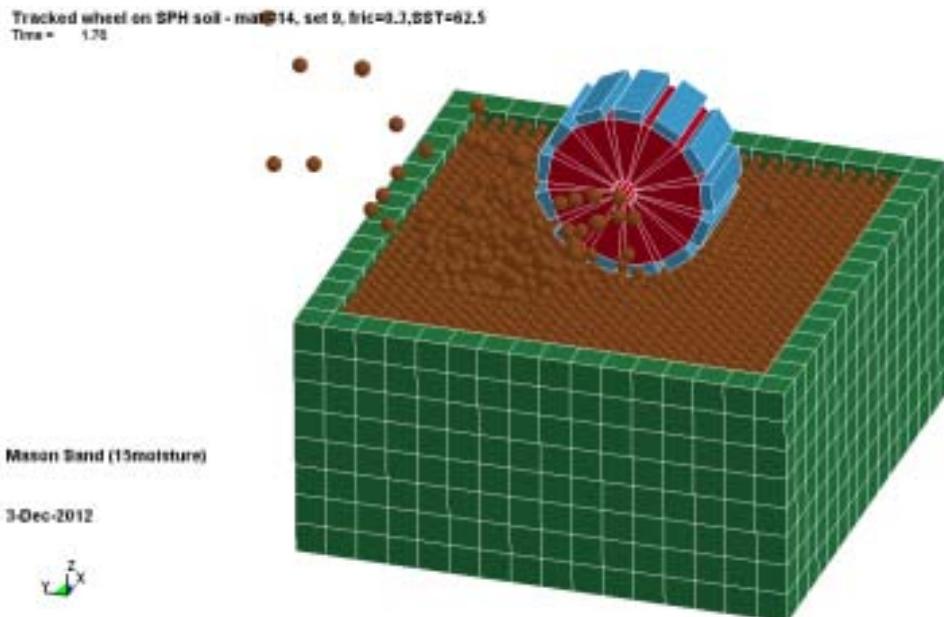


Figure 11b - Wheel at middle of SPH analysis of Mason sand (15% moisture)

Tracked wheel on SPH soil - mat #14, set 9, fric=0.3, SST=62.5
Time = 2.5

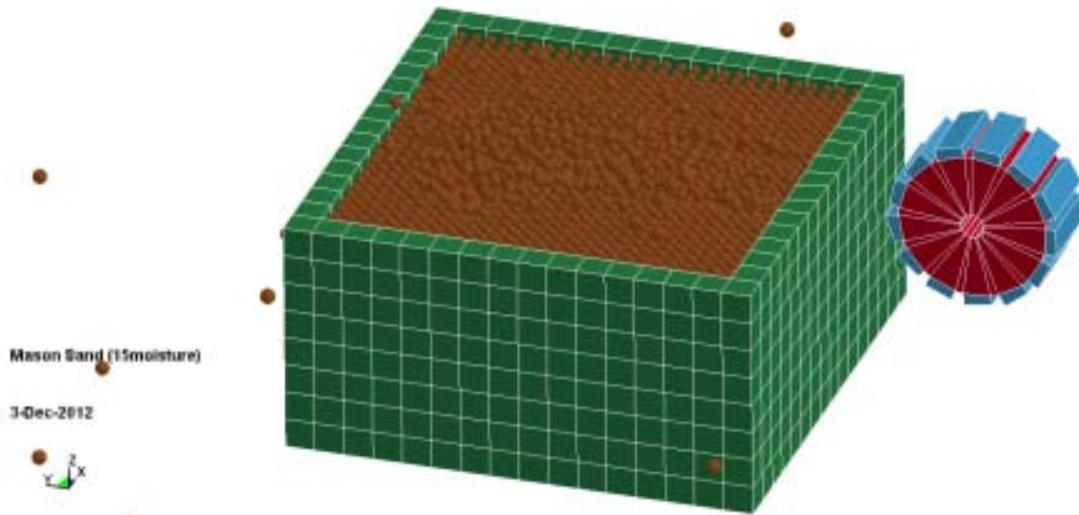


Figure 11c – Wheel at end of SPH analysis of Mason sand (15% moisture)

Here the model behaves as expected with SPH particles being thrown out of the domain as the wheel accelerates and eventually rolls off the end of the domain. Both the plastic strain and the von Mises stress in the substrate at the end of the analysis (Figures 12a-b) have reasonable values:

Tracked wheel on SPH soil - mat #14, set 9, fric=0.3, SST=62.5
Time = 2.5
Contours of Effective Plastic Strain
max IP. value
min=-0.122244, at node# 4691
max=-1.04905e-05, at node# 6075



Mason Sand (15moisture)

3-Dec-2012

$\begin{matrix} z \\ Y \\ X \end{matrix}$

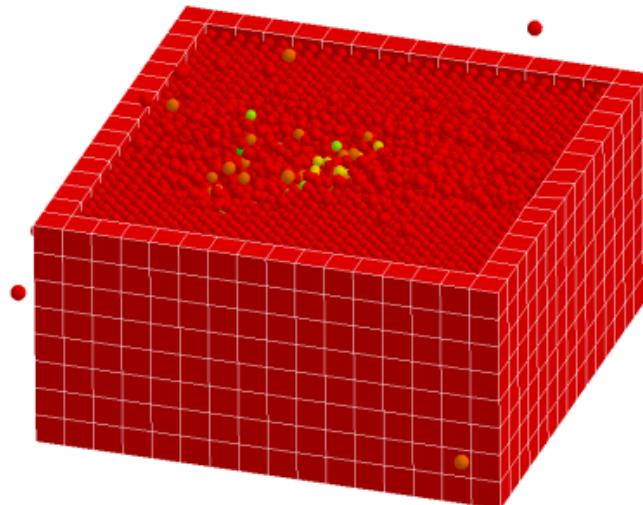


Figure 12a – Plastic strain in substrate (Mason sand, 15% moisture) after analysis

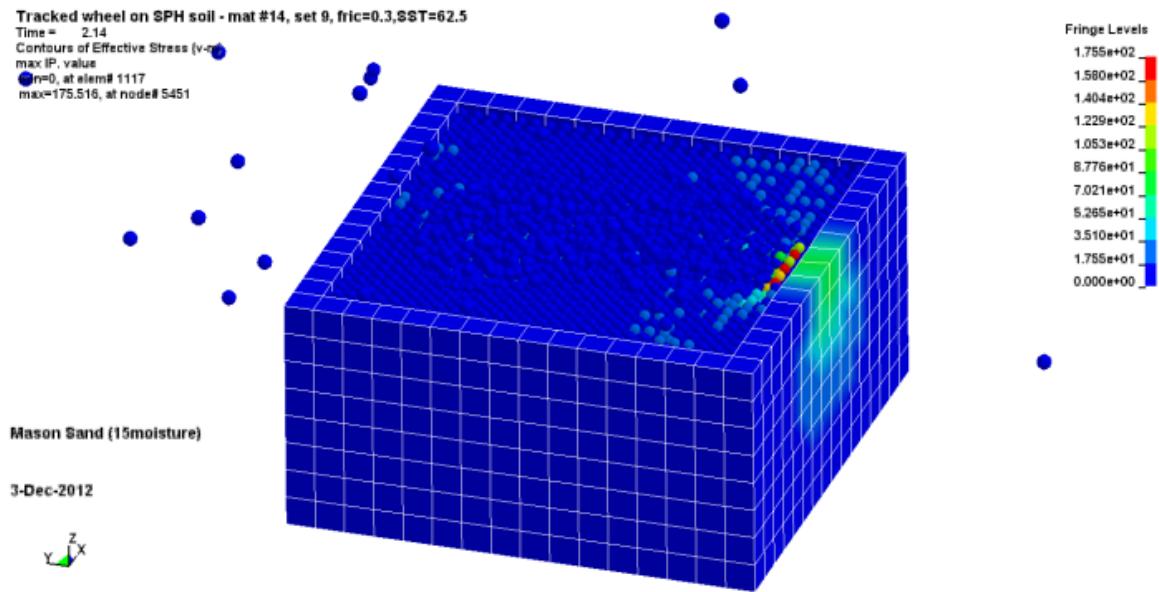


Figure 12b –von Mises stress in substrate (Mason sand, 15% moisture) after analysis

The X-moment and Y-velocity of the wheel are similar to the results from an FEA analysis:

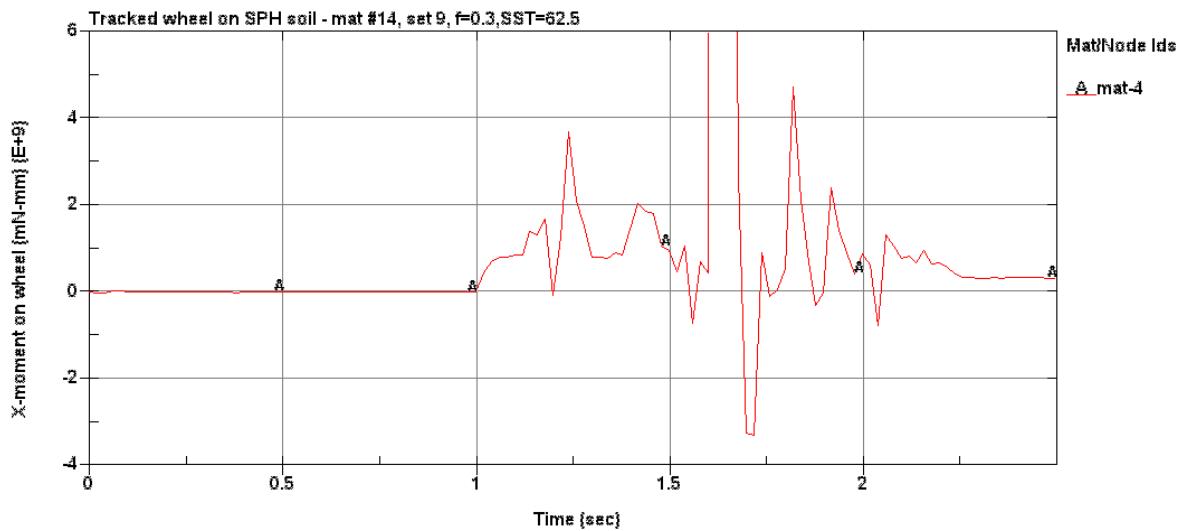


Figure 13a – X-moment of wheel during SPH analysis of Mason sand (15% moisture)

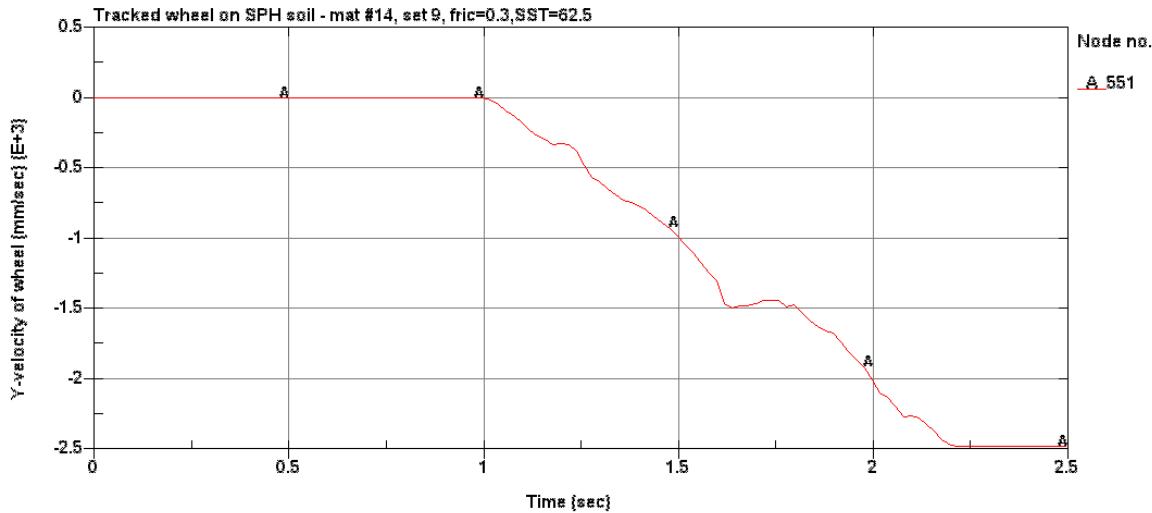


Figure 13b –Y velocity of wheel during SPH analysis of Mason sand (15% moisture)

There is a notable spike in X-moment at about 1.6 sec and corresponding bump in velocity in the above results but the cause is unknown. In general, the X-moment results from SPH analyses are even noisier than those from the FEA runs.

An identical analysis was run using the soil model for Cuddeback A, a dry lakebed soil (Material Type 14, data set 14). This harder material showed lower levels of deformation and correspondingly lower amounts of plastic strain after the wheel had passed. X-moment results were noisier while Y-velocity results were comparable to those obtained for the Mason 15% sand:

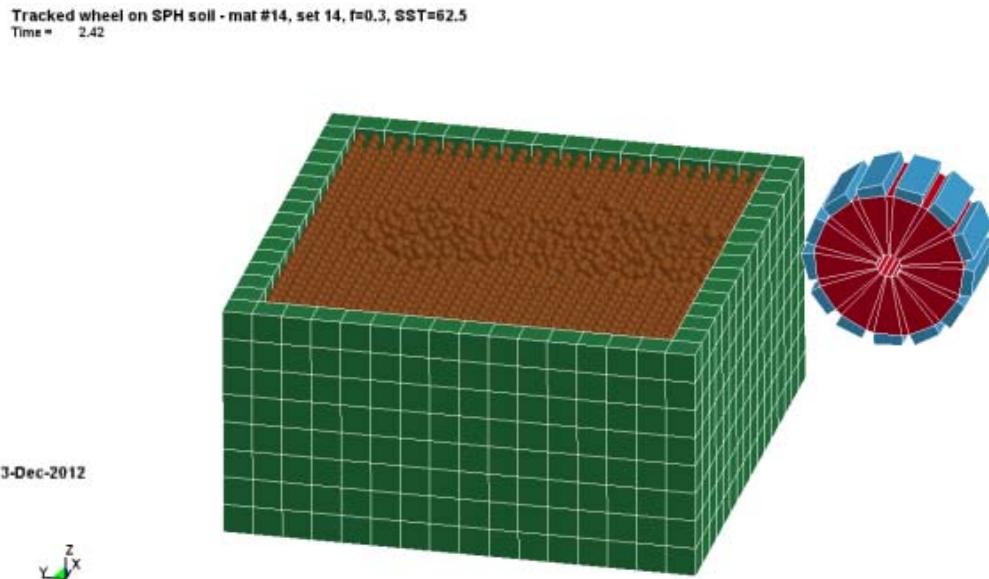


Figure 14a – Deformation after SPH analysis of Cuddeback A soil

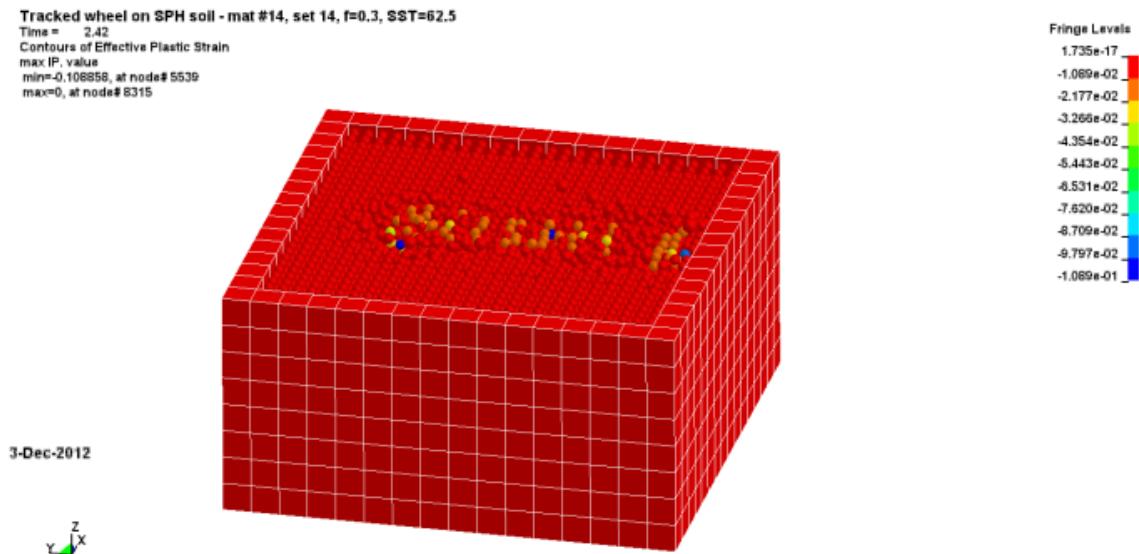


Figure 14b –Plastic strain after SPH analysis of Cuddeback A soil

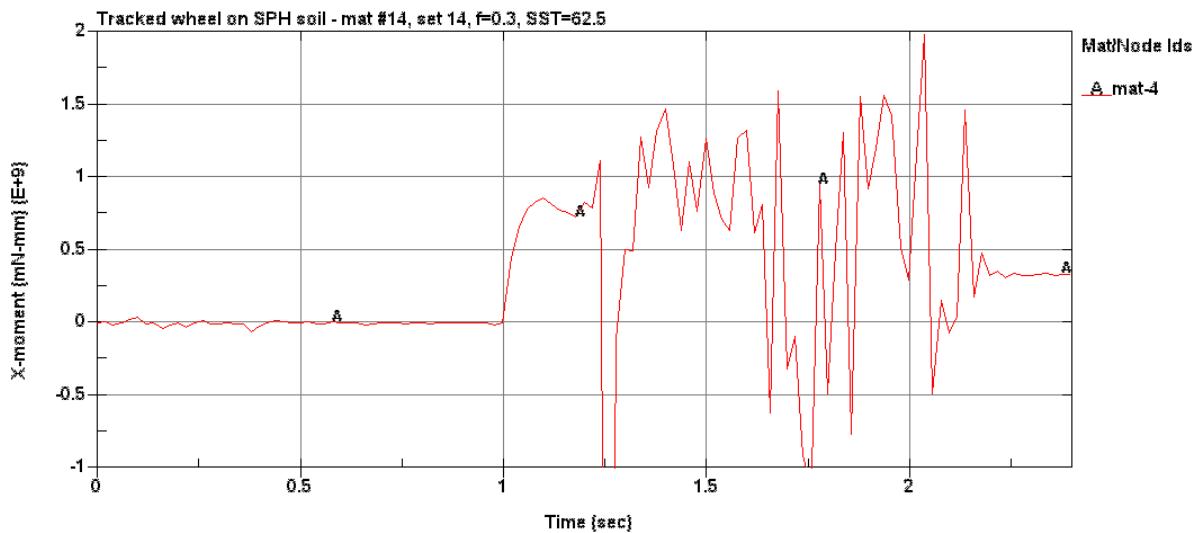


Figure 14c – X-moment of wheel during SPH analysis of Cuddeback A soil

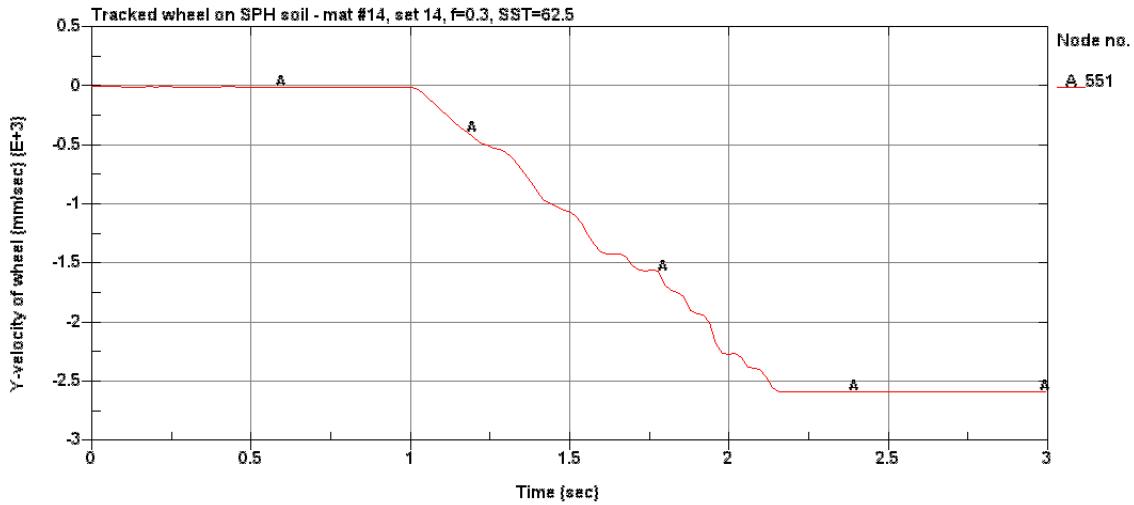


Figure 14d –Y-velocity of wheel during SPH analysis of Cuddeback A soil

Softer substrates were also simulated, with mixed results. The contact instability issue discussed in the previous sections also affected our SPH calculations. Unfortunately, the parameter values that worked so well to address this issue in solid-solid contact did not help as much in SPH calculations and, in fact, seemed to make the problem worse. Figure 15a shows the wheel on KSC dry sand (material type 5, data set 8). It has moved substantially across the substrate after sinking in noticeably during the stabilization period but the result looks good at this point:

Tracked wheel on SPH soil - mat #5, set 8, fric=0.3, SST=62.5
Time = 2.2

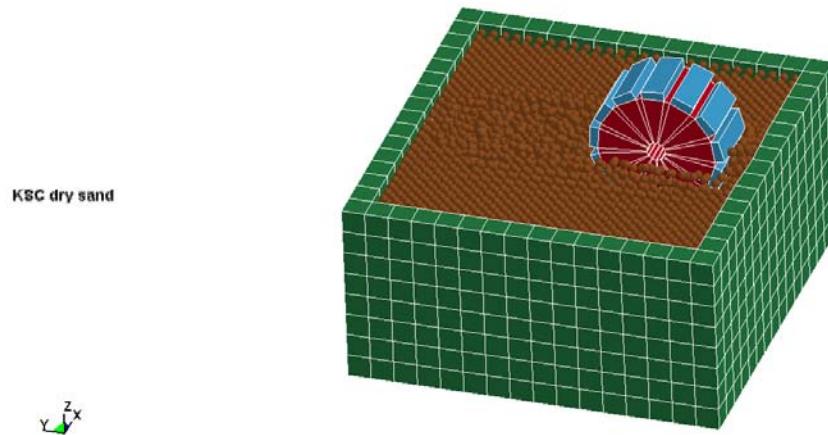


Figure 15a – Tracked wheel travelling on KSC dry sand (Material Type 5, data set 8)

Unfortunately, an instant later a contact instability occurred, terminating the analysis:

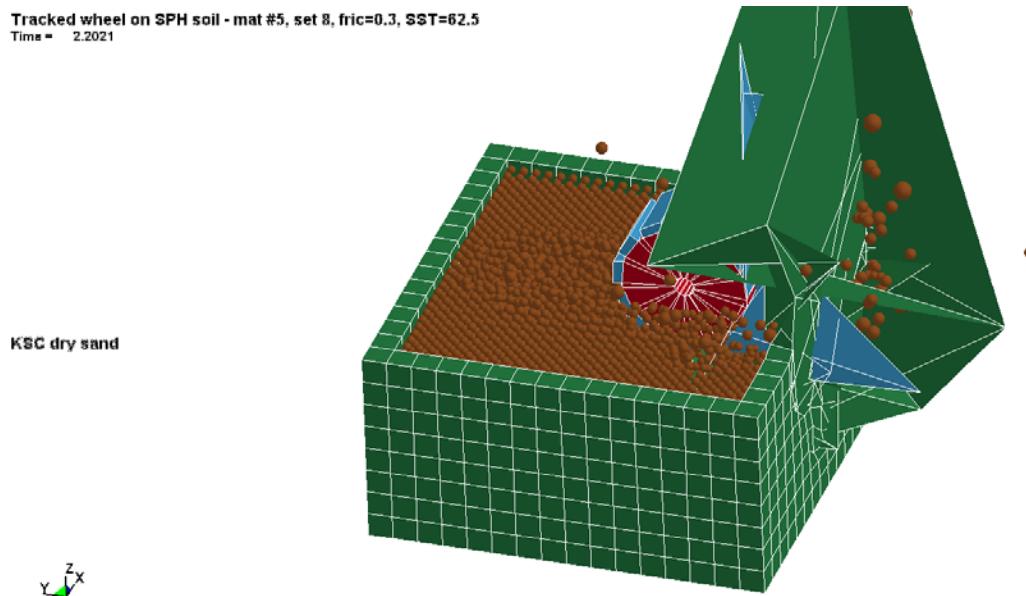


Figure 15b – Contact instability in SPH analysis of KSC dry sand (Material Type 5, data set 8)

A similar model, using Material Type 14, data set 8, also terminated due to a contact instability, but much earlier in the analysis (0.85 sec, cf. 2.2 sec) demonstrating just how unpredictable the instabilities can be:

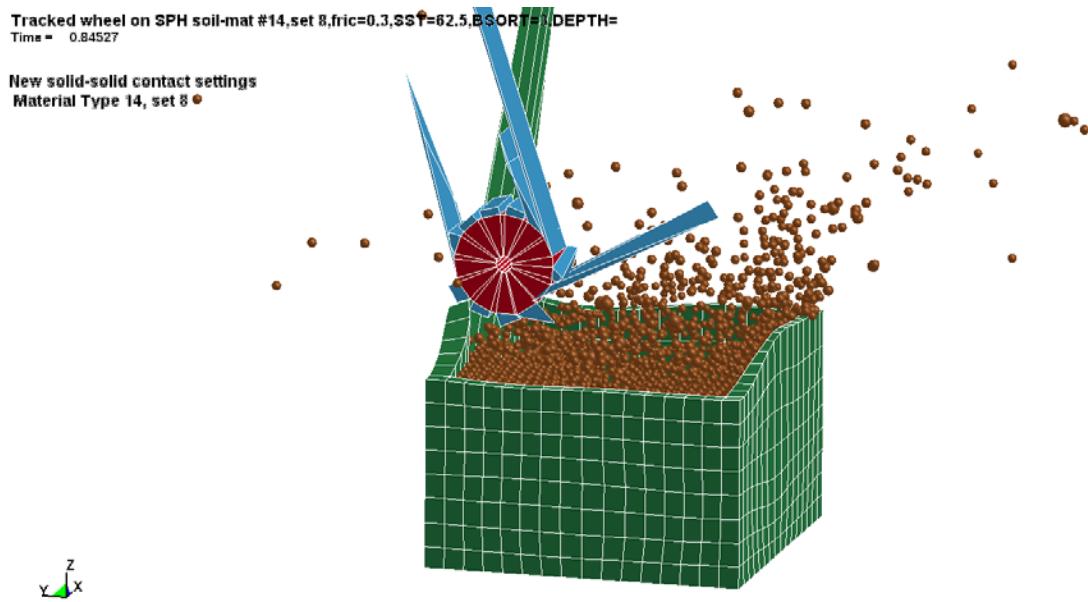


Figure 15c – Contact instability in SPH analysis of KSC dry sand (Material Type 14, data set 8)

Frustratingly, the models appear to be working well right up until the moment of the instability. Hopefully, once this issue is resolved, the good features of the models will be more apparent.

An attempt was made to run a model using a substantially finer SPH mesh (~100K nodes) using the material properties of a firm, moist sand (Kennedy Space Center, material type 5, data set 10). This model ran for about two weeks but at the time of this report had only gone just past the halfway point on the stabilization time period. Nevertheless, the below image of the pad sinking into the SPH material gives an idea as to the relative mesh density between pad shape and SPH spacing:

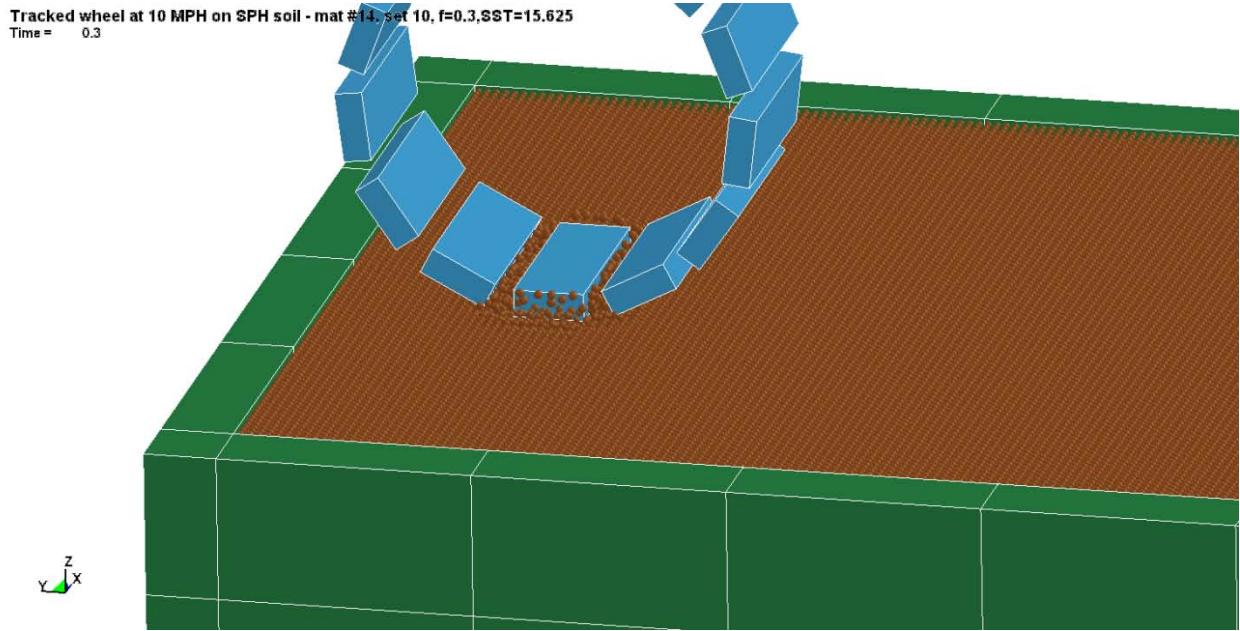


Figure 16 – Track pad on SPH material (KSC moist sand) during stabilization period

The penetration of the SPH nodes between the pads is something that the relatively coarse FEA meshes cannot duplicate as is also the case with the standard SPH mesh density used. For truly soft material simulations increased mesh density will be an essential tool for accuracy.

As extensions to the simple models we used to validate the SPH approach, we tried running two cases that would hopefully show the models' ability to quantify performance under different situations. One of these models had a rotated domain such that the wheel was attempting to climb a 10° slope. The below Figure shows the wheel at a point in the analysis just before terminating due to a contact instability:

Tracked wheel on SPH soil - mat #5, set 8, fric=0.3, 10 deg slope
Time = 1.88

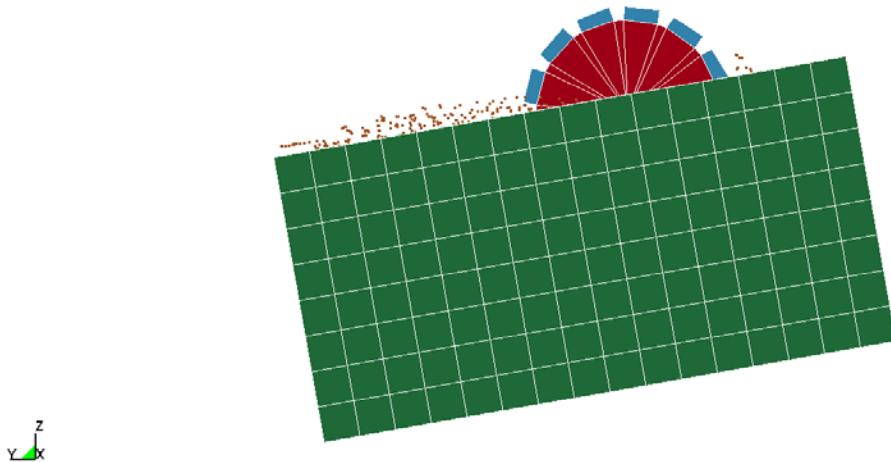


Figure 17a – Tracked wheel in KSC dry sand (Material Type 5, data set 8) at 10° slope

The wheel has dug substantially further into the substrate at this time in the analysis than at 0° slope:

Tracked wheel on SPH soil - mat #5, set 8, fric=0.3, SST=62.5
Time = 1.88

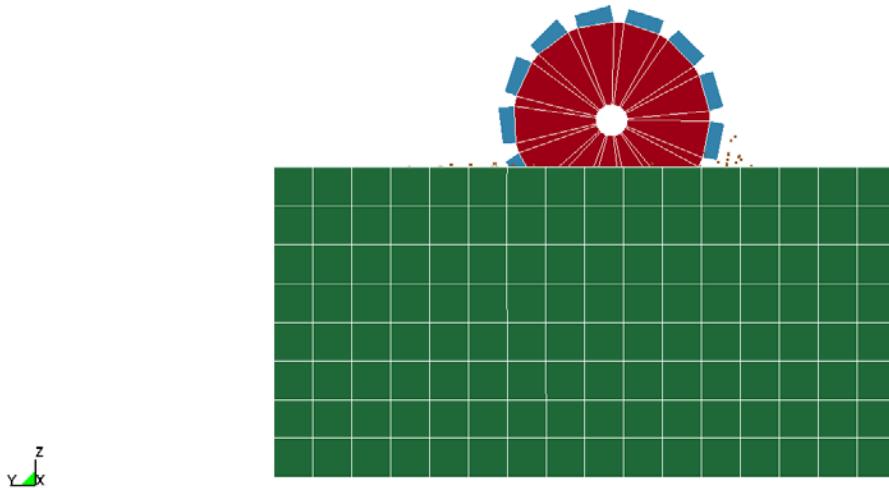


Figure 17b – Tracked wheel in KSC dry sand (Material Type 5, data set 8) at 0° slope at same time

Another model was constructed to handle an extremely rapid acceleration of the wheel to 25MPH over a 1.6 second time period. Under perfect conditions this would take a domain of ~9 m in length to still have the wheel on our substrate. Since our friction coefficient would require a longer length to obtain top velocity, the domain was extended to 15 m in order to allow the wheel to reach as close to its top velocity as possible (Figure 18).

Tracked wheel on SPH soil - mat #14, set 8, f=0.3, 25MPH, SST=50
Time = 0

Domain long enough to accomodate acceleration to 25MPH

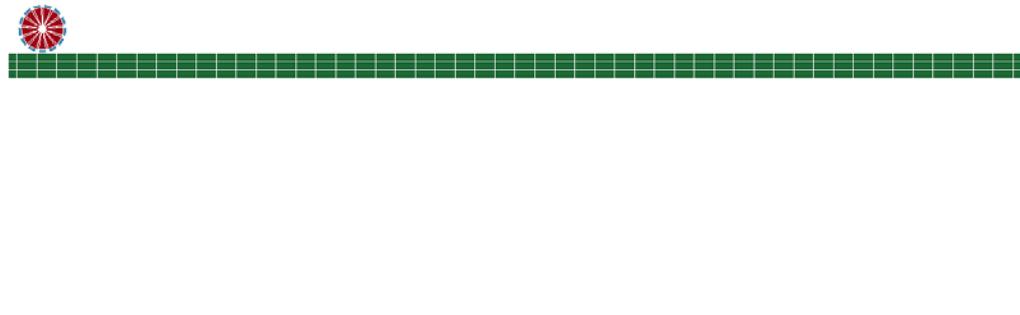


Figure 18 – Domain used for analysis of wheel accelerating to 25 MPH (11.2 m/s) in 1.6 sec
At the time of this report, the model had only reached ~0.9 seconds (0.4 seconds into the acceleration) but the results show a substantial ejection of SPH material behind the wheel due to the rapid acceleration and the wheel is digging its way into the substrate:

Tracked wheel on SPH soil - mat #14, set 8, f=0.3, 25MPH, SST=50
Time = 0.88

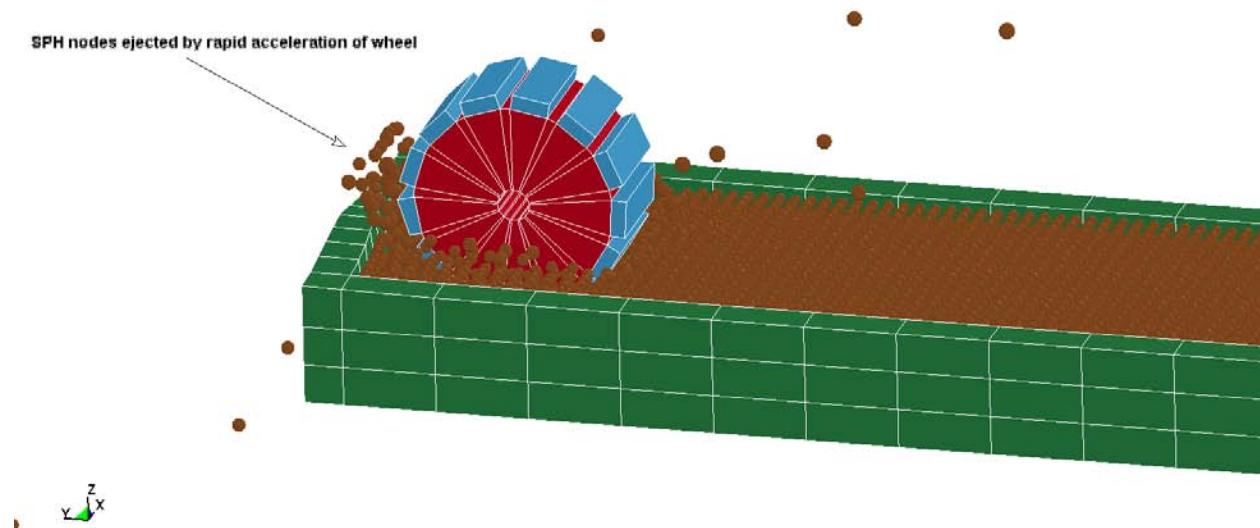


Figure 18a – SPH nodes being ejected behind rapidly accelerating wheel

These two model extensions demonstrate that even these simple SPH models can produce results showing the qualitative differences between parameters such as slope and acceleration rates. In the sloped case, the wheel dug itself deeper into the substrate, a result seen on actual tracked vehicles. In the high acceleration case, more substantial amounts of

material were ejected by the track pads than at the more modest acceleration rates used in other models.

1.4 Important Findings and Conclusions

For the general problem of simulating tracked wheels on substrates, the following solution strategies should be used:

For substrate materials that are quite hard (e.g. dry lakebed soils) typical solid elements should suffice for many analyses using solid-solid contact to address the interactions between track pads and the substrates. For hard materials that are more complex than the SOIL_AND_FOAM model in LS-DYNA is capable of representing, there are many alternative models in LS-DYNA that can be used.

For very soft materials (e.g. muds and soft sands) SPH technology should be employed as it would be the preferred mesh-free method to handle the extreme deformations likely to occur in these materials. Materials that are so soft that they easily are stressed into a “failed” regime where only their mass remains important is a situation very appropriate for SPH technology.

For those materials of intermediate hardness there are several alternatives. An eroding solid element contact approach could be used as well as the SPH method with the SPH method probably being more straightforward to implement.

The SPH models we have been able to successfully execute demonstrate an ability to capture the qualitative features we would expect for a tracked wheel on soft substrates. Ejection of the material from the domain, substantial sinking into the substrate and the formation of a “bulge” of material in front of the wheel have all been seen in these SPH analyses. After correcting our issue with contact instability the next step will be to refine the SPH models so as to obtain real, quantitative results for actual tracked wheel designs.

The most significant issue in this project involves the contact instabilities that produce the anomalous forces causing model termination. The good news is that we may have a solution in hand for solid-solid contact with the new parameter values. However, this will require extensive verification to be sure the solution is sufficiently robust and we still need to address its persistence in SPH models.

The contact instability issue is presumably present and lurking in all SPH models, but is also potentially correctable. However, we have simply run out of time to validate any further potential solutions from LSTC. There is one possible solution that should be attempted at the next opportunity (there was no time left in the project to validate it). The idea is to simply remove the solid-solid contact definitions from the input file and to expand the SPH domain so that the wheel will not encounter any of the other solid elements during the analysis. This idea was suggested by the observation that just about every instance of the contact instabilities that occurs appears to involve solid-solid contact. This may be incorrect but it is worth pursuing in a couple of models to see if the instability can be avoided. This will carry a price in computational efficiency. Before going too far down this path, however, it is critical to continue

working with LSTC support to confirm that the issue in these models is actually the solid-solid contact and not something else (e.g. SPH-solid contact).

When the instability does not present itself, we appear to have a functional set of analysis parameters that enables us to perform reasonable SPH analyses of tracked wheels on hard soils. Qualitatively the results look reasonable but further work needs to be done to refine the SPH mesh and determine how to obtain smoother X-moment results, which are almost useless in their current form. It may be possible to obtain results on soft substrates but those analyses have a high risk of being interrupted by contact instabilities.

Lastly, the SPH models in total offer a caution as to what level of CPU resources would be required to accurately simulate actual tracked vehicles on soft substrates for long periods of time (seconds). The combination of high acceleration rates/velocities, different track pad designs and soft soils will require the models to have long domain lengths, high SPH mesh density and deep SPH domains to accommodate settling/digging of the tracked wheels. The requirement for uniform SPH mesh density will mean that million node meshes will be required, at a minimum, to perform these analyses. It is worth noting that the SPH capability is compatible with the MPP (Massively Parallel Processing) version of LS-DYNA so it may be possible to reduce the computation times substantially on the right kind of equipment. As a first step to reduce the computational burden, symmetries in the SPH models need to be utilized to the maximum possible extent. Use should also be made of LS-DYNA commands that ignore SPH nodes a certain distance beyond the model.

1.5 Implications for Further Research

The DEM (Discrete Element Method) should also be evaluated as an analysis technique. For complex materials that are firm enough to not be instantly obliterated by the track pads it may provide more quantitative results than an SPH approach. LS-DYNA has a DEM capability under development but there are other commercial codes that could be used instead.

While a number of soil and sand data sets were accumulated during this project we did not find a comprehensive model that could describe a material over, say, an entire moisture range from dry to thoroughly wet. One of the materials notably absent from our collection of data is wet mud, a substance that tracked vehicles are likely to encounter.

The friction coefficients used in these analyses were averaged values from one reference. Presumably, there exists a more comprehensive set of friction data for track pads on a wide variety of soils or one can obtain such data experimentally. The role of that parameter was not investigated at all here but would be an obvious target for inclusion into a proper Design Of Experiments evaluation of the general problem.

Adaptive Solid to SPH elements is a technology that should also be pursued further, but in close cooperation with LSTC support personnel and after our other issues with SPH contact have been resolved. While the primary issues we encountered appear to be contact driven, there is still a lack of clarity from LSTC documentation on just what “coupling” between the newly-liberated SPH nodes and solid elements actually entails which makes it difficult to determine just what combination of ICPL and IOPT parameters most closely describes our

physical situation. Again, this should be a tractable issue that can be resolved with further communication with LSTC.

1.6 References

- Chitty, M. A. (2011). *Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323*. Albuquerque, NM: Applied Research Associates, Inc.
- Guo, Y. (2010). Meshless Methods in LS-DYNA: an Overview of EFG and SPH. Stuttgart, Germany: LSTC.
- Hallquist, J. (2006). *LS-DYNA Theory Manual*. Livermore, CA: LSTC.
- Kulak, R. F., & Schwer, L. (2012). Effect of Soil material Models on SPH Simulations for Soil-Structure Interaction. *12th International LS-DYNA Users Conference*. Detroit, MI.
- Lescoe, R. (2010). *Improvement of soil modeling in a tire-soil interaction using Finite Element Analysis and Smooth Particle Hydrodynamics - MS Thesis*. Penn State University.
- Livermore Software Technology Corporation. (2012). *LS-DYNA Keyword User's Manual Volume 1 Version 971*. Livermore, CA.
- Schwer, L. (2012, November 16). Personal communication.
- Tekeste, M. Z., Raper, R. L., Tollner, E. W., & Way, T. R. (2005). Effect of Soil Moisture, Soil Density, and Cone Penetrometer Material on Finite Element Prediction of Soil Hardpan Depth (Paper #: 051164). *ASAE Annual International Meeting* (p. 9 (Table 4)). Tampa, FL: American Society of Agricultural Engineers.

1.7 Annex - Material Models used in Simulations

Several material types and data sets have been used in the SPH modeling of soils. In LS-DYNA terminology, these can be described as:

- Material Type 5, Soil and Foam
- Material Type 9, Null Material
- Material Type 10, Elastic Plastic Hydro(dynamic)
- Material Type 14, Soil and Foam Failure
- Material Type 25, Inviscid Two Invariant Geologic Cap
- Material Type 79 (Hysteretic Soil (Elasto-Perfectly Plastic))

These are part of the wide variety of material models appropriate for soil available in LS-DYNA but appear to cover the range of constitutive relations used in published SPH simulations involving soil. The vast majority of our analyses were performed on materials of Type 5 or 14. Material Type 14 is a version of Material Type 5 where the material loses its

ability to carry tension when the PC (pressure cutoff) value is exceeded. The input data for Material Type 14 is the same as for Material Type 5.

An equation of state is required in SPH analyses to address the pressure-volume relationship in compression. For the Type 5 or 14 materials this is addressed by the P vs. ln (vol. strain) data. For other constitutive models a specific EOS must be chosen from the number of models available in LS-DYNA. Most of the EOS models in LS-DYNA are designed to be compatible with shock waves. Due to the relatively low-speed nature of track/substrate interactions (very low Mach number) the simplest form of EOS is all that is required. For those constitutive models that required an EOS we used the simplest representation (Linear Polynomial) with only non-zero coefficient C2=K (bulk modulus).

Below is a set of tables listing explicit values for the various material model data sets found in a brief review of the literature, organized by Material Type. Parameter values are listed with the units used in the simulations (mm-mN-kg-sec system).

Material Type 5 (Soil and Foam)

Kulak, R. F.; Bojanowski, C. 'Modeling of cone Penetration Test Using SPH and MM-ALE Approaches' 8th European LS-DYNA® Users Conference, Strasbourg (May, 2011) (Table 1)

Description	Norfolk Sandy Loam	Set 1
Variable	Value	Units
Density (RO)	1.255e-6	kg/mm ³
G	1724	mN/mm ²
K	5516	mN/mm ²
A0	0	
A1	0	
A2	0.8702	
PC	0	
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0, 0.12783, 0.19845, 0.24846, 0.31471, 0.40048	
P	0, 50, 100, 150, 250, 500	mN/mm ²

Kulak, R. F.; Schwer, L. 'Effect of Soil Material Models on SPH Simulations for Soil-Structure Interaction' 12th International LS-DYNA® Users Conference (Table 2, Figures 1&2)

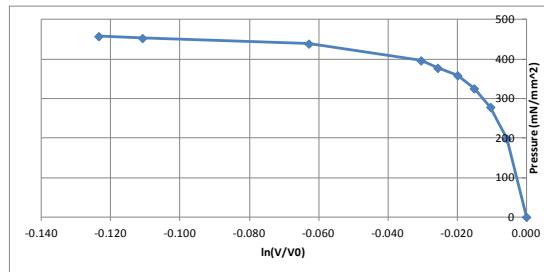
Description	Soil	Set 2
Variable	Value	Units
Density (RO)	1.64e-6	kg/mm ³
G	136000	mN/mm ²
K	4700000	mN/mm ²
A0	0	
A1	0	
A2	0.3736	
PC	0	
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.02,0.04,0.06,0.08,0.1,0.15,0.20,0.25,0.30	
P	0,7912,1,14945,22857,29011,35165,53846,84615,142308,242308	mN/mm ²

Barsotti, M. A.; Puryear, J. M. H.; Stevens, D. J.; Alberson, R. M.; McMahon, P. "Modeling Mine Blast with SPH" 12th International LS-DYNA® Users Conference

Description	Soil	Set 3
Variable	Value	Units
Density (RO)	1.37e-6	kg/mm ³
G	3.6	mN/mm ²
K ¹		mN/mm ²
A0	25300	(mN/mm ²) ²
A1		mN/mm ²
A2		
PC	-34.47	mN/mm ²
VCR	1 (off)	
REF	0	

With Pressure vs. Volumetric Strain data shown below:

Strain	ln(1-strain)	P (mN/mm ²)
0.00000	0	0.00
0.00563	-0.0056409	198.11
0.01031	-0.010366	278.30
0.01500	-0.0151136	325.47
0.01969	-0.0198839	358.49
0.02531	-0.0256384	377.36
0.03000	-0.0304592	396.23
0.06094	-0.0628732	438.68
0.10500	-0.1109316	452.83
0.11625	-0.1235811	457.55



¹ DYNA automatically calculates a value of K (35120) from the first two data points of the P vs. ln(1-strain) table.

Bojanowski, C.; Kulak, R. F. 'Comparison of Lagrangian, SPH and MM-ALE Approaches for Modeling Large Deformations in Soil' 11th International LS-DYNA® Users Conference (2010) (Table 1, Figure 3):

Description	Loose, silty clay sand	Set 4
Variable	Value	Units
Density (RO)	2.359e-6	kg/mm ³
G	34474	mN/mm ²
K	15024	mN/mm ²
A0	0	
A1	0	
A2	0.602	
PC	0	
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.023,0.04,0.06,0.07,0.08,0.088,0.101,0.112,0.128	
P	0,80,130,280,350,405,490,610,700,900	mN/mm ²

Fasanella, E. L.; Lyle, K. H.; Jackson, K. E. "Developing Soil Models for Dynamic Impact Simulations" (Table 2, Figure 14)

Description	Unwashed Sand	Set 5
Variable	Value	Units
Density (RO)	2.095e-6	kg/mm ³
G	23029	mN/mm ²
K	133556	mN/mm ²
A0	300.7	(mN/mm ²) ²
A1	25.56	mN/mm ²
A2	0.543	
PC	-6.895	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.00251,0.00474,0.007,0.0092,0.0103,0.0114,0.0125,0.0137,0.016	
P	0,66.73,137.16,203.88,274.32,311.39,344.75,378.11,411.48,489.32	mN/mm ²

Fasanella, E. L.; Lyle, K. H.; Kellas, S. "Soft Soil Impact Testing and Simulation of Aerospace Structures" 10th International LS-DYNA® Users Conference (Table 1, Figure 5)

Description	Soft Soil	Set 6
Variable	Value	Units
Density (RO)	1.4535e-6	kg/mm ³
G	1841	mN/mm ²
K	68950	mN/mm ²
A0	0	(mN/mm ²) ²
A1	0	mN/mm ²
A2	0.3	
PC	0	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.00396,0.0119,0.032,0.0568,0.0891,0.12,0.157	
P	0,7.51,13.94,27.89,41.83,55.24,68.64,82.59	mN/mm ²

Palmer, T.; Honken, B.; Chou, C.; 'Rollover Simulations for Vehicles using Deformable road Surfaces' 12th International LS-DYNA® Users Conference (Page 8)

Description	Damp Sand	Set 7
Variable	Value	Units
Density (RO)	1.6e-6 (assumed; not listed)	kg/mm ³
G	11800	mN/mm ²
K	107500	mN/mm ²
A0	0	
A1	0	
A2	0.6	
PC	0	
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.006,0.009,0.012,0.015,0.018,0.021,0.024,0.027,0.030	
P	0,79.2,194.4,329.2,548.5,740.5,1024.5,1284.7,1562.2,1925.2	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 41)

Description	KSC Low Density, Dry Sand	Set 8
Variable	Value	Units
Density (RO)	1.2835e-6	kg/mm ³
G	508.1615	mN/mm ²
K	257804.05	mN/mm ²
A0	4.2687086	(mN/mm ²) ²
A1	2.915206	mN/mm ²
A2	0.4978	
PC	-3.4475	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.01586,0.3794,0.04539,0.0525,0.05748,0.0656,0.0694,0.0732	
P	0,18.51,19.65,125.08,128.66,228.91,337.17,607.1,767,929.45	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 110)

Description	KSC Mason Sand, 15% moisture	Set 9
Variable	Value	Units
Density (RO)	1.8397e-6	kg/mm ³
G	13755.525	mN/mm ²
K	431489.1	mN/mm ²
A0	62.896776	(mN/mm ²) ²
A1	13.962375	mN/mm ²
A2	0.7752	
PC	-6.895	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.0092	
P	0,21.175,66.985,136.935,231.258,355.437,506.507,687.983,895.66, 1181.11	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 58)

Description	KSC High Density In-situ moisture Sand	Set 10
Variable	Value	Units
Density (RO)	1.6044e-6	kg/mm ³
G	9701.265	mN/mm ²
K	299436.06	mN/mm ²
A0	67.555797	(mN/mm ²) ²
A1	11.95593	mN/mm ²
A2	0.5290	
PC	-6.895	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.001,0.002,0.003,0.004,0.006,0.008,0.01,0.012,0.01394	
P	0,17.493,49.582,89.635,138.727,259.804,414.665,586.558,780.514, 978.4	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 74)

Description	Mason 4% water	Set 11
Variable	Value	Units
Density (RO)	1.615e-6	kg/mm ³
G	12080.04	mN/mm ²
K	389360.65	mN/mm ²
A0	64.750876	(mN/mm ²) ²
A1	13.176345	mN/mm ²
A2	0.6700	
PC	-3.4475	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.009,0.01031	
P	0,19.03,58.125,98.323,196.439,296.209,413.011,545.739,866.219, 1108.716	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 86)

Description	Mason 8% water	Set 12
Variable	Value	Units
Density (RO)	1.6578e-6	kg/mm ³
G	16368.73	mN/mm ²
K	347825.17	mN/mm ²
A0	65.036122	(mN/mm ²) ²
A1	13.203925	mN/mm ²
A2	0.6700	
PC	-3.4475	mN/mm ²
VCR	0	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.009,0.01044	
P	0,26.270,65.296,121.904,196.025,291.521,401.565,530.846, 851.050,1116.507	mN/mm ²

Chitty, M. A. (2011). Constitutive Soil Properties for Mason Sand and Kennedy Space Center NASA/CR-2011-217323. Albuquerque, NM: Applied Research Associates, Inc. (Page 98)

Description	Mason 5% water	Set 13
Variable	Value	Units
Density (RO)	1.6792e-6	kg/mm ³
G	24539.305	mN/mm ²
K	493544.1	mN/mm ²
A0	62.421366	(mN/mm ²) ²
A1	14.279545	mN/mm ²
A2	0.8164	
PC	-3.4475	mN/mm ²
VCR	0	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.001,0.002,0.003,-0.004,0.005,0.006,0.007,0.008,0.00861	
P	0,36.930,99.357,187.682,308.275,455.277,635.374,845.327, 1083.205,1236.274	mN/mm ²

20080045436_2008045156 NASA report - Constitutive Soil Properties for Cuddeback Lake, CA and Carson Sink, NV (Page 54)

Description	Cuddeback A	Set 14
Variable	Value	Units
Density (RO)	1.4546e-6	kg/mm ³
G	17237.5	mN/mm ²
K	119973	mN/mm ²
A0	3685.3803	(mN/mm ²) ²
A1	105.2177	mN/mm ²
A2	0.7510	
PC	-13.79	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.0089,0.0104,0.012,0.0138,0.0155,0.019,0.0226,0.0263,0.0291	
P	0,206.85,241.325,275.8,310.275,344.75,413.7,482.65, 551.6,605.381	mN/mm ²

20080045436_2008045156 NASA report - Constitutive Soil Properties for Cuddeback Lake, CA and Carson Sink, NV (Page 73)

Description	Cuddeback B	Set 15
Variable	Value	Units
Density (RO)	1.2942e-6	kg/mm ³
G	3171.7	mN/mm ²
K	168238	mN/mm ²
A0	6.1803333	(mN/mm ²) ²
A1	5.14367	mN/mm ²
A2	1.0680	
PC	0	mN/mm ²
VCR	0 (on)	
REF	0 (off; ref. geometry)	
-ln(V/V0)	0,0.0222,0.0317,0.0379,0.0436,0.0491,0.0592,0.0683,0.0766,0.0827	
P	0,124.11,172.375,206.85,241.325,275.8,344.75,413.7, 482.65,537.81	mN/mm ²

Material Type 9 (Null Material)

The Null Material model can be constructed from any soil model that has a density and a bulk modulus available. For example, Data Set 5 from Material Type 5 above would have these values as a NULL material:

Description	Unwashed Sand	
Variable	Value	Units
Density (RO)	2.095e-6	kg/mm ³
K	133556	mN/mm ²

Material Type 10 (Elastic Plastic Hydrodynamic)

Kulak, R. F.; Schwer, L. 'Effect of Soil Material Models on SPH Simulations for Soil-Structure Interaction' 12th International LS-DYNA® Users Conference (Table 2, Figures 1&2)

Description	Soil	
Variable	Value	Units
Density (RO)	1.64e-6	kg/mm ³
G	136000	mN/mm ²
PC	-510	mN/mm ²
A1	1057.8	mN/mm ²
-ln(V/V0)	0,0.02,0.04,0.06,0.08,0.1,0.15,0.20,0.25,0.30	
P	0,7912.1,14945,22857,29011,35165,53846,84615,142308,242308	mN/mm ²

Lescoe, R. "Improvement of soil modeling in a tire-soil interaction using finite element analysis and smooth particle hydrodynamics" Master's Thesis, Penn State University, 2010.

Description	Dense Sand/Sandy Loam	
Variable	Value	Units
Density	1.60e-6	kg/mm ³
E	22000	mN/mm ²
K	15000	mN/mm ²
G	9000	mN/mm ²
Yield Stress Y	16	mN/mm ²
Hardening Modulus	100	mN/mm ²

For an Equation of State, Lescoe used a linear polynomial with the only non-zero coefficient C1=K.

Material Type 25 (Inviscid Two Invariant Geologic Cap)

Kulak, R. F.; Schwer, L. 'Effect of Soil Material Models on SPH Simulations for Soil-Structure Interaction' 12th International LS-DYNA® Users Conference (Table 2)

Description	Soil	
Variable	Value	Units
Density (RO)	1.64e-6	kg/mm ³
G	5289000	mN/mm ²
K	15000000	mN/mm ²
Theta	0.20375	radians
R	2.3	
D	1.6e-3	
W	0.49	
X0	46500	mN/mm ²

Surf-to-Shore Visualization

Ryan B. Foss
BAE Systems

Motivation

Visualization via 3D rendering employing heuristic-based¹ physics models can provide insight into environmental behavior that is not always available through conventional analysis and simulation models. A visualization approach can also reveal missing behaviors or corner cases not exposed by a quantitative model. This is particularly the case with highly complex environments such as a surf-zone. And of course, driving simulations have long been useful as supplemental sources of information to model-based design. In the following, we describe a surf-to-shore visualization that extends from a conventional driving simulation (both of these documented and released as part of the project deliverables).

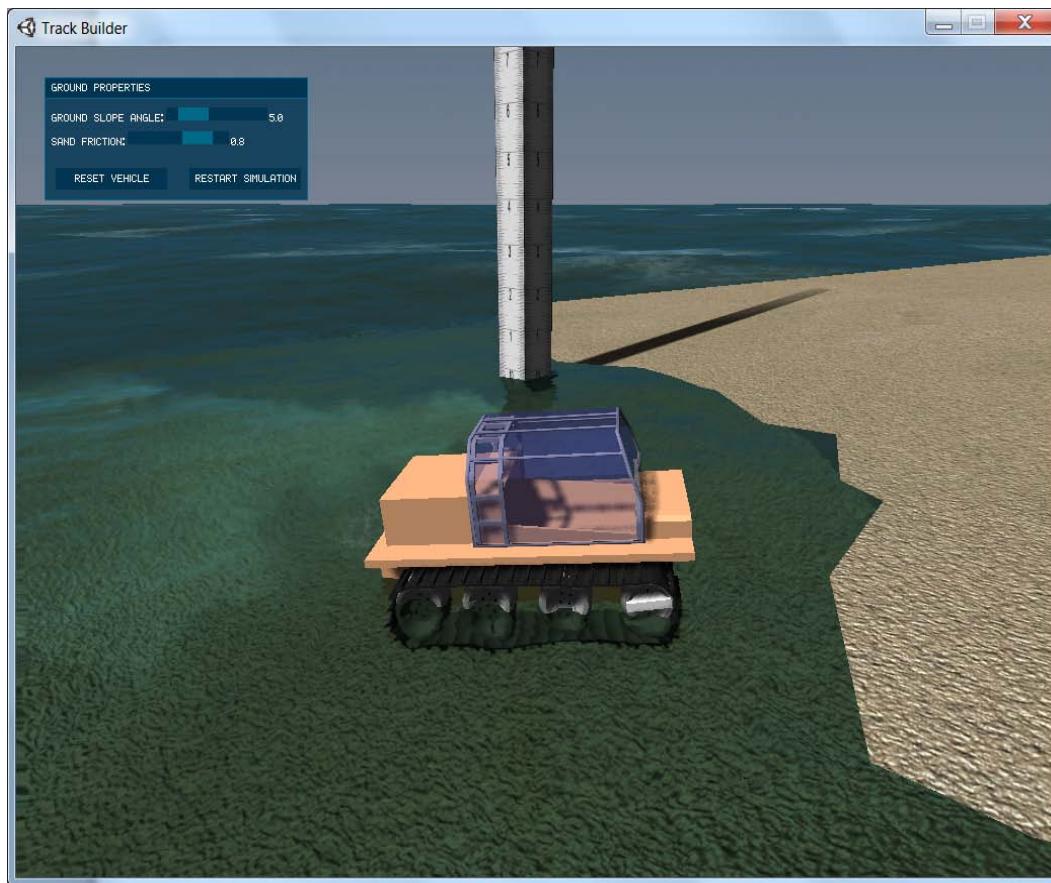


Figure 1: Surf-to-shore visualization context

¹ These are referred to as “gaming” physics models, yet gaming physics often contains a significant fundamental underpinning in real mathematical-based physics. See [1] for example.

Basic Detail

The vehicle is simulated using physics provided by the Unity 3D engine[2]. Physical properties are assigned to the geometry in the scene. Properties include static and dynamic frictions. Forces and torques are applied to a rigid body, which cause the vehicle to move. Buoyancy forces act while the body of the vehicle is in water. Friction properties react between the wheels and linkages making the track, as well as the track with the ground when it is in contact.

Additional interaction and forces have not been added yet. The current state only considers the friction of the land regardless of its interaction with the water.

Water Model (Ocean)

Many real-time variations exist for ocean visualization. In this case, a publicly available open source ocean was used. It uses a typical Fourier FFT to deform a mesh representing the ocean. The ocean is can be queried for wave height at a position, which is what is used for physical interaction calculations in the simulation.

Buoyancy

The buoyancy is calculated based on a simpler representation of the shape of the vehicle, a rectangular volume. This volume is divided into multiple sections and for each a buoyancy calculation is performed. The buoyancy force is based on how much of the volume is submerged.

For the sake of simplicity, the calculations are assumed to counter act the forces of gravity, thus making the volume float on the water when essentially half submerged.

Additionally, the drag impacted on the vehicle changes depending on the amount that is submerged.

Vehicle, how it works

Many approaches were considered and tested when building the vehicle. This particular iteration uses much of the internal Unity physics to provide motion and physical interaction. The tracks are modeled using sectional linkages that are powered by wheels. A simple drive train was added to ramp up and down the control inputs, specifically providing torque to the wheels and propellers.

When the vehicle is in the water, the propellers provide force to the vehicle only if it is submerged.

Note that the Unity vehicle used in the amphibious settings differs from the vehicle used in rigid course and track based simulations, described briefly next.

Terrain Model (Track)

Figure 3 below illustrates the 3D track generation and Figure 4 the visual rendering

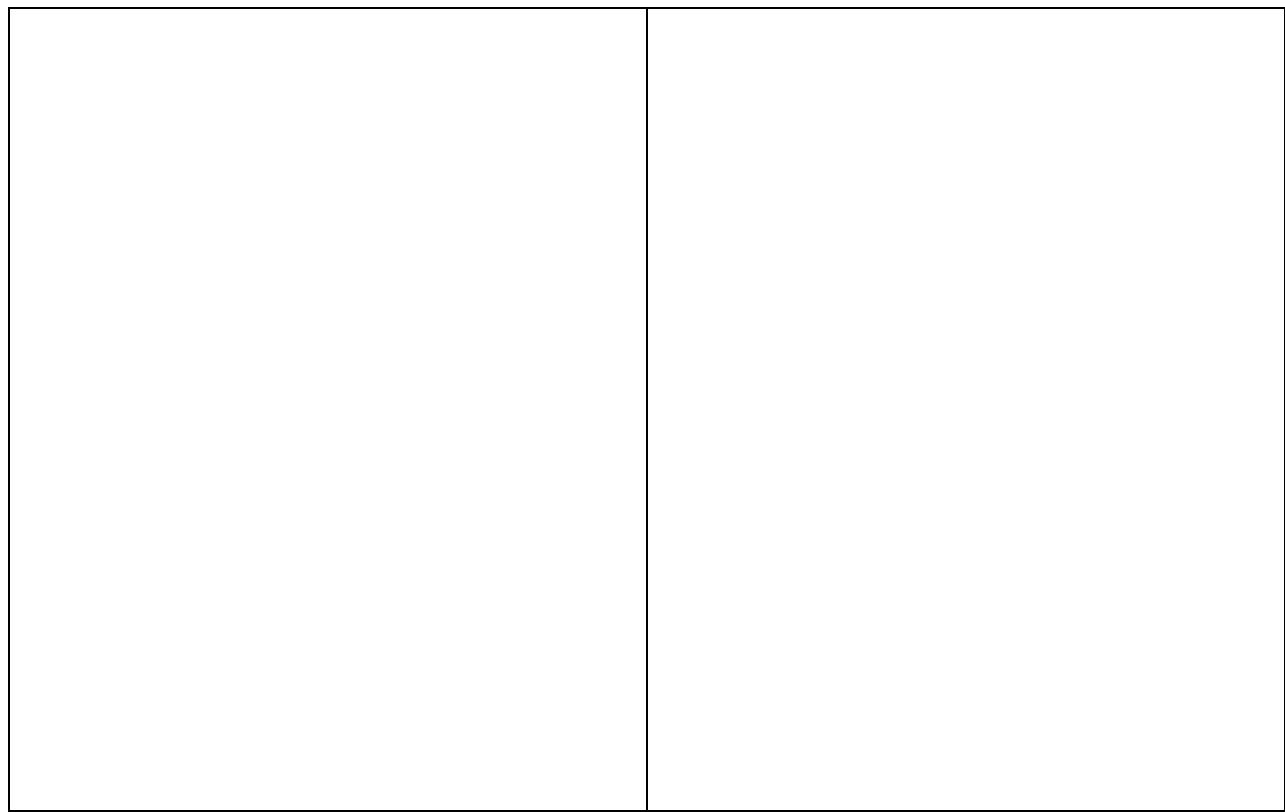


Figure 2: Overview of visual rendering of tracks via a 3D track generation process



Figure 3: Driving visualization where either stochastic roughness (left) or deterministic obstacles (right) track profiles are rendered, with vehicle interactions applied.

The usage of these models are described in two documents

\Models\Aquatic\features\surf\doc\UserGuide-SurfToShore.docx
 \Models\Land\terrains\grades_slopes\doc\UserGuide-TerrainBuilder.docx

Concerns

I attempted a number of different models to represent this effect and settled on this one. It is a satisfying simulation, but not without limitations and frustrations. I would consider it to be a first iteration which provides insight into the complexity of the problem. Additionally, I have much better understanding of the physics engine Unity uses and the methods available, as well as potential limitations it contains.

The physical interaction between the linkages representing the rubber track and the powered wheels produces a very satisfying simulation, but its relation to a real world interaction is questionable. At the moment, the tracks limit the vehicle speed regardless of inputting larger torques. **I suspect this is because the wheels are pulling and pushing, and because the track is constructed the way it is, it's like pushing a rope.**

The drive train and propeller drives are separate from each other, and produce the same forces regardless of being engaged or not. For instance, if the engine is powering the propellers and tracks it could be assumed that the power is divided between the two systems. The drive train does supply some ramping and smoothing to the inputs as the engine is engaged and disengaged.

Steering is a very simplistic representation that does not work well.

Not Addressed

A number of elements are not addressed in this simulation.

- Sink,

- Slip,
- Impact of water on the ground with regards to traction,
- Steering,
- Wave direction and body movement

Future Improvements

There are a number of ways the simulation could be improved. The terrain could be modeled with various sections or transition zones with different physical properties, such as slippery earth closer to the shore.

The drive train needs improvement to provide more appropriate torques to the tracks and propellers.

Investigate and improve the track model so forces are translated through the system properly.

Conclusions

These kinds of tools are useful for rapid validation of high level design performance on test tracks and in aquatic environments.

References

- [1] J. Tessendorf, "Simulating ocean water," *Simulating Nature: Realistic and Interactive Techniques*. SIGGRAPH, 2001.
- [2] "Unity (game engine) - Wikipedia, the free encyclopedia." [Online]. Available: [http://en.wikipedia.org/wiki/Unity_\(game_engine\)](http://en.wikipedia.org/wiki/Unity_(game_engine)). [Accessed: 12-Feb-2013].

3-D Amphibious Vehicle Model

1. Summary

As part of the C2M2L modeling effort, we are delivering non-compliant (i.e. “fixed”) terrain profiles that will be used for testing AVM designs. As the terrain profiles are generated in a custom coordinate system, we are providing example vehicle models to illustrate how a typical vehicle is orientated with respect to the terrain model. The specification of the terrain profile is described in Section X, while this appendix illustrates how an amphibious tracked vehicle design can be applied to a set of terrain context models featuring straight, curved, banked and sloped surface sections. The findings suggest that the choice of the coordinate system is crucial to allowing the terrain profiles to be used in a portable way across the surface vehicle standards.

An existing commercial amphibious vehicle, the Hydratrek D2488, was selected to serve as the basis of the vehicle model developed for this task. The current model does not realize the track system on the commercial vehicle, but instead implements a wheeled version. This model interacts with terrain in the form of OpenCRG data files via one of two tire models, a high quality model which uses Pacejka’s “magic formula,” and a fast lower quality model. Both models assume that the ground is rigid.

If the MATLAB Instrument Control Toolbox is available, the example code can be modified¹ so that the example code directly drives the Zulu visualization tool; currently the simulation data is written to a file which can be read by Zulu. It is possible to use the model to drive interactively in Zulu, watching the model react to steering commands in near real time.

The equations of motion developed incorporate a number of approximations designed to speed up the model. There are a number of limitations to the model, and further development is needed to make it truly useful for design work.

1.1 Technical Challenges

The exact equations of motion for a wheeled or tracked vehicle interacting with an uneven compliant terrain are highly nonlinear and extremely complicated. The basic technical challenge in creating a 3-D amphibious vehicle model was to simplify the model enough so that it can run quickly but still retain essential physical authenticity.

1.2 General Methodology

The full equations of motion were derived using no small angle approximations. A MATLAB model was then created which implemented these equations with various approximations

¹ Set the variable DEBUG_VAR_use_udp to true in the file `amphibious_gui.m`.

intended to increase the execution speed of the code. The resulting code was integrated into a simple MATLAB graphical user interface shown in Figure 1.

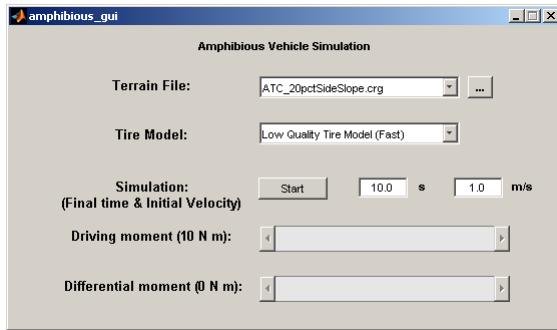


Figure 1. The amphibious vehicle graphical user interface.

The user may select the OpenCRG terrain file and choose which tire model to use from the menus and adjust the final simulation time and the initial velocity before starting the simulation. While the simulation is running the driving and differential moment sliders may be used to interactively drive the vehicle. If the Zulu visualization tool is running and the code has been modified to use it directly (see footnote 1 above), the vehicle will be seen moving in the Zulu window.

1.3 Technical Results

This report documents the development of the equations of motion for an n -axle vehicle, similar to the Hydratrek D2488², but without the over-the-tire rubber track system. (For the vehicle implemented in the example code $n = 4$.) A number of approximations, intended to simplify the analysis, are acknowledged here.

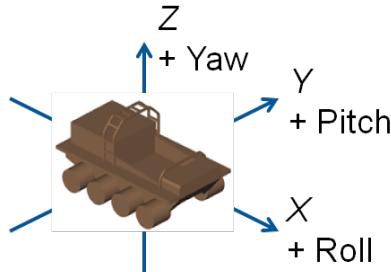


Figure 2. Sign conventions used for developing the equations of motion.

As much as convenient, uppercase vectors are body-fixed, while lowercase vectors are spatial³. Let \mathbf{X} be body-fixed coordinates originating at the center of mass of the vehicle, with X going

² Hydratrek website <<http://www.hydratrek.com/d2488.php>>.

³ Physical quantities which transform by $Q\mathbf{U} = \mathbf{u}$ and $QTQ^T = \boldsymbol{\tau}$ will be referred to as vectors and tensors, respectively. Quantities which are merely gathered together to take advantage of Matlab's matrix capabilities will be called arrays.

forward, Y to the left, and Z upward, and let $Q(t)$ be the rotation (special orthogonal⁴) matrix. See Figure 2. Then the rigid-body motion of the vehicle sprung mass is given by

$$\mathbf{x} = Q(t)\mathbf{X} + \mathbf{c}(t), \quad Q(t) \in SO(3),$$

where $\mathbf{c}(t)$ is the motion of the center of mass. Body-fixed vectors \mathbf{U} are transformed into spatial vectors \mathbf{u} via $Q(t)\mathbf{U} = \mathbf{u}$, and vice-versa using $Q^T(t)$: $Q^T(t)\mathbf{u} = \mathbf{U}$. The velocity is given by

$$\mathbf{v} = \dot{Q}(t)\mathbf{X} + \dot{\mathbf{c}}(t), \quad \mathbf{v} = \dot{Q}(t)Q^T(t)(\mathbf{x} - \mathbf{c}(t)) + \dot{\mathbf{c}}(t).$$

The spatial angular velocity tensor is $\hat{\omega}(t) = \dot{Q}(t)Q^T(t)$, while the body-fixed angular velocity tensor is $\hat{\Omega}(t) = Q^T(t)\dot{Q}(t)$. Both $\hat{\omega}$ and $\hat{\Omega}$ are skew-symmetric. These are related to the spatial and body-fixed angular velocity vectors $\boldsymbol{\omega}$ and $\boldsymbol{\Omega}$ via

$$\hat{\omega}\mathbf{u} = \boldsymbol{\omega} \times \mathbf{u}, \quad \hat{\Omega}\mathbf{U} = \boldsymbol{\Omega} \times \mathbf{U}.$$

It is assumed that the wheel centers may only move vertically in the body-fixed frame, corresponding to an ideal suspension constraint. The sprung mass (the vehicle body) interacts with the wheels through suspension forces $\mathbf{P}_k(t)$ (body-fixed) and $\mathbf{p}_k(t) = Q(t)\mathbf{P}_k(t)$ (spatial). (Going forward, a subscript k is implied to run from $1, \dots, 2n$, accounting for the $2n$ wheels.) If the static equilibrium positions of the $2n$ wheel centers in the body-fixed coordinates are \mathbf{R}_k and $\zeta_k(t)$ the vertical suspension displacements, then the spatial positions of the wheel centers are

$$\mathbf{r}_k(t) = Q(t)\{\mathbf{R}_k + \zeta_k(t)\boldsymbol{\partial}_z\} + \mathbf{c}(t).$$

The dynamical equations are Newton's second law for the sprung mass

$$m\ddot{\mathbf{c}}(t) = -mg\boldsymbol{\partial}_z + \sum_{k=1}^{2n} Q(t)\mathbf{P}_k(t) = -mg\boldsymbol{\partial}_z + Q(t) \sum_{k=1}^{2n} \mathbf{P}_k(t),$$

where m is the sprung mass, g is the acceleration due to gravity, and $\boldsymbol{\partial}_z$ is a unit vector pointing vertically, and the moment (Euler) equation for the motion about the center of mass

$$\boldsymbol{\tau}_c(t) \equiv \sum_{k=1}^{2n} (\mathbf{r}_k(t) - \mathbf{c}(t)) \times \mathbf{p}_k(t) - \mathbf{m}_k(t) = \dot{\mathbf{h}}_c(t),$$

where $\boldsymbol{\tau}_c(t)$ is the torque about the center of mass, $\mathbf{r}_k(t)$ are the spatial positions of the wheel centers, $\mathbf{m}_k(t)$ is the spatial moment applied to wheel k , and $\mathbf{h}_c(t)$ is the angular momentum. Transforming to the body-fixed frame we obtain the Euler-Poincaré form

$$\mathbb{I}_c \dot{\boldsymbol{\Omega}}(t) = \mathbf{T}_c(t) - \boldsymbol{\Omega}(t) \times \mathbb{I}_c \boldsymbol{\Omega}(t),$$

where \mathbb{I}_c is the inertia tensor about the center of mass in body-fixed coordinates, $\boldsymbol{\Omega}(t) = Q^T(t)\boldsymbol{\omega}(t)$ is the body-fixed angular velocity vector, $\mathbf{h}_c(t) = Q(t)\mathbf{H}_c(t)$, $\mathbf{H}_c(t) = \mathbb{I}_c \boldsymbol{\Omega}(t)$, and

⁴ The special orthogonal group of matrices in three dimensions $SO(3)$ is the set of matrices satisfying $Q^T Q = I$, $\det Q = +1$.

$\mathbf{T}_c(t) = Q^T(t)\boldsymbol{\tau}_c(t)$ is the body-fixed torque about the center of mass. If $\mathbf{M}_k(t) = Q^T(t)\mathbf{m}_k(t)$ is the body-fixed moment applied to wheel k , it follows that

$$\mathbf{T}_c(t) = \sum_{k=1}^{2n} (\mathbf{R}_k(t) - \zeta_k(t)\boldsymbol{\partial}_z) \times \mathbf{P}_k(t) - \mathbf{M}_k(t) \approx \sum_{k=1}^{2n} \mathbf{R}_k(t) \times \mathbf{P}_k(t) - \mathbf{M}_k(t),$$

if the (assumed small) suspension displacement $\zeta_k(t)$ are ignored.

The terrain height at the positions of the wheel centers are obtained from the OpenCRG⁵ function `crg_eval_xy2z`: $z_k^{CRG} = \text{crg_eval_xy2z}(\mathbf{r}_k)$, $k = 1, \dots, 2n$.

The difference between the wheel center height and the terrain height is used to calculate the deflection δ_k of the each tire (Figure 3) and, assuming a simple linear tire stiffness model, the vertical contact forces on the wheels:

$$\delta_k = \max(0, z_k^{CRG} + R - z_k), \quad F_k^Z = K_k \delta_k, \quad k = 1, \dots, 2n$$

The deflection is limited so that the tire only supports a compressive contact force ($\delta_k \geq 0$).

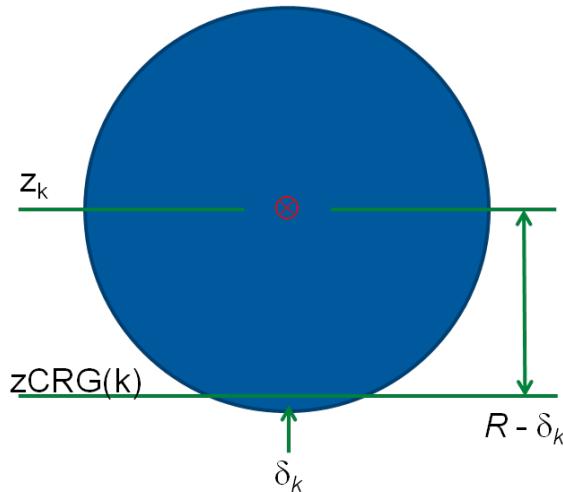


Figure 3. Calculating the tire deflection.

The wheel center velocities and the angular velocities of the wheels are combined to determine the slip velocity. Given a wheel center velocity of \mathbf{v}_k and a rotation speed of $\dot{\rho}_k$ about the body-fixed Y axis (spatial axis $Q(t)\boldsymbol{\partial}_Y$), the velocity \mathbf{v}_k^s of the tire contact patch center at $-RQ(t)\boldsymbol{\partial}_Z$ from the wheel center is

$$\mathbf{v}_k^s = \mathbf{v}_k + (\dot{\rho}_k Q(t)\boldsymbol{\partial}_Y) \times (-RQ(t)\boldsymbol{\partial}_Z) = \mathbf{v}_k - \dot{\rho}_k R Q(t) \boldsymbol{\partial}_X.$$

See Figure 4. In the body-fixed frame this becomes $\mathbf{V}_k^s = \mathbf{V}_k - \dot{\rho}_k R \boldsymbol{\partial}_X$.

⁵ OpenCRG website <<http://www.openerg.org/>>.

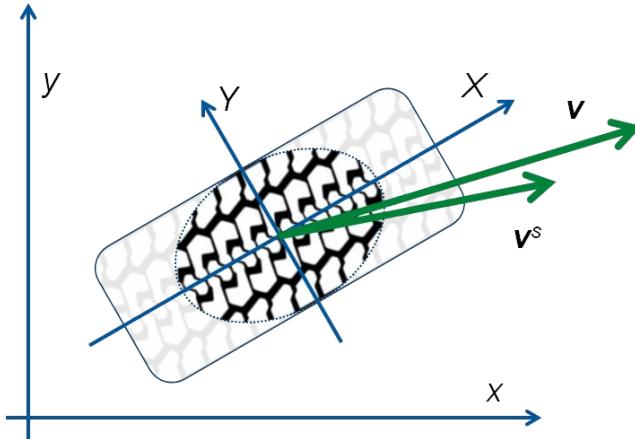


Figure 4. Calculating the slip velocity v^s .

The slip velocity components are thus $V_k^{sx} = V_k^X - \dot{\rho}_k R$ and $V_k^{sy} = V_k^Y$. Therefore the longitudinal slip is $\kappa_k = (\dot{\rho}_k R - V_k^X)/\max(|V_k^X|, |\dot{\rho}_k R|)$, and the slip angle is $\tan \alpha_k = -V_k^Y/V_k^X$. The longitudinal slip definition is modified from most of the references to account for slip without forward motion ($V_k^X = 0$) and locked wheel sliding ($\dot{\rho}_k = 0$); these modified expressions are used in the model. The Pacejka “magic formula” model⁶ uses the normal force, the longitudinal slip, and the slip angle to compute the tire forces and moments.

Once the tire forces and moments are determined, the suspension forces and moments may be approximated, and finally the accelerations of the vehicle body and the wheels may be calculated using the dynamical equations. In order to carry this out a form must be chosen for the rotation matrix; the choice here is $Q = Q_{yaw}(\psi) Q_{pitch}(\theta) Q_{roll}(\phi)$:

$$Q = \begin{bmatrix} \cos \psi \cos \theta & \sin \phi \cos \psi \sin \theta - \cos \phi \sin \psi & \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi \\ \sin \psi \cos \theta & \sin \phi \sin \psi \sin \theta + \cos \phi \cos \psi & \cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

where the roll, pitch and yaw angles are ϕ , θ , and ψ , respectively.

1.4 Important Findings and Conclusions

The model has been exercised to confirm that the amphibious vehicle can traverse straight, curved, banked and sloped terrain, individually or in combination. Certain limitations of the model have been noted: a tendency to oversteer with the low fidelity tire model and understeer with the Pacejka model. More research into tire models and their interaction with the vehicle dynamics would be advisable. Additionally implementing a fast and accurate tracked vehicle model would be highly useful.

⁶ Egbert Bakker, Lars Nyborg and Hans B. Pacejka, “Tyre modelling for use in vehicle dynamics studies,” SAE Technical Paper 870421, 1987, and H. B. Pacejka, *Tire and Vehicle Dynamics*, Butterworth-Heinemann, first edition 2002, second edition 2006, third edition 2012.

A number of internal variables, such as the tire and suspension forces, the acceleration of the center of mass, etc. should be saved to allow quantitative investigations of the vehicle motion.

1.5 Implications for Further Research

This model would be much more useful with a fast and accurate tracked vehicle option. Further work to make the existing model more robust would also increase its value. Additionally most automotive calculations are done using the SAE J670E surface vehicle standard for coordinate frame and rotation direction definitions; this differs from mine since the SAE J670E y axis is to the right and the z axis is down. Using the SAE J670E standard would make including models from the existing literature easier.

1 ECTo-VATR

1.1 Summary

ECTo-VATR is a system design tool that enables editing of the master model primarily through the hierarchical assembly and manipulation of components from the CML. It is focused primarily on empowering a designer in the early design phase to be able to incorporate and manipulate major design drivers and rapidly assess the qualities of system concepts. Resultant concepts can be used as the basis for more detailed design.

ECTo-VATR includes a 3D viewer called Zulu to represent the vehicle design concept and to aid in initial spatial layout and rudimentary packaging without the burden of a commercial CAD tool.

1.2 Launching ECTo-VATR

ECTo-VATR and Zulu are best run as client-side tools and running these on a remote machine will likely result in slow or degraded performance, especially the 3D visualization. ECTo-VATR and Zulu have currently only been built and tested on the Windows OS, however ECTo-VATR is based on the Qt framework and should be portable to linux or Mac without significant effort. Zulu is built using Unity which itself uses Mono and is supported only on Windows or Mac platforms.

ECTo-VATR normally runs in the following working directory: <installdir>\exec\ECTo-VATR\.

To run ECTo-VATR and Zulu (the 3D visualization) run the following bat files:

- <installdir>\exec\ECTo-3D\Zulu.exe (select the desired resolution and press ‘Play’ when the popup window comes up)
- <installdir>\exec\ECTo-VATR\startEcto.bat

1.3 Tool Overview

1.3.1 ECTo-VATR Main Window

ECTo-VATR is made up of several configurable panels. All but the system design panel are dockable widgets which can be undocked or moved around by dragging the title-bar of the panel. ECTo-VATR saves the layout configuration and window position when the program is closed and restores this when opened. This is saved in the layout.ini file which can be deleted or modified if desired. Initially there is no layout.ini file so the system comes up in a typical layout which looks something like the following:

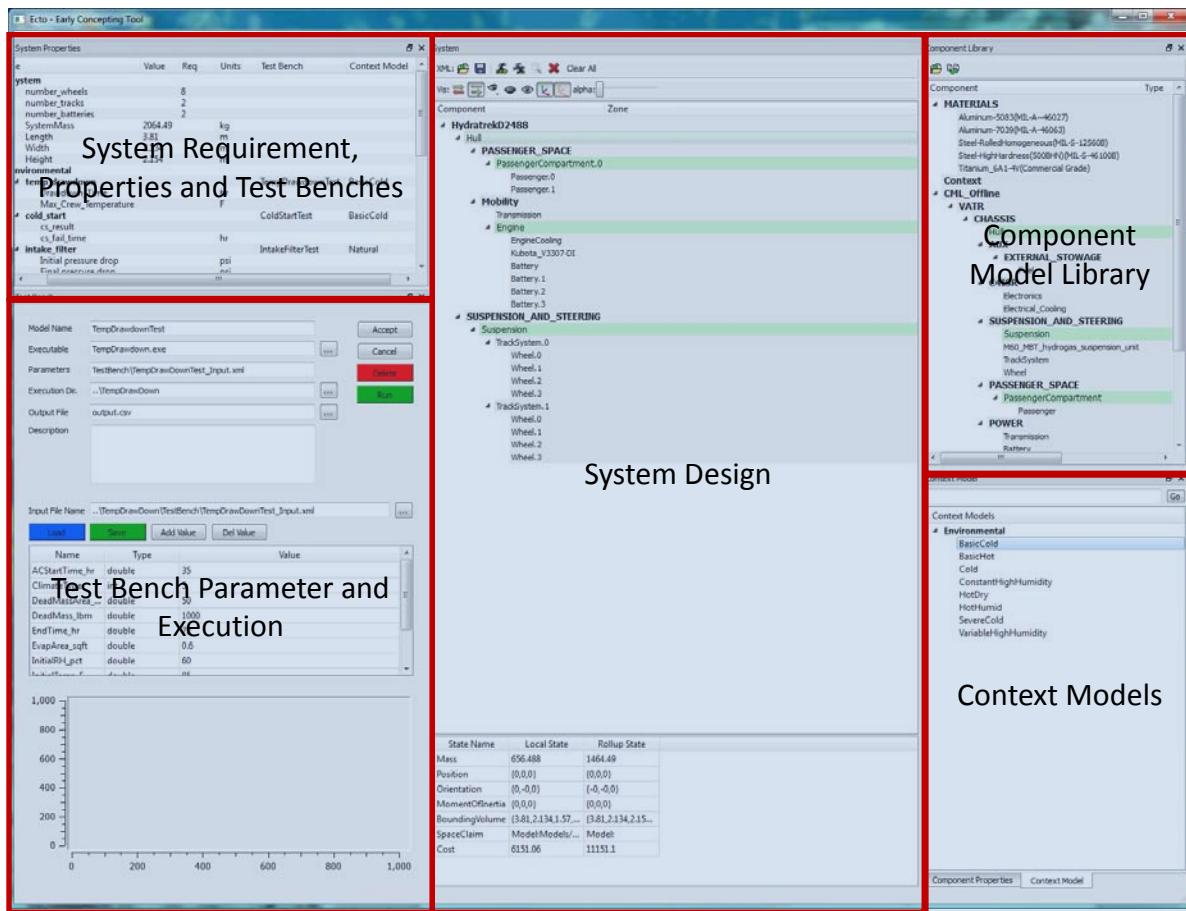


Figure 1 - ECTo-VATR Overview

By default when ECTo-VATR comes up the file system.xml is loaded. If you want to start from nothing you can press the clear all button.

1.3.2 Zulu

The 3D visualization component to ECTo-VATR, called Zulu runs as a separate process. No state is stored in the visualization and it is not required to be run for ECTo-VATR to function, though this provides the easiest way to move components around and get a spatial view of the system. If Zulu is started after ECTo-VATR or gets out of synch for any reason you can always press the 'Synch Vis with the system' button  located in the System Visualization Toolbar.

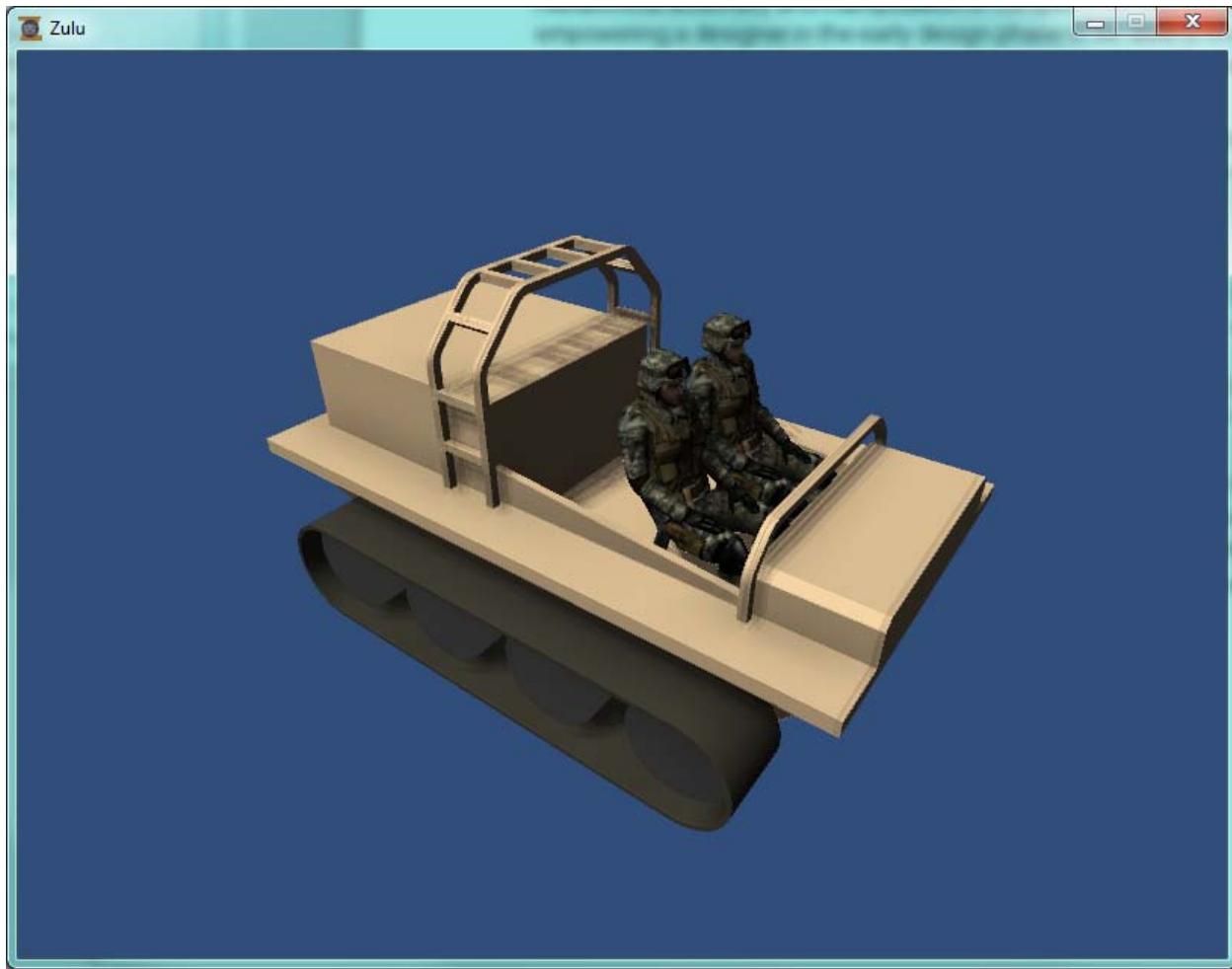


Figure 2 - Zulu (ECTo-VATR's 3D Visualization)

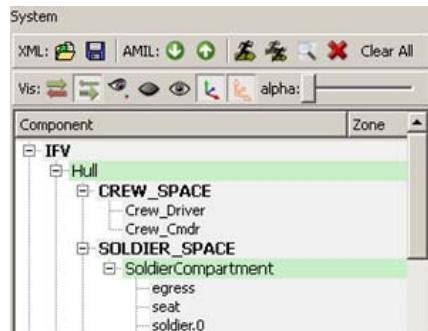
1.3.2.1 Zulu Camera Control

Zulu starts out in a free fly camera mode where by pressing the left mouse button and moving the mouse changes the direction of the camera. When the left mouse button is down the WASD keys translate the camera around the scene.

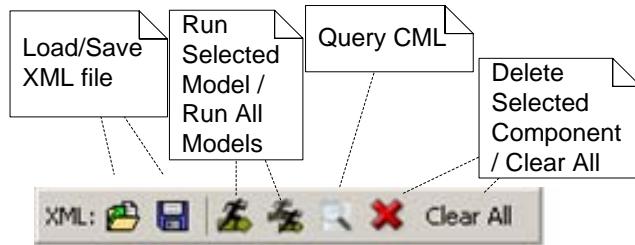
You can tether the camera to look at any node in your system by selecting a node and pressing the 'Tether Camera to Selected Node' button . Once tethered the camera spins around the selected

node by holding the left mouse button down and dragging it. Rolling the middle mouse button will zoom the view in/out. To go back to free fly mode press the ‘g’ key or ‘tab’ to get the Buttons Window and press the ‘Free Camera’ button under the Controls tab.

1.3.3 System Design panel



1.3.3.1 System Design Toolbar



XML load and Save buttons store/load the state of the system design using local XML files. Loading a new system files clears the existing system.

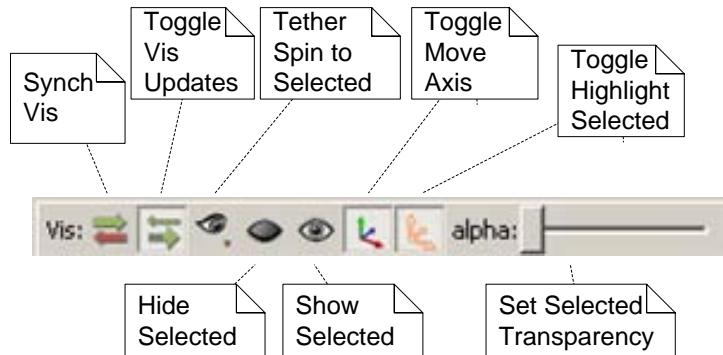
Run Selected Model/Run all Models: ECTo-VATR models can be run explicitly by selecting a model or more commonly all at once by pressing the ‘Run All Models’ button in the System Design Toolbar. Run all models execute all of the models in the System Hierarchy. See ECTo-VATR Models section below for more information.

Query CML creates a SPARQL query for refinements of the selected component. It generated as query for things in the ontology that are valid refinements of abstract item selected. Note for the demo as delivered the only component that had the CML populated for it to illustrate how this works is the ‘Engine’. See the Demo walkthrough below for an example.

Delete Selected Component: removes the selected component *and all its children* from the system hierarchy.

Clear All: deletes the entire system hierarchy.

1.3.3.2 System Visualization Toolbar



The synch vis button clears what is in the Vis and resends the entire system design. No state is ever persisted in the visualization so it is generally ok to do this any time. Once synched the vis typically should stay in synch as changes are made. Zulu can be closed and restarted anytime and the synch can be used to re-establish the 3D representation of the system in ECTo-VATR.

Toggle Vis Updates is enabled by default and generally does not need to be changed. When enabled tells the Zulu visualization to automatically send positional updates back to ECTo-VATR when things are moved around. A designer may want to disable this if they want to manipulate the 3D visualization without changing the system.

Hide selected tells the visualization to not render the component in the visualization. This is only a visual preference and does not affect the system design and is not saved in XML or AMIL.

Show selected turns a previously hidden component visible again.

Toggle Move Axis turn on/off the Axis control in Zulu which allows you to move or rotate a component. Toggle Highlight Selected turn on/off the highlighting of edges around the component which is currently selected.

Set Selected Transparency makes the selected component's alpha value so it becomes transparent.

1.3.3.3 System Hierarchy

This is where the hierarchical representation of the system is displayed. You can drag and drop components around.

As you select components in the tree you will see the values of the major states displayed in a table under the system hierarchy tree. The 'Local State' column is the state values for *only the component selected*. The 'Rollup State' column displays a rollup of values for the selected state and all that state's children. System level rollups for Mass, Cost, Length, Width and Height are always displayed in the System Properties panel.

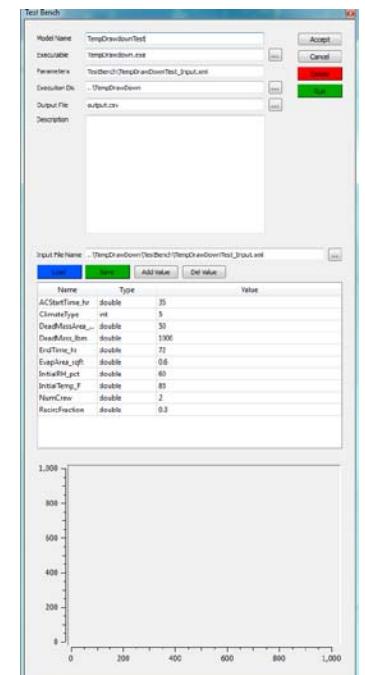
1.3.4 System Requirements and Properties Panel

This panel stores all the system level properties or any properties that need to be exchanged between component models. The ‘Value’ column indicates the current systems estimated performance and the ‘Req’ column is used to maintain the system requirement or derived requirements for that property. Most key system inputs are considered requirements but they can be adjusted by a designer so they can assess how particular inputs drive a design. The test benches are seen in the ‘Test Bench’ column and they are selected by double clicking the text. When a context is selected for the current test bench it is seen in the ‘Context Model’ field.

Name	Value	Req	Units	Test Bench	Context Model
System					
number_wheels	8				
number_tracks	2				
number_batteries	2				
Environmental					
temp_dropdown					
Max_Crew_Temperature	hr	F			TempDropdownTest
cold_start					
c_cool					
c_fall_time	hr				ColdStartTest
intake_filter					
Initial pressure drop	psi				
Final pressure drop	psi				
Final flow	cfm				
heat_exchanger					
Tin/Out	C				HeatExchangeTest
Transmission	C				

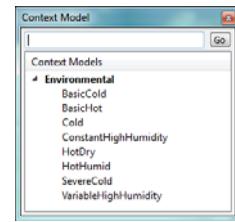
1.3.5 Test Bench Parameters and Execution Panel

This panel shows and allows one to edit the parameters for the current test bench. The panel is broken down into three sections. The top section is the information needed to run the test bench executable. The middle section is the input parameters for the test bench. The bottom section is the output graph that was generated during the execution of the test bench. To run the test bench the user needs to click on the Run button. If the execution of the test bench takes a long time the panel will display a progress indicator.



1.3.6 Context Model Panel

This panel shows the available context models for the current test bench. Double clicking the text will select which of the context that is used when the next run of the test bench. ECTo-VATR will automatically change the parameters in the current test bench that corresponds to that context.



1.3.7 Component Panel

This panel shows and allows one to edit the properties of a specific component. Components with associated models will frequently have additional properties that are specific to that model. For example, the PassengerCompartment model uses additional properties to construct the soldier compartment such as seating arrangement, squad size assumptions, clearances, etc.

Component Properties	
Property	Value
objectName	
type	
x	-6.7
y	0
z	0
h	0
p	0
r	0
SeatingConfiguration	Inline
ManSizeConfiguration	LargeMale
HeadClearance	0.0508
SeatThickness	0.0254

1.3.8 Component Model Library Panel

The Component Model Library panel shows components from the CML which can be included into the system hierarchy by dragging a component from this panel to the parent in the system hierarchy you want to attach this component to. Alternatively, you can select a parent in the system hierarchy and select a component in the CML panel and press the Copy Selected Component to CML button .

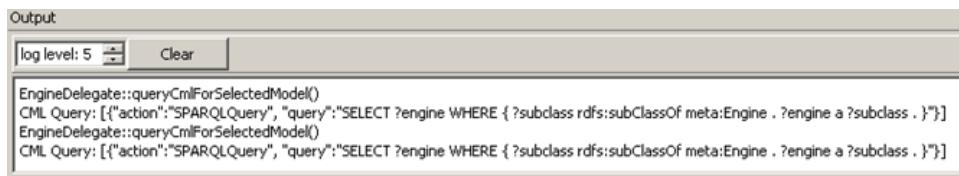
Components can be loaded into this panel either through XML files using the load file button .

As additional ways to interact with and search a CML are established this panel can evolve to support additional approaches. Additionally, CML repositories should be able to easily export indexes of components into a compatible XML file. The file CML.xml in ECTo-VATR's working directory provides an example of various CML components used to build up the IFV in the demo walkthrough below.

Component Library	
Component	
Query Results	<ul style="list-style-type: none"> LM500 GE_T700-T6E1 GE_T700-T6A1 GE_T700-701C AGT1500_turbine_engine Detroit_Diesel_S71_12V-71TA Detroit_Diesel_S92_8V-92TA Detroit_Diesel_S53_6V-53T Perkins_Phaser_180_Ti Perkins_Phaser_110_T MTU_MB_871-Ka501 MTU_MB_873-Ka501 Renault_MIDR_06-20-45 Renault_MIDR_06-02-26 NORINCO_X150-960 NORINCO_12150L-7BW NORINCO_12150L
MATERIALS	<ul style="list-style-type: none"> Aluminum-5083(MIL-A-46027) Aluminum-7039(MIL-A-46063) Steel-RolledHomogeneous(MIL-S-12560B) Steel-HighHardness(500BHN)(MIL-S-46100B) Titanium_6Al-4V(Commercial Grade)
Context	<ul style="list-style-type: none"> Tunnels <ul style="list-style-type: none"> Tunnel_NATO_M Tunnel_AAR Tunnel_GIC Tunnel_DOD Flat_Rail_Car
CML_Offline	
IFV	<ul style="list-style-type: none"> CHASSIS <ul style="list-style-type: none"> AUX <ul style="list-style-type: none"> EXTERNAL_STOWAGE Fuel

1.3.9 Tool Log

The tool log provides informational output to the user as the tool is used. The output level can be adjusted so more or less output is displayed in this panel. The clear button wipes the current contents away.



1.4 ECTo-VATR Models

Certain nodes, indicated by having a green background in the System Hierarchy, have Models associated with them. These are typically parametrically driven components or systems that change their properties or are ‘built’ based on inputs to those models. For example the ‘PassengerCompartment’ Model, when executed takes the requirement from the System Properties panel for ‘number_passengers’ and various other inputs in the Component Properties panel to construct the space claim and mass for a the Passenger Compartment. If you change the number_passengerss requirement and run this model you will see the number of soldiers and the various properties of the compartment, bench, egress volumes, etc. change to reflect this.

ECTo-VATR models are run from the System Design Toolbar, either individually on a selected model or by executing all models in the system. Running all models traverses the tree and executes models from children models up, models under the sub-tree of another model node are executed before the ancestor node’s model. Executing Models can retrieve and set data in the System Properties widget as they run, so one models output can be used as input or modified by another model. This data flow can also be used to control the execution order of models or to establish an implicit causal network of models.

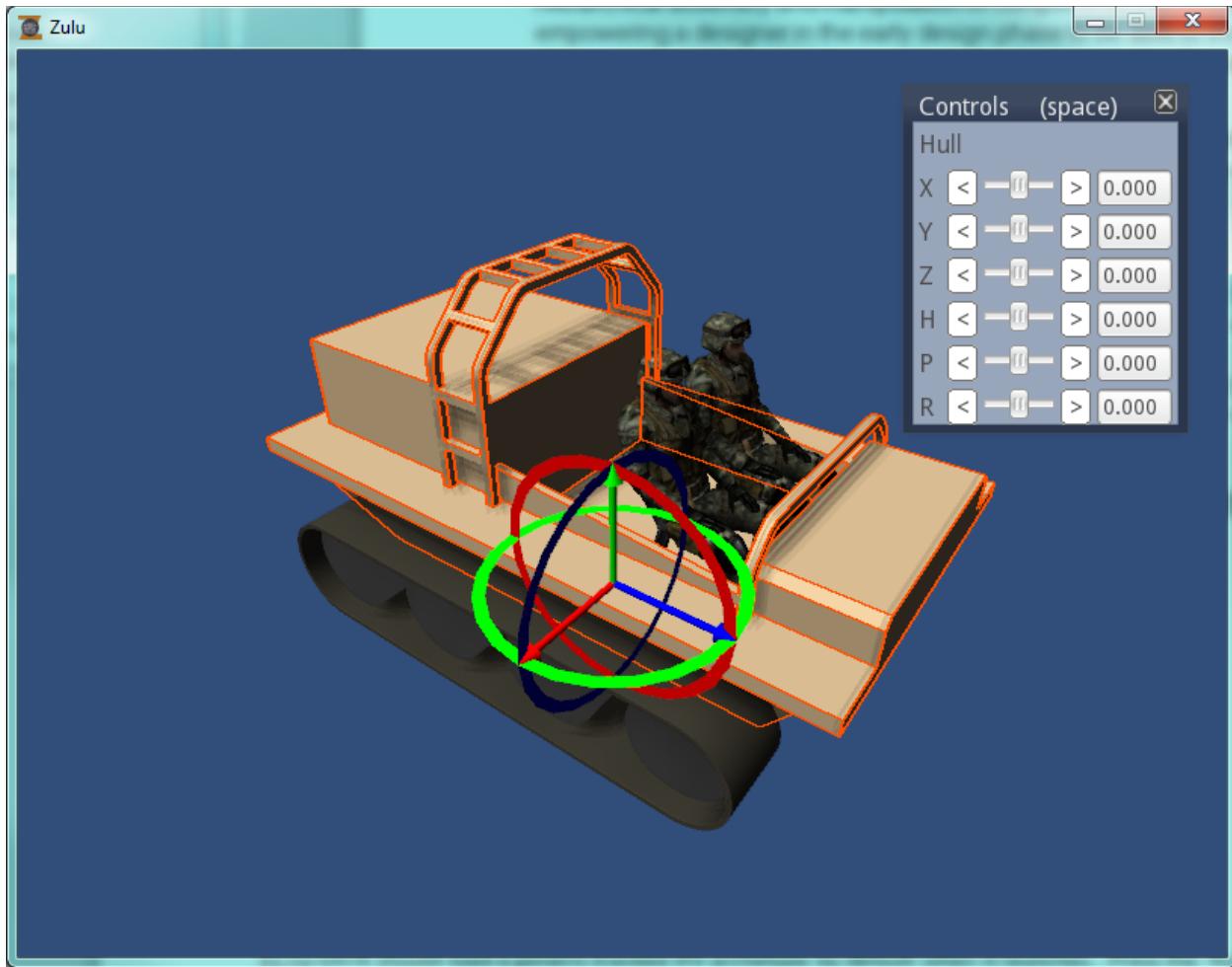


Figure 3 - Hull Positioning

1.5 Example Walkthrough

The following demo walks through an example of how ECTo-VATR could be used to quickly assess different system concepts and how that system operates in against a set of test benches.

1.5.1 Load Tracked VATR archetype

ECTo-VATR should load a generic tracked VATR archetype by default when it launches. Press the ‘Run All Models’ button in the System Design Toolbar.

1.5.2 Synch with 3D Vis (Zulu) and system properties

If Zulu does not show current vehicle press ‘Sync Vis with System’ button

1.5.3 Modify system requirements (increase number squad from 6 to 9) and execute all models again (rebuild system)

Change number_passengers from 4 to 2 in the Systems Properties panel.

Press the ‘Run All Models’ button in the System Design Toolbar.

1.5.4 Observe quick look mobility model and Engine Power derived requirement

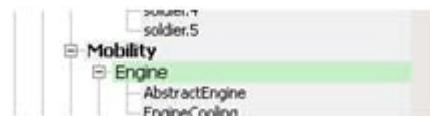
In the System Properties panel, as you modify the vehicle a quick look calculation is made that generates a minimum Engine Power that would be required for a vehicle of this weight class to achieve the required maximum speed (as show in the System Properties as top_speed). If you modify the system to make it heavier the required Engine Power will go up. Additionally, if you modify the top_speed requirement in the System Property panel this will also change the minimum required Engine Power. This calculated, or derived, requirement for minimum Engine Power is used when you query for viable refinements of Engine.

One way to quickly change the weight of the vehicle is to modify the ‘AvgThickness’ component property for the ‘Hull’ Model component. You can change the average hull thickness from 0.02 m to 0.03 and run the Hull model (or Run All Models) to represent a heavier armored vehicle.

1.5.5 Replace abstract engine with specific engine

Delete ‘AbstractEngine’ by selecting it in system hierarchy and pressing ‘Delete Selected Component’ button.

Drag an engine from the CML to the ‘Engine’ model component.



Press the ‘Export to AMIL’ button to save change to AMIL graph (Master Model).

1.6 Building ECTo-VATR

ECTo-VATR is built on top of the Qt framework and should be able to be compiled and run under windows/linux/Mac though only project files are supplied for Visual Studio.

Building using Visual Studio for Windows (XP or newer)

Requires:

- Visual Studio 2010
- Qt 4.5 or newer built for the version of Visual Studio you are using
- Visual Studio Qt addin (tested using version 1.1.9)

Open <svnroot>\VATR\VATR_VS2010.sln in Visual Studio 2010 and build release configuration.

1.7 Developers notes

ECTo-VATR is written in C++ and built on top of the Qt application framework. Zulu is built using the Unity Engine and communicates with ECTo-VATR using UDP messages.

1.8 Path Forward

While useful in its current state, ECTo-VATR is a prototype of an early system conception, analysis and test tool and has the potential to evolve into a more powerful system engineering asset.

Architectural modifications

- Migrate VATR specific delegates and models into dynamically loadable libraries, these libraries can then be stored and brought into ECTo-VATR from the CML.
- Include Ability to do multidimensional sweeps across input values or enumerated sets. Add output tables and data export utilities to facilitate analysis of this data.
- As statistical models are incorporated allow for execution of Monte Carlo runs.
- Fully incorporate a Design Set node that can be used to represent alternatives and any level in the hierarchy and tools.
- Enhance the System Properties panel to allow for a designer to adjust values and still maintain original requirements perspective. Could add a third column for actual requirement, a column for value to use as system input and a value for actual estimated output.
- Expand and make it easier to use server based context models such as those provided by OSCAR, the Dynamic Context Server and the Terrain Server.

ECTo-VATR then becomes more useful and powerful as more models are added and integrated together and the fidelity and accuracy of the models are improved. While the example models and analysis that currently exists in ECTo-VATR are related to the sample VATR with some relatively minor changes the underlying infrastructure can support the design of many types of systems.

LIST OF EXECUTABLE STAND-ALONE EXAMPLES

This section describes context models use with stand-alone executable vehicle simulations. The simulations were developed in Matlab, C++, Excel, either directly linked to context models, or run-time linked via FMU interfaces.

COLD START DEMO

This simple code demonstrates the usage of the environmental context models for C2M2L-1. In particular, the AR-70-32 environmental specification is used to evaluate an abstract vehicle design.

Models/Composite/FluidThermal/doc/Cold Start Demo Notes.docx

RUN SIX WATT STAND ALONE EXAMPLE

The run six watt stand alone example is a demo example that connects to the terrain server and pulls a synthesized terrain profile for simulation analysis. The terrain examples all employ a terrain generator shown in Figure 1.

Models/Land/terrains/grades_slopes/doc/Demo_RunSixWattStandAlone.docx

DISCRETE OBSTACLES EXAMPLE

The Discrete Obstacles Pitch Plane demo example demonstrates the use of discrete obstacle context with a simulation model to determine the go/no-go performance of the design.

Models/Land/obstacles/doc/Demo_DiscreteObstaclesPitchPlane.docx

DRIVER SPECTRA AND OBSTACLES EXAMPLE

The driver spectra and obstacles example demonstrates the use of a spatial design model with suitable spatial context.

Models/Land/terrains/grades_slopes/doc/Demo_DriverSpectraNObstacles.docx

DRIVER LIMITED SPEED SPECTRA EXAMPLE

The Driver Limited Speed Spectra demo example demonstrates the generation of a synthetic terrain for simulated model crossing at speed.

Models/Land/terrains/grades_slopes/doc/Demo_DrvLmtdSpdSpctr.docx

FUEL EFFICIENCY EXAMPLE

The Fuel Efficiency demo example demonstrates the generation of a synthetic terrain for use in a fuel efficiency calculation for design evaluation.

Models/Land/terrains/grades_slopes/doc/Demo_Fuel_Efficiency.docx

Appendix O: Standalone Executable Examples

AMPHIBIOUS CONTEXT MODEL

To determine the reserve buoyancy, range and stability of an amphibious vehicle within a water condition an example vehicle was chosen.

Models/Aquatic/properties/buoyancy/Summary of Amphibious Context Model.docx

SOIL TRACTION EXAMPLE

Demonstrate how a soil cone model can be used to determine the soil cone index with a given humidity value, along with a simple application to demonstrate how a mobility model can be used to determine the rolling resistance and the theoretical max speed of a vehicle.

Models/Land/doc/SoilConeIndexDemo.docx

Models/Land/doc/MobilityDemo.docx

GENERATING OBSTACLES AND TERRAINS EXAMPLES

The generation of discrete obstacles parameterized with static friction and other properties is demonstrated for two specific geometries, along with a random terrain generator.

Models/Land/obstacles/doc/DiscreteObstacleDemo.docx

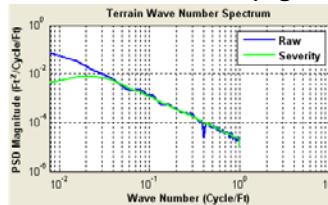
Models/Land/obstacles/doc/AngleDitchCrossingDemo.docx

Models/Land/terrains/grades_slopes/doc/TerrainGeneratorDemo.docx

COMMON FEATURES

Several of the examples use PSD representations to synthesize terrain (see Figure 1). For discrete obstacles, deterministic profiles were used.

**Power Spectral Density representation
of Terrain generated from Raw
Measurement Data or “Mined” from
Unclassified Sources (e.g., TOPS)**



**PSD used to generate
Synthetic Terrain (X,Z)**

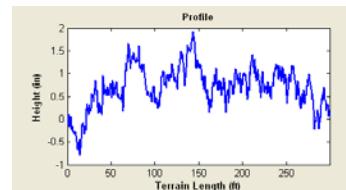


Figure 1: PSD to terrain generation

COLD START DEMO

This simple code demonstrates the usage of the environmental context models for C2M2L-1. In particular, the AR-70-32 environmental specification is used to evaluate an abstract vehicle design. The situation is similar to a typical US military vehicle requirement verification test. The vehicle is conditioned to temperatures corresponding to a climate type indicated in AR-70-32 and starting attempted. In the verification test, the vehicle must be “combat ready” within a certain time limit. After the vehicle starts, the engine warms up to an operational threshold, the electronics and computer systems are booted and ready. In this example, the “vehicle” is deemed able to start and combat ready if the available battery power is equal to or greater than the power needed to turn over the engine.

LIMITATIONS OF THE COLD START DEMO CODE:

In general, the code is in the alpha stage. Limited testing has occurred but is has not been finished on all aspects of the models. Documentation is incomplete. Some inputs are parameterized in an input file, but the vehicle architecture is hard-coded and unalterable except through recompilation of the code. The code was developed using the Microsoft Visual C++ 2010 compiler. Some of this code uses features of the C++ 11 language standard not available in earlier compiler versions.

The vehicle is abstract. Nothing is assumed or modeled about the internal heat generation of components due to running the engine, engine heaters, or other sources. The vehicle model is a simple thermal network solver that uses lumped-mass approximations for the individual components. The vehicle data is local to the simulation. The vehicle behavior is local to the simulation. Vehicle data and behavior will ultimately be defined within META.

The thermal network solver used to define the vehicle’s response to the environment is limited in functionality. A 4th order Runge-Kutta solver is used, but with a fixed time step. Full vehicle models would be sufficiently complex that an automatically adjustable time step solver would be preferred to keep the simulation time to a reasonable limit. The thermal network solver does not yet poses a feature to allow component heat generation, a critical element in the thermal architecture of a vehicle. The thermal network solver does not possess interfaces for more complex components. For example, a 1D or heat conduction model of an insulated vehicle wall would be useful in some situations.

Code for the thermal and fluid context models included, but the feature set is incomplete. Humidity information, which is specified in AR-70-38 is not yet available.

The Alglib math library used in this example has been replaced in more recent versions of the fluid and thermal context models.



Figure 2: Cold Start GUI

RUN SIX WATT STAND ALONE EXAMPLE

The run six watt stand alone example is a demo example that connects to the terrain server and pulls a synthesized terrain profile for simulation analysis. The demo example accepts an input file describing the terrain spectra to be analyzed and some user configurable design model parameters. The simulation model is repeatedly exercised across the terrain until a limit speed is reached. The limit speed is either a six watt average absorbed power limit or an upper bound speed set in the code. The simulation outputs two files, one providing chart data showing driver absorbed power as a function of terrain crossing speed, the other listing the driver absorbed power and terrain rms value. The terrain server must be started prior to calling the six watt stand alone model. The terrain server window is shown below.

Appendix O: Standalone Executable Examples

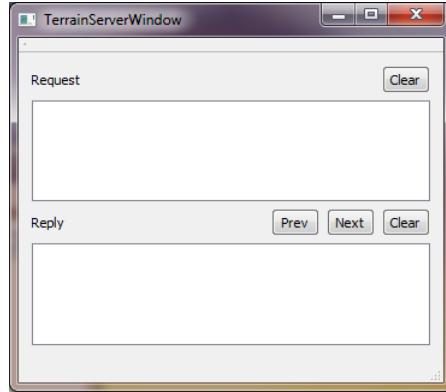


Figure 3

The call to the six watt stand alone demo example accepts input arguments. The first input argument is taken to be the model input file. A sample input file is shown below. The model input is kept relatively simple for this demo. The parameters listed below from ServerHost to seed are specifications for the call to the terrain server to retrieve a terrain profile. The terrain profile requested in this case is spectral representation number 20 "Perryman 3", with a synthesized length of 30 meters at 0.025 meter resolution. Additionally, the context model is seeded with a seed value of 1. The sprungmass and wheelposition values are design model parameters. The design model in this case consists of a 1000 kilogram sprung mass supported by three wheels on each side. The simulation model is only in the pitch plane for this example as the point is to illustrate the effect of design and context model changes on performance.

```
<?xml version="1.0" encoding="UTF-8"?>

<ModelInput>
  <ServerHost string_val="127.0.0.1"/>
  <ServerPort int_val="8080"/>
  <spectral int_val="20"/>
  <length double_val="30"/>
  <int double_val="0.025"/>
  <seed int_val="1"/>
  <SprungMass double_val="1000"/>
  <WheelXPosition.0 double_val="1.67258"/>
  <WheelXPosition.1 double_val="2.67258"/>
  <WheelXPosition.2 double_val="3.67258"/>
</ModelInput>
```

The simulation model exercises the design model repeatedly across the terrain profile at increasing speeds. At the end of each crossing, the driver average absorbed power is examined and recorded. The speed is increased until the limit speed is achieved. The results are placed in two output comma separated value files. The names of the files can be passed as the optional second and third arguments to the model call.

Influences of either design changes or context model changes can be studied and quantified by changes to the input file. Suggested changes for the context include the selection of different terrain spectra profiles, different terrain lengths, and different terrain seeds. Potential changes for the design model include changes to the sprung mass, the number of supporting wheels, and the spacing between the supporting wheels.

DISCRETE OBSTACLES EXAMPLE

The Discrete Obstacles Pitch Plane demo example demonstrates the use of discrete obstacle context with a simulation model to determine the go/no-go performance of the design. Course profiles and discrete obstacle profiles are loaded individually and the simulation model is exercised at a user-controlled speed across the terrain. Success or failure to cross the terrain profile is observed from the simulation graphic.

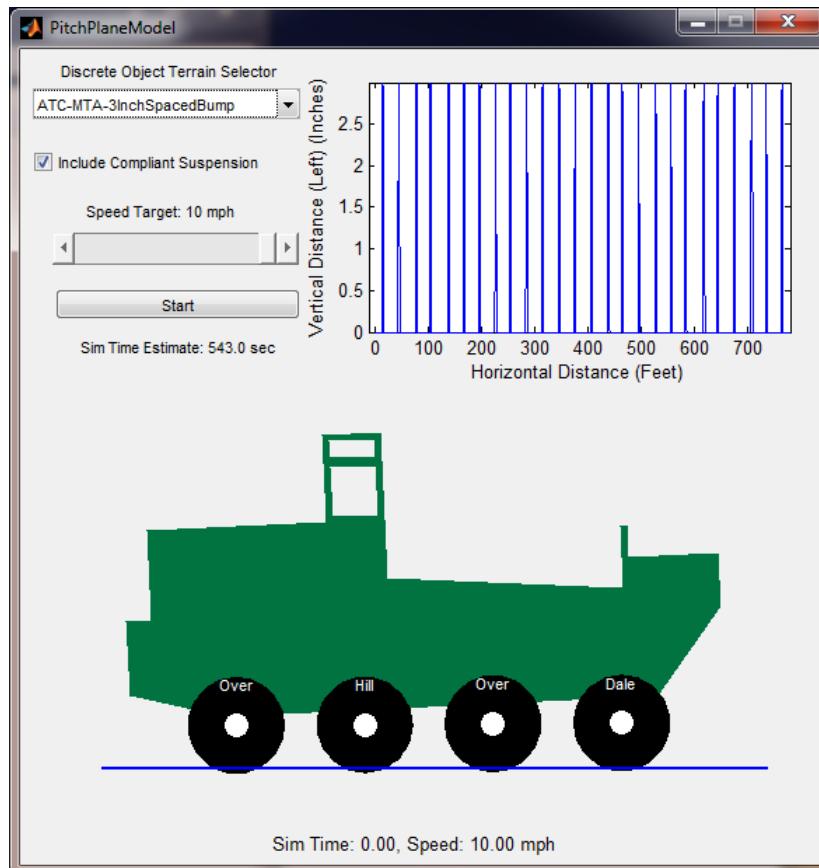


Figure 4

The demo GUI, shown above, consists of one plot, the simulation graphic, and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is a multiple degree-of-freedom pitch plane model that queries the terrain at multiple points beneath the wheels. This design model terrain contact interface is not constrained to the terrain and can separate.

The discrete object terrain selector in the upper left corner select terrain context from the available courses and profiles and loads them individually into the simulation model. The include compliant checkbox changes the simulation model from having a compliant suspension when checked, to a non-compliant suspension when unchecked. Performance differences can be observed in the simulations to the selection of compliant and non-compliant. This option is not selectable during run-time. The simulation speed target is user-adjustable before and during run-time. The simulation start pushbutton starts the model execution. The model execution will stop after an estimated terrain crossing time is reached or when the user presses the simulation start pushbutton again.

The simulation graphic and terrain profile plot are updated continuously during the simulation. The terrain profile displays a windowbox to indicate the current portion of the terrain profile that the simulation model is in. This is the terrain portion shown in the simulation graphic window on the bottom. The design response to the terrain context can be viewed here as the simulation progresses. Failure to cross/traverse an obstacle/terrain profile is observed as a stagnant design model, or a design model in an undesirable configuration (such as inverted).

DRIVER SPECTRA AND OBSTACLES EXAMPLE

The driver spectra and obstacles example demonstrates the use of a spatial design model with suitable spatial context. The demo example can use either discrete obstacles, profiles, or generate synthetic terrain from the spectral representations. The demo uses context in a manner similar to the Driver Limited Speed demo example when selecting terrain spectra. The demo example generates synthetic terrain with wavenumber content from the reference spectra profile. The demo just loads up the obstacle or course profile content when selecting a discrete obstacle or profile, similar to the Discrete Obstacles Pitch Plane demo model. The key difference is the use of a spatial design model to assess the performance across the context.

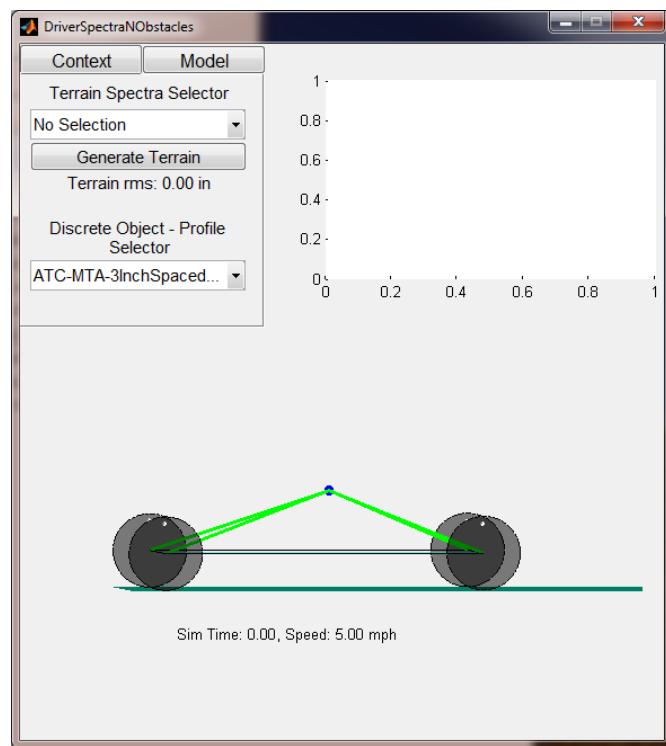


Figure 5

The driver spectra and obstacles gui, with initial appearance shown above, consists of context and model control panels, a context profile window, and a simulation graphic. The context panel, displayed in above graphic, permits the selection of either terrain spectra or discrete object profiles. If a spectral terrain representation is selected, terrain must be generated before continuing to the model panel for simulation. The model panel contains the controls for interaction with the design model, displayed in the figure below. The include compliant checkbox changes the simulation model from having a compliant suspension when checked, to a non-compliant suspension when unchecked. Performance differences can be observed in the simulations to the selection of compliant and non-compliant. This option is not selectable during run-time. The simulation speed target is user-adjustable

Appendix O: Standalone Executable Examples

before and during run-time. The simulation start pushbutton starts the model execution. The model execution will stop after an estimated terrain crossing time is reached or when the user presses the simulation start/stop pushbutton again.

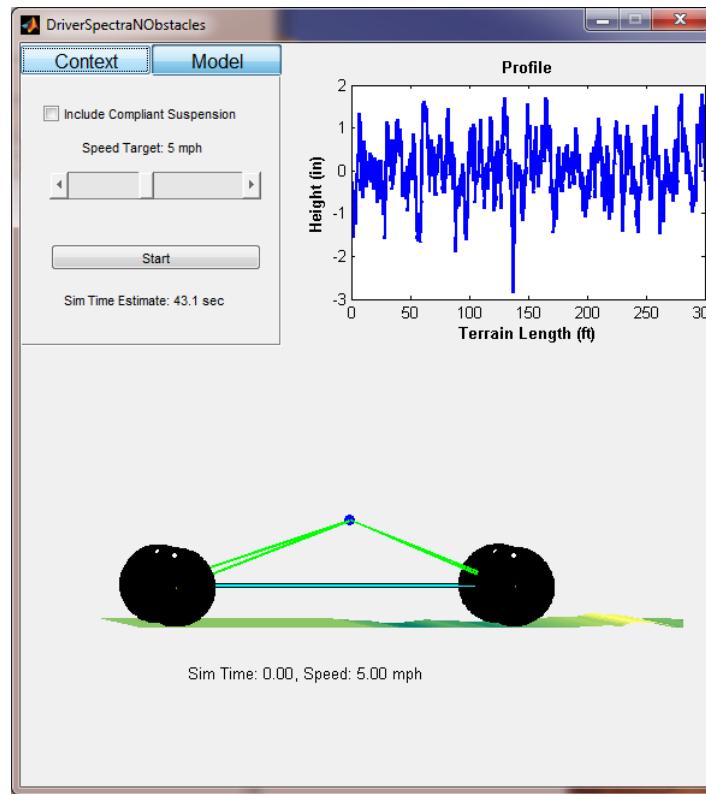


Figure 6

The simulation graphic and terrain profile plot are updated continuously during the simulation. The terrain profile displays a windowbox to indicate the current portion of the terrain profile that the simulation model is in. This is the terrain portion shown in the simulation graphic window on the bottom. The design response to the terrain context can be viewed here as the simulation progresses. Failure to cross/traverse an obstacle/terrain profile is observed as a stagnant design model, or a design model in an undesirable configuration (such as inverted).

DRIVER LIMITED SPEED SPECTRA EXAMPLE

The Driver Limited Speed Spectra demo example demonstrates the generation of a synthetic terrain for simulated model crossing at speed. This demo illustrates the coordination of context modeling and system modeling for simulated evaluation of design performance against the requirement of driver limited speed. The driver limited speed requirement is the evaluation of the average absorbed power to the driver for crossing severe terrains at speed. There exists a power limit at which the driver will slow down for a smoother ride versus subjecting him/herself to excessive vibration. The limiting speed is typically termed 6 watts and the design will likely be required to cross a specific severity terrain at a speed to keep combat formation.

Appendix O: Standalone Executable Examples

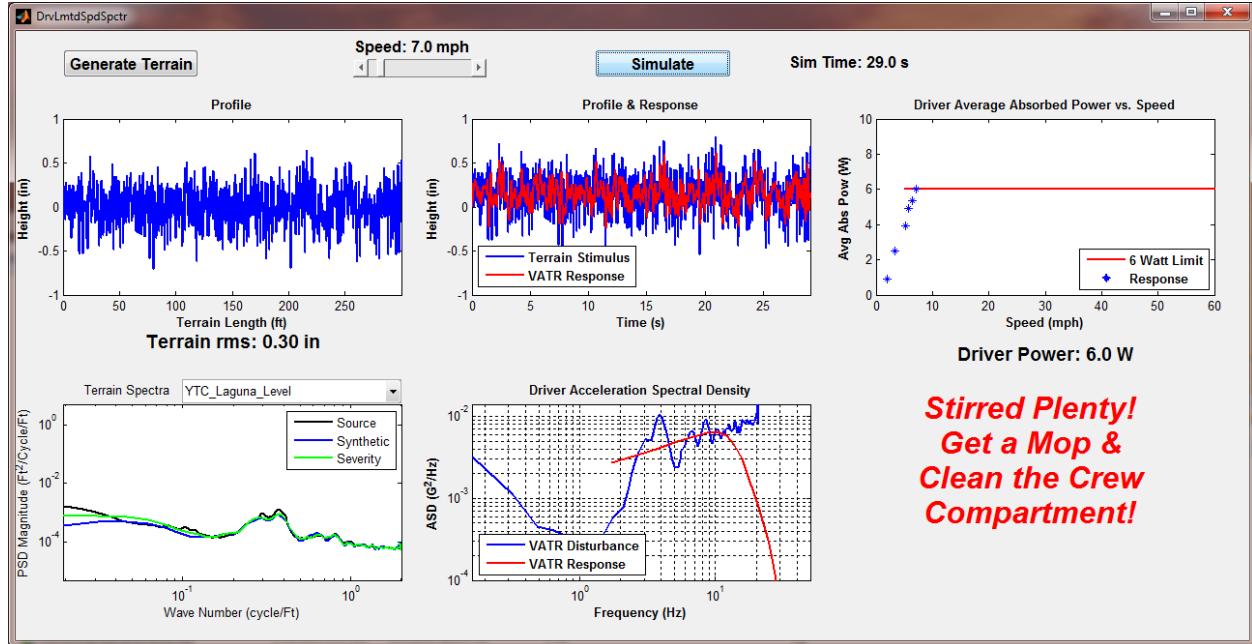


Figure 7

The driver limited speed demo GUI, shown above, consists of five plots and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is a single degree-of-freedom sprung mass model that queries the terrain at a singular point. This design model terrain contact interface is constrained to the terrain and does not separate.

Terrain spectra is selected in the lower left pull-down menu from the terrain spectra JSON files in the data directory. The “Generate Terrain” pushbutton synthesizes a 300 foot terrain length containing spectral wavelength content shown by the blue curve in the lower left plot, with similar content to the reference black curve. This range of wavelength content represents the wavelength relevant to terrain severity calculation. The terrain severity is shown by the “Terrain rms” value between the upper and lower left plots. This particular example synthesizes terrains within that rms range and contains spectral content along the logarithmic line shown in the lower plot. Note that the terrain profile consistently contains similar spectral content, yet is not unique in the terrain profile plot with each press of the “Generate Terrain” pushbutton. The green plot in the wave number spectrum represents the detrended terrain content that is used in the severity calculation (the Terrain rms number).

The synthetic terrain shown in the upper left is used as context input to the simulation model. The “Speed” slider control adjusts the model’s speed of crossing the terrain profile. The terrain profile is turned into model stimulus by crossing the terrain at the specified speed.

The “Simulate” pushbutton simulates the system response to the context model with the stimulus and response shown in the central top plot in the GUI. The driver acceleration spectral density and corresponding stimulus input is shown below that. The average absorbed power to the driver is then calculated and plotted as a data point in the right plot for comparison to the six watt limit. The intent is to adjust the driven speed to find the limit speed at which the six watt line is crossed. Suggestions to speed adjustment are provided in the lower right, below the most recent simulated driver absorbed power value.

FUEL EFFICIENCY EXAMPLE

The Fuel Efficiency demo example demonstrates the generation of a synthetic terrain for utilization in a fuel efficiency calculation for design evaluation. This demo shows the coordination of context modeling and system modeling for simulated evaluation of design performance against the requirement of vehicle range with on-board consumables. The vehicle range is specified in program requirements and is limited to on-board fuel as the consumable source. The evaluation of compliance to this requirement is a necessary design effort. This demo illustrates the coupling of context and system modeling with a simple arithmetic design model.

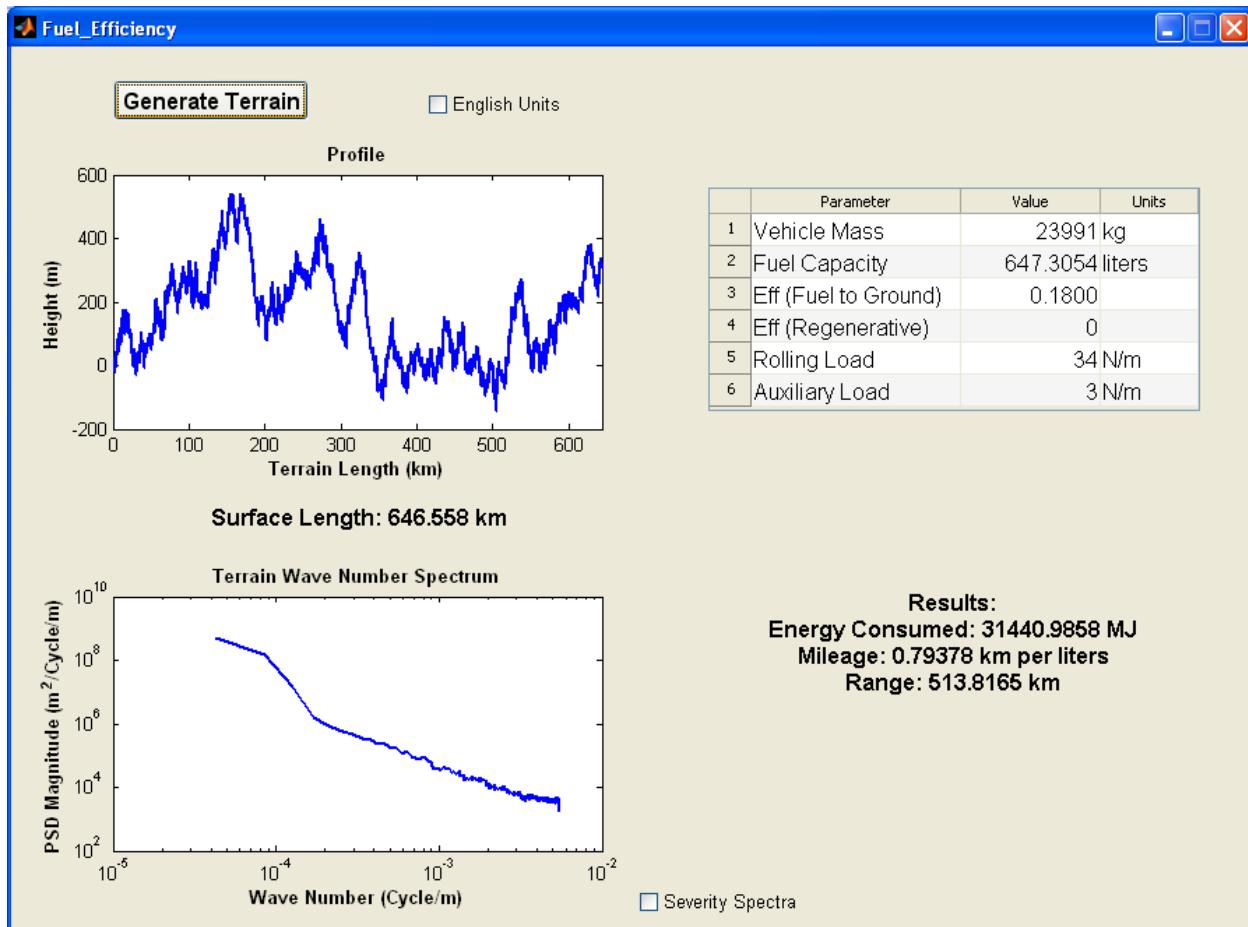


Figure 8

The fuel efficiency demo GUI, shown above, consists of two plots, a results statement, and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is an arithmetic calculation based on the synthetic terrain and design parameters.

The “Generate Terrain” pushbutton user interface control synthesizes a 650 km terrain from a probabilistic density function fitted to CONUS DEM data. This is the context model. It is displayed in the upper left plot as a terrain profile. It changes with every press of the “Generate Terrain” button from a series of randomly generated seed values. The terrain is synthesized from a random walk on the probability density function.

Appendix O: Standalone Executable Examples

Design model influential characteristics of the terrain profile are extracted and design efficiency results are displayed in the lower right. The energy consumed, mileage, and design range are shown for the design parameters in the upper right. The energy content of the fuel is coded into the example for the results calculations.

The wave number spectrum is shown in the lower plot for continuity with the driver limited speed example. The wave numbers are limited to long wavelengths that represent large changes in slope and elevation. This plot introduces users to the concept of wave spectrum for terrains.

A units checkbox is provided to express the model in English or Metric units. When the model is in the English units configuration, the “Severity” checkbox will permit the severity spectra of the generated terrain to be displayed (green plot). This capability illustrates separation of the wave number ranges for severity and fuel efficiency. The relative intensity of the severity spectra is substantially lower than the synthetic terrain generated.

Model parameters can be set in the upper right dialog to determine the influence and sensitivity of the model parameters.

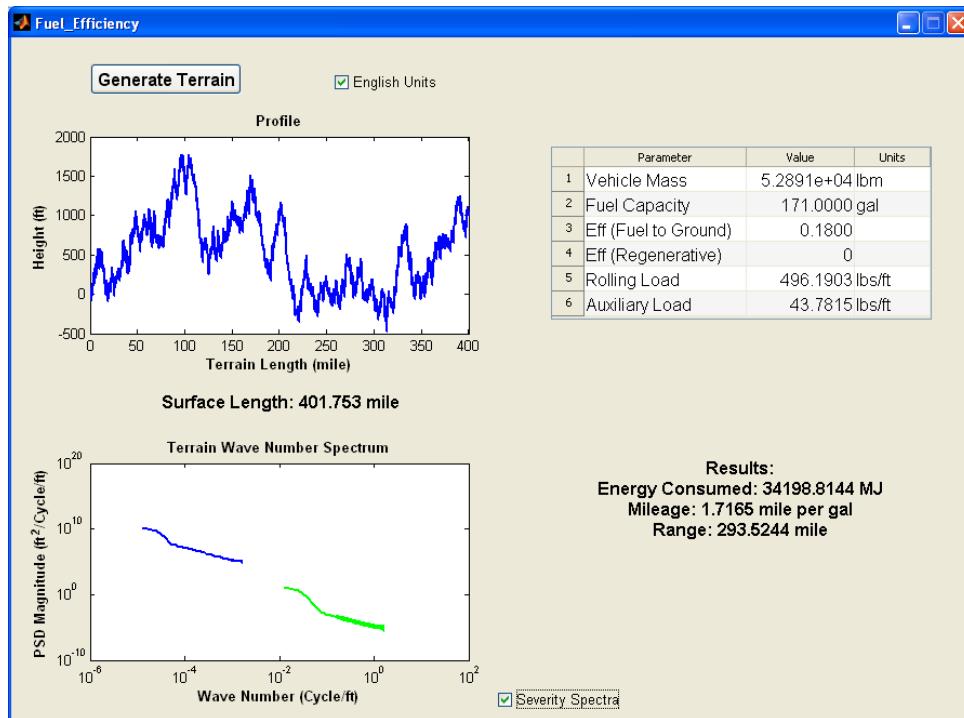


Figure 9

AMPHIBIOUS CONTEXT MODEL

In order to determine the reserve buoyancy, range and stability of an amphibious vehicle within a water condition an example vehicle was chosen. The example vehicle is a commercial amphibian (hydrotrek) whose weight and dimensional values are obtainable to the public online. This vehicle was set up within Pro/Engineer and created in such a way that the major dimensions were parameterized. Any change to one of the parameters will adjust the rest of the model through relation calculations. This allows for several size conditions to give the outputs needed for the buoyancy and stability calculations. The inputs needed are length, width, height, wheel diameter and

Appendix O: Standalone Executable Examples

either 2 or 4 occupants. The outputs produced are total volume, center of gravity, and vehicle mass. Again these outputs are based on relations to the initial model.

Once the outputs are obtained, either through the Pro/e model or direct inputs, visual basic code is utilized to determine the reserve buoyancy, range and stability of the model.

- For the buoyancy calculations fluid type, temperature and elevation can be taken into account or can be left as the baseline values. The temperature and elevation adjust the result slightly while the fluid type can have a larger effect if it is a mixture. Options are in place in order to allow for 5 different particulate materials mixed with fresh and/or salt water to create the fluid mixture. The correct fluid density is determined by the percentage of each.
- Along with fluid density several other outputs are calculated including object density, buoyant mass and buoyant force, but reserve buoyancy is most important. The reserve buoyancy is the percentage of the vehicle which remains above water therefore as long as the value is positive the vehicle will float. If the value is negative, than the vehicle will be completely submerged under the water.
- In order to calculate the range and maximum speed of the amphibious vehicle within the water several inputs need to be known. These include, fuel efficiency, fuel tank capacity, run time, power, drag coefficient and reference area. A table is provided for several different frontal geometry configurations to estimate the drag coefficient. Range is given in both miles and nautical miles and speed is given in miles per hour as well as knots.
- The stability of the model is determined by using the weight, center of gravity and the reserve buoyancy of the vehicle in self righting calculations. The calculations use the maximum width and height of the vehicle as a bounding box in order to determine the stability curve. The stability curve shows at any particular heel angle, or angle at which the vehicle is tilted in the water, the likelihood that the vehicle will right itself or will tip over. If the slope of the curve is positive at the angle of interest than the vehicle will right itself back to equilibrium. If the slope is negative the angle will continue to increase and the vehicle will tip over. The static attitude, or forward tilt angle, of the vehicle is also calculated. This is done using a bounding box for the length and height and depends on the height of the water and the location of the center of gravity.

Appendix P contains more details on buoyancy properties. The files describing the Pro-E model and calculations for buoyancy, reserve buoyancy, and stability can be found in following location in SVN:
\Models\Aquatic\properties\buoyancy.

SEA-STATE PROTOTYPE

Due to the effect of waves, an additional sea-state-influenced buoyancy example was prototyped

```
./Aquatic/features/sea_state/test/PotentialFlowSeaApp
```

Inputs:

- Sea State
 - Implies values for significant wave height, wave period and wave length
 - Limited to 0-3. Non-linear effects, which are not modeled, participate in higher sea states.
- Vessel heading, radians – can change during simulation, affects wave encounter frequency
 - Vessel speed, m/s – can change during simulation, affects wave encounter frequency
 - Position:

Appendix O: Standalone Executable Examples

- x (m), y (m), z (m)
- Time: from simulation, seconds

Outputs:

- Surface height, $\eta(x, y, t)$, m
- Pressure, $P(x, y, z, t)$, Pascals
- Water velocity components $u(x, y, z, t)$, $v(x, y, z, t)$, $w(x, y, z, t)$, m/s

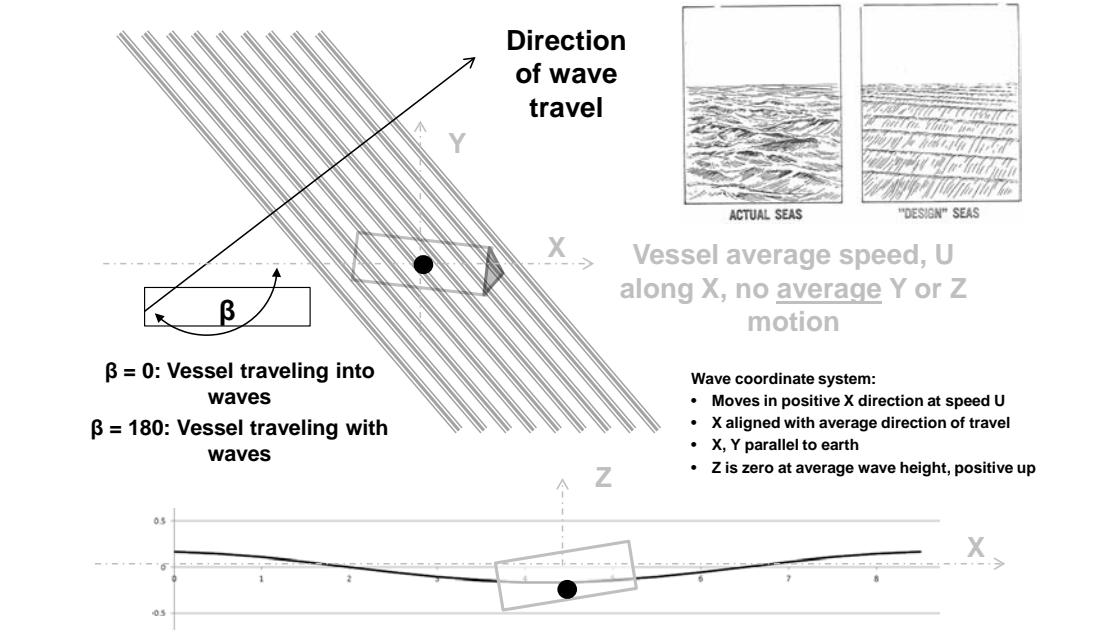


Figure 10: Wave coordinate system geometry

The introduction of waves to buoyancy requires dynamic modeling as shown below:

Appendix O: Standalone Executable Examples

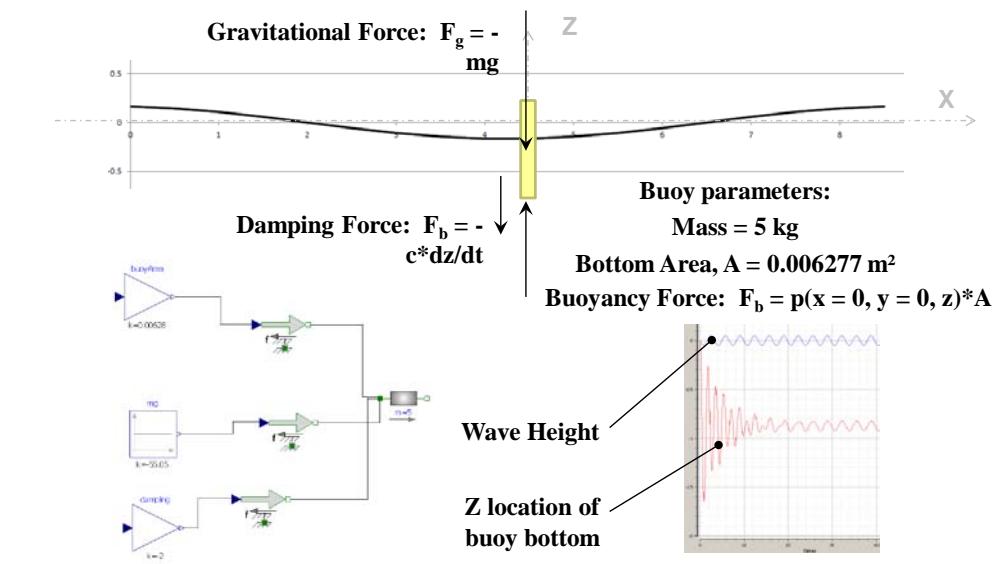


Figure 11: Example of dynamic output

SOIL TRACTION EXAMPLE

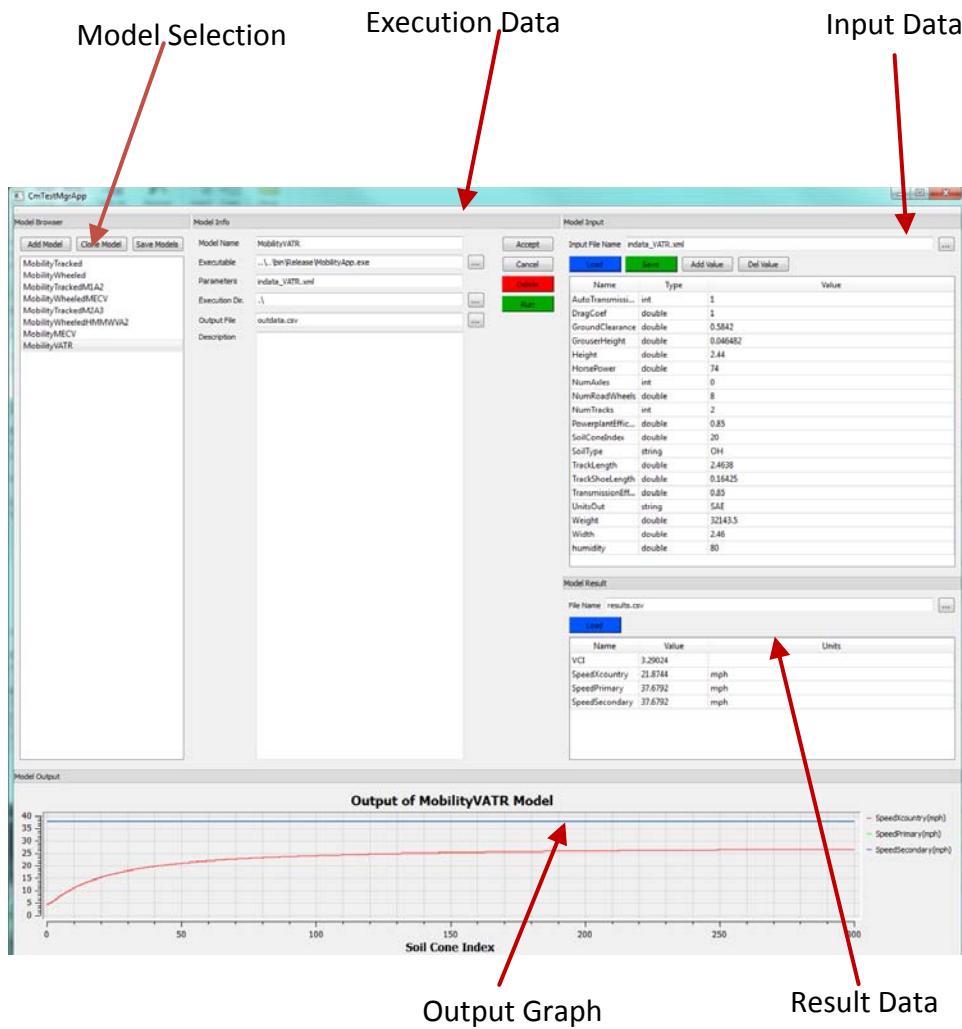
The Soil Cone Index Demo is a simple application to demonstrate how soil cone model can be used to determine the soil cone index with a given humidity value. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.

Appendix O: Standalone Executable Examples



The Mobility Demo is a simple application to demonstrate how mobility model can be used to determine the rolling resistance and the theoretical max speed of a vehicle. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.

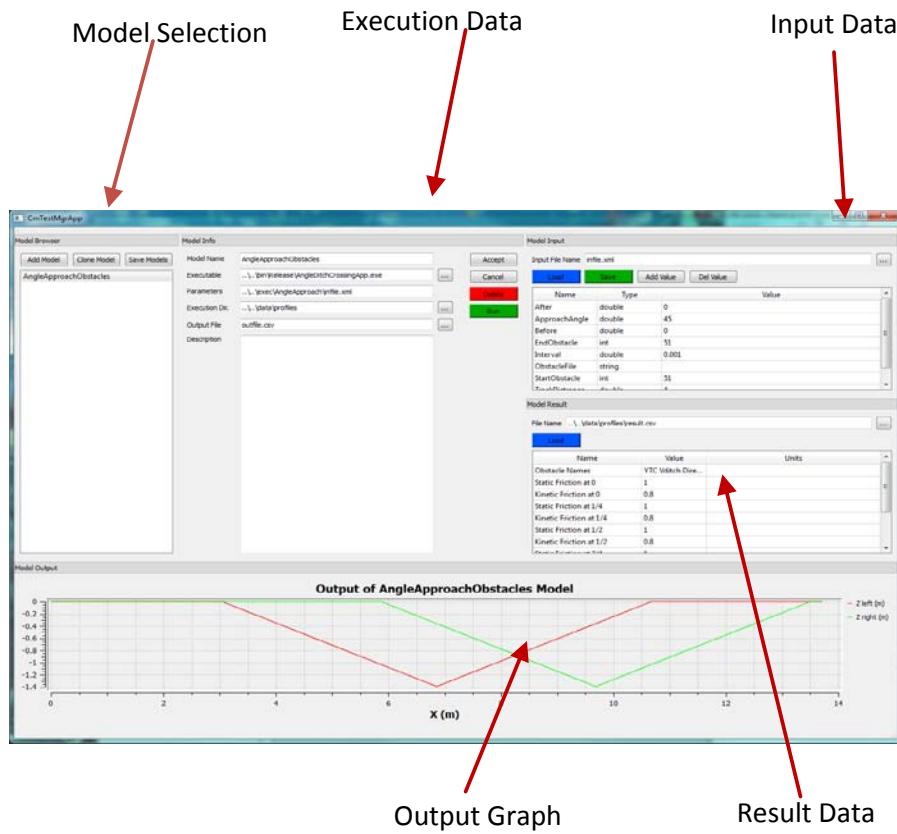
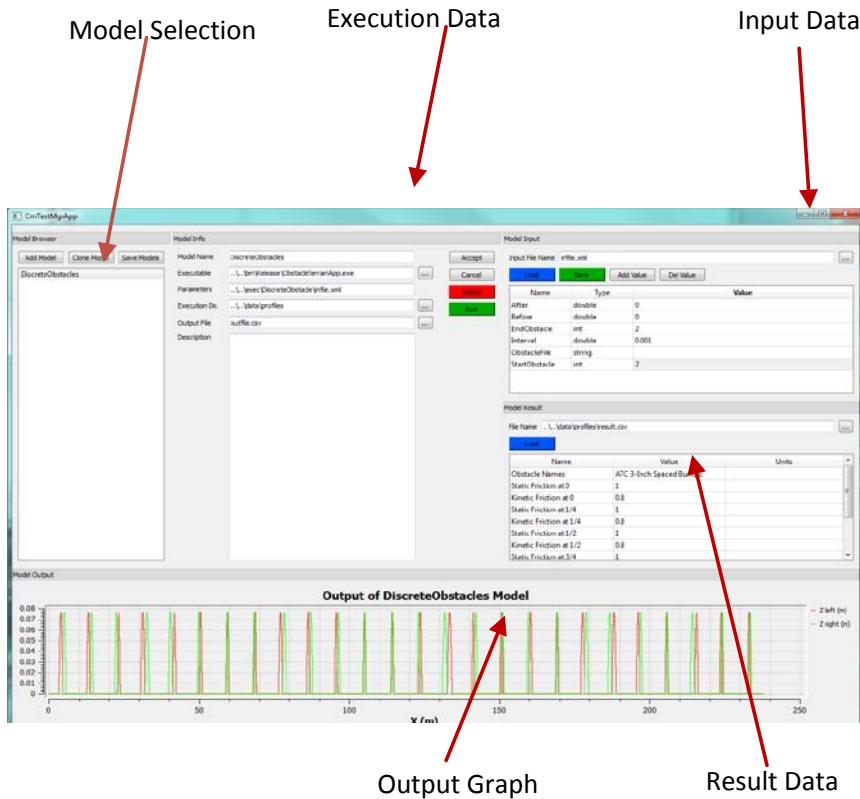
Appendix O: Standalone Executable Examples



GENERATING OBSTACLES AND TERRAINS EXAMPLES

The Discrete Obstacle Demo is a simple application to demonstrate how the discrete obstacle model can be used to create terrains. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.

Appendix O: Standalone Executable Examples



Appendix O: Standalone Executable Examples

The Terrain Generator Demo is a simple application to demonstrate how the spectral terrain model can be used to generate terrains.



Amphibious Vehicle Buoyancy Modeling

1. Summary

Several different attributes make up the amphibious context model but the main results are reserve buoyancy, range and stability. An excel spreadsheet was put together to contain all of these results based on inputs of a vehicle's basic dimensions and weight information. An example vehicle was chosen in order to prove out the calculations; this example was created using Pro/engineer and basic information from a public domain website, Hydrotek.

1.1 Technical Challenges

Assumptions were made for some of the calculations. First for any vehicle chosen, the results are in relation to the example vehicle unless specific values are known. Since exact volume and mass values are not always known this information can be set as a relation to the initial example vehicle that was used. Another assumption is that the area used for the stability curve calculation was a bounding box instead of the actual geometric shape of the vehicle. This allows for simple calculations that can be done quickly and higher powered tools such as CFD are not needed. As long as max width and height are known a quick check on the stability of the vehicle can be done.

1.2 General Methodology

In order to determine the reserve buoyancy, range and stability of an amphibious vehicle within a water condition an example vehicle was chosen. The example vehicle is a commercial amphibian (hydrotrek) whose weight and dimensional values are obtainable to the public online. This vehicle was set up within Pro/Engineer and created in such a way that the major dimensions were parameterized. Any change to one of the parameters will adjust the rest of the model through relation calculations. This allows for several size conditions to give the outputs needed for the buoyancy and stability calculations. The inputs needed are length, width, height, wheel diameter and either 2 or 4 occupants. The outputs produced are total volume, center of gravity, and vehicle mass. Again these outputs are based on relations to the initial model.

Once the outputs are obtained, either through the Pro/e model or direct inputs, visual basic code is utilized to determine the reserve buoyancy, range and stability of the model.

Table 1: Example Vehicle Inputs and Outputs

<u>Inputs:</u>	Max width	2451.1	mm
	Max length	3810	mm
	Max height	2082.8	mm
	wheel diameter	608	mm
	2 or 4 persons	4	
<u>Outputs:</u>	Volume	5.20E+09	mm ³
Center of Gravity (mm)	x	y	z
	0.00	2117.65	798.86

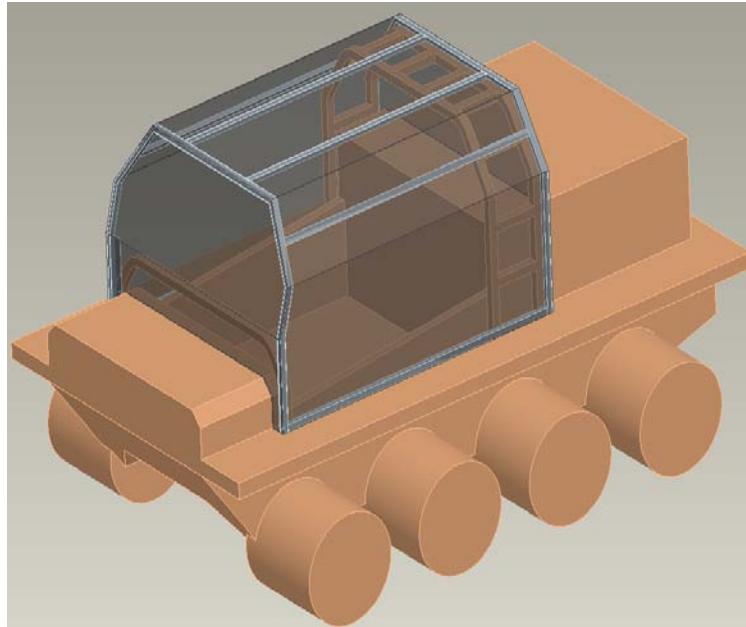


Figure 1: Pro/Engineer model of the Hydrotek vehicle

1.2.1 Reserve Buoyancy

For the buoyancy calculations fluid type, temperature and elevation can be adjusted or can be left as the baseline values. Along with reserve buoyancy several other outputs are calculated including fluid density, object density, buoyant mass and buoyant force. The reserve buoyancy is the percentage of the vehicle which remains above water therefore as long as the value is positive the vehicle will float. When the value is negative the vehicle will be completely submerged under the water.

Table 2: Baseline values for Reserve Buoyancy Calculations of the Example Vehicle

<u>Inputs:</u>	Object Mass	3179.4	kg	7009.3	lb
	Object Volume	5.2	m ³	183.6	ft ³
	Fluid	Fresh Water			
	Temperature	4	°C	39.2	°F
	Elevation	0	m	0	ft
<u>Outputs:</u>	Fluid Density	1000.0	kg/m ³	62.4	lb/ft ³
	Object Density	611.4	kg/m ³	38.2	lb/ft ³
	Buoyant Mass	-2021	kg	-4454.6	lb
	Buoyant Force	31190	N	7012.1	lbf
	Reserve Buoyancy	38.9	%		

The amount of particulates and debris within the fluid that the vehicle is entering can also affect the buoyancy results. If there is a large enough percentage of mud, sand, clay, gravel, etc. than the density of the mixture is changed and thus the buoyancy of whatever object rests on that mixture changes as well. Therefore coding was developed to allow for a mixture to be used and several given particulates are available to choose from. If a different substance is desired the density is required to complete the calculations.

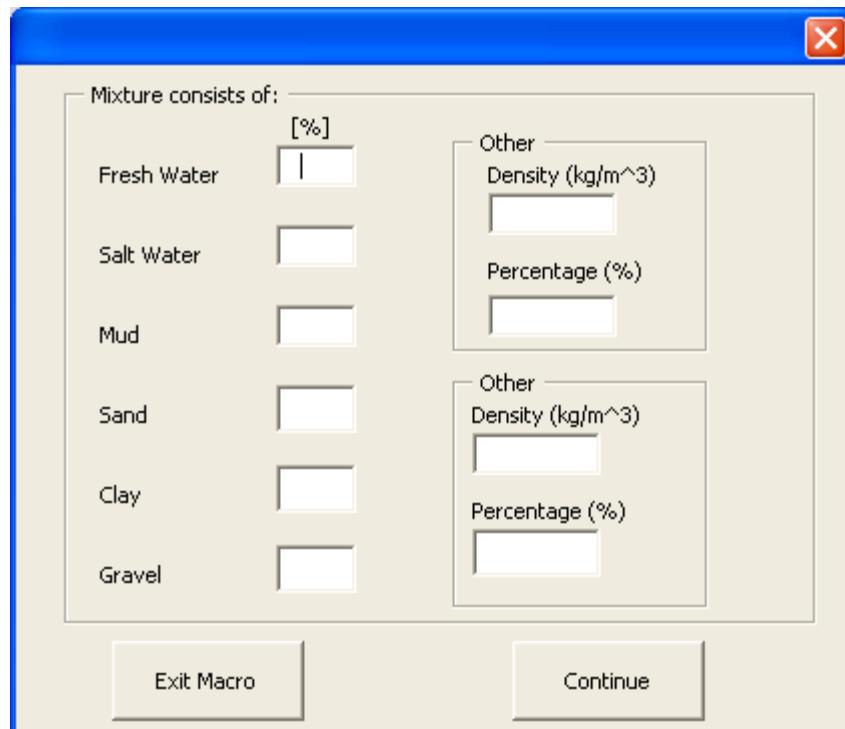


Figure 2: Mixture options

1.2.2 Range

In order to calculate the range and maximum speed of the amphibious vehicle within the water several inputs are needed. These include, fuel efficiency, fuel tank capacity, run time, power, drag coefficient and reference area. A table is provided for several different frontal geometry configurations to estimate the drag coefficient. Range is given in both miles and nautical miles and speed is given in miles per hour as well as knots.

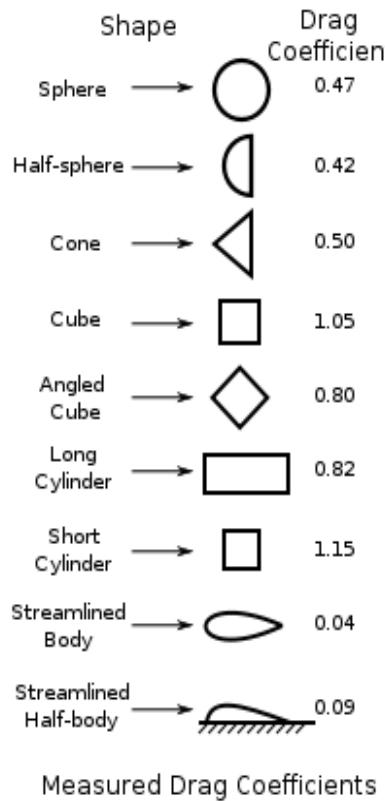


Figure 3: Drag Coefficients Values

Table 3: Baseline values for the Range Calculations of the Example Vehicle

<u>Inputs:</u>	Fuel Efficiency	MPG	
Fuel Tank Capacity	10.5	gal	
Run time	12.0	hr	
Power	74.0	HP	
Drag Coefficient	1.05		
Reference Area	0.56	m ²	
<u>Outputs:</u>	Range	miles	Naut miles
Max Speed	12.1	MPH	10.5 Knotts

1.2.3 Stability

The stability of the model is determined by using the weight, center of gravity and the reserve buoyancy of the vehicle in self righting calculations. The calculations use the maximum width and height of the vehicle as a bounding box in order to quickly maintain a result instead of using a more accurate depiction of the geometry.

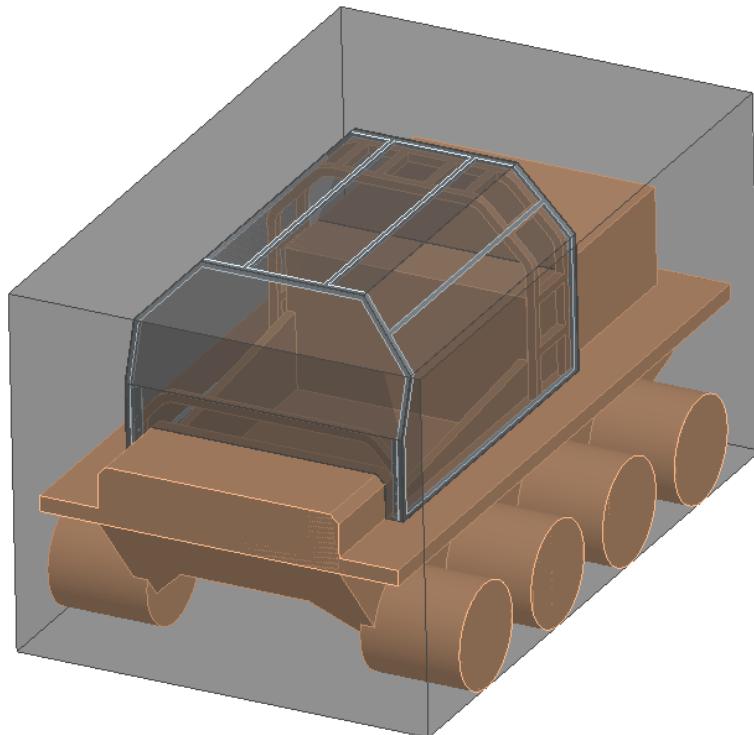


Figure 4: Bounding Box

The center of buoyancy is first determined at each tilt angle and then the perpendicular distance from the center of gravity to the center of buoyancy is calculated. The results are then graphed verse the tilt angle to form the stability curve. The stability curve shows at any particular heel angle, or angle at which the vehicle is tilted in the water, the likelihood that the vehicle will right itself or will tip over. If the slope of the curve is positive at the angle of interest than the vehicle will right itself back to equilibrium. If the slope is negative the angle will continue to increase and the vehicle will tip over. The higher the righting arm value, the more difficult, or the more force is required to tip as well.

Table 4: Righting Arm and Center of Buoyancy at each Heel Angle for the baseline vehicle

Stability Curve		Center of Buoyancy	
Heel Angle (deg)	Righting Arm (mm)	X (mm)	Y (mm)
0	0	1225.55	636.3
10	41.1	1156.181	642.4158
20	87.9	1082.359	662.3585
30	148.2	998.413	701.8687
40	222.5	907.1746	765.67
50	282.4	833.8081	838.8575
60	307.6	789.0677	902.0399
70	302.9	764.826	953.5452
80	279	752.5834	998.8378
90	242.5	748.8165	1041.4
100	198.6	752.5884	1083.962
110	152.9	764.8445	1129.253
120	112.5	789.1062	1180.753
130	89.1	833.8715	1243.923
140	89.2	907.1746	1317.13
150	94.3	998.4004	1380.963
160	78	1082.356	1420.456
170	43.1	1156.18	1440.388
180	0	1225.55	1446.5

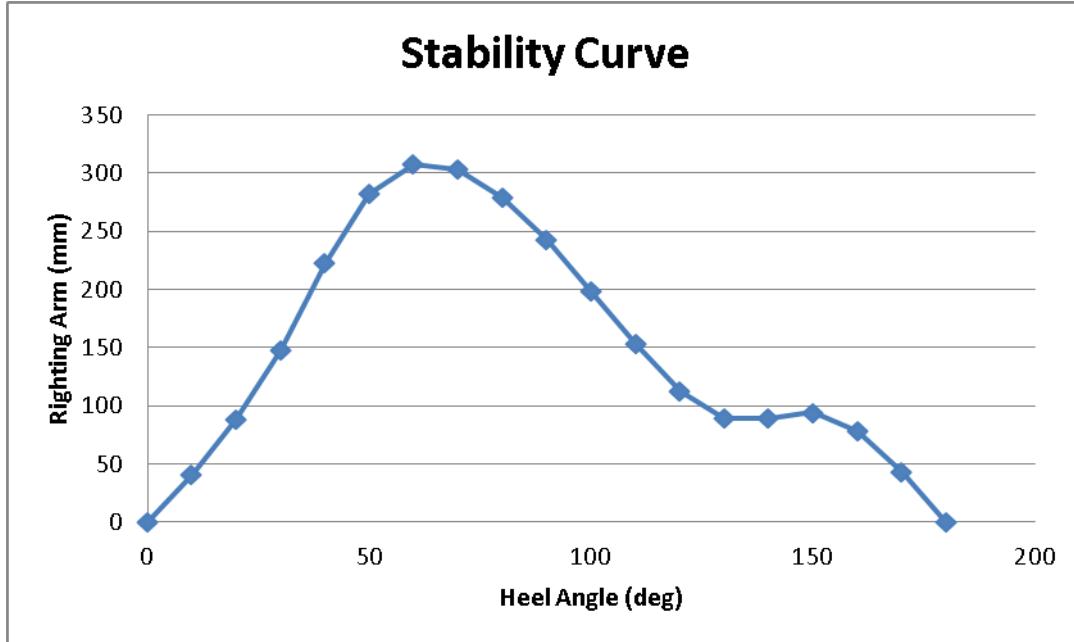


Figure 5: Stability Curve of the Baseline Vehicle

The static attitude, or pitch angle of the vehicle is also determined and is based on the length of the vehicle and the center of gravity location in relation to that length.

The static attitude, or tilt angle,
of the vehicle in the water is:
15 degrees.

Figure 6: Static Attitude of the Baseline Vehicle

1.3 Technical Results

In comparison with the baseline values several different scenarios are shown below for reserve buoyancy, range and stability to show the effects of changing one or more of the parameters used.

1.3.1 Reserve Buoyancy

As can be seen in table 5 through 8, changing the temperature and elevation do not affect the reserve buoyancy by very much but the changes in fluid can. A slight increase in the reserve buoyancy is seen when using salt water as opposed to fresh water. But an even higher increase results when the fluid is a mixture. The mixture used in the results of table 8 was 75% fresh water and 25% mud. The increase in density from the addition of the mud raised the reserve buoyancy by almost 10%. This means that a vehicle in only fresh water will sit lower

Appendix P: Amphibious Vehicle Buoyancy Modeling

in the water than if the water has a mixture of mud in it as well. The positioning of the vehicle in the water affects the stability which is shown further in the report.

Table 5: Change in Fluid Type Effects on Baseline results

<u>Inputs:</u>	Object Mass	3179.4	kg	7009.3	lb
	Object Volume	5.2	m ³	183.6	ft ³
	Fluid	Salt Water			
	Temperature	4	°C	39.2	°F
	Elevation	0	m	0	ft
<u>Outputs:</u>	Fluid Density	1030.0	kg/m ³	64.3	lb/ft ³
	Object Density	611.4	kg/m ³	38.2	lb/ft ³
	Buoyant Mass	-2177	kg	-4798.5	lb
	Buoyant Force	31190	N	7012.1	lbf
	Reserve Buoyancy	40.6	%		

Table 6: Change in Temperature Effects on Baseline results

<u>Inputs:</u>	Object Mass	3179.4	kg	7009.3	lb
	Object Volume	5.2	m ³	183.6	ft ³
	Fluid	Fresh Water			
	Temperature	20	°C	68	°F
	Elevation	0	m	0	ft
<u>Outputs:</u>	Fluid Density	996.8	kg/m ³	62.2	lb/ft ³
	Object Density	611.4	kg/m ³	38.2	lb/ft ³
	Buoyant Mass	-2004	kg	-4418.0	lb
	Buoyant Force	31190	N	7012.1	lbf
	Reserve Buoyancy	38.7	%		

Table 7: Change in Elevation Effects on Baseline results

<u>Inputs:</u>	Object Mass	3179.4	kg	7009.3	lb
	Object Volume	5.2	m ³	183.6	ft ³
	Fluid	Fresh Water			
	Temperature	4	°C	39.2	°F
	Elevation	10000	m	32800	ft
<hr/>					
<u>Outputs:</u>	Fluid Density	1000.0	kg/m ³	62.4	lb/ft ³
	Object Density	611.4	kg/m ³	38.2	lb/ft ³
	Buoyant Mass	-2021	kg	-4454.6	lb
	Buoyant Force	31190	N	7012.1	lbf
	Reserve Buoyancy	38.9	%		

Table 8: Effects of a fluid mixture on Baseline results

<u>Inputs:</u>	Object Mass	3179.4	kg	7009.3	lb
	Object Volume	5.2	m ³	183.6	ft ³
	Fluid	Mixture			
	Temperature	4	°C	39.2	°F
	Elevation	0	m	0	ft
<hr/>					
<u>Outputs:</u>	Fluid Density	1182.5	kg/m ³	73.8	lb/ft ³
	Object Density	611.4	kg/m ³	38.2	lb/ft ³
	Buoyant Mass	-2970	kg	-6546.8	lb
	Buoyant Force	31190	N	7012.1	lbf
	Reserve Buoyancy	48.3	%		

1.3.2 Range

There are several different inputs that can affect the range and speed outputs of a vehicle but in this case the effects of the drag coefficient were focused on. As the drag decreases, the number of miles the vehicle can cross and the speed at which it can do so both increase.

Table 9: Change in Drag Coefficient Effects on Baseline Results

Drag Coefficient	Range (naut miles)	Speed (knotts)
1.05	133.6	11.1
0.9	140.6	11.7
0.75	149.4	12.5
0.6	160.9	13.4
0.45	177.1	14.8
0.3	202.8	16.9
0.15	255.5	21.3

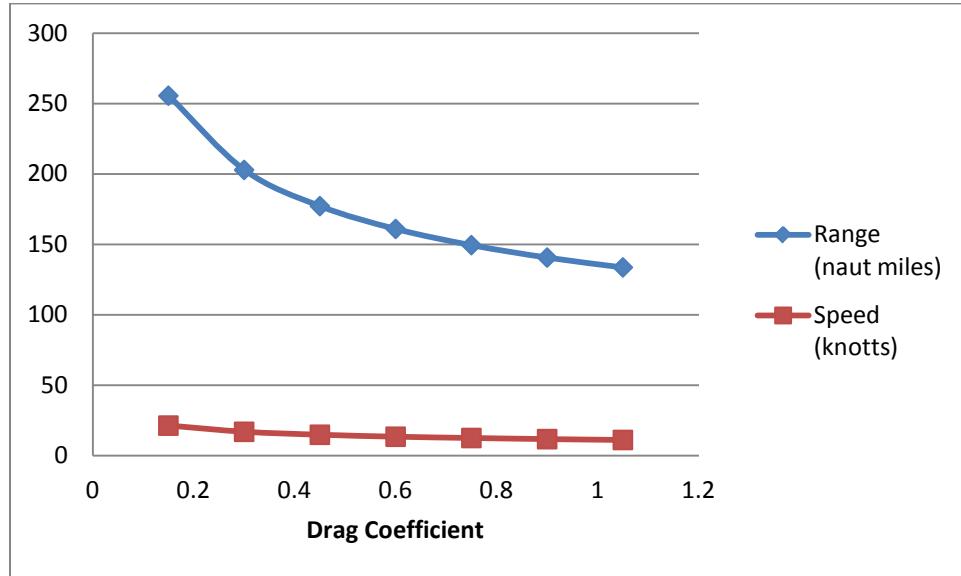


Figure 7: Effects of Drag Coefficient on Range and Speed

1.3.3 Stability

There are many different scenarios for the stability curve. Several of these are shown in the following figures. The baseline values of the bounding box are that the height is equal to 1082.8mm and the width is equal to 2451.1mm. The reserve buoyancy is 38.9% and the center of gravity height is equal to 798.86mm. Using these baseline values and changing only one per scenario gives very different curves.

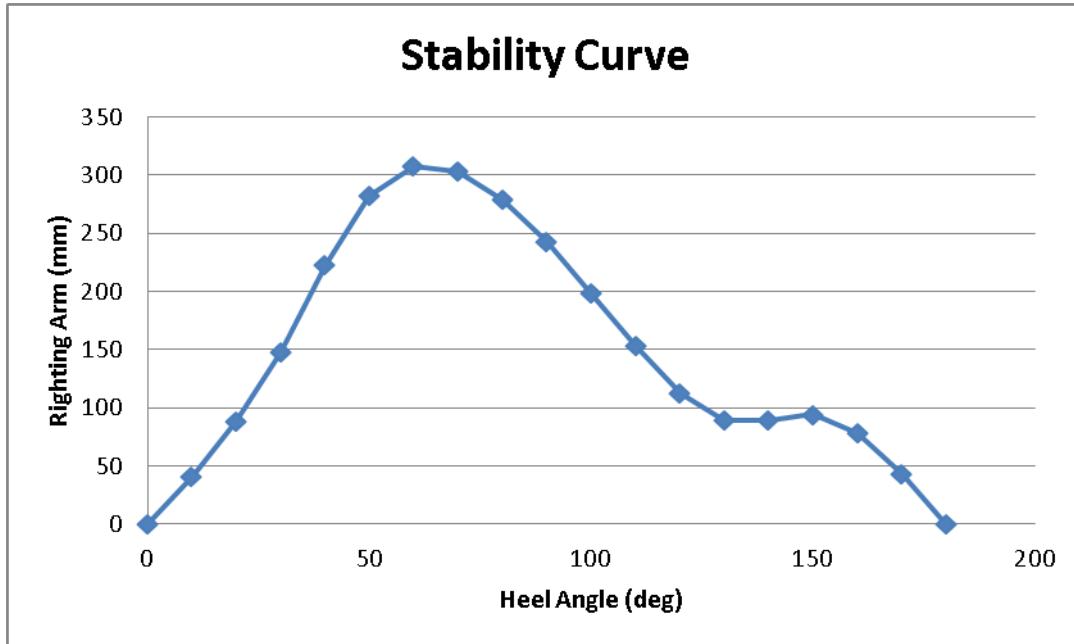


Figure 8: Stability Curve of the Baseline Vehicle

From reviewing the baseline curve it can be seen that the stability point occurs at a 60 degree tilt angle but there is another stability point at 150 degrees as well. Also noted is that the righting arm is just above 300 mm at the first stability point. When changing the center of gravity location to be half the height, or centered on the bounding box shape the curve becomes very different. The initial stability point is still at approximately 60 degrees but the righting arm at this point is only 100mm. Also the curve drops below zero and a negative stability point occurs at 120 degrees and -100mm.

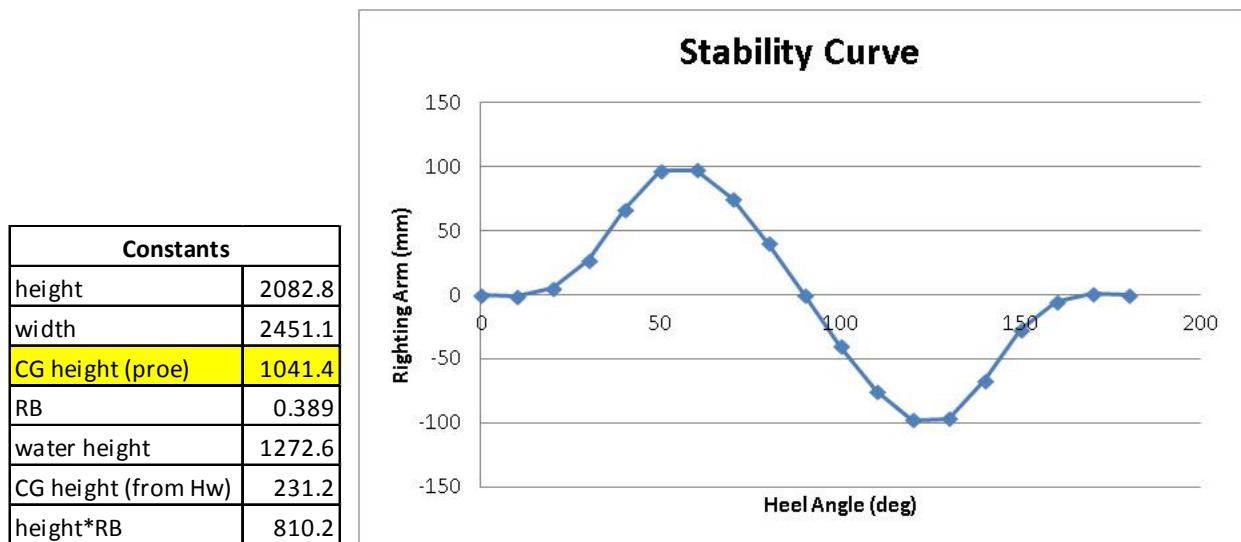


Figure 9: CG Height changed to be half the bounding box height

Again changing the center of gravity location changes the curve drastically. When the CG is above the waterline the vehicle immediately wants to tip and does not begin to right itself until approximately 110 degrees. It is also requires much more force to right itself noted by the righting arm being over -400mm.

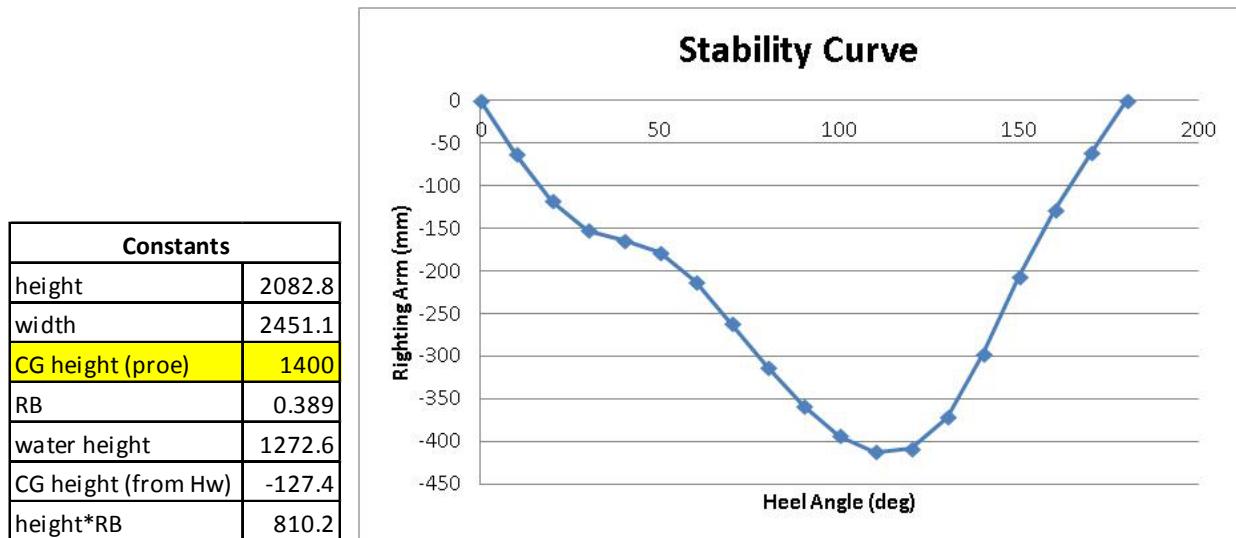


Figure 10: CG Height changed to be above the water line

With the center of gravity back to the baseline value the width and the height of the bounding box are adjusted. This does not have nearly the same effect on the curve as changing the CG did. Although the righting arm values do change the stability points are still very similar to the baseline curve.

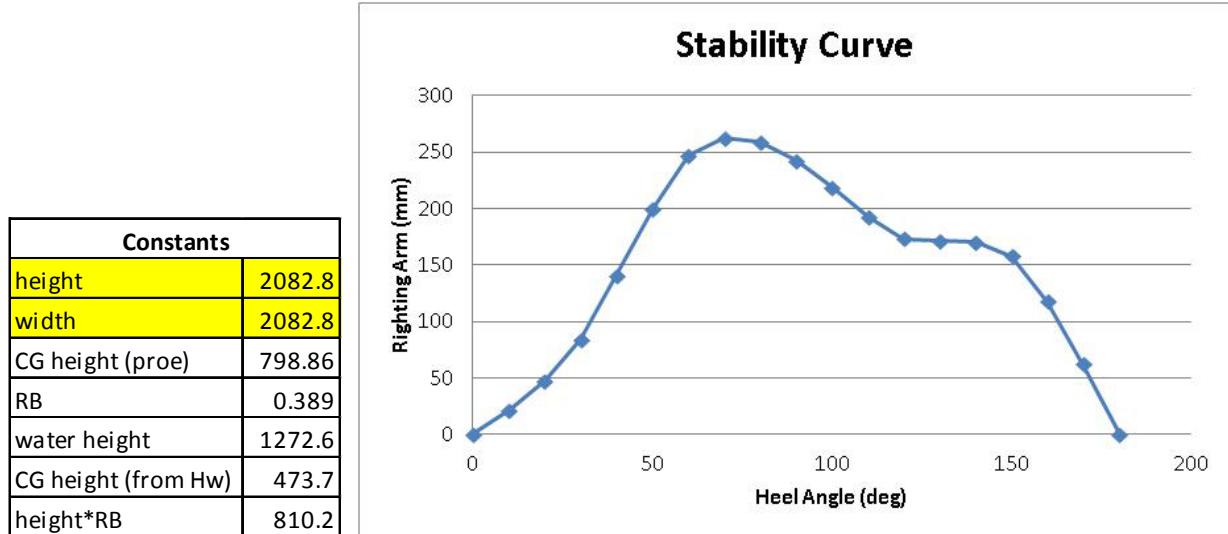


Figure 11: Height and Width are equal

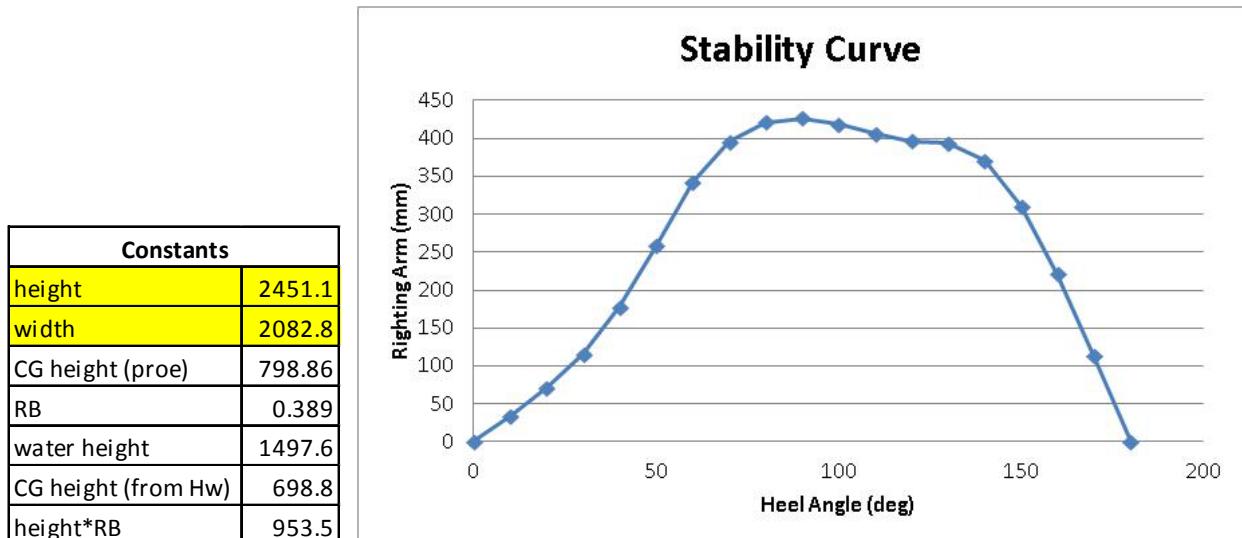


Figure 12: Height is greater than Width

The final two scenarios shown have changes to the reserve buoyancy which results in a different waterline location. Of the different scenarios tested this changes the baseline curve the least. The shape of the curve remains the same with only slight changes to the righting arm values at each stability point along the curve.

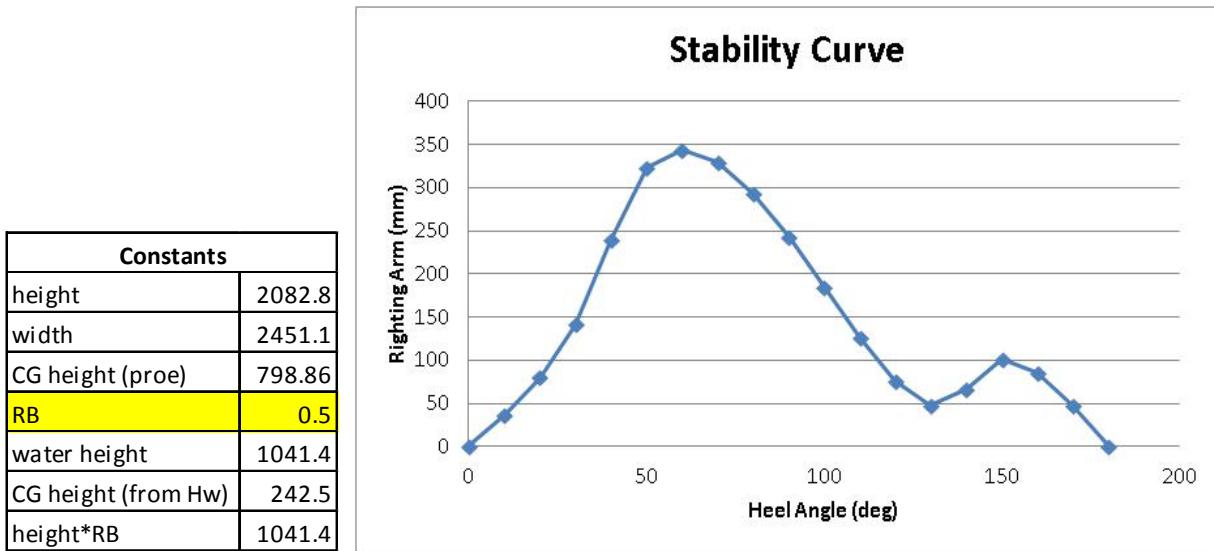


Figure 13: Reserve Buoyancy is equal to 50%

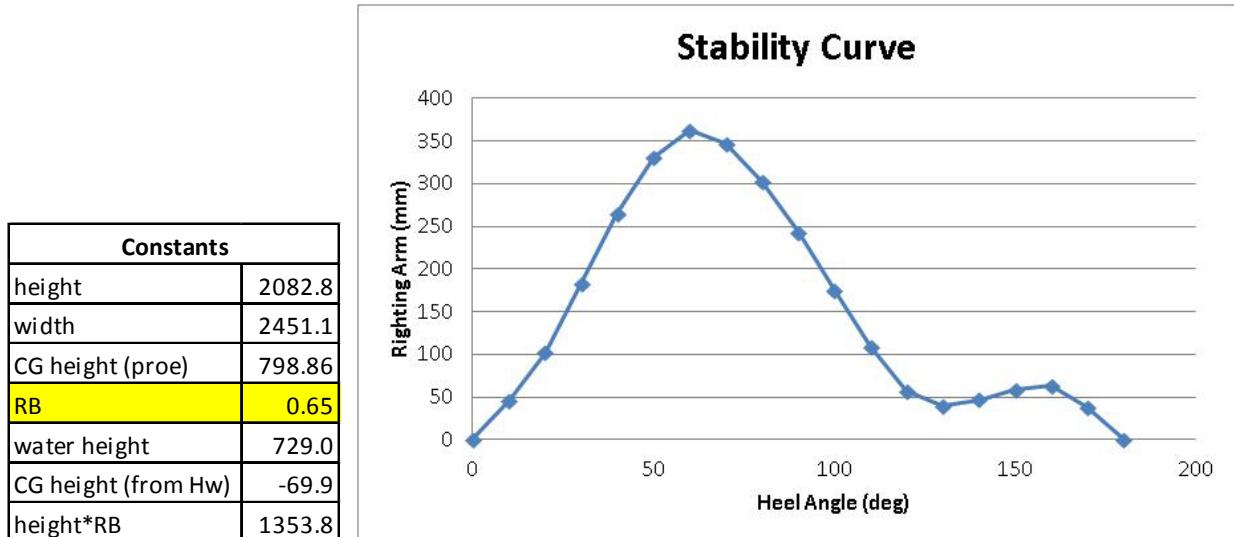


Figure 14: Reserve Buoyancy is equal to 65%

1.4 Important Findings and Conclusions

In conclusion, it is proven through this amphibious context model that there are certain parameters that effect results more than others, and if these parameters can be focused on and adjusted appropriately desired requirements can be met. For the reserve buoyancy, the density of the fluid in which the vehicle must drive through has the greatest impact, since it would be difficult to change this parameter it is best to try and adjust the density of the vehicle itself. For the range and speed, the drag coefficient has a great impact on the results and this is something easily manageable with changes in the vehicle design. Lastly the stability of the vehicle is greatly affected by the center of gravity location, therefore if it is possible to arrange the masses of the vehicle to allow for a lower CG that is best for a desirable outcome.

1.5 Implications for Further Research

Further work may be important to include more accurate bounding shapes for the stability curve results. The more closely related the bounding geometry is to the actual, the more accurate the curve will be. Therefore if more precise results are necessary incorporating a bounding polygon or other type shape will be needed. These shapes require much larger calculations and coding so they were not initially included.

Context Model FMI Integration

Standalone simulations that employ context model libraries are commonly written in languages that require specific interface protocols. The rationale and motivation for enabling a reliable co-simulation environment via contractual interfaces is described elsewhere [1][2]. As a potential solution proffered, the interoperability among simulations is proscribed by the MODELISAR project [3] (in support of the AUTOSAR vehicle design initiative) [4]. This appendix describes the process of prototyping context model interfaces to be exercised in a larger vehicle simulation environment.

1. Summary

There was a desire to make the context models available for use within the Modelica or Dymola environment. One of the ways this could be accomplished is to use the Functional Mockup Interface (FMI) that is a model import feature of Modelica/Dymola and can be purchased as a 3rd party toolkit in MATLAB. In order to demonstrate how to import models using FMI we created several Functional Mockup Units (FMUs) for several of the context models as prototypes on how the context models could be used to create components that could be imported into Modelica or Dymola.

1.1 Technical Challenges

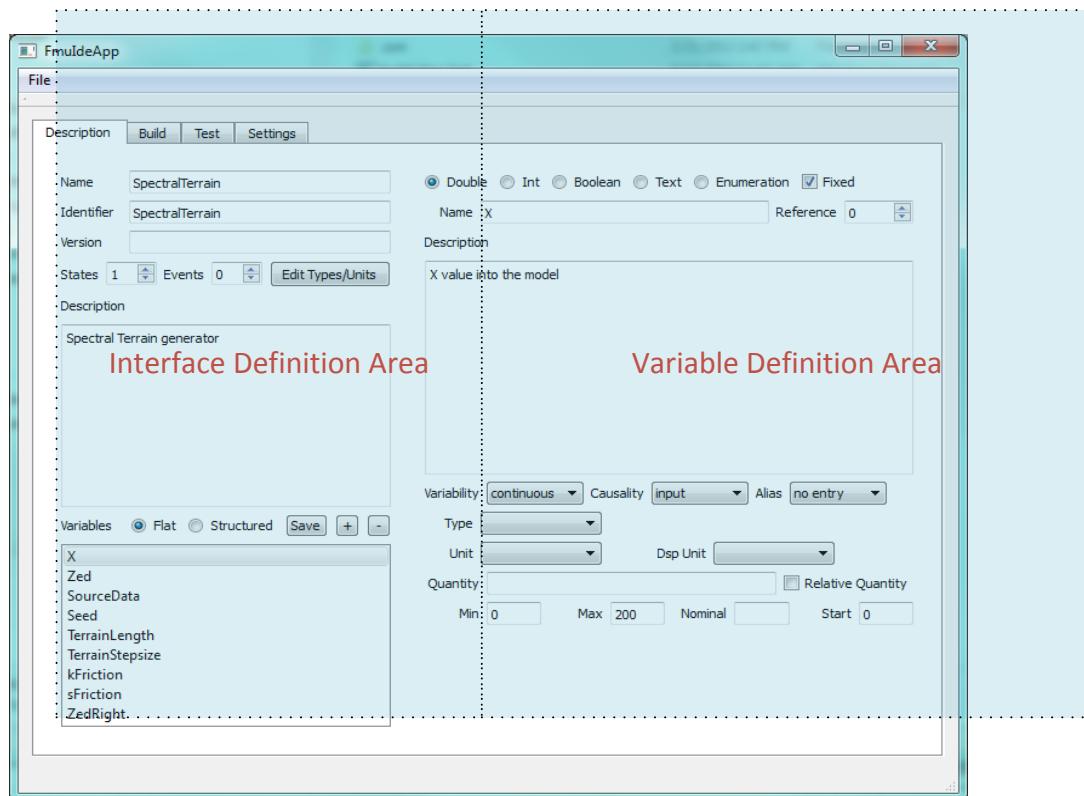
We looked into what tools were available for creating FMUs. What we found was a lack of open source tools for creating FMUs and the ones we found were limited in their capabilities and ease of use. Although the FMI specification was found to be sufficient for the designed purpose of integration with multiple tools we found that OpenModelica, Dymola and MATLABs implementation to be inconsistent. We found numerous bugs and non-standard implementations during our development indicative of an immature specification that isn't widely used.

1.2 General Methodology

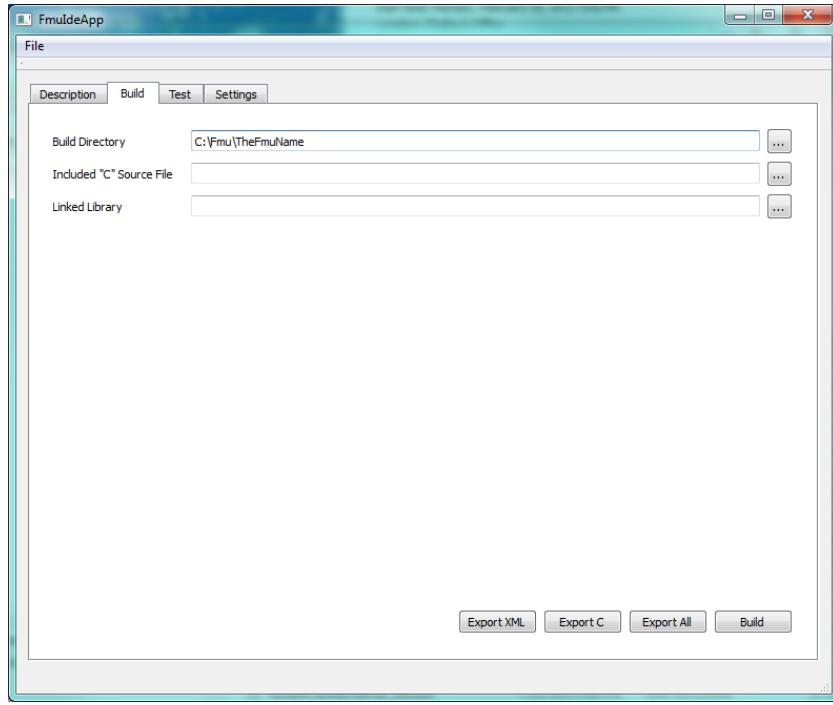
After a search of the available tools we settled on the FMU SDK as an open source toolkit that would allow us to package our C++, Visual Studio based models into FMUs that could be imported using the FMI import features of the META development tools. Creating the needed files to create an FMU using the FMU SDK was found to be a manual process that is prone to error. In order to reduce the chances of error and to ensure that the resulting interface was compliant with the FMI specification we decided that a FMU development tool was needed. This tool could be used to define the FMU using an easy to use GUI interface that would only aid the user in developing the interface and then export the files needed for implementing the interface using the FMU SDK.

1.3 Technical Results

We created a tool called the FMU IDE (Integrated Development Environment) to aid in FMU creation. The tool (see below) has an interface definition tab that has an interface definition area that takes up the left half of the tab and on the right half is the variable definition area that shows the information for the currently selected variable.



When the user is satisfied that the interface is defined they can use the build tab to export the XML, the C code or both. The tool will export to the directory define in the Build Directory text box. The user can use the “...” button to browse to the desired directory. As an option the user can define a C language source file to be included in the exported file via a “#include” statement. This included source file could have additional links to source files or functions that are used to define the behavior of the interface.



Prior to building the FMU the developer must unzip the fmusdk.zip file that is in the ...\\Models\\Tools directory into that directory.

In order to build the FMU the developer must do the following steps:

1. In the directory directly below the directory containing the FMU's C and XML file make a copy of the build_all.bat file that is in ...\\Models\\Tools\\exec
2. Modify the file for the specific FMU
3. Run the modified build_all.bat from the command line or by double clicking from the Windows Explorer.
4. It will create an FMU in the directory ..\\fmu\\me for model exports FMUs or ..\\fmu\\cs for co-simulation FMUs.

1.4 Important Findings and Conclusions

As stated earlier the acceptance of the FMI specification is in its infancy and in need of a greater breadth of tools that use FMUs for importing models. To that end we created a tool that could be available as an open source addition to the FMU SDK. This tool would enable standardized models to be available via FMUs. Although this tool and the resulting context model FMUs were developed using Visual Studio 2010 on Windows 7 it should be fairly easy to port to other environments because the tool uses QT and doesn't use Windows only extensions to standard C++.

We created six FMUs and examples that demonstrate their use. The six are:

1. TerrainService.fmu: This model is similar to the SpectralTerrain.fmu and the DiscreteObstacle.fmu but this uses the terrain server. An example that uses this FMU

- is the TerrainCrawler2server.mo that is located in the directory
`...\\Models\\Land\\terrains\\grades_slopes\\interface\\Modelica.`
2. TempFMU.fmu: This model returns the diurnal temp. It uses the time from the simulation environment to determine the temperature.
 3. SpectralTerrain.fmu: This FMU returns two Z values for right and left (ZedRight and Zed) values (m) for every x value input into the model. The initialization values a seed for the random number generator, the desired terrain length (m), the step size for the terrain (m), needed are an enumerated value that is defined in the file Terrain PSD metadata.xlsx. The file resides in the directory
`...\\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra.` It also returns the kinetic and static friction for the course. Examples that use this FMU are TerrainCrawler.mo and FrictionTerrainCrawler.mo that reside in the directory
`...\\Models\\Land\\terrains\\grades_slopes\\interface\\Modelica.`
 4. SeaState.fmu: This FMU provides a pressure value (in Pa), the height of the water surface, and the u,v,w of the water velocity values for x,y,z (in m/s). The input is the heading of the vessel (in radians), the speed of the vessel (in m/s) the ocean depth (in meters) and the u,v,w of the water velocity value. An example that uses the SeState.fmu is buoy.mo in the directory
`...\\Models\\Aquatic\\features\\sea_state\\interface\\Modelica.`
 5. DiscreteObstacle.fmu: Provide two Z values (in meters), for each X input value (in meters), for left and right tracks for a defined obstacle course. The main initialization input variable is an enumerated value for the obstacle that is documented in
`...\\Models\\Land\\obstacles\\data\\Obstacle metadata.xlsx.` The state of the model is determined by varying X and the model returns a Zr(right) and Zl(left). An example that uses this FMU is the ObstacleTerrainCrawler.mo in the directory
`...\\Models\\Land\\obstacles\\interface\\Modelica.`
 6. ClimateServe.fmu – Provides the current temperature. The input is an enumerated value for the climate design type:
 - 0) Hot dry: QSTAG 360 = A1 (north Africa, Middle East, Pakistan, India, southwest US, northern Mexico)
 - 1) Hot humid: QSTAG 360 = B3 (Persian Gulf, Red Sea)
 - 2) Constant high humidity: QSTAG 360 = B1
 - 3) Variable high humidity: QSTAG 360 = B2
 - 4) Basic hot, QSTAG 360 = A2 (US, Mexico, Africa, Asia, Australia, South America, southern Spain, SW Asia)
 - 5) Basic cold, QSTAG 360 = C1 (high latitude coasts, eg, southern Alaska)
 - 6) Cold, QSTAG 360 = C2 (Canada, Alaska, Greenland, northern Scandinavia, northern Asia, Tibet, Alps, Himalayas, Andes)
 - 7) Severe cold, QSTAG 360 = C3 (interior Alaska, Canadian Yukon, Greenland icecap, north Asia)

Also the other input is the enumerated value for the condition type as follows:

- 0) Operational: conditions in open to which materiel may be subjected to during operations or standby
 - 1) Storage transit: Conditions materiel may be subjected to in storage or transit
- The examples for this FMU are the ColdStart.mo and the WeatherClient.mo

1.5 Implications for Further Research

Expanding the FMU IDE to include build and test capabilities would make the FMU creation easier. The current hand building of the build batch file could easily be done under the covers of the FMU IDE. There is also plenty of room to mature the implementation of the the FMI specification in OpenModelica or teaming with Dassault Systemes to mature the implementation in Dymola. Also an open source implementation for MATLAB and other tools is an area to needs work.

1.6 References

- [1] H. Heinecke, W. Damm, B. Josko, A. Metzner, H. Kopetz, A. Sangiovanni-Vincentelli, and M. Di Natale, “Software components for reliable automotive systems,” presented at the Proceedings of the conference on Design, automation and test in Europe, 2008, pp. 549–554.
- [2] S. Bankes, D. Challou, T. Haynes, H. Holloway, P. Pukite, J. Tierno, and C. Wentland, “META Adaptive, Reflective, Robust Workflow (ARRoW),” BAE Systems, Final Report TR-2742, 2011.
- [3] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauß, H. Elmquist, M. Friedrich, A. Junghanns, J. Mauss, and D. Neumerkel, “Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models,” presented at the 9th International Modelica Conference, Munich, 2012.
- [4] “MODELISAR - Wikipedia, the free encyclopedia.” [Online]. Available: <http://en.wikipedia.org/wiki/MODELISAR>. [Accessed: 26-Feb-2013].

Appendix Z : Context Model Map

	Repository View (via the Subversion version control system)			Ontological Views (via a browser)	
File Location					
				Dynamic Context Server	OSCAR
Context Descriptions	Model Location		Documentation	Examples	URL (base http://localhost:3020)
Atmospheric Environment					
Air Properties					
Pressure	\Models\Composite\FluidThermal\src\stdAtmos1976.cpp \Models\Composite\FluidThermal\src\stdAtmos1976.h \Models\Composite\FluidThermal\src\stdAtmos1962.cpp \Models\Composite\FluidThermal\src\stdAtmos1962.h \Models\Composite\FluidThermal\src\idealGas0.cpp \Models\Composite\FluidThermal\src\idealGas0.h \Models\Composite\FluidThermal\src\hvaco.cpp \Models\Composite\FluidThermal\src\AR7038environ.h \Models\Composite\FluidThermal\src\AR7038environ.cpp \Models\Composite\FluidThermal\src\hvaco.h		\Models\Composite\FluidThermal\doc\namespace_AR7038Environment.doc \Models\Composite\FluidThermal\doc\namespace_HVAC.doc	\Models\Composite\FluidThermal\test\TempDrawdown	/context_atm/navigate
Density	\Models\Composite\FluidThermal\src\stdAtmos1976.h \Models\Composite\FluidThermal\src\stdAtmos1962.cpp \Models\Composite\FluidThermal\src\stdAtmos1962.h \Models\Composite\FluidThermal\src\idealGas0.cpp \Models\Composite\FluidThermal\src\idealGas0.h \Models\Composite\FluidThermal\src\hvaco.cpp \Models\Composite\FluidThermal\src\hvaco.h		\Models\Composite\FluidThermal\doc\namespace_HVAC.doc	\Models\Composite\FluidThermal\test\TempDrawdown	/context_atm/navigate
Moisture	\Models\Composite\FluidThermal\src\climateAR7038.cpp \Models\Composite\FluidThermal\src\climateAR7038.h \Models\Composite\FluidThermal\src\moistAirMix.cpp \Models\Composite\FluidThermal\src\moistAirMix.h \Models\Composite\FluidThermal\src\saturatedWater.cpp \Models\Composite\FluidThermal\src\saturatedWater.h		\Models\Composite\FluidThermal\doc\classes_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\classes_moistAirMixture.doc \Models\Composite\FluidThermal\doc\namespace_HVAC.doc	\Models\Composite\FluidThermal\test\TempDrawdown	/context_climate_AR7038/navigate

Appendix Z : Context Model Map

Temperature (Arctic, Cold, Normal, Hot)	\Models\Composite\FluidThermal\src\stdAtmos1976.cpp \Models\Composite\FluidThermal\src\stdAtmos1976.h \Models\Composite\FluidThermal\src\stdAtmos1962.cpp \Models\Composite\FluidThermal\src\stdAtmos1962.h \Models\Composite\FluidThermal\src\idealGas0.cpp \Models\Composite\FluidThermal\src\idealGas0.h \Models\Composite\FluidThermal\src\hvacio.cpp \Models\Composite\FluidThermal\src\hvacio.h	\Models\Composite\FluidThermal\doc\classes_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\namespace_HVAC.doc	\Models\Composite\FluidThermal\test\TempDrawdown \Models\Composite\FluidThermal\test\ColdStart	/context_climate_AR7038/navigate	
Temperature (Locally Induced)	\Models\Composite\FluidThermal\src\ThermalComp.cpp \Models\Composite\FluidThermal\src\ThermalComp.h \Models\Composite\FluidThermal\src\ThermalLink.cpp \Models\Composite\FluidThermal\src\ThermalLink.h \Models\Atmospheric\properties\temperature\src\TempDiurnal.cpp \Models\Composite\FluidThermal\src\ThermalLink.h \Models\Composite\FluidThermal\src\ThermalNet.cpp \Models\Composite\FluidThermal\src\ThermalNet.h \Models\Composite\FluidThermal\src\diurnalCycle.h \Models\Composite\FluidThermal\src\diurnalCycle.cpp	\Models\Composite\FluidThermal\doc\classes_ThermalNet.doc \Models\Composite\FluidThermal\doc\namespace_HVAC.doc	\Models\Composite\FluidThermal\test\ColdStart \Models\Atmospheric\properties\temperature\test\Thermometer	/context_select/navigate?category=thermal	
Atmospheric Features					
Wind	\Models\Composite\FluidThermal\src\AR7038environ.cpp \Models\Composite\FluidThermal\src\AR7038environ.h \Models\Composite\FluidThermal\src\climateAR7038.cpp \Models\Composite\FluidThermal\src\climateAR7038.h	\Models\Composite\FluidThermal\doc\classes_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\namespace_Climate_AR7038.doc	\Models\Composite\FluidThermal\test\TempDrawdown	/context_select/navigate?category=wind	
Solar Radiation	\Models\Composite\FluidThermal\src\AR7038environ.cpp \Models\Composite\FluidThermal\src\AR7038environ.h \Models\Composite\FluidThermal\src\climateAR7038.cpp \Models\Composite\FluidThermal\src\climateAR7038.h \Models\Composite\FluidThermal\src\solar_ASHRAE.cpp \Models\Composite\FluidThermal\src\solar_ASHRAE.h \Models\Composite\FluidThermal\src\solar_ASHRAE2009.cpp \Models\Composite\FluidThermal\src\solar_ASHRAE2009.h	\Models\Composite\FluidThermal\doc\classes_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\namespace_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\classes_solarLoad_ASHRAE.doc \Models\Composite\FluidThermal\doc\classes_solarLoad_ASHRAE2009.doc	\Models\Composite\FluidThermal\test\TempDrawdown	/context_solar/navigate /context_climate_AR7038/navigate	
Contaminants					
Corrosive Components (Salt spray, SO ₂ , NO _x)				/context_select/navigate?category=corrosion	
Particulates (Dust, Sand, Volcanic Ash, Rain, Snow, Ice Crystals)	\Models\Composite\FluidThermal\src\AR7038environ.cpp \Models\Composite\FluidThermal\src\AR7038environ.h \Models\Composite\FluidThermal\doc\Example Filter Performance with Particle Size.xlsx	\Models\Composite\FluidThermal\doc\classes_Climate_AR7038.doc \Models\Composite\FluidThermal\doc\namespace_Climate_AR7038.doc	\Models\Composite\FluidThermal\test\IntakeFilter	/context_select/navigate?category=particles	

Appendix Z : Context Model Map

Electro Magnetic Interference (EMI)/ <i>Electro Magnetic Pulse (EMP)</i>				/context_select/navigate?category=clutter	
Land Environment					
Surface Characteristics					
Concrete	<p>Obstacle surface material descriptions included in obstacle JSONs in \Models\Land\obstacles\data\profiles\ cross-reference with \Models\Land\obstacles\data\ObstacleMetadata.xml crg model files located in \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ cross-reference with \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx Terrain profile in \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced cross-reference with \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	<p>\Models\Land\obstacles\data\ObstacleMetadata.xml \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	<p>\Models\Land\terrains\grades_slopes\interface\Mod elica\FrictionTerraincrawler.mo</p>	search on "concrete"	search on "concrete"
Paved	<p>Obstacle surface material descriptions included in obstacle JSONs in \Models\Land\obstacles\data\profiles\ cross-reference with \Models\Land\obstacles\data\ObstacleMetadata.xml crg model files located in \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ cross-reference with \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx Terrain profile in \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced cross-reference with \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	<p>\Models\Land\obstacles\data\ObstacleMetadata.xml \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	<p>\Models\Land\terrains\grades_slopes\interface\Mod elica\FrictionTerraincrawler.mo</p>	search on "paved"	search on "paved"

Appendix Z : Context Model Map

Dirt	<p>Obstacle surface material descriptions included in obstacle JSONs in \\Models\\Land\\obstacles\\data\\profiles\\ cross-reference with \\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml</p> <p>crg model files located in \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\crg-lx\\ cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx</p> <p>Terrain profile in \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced</p> <p>cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx</p>	\\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx	\\Models\\land\\terrains\\grades_slopes\\interface\\Mod elica\\FrictionTerraincrawler.mo	search on "dirt" or "soil"	search on "dirt" or "soil"
Sand	<p>Obstacle surface material descriptions included in obstacle JSONs in \\Models\\Land\\obstacles\\data\\profiles\\ cross-reference with \\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml</p> <p>crg model files located in \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\crg-lx\\ cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx</p> <p>Terrain profile in \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced</p> <p>cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx</p>	\\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx	\\Models\\land\\terrains\\grades_slopes\\interface\\Mod elica\\FrictionTerraincrawler.mo	search on "sand" or "silt"	search on "sand" or "silt"
Wet	<p>surface material descriptions included in obstacle JSONs in \\Models\\Land\\obstacles\\data\\profiles\\ cross-reference with \\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml</p> <p>crg model files located in \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\crg-lx\\ cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx</p> <p>Terrain Profile in \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced</p> <p>cross-reference with \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx</p>	\\Models\\Land\\obstacles\\data\\ObstacleMetadata.xml \\Models\\Land\\terrains\\grades_slopes\\data\\OpenCRG\\CRG metadata.xlsx \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\LogSpaced \\Models\\Land\\terrains\\grades_slopes\\data\\WaveNumberSpectra\\Terrain PSD metadata.xlsx	\\Models\\land\\terrains\\grades_slopes\\interface\\Mod elica\\FrictionTerraincrawler.mo	search on "wet" or "rain"	search on "wet" or "rain"

Appendix Z : Context Model Map

Mud	<p>surface material descriptions included in obstacle JSONs in \Models\Land\obstacles\data\profiles\ cross-reference with \Models\Land\obstacles\data\ObstacleMetadata.xml crg model files located in \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ cross-reference with \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx Terrain Profile in \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced cross-reference with \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	\Models\Land\obstacles\data\ObstacleMet adata.xml \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx	\Models\land\terrains\gra des_slopes\interface\Mod elica\FrictionTerraincrawl er.mo	search on "mud" or "clay"	search on "mud" or "clay"
Snow	<p>surface material descriptions included in obstacle JSONs in \Models\Land\obstacles\data\profiles\ cross-reference with \Models\Land\obstacles\data\ObstacleMetadata.xml crg model files located in \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ cross-reference with \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx Terrain profile in \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced cross-reference with \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	\Models\Land\obstacles\data\ObstacleMet adata.xml \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx	\Models\land\terrains\gra des_slopes\interface\Mod elica\FrictionTerraincrawl er.mo	search on "snow"	search on "snow"
Ice	<p>surface material descriptions included in obstacle JSONs in \Models\Land\obstacles\data\profiles\ cross-reference with \Models\Land\obstacles\data\ObstacleMetadata.xml crg model files located in \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ cross-reference with \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx Terrain profile in \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced cross-reference with \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx</p>	\Models\Land\obstacles\data\ObstacleMet adata.xml \Models\Land\terrains\grades_slopes\data\OpenCRG\CRG metadata.xlsx \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx	\Models\land\terrains\gra des_slopes\interface\Mod elica\FrictionTerraincrawl er.mo	search on "ice"	search on "ice"

Appendix Z : Context Model Map

Discrete Obstacles (Forward and Reverse, and at Angles)					
Step Climb	\Models\Land\obstacles\data\profiles\ATC_MTA_18InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_24InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_36InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_42InchWall.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "step" or "stair"
Step Descend	\Models\Land\obstacles\data\profiles\ATC_MTA_18InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_24InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_36InchWall.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_42InchWall.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "step" or "stair"
Gap Crossing	\Models\Land\obstacles\data\profiles\ATC_MTA_BridgingDeviceExit.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_BridgingDeviceEntry.JSON \Models\Land\obstacles\exec\DiscreteObstacle\ATC_Bridging_Device.txt \Models\Land\obstacles\exec\DiscreteObstacle\RunDiscreteObstacles.bat	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "gap"
V-Ditch	\Models\Land\obstacles\data\profiles\ATC_MTA_StandardDitch.JSON \Models\Land\obstacles\exec\AngleApproach\RunAngleApproach.bat	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "ditch"

Appendix Z : Context Model Map

Half-Round	\Models\Land\obstacles\data\profiles\ATC_MTA_4_Inch_Half_Round.JSON \Models\Land\obstacles\data\profiles\YTC_MOUT_9inch_Half_Rounds.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_6_Inch_Half_Round.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_8_Inch_Half_Round.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_10_Inch_Half_Round.JSON \Models\Land\obstacles\data\profiles\ATC_MTA_12_Inch_Half_Round.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "half-round"
Curb	\Models\Land\obstacles\data\profiles\YTC_MOUT_Curb.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "curb"
Features found in MOUT (Military Operations in Urban Terrain)	\Models\Land\obstacles\data\profiles\YTC_MOUT_Staircase.JSON \Models\Land\obstacles\data\profiles\YTC_MOUT_9inch_Half_Rounds.JSON \Models\Land\obstacles\data\profiles\YTC_MOUT_Curb.JSON \Models\Land\obstacles\data\profiles\YTC_MOUT_Log.JSON \Models\Land\obstacles\data\profiles\YTC_MOUT_Washboard.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "mout"
Jersey Barrier (Highway Divider)	\Models\Land\obstacles\data\profiles\Jersey_BARRIER_inset.JSON \Models\Land\obstacles\data\profiles\Jersey_BARRIER_whole.JSON	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "jersey"

Appendix Z : Context Model Map

Improvised Obstacles (e.g., passenger cars)	See MOUT	\Models\Land\obstacles\doc\Discrete_Obstacles.pptx \Models\Land\obstacles\doc\Demo_DiscreteObstaclesPitchPlane.docx \Models\Land\terrains\grades_slopes\doc\Demo_DriverSpectraNObstacles.docx \Models\Land\obstacles\data\ObstacleMetadata.xlsx	\Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles	/context_obstacles/navigate	search on "mout"
Terrains					
Terrains of varying roughness (Flat to 5" in rms)	TC_Belgian_Block_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Churchville_B_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Churchville_C_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Munson_Gravel_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Perryman_1_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Perryman_2_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Perryman_3_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\ATC_Perryman_A_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\YTC_Desert_March_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\YTC_KOFA_Level_Gravel_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\YTC_Laguna_Hilly_Trails_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\YTC_Laguna_Level_logspaced.JSON \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\LogSpaced\YTC_Laguna_Level_Trails_East_logspaced.JSON		\Models\Land\terrains\grades_slopes\RunDrvLmtdSpdSpctr.bat \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\terrains\grades_slopes\RunSixWattStandAlone.bat \Models\Land\terrains\grades_slopes\RunSixWattStandAlone.docx \Models\Land\terrains\grades_slopes\TerrainGeneratorApp \Models\Land\terrains\grades_slopes\tools\TerrainServer \Models\Land\terrains\grades_slopes\tools\TerrainPSD \Models\Land\terrains\grades_slopes\data\WaveNumberSpectra\Terrain PSD metadata.xlsx	/context_psd_workflow/navigate	search "PSD"

Appendix Z : Context Model Map

Longitudinal Grades (Forward and Reverse)	\Models\Land\obstacles\data\profiles\ATC_10%SoilSlope.JSON \Models\Land\obstacles\data\profiles\ATC_15%SoilSlope.JSON \Models\Land\obstacles\data\profiles\ATC_20%SoilSlope.JSON \Models\Land\obstacles\data\profiles\ATC_25%SoilSlope.JSON \Models\Land\obstacles\data\profiles\ATC_60%SoilSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA5%AsphaltSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA10%AsphaltSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA15%AsphaltSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA20%AsphaltSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA30%ConcreteSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA40%ConcreteSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA45%ConcreteSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA50%ConcreteSlope.JSON \Models\Land\obstacles\data\profiles\ATC_MTA60%ConcreteSlope.JSON	\Models\Land\terrains\grades_slopes\test\Fuel_Efficiency \Models\Land\obstacles\test\PitchPlaneModel \Models\Land\terrains\grades_slopes\test\DriverSpectraNObstacles \Models\Land\obstacles\src\DiscreteObstacles \Models\Land\terrains\grades_slopes\toolsTerrainServer \Models\Land\terrains\grades_slopes\tools\Unity\RoadExperiments Demo_Fuel_Efficiency.docx Demo_DriverSpectraNObstacles.docx	\Models\Land\terrains\grades_slopes\test\Amphibious3DVehicle \Models\Land\terrains\grades_slopes\tools\TerrainServer \Models\Land\terrains\grades_slopes\tools\Unity\RoadExperiments	/context_obstacles/navigate	search "slope"
Side-to-Side Slopes (Either side up-hill)	\Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ATC_20pctSideSlope.crg \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ATC_30pctSideSlope.crg \Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\ATC_40pctSideSlope.crg \Models\Land\obstacles\data\profiles\ATC_40%SoilSideSlope.JSON \Models\Land\obstacles\data\profiles\ATC_30%SoilSideSlope.JSON	\Models\Land\terrains\grades_slopes\data	\Models\Land\terrains\grades_slopes\test\Amphibious3DVehicle \Models\Land\terrains\grades_slopes\tools\TerrainServer \Models\Land\terrains\grades_slopes\tools\Unity\RoadExperiments	/context_obstacles/navigate	search "slope"
Combined Grade and Slope (Fore-Aft and Side- to-Side)	\Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\Fictitious_slopebank.crg	\Models\Land\terrains\grades_slopes\data	\Models\Land\terrains\grades_slopes\test\Amphibious3DVehicle \Models\Land\terrains\grades_slopes\tools\TerrainServer \Models\Land\terrains\grades_slopes\tools\Unity\RoadExperiments	/context_obstacles/navigate	search "slope"

Appendix Z : Context Model Map

			\Models\Land\terrains\grades_slopes\test\Amphibious3DVehicle		
Curvature (Turns, Crown, Trough)	\Models\Land\terrains\grades_slopes\data\OpenCRG\crg-txt\Fictitious_Turning_Course.crg	\Models\Land\terrains\grades_slopes\data	\Models\Land\terrains\grades_slopes\tools\TerrainServer	/context_obstacles/navigate	search "turn"
Aquatic Environment			\Models\Land\terrains\grades_slopes\tools\Unity\RoadExperiments		
Water Properties					
Density	\Models\Composite\FluidThermal\src\seaWaterTPProps.h \Models\Composite\FluidThermal\src\seaWaterTPProps.cpp \Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx	\Models\Composite\FluidThermal\doc\Sea water_Property_Tables.pdf \Models\Composite\FluidThermal\src\class)SeaWaterTPProps.docx	\Models\Composite\FluidThermal\test\HeatExchanger	/context_select/navigate?category=wave	
Temperature				/context_select/navigate?category=wave	
Viscosity	\Models\Composite\FluidThermal\src\seaWaterTPProps.h \Models\Composite\FluidThermal\src\seaWaterTPProps.cpp	\Models\Composite\FluidThermal\doc\Sea water_Property_Tables.pdf \Models\Composite\FluidThermal\src\class)SeaWaterTPProps.docx	\Models\Composite\FluidThermal\test\HeatExchanger	/context_select/navigate?category=wave	
Thermal Conductivity	\Models\Composite\FluidThermal\src\seaWaterTPProps.h \Models\Composite\FluidThermal\src\seaWaterTPProps.cpp	\Models\Composite\FluidThermal\doc\Sea water_Property_Tables.pdf \Models\Composite\FluidThermal\src\class)SeaWaterTPProps.docx	\Models\Composite\FluidThermal\test\HeatExchanger	/context_physical/navigate	
Specific Heat	\Models\Composite\FluidThermal\src\seaWaterTPProps.h \Models\Composite\FluidThermal\src\seaWaterTPProps.cpp	\Models\Composite\FluidThermal\doc\Sea water_Property_Tables.pdf \Models\Composite\FluidThermal\src\class)SeaWaterTPProps.docx	\Models\Composite\FluidThermal\test\HeatExchanger	/context_physical/navigate	
Water Body Features					
Depth/Area				/context_select/navigate?category=lakes /context_select/navigate?category=wave	
Calm	\Models\Aquatic\features\calm\src\drag	\Models\Aquatic\features\calm\doc\waterMobility.pptx		search "calm"	
Surf	\Models\Aquatic\features\surf\tools\Unity\SurfShore \Models\Aquatic\features\surfexec\SurfToShore		\Models\Aquatic\features\surf\tools\Unity\SurfShore \Models\Aquatic\features\surfexec\SurfToShore	search "surf"	

Appendix Z : Context Model Map

Currents				/context_fording/navigate	
Sea-State	\svn\Models\Aquatic\features\sea_state\src\PotentialFlowSea\PotentialFlowSea.cpp \svn\Models\Aquatic\features\sea_state\src\PotentialFlowSea\PotentialFlowSea.h		\Models\Aquatic\features\sea_state\test\PotentialFlowSeaApp	/context_select/navigate?category=wave	
Contaminants					
Salt	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	/context_water/navigate	
Particulates (Sand, Volcanic Ash)	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	/context_select/navigate?category=particles	
Debris (Vegetation, Spills)	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	\Models\Aquatic\properties\buoyancy\amphibious-context-model.xlsx - see all .xlsx files and Summary	search "vegetation" or "debris" /context_friction/coefficient_friction_table /context_water/densities	