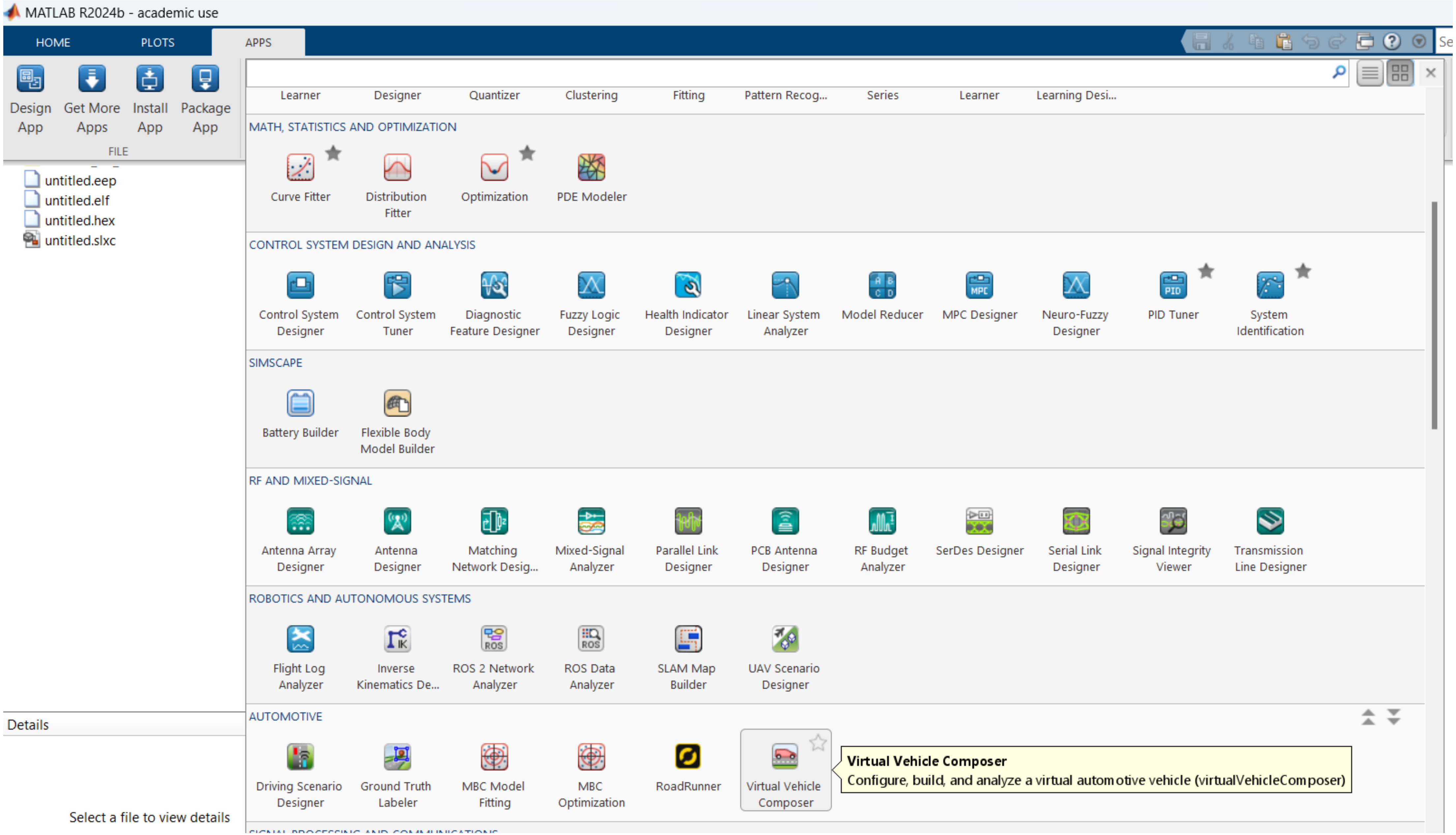
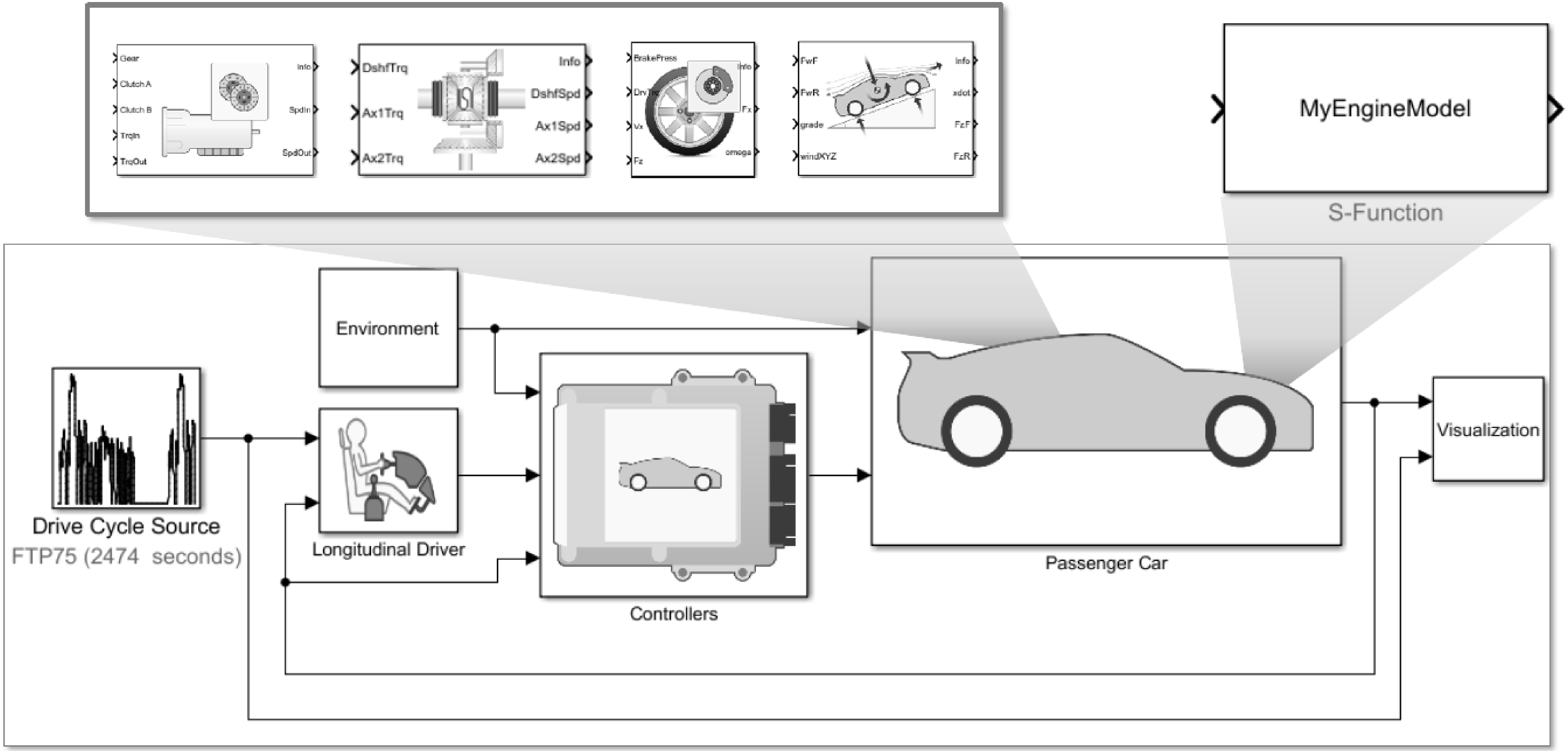
Powertrain MATLAB

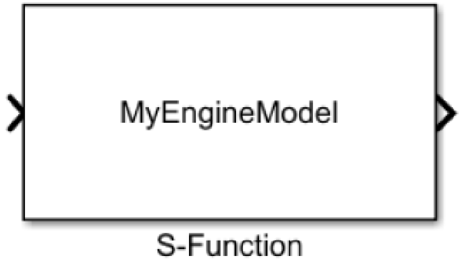
Prof. Juan Moises Mauricio Villanueva

E-mail: jmauricio@cear.ufpb.br


MATLAB R2024b - academic use
HOME
PLOTS
APPS
Design App
Get More Apps
Install App
Package App
FILE
untitled.eep
untitled.elf
untitled.hex
untitled.slxc
Details
Select a file to view details
Learner
Designer
Quantizer
Clustering
Fitting
Pattern Recog...
Series
Learner
Learning Desi...
MATH, STATISTICS AND OPTIMIZATION
Curve Fitter
Distribution Fitter
Optimization
PDE Modeler
CONTROL SYSTEM DESIGN AND ANALYSIS
Control System Designer
Control System Tuner
Diagnostic Feature Designer
Fuzzy Logic Designer
Health Indicator Designer
Linear System Analyzer
Model Reducer
MPC Designer
Neuro-Fuzzy Designer
PID Tuner
System Identification
SIMSCAPE
Battery Builder
Flexible Body Model Builder
RF AND MIXED-SIGNAL
Antenna Array Designer
Antenna Designer
Matching Network Desig...
Mixed-Signal Analyzer
Parallel Link Designer
PCB Antenna Designer
RF Budget Analyzer
SerDes Designer
Serial Link Designer
Signal Integrity Viewer
Transmission Line Designer
ROBOTICS AND AUTONOMOUS SYSTEMS
Flight Log Analyzer
Inverse Kinematics De...
ROS 2 Network Analyzer
ROS Data Analyzer
SLAM Map Builder
UAV Scenario Designer
AUTOMOTIVE
Driving Scenario Designer
Ground Truth Labeler
MBC Model Fitting
MBC Optimization
RoadRunner
Virtual Vehicle Composer
Virtual Vehicle Composer
Configure, build, and analyze a virtual automotive vehicle (virtualVehicleComposer)

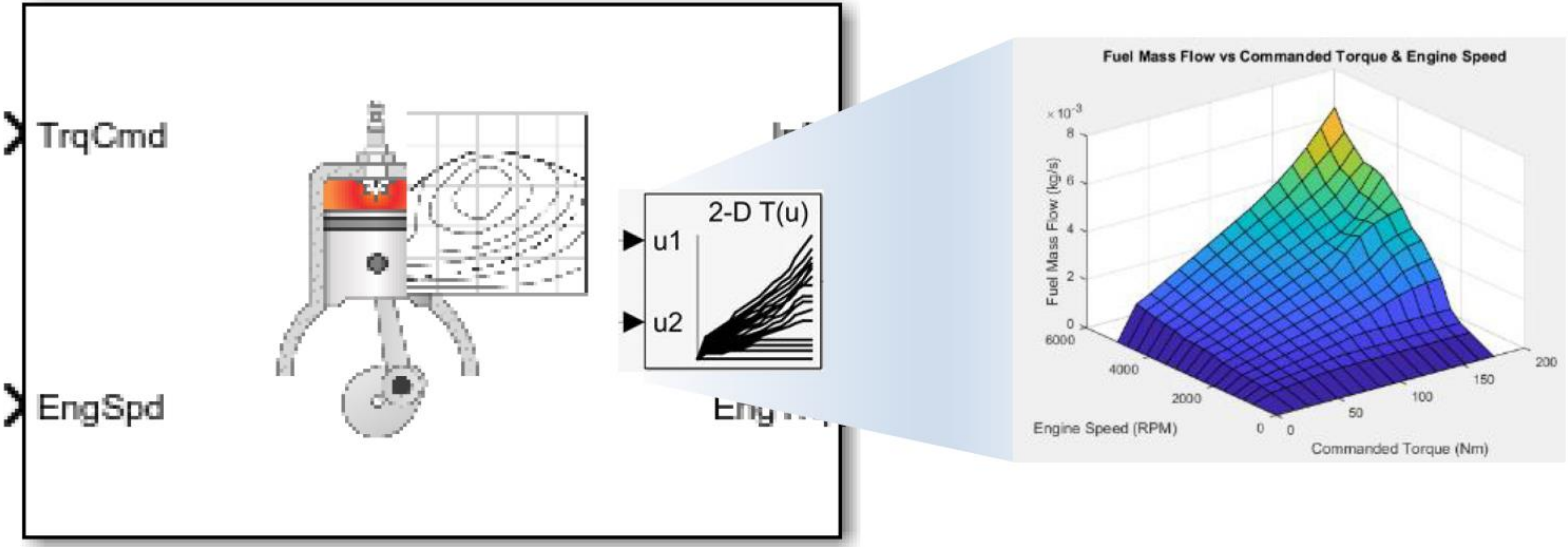
Controls Validation with Engine Model Co-Simulation



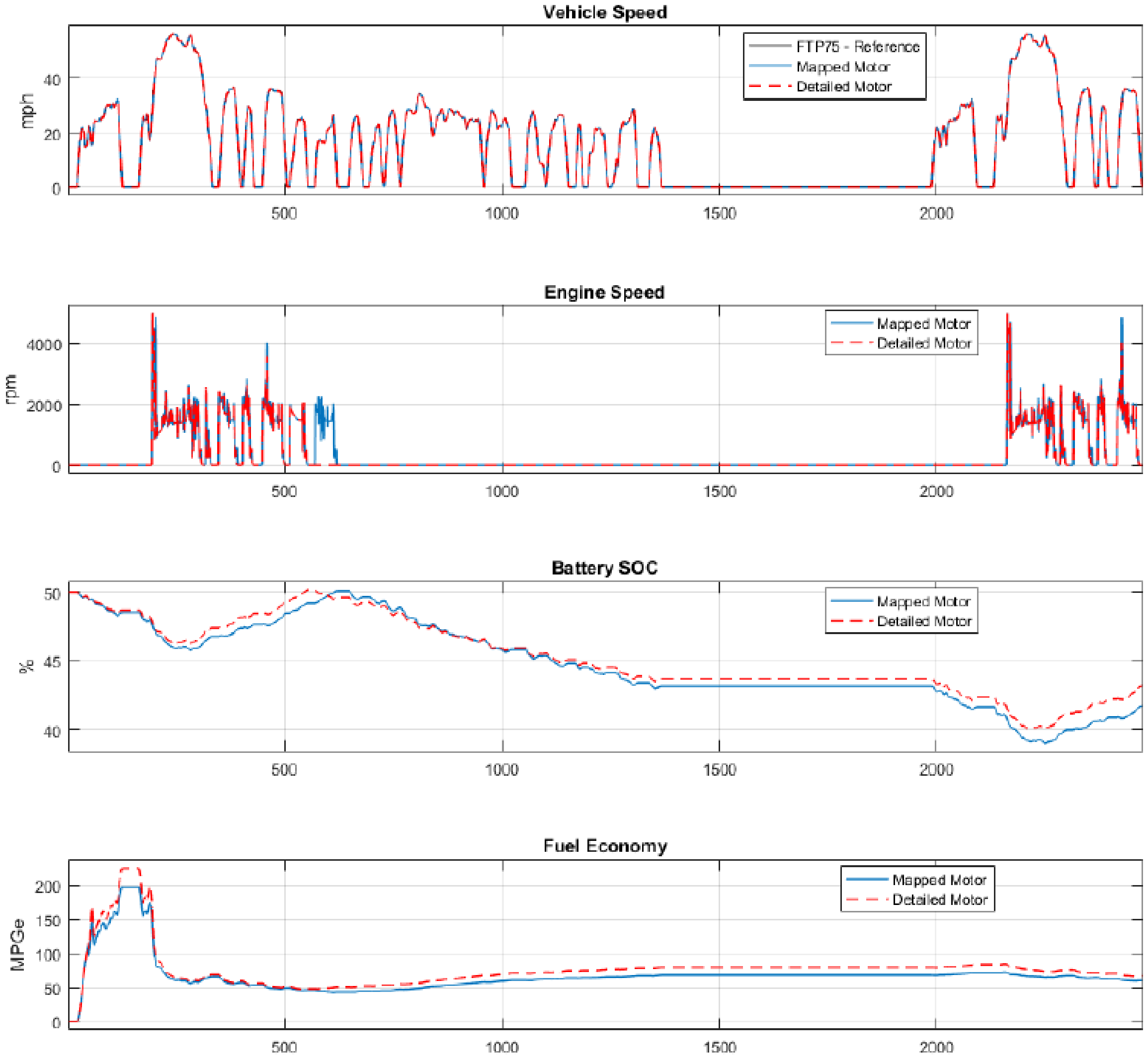


Detailed, design-oriented model

Fast, but accurate controls-oriented model

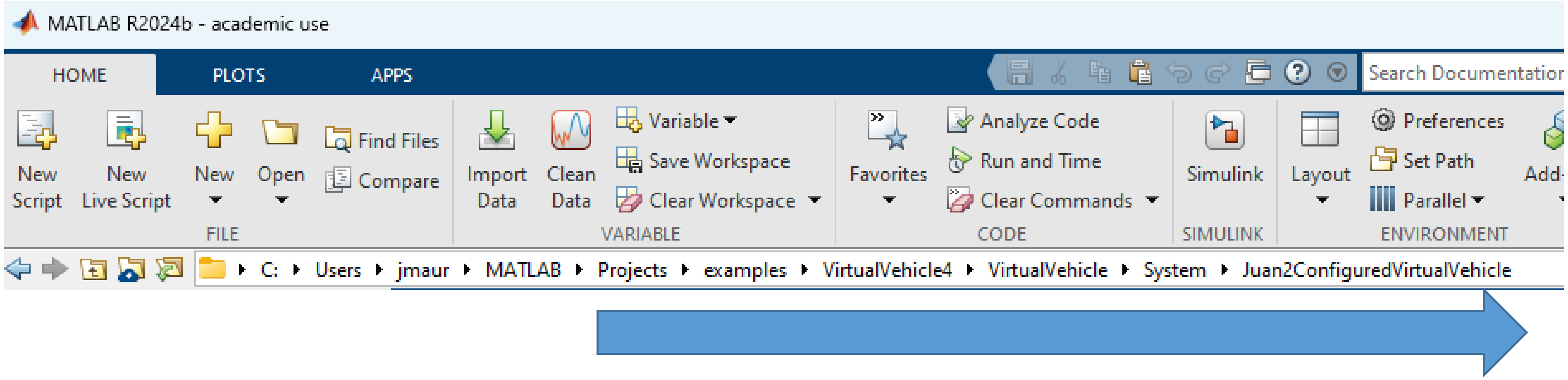


Detailed Model Variant Simulation

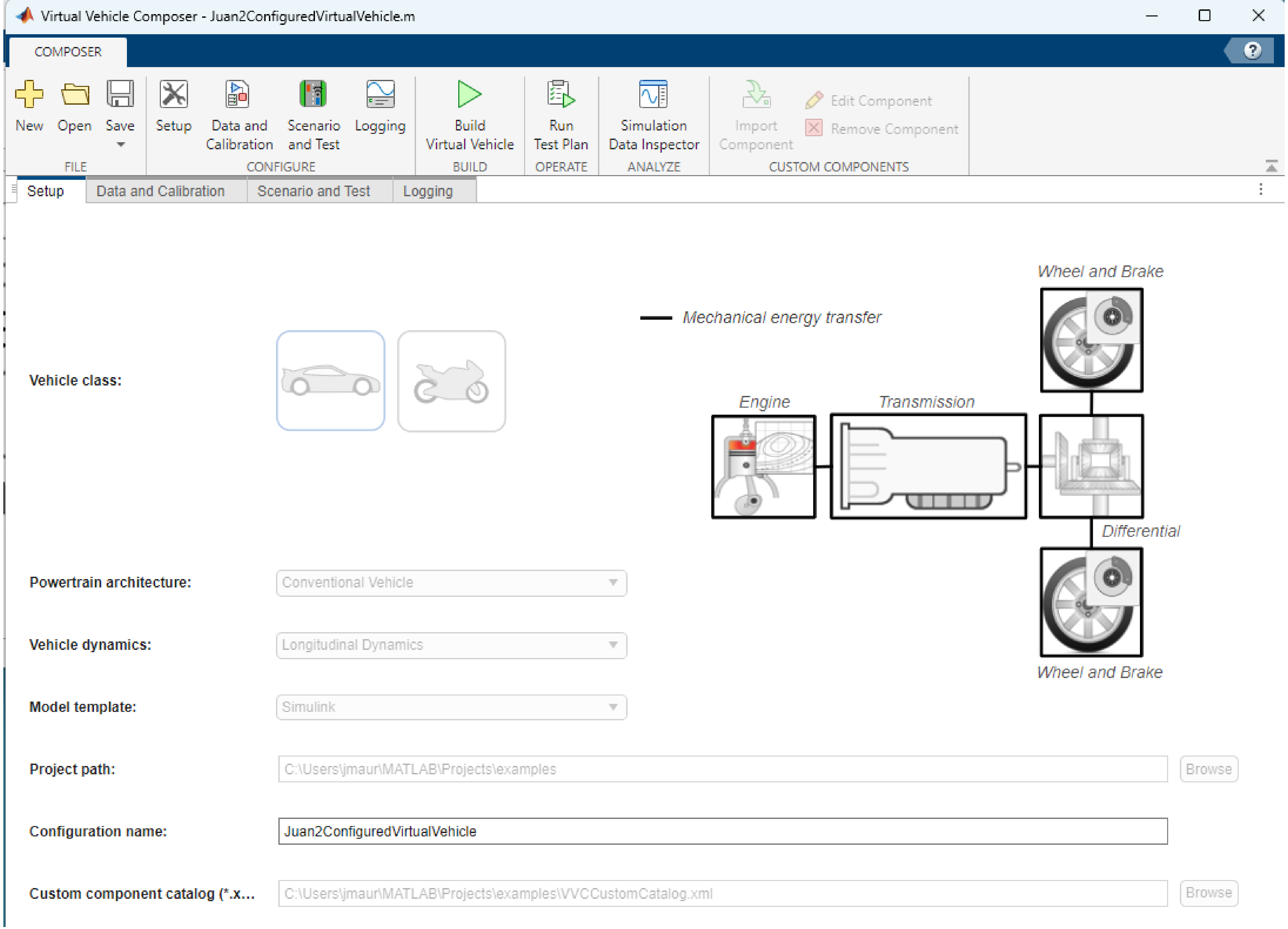


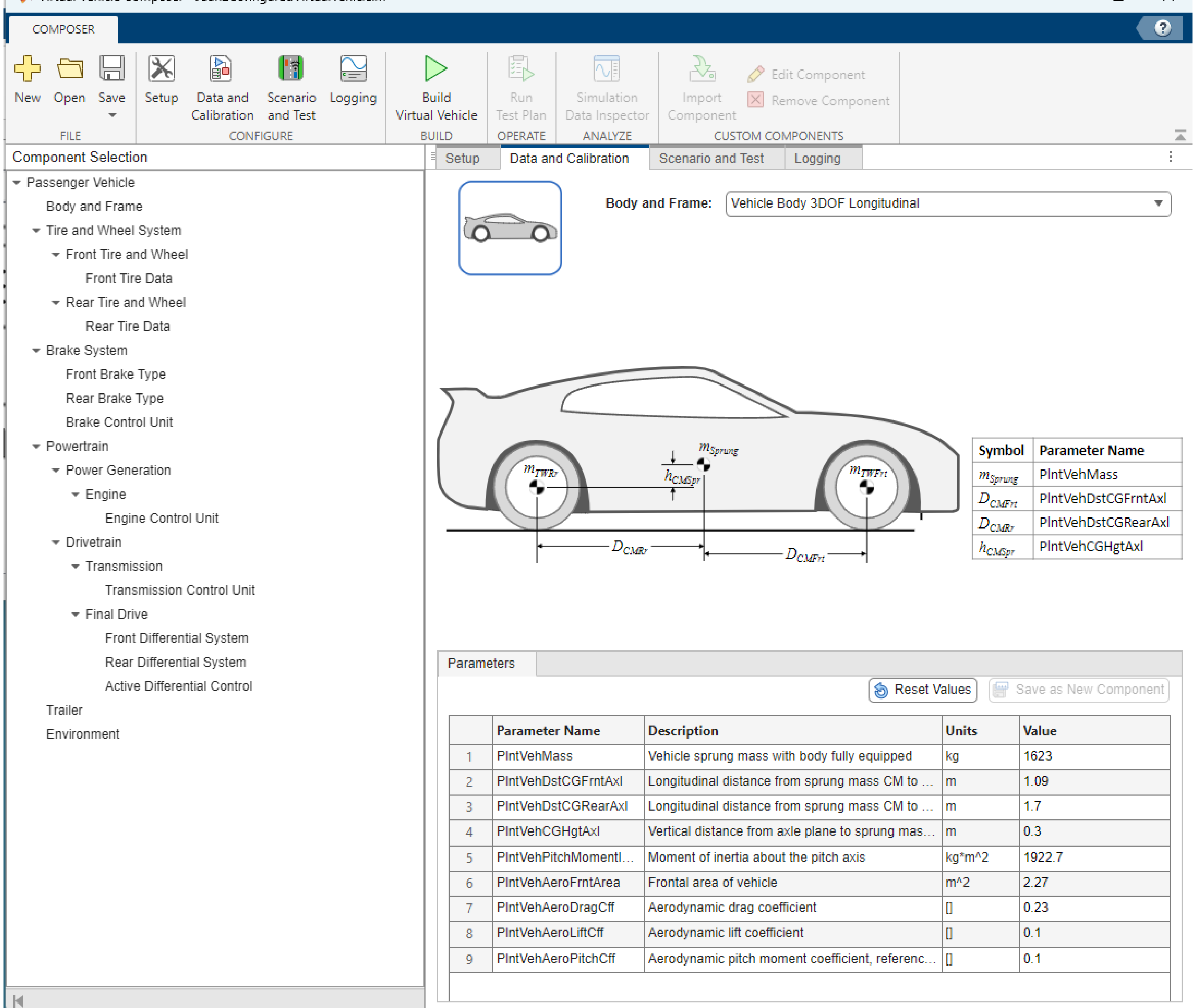
Cycle Name	Final SOC (%)		MPGe	
	Mapped	Detailed	Mapped	Detailed
HWFET	42	44	50.5	51.8
FTP75	41.4	42.8	59.6	66.4

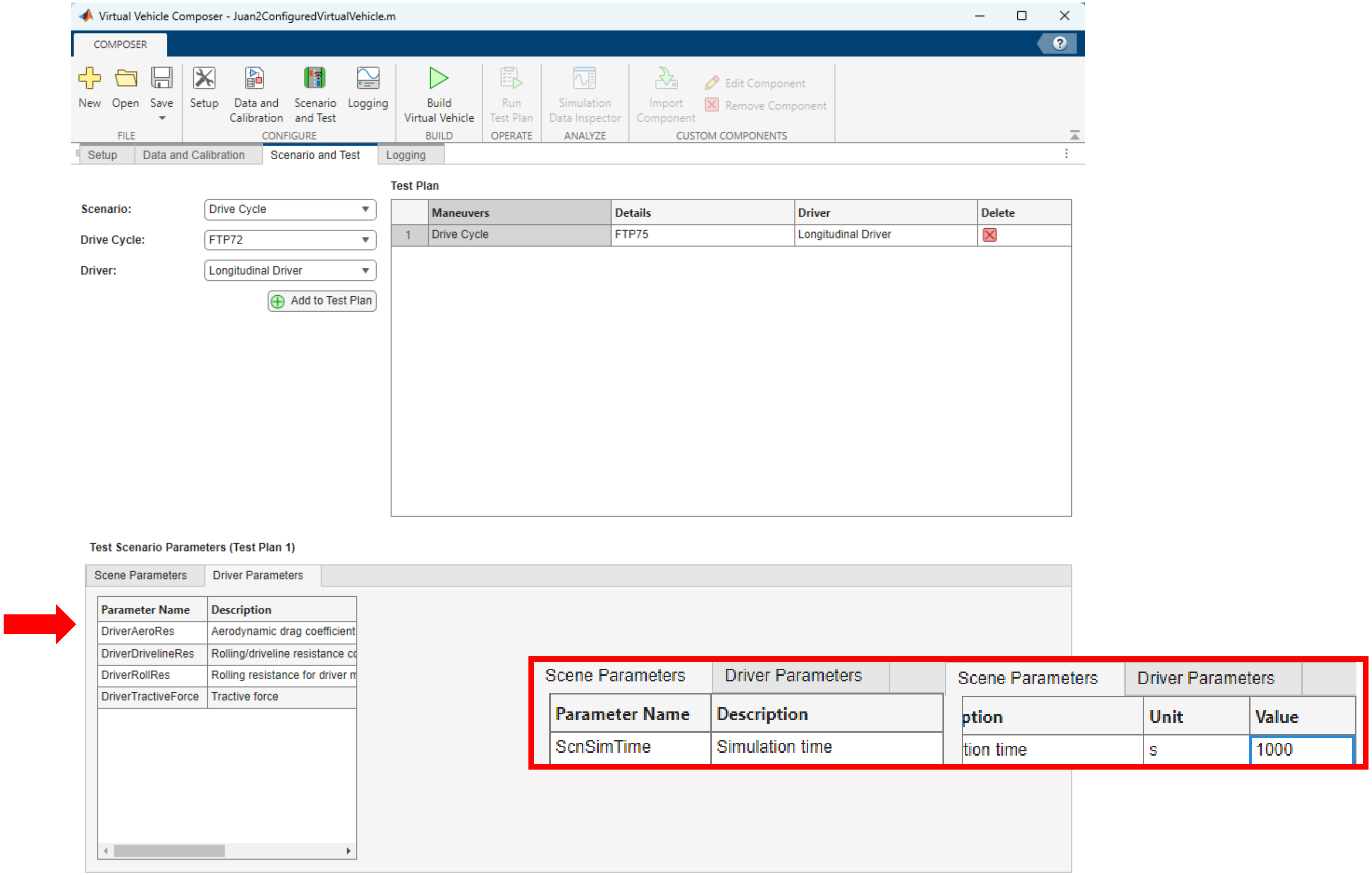
- Detailed variant gives comparable response
- Supervisory controller handles both motor variants
- Motor controller requires further verification


MATLAB R2024b - academic use
HOME
PLOTS
APPS
Search Documentatior
New Script
New Live Script
New
Open
Find Files
Compare
FILE
Import Data
Clean Data
Variable
Save Workspace
Clear Workspace
VARIABLE
Favorites
Analyze Code
Run and Time
Clear Commands
CODE
Simulink
SIMULINK
Layout
Preferences
Set Path
Parallel
ENVIRONMENT
Add-
C: ▸ Users ▸ jmaur ▸ MATLAB ▸ Projects ▸ examples ▸ VirtualVehicle4 ▸ VirtualVehicle ▸ System ▸ Juan2ConfiguredVirtualVehicle

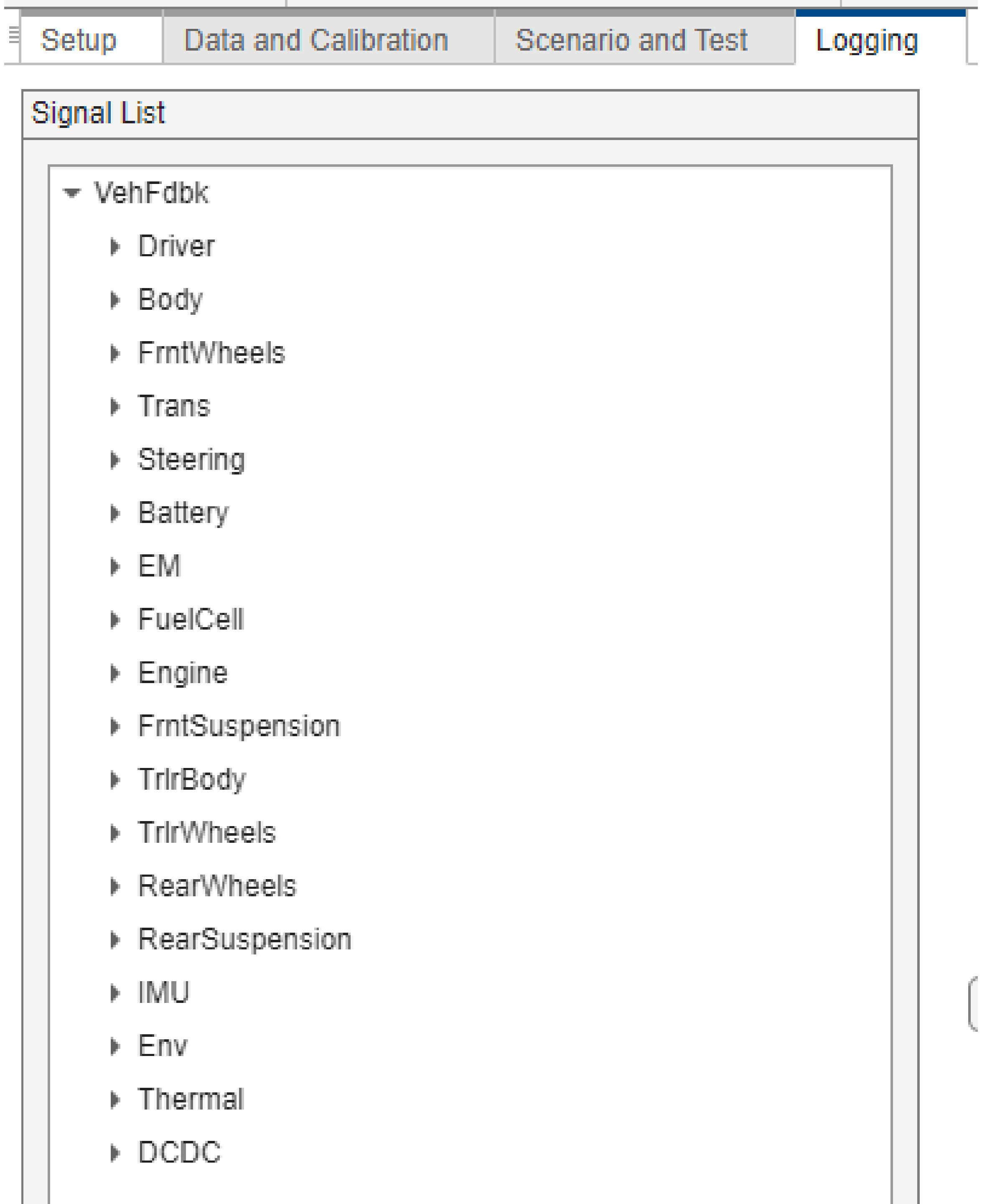
Modelo Inicial

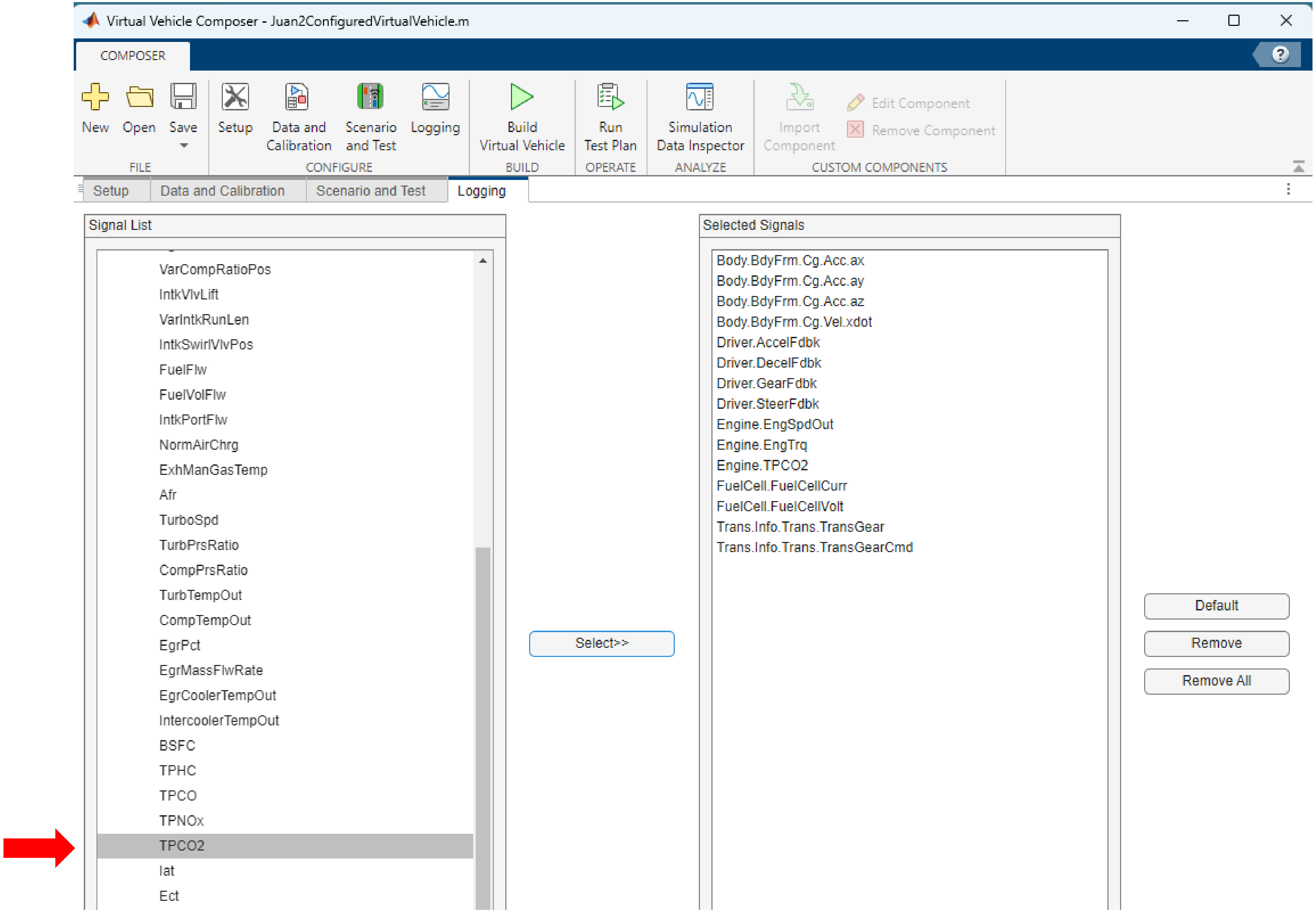

Virtual Vehicle Composer - Juan2ConfiguredVirtualVehicle.m
COMPOSER
New
Open
Save
FILE
Setup
Data and Calibration
Scenario and Test
Logging
CONFIGURE
Build Virtual Vehicle
BUILD
Run Test Plan
OPERATE
Simulation Data Inspector
ANALYZE
Import Component
Edit Component
Remove Component
CUSTOM COMPONENTS
Setup
Data and Calibration
Scenario and Test
Logging
Vehicle class:
Mechanical energy transfer
Wheel and Brake
Engine
Transmission
Differential
Wheel and Brake
Powertrain architecture:
Conventional Vehicle
Vehicle dynamics:
Longitudinal Dynamics
Model template:
Simulink
Project path:
C:\Users\jmaur\MATLAB\Projects\examples
Browse
Configuration name:
Juan2ConfiguredVirtualVehicle
Custom component catalog (*.x...
C:\Users\jmaur\MATLAB\Projects\examples\VVCCustomCatalog.xml
Browse


COMPOSER
New
Open
Save
FILE
Setup
Data and Calibration
Scenario and Test
Logging
CONFIGURE
Build Virtual Vehicle
BUILD
Run Test Plan
OPERATE
Simulation Data Inspector
ANALYZE
Import Component
Edit Component
Remove Component
CUSTOM COMPONENTS
Component Selection
Passenger Vehicle
Body and Frame
Tire and Wheel System
Front Tire and Wheel
Front Tire Data
Rear Tire and Wheel
Rear Tire Data
Brake System
Front Brake Type
Rear Brake Type
Brake Control Unit
Powertrain
Power Generation
Engine
Engine Control Unit
Drivetrain
Transmission
Transmission Control Unit
Final Drive
Front Differential System
Rear Differential System
Active Differential Control
Trailer
Environment
Setup
Data and Calibration
Scenario and Test
Logging
Body and Frame:
Vehicle Body 3DOF Longitudinal
m_{Sprung}
m_{TWRr}
h_{CMSpr}
m_{TWFrt}
D_{CMRr}
D_{CMFrt}
Symbol
Parameter Name
m_{Sprung} PlntVehMass
D_{CMFrt} PlntVehDstCGFrntAxl
D_{CMRr} PlntVehDstCGRearAxl
h_{CMSpr} PlntVehCGHgtAxl
Parameters
Reset Values
Save as New Component
Parameter Name | Description | Units | Value
1 PlntVehMass | Vehicle sprung mass with body fully equipped | kg | 1623
2 PlntVehDstCGFrntAxl | Longitudinal distance from sprung mass CM to ... | m | 1.09
3 PlntVehDstCGRearAxl | Longitudinal distance from sprung mass CM to ... | m | 1.7
4 PlntVehCGHgtAxl | Vertical distance from axle plane to sprung mas... | m | 0.3
5 PlntVehPitchMomentI... | Moment of inertia about the pitch axis | kg*m^2 | 1922.7
6 PlntVehAeroFrntArea | Frontal area of vehicle | m^2 | 2.27
7 PlntVehAeroDragCff | Aerodynamic drag coefficient | [] | 0.23
8 PlntVehAeroLiftCff | Aerodynamic lift coefficient | [] | 0.1
9 PlntVehAeroPitchCff | Aerodynamic pitch moment coefficient, referenc... | [] | 0.1

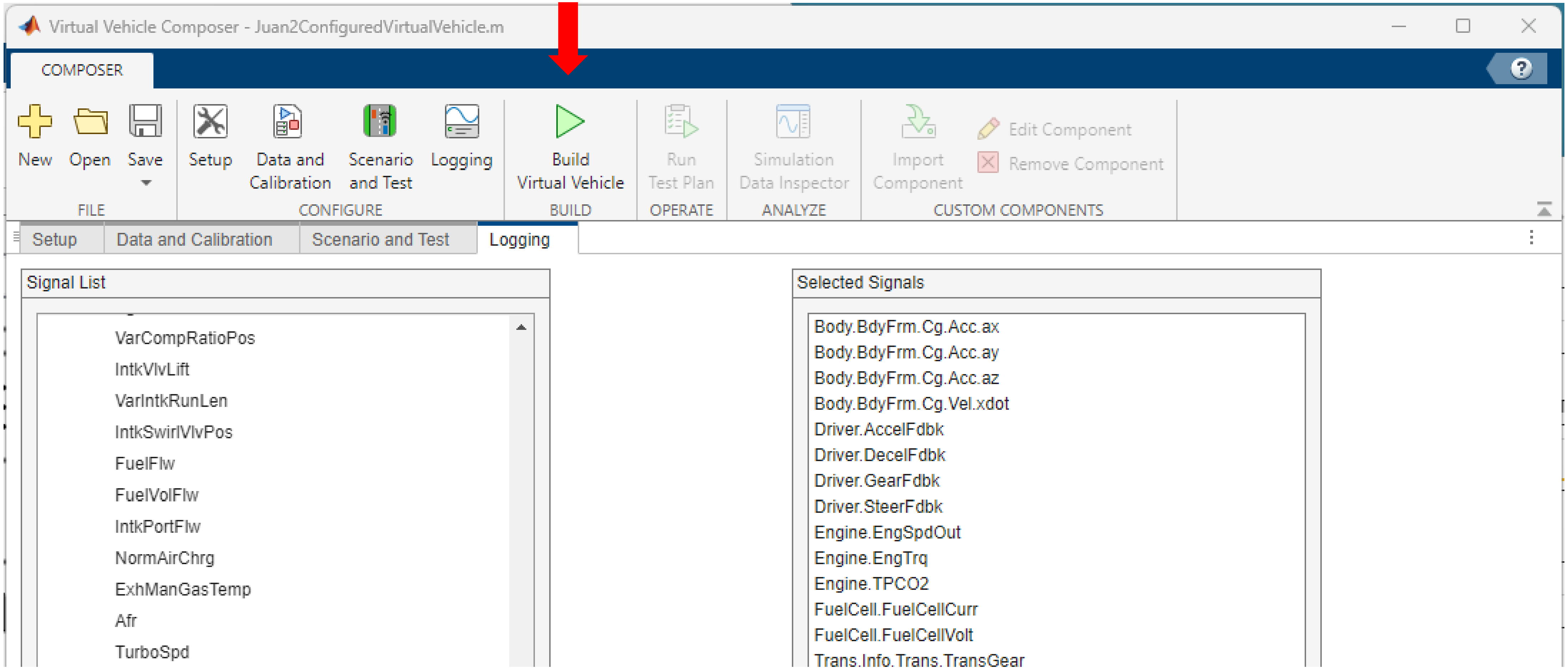


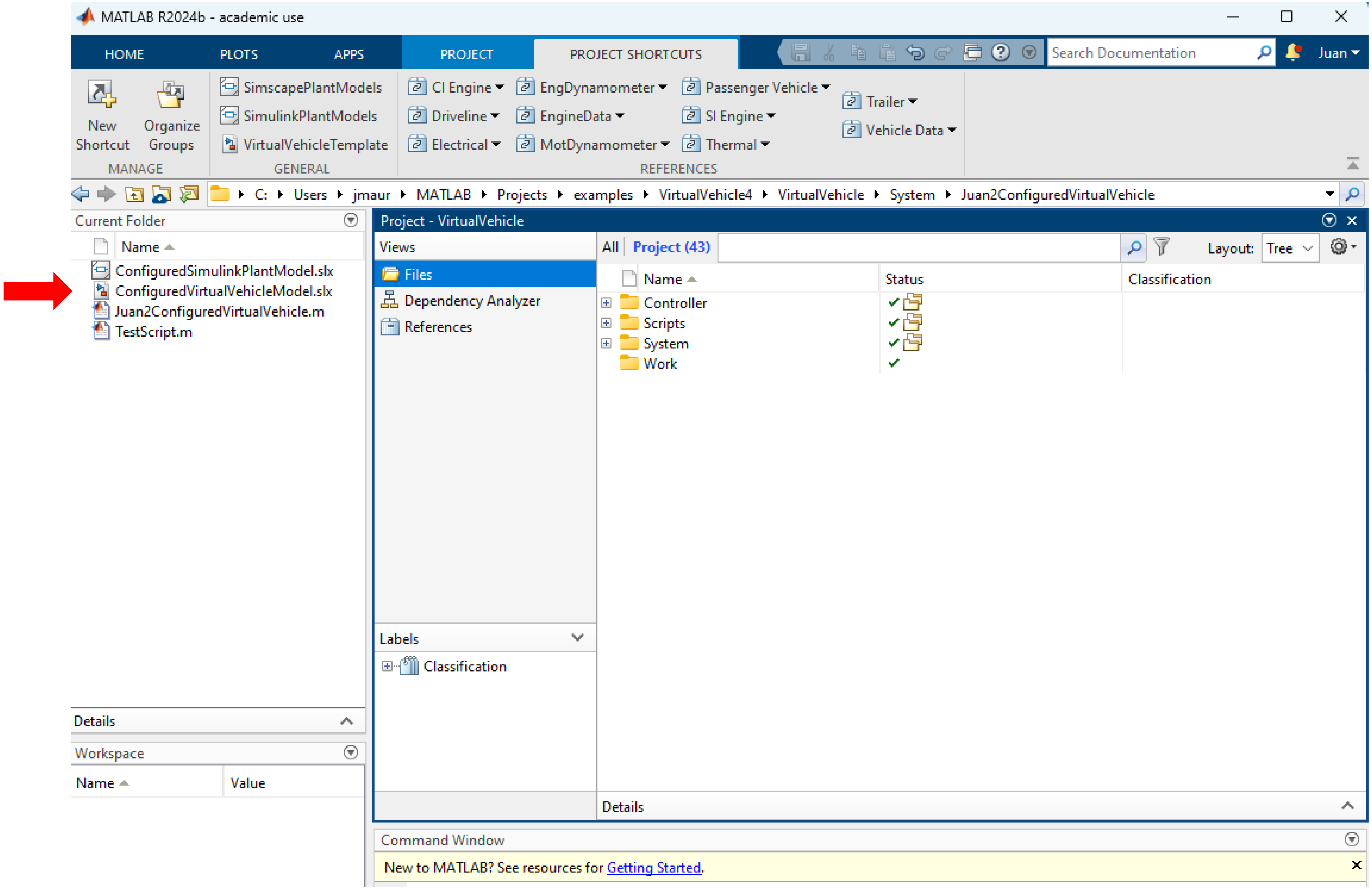
Virtual Vehicle Composer - Juan2ConfiguredVirtualVehicle.m
COMPOSER
New
Open
Save
FILE
Setup
Data and Calibration
Scenario and Test
Logging
CONFIGURE
Build Virtual Vehicle
BUILD
Run Test Plan
OPERATE
Simulation Data Inspector
ANALYZE
Import Component
Edit Component
Remove Component
CUSTOM COMPONENTS
Setup
Data and Calibration
Scenario and Test
Logging
Scenario:
Drive Cycle
Drive Cycle:
FTP72
Driver:
Longitudinal Driver
Add to Test Plan
Test Plan
Maneuvers
Details
Driver
Delete
1
Drive Cycle
FTP75
Longitudinal Driver
Test Scenario Parameters (Test Plan 1)
Scene Parameters
Driver Parameters
Parameter Name
Description
DriverAeroRes
Aerodynamic drag coefficient
DriverDrivelineRes
Rolling/driveline resistance co
DriverRollRes
Rolling resistance for driver m
DriverTractiveForce
Tractive force
Scene Parameters
Driver Parameters
Parameter Name
Description
ScnSimTime
Simulation time
Scene Parameters
Driver Parameters
ption
Unit
Value
tion time
s
1000

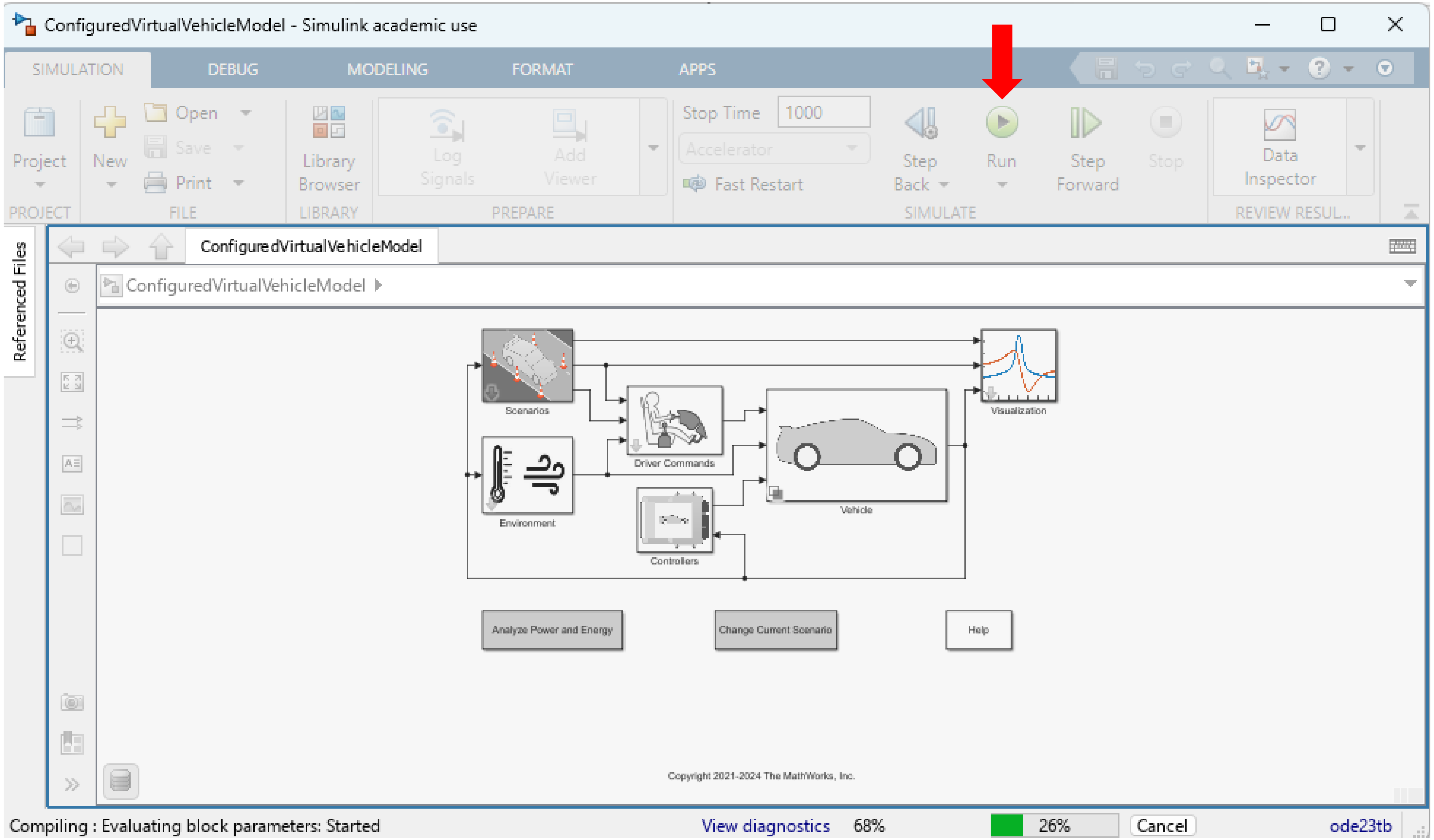

Setup
Data and Calibration
Scenario and Test
Logging
Signal List
VehFdbk
Driver
Body
FrntWheels
Trans
Steering
Battery
EM
FuelCell
Engine
FrntSuspension
TrlrBody
TrlrWheels
RearWheels
RearSuspension
IMU
Env
Thermal
DCDC

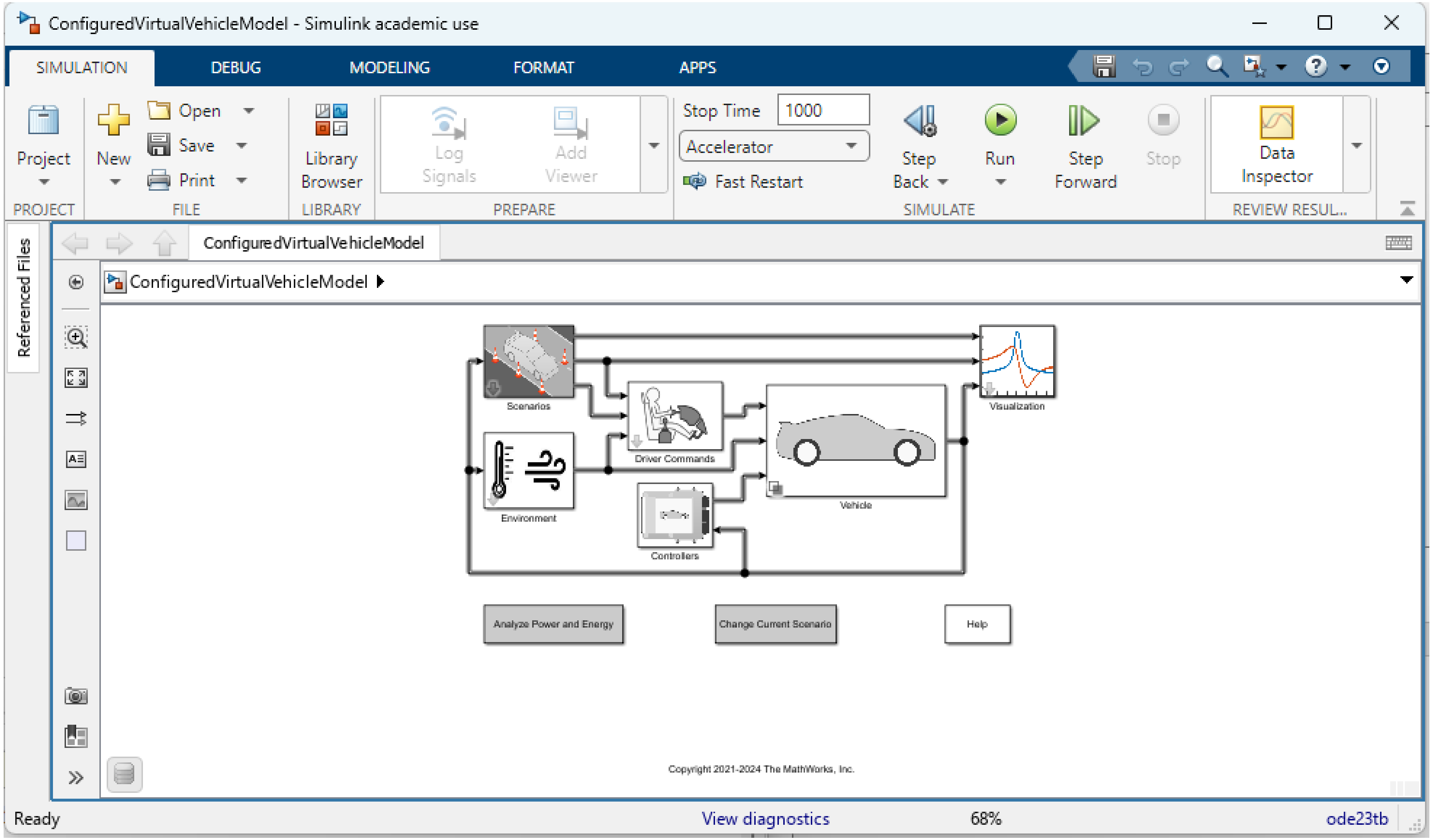

Virtual Vehicle Composer - Juan2ConfiguredVirtualVehicle.m
COMPOSER
New
Open
Save
FILE
Setup
Data and Calibration
Scenario and Test
Logging
CONFIGURE
Build Virtual Vehicle
BUILD
Run Test Plan
OPERATE
Simulation Data Inspector
ANALYZE
Import Component
Edit Component
Remove Component
CUSTOM COMPONENTS
Setup
Data and Calibration
Scenario and Test
Logging
Signal List
VarCompRatioPos
IntkVlvLift
VarIntkRunLen
IntkSwirlVlvPos
FuelFlw
FuelVolFlw
IntkPortFlw
NormAirChrg
ExhManGasTemp
Afr
TurboSpd
TurbPrsRatio
CompPrsRatio
TurbTempOut
CompTempOut
EgrPct
EgrMassFlwRate
EgrCoolerTempOut
IntercoolerTempOut
BSFC
TPHC
TPCO
TPNOx
TPCO2
Iat
Ect
Select>>
Selected Signals
Body.BdyFrm.Cg.Acc.ax
Body.BdyFrm.Cg.Acc.ay
Body.BdyFrm.Cg.Acc.az
Body.BdyFrm.Cg.Vel.xdot
Driver.AccelFdbk
Driver.DecelFdbk
Driver.GearFdbk
Driver.SteerFdbk
Engine.EngSpdOut
Engine.EngTrq
Engine.TPCO2
FuelCell.FuelCellCurr
FuelCell.FuelCellVolt
Trans.Info.Trans.TransGear
Trans.Info.Trans.TransGearCmd
Default
Remove
Remove All

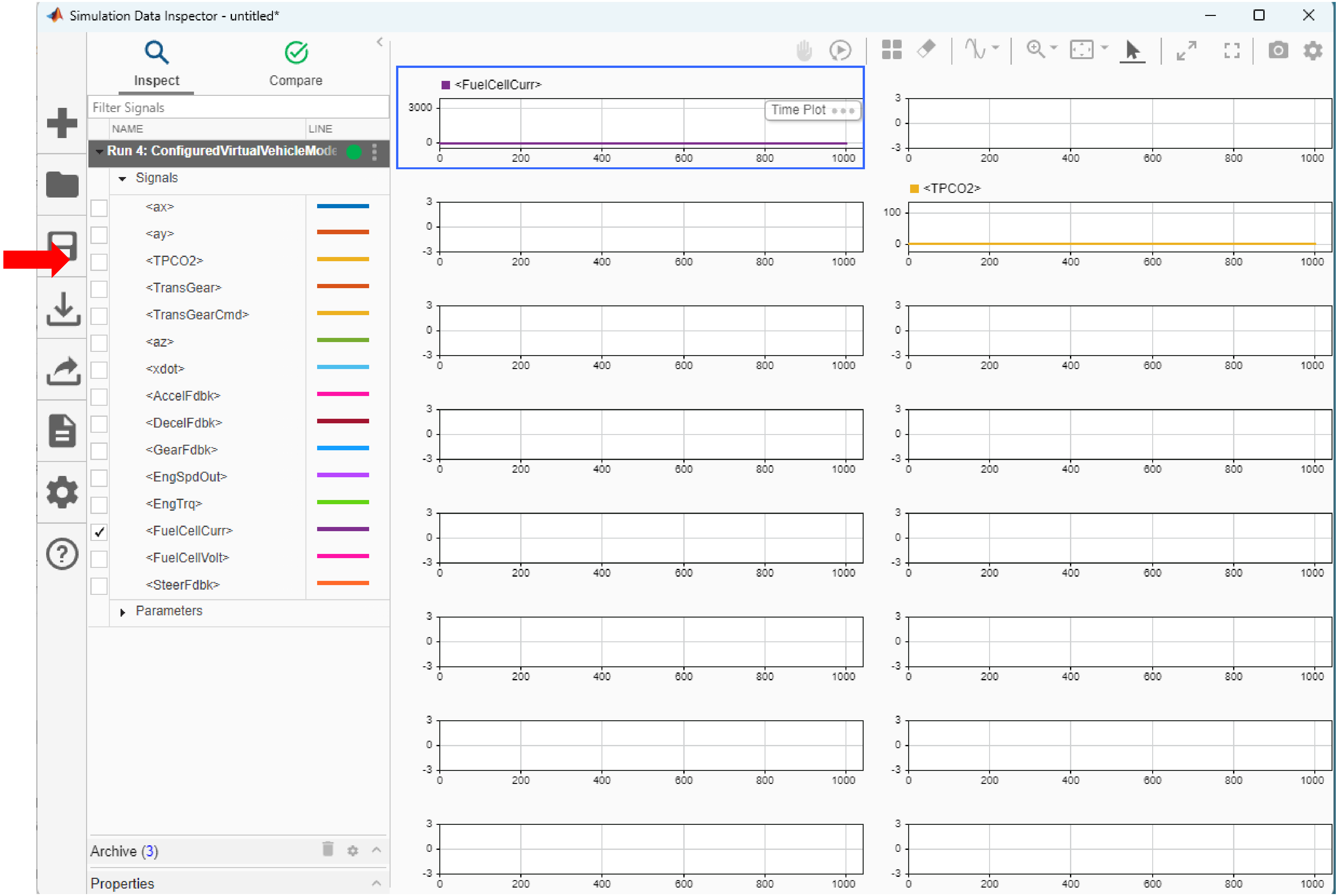
Build Virtual Vehicle

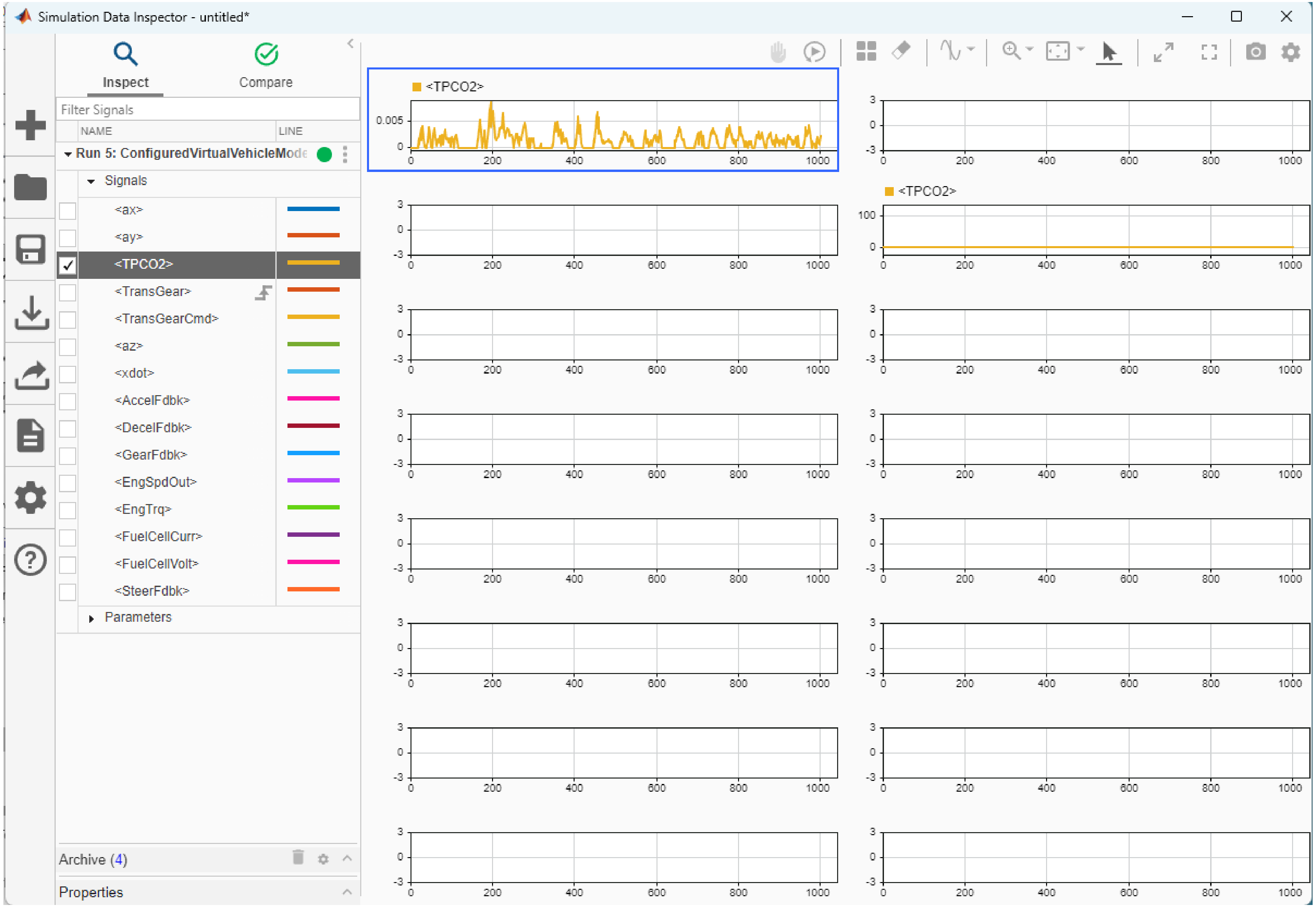



MATLAB R2024b - academic use
HOME
PLOTS
APPS
PROJECT
PROJECT SHORTCUTS
Search Documentation
Juan
New Shortcut
Organize Groups
MANAGE
SimscapePlantModels
SimulinkPlantModels
VirtualVehicleTemplate
GENERAL
CI Engine
Driveline
Electrical
EngDynamometer
EngineData
MotDynamometer
Passenger Vehicle
SI Engine
Thermal
Trailer
Vehicle Data
REFERENCES
C: ▸ Users ▸ jmaur ▸ MATLAB ▸ Projects ▸ examples ▸ VirtualVehicle4 ▸ VirtualVehicle ▸ System ▸ Juan2ConfiguredVirtualVehicle
Current Folder
Name
ConfiguredSimulinkPlantModel.slx
ConfiguredVirtualVehicleModel.slx
Juan2ConfiguredVirtualVehicle.m
TestScript.m
Details
Workspace
Name
Value
Project - VirtualVehicle
Views
Files
Dependency Analyzer
References
Labels
Classification
All
Project (43)
Layout:
Tree
Name
Status
Classification
Controller
Scripts
System
Work
Details
Command Window
New to MATLAB? See resources for Getting Started.


ConfiguredVirtualVehicleModel - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Project
New
Open
Save
Print
Library Browser
Log Signals
Add Viewer
Stop Time
1000
Accelerator
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
PROJECT
FILE
LIBRARY
PREPARE
SIMULATE
REVIEW RESUL...
Referenced Files
ConfiguredVirtualVehicleModel
ConfiguredVirtualVehicleModel
Scenarios
Driver Commands
Vehicle
Visualization
Environment
Controllers
Analyze Power and Energy
Change Current Scenario
Help
Copyright 2021-2024 The MathWorks, Inc.
Compiling : Evaluating block parameters: Started
View diagnostics
68%
26%
Cancel
ode23tb


ConfiguredVirtualVehicleModel - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
APPS
Project
New
Open
Save
Print
Library Browser
Log Signals
Add Viewer
Stop Time
1000
Accelerator
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
PROJECT
FILE
LIBRARY
PREPARE
SIMULATE
REVIEW RESUL...
Referenced Files
ConfiguredVirtualVehicleModel
Scenarios
Visualization
Driver Commands
Vehicle
Environment
Controllers
Analyze Power and Energy
Change Current Scenario
Help
Copyright 2021-2024 The MathWorks, Inc.
Ready
View diagnostics
68%
ode23tb


Simulation Data Inspector - untitled*
Inspect
Compare
Filter Signals
NAME
LINE
Run 4: ConfiguredVirtualVehicleMode
Signals
<ax>
<ay>
<TPCO2>
<TransGear>
<TransGearCmd>
<az>
<xdot>
<AccelFdbk>
<DecelFdbk>
<GearFdbk>
<EngSpdOut>
<EngTrq>
<FuelCellCurr>
<FuelCellVolt>
<SteerFdbk>
Parameters
Archive (3)
Properties
<FuelCellCurr>
Time Plot
<TPCO2>


Simulation Data Inspector - untitled*
Inspect
Compare
Filter Signals
NAME
LINE
Run 5: ConfiguredVirtualVehicleMode
Signals
<ax>
<ay>
<TPCO2>
<TransGear>
<TransGearCmd>
<az>
<xdot>
<AccelFdbk>
<DecelFdbk>
<GearFdbk>
<EngSpdOut>
<EngTrq>
<FuelCellCurr>
<FuelCellVolt>
<SteerFdbk>
Parameters
Archive (4)
Properties
<TPCO2>
<TPCO2>

```
>> out

out =

  Simulink.SimulationOutput:

                   logsout: [1x1 Simulink.SimulationData.Dataset]
                      tout: [100801x1 double]

        SimulationMetadata: [1x1 Simulink.SimulationMetadata]
              ErrorMessage: [0x0 char]

>>
```

```
>> out.logsout

ans =

Simulink.SimulationData.Dataset 'logsout' with 15 elements

                         Name            BlockPath
                         ______________  ________________________________________

    1  [1x1 Signal]      <AccelFdbk>     ...isualization/DataLogging/Bus Selector
    2  [1x1 Signal]      <DecelFdbk>     ...isualization/DataLogging/Bus Selector
    3  [1x1 Signal]      <EngSpdOut>     ...isualization/DataLogging/Bus Selector
    4  [1x1 Signal]      <EngTrq>        ...isualization/DataLogging/Bus Selector
    5  [1x1 Signal]      <FuelCellCurr>  ...isualization/DataLogging/Bus Selector
    6  [1x1 Signal]      <FuelCellVolt>  ...isualization/DataLogging/Bus Selector
    7  [1x1 Signal]      <GearFdbk>      ...isualization/DataLogging/Bus Selector
    8  [1x1 Signal]      <SteerFdbk>     ...isualization/DataLogging/Bus Selector
    9  [1x1 Signal]      <TPCO2>         ...isualization/DataLogging/Bus Selector
   10  [1x1 Signal]      <TransGear>     ...isualization/DataLogging/Bus Selector
   11  [1x1 Signal]      <TransGearCmd>  ...isualization/DataLogging/Bus Selector
   12  [1x1 Signal]      <ax>            ...isualization/DataLogging/Bus Selector
   13  [1x1 Signal]      <ay>            ...isualization/DataLogging/Bus Selector
   14  [1x1 Signal]      <az>            ...isualization/DataLogging/Bus Selector
   15  [1x1 Signal]      <xdot>          ...isualization/DataLogging/Bus Selector

  - Use braces { } to access, modify, or add elements using index.

>>
```

```
>> out.logsout{9}

ans =

  Simulink.SimulationData.Signal
  Package: Simulink.SimulationData

  Properties:
              Name: '<TPCO2>'
    PropagatedName: ''
         BlockPath: [1×1 Simulink.SimulationData.BlockPath]
          PortType: 'outport'
         PortIndex: 11
            Values: [1×1 timeseries]


  Methods, Superclasses
>>
```

```
>> out.logsout{9}.Values
  timeseries

  Common Properties:
            Name: '<TPCO2>'
            Time: [100801x1 double]
        TimeInfo: [1x1 tsdata.timemetadata]
            Data: [100801x1 double]
        DataInfo: [1x1 tsdata.datametadata]

  More properties, Methods

>>
```

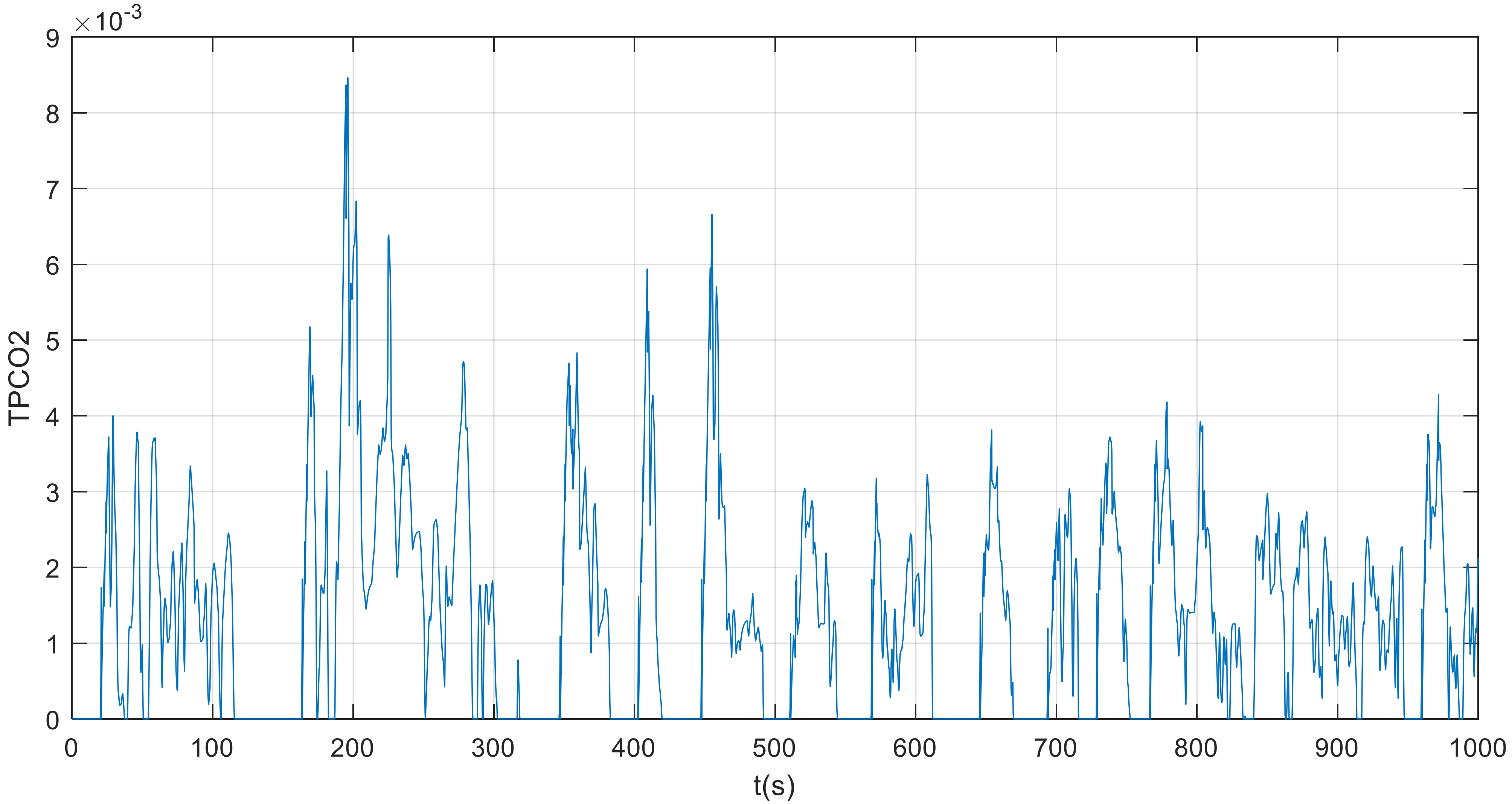
Base de Dados

```
>> Dados = out.logsout{9}.Values.Data;
>> size(Dados)

ans =

      100801           1
```

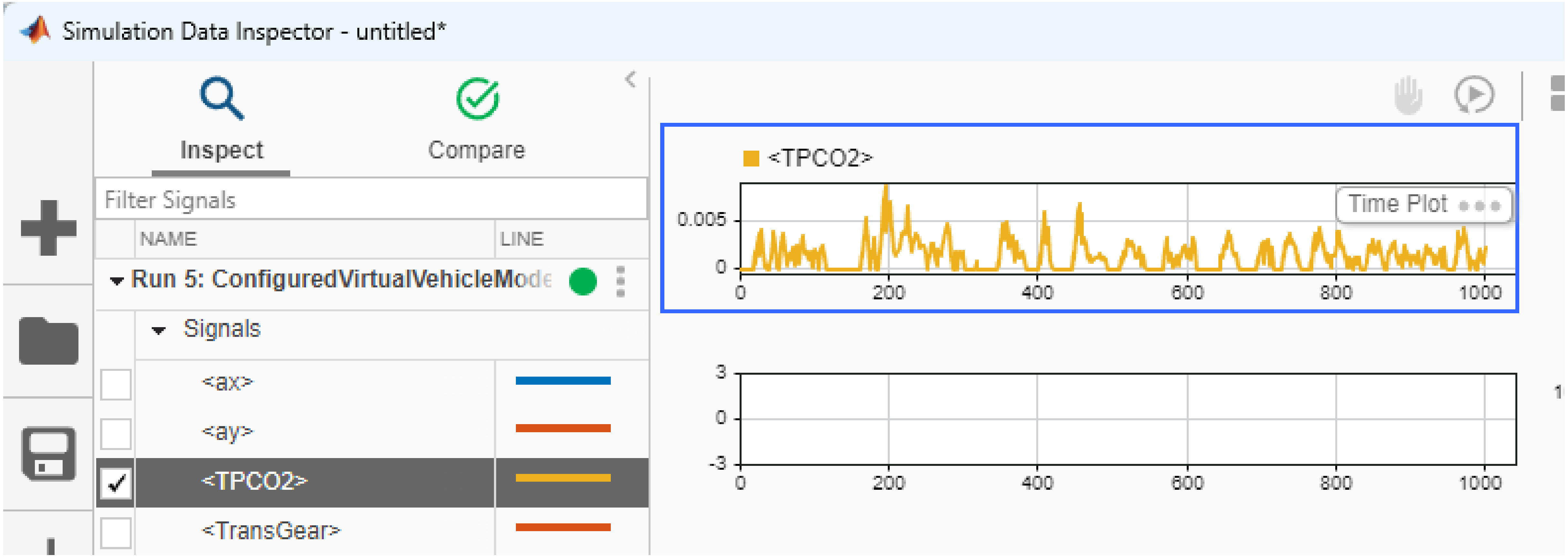
Base de tempo

```
>> size(out.tout)

ans =

      100801
```

```
>> plot(out.tout,out.logsout{9}.Values.Data)
>> grid
>> xlabel('t(s)')
>> ylabel('TPCO2')
```



×10^-3
9
8
7
6
5
4
3
2
1
0
TPCO2
0
100
200
300
400
500
600
700
800
900
1000
t(s)



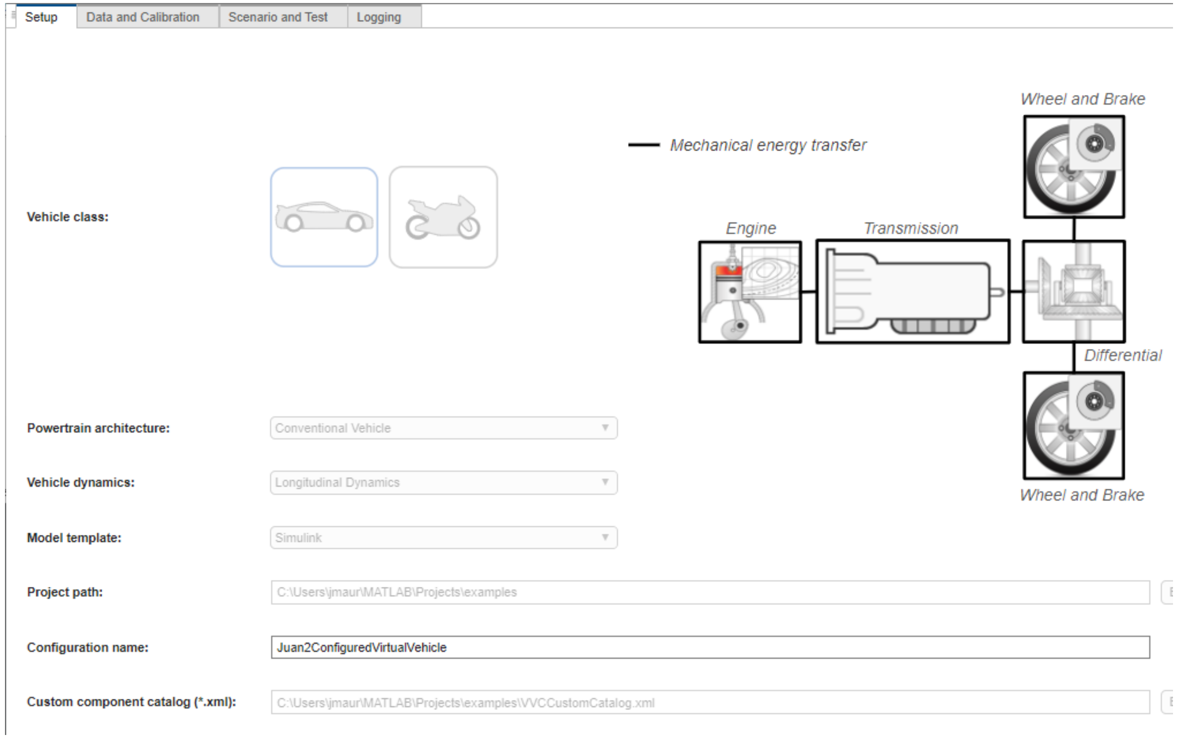
Simulation Data Inspector - untitled*
Inspect
Compare
Filter Signals
NAME
LINE
Run 5: ConfiguredVirtualVehicleMode
Signals
<ax>
<ay>
<TPCO2>
<TransGear>
<TPCO2>
Time Plot
0.005
0
0
200
400
600
800
1000
3
0
-3
0
200
400
600
800
1000

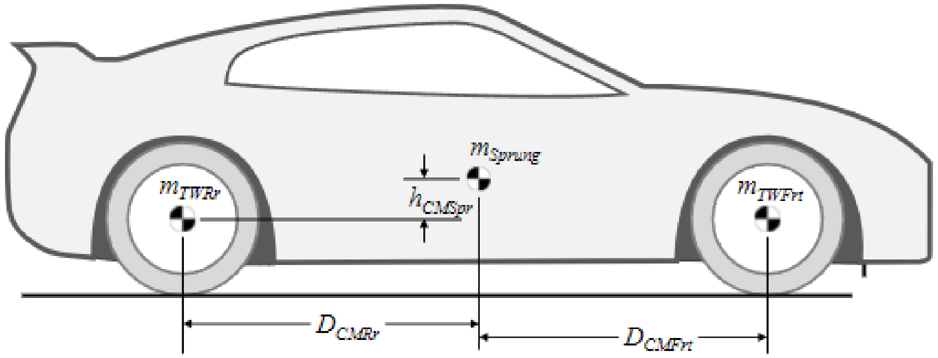
Resultados 1

Resultados

Body and Frame: Vehicle Body 3DOF Longitudinal

https://www.anl.gov/taps/autonomie-vehicle-system-simulation-tool





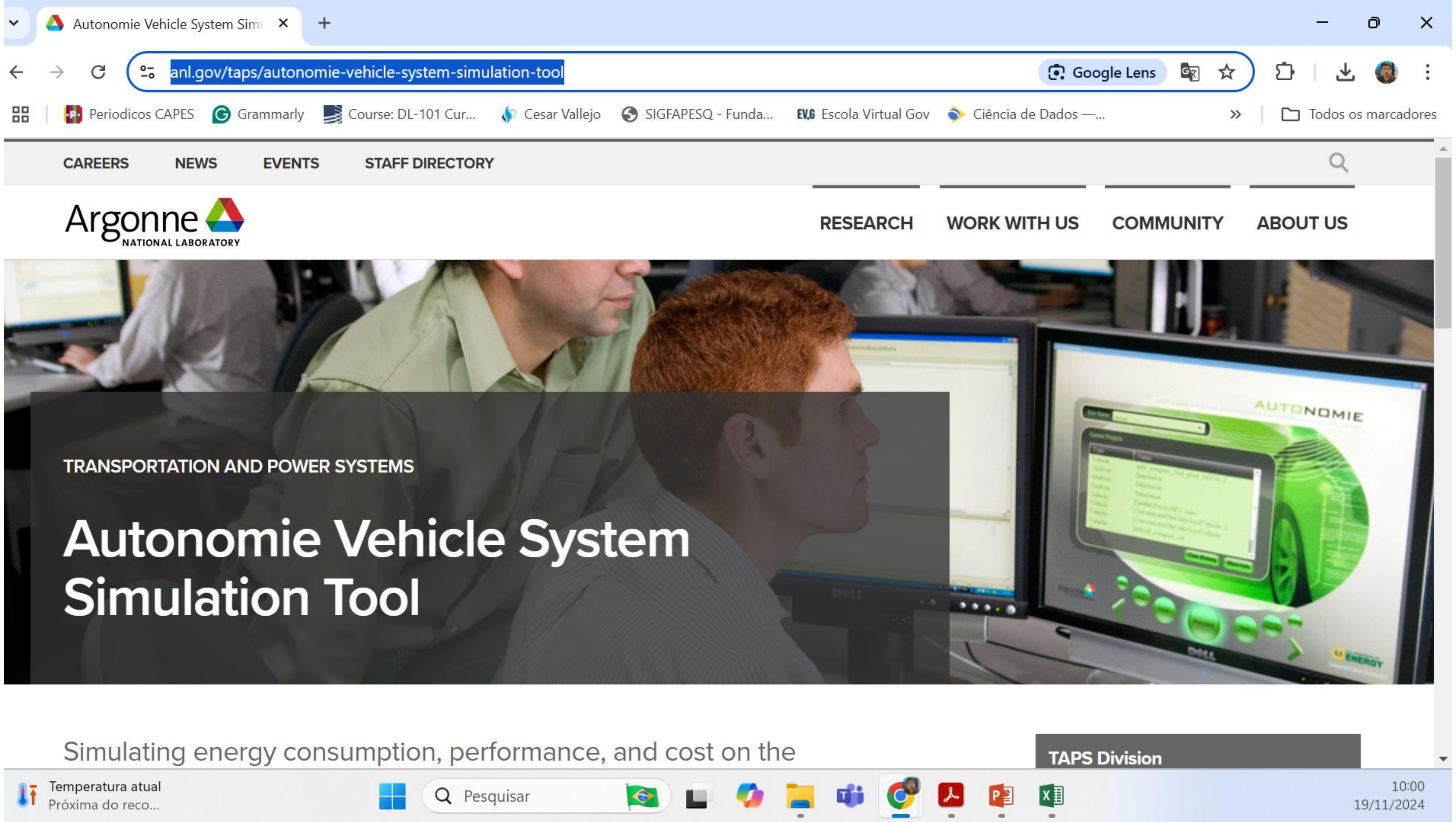
Symbol	Parameter Name
m_{Sprung}	PlntVehMass
D_{CMFrt}	PlntVehDstCGFrntAxl
D_{CMRr}	PlntVehDstCGRearAxl
h_{CMSpr}	PlntVehCGHgtAxl

Parameters

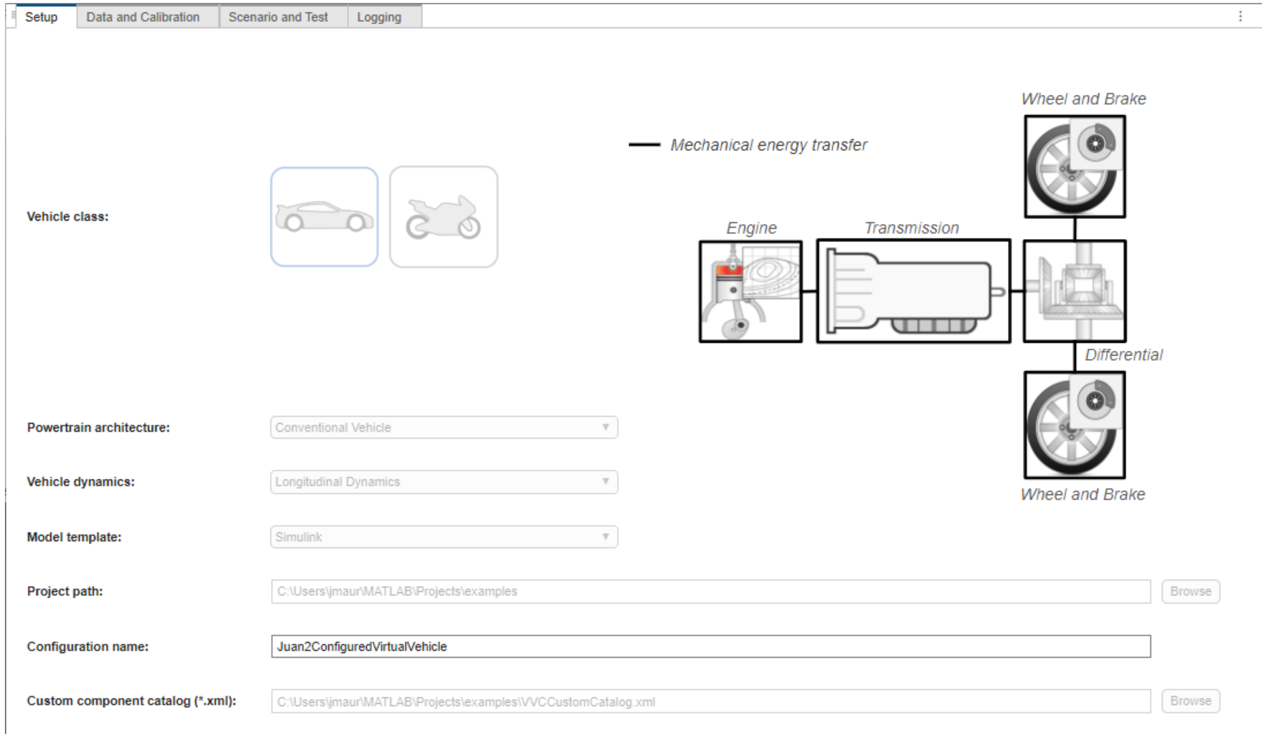
Reset Values   Save as New Component

	Parameter Name	Description	Units	Value
1	PlntVehMass	Vehicle sprung mass with body fully equipped	kg	1623
2	PlntVehDstCGFrntAxl	Longitudinal distance from sprung mass CM to front axle	m	1.09
3	PlntVehDstCGRearAxl	Longitudinal distance from sprung mass CM to rear axle	m	1.7
4	PlntVehCGHgtAxl	Vertical distance from axle plane to sprung mass CM	m	0.3
5	PlntVehPitchMomentInertia	Moment of inertia about the pitch axis	kg*m^2	1922.7
6	PlntVehAeroFrntArea	Frontal area of vehicle	m^2	2.27
7	PlntVehAeroDragCff	Aerodynamic drag coefficient	[]	0.23
8	PlntVehAeroLiftCff	Aerodynamic lift coefficient	[]	0.1
9	PlntVehAeroPitchCff	Aerodynamic pitch moment coefficient, reference length: ...	[]	0.1

https://www.mathworks.com/help/releases/R2024b/autoblks/ref/virtualvehiclecomposer-app.html

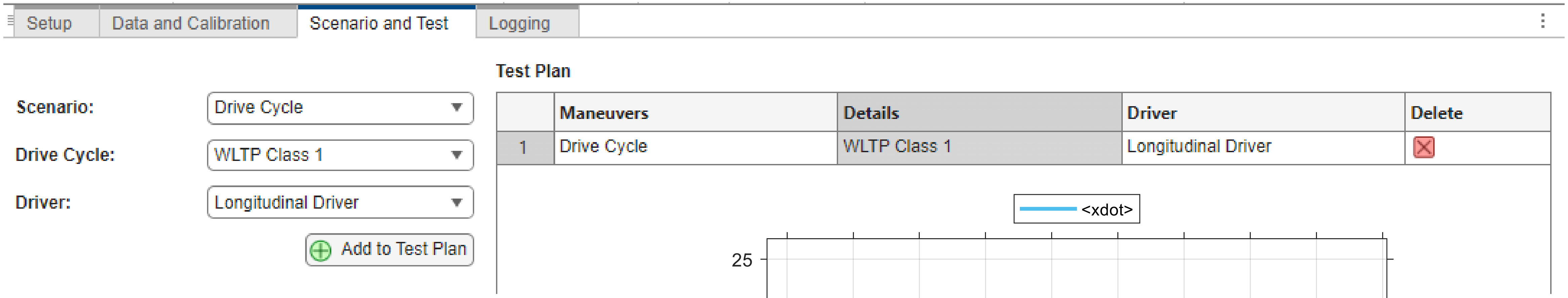

Autonomie Vehicle System Sim
anl.gov/taps/autonomie-vehicle-system-simulation-tool
Google Lens
Periodicos CAPES
Grammarly
Course: DL-101 Cur...
Cesar Vallejo
SIGFAPESQ - Funda...
Escola Virtual Gov
Ciência de Dados —...
Todos os marcadores
CAREERS
NEWS
EVENTS
STAFF DIRECTORY
Argonne
NATIONAL LABORATORY
RESEARCH
WORK WITH US
COMMUNITY
ABOUT US
TRANSPORTATION AND POWER SYSTEMS
Autonomie Vehicle System Simulation Tool
AUTONOMIE
Simulating energy consumption, performance, and cost on the
TAPS Division
Temperatura atual
Próxima do reco...
Pesquisar
10:00
19/11/2024

Resultados

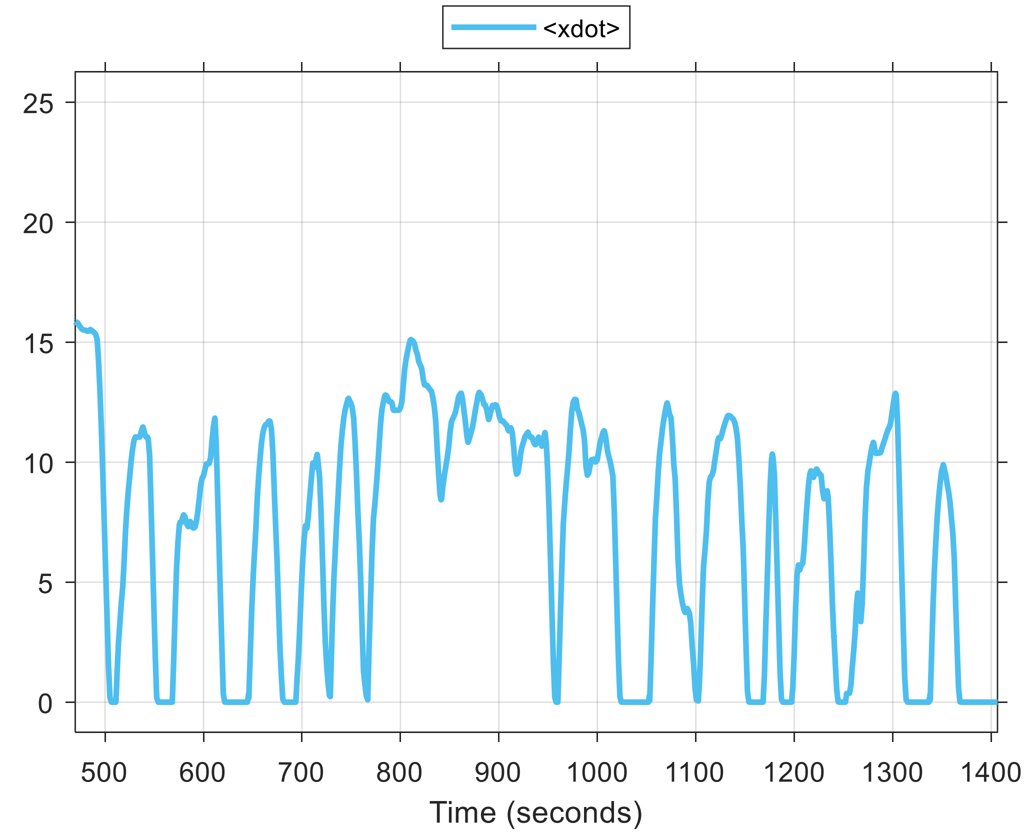


		Name
1	[1x1 Signal]	<AccelFdbk>
2	[1x1 Signal]	<DecelFdbk>
3	[1x1 Signal]	<EngSpdOut>
4	[1x1 Signal]	<EngTrq>
5	[1x1 Signal]	<FuelCellCurr>
6	[1x1 Signal]	<FuelCellVolt>
7	[1x1 Signal]	<GearFdbk>
8	[1x1 Signal]	<SteerFdbk>
9	[1x1 Signal]	<TPCO2>
10	[1x1 Signal]	<TransGear>
11	[1x1 Signal]	<TransGearCmd>
12	[1x1 Signal]	<ax>
13	[1x1 Signal]	<ay>
14	[1x1 Signal]	<az>
15	[1x1 Signal]	<xdot>

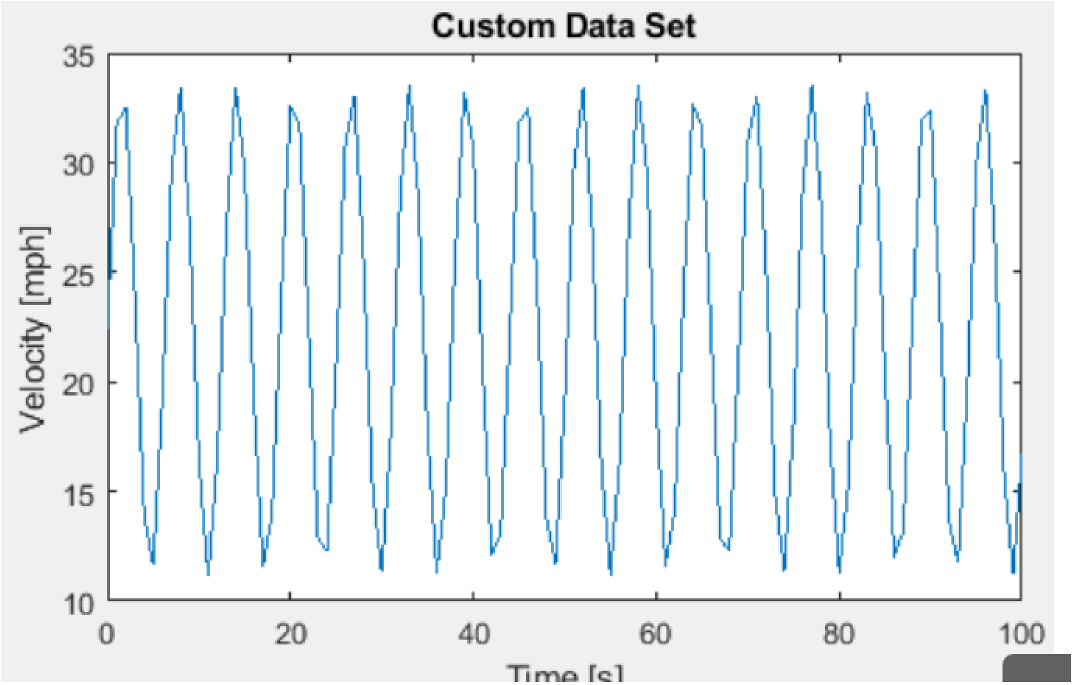
Configuração do Teste




```
>> plot(out.tout,out.logsout{15}.Values.Data)
>> grid
```



Workspace Variable	Source Velocity Units	Output Velocity Units	Drive Cycle Plot

Structure without a gear shift schedule, with **From workspace** set to `myCycleS`.
`t = 0:1:100;`
`xdot = 5.*sin(t)+10;`
`myCycleS.time = t';`
`myCycleS.signals.values = xdot';`
	`m/s`	`mph`	

https://www.mathworks.com/help/vdynblks/ref/drivecyclesource.html

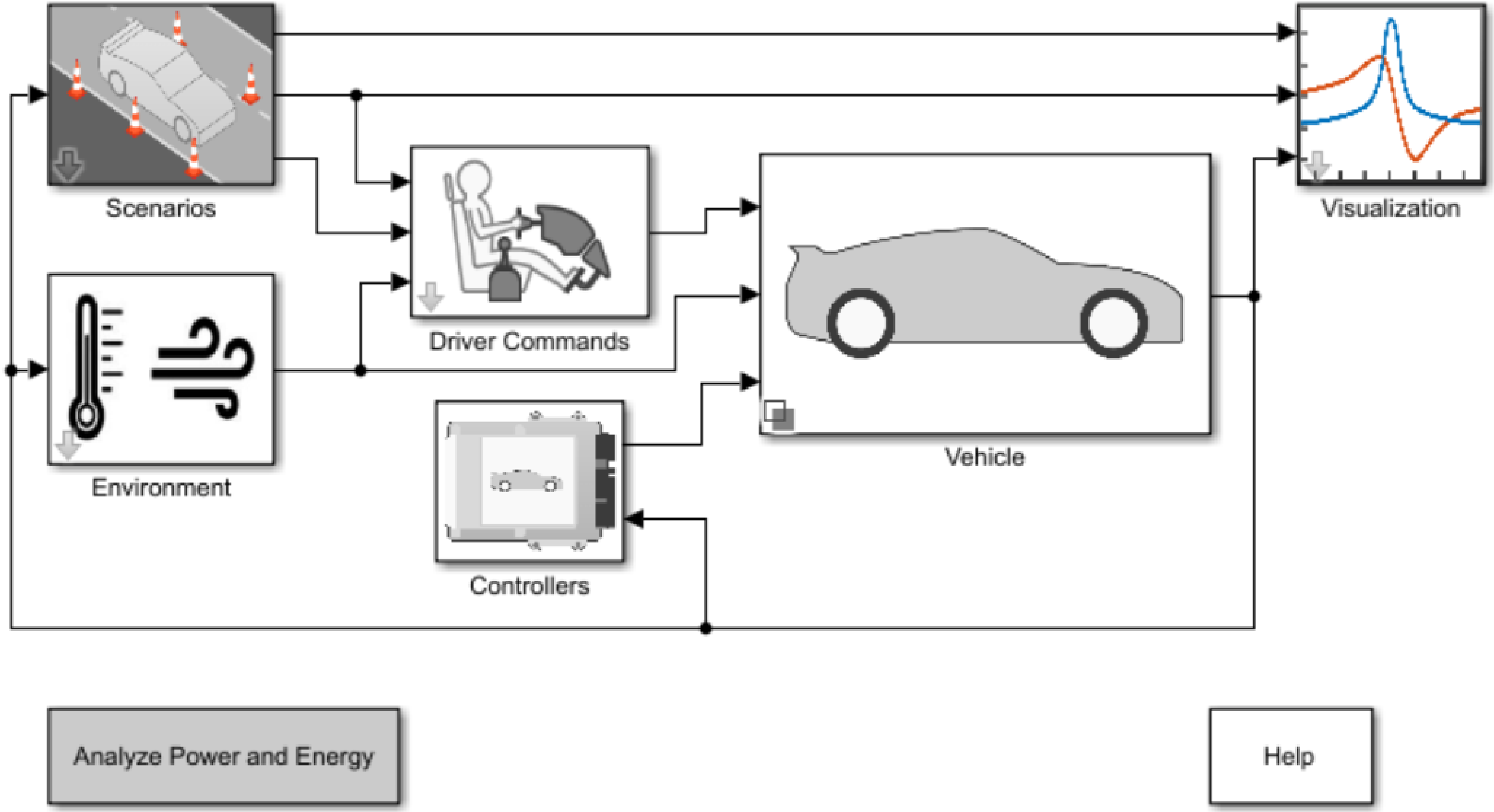
Configuração do Teste

Scene Parameters | Driver Parameters

Parameter Name	Description	Unit	Value
ScnSimTime	Simulation time	s	1800
ScnLongVelUnit	Longitudinal velocity units	[]	mph
PlntVehInitVertPos	Vehicle initial vertical position	m	0
DriverPreviewDist	Preview distance	m	4
DriverTimeConst	Time constant	s	0.3
PlntVehInitLatPos	Initial lateral position	m	0

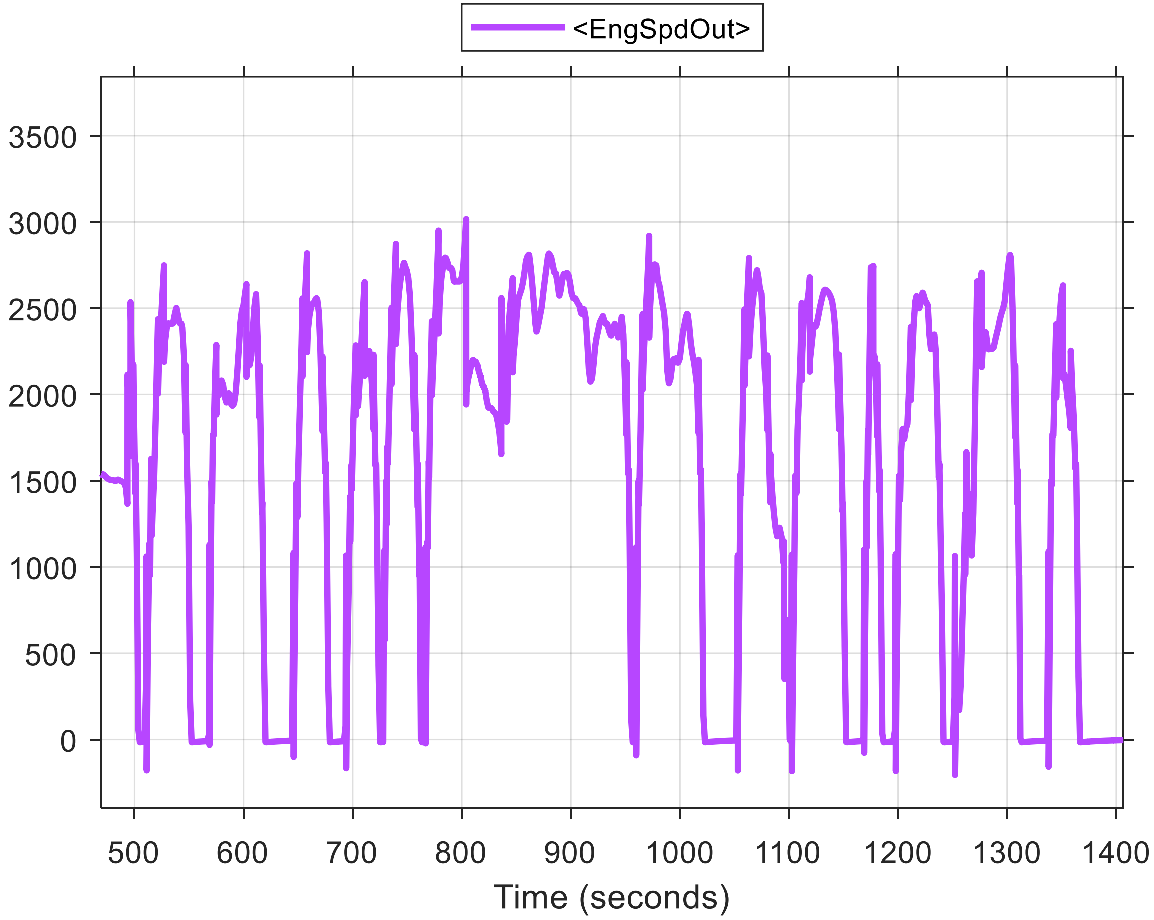
Scene Parameters | Driver Parameters

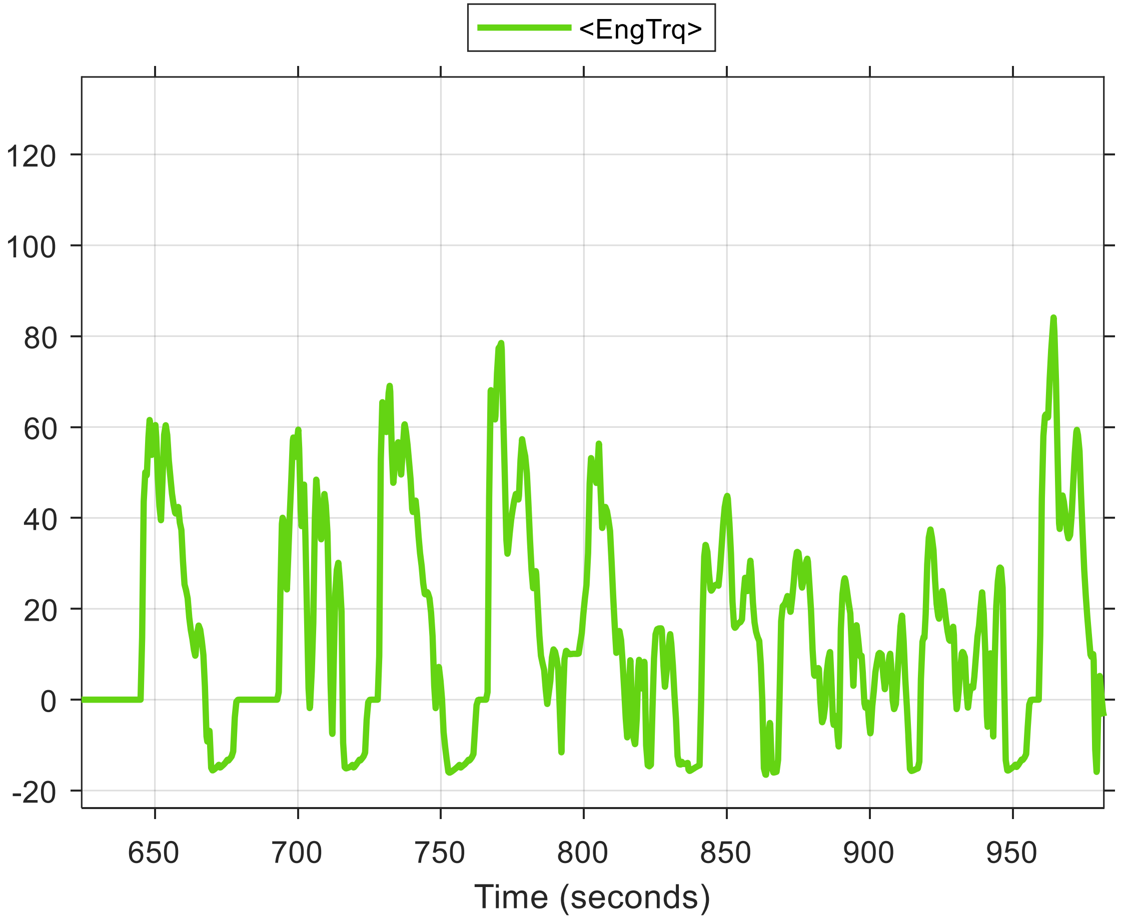
Parameter Name	Description	Unit	Value
DriverAeroRes	Aerodynamic drag coefficient	[]	0.5
DriverDrivelineRes	Rolling/driveline resistance coefficient for driver mod...	N*s/m	2.5
DriverRollRes	Rolling resistance for driver model, constant compon...	N	200
DriverTractiveForce	Tractive force	N	14000

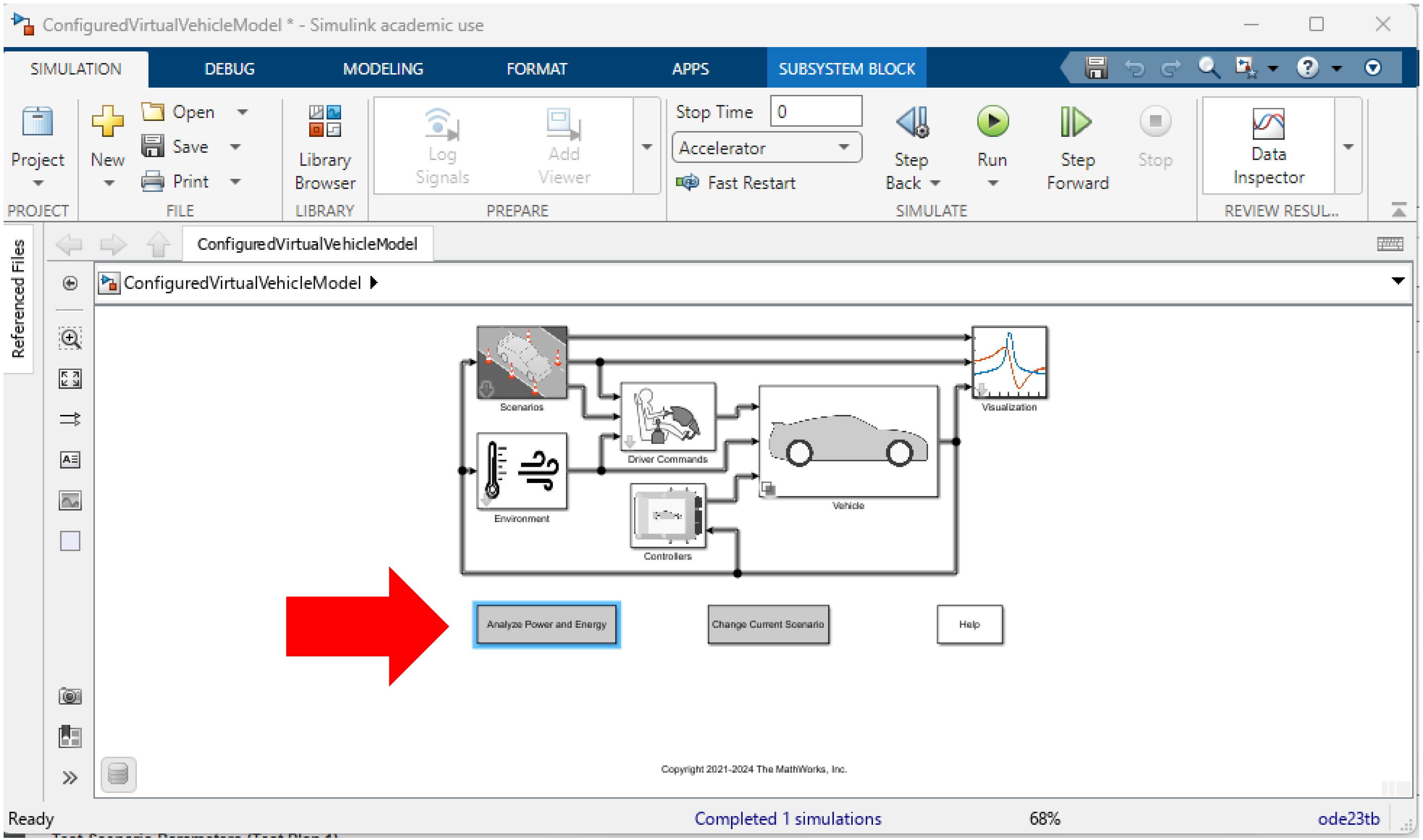

Scenarios
Visualization
Driver Commands
Vehicle
Environment
Controllers
Analyze Power and Energy
Help

This table describes the blocks and subsystems in the reference application.

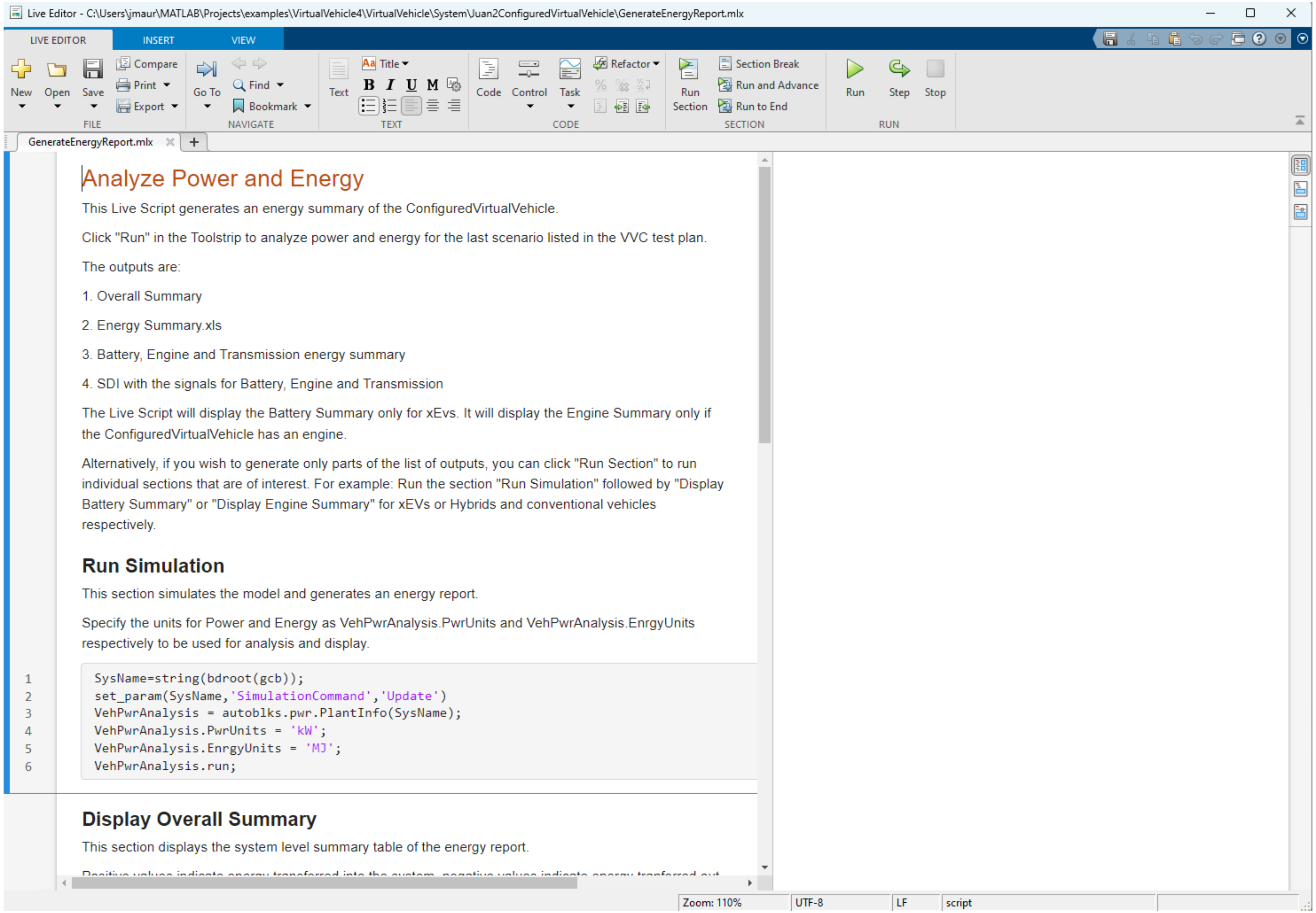
Reference Application Element	Description
Analyze Power and Energy	Double-click **Analyze Power and Energy** to open a live script. Run the script to evaluate and report power and energy consumption at the component- and system-level. For more information about the live script, see Analyze Power and Energy.
Scenarios	Implements the Drive Cycle Source block to generate a FTP75 (2474 seconds) drive cycle.
Environment	Creates environment variables, including road grade, wind velocity, and ambient temperature and pressure.
Driver Commands	Implements the Longitudinal Driver to generate normalized acceleration and braking commands.
Controllers	Implements a powertrain control module (PCM) containing a transmission control unit (TCU) and engine control unit (ECU).
Vehicle	Implements a passenger car that contains transmission drivetrain and engine plant model subsystems.
Visualization	Displays vehicle-level performance, fuel economy, and emission results that are useful for powertrain matching and component selection analysis.


<EngSpdOut>
3500
3000
2500
2000
1500
1000
500
0
500
600
700
800
900
1000
1100
1200
1300
1400
Time (seconds)


<EngTrq>
120
100
80
60
40
20
0
-20
650
700
750
800
850
900
950
Time (seconds)



ConfiguredVirtualVehicleModel * - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
APPS
SUBSYSTEM BLOCK
Project
New
Open
Save
Print
Library Browser
Log Signals
Add Viewer
Stop Time
0
Accelerator
Fast Restart
Step Back
Run
Step Forward
Stop
Data Inspector
PROJECT
FILE
LIBRARY
PREPARE
SIMULATE
REVIEW RESUL...
Referenced Files
ConfiguredVirtualVehicleModel
Scenarios
Visualization
Driver Commands
Vehicle
Environment
Controllers
Analyze Power and Energy
Change Current Scenario
Help
Copyright 2021-2024 The MathWorks, Inc.
Ready
Completed 1 simulations
68%
ode23tb



Analyze Power and Energy

This Live Script generates an energy summary of the ConfiguredVirtualVehicle.

Click "Run" in the Toolstrip to analyze power and energy for the last scenario listed in the VVC test plan.

The outputs are:

1. Overall Summary

2. Energy Summary.xls

3. Battery, Engine and Transmission energy summary

4. SDI with the signals for Battery, Engine and Transmission

The Live Script will display the Battery Summary only for xEvs. It will display the Engine Summary only if the ConfiguredVirtualVehicle has an engine.

Alternatively, if you wish to generate only parts of the list of outputs, you can click "Run Section" to run individual sections that are of interest. For example: Run the section "Run Simulation" followed by "Display Battery Summary" or "Display Engine Summary" for xEVs or Hybrids and conventional vehicles respectively.

Run Simulation

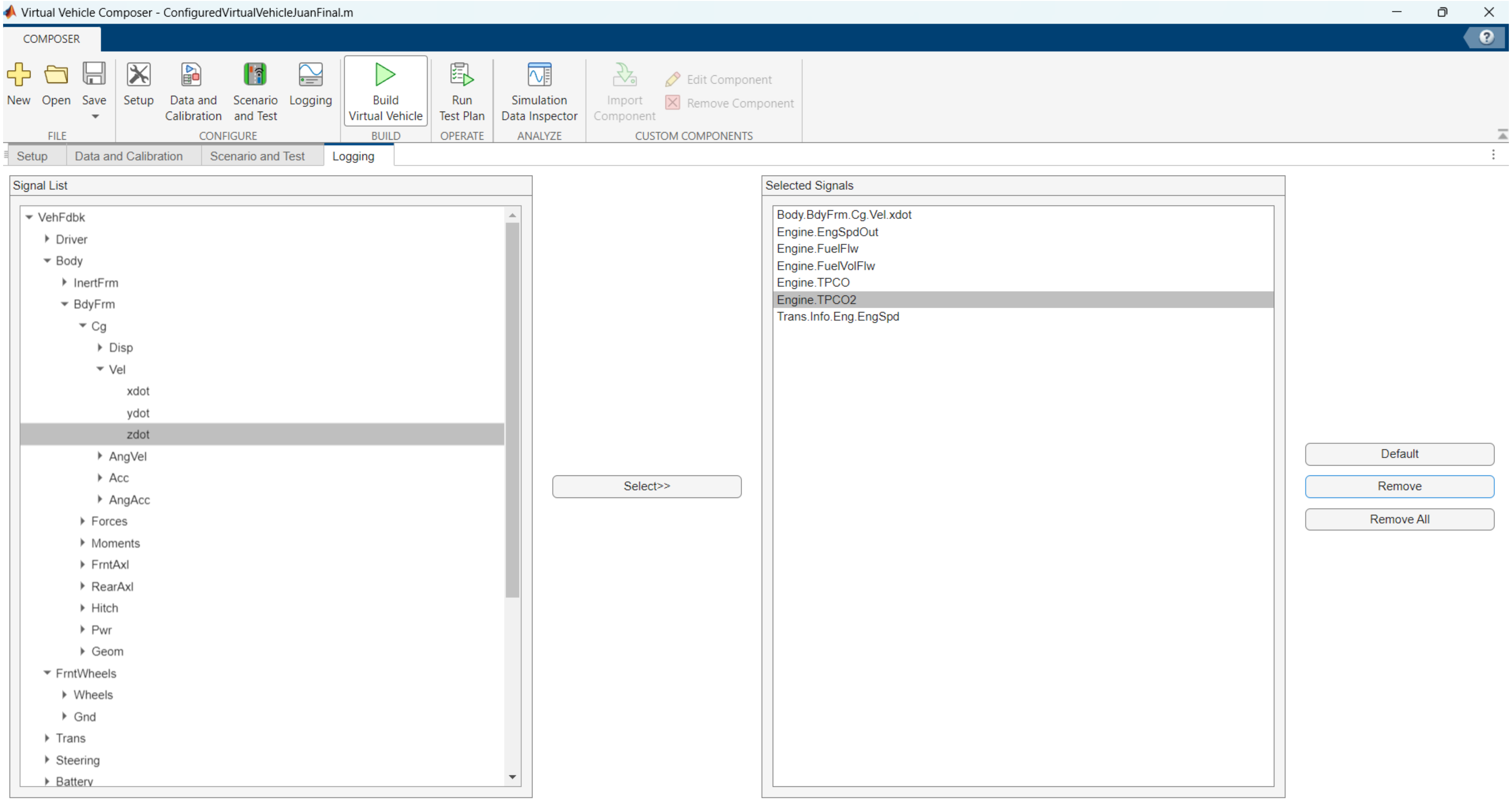
This section simulates the model and generates an energy report.

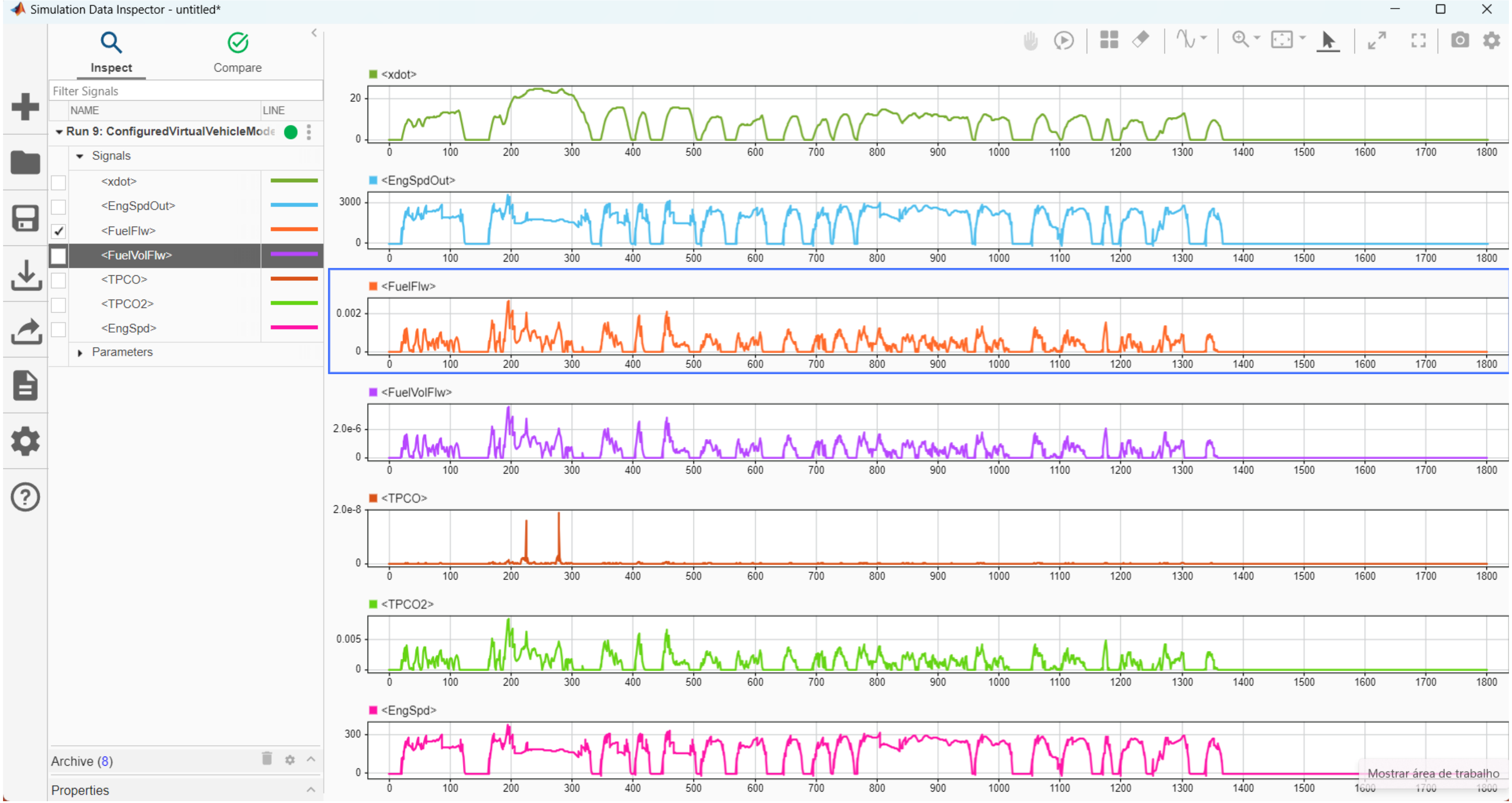
Specify the units for Power and Energy as VehPwrAnalysis.PwrUnits and VehPwrAnalysis.EnrgyUnits respectively to be used for analysis and display.

```
SysName=string(bdroot(gcb));
set_param(SysName,'SimulationCommand','Update')
VehPwrAnalysis = autoblks.pwr.PlantInfo(SysName);
VehPwrAnalysis.PwrUnits = 'kW';
VehPwrAnalysis.EnrgyUnits = 'MJ';
VehPwrAnalysis.run;
```

Display Overall Summary

This section displays the system level summary table of the energy report.


Virtual Vehicle Composer - ConfiguredVirtualVehicleJuanFinal.m
COMPOSER
New
Open
Save
FILE
Setup
Data and Calibration
Scenario and Test
Logging
CONFIGURE
Build Virtual Vehicle
BUILD
Run Test Plan
OPERATE
Simulation Data Inspector
ANALYZE
Import Component
Edit Component
Remove Component
CUSTOM COMPONENTS
Setup
Data and Calibration
Scenario and Test
Logging
Signal List
VehFdbk
Driver
Body
InertFrm
BdyFrm
Cg
Disp
Vel
xdot
ydot
zdot
AngVel
Acc
AngAcc
Forces
Moments
FrntAxl
RearAxl
Hitch
Pwr
Geom
FrntWheels
Wheels
Gnd
Trans
Steering
Battery
Select>>
Selected Signals
Body.BdyFrm.Cg.Vel.xdot
Engine.EngSpdOut
Engine.FuelFlw
Engine.FuelVolFlw
Engine.TPCO
Engine.TPCO2
Trans.Info.Eng.EngSpd
Default
Remove
Remove All


Simulation Data Inspector - untitled*
Inspect
Compare
Filter Signals
NAME
LINE
Run 9: ConfiguredVirtualVehicleMode
Signals
<xdot>
<EngSpdOut>
<FuelFlw>
<FuelVolFlw>
<TPCO>
<TPCO2>
<EngSpd>
Parameters
Archive (8)
Properties
<xdot>
20
0
<EngSpdOut>
3000
0
<FuelFlw>
0.002
0
<FuelVolFlw>
2.0e-6
0
<TPCO>
2.0e-8
0
<TPCO2>
0.005
0
<EngSpd>
300
0
0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800
Mostrar área de trabalho

The TPCO2 engine model refers to a transient powertrain cycle simulation that typically represents an internal combustion engine or a hybrid engine model in Simulink, specifically for the purpose of optimizing emissions and fuel efficiency (often targeting CO2 reduction, hence the "TPCO2"). In MATLAB Simulink, you can set up a TPCO2 engine model to evaluate powertrain performance and emissions through various drive cycles.

Steps to Build a TPCO2 Powertrain Model in Simulink:

1.Engine Subsystem:
1. Use the *Internal Combustion Engine* block from Powertrain Blockset or the *Engine* block in Simulink if modeling a combustion engine.
2. Configure parameters such as fuel type, displacement, cylinder number, and engine efficiency.
3. For CO2 emissions, add an emissions calculation model to measure output based on fuel combustion.

2.Transmission Subsystem:
1. Use the *Transmission* block from the Powertrain Blockset to model different types of transmissions (manual, automatic, or CVT).
2. Connect the transmission block to the engine and configure gear ratios, shift logic, and power loss characteristics.

3.Driveline and Wheels:
1. Add driveline components like the *Differential*, *Driveshaft*, and *Tire* models to connect the transmission to the vehicle's wheels.
2. Tire models, like those from the *Vehicle Dynamics Blockset*, help simulate real-world road interaction, including rolling resistance and grip.

4.CO2 Emissions and Fuel Consumption Calculation:
1. For CO2 emissions, set up a subsystem to calculate emissions based on fuel flow rate and engine efficiency.
2. Integrate real-time monitoring of fuel usage to calculate grams of CO2 per km or per cycle.

5.Vehicle Model and Power Demand:
1. Create a *Vehicle Dynamics* block that includes weight, drag coefficient, and other resistance factors.
2. Include an input for throttle demand based on driving cycle (e.g., FTP, WLTP) to test the engine's response under various loads.

6.Controller and Drive Cycle:
1. Use a *PID Controller* to simulate the engine management system (EMS) which adjusts throttle, fuel injection, and other variables.
2. Load standardized drive cycles (such as WLTP or FTP-75) to simulate urban, highway, or mixed driving conditions.

7.Simulation and Analysis:
1. Run the simulation for a predefined time or cycle to analyze power output, fuel consumption, and CO2 emissions.
2. Plot outputs for torque, speed, emissions, and fuel economy to evaluate performance under different scenarios.

Using Powertrain Blockset in Simulink

The **Powertrain Blockset** in MATLAB Simulink is specifically useful here, as it includes pre-built models and subsystems for internal combustion engines, transmission, vehicle dynamics, and emissions estimation. It simplifies setting up and running complex powertrain simulations, including TPCO2 and other emissions-focused cycles.

O modelo de motor TPCO2 refere-se a uma simulação de ciclo de trem de força (powertrain) em regime transitório, que geralmente representa um motor de combustão interna ou um modelo de motor híbrido no Simulink, visando a otimização de emissões e eficiência de combustível (frequentemente com foco na redução de CO2, daí o nome "TPCO2"). No MATLAB Simulink, é possível configurar um modelo de motor TPCO2 para avaliar o desempenho do trem de força e as emissões em diferentes ciclos de condução.

Passos para Construir um Modelo de Trem de Força TPCO2 no Simulink:

1. **Subsistema do Motor**:
    1. Use o bloco *Internal Combustion Engine* do Powertrain Blockset ou o bloco *Engine* no Simulink para modelar um motor de combustão.
    2. Configure parâmetros como tipo de combustível, deslocamento, número de cilindros e eficiência do motor.
    3. Para calcular as emissões de CO2, adicione um modelo de cálculo de emissões para medir a saída com base na combustão de combustível.
2. **Subsistema de Transmissão**:
    1. Use o bloco *Transmission* do Powertrain Blockset para modelar diferentes tipos de transmissões (manual, automática ou CVT).
    2. Conecte o bloco de transmissão ao motor e configure as relações de engrenagem, lógica de mudança de marcha e características de perda de potência.
3. **Linha de Transmissão e Rodas**:
    1. Adicione componentes da linha de transmissão, como os modelos de *Differential*, *Driveshaft* e *Tire*, para conectar a transmissão às rodas do veículo.
    2. Os modelos de pneu, como aqueles no *Vehicle Dynamics Blockset*, ajudam a simular a interação real com a estrada, incluindo resistência ao rolamento e aderência.
4. **Cálculo de Emissões de CO2 e Consumo de Combustível**:
    1. Para as emissões de CO2, configure um subsistema para calcular emissões com base na taxa de consumo de combustível e na eficiência do motor.
    2. Integre um monitoramento em tempo real do consumo de combustível para calcular as emissões de CO2 em gramas por km ou por ciclo.
5. **Modelo do Veículo e Demanda de Potência**:
    1. Crie um bloco *Vehicle Dynamics* que inclua peso, coeficiente de arrasto e outros fatores de resistência.
    2. Inclua uma entrada de demanda de aceleração baseada no ciclo de condução (como FTP ou WLTP) para testar a resposta do motor sob diferentes cargas.
6. **Controlador e Ciclo de Condução**:
    1. Use um *Controlador PID* para simular o sistema de gerenciamento do motor (EMS), que ajusta a aceleração, injeção de combustível e outras variáveis.
    2. Carregue ciclos de condução padronizados (como WLTP ou FTP-75) para simular condições de condução urbana, rodoviária ou mista.
7. **Simulação e Análise**:
    1. Execute a simulação por um tempo ou ciclo predefinido para analisar a potência de saída, consumo de combustível e emissões de CO2.
    2. Trace gráficos de torque, velocidade, emissões e economia de combustível para avaliar o desempenho em diferentes cenários.

Usando o Powertrain Blockset no Simulink

O **Powertrain Blockset** no MATLAB Simulink é especialmente útil aqui, pois inclui modelos e subsistemas pré-construídos para motores de combustão interna, transmissão, dinâmica veicular e estimativa de emissões. Ele simplifica a configuração e execução de simulações complexas de trem de força, incluindo o ciclo TPCO2 e outros focados em emissões.

- **CO2 Emissions**: CO2 emissions based on fuel consumption.
- **Engine Speed**: Engine rotation speed (in RPM)

Como calcular o consumo e a eficiência?

Command Window

New to MATLAB? See resources for Getting Started.

```
>> who

Your variables are:

ConfigInfos     ans             data
M               dadosJuanFinal  out

>> out

out =

  Simulink.SimulationOutput:

                logsout: [1x1 Simulink.SimulationData.Dataset]
                   tout: [181162x1 double]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]
```

```
>> out.logsout

ans =

Simulink.SimulationData.Dataset 'logsout' with 7 elements

                          Name          BlockPath
                          ____________  ________________________________________
    1  [1x1 Signal]       <EngSpd>      ...isualization/DataLogging/Bus Selector
    2  [1x1 Signal]       <EngSpdOut>   ...isualization/DataLogging/Bus Selector
    3  [1x1 Signal]       <FuelFlw>     ...isualization/DataLogging/Bus Selector
    4  [1x1 Signal]       <FuelVolFlw>  ...isualization/DataLogging/Bus Selector
    5  [1x1 Signal]       <TPCO2>       ...isualization/DataLogging/Bus Selector
    6  [1x1 Signal]       <TPCO>        ...isualization/DataLogging/Bus Selector
    7  [1x1 Signal]       <xdot>        ...isualization/DataLogging/Bus Selector

  - Use braces { } to access, modify, or add elements using index.
```

1

```
>> out.logsout{1}

ans =

  Simulink.SimulationData.Signal
  Package: Simulink.SimulationData

  Properties:
              Name: '<FuelFlw>'
    PropagatedName: ''
         BlockPath: [1×1 Simulink.SimulationData.BlockPath]
          PortType: 'outport'
         PortIndex: 1
            Values: [1×1 timeseries]
```

2

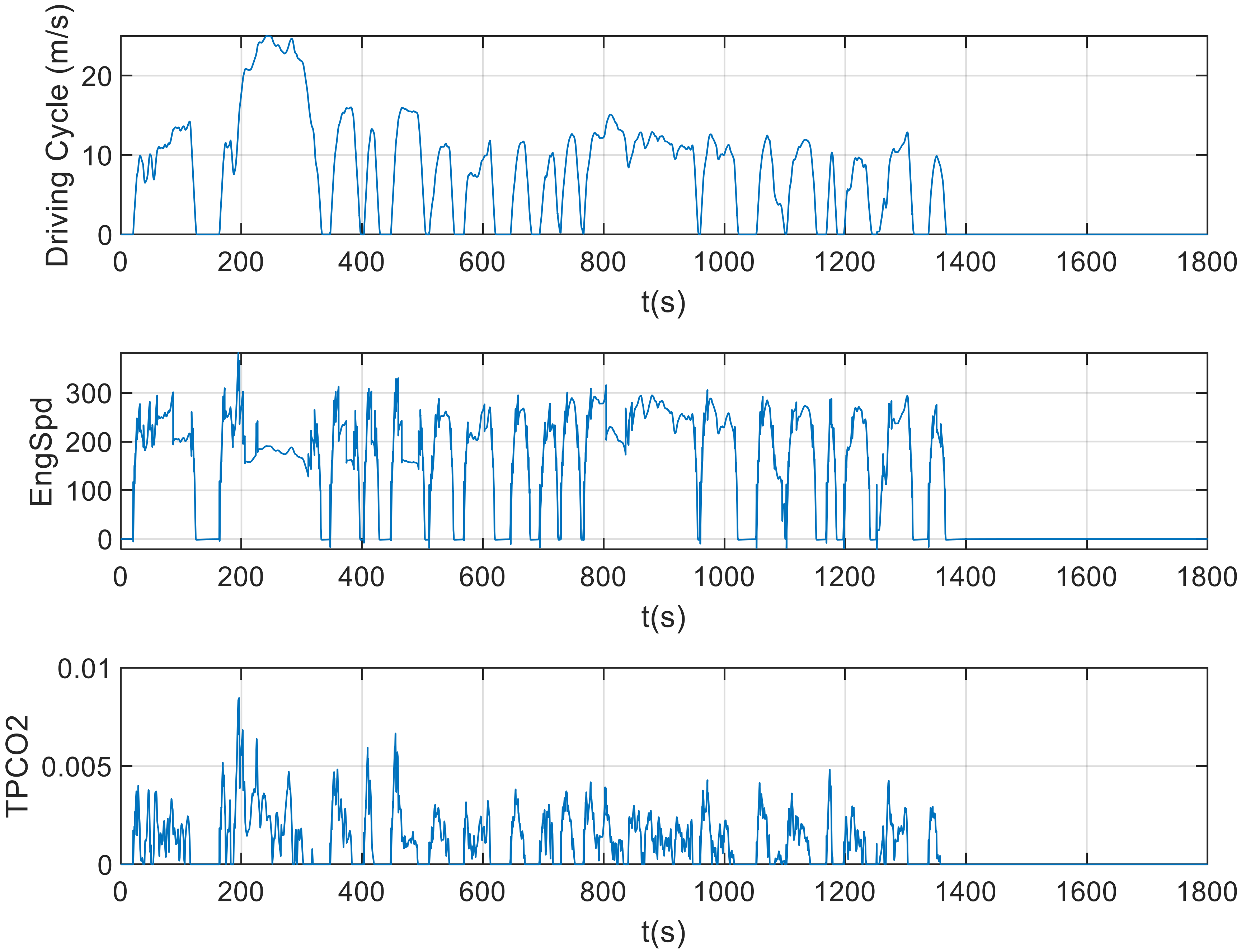
```
>> out.logsout{1}.Values
  timeseries

  Common Properties:
            Name: '<FuelFlw>'
            Time: [181162x1 double]
        TimeInfo: [1x1 tsdata.timemetadata]
            Data: [181162x1 double]
        DataInfo: [1x1 tsdata.datametadata]

  More properties, Methods
```

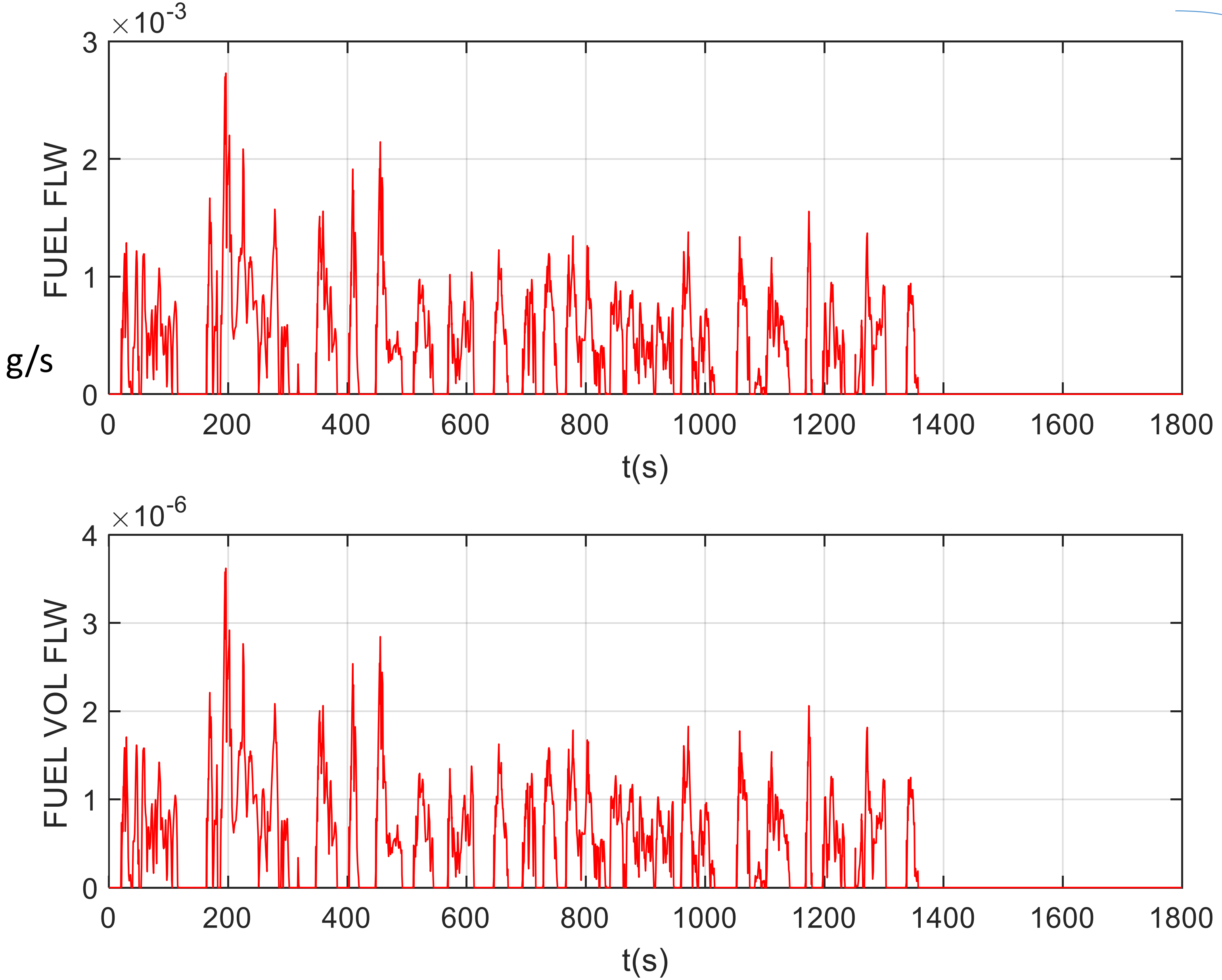
3

```
>> out.logsout{1}.Values.Data
```

```
clc
close all
%save dadosJuanFinal
%load dadosJuanFinal.mat
tempo = out.tout;
engspd = out.logsout{1}.Values.Data;
engspdout = out.logsout{2}.Values.Data;
fuelflw = out.logsout{3}.Values.Data;
fuelvolflw = out.logsout{4}.Values.Data;
tpco2 = out.logsout{5}.Values.Data;
tpco = out.logsout{6}.Values.Data;
xdot = out.logsout{7}.Values.Data;

figure
subplot(3,1,1)
plot(tempo,xdot)
xlabel('t(s)'),ylabel('Driving Cycle (m/s)'),grid

subplot(3,1,2)
plot(tempo,engspd)
xlabel('t(s)'),ylabel('EngSpd'),grid

subplot(3,1,3)
plot(tempo,tpco2)
xlabel('t(s)'),ylabel('TPCO2'),grid
```

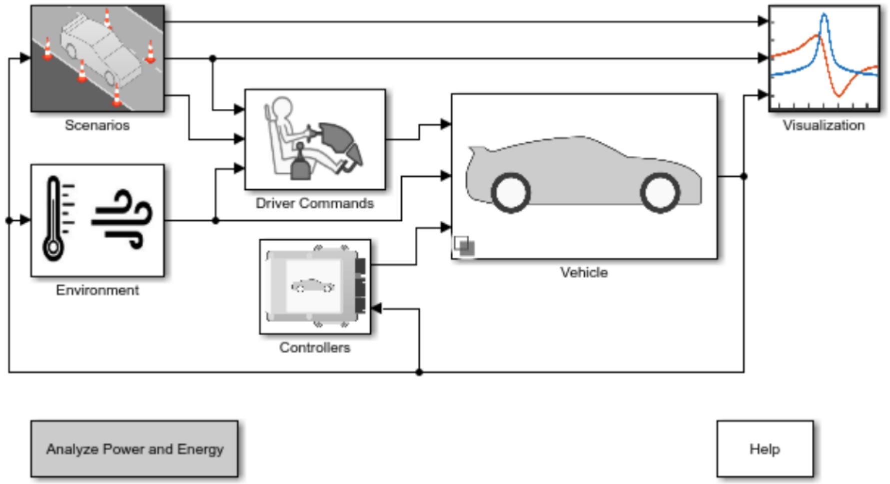
```
%Consumo de combustivel
figure
subplot(2,1,1)
plot(tempo,fuelflw,'r')
xlabel('t(s)'),ylabel('FUEL FLW'),grid


subplot(2,1,2)
plot(tempo,fuelvolflw,'r')
xlabel('t(s)'),ylabel('FUEL VOL FLW'),grid
```


Driving Cycle (m/s)
20
10
0
0
200
400
600
800
1000
1200
1400
1600
1800
t(s)
EngSpd
300
200
100
0
t(s)
TPCO2
0.01
0.005
0
t(s)



Consumo em L?

9,8 kWh/L



Prepare the Conventional Vehicle Reference Application for Simulation

Name the Drive Cycle Source block and Visualization subsystem.

```
model = 'ConfiguredConventionalVirtualVehicle';
dcs = [model, '/Scenarios/Reference Generator/Drive Cycle/Drive Cycle Source'];
vis_sys = [model, '/Visualization'];
```

In the Visualization subsystem, log the emissions signal data.

```
pt_set_logging([vis_sys, '/Performance Calculations'], 'US MPG', 'Fuel Economy [mpg]', 'both');
pt_set_logging([vis_sys, '/Emission Calculations'], 'TP HC Mass (g/mi)', 'HC [g/mi]', 'both');
```

```
pt_set_logging([vis_sys, '/Emission Calculations'], 'TP CO Mass (g/mi)', 'CO [g/mi]', 'both');
pt_set_logging([vis_sys, '/Emission Calculations'], 'TP NOx Mass (g/mi)', 'NOx [g/mi]', 'both');
pt_set_logging([vis_sys, '/Emission Calculations'], 'TP CO2 Mass (g/km)', 'CO2 [g/km]', 'both');
```

Run City Drive Cycle Simulation

Configure the Drive Cycle Source block to run the city drive cycle (FTP75).

```
set_param(dcs,'cycleVar','FTP75');
```

Run a simulation of the city drive cycle. Plot the results.

```
tfinal = get_param(dcs, 'tfinal');
tf = extractBefore(tfinal ,' ');
save_system(model);
simout1 = sim(model,'ReturnWorkspaceOutputs','on', 'StopTime', tf);

logsout1 = simout1.get('logsout');
FECity = logsout1.get('Fuel Economy [mpg]');
plot(FECity);
```

```
### Searching for referenced models in model 'ConfiguredConventionalVirtualVehicle'.
### Found 7 model reference targets to update.
### Starting serial model reference simulation build.
### Successfully updated the model reference simulation target for: NoBMS
```

Extract Results

Extract the final city and highway fuel economy results for the city and highway drive cycles from the logged data.

```
logsout1 = simout1.get('logsout');
FuelEconomyCity = logsout1.get('Fuel Economy [mpg]').Values.Data(end);
logsout2 = simout2.get('logsout');
FuelEconomyHwy = logsout2.get('Fuel Economy [mpg]').Values.Data(end);
```

Use the city and highway fuel economy results to compute the combined sticker mpg.

```
FECombined = 0.55*FuelEconomyCity + 0.45*FuelEconomyHwy;
```

Extract the tailpipe emissions from the city drive cycle.

```
HC = logsout1.get('HC [g/mi]').Values.Data(end);
CO = logsout1.get('CO [g/mi]').Values.Data(end);
NOx = logsout1.get('NOx [g/mi]').Values.Data(end);
CO2 = logsout1.get('CO2 [g/km]').Values.Data(end);
```

Display the fuel economy and city drive cycle tailpipe emissions results in the command window.

```
fprintf('\n**********************\n')
fprintf('FUEL ECONOMY\n');
fprintf('   City:     %4.2f mpg\n', FuelEconomyCity);
fprintf('   Highway:  %4.2f mpg\n', FuelEconomyHwy);
fprintf('   Combined: %4.2f mpg\n', FECombined);
fprintf('\nTAILPIPE EMISSIONS\n');
fprintf('   HC:   %4.3f [g/mi]\n',HC);
fprintf('   CO:   %4.3f [g/mi]\n',CO);
fprintf('   NOx:  %4.3f [g/mi]\n',NOx);
fprintf('   CO2:  %4.1f [g/km]\n',CO2);
fprintf('   NMOG: %4.3f [g/mi]',HC+NOx);
fprintf('\n**********************\n');
```

```
**********************
FUEL ECONOMY
   City:     39.73 mpg
   Highway:  54.42 mpg
   Combined: 46.34 mpg

TAILPIPE EMISSIONS
   HC:   0.001 [g/mi]
   CO:   0.000 [g/mi]
   NOx:  0.001 [g/mi]
   CO2:  138.2 [g/km]
   NMOG: 0.002 [g/mi]
**********************
```

Fuel Economy (economia de combustível) com unidades em **mpg** significa "**miles per gallon**," ou milhas por galão, que é uma medida de quantas milhas um veículo pode percorrer com um galão de combustível. Essa unidade é amplamente usada nos Estados Unidos e no Reino Unido para indicar a eficiência de combustível de um veículo.

<u>Interpretação de mpg</u>

•**Quanto maior o valor em mpg**, maior é a eficiência de combustível do veículo, pois ele consegue percorrer mais milhas com a mesma quantidade de combustível.

•Valores mais baixos em mpg indicam um consumo de combustível maior, ou seja, o veículo percorre menos milhas por galão, sendo menos eficiente.

Conversão para km/L

Para entender o valor de mpg em um contexto mais familiar, como em km/L, uma conversão comum é:

$$\boxed{\mathrm{km/L} = \mathrm{mpg} \times 0.4251}$$

Por exemplo, um veículo que tem 30 mpg equivale a aproximadamente **12,75 km/L**.

No MATLAB Powertrain Blockset, a variável que representa o **consumo de combustível** em um modelo de veículo é geralmente chamada de **Fuel Consumption** ou **Fuel Rate**. Essa variável pode variar dependendo da configuração e dos componentes usados no modelo. Algumas das variáveis mais comuns para o consumo de combustível incluem:

1. **fuelConsumption**: Total de combustível consumido ao longo do ciclo de simulação, geralmente medido em litros por 100 km (L/100 km) ou gramas por quilômetro (g/km).

2. **fuelFlowRate** ou **Fuel Rate**: Taxa instantânea de consumo de combustível, medida em gramas por segundo (g/s) ou quilogramas por segundo (kg/s). Esse valor é útil para monitorar o consumo de combustível em tempo real durante a simulação.

3. **cumulativeFuel**: Combustível acumulado consumido desde o início da simulação, que permite calcular a eficiência de combustível e o consumo total ao final de um ciclo de condução.

Essas variáveis são geralmente obtidas de sensores virtuais ou blocos de medição conectados ao modelo do motor, como o bloco de *Internal Combustion Engine*. Elas podem ser visualizadas usando blocos de escopo (scope) ou registradas para análise pós-simulação.