CIGI 2021
A Computer-Aided Verification Process for Engineered Systems

VICTOR ROMERO[1], ROMAIN PINQUIE[1], FREDERIC NOEL[1]

[1] UNIV. GRENOBLE ALPES, CNRS, GRENOBLE INP*, G-SCOP, 38000 GRENOBLE, FRANCE
* INSTITUTE OF ENGINEERING UNIV. GRENOBLE ALPES
46 avenue Félix Viallet, 38 000 GRENOBLE, France
{victor.romero, romain.pinquie, frederic.noel}@grenoble-inp.fr

***Résumé* – Les acteurs de l'industrie du futur essayent de systématiquement numériser leurs produits et leurs processus. La numérisation systématique des objets techniques du bureau d'études reste néanmoins très inégalitaire. En effet, les outils qui supportent la conception détaillée sont plus sophistiqués que les outils de spécification ou d'architecture. Notons tout de même la récente évolution d'une ingénierie système basée sur les modèles plutôt que sur les documents. Cependant, les modèles assimilables à des diagrammes 2D inertes, ne facilitent pas assez la détection des erreurs de spécification (validation) et de conception (vérification) en amont des tests d'intégration et des essais de qualification. Nous proposons alors un processus holistique de vérification assistée par ordinateur. Celui-ci permet de s'assurer qu'une solution de conception satisfait ou non aux exigences, et ce, dans des conditions opérationnelles dont l'amélioration progressive du niveau de réalisme assure une continuité entre le système virtuel et le système réel. Notre processus de vérification assistée par ordinateur repose, d'une part, sur la cosimulation d'un modèle de spécification et d'un modèle de conception et, d'autre part, sur une immersion dans une scène virtuelle qui permet de mieux apprécier les propriétés du système, et ce, avant de progressivement substituer le virtuel par le réel au moyen de simulations Hardware-In-the-Loop.**

***Abstract* – Although companies systematically strive for a full digitalisation of their products and their processes, the design phase shows that the quality of models is very unequal. Indeed, detailed design benefit from much more sophisticated methods and tools than the specification and architecture activities. However, we should note the recent paradigm shift from document-based to model-based system engineering. However, these models, which are mainly static 2D diagrams, remain too poor to facilitate the early detection of design errors. Thus, to detect most errors that occur during the design phase, companies have no other alternative than to wait up to the testing phase which occurs after several years for complex systems. Thus, we propose a holistic user-centred computer-aided verification process to ensure that the design meets the requirements under realistic operational conditions. The verification process provides a progressive immersion into the virtual system before seamlessly transitioning from the virtual to the real system. Our work relies on state-of-the-art MBSE methods such as the Property Model Methodology, which enables systems engineers to co-simulate specification models and design models. We improve such MBSE methods by increasing the level of realism that experiences the end-user during the verification of a design by the original combination of Model-In-the-Loop, Immersive Model-In-the-Loop, Human-In-the-Loop, and Hardware-In-the-Loop simulation strategies.**

***Mots clés* – Ingénierie Système, Vérification, Réalité Virtuelle, Modélisation et Simulation, Retours Haptiques**

***Keywords* – Model-Based Systems Engineering, Verification, Virtual Reality, Human-In-the-Loop Simulation**

1 INTRODUCTION

1.1 Context

In the past decades, the design of increasingly complex engineered systems forces the designers to replace the traditional document-based systems engineering approach with Model-Based Systems Engineering (MBSE). In its systems engineering vision 2020 (Crisp, 2007), the International Council On Systems Engineering (INCOSE) defined Model-Based Systems Engineering (MBSE) as *"the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases*". To date, we perceive the motivation for engineering systems using models through two different lenses.

On the one hand, some practitioners argue that the use of models to support an engineering activity is an age-old solution since models have been used for decades to engineer technological systems. Indeed, the roots of MBSE can be traced back to the pioneer work achieved by Wayne Wymore. In his book entitled "*Model-Based Systems Engineering*" (Wymore, 1993), Wymore has shown the crucial role of mathematics in the specification and design of systems. By theorising systems engineering concepts, he paved the way for MBSE, but his visionary mathematical theory of systems engineering was so far ahead of systems engineers that it took several decades before his contribution was recognized (Bahill, 2011).

On the other hand, the model-based approach is in the limelight today because there is another community of engineers who see MBSE as the salvation of systems engineering. The intellectual Renaissance of MBSE results from academic researchers and industrialists who have worked out a set of theoretical and methodological concepts related to systems engineering, a branch that originally was more an irrational practice rather than a rigorous method (Suh, 2001).
MBSE rests on a four-activity process: specification, validation, design, and verification (INCOSE, 2015, sec. 9.2.1). Our study focuses on the verification activity. It is important not to confuse verification that intends to ensure that the "product is built right" and validation that intends to ensure that the "right product is built" (INCOSE, 2015).
According to the INCOSE, there exists four verification techniques: inspection, demonstration, testing, and analysis. Here we concentrate on the analysis.

> *"Analysis – This technique is based on analytical evidence obtained without any intervention on the submitted element, using mathematical or probabilistic calculation, logical reasoning (including the theory of predicates), modelling, and/or simulation under defined conditions to show theoretical compliance. Mainly used where testing to realistic conditions cannot be achieved or is not cost-effective."*

The recent development of cutting-edge computational capabilities has opened new vistas for the analytical verification of design solutions (Hoppe, Engel, & Shachar, 2007). Carried out early in the design process, analytical verification using executable models enables companies to drastically reduce costs and time-to-market while improving quality (Becquet et al., 2018; Fabre, Micouin, Gaurel, & Pandolfi, 2020).
Nevertheless, when we observe the verification activity in the industry, we notice that there are several – more or less advanced – practices including, but not limited to, the sharing of 2D captures of models, dynamic co-simulation of specification and design models, immersive 3D design reviews, hardware-in-the-loop simulation, physical testing, etc., but there is no holistic computer-aided verification process that aims at logically articulating them.

1.2 Problem(s)

Document-Based System Engineering (DBSE) is limited for the design of complex engineered systems. Indeed, the content of documents is difficult to maintain, synchronise, and access (INCOSE, 2015). When looking for design errors during the verification activity, engineers do not have other alternatives than to get together for a pragmatic design review where they share screenshots of their models (SysML, CAD, physics-based simulation, testing reports, etc.) supporting a relatively subjective argumentation process that aims at justifying the satisfaction of requirements stored in a large and ambiguous document-based specification.
To make the verification activity more objective, recent MBSE methods propose to verify whether the design meets the requirements by the co-simulation of design models and specification models (Becquet et al., 2018). Requirements are not natural language statements but logical assertions that monitor the state of design variables in near real-time. Although these new MBSE methods and tools are essentials for the early detection of design errors, the lack of visualisation and interactions of MBSE virtual environments do not enable stakeholders to experience the virtual system in realistic conditions (Figure 1).

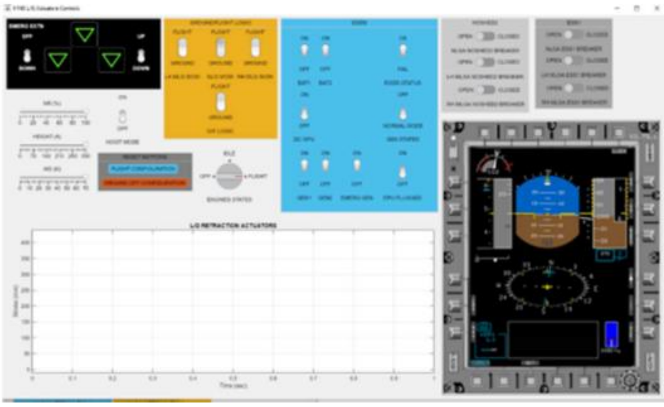

Figure 1: Visualisation virtual cockpit in the MathWorks suite (Fabre et al., 2020)

There are numerous ways to experience a virtual system in realistic conditions, but there is no integrated and continuous verification strategy. For instance, to increase the appreciation of structural properties, we can immerse stakeholders in an immersive environment using Virtual Reality (VR) (Han & Black, 2011). However, the isolation of the specification and the design also leads to a subjective design review. Indeed, designers look for potential design errors in immersion without a reliable strategy since requirements are stored in external ambiguous documents. Regarding the behavioural properties of a system, we can replace models of sub-systems with real components and carry out Hardware-In-the-Loop simulations to get rid of some modelling assumptions and therefore be more realistic (Sarhadi & Yousefpour, 2015), but the specifications remain isolated from the design too.

Those limits motivate us to ask the question:

How to logically combine advanced V&V technologies (MBSE, Simulation, VR, HIL, etc.) to perform an objective verification with a user-centred process, in realistic conditions?

1.3 Proposal

To connect the dots, we propose a holistic computer-aided verification process for continuously verifying that the design satisfies the requirements with objective evidence in realistic conditions. Our computer-aided verification process is also a continuous model-centric integration strategy from the specification phase to the first prototype.
This process relies on the use of virtual reality to visualise and interact with the simulation, to improve the quality of the visualisation and add natural interactions with the simulated system adding then operational randomness.

2 LITERATURE REVIEW

In this section, we start with a fundamental definition of the verification activity and briefly introduce model- and simulation-based systems engineering methods that will serve as a starting point for our research study.

2.1 Verification

According to the main systems engineering standards: "*The purpose of the verification process is to provide objective evidence that a system or a system element fulfils its specified requirements and characteristics*" (ISO/IEC/IEEE 15288, [6.4.9.1]) (INCOSE, 2015).

Except for analysis, all the verification types require a physical prototype. Building a physical prototype is excessively expensive for complex systems. To pass by this issue, the MBSE approach emerged in the past decades.

The starting point of the verification activity is the formalisation of the requirements, to allow an objective comparison between the actual system and what is expected. For this purpose, standardised graphical annotations such as SysML (Karban, Hauber, & Weilkiens, 2015; OMG, 2017) and Capella (Roques, 2016) help to develop and manage the requirements but do not natively provide any formal validation and verification capabilities. However, the next section will discuss recent studies that focus on the use of simulation software to formally model computable requirements that fulfil a monitoring function. Tight integration of specification models and design models in a co-simulation environment helps to identify design errors early on.

2.2 Model- and Simulation-Based Verification

Engineers use modelling and simulation technologies for verifying complex engineered systems in a very subjective way. Indeed, as Suh (Suh, 2001) reports "*Often designers find that the precise description of "what we want to achieve" is a difficult task. Many designers deliberately leave them implicit rather than explicit and then, start working on design solutions even before they have clearly defined their design goals. They measure their success by comparing their design with the implicit design goals they had in mind. They spend a great deal of time to improve and iterate the design until the design solution and "what they had in mind" converge, which is a time-consuming process at best.*" (Suh, 2001). Today, the most common verification approach consists in modelling and simulating the properties of the system and organising design reviews to approve or reject a design candidate based on the results of the simulations without a rigorous definition of the requirements and their integration with design solutions.

To satisfy Suh's recommendation "*To be efficient and to generate the design that meets the perceived needs, we must specifically state the design goals in terms of "what we want to achieve" and begin the design process.*" (Suh, 2001), recent MBSE methods and tools propose formal specification and verification activities. For instance, Micouin proposed the Property Model Methodology (PMM) (Micouin, 2013). PMM was successfully applied to an academic case study using Modelica (Pinquie et al., 2016) before to be evaluated on an operational Airbus Helicopter program (Becquet et al., 2018; Fabre et al., 2020). The PMM process is based on formally defined requirements using the concept of Property-Based Requirements (PBR) (Micouin, 2010). After the specification, validation, and design activities comes the verification thanks to the co-simulation of design models and specification models. PBR can be supported by various simulation language such as Modelica (Buffoni & Fritzson, 2016; Nguyen, 2014; Otter et al., 2015) or the MathWorks suite (Nilsson, 2014).

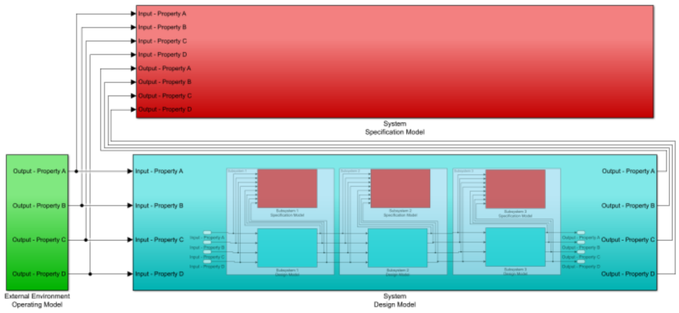

Figure 2: With the Property Model Methodology, the model- and simulation-based verification activity requires: the top-down modelling of (1) the system specification model (block in red), (2) the system design model (block in blue), (3) the sub-systems specification models (blocks in red within the design model), (4) the sub-systems' design models (block in blue within the design model), (5) the operating scenarios that stimulates the specification and design models (block in green).

The limits of these model- and simulation-based verification approaches pointed out during the introduction are the weak visualisation and interaction capabilities of the physics-based simulation tools that support the method, and which lead to a low level of realism for the end-user experience.

2.3 Virtual Reality-Based Verification

Casenave & Lugo (2018) demonstrated that Virtual Reality technologies, which support the 3D representation of a system, give much more natural access to the product for the end-user, improving the understanding of the data. This encourages the industries to adopt these technologies for various applications including the verification activity (Berg & Vance, 2017; PWC, 2015), but the experience is still limited to the inspection-like verification. Indeed, the immersive environments increase the end-user experience when analysing structural properties, but they hardly integrate a physics-based simulation for emulating the behavioural properties and do not integrate formal specification models. Therefore, the design remains separated from the specification.

The literature review shows that there are various methods and tools for verifying designs early on. However, there is no holistic computer-aided verification process that logically integrates advanced practices.

3 A Holistic Computer-Aided Verification Process

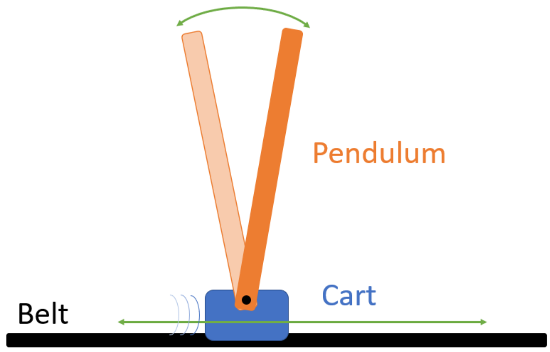

Figure 3: Inverted Pendulum

To illustrate our computer-aided verification process, we will use an inverted pendulum (Figure 3), a system that is not complex but sufficiently rich to illustrate our proposal as it contains five multi-engineering subsystems (Figure 4).

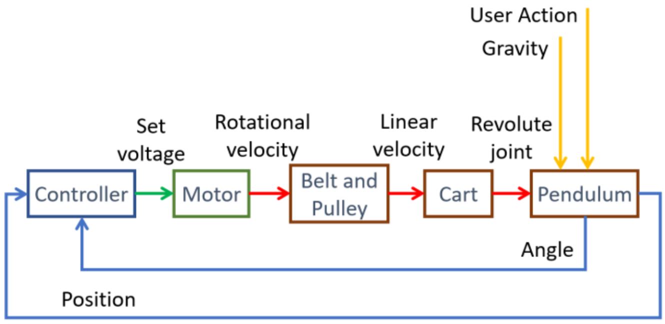

Figure 4: Architecture of the inverted pendulum

We assume that an efficient computer-aided verification process (Figure 5), which provides increasing analytical evidence in realistic conditions, starts with a state-of-the-art analytical verification activity: the co-simulation of design models with specification models – **Model-In-the-Loop Verification**.

The second activity of the computer-aided verification process consists of immersing the end-user in a virtual world to improve his experience in terms of visualisation – **Immersive Model-In-the-Loop Verification**.

Always with the desire to improve the degree of realism, especially the interactions with the virtual world, the third activity integrates the human to realistically stimulate the models in near real-time. – **Immersive Model- and Human-In-the-Loop Verification**.

The fourth activity removes modelling assumptions by a continuous substitution of virtual building blocks from the design model by real ones up to the first real prototype of the system - **Immersive Model-, Human- and Hardware-In-the-Loop Verification**.

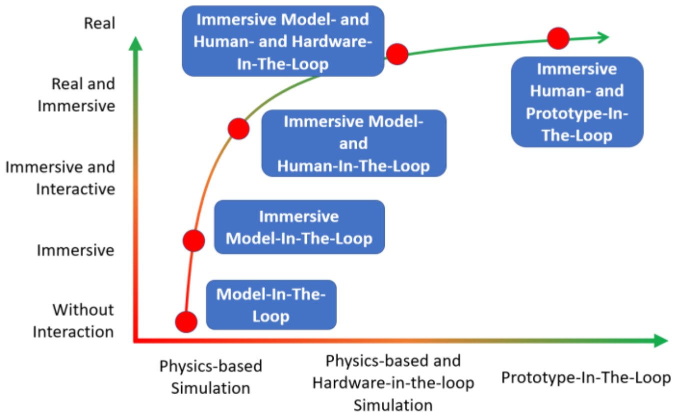

Figure 5: A continuous and holistic computer-aided verification process from virtuality to reality

3.1 Model-In-The-Loop Verification

3.1.1 Definition

We start from the state-of-the-art MBSE methods by using the Property Model Methodology (Micouin, 2013) for its Model-In-the-Loop verification strategy implemented with the MathWorks suite (Becquet et al., 2018; Fabre et al., 2020). Model-In-the-Loop verification consists of the co-simulation of a formal specification model and a discrete and/or continuous design model. The specification model monitors the design models (Figure 6). We assume that these models are already created in a previous design step and we expect to check that the design matches the specifications. Currently, our process starts at this design stage.

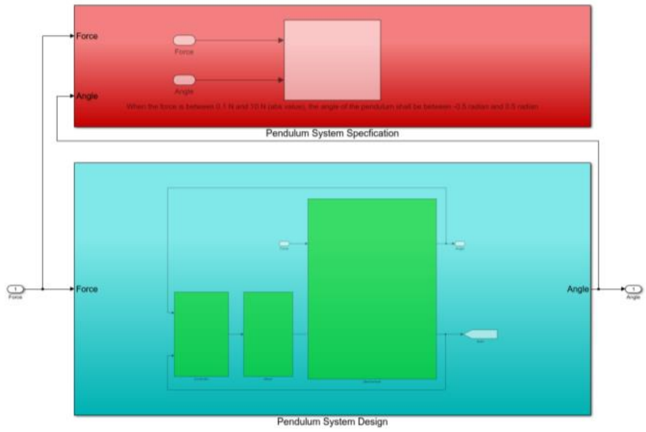

Figure 6: The specification model (in red) monitors the design model (in blue for the system and in green for the sub-systems)

3.1.2 Implementation

The quality of the implementation depends on the capabilities of software that supports the PMM method. We use the MathWorks suite to co-simulate our specification models and our design models because past experiences (Fabre et al., 2020; Pinquie et al., 2016) demonstrated that its maturity outperforms existing alternatives such as Modelica-based solutions. As an example, Figure 7 shows a specification model (in orange) containing a formal requirement. The inputs of the specification model correspond to the external stimulus – i.e. the force applied to the pendulum – and the intended effect – i.e. the angular position of the pendulum. The specification, which is simplified to a single requirement here, makes sure that the design behaves as expected under specific conditions, that is, the pendulum remains in equilibrium if the magnitude of the force acting on the pendulum belongs to the specified interval. Here we assume that the equilibrium state corresponds to an angular position between -0,5 and 0,5 radian. Figure 7 show that the specification of the system is derived into sub-systems specifications by satisfying the prime contractor theorem which makes sure that sub-systems requirements are equally or more constraining than the systems requirements.

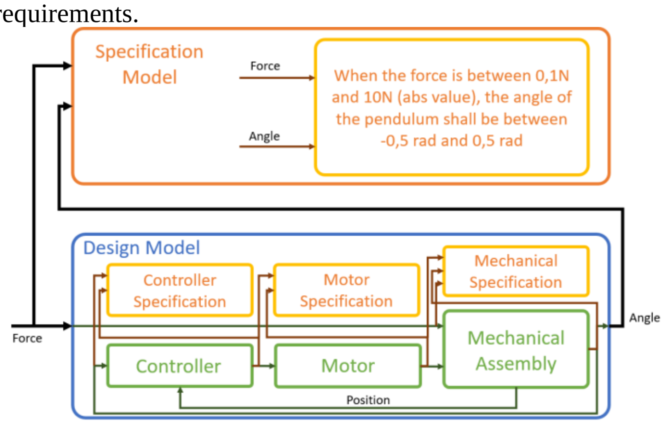

Figure 7: Representation of the Model-In-The-Loop verification in the MathWorks suite.

3.1.3 Limits

The modelling scheme of Model-In-the-Loop verification is limited to predefined operational scenarios, whereas sound verification requires extensively to stimulate the system. There is therefore a need to define numerous and heterogeneous operational scenarios to make sure the design meets the requirements. The definition of operational scenarios is limited to the nominal and well-known critical scenarios while missing potential unintended situations. Furthermore, the visuals are graphs, tables, or 2/3D animations, whereas simulation outputs require an intuitive visualisation thanks to immersive visualisation techniques such as virtual reality (Han & Black, 2011; Pausch, Proffitt, & Williams, 1997). By integrating virtual reality into PMM, the level of realism would be higher because of the improvement of many visualisation performance criteria, including but not limited to stereoscopy, field of view, display size and resolution, frames per second (Bowman & McMahan, 2007).

3.2 Immersive Model-In-The-Loop Verification

3.2.1 Definition

Reminding that our goal is to be closer to a test in *realistic conditions*, we expect to improve the realism of the Model-in-the-Loop verification with visual immersion. Thus, to perform a much more intuitive visualisation of the system properties, we conserve the co-simulation of specification models and design models, but we replace the native visualisation capabilities of the MathWorks suite with an immersive environment giving birth to the Immersive Model-in-the-Loop verification strategy (Figure 8).

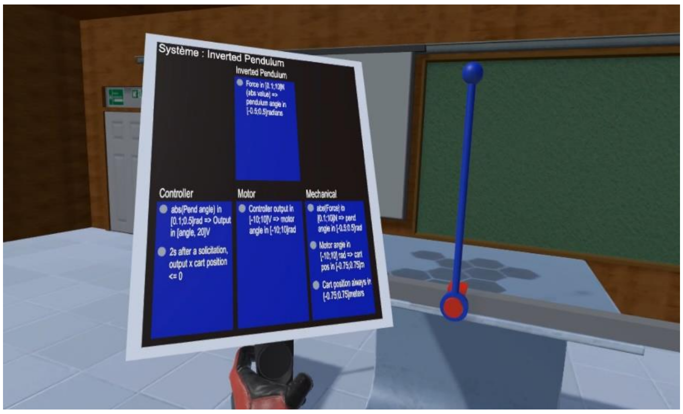

Figure 8: User view of the inverted pendulum in a virtual environment and the requirements on a pad.

3.2.2 Implementation in the use case

Immersive Model-in-the-Loop verification requires a VR engine to communicate with the Model-In-the-Loop simulation. This interconnection uses UDP network communication because there is no verification from the server of the data received by the client leading to faster data exchanges. The VR visualisation module was developed within the Unity3D framework (Unity Technologies, 2020). We use the Simulink block SendUDP to send the position of the cart and the rotation of the pendulum around the cart axis to another computer in which Unity3D is running (Figure 9).

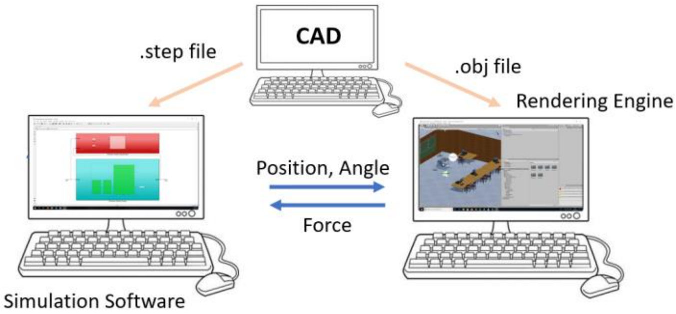

Figure 9: Immersive Model-In-The-Loop verification

In this second software, to animate the system in the virtual environment created in the Unity3D graphic window, we can open the appropriate ports and read the received data thanks to C# scripts. Regarding the visual metaphors for the specification model, we use a pad that the user activates and deactivates using a key of the left-hand remote controller. This pad contains a representation of the system architecture with the system at the top and the sub-systems at the bottom (Figure 10). Each system-of-interest contains a textual representation of the requirements contained in its specification model. The virtual lamp that stands next to each requirement turns green when the requirements are tested and satisfied, red when the requirements are tested and unsatisfied, and remains grey when it is untested.

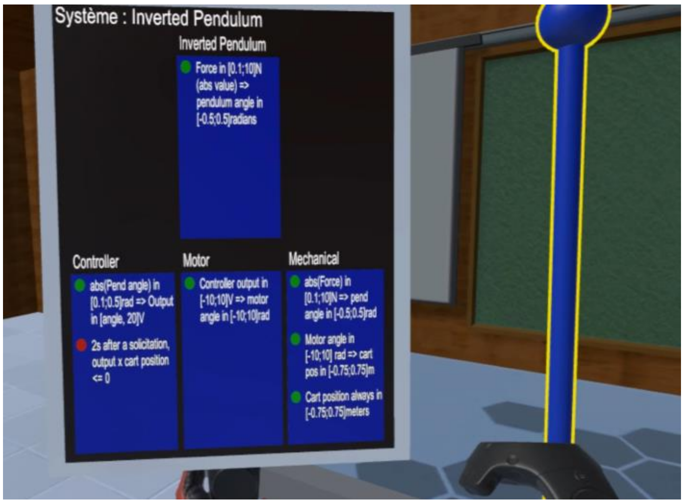

Figure 10: User view of interaction with the pendulum (highlighted when touching with the user's virtual hand)

3.2.3 Limits

Immersive Model-in-the-Loop improves the visual appreciation of the structural and behavioural properties. However, we still have a lack of realism in the interactions provided by this simulation. Indeed, the user cannot directly interact with the system limiting the virtual experience to planned (models or recordings embedded in the behavioural model) operational scenarios without unintended events.

3.3 Immersive Model- and Human-In-The-Loop Verification

3.3.1 Definition

In the last section, we improved the realism during the verification activity, but VR must not be restricted to 3D visual perception as it offers advanced interactive capabilities. The integration of Human-In-the-Loop simulation into the Immersive Model-In-the-Loop Verification aims at providing the end-user with the opportunity to naturally operate the system. We name this new verification strategy Immersive Model- and Human-In-the-Loop verification (Figure 10, Figure 11 and Figure 12).

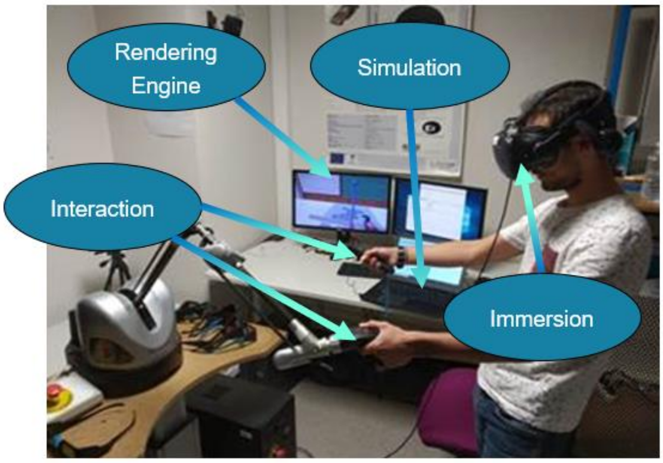

Figure 11: Example of an Immersive Model- and Human in the loop verification

3.3.2 Implementation in the use case

To achieve an Immersive Model- and Human-In-the-Loop verification, we need changes in the simulation and visualisation software. The simulation will have a new input standing for the end-user stimulus. We can use the same communication protocol as for the visualisation but in the other direction: the VR software sends the interaction data to the physics-based simulation through the communication network using a UDP protocol and the simulation software receives it and uses it as an input of the system. These interactions are captured by the remote controllers provided with the HMD HTC Vive. We also use a haptic arm, the Virtuose 6D 35-45, which captures the position of the hand and gives force feedback when the user touches the virtual pendulum. When providing significant force feedback, the user develops a certain apprehensiveness, not to say fear, to touch the pendulum again leading to an increase of the level of presence, that is, the sense of “being in” the scene and the user starts behaving as he was experiencing the system in the real world.

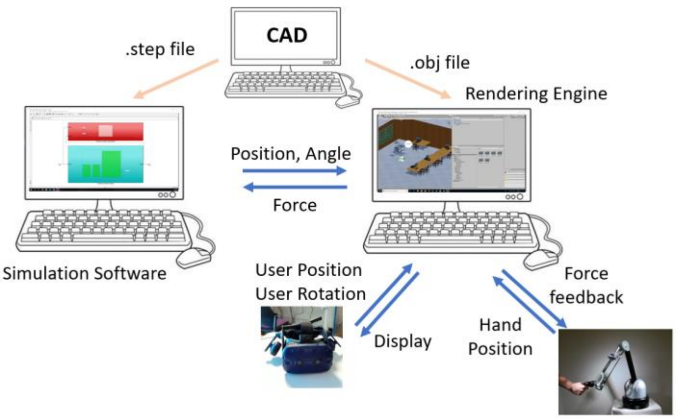

Figure 12: Immersive Model- and Human-In-The-Loop verification including haptic feedbacks

3.3.3 Limits

The natural interaction with realistic behaviour and structural properties improves the early verification activity. However, as the system is fully virtual, modelling assumptions may lead to unintended events.

3.1 *Immersive Model-, Human-, and Hardware In-The-Loop Verification*

3.1.1 Definition

Immersive Model, Human- and Hardware-In-the-Loop Verification consists of substituting virtual building blocks from the design model by real ones up to the first real prototype instrumented with sensors that feed the system specification model. The simulation of the system under design embedding real components enable designers to get rid of modelling assumptions leading to potential design errors.

3.1.2 Implementation in the use case

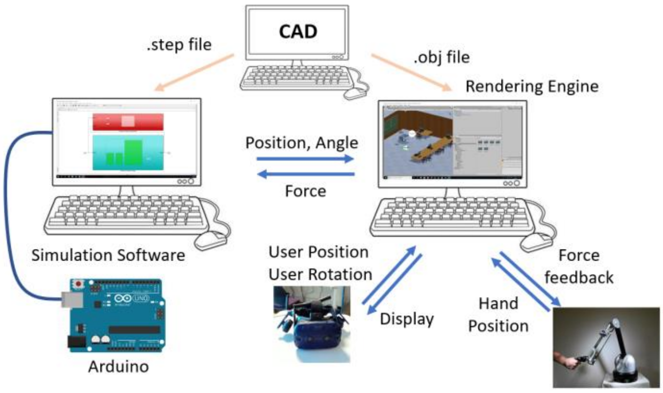

Figure 13: Immersive Hardware- and Human- and Model-In-The-Loop verification

In our case study, we can start replacing the design model of the “Controller” sub-system (Figure 4) with the Arduino Uno card (Figure 13). With this strategy, we can proceed to a step-by-step verification of the real components before integrating them together up to the verification of the full prototype instrumented with sensors (Figure 14).

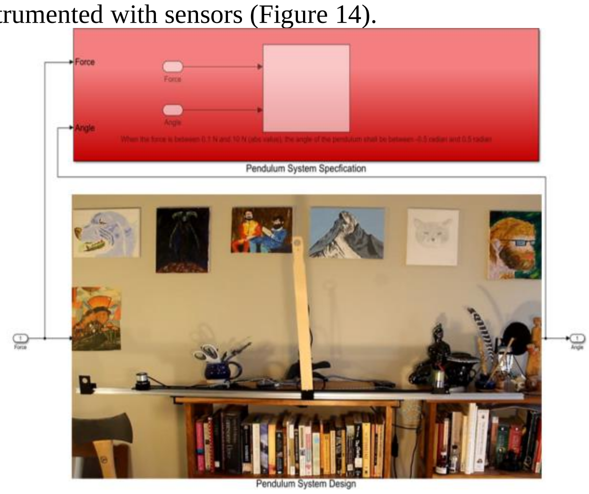

Figure 14: Verification of the first functional prototype with the system specification model in the loop

4 CONCLUSION

Model-based requirements and systems engineering, virtual reality, physics-based simulation, and hardware-in-the-loop simulations are key technologies for the early verification of engineered systems. However, there was a need for continuous integration of these technologies into a holistic computer-aided verification process for verifying engineered systems in realistic conditions.

We demonstrated that it is possible and relevant to integrate VR technologies with an MBSE method to recreate realistic operational conditions while keeping the added-value of the co-simulation of specification models and design models. Also, following the growing maturity of the designed system by progressively substituting the virtual building blocks with real ones, the designer can be confident that the structural and behavioural properties of the first prototype will satisfy the requirements, avoiding then extra costs due to design errors.

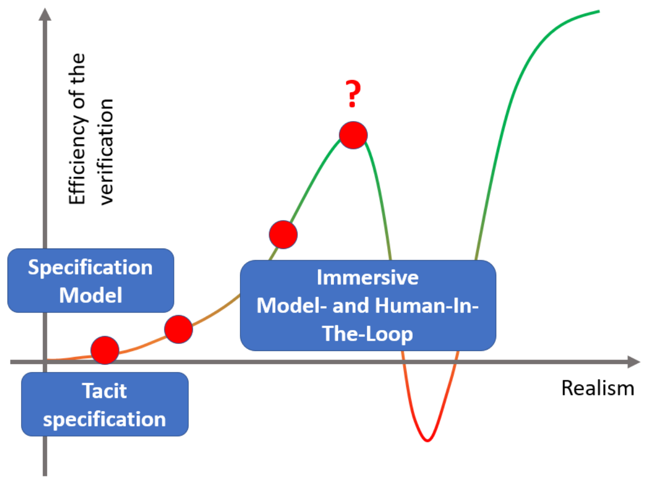

Figure 15: An attempt to conceptually define the uncanny valley of immersive verification

As future works, we plan to develop a reporting dashboard in Unity 3D containing indicators related to the verification activity. We will also test the proposed computer-aided verification process in an industrial context. Moreover, analytical verifications strategies strive to integrate numerous technologies and develop immersive virtual environments to experience more and more realistic conditions. Nevertheless, we hardly know: how much realism is enough to verify a design? We may therefore carry out new research studies to derive sound and parsimonious recommendations that would help companies to reach the right level of realism according to the verification goals and the available resources.

To provide more objective criteria, we may borrow the phenomenon of the uncanny valley suggested by the researches applied to humanoids (Mathur & Reichling, 2016). The uncanny valley translates the unexpectedly negative dislike reactions provoked by imperfect human-likeness in human robots (Mori, 1970). By analogy, we may assume that more sensorial feedback does not systematically increase the level of realism. For example, studies show that is the case for haptics feedback (Berger, Gonzalez-Franco, Ofek, & Hinckley, 2018). We may therefore assume that, at a certain point, more realism does not automatically increase the efficiency of the verification process. Figure 15 illustrate a new research hypothesis stating that this phenomenon may exist in our virtual environment for systems verification. Therefore, as future work, we can try to demonstrate the existence or not of the uncanny valley phenomenon in the virtual phases of our computer-aided verification process to finally be able to characterise the "right need" of the user, that is, the perfect balance between the time and cost spent for developing a realistic virtual environment and its efficiency in terms of verification (e.g. the number of design errors identified).

5 References

Bahill, T. (2011). In Memoriam A. Wayne Wymore. *INSIGHT*, *14*(2), 57–61. https://doi.org/10.1002/inst.201114257

Becquet, R., Micouin, P., Fabre, L., Guérin, F., Paper, P., & Razafimahefa, T. (2018). Property Model Methodology: A Landing Gear Operational Use Case. *INCOSE International Symposium*, *28*(1), 321–336. https://doi.org/10.1002/j.2334-5837.2018.00484.x

Berg, L. P., & Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality*, *21*(1). https://doi.org/10.1007/s10055-016-0293-9

Berger, C. C., Gonzalez-Franco, M., Ofek, E., & Hinckley, K. (2018). The uncanny valley of haptics. *Science Robotics*, *3*(17), eaar7010. https://doi.org/10.1126/scirobotics.aar7010

Bowman, D. A., & McMahan, R. P. (2007). Virtual reality: How much immersion is enough? *Computer*, *40*(7), 36–43. https://doi.org/10.1109/MC.2007.257

Buffoni, L., & Fritzson, P. (2016). Expressing Requirements in Modelica. *SNE Simulation Notes Europe*, *25*(3–4), 21–22. https://doi.org/10.11128/sne.25.tn.10314

Crisp, H. (2007). *INCOSE systems engineering vision 2020*.

de Casenave, L., & Lugo, J. E. (2018). Effects of Immersion on Virtual Reality Prototype Design Reviews of Mechanical Assemblies. *Volume 7: 30th International Conference on Design Theory and Methodology*, *7*, 1–11. https://doi.org/10.1115/DETC2018-85542

Fabre, L., Micouin, P., Gaurel, C., & Pandolfi, P. (2020). *Advances in Property Model Methodology (PMM ®)*.

Han, I., & Black, J. B. (2011). Incorporating haptic feedback in simulation for learning physics. *Computers and Education*, *57*(4), 2281–2290. https://doi.org/10.1016/j.compedu.2011.06.012

Hoppe, M., Engel, A., & Shachar, S. (2007). SysTest: Improving the verification, validation, and testing process— Assessing six industrial pilot projects. *Systems Engineering*, *10*(4), 323–347. https://doi.org/10.1002/sys.20082

INCOSE. (2015). *Systems engineering handbook A guide for system life cycle processes and activities*.

Karban, R., Hauber, R., & Weilkiens, T. (2015). MBSE in Telescope Modeling. *INSIGHT*, *12*(4), 24–31. https://doi.org/10.1002/inst.200912424

Mathur, M. B., & Reichling, D. B. (2016). Navigating a social world with robot partners: A quantitative cartography of the Uncanny Valley. *Cognition*, *146*, 22–32. https://doi.org/10.1016/j.cognition.2015.09.008

Micouin, P. (2010). Toward a Property BasedRequirements Theory:System RequirementsStructured as a Semilattice. *Wiley InterScience*, *14*(3), 305–326. https://doi.org/10.1002/sys

Micouin, P. (2013). Property-Model Methodology: A Model-Based Systems Engineering Approach Using VHDL-AMS. *Wiley Online Library*, *14*(3), 305–326. https://doi.org/10.1002/sys

Mori, M. (1970). The uncanny valley. *Energy*, *7*, 33–35.

Nguyen, T. (2014). FORM-L: A MODELICA Extension for Properties Modelling Illustrated on a Practical Example. *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*, *96*, 1227–1236. https://doi.org/10.3384/ecp140961227

Nilsson, V. (2014). *Model-Based Testing with Simulink Design Verifier*. Chalmers University of Technology.

OMG. (2017). OMG SysML Home | OMG Systems Modeling Language. Retrieved from http://www.omgsysml.org/

Otter, M., Thuy, N., Bouskela, D., Buffoni, L., Elmqvist, H., Fritzson, P., … Tundis, A. (2015). Formal Requirements

Modeling for Simulation-Based Verification. *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, *118*, 625–635. https://doi.org/10.3384/ecp15118625
Pausch, R., Proffitt, D., & Williams, G. (1997). Quantifying immersion in virtual reality. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, (August 1997), 13–18. https://doi.org/10.1145/258734.258744
Pinquie, R., Micouin, P., Veron, P., Segonds, F., Pinquié, R., Véron, P., … Segonds, F. (2016). Property Model Methodology : A Case Study with Modelica. *Proceedings of Tools and Methods for Competitive Engineering*, (April), 1–12. https://doi.org/10.13140/RG.2.1.1925.1603
PWC. (2015). For US manufacturing, virtual reality is for real. Retrieved May 23, 2019, from 2015 Disruptive Manufacturing Innovations Survey website: https://www.pwc.com/us/en/industrial-products/next-manufacturing/augmented-virtual-reality-manufacturing.html
Roques, P. (2016). MBSE with the ARCADIA Method and the Capella Tool. *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*.
Sarhadi, P., & Yousefpour, S. (2015). State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software. *International Journal of Dynamics and Control*, *3*(4), 470–479. https://doi.org/10.1007/s40435-014-0108-3
Suh, N. P. (2001). *Axiomatic design : advances and applications*. Oxford University Press.
Unity Technologies. (2020). Multiplatform - VR-AR. Retrieved March 10, 2020, from https://unity3d.com/fr/unity/features/multiplatform/vr-ar
Wymore, A. (1993). *Model-based systems engineering*.