

Exploring Validity Frames in Practice

Simon Van Mierlo^[0000-0002-4043-6883],
Bentley James Oakes^[0000-0001-7558-1434], Bert Van Acker^[0000-0002-3854-5159],
Raheleh Eslampanah^[0000-0001-7213-9536], Joachim Denil^[0000-0002-4926-6737],
and Hans Vangheluwe^[0000-0003-2079-6643]

University of Antwerp - Flanders Make vzw
{simon.vanmierlo,bentley.oakes,bert.vanacker,
raheleh.eslampanah,joachim.denil,hans.vangheluwe}@uantwerpen.be

Abstract. Model-Based Systems Engineering (MBSE) provides workflows, methods, techniques and tools for optimal simulation-based design and realization of complex Software-Intensive, Cyber-Physical Systems. One of the key benefits of this approach is that the behavior of the realized system can be reasoned about and predicted *in-silico*, before any prototype has been developed. Design models are increasingly used after the system has been realized as well. For example, a (design) digital twin can be used for runtime monitoring to detect and diagnose discrepancies between the simulated and realized system. Inconsistencies may arise, however, because models were used at design time that are not valid within the operating context of the realized system. It is often left to the domain expert to ensure that the models used are valid with respect to their realized counterpart. Due to system complexity and automated Design-Space Exploration (DSE), it is increasingly difficult for a human to reason about model validity. We propose validity frames as an explicit model of the contexts in which a model is a valid representation of a system to rule out invalid designs at design time. We explain the essential and conceptual, yet practical, structure of validity frames and a process for building them using an electrical resistor in the optimal design of a high-pass filter as a running example. We indicate how validity frames can be used in a DSE process, as well as for runtime monitoring.

Keywords: Validity Frames, Design-Space Exploration, Digital Twin

1 Introduction

The systems we build today are characterized by an ever-increasing complexity. In recent decades, we have moved from systems that are largely restricted to one domain (mechanical, electrical, software) to large, integrated systems that combine components from many different domains, each with their own formalisms, techniques and tools. This has led to a plethora of applications, typically called Software-Intensive Systems, and more recently Cyber-Physical Systems. Model-Based Systems Engineering (MBSE) provides workflows, methods, techniques and tools for optimal simulation-based design and realization of

such systems [17]. Typically, domain experts create designs in specialized modeling and simulation languages and their accompanying tools. Such designs are an abstraction of the realized systems. For a specific purpose, they allow for the *in-silico* analysis of a limited set of properties of the system-to-be-built. This leads to a tremendous increase in productivity and other side-effects, such as improved communication between different domain experts, traceability to the final design, easier response to changing requirements, etc. [1].

Looking forward, the *digital twin* is seen as an enabler for advanced runtime monitoring [2, 26], fault detection, and (automatic) recovery or reconfiguration [15]. A (design) digital twin, which is a (tracking) simulation of the design model, is run in real-time alongside the realized system and is provided with the same inputs as the realized system. When the results observed in the running system do not correspond to what is predicted by the digital twin, a *discrepancy* occurs, which might be due to a fault in the realized system (due to wear and tear, for example), an error made in the design model, or in simulation solver parameter settings. This might lead to further actions, such as fault localization, recovery of the system, and even a reconfiguration or redesign of the system. In this paper, design models are assumed to be used in some form to create a *design digital twin* that is used as a (advanced) runtime monitor as discussed above.

The search for an optimal design is not only performed manually, but is increasingly automated by providing a Design-Space Exploration (DSE) solver with a number of components, a set of design constraints, and a set of system properties that need to be satisfied and/or that have to be optimized. Exploiting vast computational power currently available, DSE allows exploring many more design candidates than before, possibly leading to original new designs [5].

The problem in any design endeavor, which is magnified when DSE is automated, is to come up with a design that is *valid*, *i.e.*, properties of interest obtained from the realized system are equivalent to those obtained from simulations. In DSE, some domain constraints are already encoded and provided to the solver in order to not come up with designs that can not be realized. However, often overlooked is the *validity* of the used models with respect to their realized counterparts, in their intended operational context. If such validity can not be guaranteed, simulation results and hence DSE results *cannot be trusted!*

Validity frames, first introduced by Zeigler [27], explicitly encode the conditions under which a model is a valid abstraction of the realized system. In the decades after its introduction, other researchers have extended the original definition and have attempted to apply these frames to certain applications in different domains. However, a practical guide for creating and using validity frames with a concrete example is currently missing from the state-of-the-art. In this paper, we introduce the concept of validity frames using a practical example, and show how frames can enhance the (automatic) DSE for a multi-component system.

Structure Section 2 introduces the notion of validity frames with a running example: an electrical resistor whose behavior is specified using four different models. Section 3 demonstrates the use of validity frames on the design of a

multi-component system: a (passive) high-pass filter (HPF). Section 4 discusses related work. Section 5 concludes the paper and provides pointers for future work.

2 Validity Frames

This section presents an overview of our vision on validity frames. In particular, the precise objective of checking the validity of the model will be detailed. Then, we will present the essential components of validity frames and their relations, using an electrical resistor as an example.

2.1 Example Component

An electrical resistor is a common element in many electrical circuits. Many different types of resistors exist; their material, geometry, connection points, insulating material, etc. can vary depending on the intended application. All resistors share a common property though in their reduction of current flow, typically by dissipating the energy as heat.

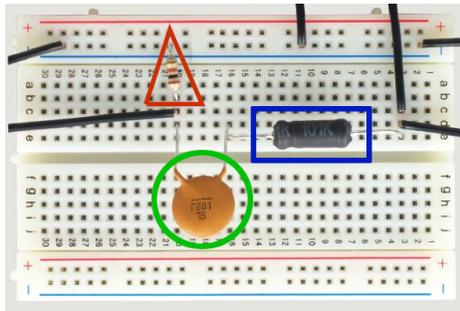


Fig. 1: A realized design of a high-pass filter¹: a circuit consisting of a resistor, capacitor, and inductor.

Figure 1 shows an example electrical circuit consisting of a resistor, a capacitor, and an inductor, marked by a triangle, circle, and square respectively. They are connected through wires, and when the circuit is connected to a voltage source, the effect that the elements have on the electrical signal (governed by the appropriate physical laws) can be observed by measuring the electrical signal at certain points in the circuit using “probe wires”. Typically, such circuits are designed by domain experts (*i.e.*, electrical engineers) who have deep knowl-

edge of these physical effects and can take into account the amount of current that will flow through the complete circuit (induced by the voltage source that is connected to the circuit). This may cause other physical effects to occur: the elements might heat up, their behavior might change if the frequency of the voltage signal is high, or magnetic fields might be induced, which causes (potentially unwanted) behavior changes in the components, or interactions between elements that are placed close together.

When modeling electrical networks, *models* of the electrical elements are used. In Figure 2, four models for the resistor are shown. All of these models have the same interface, a set of physical ports, highlighted in the figures in

¹ Explained further as a running example in Section 3.

yellow: a positive electrical connection P and a negative electrical connection N (which, when connected, generate appropriate equations based on Kirchoff's laws), and a thermal port T . As such, they can be used interchangeably, but their behavior (both internal to the component and visible to the environment) will be different. We consider two influencing factors in this work: the frequency of the electrical signal (connected through the electrical interface of the resistor) and the temperature of the environment. The effects of these influencing factors can either be low or high on the behavior of the component. In Figure 2, we have classified whether the model takes into account the high effects or not.

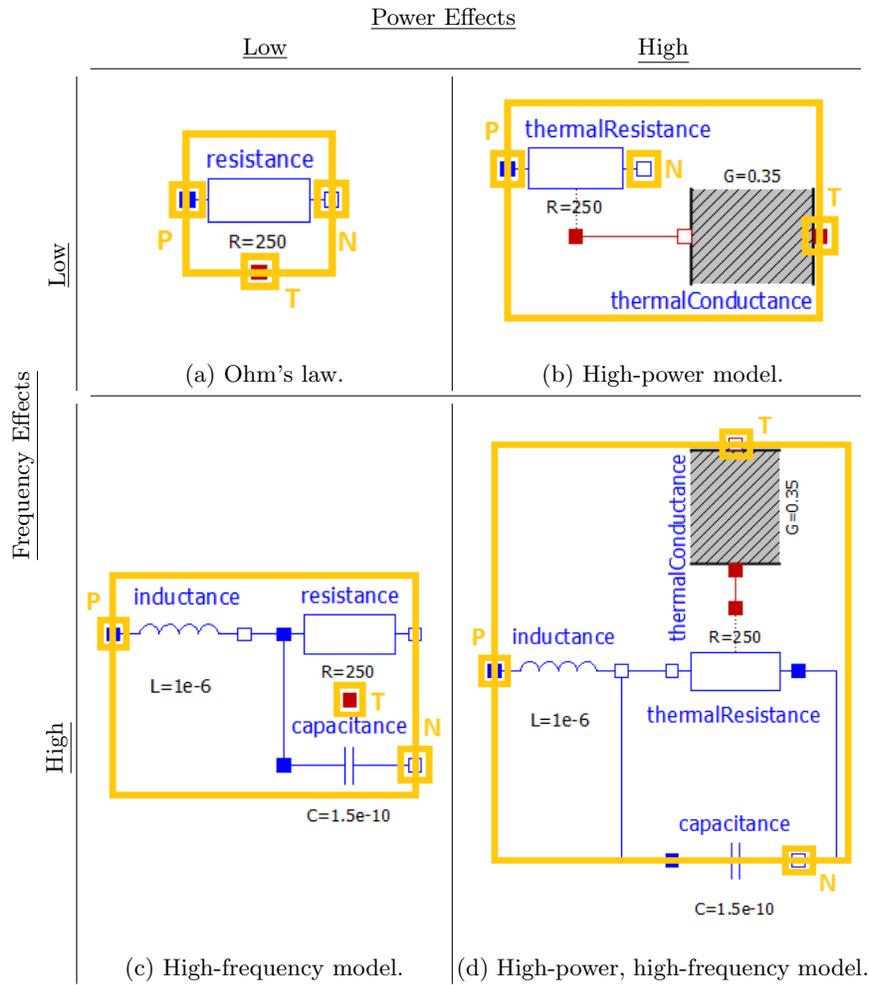


Fig. 2: Different models of a resistor in Modelica

Model (a), Ohm's law, models the resistor behavior as follows:

$$V = R_{nominal} * (1 + alpha * (T_{res} - T_{reference})) * I \quad (1)$$

where $R_{nominal}$ is the nominal resistance value (equal to 250 in the model), $alpha$ is the temperature coefficient, $T_{reference}$ is the reference temperature of the resistor at which its resistance value is equal to the nominal resistance value, and T_{res} is the temperature of the resistor, in this case equal to the ambient temperature (coming into the red thermal port under the resistor in the figure). This effectively models a linear relation between the voltage V and the current I , where the gradient of the function depends on the ambient temperature.

Model (b) takes into account the fact that the resistor itself will heat up over time as a current flows through it. The behavior of the resistor is extended with the following equations:

$$LossPower = v * i \quad (2)$$

$$\frac{dT_{res}}{dt} = \frac{-LossPower + v * i}{tc * K_d} \quad (3)$$

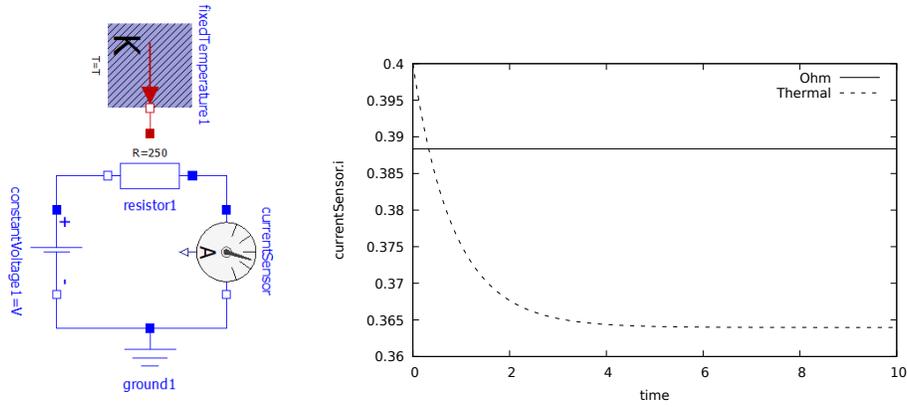
where tc is the thermal time constant, a time unit that specifies how fast the resistor will heat up, and K_d is the dissipation factor, specifying how fast the heat of the resistor is dissipated into the environment.

Model (c) accounts for the fact that a resistor does not only resist the current flowing through it, but it also has slight capacitive and inductive properties. In this high-frequency model, such effects are modeled for AC applications. The electrical network now models the single (real-world) resistor component. The behavior of the inductor is governed by $\frac{di}{dt} = \frac{v}{L}$ where L is the inductance, and the behavior of the capacitor is governed by $\frac{dv}{dt} = \frac{i}{C}$ where C is the capacitance. Last, in model (d), both frequency and temperature effects are taken into account by combining the second and third models of the resistor.

We can run simulations using these models. Suppose we model an electrical network where a resistor is connected to a voltage source with a particular constant voltage V and a current sensor that senses the amount of current that is output by the resistor. This is illustrated in Figure 3a, where we show a resistor model within an experimental context: it is connected to a voltage source, a current sensor, and a fixed ambient temperature. In Figure 3, the results of the simulation are shown: the purple (constant) line is the result of using the simple model for the resistor, while the green (descending) line is the result of using the model of the resistor that takes into account heating effects. The linear model arrives at a value slightly below 0.39 for the current flowing through the resistor, while the model of the thermal resistor settles on a value slightly below 0.365.

2.2 Objective

Figure 4 displays the relationship between a *system* and its *model*, in an adaptation from the works of Zeigler [27]. Starting from a system to be realized, the



(a) The simulation model for the resistor. (b) Comparing the simulation results.

Fig. 3: A simulation and its results for two models of the resistor.

context of the model and the *properties of interest* are defined. Then the model can be built or selected from a *model library* such as the Modelica library. If a model's *simulation results* reproduce the real-world system *experimental results* faithfully within an *experimental context* (and with a particular tolerance, since a perfect match will never be achievable), we say that the model is *valid* with respect to the real system. This validity condition is always in relation to the *properties of interest*; other properties that the system or model exhibits beyond these (as long as they do not influence the considered properties) are irrelevant with respect to the validity condition.

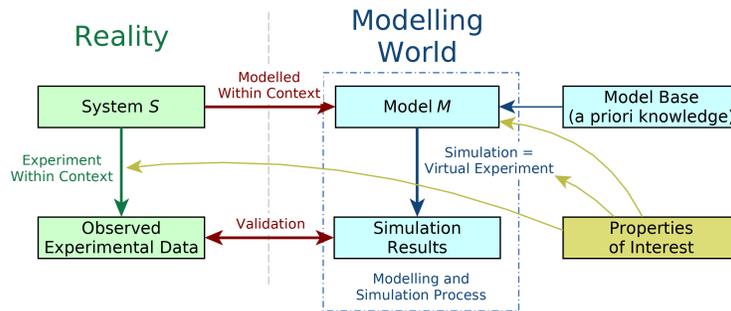


Fig. 4: The modelling relationship between the system and the model.

The objective of validity frames is to explicitly encode the *contexts* in which a *model* provides valid results for a specific set of *properties* with respect to a *real-world system*. Ultimately, this will enable the building of *frame-enabled model repositories*, where the repository contains (1) a set of models, (2) a set of validity frames, and (3) relations between models and validity frames, specifying which frames are *applicable* to which models, and vice versa, which models can be *accommodated* by which frames. This enables correct and efficient design of systems, in particular automated design-space exploration.

In the following subsections, we will provide additional details for these components of validity frames, applied to the electrical resistor.

2.3 Properties

Properties in the validity frame are those properties of the system-under-study that are relevant in the modeling activity, and which can be measured in some way, termed *value properties*. These properties therefore scope the considered context of the validity frame. They are at the core of the modeling activity and of validity frames. A property can be *evaluated* within an *experimental context*. The set of properties of interest gives rise to the abstraction relation between model and system: a model is only supposed to provide (correct) answers with respect to these properties, while other properties are abstracted away.

For the electrical resistor example, let us consider the constituent relation between the voltage V and the current I as the property of interest. In the most basic case, this relation is given by Ohm's law, *i.e.*, $I = \frac{V}{R}$, which constitutes a linear relation, since the resistance value R is supposed to be constant. Considering the models in Figure 2, however, that relation will look different (and more difficult to solve analytically) as more effects are taken into account. We can compare the *fidelity* of a model with respect to a property when compared to the real system that the model represents. The evaluation of that fidelity (and, extended, of the range of validity of the model) can be performed by comparing the result of an experiment in the virtual world (on the model) with the result of a representative experiment in the real world (on the system), or by comparing the results of two experiments in the virtual world, one on the model, and the other on a higher-fidelity model (with a wider validity range).

2.4 Influencing Factors

Influencing factors are properties of the environment that are outside of the control of the model, but can potentially influence its behavior and/or the range of validity of the model with respect to a certain property. For example, in the resistor example from Section 2.1, the ambient temperature and the frequency and amplitude of the electrical signal are influencing factors. Therefore, certain frames may provide validity conditions using these influencing factors.

2.5 Experiments

The validation process relies on *experiments* to determine to what extent the provided model is valid to the real-world system. These experiments can be run in the real world (on the realized system) or in the virtual world (on a simulated model representing the system). An experiment places a model or a system in an *experimental context*. It also consists of an experimental setup, and the process to take the measurements; for details we refer to Denil *et al.* [8]. These experimental conditions must be precisely defined, such that the experiments can be accurately replicated. The results are used by the validation procedure to make the determination of validity for the provided model, including a measure of the model's fidelity with respect to the properties of the context.

In a basic form, these experiments may be input and output trajectories which the model is expected to match with some degree of tolerance. Additionally, the experiment needs to provide the influencing factors. For example, an experiment for the resistor may place a model of the resistor in an electrical network with an AC voltage source and a particular environment with fixed temperature (both influencing factors). It then can measure, as an output, the value property of interest, namely the constituent relation between (input) voltage and (output) current (as measured by a current sensor). Along with sufficient experimental conditions [13] (which includes, for example, the simulation platform which was used, the solver settings, etc.), and a tolerance, these experiments can determine the validity of the model with respect to the real system.

2.6 Validation Procedure

The *validation procedure* is an explicit activity stored within the validity frame to determine the fidelity of the provided model with respect to the real system, and to determine for each of the selected properties and the ranges in which the model is a valid abstraction of the system, considering the influencing factors. Along with performing the experiments, the validation procedure also has the responsibility of relating the output of each experiment stored in the validity frame to the model's results to determine the model's fidelity with respect to each property. That is, the validation procedure answers a) whether the model's behavior match (with some tolerance) the results of the experiments performed on the real-world system or high-fidelity model, and b) how well does the model perform with respect to each of the properties of the context captured by the validity frame. For this, a (validation) evaluation function needs to be defined.

A validation procedure for the resistor example could be as follows. Given a model of the resistor, it is placed in an electrical network with a variable (AC or DC) voltage source, an ambient temperature, and a current sensor. Then, a number of experiments are performed that excite the resistor with a number of different voltages, and the current flow is collected from the current sensor. The results of the real-world experiments would be compared to the model's results, and determine if the results were sufficiently close, within some tolerance. Finally, a robustness metric would determine how well the resistor model is valid with

respect to temperature and frequency, based on the experimental results from the model and which properties are involved in each experiment.

2.7 Validity Conditions

Finally, *validation conditions* are synthesized from the experiments that were performed and they are stored as part of the relation between a model and its frame. These conditions are useful when using frames such as in a model catalog, where these relations can be exploited to choose the most appropriate model based on the context it will be used in, and which properties are of interest. An example validity condition could define a range for parameters, input values, or influencing factors; within these ranges, the condition specifies that the model is valid (considering the validation process used).

For the resistor example from Section 2.1, validity conditions could be extracted from the range of experiments performed on the model. If we have run a series of experiments for a particular ambient temperature (for example, 50 degrees Centigrade) with a varying voltage value for the voltage source, we know for each experiment whether the model provides a response that is close to its real-world counterpart value. From these responses, a validity condition will then state that the model is valid for voltages up to x volts, where x is the last voltage value for which the model provided a response within the tolerance range (at 50 degrees Centigrade).

2.8 Capturing Validity

To summarize, we capture the validity of the simplest model for the resistor, namely Ohm's law in Figure 2a for one property, the constituent relation between voltage and current, and one influencing factor, ambient temperature. To do this, we take an experimental setup in the real world similar to Figure 1, where we hook up a single resistor to a DC voltage source and put it in an environment where we can control the ambient temperature. We then probe the circuit to measure the current flowing through it, which will be the result of the resistor's physical properties as it dissipates energy as heat. We then note this value for different values of voltage input; these measurements, for an ambient temperature of 50 degrees Centigrade are plotted in Figure 5 as red dots.

Taking these measurements, we can define a frame that specifies a validity condition that states that a model is valid with respect to these measured values if the simulated current value is equal to the measured value with a tolerance of 0.1 amperes. This allows us to put our model in an equivalent experimental setup (in the virtual world), such as can be seen in Figure 3a and repeat the experiments by setting the ambient temperature to a fixed value (50 degrees Centigrades, in this case) and by sweeping over a voltage range (0-100) with a certain step size, and then collecting the value that is observed in the current sensor. When we plot these values, we get the green line in Figure 5, a linear relation as specified by Ohm's law. For a particular range of voltages, the measured voltages almost follow this linear relation; after that other physical effects

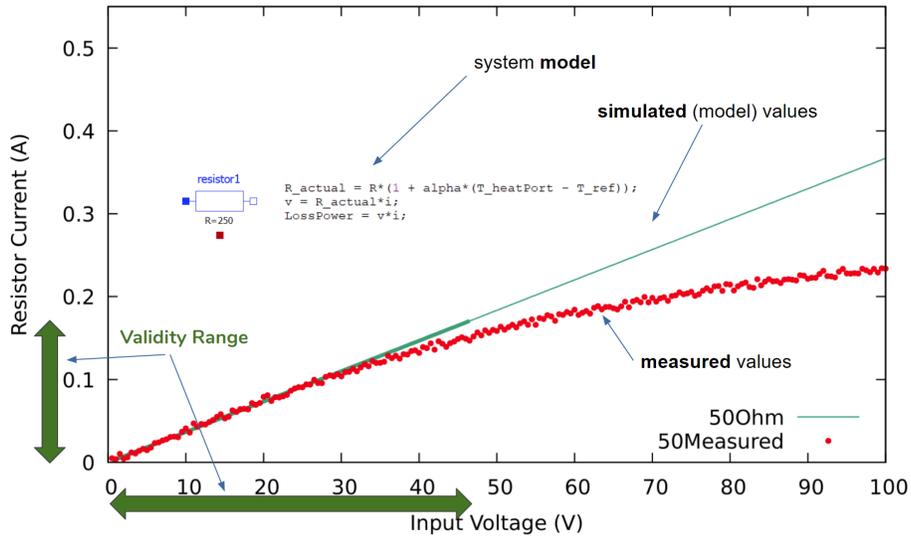


Fig. 5: Plot of validity range for Ohm’s law at 50 degrees Centigrade.

come into play. The range of voltages in which the simple model is valid, in this particular ambient temperature, is [0V, 48.1V] with a tolerance of 0.001A.

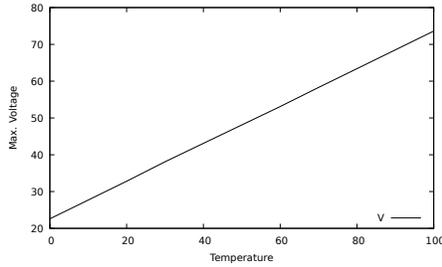


Fig. 6: Temperature Dependence.

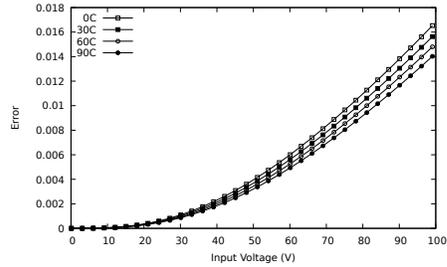


Fig. 7: Sensitivity.

This maximum voltage (for 50 degrees Centigrade, equal to 48.1V) depends on the ambient temperature. If we repeat these experiments for different ambient temperatures, that maximum voltage value will change as well. These results can also be recorded in the relation between model and validity frame, and can be seen in Figure 6. Furthermore, we can also store the sensitivity of the model (the error that is made in the simulation results with respect to the real system). This can be seen in Figure 7. This information can help us to see what the increase in error is if we go outside of the validity range of the model; if it does not increase

too much, it might mean there is some “stretch” on the validity range depending on the application and whether or not it is safety-critical.

3 Design-Space Exploration Using Validity Frames

In this section, we demonstrate the value of validity frames when used in the (automatic) domain-space exploration of a multi-component system. As a running example, we design a (passive) high-pass filter, which is implemented as an electrical network consisting of a number of electrical components, including a resistor. We demonstrate that performing the design-space exploration naively (without the use of validity frames) might result in a design that is invalid with respect to the behavior of the realized system, which means that the requirements are not satisfied. By using validity frames, we can avoid this costly error and ensure that the models of the components are used in a valid context; we focus on the resistor component.

3.1 Requirements

A (passive) high-pass filter is an electrical circuit that “lets through” electrical signals if their frequency is above a certain value. Signals below that cut-off frequency value are dampened (*i.e.*, only part of the magnitude of the signal passes). An ideal filter would dampen the signal completely if it is below the cut-off frequency (*i.e.*, the voltage would drop down to zero) and have a discontinuity at the cut-off frequency, letting through the whole signal.

This is, however, physically not possible to achieve, and instead, the signal will gradually be let through as the frequency increases. In the example, we will design a simple high-pass filter, consisting of only passive electrical components, whose behavior will be far from ideal.

Typically, the (required) behavior of a high-pass filter is expressed as a Bode plot. It shows for increasing frequency values how much the input electrical signal is attenuated by the circuit. This attenuation is expressed in decibels (dB), which is the logarithm of the ratio of input and output voltage, as follows:

$$value_{dB} = 20 * \log_{10}\left(\frac{V_{out}}{V_{in}}\right) \quad (4)$$

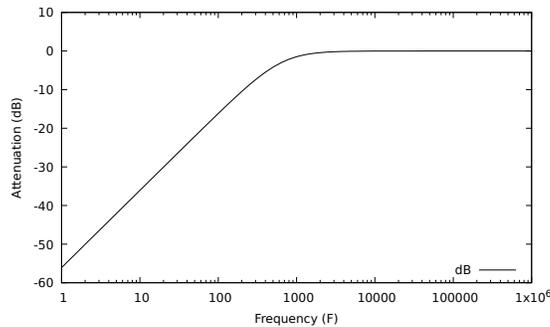


Fig. 8: Bode plot of the required behavior of the high-pass filter circuit.

A dB value of zero means that the signal is not attenuated; a negative value means the signal is attenuated (where lower values mean the signal is attenuated more - with a logarithmic scale). The Bode plot expressing the requirement of our example is shown in Figure 8:

- The cutoff frequency (where the dB value is -3) is approximately 460Hz.
- After the cutoff frequency, the dB value should be zero.
- The slope of the curve before the cutoff frequency is not a strict requirement for the process.
- The high-pass filter needs to operate for an AC voltage source whose frequency can range between 0Hz (DC voltage) and 1MHz; the amplitude of the AC voltage source can range between 0.1V and 100V.
- The circuit will be used in an environment of [20C, 50C] ambient temperature.

3.2 Initial Design

To design this system, we want to build an electrical circuit that, when connected to a voltage source for which we can vary the frequency, attenuates the signal as shown in Figure 8. For this, we set up an experimental setups in the virtual and the real world from which we can deduce the dB values of the circuit. We assume that these setups lead to results we can trust, and any error in the measured values is a result from a fault in the system or model.

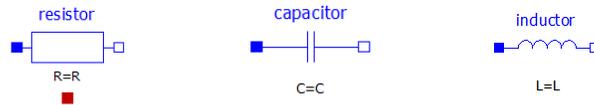


Fig. 9: Ideal physical models of electrical elements.

To create an initial design, we use a method closely related to what is described in [16]. We consider a library of component models, in this case consisting of a resistor, a capacitor, and an inductor. Their models are shown in Figure 9. We further restrict any design with domain-specific rules, including the fact that there needs to be exactly one element of each type, the electrical elements need to be connected in a legal way, and their parameter values are filled in.

From this design-space exploration, an electrical circuit is generated, shown in Figure 10, and its simulation result (as a Bode plot) is shown in Figure 11 alongside the experimental results of the realized system. As can be seen, the simulation results predict a (near-)perfect match between the behavior of the circuit and the required behavior. As such, we can realize the circuit by placing the physical components on a bread board, similar to Figure 1. However, when the experiment is run on this circuit, there is a significant discrepancy between these results. We can conclude that the virtual design does not result in a system that satisfies the required property; as such, the design model is invalid.

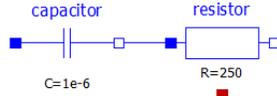


Fig. 10: The initial design of the high-pass filter circuit.

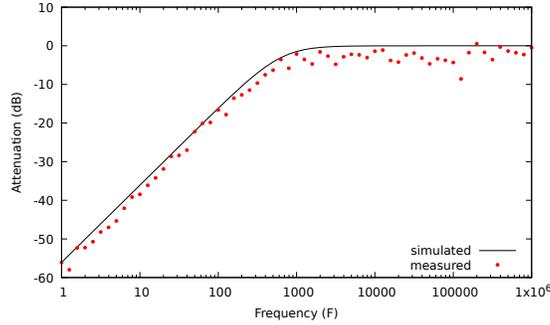


Fig. 11: The simulation results of the initial design and real-world experiment results.

3.3 The Resistor Frame

To deduce what has gone wrong during design-space exploration by focusing on one component of the electrical network, the resistor. We can ask whether, within the experimental context that the system is tested, the model we use is a valid abstraction of the resistor component.

We consider the validity range of the ideal physical model—a term introduced by Jan Broenink, see Chapter 2 of [6]—of the resistor for two influencing factors that are important for our application: ambient temperature (and power dissipation which may affect temperature) and frequency. The validity ranges (maximum voltage for which the model is valid) are plotted in Figure 6 and Figure 12. The validity range for the temperature influence is linear; the current running through the resistor has to be below $\frac{maxV}{R}$. For the frequency effects, the validity range decreases quickly up to 10MHz, and as such is almost unusable in that range. The current running through the resistor in the circuit shown in Figure 10 is higher than the value for which the model is valid; as such, the resistor model is used outside of its validity frame for the ambient temperature. Further, since the range of frequencies between 0Hz and 1MHz needs to be supported by the circuit, the model is also used outside of its validity frame for the frequency influencing factor.

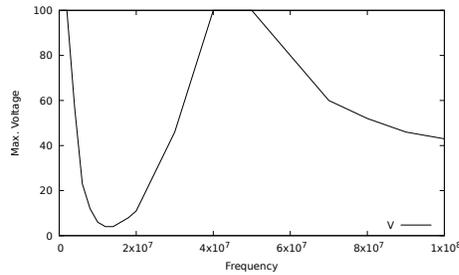


Fig. 12: Frequency Dependence.

3.4 Modified Design

To arrive at a modified design, we need to add further information to the design-space exploration process, considering the validity information of the resistor. When we do this, we have two options:

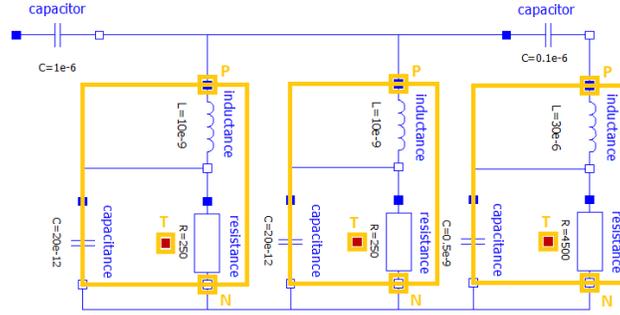


Fig. 13: The modified design of the high-pass filter circuit.

- We let the design-space exploration search for a design where the provided models have to stay within their validity range, which might mean we have to drop the constraint that states there should only be one “instance” of each component in the electrical network.
- We provide the design-space exploration with more detailed models of the component, considering the influencing factors. In Figure 2, several models are shown that provide higher-fidelity results by considering the effects of power and frequency environments.

The advantage of the first option is efficiency; we keep the models simple and ensure that they are used within a range that they are valid. However, this might lead to more complex designs with more components. The advantage in the second option is that the component is modeled in more details, which means that it can be used in more contexts, since it has a wider validity range. However, simulation performance might be negatively affected.

In Figure 13, the result of the design-space exploration, augmented with the extra constraints of the components staying within their validity range, is shown. The circuit is a second-order passive high pass filter with high fidelity resistor models. Figure 14 shows the simulation results of the modified design, as well as the result of experimenting with the real system when it is realized. These results are similar which means that now the model faithfully represents the realized system.

Once we have a valid design, we can also use it as a *design digital twin*, as explained in Section 1. The information in the validity frames of the components can be used to deploy runtime monitors. These monitors continuously run the experiments that are defined in the frame, but feed it with the input data from the real system. For the resistor, this could mean that we monitor the input of the voltage source and probe the output electrical signal of the resistor. We can then check whether these values are within the validity range of the model that was used to design the system; otherwise the predictions made by the design model will no longer be valid. By continuously monitoring whether components do not

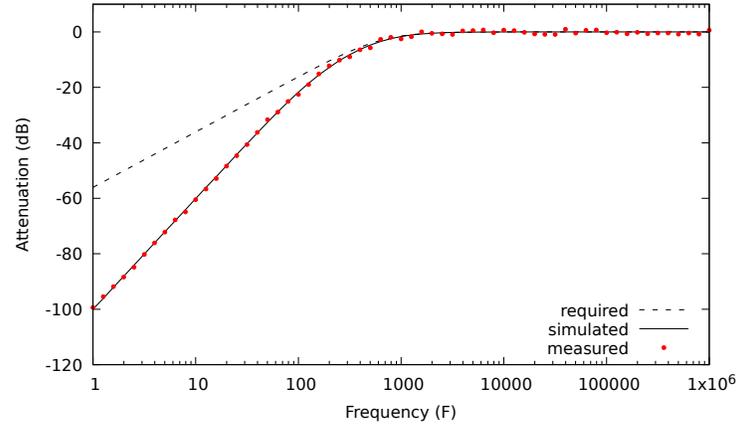


Fig. 14: The result of running the experiment on the realized high-pass filter circuit.

go outside the range of the models that were used to represent them in the design, we can already raise a warning or error when that range is exceeded, effectively creating an efficient (but limited) digital twin. Using sensitivity information, as demonstrated in Figure 7, a decision could be made whether the component is allowed to stay outside of its validity range for a while (because it is not very sensitive to its input), or whether the system should be brought to a failsafe mode as soon as possible.

4 Related Work

The notion of ‘frames’ in modeling and simulation is an old concept [28] where information about a model’s context is formalized. For example, *experimental frames* are a “specification of the conditions under which the system is observed or experimented with” [27]. These experimental frames are composed of three components: a) a *generator* of input traces to the model, b) an *acceptor* which ensures that the experimental conditions are met, and c) a *transducer* to examine the output signals. Formal definition of these experimental frames allows for (semi-)automated creation [18, 7], use in (semi-)automated trade-off analysis [12], and enabling frame-enabled hierarchical decomposition of the system [19].

The work of Traoré and Muzy generalize the experimental frames concept to instead define multiple frame utilizations [22]. An important aspect of their work is to relate a *system* and its *context*, a *model* and its *frame*, and a *simulator* and its *experimenter*. Here it is clear that the frame of a model represents the context of the original system which it captures. Further work by Foures *et al.* examines the generator, acceptor, and transducer components of the experimental frame to reason about how their inputs and output can be used to determine constraints

on the model [10, 11]. For example, the generator can be restricted in what events will be sent as input to the model, thereby structuring the context of the model.

Schmidt *et al.* approach our validity frame concept, as they define an experimental test framework for automatically evaluating fidelity of models with respect to complicated cyber-physical systems [21]. Their framework operates within the MATLAB/Simulink environment to implement test control and admissible model parameter configurations. The structure of modeling frames was addressed in work by Klikovits *et al.* [14], where a frame is structured as three components: a) a *modeling activity* with inputs, outputs, and a process, b) a *context* with objectives, assumptions, and constraints, and c) zero or more *sub-frames*. They emphasize the processes stored within frames, which can be used to (semi-)automatically perform the frame activity.

The work of Denil *et al.* specifically addresses validity frames, with an emphasis on capturing information to reproduce the results of experiments on the real system in the virtual world [8]. That work defines two uses for validity frames as *testing* the validity of the model against a real system, and on *calibrating* the model such that the model is valid. Recent work on validity frames by Van Acker *et al.* focuses on validity frames for models-of-the-physics in embedded domains [23]. These validity frames have two purposes: a) to assess the validity range of reduced-order models such as a neural net, and b) quantify the fidelity of these models versus the high-fidelity model. In that work, the validity frame is divided into (1) the *meta-data* about the model and its parameters, which includes information on the model solver and parameter datatype; (2) the *operational* part, which includes information about the equivalence ranges for parameters; and (3) the *process* part of the validity frame defines the validity and calibration processes as defined in [8].

5 Conclusion and Future Work

In this paper, we show how validity frames for individual components can be constructed based on the simulation results of their models and the experimental data obtained from real-world experiments. We demonstrate this by creating a validity frame for an electrical resistor, for two influencing factors: ambient temperature and frequency, and one property: the constituent relation between voltage and current. We use the frame information in an (automated) design-space exploration process and show that this information is crucial to arrive at valid design models and ultimately valid systems. We demonstrate this on the design of a (passive) high-pass filter, an example of a multi-component electrical circuit. We see several directions for future work. The validity frame concept can be investigated further in the optimization of the (design) digital twin, and its role explored in fault localization and (runtime) adaptivity. Further, we envision the creation of a model repository and framework which embraces the validity frame concept fully; it would not allow the creation and use of models without reasoning about its range of validity.

Acknowledgments

This work was partially funded by Flanders Make vzw, the strategic research centre for the Flemish manufacturing industry and by the University of Antwerp's Industrial Research Fund Strategic Basic Research (IOF-SBO).

References

1. Van der Auweraer, H., Anthonis, J., De Bruyne, S., Leuridan, J.: Virtual engineering at work: the challenges for designing mechatronic products. *Engineering with Computers* **29**(3), 389–408 (2013)
2. Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Ničković, D., Sankaranarayanan, S.: Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In: *Lectures on Runtime Verification*, pp. 135–175. Springer (2018)
3. Benveniste, A., Caillaud, B., Nickovic, D., Passerone, R., Raclet, J.B., Reinke-meier, P., Sangiovanni-Vincentelli, A., Damm, W., Henzinger, T.A., Larsen, K.G., et al.: Contracts for system design. *Foundations and Trends® in Electronic Design Automation* **12**(2-3), 124–400 (2018)
4. Breunese, A., Broenink, J.F., Top, J., Akkermans, J.: Libraries of reusable models: theory and application. *Simulation* **71**(1), 7–22 (1998)
5. Canedo, A., Richter, J.H.: Architectural design space exploration of cyber-physical systems using the functional modeling compiler. *Procedia CIRP* **21**, 46–51 (2014)
6. Carreira, P., Amaral, V., Vangheluwe, H.: *Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems*. Springer (2020)
7. Daum, T., Sargent, R.G.: Experimental frames in a modern modeling and simulation system. *Iie Transactions* **33**(3), 181–192 (2001)
8. Denil, J., Klikovits, S., Mosterman, P.J., Vallecillo, A., Vangheluwe, H.: The experiment model and validity frame in M&S. In: *Proceedings of the Symposium on Theory of Modeling & Simulation*. pp. 1–12 (2017)
9. Ewald, R., Uhrmacher, A.M.: Sessl: A domain-specific language for simulation experiments. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **24**(2), 1–25 (2014)
10. Foures, D., Albert, V., Nketsa, A.: Formal compatibility of experimental frame concept and finite and deterministic devs model. In: *International Conference on Modeling Optimization SIMulation (MOSIM 2012)*, pages–10. p. 94 (2012)
11. Foures, D., Albert, V., Nketsa, A.: Simulation validation using the compatibility between simulation model and experimental frame (2013)
12. Hu, J., Rozenblit, J.W.: Towards automatic generation of experimental frames in simulation-based system design. In: *AI and Simulation Conference*. pp. 1–6 (1988)
13. Ivie, P., Thain, D.: Reproducibility in scientific computing. *ACM Computing Surveys (CSUR)* **51**(3), 1–36 (2018)
14. Klikovits, S., Denil, J., Muzy, A., Salay, R.: Modeling frames. In: *CEUR workshop proceedings*. pp. 315–320 (2017)
15. Madni, A.M., Madni, C.C., Lucero, S.D.: Leveraging digital twin technology in model-based systems engineering. *Systems* **7**(1), 7 (2019)
16. Meyers, B., Denil, J., Vanherpen, K., Vangheluwe, H.: Enabling design-space exploration for domain-specific modelling. In: *Proceedings of the Model-driven Approaches for Simulation Engineering Symposium*. p. 5. Society for Computer Simulation International (2018)

17. Ramos, A.L., Ferreira, J.V., Barceló, J.: Model-based systems engineering: An emerging approach for modern systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(1), 101–111 (2011)
18. Rozenblit, J.W.: Exp—a software tool for experimental frame specification in discrete event modelling and simulation. In: *Proceedings of the Summer Computer Simulation Conference*. pp. 967–971 (1984)
19. Rozenblit, J.W.: Experimental frame specification methodology for hierarchical simulation modeling. *International Journal Of General System* **19**(3), 317–336 (1991)
20. dos Santos, C.A.R., Saleh, A.H., Schrijvers, T., Nicolai, M.: Condense: Contract based design synthesis. In: *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*. pp. 250–260. IEEE (2019)
21. Schmidt, A., Durak, U., Pawletta, T.: Model-based testing methodology using system entity structures for MATLAB/Simulink models. *Simulation* **92**(8), 729–746 (2016)
22. Traoré, M.K., Muzy, A.: Capturing the dual relationship between simulation models and their context. *Simulation Modelling Practice and Theory* **14**(2), 126–142 (2006)
23. Van Acker, B., De Meulenaere, P., Denil, J., Durodie, Y., Van Bellinghen, A., Vanstechelman, K.: Valid (re-)use of models-of-the-physics in cyber-physical systems using validity frames. In: *2019 Spring Simulation Conference (SpringSim)*. pp. 1–12 (April 2019). <https://doi.org/10.23919/SpringSim.2019.8732858>
24. Van Acker, B., Denil, J., De Meulenaere, P., Aelvoet, B., Mahieu, D., Van Den Oudenhoven, J.: Generation of test strategies for model-based functional safety testing using an artifact-centric approach. In: *Proceedings of MODELS 2018 Workshops co-located with ACM/IEEE 21st International Conference on Model Driven Engineering Languages and Systems (MODELS 2018) Copenhagen, Denmark, October, 14, 2018/Hebig, Regina [edit.]*. pp. 563–569 (2018)
25. Vangheluwe, H.: *Multi-formalism Modelling and Simulation*. D.Sc. dissertation, Faculty of Science, Ghent University (2000)
26. Weyns, D.: *Software engineering of self-adaptive systems: an organised tour and future challenges*. Chapter in *Handbook of Software Engineering* (2017)
27. Zeigler, B.P., Kim, T.G., Praehofer, H.: *Theory of modeling and simulation*. Academic Press (2000)
28. Zeigler, B. P.: *Multifaceted modelling and discrete event simulation*. London, Academic Press (1984)