Validation and Uncertainty in Model-Based Design Space Exploration - an Experience Report

Yon Vanommeslaeghe
yon.vanommeslaeghe@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

David Ceulemans
david.ceulemans@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

Bert van Acker
bert.vanacker@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

Joachim Denil
joachim.denil@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

Stijn Derammelaere
stijn.derammelaere@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

Paul De Meulenaere
paul.demeulenaere@uantwerpen.be
Cosys-Lab
University of Antwerp
Antwerp, Belgium
Flanders Make
Belgium

ABSTRACT

Model-based systems engineering (MBSE) techniques can help manage the growing complexity in the design and development of cyber-physical systems, and can even allow for the optimization of a system under design in simulation. However, models are always an abstraction of the real-world systems they represent. This introduces uncertainty at the model level, which affects the validity of simulation results, and thus also the results of the optimization. This, together with variations in real-world system parameters, significantly complicates the validation of simulation and optimization results. In this experience report, we first use a descriptive process model to describe our efforts to validate the results of a model-based design space exploration (DSE) process given this uncertainty. After this, we discuss lessons learned and insights gained, and identify future challenges. We present a possible prescriptive process model for future validation efforts, which specifically takes into account uncertainty.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**; • **Computing methodologies → Model verification and validation**; • **General and reference → Validation**.

KEYWORDS

cyber-physical systems, model-based systems engineering, design space exploration, validation, uncertainty

MPM4CPS'22, October 23–28, 2022, Montreal, Canada

ACM ISBN 978-1-4503-9467-3/22/10…$15.00
https://doi.org/10.1145/3550356.3561581

ACM Reference Format:
Yon Vanommeslaeghe, David Ceulemans, Bert van Acker, Joachim Denil, Stijn Derammelaere, and Paul De Meulenaere. 2022. Validation and Uncertainty in Model-Based Design Space Exploration - an Experience Report. In *ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS '22 Companion), October 23–28, 2022, Montreal, QC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3550356.3561581

1 INTRODUCTION

Cyber-physical systems are becoming increasingly complex, with both the dependencies between system components and the complexity of the components themselves increasing rapidly. As these systems become more complex, so does their design and development. Here, model-based systems engineering (MBSE) techniques can help manage this complexity. Such techniques provide a safer, more efficient, and less costly way of developing such systems, compared to analyzing and evaluating each component during the construction and integration phase [3]. They also enable model-based design space exploration (DSE), whereby the system under design can be optimized in simulation, possibly even before its actual implementation or construction. However, models are always an abstraction of the real-world system they represent, with certain details or effects being omitted or otherwise simplified. This introduces uncertainty at the model level, which affects the validity of the simulation results, and by extent also of the DSE results. This complicates validation efforts, as this can cause discrepancies between modelled and observed system behavior. Additionally, variations in real-world system parameters, e.g. due to variations in the production process or measurement uncertainty, can cause further discrepancies or can even cause the system behavior to divergence from simulation results.

In previous work [11], we demonstrated an ontology-based approach to DSE, where we considered the optimal embedded deployment of an advanced control algorithm for brushless DC (BLDC)

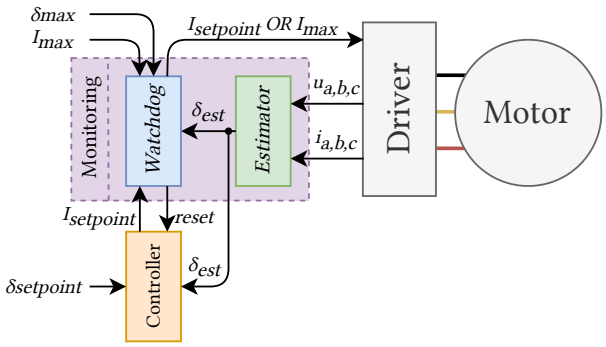

Figure 1: Schematic overview of the load angle controller [10]

motors. Using this approach, we were able to determine good, non-obvious design candidates for embedded deployment of this algorithm. However, in the DSE, the determined design candidates were only evaluated in simulation using models. As such, it remained unknown how well they would actually perform in a real-world setup. In this experience report, we present our efforts to validate these results on a real-world setup, along the way encountering difficulties regarding uncertainty. We start by deploying the identified design candidates on an embedded platform and evaluating their performance on a real-world test setup. After this, we further validate our results and the simulation models themselves by repeating these evaluations for multiple other (non-optimal) design candidates. Then, to quantify the obtained results, we investigate the impact of uncertainty regarding different model parameters on the predicted performance of the design candidates. These different steps are all captured in a *descriptive* process model. After this, we discuss what this impact means in retrospect for the DSE results and what this may mean in the future, based on gained insights and lessons learned. For example, how statistical information may be used to inform the optimization problem definition. This discussion is accompanied by a possible *prescriptive* process model.

The rest of this paper is structured as follows. First, in Section 2, we introduce the example case used in previous work and the current paper. Next, in Section 3, we describe the followed validation process (*descriptive*). Lastly, in Section 4, we discuss the results of the *descriptive* process and the insights gained, and present a possible *prescriptive* process, taking into account lessons learned.

2 EXAMPLE USE CASE

In the current paper, as in previous work, we consider the embedded deployment of an advanced, energy-efficient load angle control system for BLDC motors, based on the algorithm proposed by De Viaene *et al.* [2]. In short, this algorithm consists of two main parts, mapped to different tasks on the embedded platform: a monitoring part, and a control part, as shown in Figure 1. The monitor part contains an algorithm that estimates the load angle, i.e. the angle between the stator current vector and the magnetic rotor field of the motor. This estimated load angle serves as feedback to the controller, which will attempt to drive the load angle to a predefined setpoint by varying the motor current amplitude. Here, a larger load angle increases energy efficiency but reduces the margin for error, increasing the chance of loss of synchronism. Therefore, the monitor contains a watchdog that guards against loss of synchronism by overruling the controller when the estimated load angle becomes too large. This algorithm was developed using a MBSE approach and was implemented in MATLAB® Simulink®.

What complicates matters is that there is an optimization problem associated with the embedded deployment of this algorithm. Indeed, we want to find the optimal combination of *estimator period*, *control period*, and *load angle setpoint*, which will result in the best possible system performance, i.e. the lowest energy consumption, while maintaining system stability and ensuring schedulability on the embedded platform. To determine this optimal configuration, we make use of (model-based) DSE techniques. Here, a model of the control algorithm is combined with a plant model, i.e. a model of the physical system it will control, to predict the performance of different algorithm configurations in simulation. The specific DSE approach we used is discussed in the following subsection.

3 VALIDATION PROCESS (DESCRIPTIVE)

In this section, we present the process we followed to validate the performance of the previously determined design candidates on a real-world test setup. To model this process and the different formalisms and transformations used therein, we make use of the formalism transformation graph + process model (FTG+PM) formalism [4]. The FTG part is shown in Figure 3, while the PM is shown in Figure 4. Note that we add a small extension to the standard FTG+PM formalism to allow the inclusion of real-world entities. The major sections of the PM are further described in the following subsections. The PM shown here is a *descriptive* process model, i.e. it describes the process steps in the order they were performed. As such, the shown process is rather inefficient. In Section 4, we discuss how, in retrospect, this process could have been improved and show a possible *prescriptive* process.

3.1 Experimental Setup

Although many definitions exist for the term *validation*, we regard *model validation* as *the assessment of model accuracy by way of comparison of simulation results with experimental measurements*, as defined by Oberkampf and Roy [7]. As such, a real-world test setup is needed to perform the required measurements described in the following subsections. It consists of four main parts, as shown in Figure 2:

- A **brushless DC motor** to be controlled by the load angle control algorithm. A high-resolution **encoder** attached to the motor shaft allows us to measure the rotor angle (θ_{rotor}) to determine the ground-truth load angle (δ_{ref}) for validation purposes.
- The **embedded platform** on which the load angle control algorithm is to be deployed, in this case a MicroZed™ 7020, paired with the required **power electronics** to drive the BLDC motor.
- A **permanent magnet synchronous motor (PMSM)** paired with a **Siemens® SINAMICS®** drive. The PMSM is operated in torque control mode, which allows us to apply a precise load torque to the BLDC motor in a controlled and repeatable manner.
- A **Speedgoat educational real-time target (RTT) machine** is used both to sequence experiments and to log data over the course of each experiment. During an experiment, it is responsible for

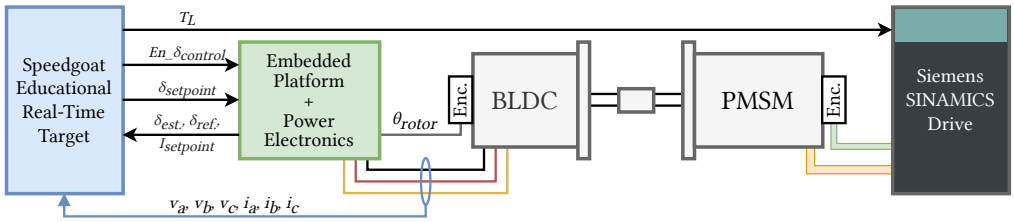

Figure 2: Simplified diagram of the real-world test setup.

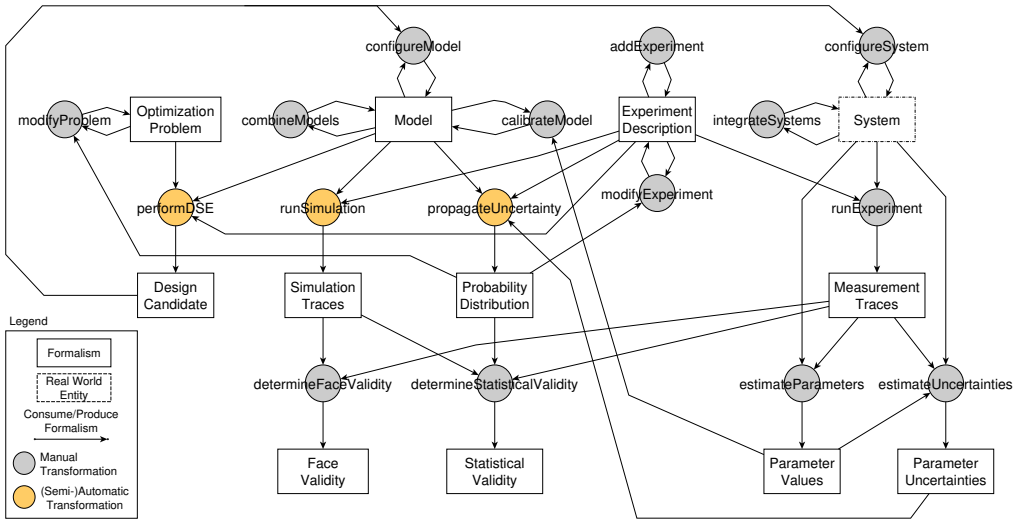

Figure 3: Formalism transformation graph showing the formalisms and transformations used in the validation process.

configuring ($\delta_{setpoint}$) and enabling ($En\_\delta_{control}$) the load angle controller, and commanding the SINAMICS® drive to apply a specific load torque (T_L) at a specific time. At the same time, it collects information about the estimated load angle (δ_{est}), the ground-truth load angle (δ_{ref}), and the requested motor current ($I_{setpoint}$). It also records the motor phase voltages (v_a, v_b, v_c) and the resulting phase currents (i_a, i_b, i_c) for later analysis.

This setup allows us to (semi-)automatically run pre-defined experiments. It is used in full, or in part, in all experiments, characterization, and validation efforts described in the current paper. Here, the **brushless DC motor**, **power electronics**, and the **PMSM** with its **SINAMICS®** drive are considered the *plant* (Figure 4), while the **embedded platform** is the *(configured) controller*. As such, these parts together become the *configured system*.

3.2 Plant Model Calibration

Our previous work was performed using experiments in-silico. We used a model of the load angle control system, together with a plant model, to evaluate design candidates in a (simulated) test scenario. The used plant model was constructed using parameter values from datasheets and other theoretical values. During initial testing on the real-world test setup, we noticed discrepancies between the observed behavior of the real system and the simulated behavior of the modeled system. Such discrepancies could originate from a poorly calibrated plant model, which does not match its real-world counterpart. As such, it was deemed necessary to first calibrate the plant model. We identified and calibrated six important parameters using a series of experiments (shown as a loop in Figure 4):

(1) The **motor velocity constant (K_v)** specifies the ratio between the motor's unloaded rotational speed and the voltage across its windings. As this constant is closely related to the back electromotive force (back EMF), we experimentally determined K_v by measuring the back EMF voltage at different rotational speeds.
(2) The **motor torque constant (K_T)** specifies the ratio between the stator current and the produced torque. K_T can be calculated from K_v. As such, we calculated this from our experimentally determined K_v.
(3) The **motor winding resistance (R)** was determined using a four point resistivity measurement and later verified using the parameter estimation capabilities of the vedder electronic speed controller (VESC) [12].
(4) The **motor winding impedance (L)** was similarly measured in a lab setup and later verified using the VESC.
(5) The **moment of inertia (J) of the system**, i.e. both motor rotors with encoders and shaft couplings, was determined using the parameter estimation capabilities of the Siemens® SINAMICS® drive used to control the PMSM.

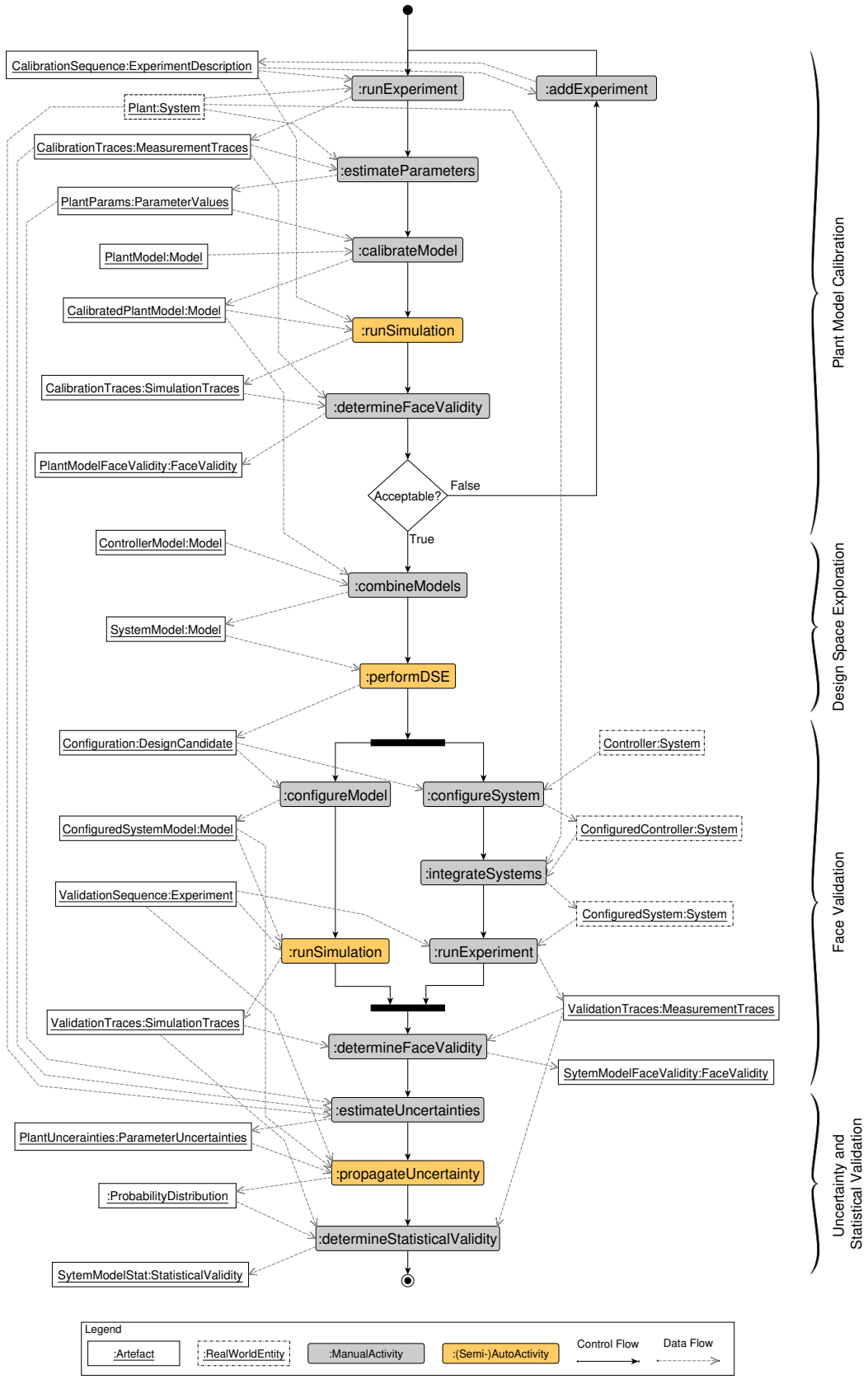

Figure 4: Descriptive process model of the followed model calibration, design space exploration and system validation process.

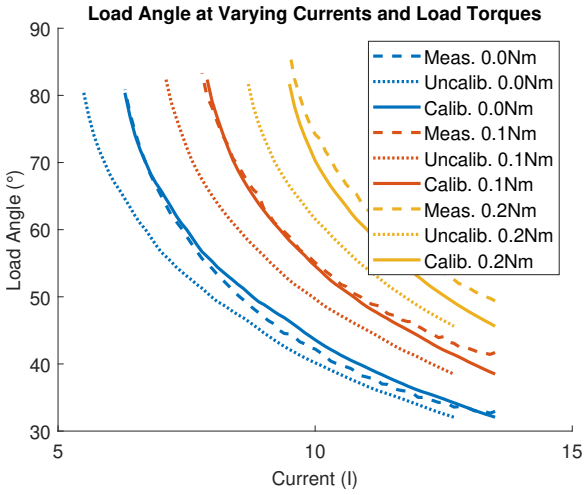

Figure 5: Load angle at varying current levels under different loads on the physical system (meas.) and in simulation, using both the uncalibrated (uncalib.) and calibrated model (calib.).

(6) The **friction inherent in the system** due to, e.g. bearing friction, viscous friction, etc., Characterizing this is less straightforward than the previous parameters. An initial estimate was made by running the motor up and down its operating speed range and recording the required current, and thus torque (using K_T), at each speed. Using this data, we were able to approximate the friction in function of the rotational speed using the quadratic function $T_{friction} = p_1 \cdot \omega^2 + p_2 \cdot \omega + p_3$. Note that this does ignore breakaway torque. While this is not important for our purposes, this does affect the validity range of the model.

Most of the listed parameters (everything except friction) are generally also provided by the manufacturer. While most of our measurements resulted in values close to those listed in the datasheet (within ~1-2%), the measured impedance was an order of magnitude lower than stated by the manufacturer. As multiple independent measurements using different techniques provided similar results, we do assume these to be correct. This has a major impact on system behavior. First, a significantly lower impedance means the current in the motor windings rises and falls more quickly, leading to worse than expected performance of the current control system. This in turn affected the precision of the estimated load angle. As such, modifications were made to the implementation of the system to mitigate these effects. Additionally, the load angle estimator requires the values of R and L to be known. A significant mismatch with the actual resistance and impedance of the motor windings affects the accuracy of the estimate and thus the overall behavior of the load angle control system.

With these changes and other small modifications to the model, the behavior of the system in simulation much more closely matches the behavior of the real system during our test scenario. This is illustrated in Figure 5, which shows the reference load angle at different motor current amplitudes under different loads, both in simulation and measured on the real-world test setup (meas.). Simulation results are shown for both the uncalibrated (uncalib.) and calibrated (calib.) models. Note that the curves do not match exactly between the simulation and the real world. This is likely due to the fact that the model is still an abstraction of reality. While this does further limit the validity of the model, the shown simulation results are sufficiently close to reality for our purposes as we are mostly concerned with the maximum load angle, specifically with 0-0.1Nm external load torque. However, as the model has changed, the results of the previously performed design space exploration may no longer be valid. As such, we repeated the DSE process using the updated models. This is explained in more detail in the following section.

3.3 Design Space Exploration

As the model has changed, the results of the design space exploration presented in previous work may no longer be valid. Additionally, some modifications were made to the implementation to reduce some undesired effects, further invalidating the previous DSE results. As such, we reuse the DSE workflow presented in previous work [11] to once again determine two good design candidates for deployment on respectively a single- and dual-core embedded platform. Each candidate consists of a specified period for the *monitor* and *control* parts of the algorithm, together with an optimal *load angle setpoint*. Using this approach, we arrive at the following configurations:

(a) **Single-Core Candidate**
Monitor period: 24 µs
Control period: 40 µs
Load angle setpoint: 80°

(b) **Dual-Core Candidate**
Monitor period: 12 µs
Control period: 36 µs
Load angle setpoint: 85°

For reference, we also exhaustively evaluated the entire design space. All schedulable configurations are shown in Figure 6, with the single- and dual-core candidates listed above highlighted. This figure shows the corresponding optimal load angle setpoint for each schedulable combination of monitor and control period.

3.4 Face Validation

To validate the performance of the proposed design candidates, we deploy them on the embedded platform and evaluate them on the physical setup. To evaluate their performance, we record the motor power consumption during a predefined validation sequence. In this part of the process, we only considered *face validity*, i.e. does the model seem reasonable to an expert [8], which is subjective.

3.4.1 Validation Sequence. The used validation sequence is illustrated in Figure 7, which shows (a) the load torque, (b) the load angle, and (c) the motor current amplitude during one validation run. The validation sequence starts with the motor running at a predefined speed, with no external load. After 15 s however, the load is abruptly increased to 0.1 Nm to exercise the watchdog. In the figure, the watchdog is active from 15 s to 25 s, as can be seen by the increased motor current. The load torque of 0.1 Nm is maintained until the end of the sequence, at 35 s. Performing this test for a range of load angle setpoints allows us to find the setpoint with

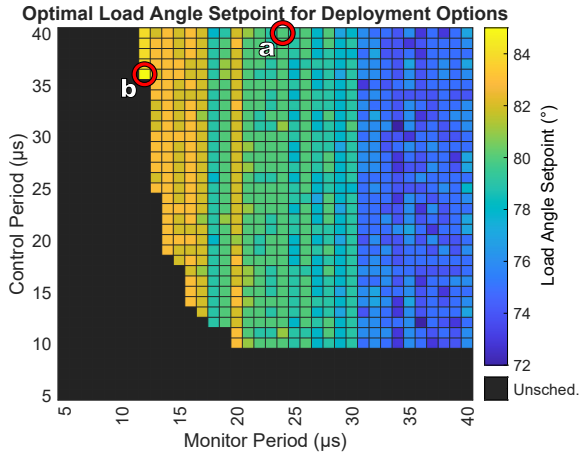

Figure 6: Optimal load angle setpoint for all schedulable configurations.

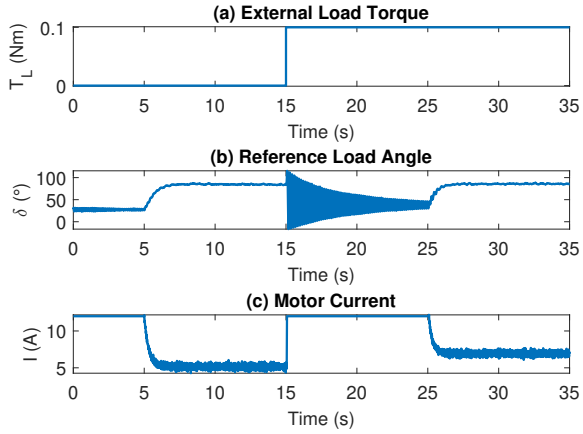

Figure 7: Measurement data showing the externally applied load, load angle, and motor current during a test sequence.

the lowest energy consumption, i.e. the highest possible load angle before loss of synchronism occurs, for each specific design candidate. Overall, better candidates will be able to maintain a larger load angle, resulting in lower power consumption.

3.4.2 Determine Face Validity. We use this approach to validate the single- and dual-core solution determined using the DSE approach described in the previous subsection. For each solution, we compare the actual optimal load angle setpoint to one predicted in simulation. For the single-core solution, this results in an optimal load angle of 77 −78° on the real-world system compared to 80° in simulation. Similarly, for the dual-core solution, we find an optimal load angle of 81 −83° instead of the predicted 85°. In short, both solutions perform slightly worse on the real-world setup when compared to the simulation. Overall, this is not surprising, as we know the simulation models used are always an abstraction of their real-world counterparts [13]. They might not capture all effects that can affect system performance. As such, a difference between the real-world and simulation results is to be expected.

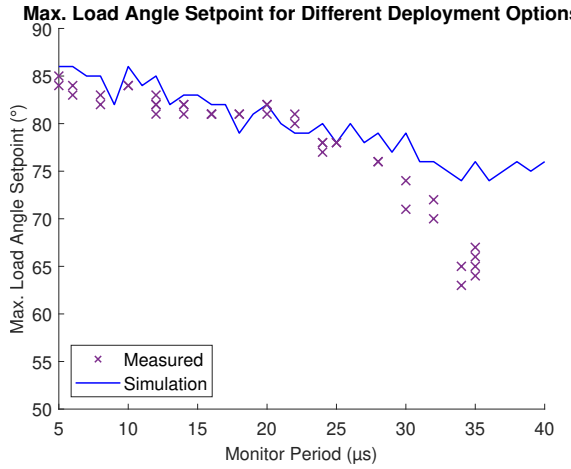

Figure 8: Predicted performance of different algorithm configurations compared to their real-world performance.

To further validate our simulation results, we additionally evaluate multiple (non-optimal) design candidates on the physical setup. Here, we take a row of the design space seen in Figure 6 at control period 36 µs and evaluate design candidates for a subset of all possible estimator periods. We choose a row, as the estimator period appears to have a much larger impact on system performance than the control period. Nevertheless, this was first confirmed by evaluating a few configurations within a column, resulting in the same observed performance. First, however, the precision of the measurements was determined by repeating the same measurements ten times each for two different configurations, resulting in a repeatability of 1°. To gauge the influence of temperature changes, we also repeated the same measurements multiple times over a longer period of time (3 −5 h), which showed that once the motor had warmed up for around an hour, the same repeatability of 1° could be achieved over multiple measurements. As such, the motor was allowed to warm up for more than one hour at the start of each measurement session. Nevertheless, to minimize the influence of environmental factors, measurements were performed out of order to avoid introducing a potentially misleading trend in the results.

After this, each candidate from the chosen subset was evaluated at least twice, the results of which can be seen in Figure 8, which shows the measurement results together with the predicted performance from the simulation runs. Overall, the measurement results confirm the expected trend, with shorter monitor periods resulting in better system performance (larger maximum load angle). Additionally, we see that most of the measurements are within 2 −3° of the predicted value and are especially close for estimator periods around 20 −25 µs. However, we do see that the measurements start to diverge from the simulation at around 28 −30 µs, possibly some effect(s) not considered in the model start to have a larger impact here. As such, the model might no longer be valid from this point on. In short, the model *appears* valid up to around 28 µs. Nevertheless, without a relevant frame of reference, it is hard to quantify how good or bad these results actually are. As such, in

the following subsection, we investigate the impact of potential sources of uncertainty on the predicted performance.

3.5 Uncertainty and Statistical Validation

In the previous section, we compared the performance of the deployed load angle control algorithm in a real-world test setup to the expected performance predicted using simulations. However, while we went to great lengths to calibrate our model to obtain representative simulation results (Subsection 3.2), there is still uncertainty associated with many of the calibrated model parameters. This uncertainty can stem from the methods used to determine these parameters, which may not be entirely accurate, but certain parameters may also change depending on the operating conditions of the system, e.g. because they are temperature dependent. In the current section, we seek to determine the impact of this uncertainty on the simulation results and, thus, in retrospect, what this means for our previously obtained DSE results and the DSE workflow we used to obtain them.

3.5.1 Estimating Uncertainties. First, we identified potential sources of uncertainty that are likely to impact the accuracy of the model and thus the reliability of the simulation results. For each item, a realistic range of possible values was identified. This resulted in six potential sources of uncertainty:

(1) **Motor winding resistance (R):** While we have taken care to measure the resistance as accurately as possible, this was done so in a static setup, i.e. without the motor running. However, the resistance can change depending on the operating point of the motor (rotational speed and phase current). From measurements performed on a similar setup [1], we expect the phase resistance to increase by about 78% to 109% in our test scenario. As such, we consider the following likely values for R: $[1, 1.78, 2.09] \cdot R_{measured}$.
(2) **Motor winding impedance (L):** Similar to the winding resistance, the impedance can change depending on the operating point. Based on the same measurements, we estimate that the impedance may increase by 9% to 22% in our test scenario. As such, we consider the following likely values for L: $[1, 1.09, 1.22] \cdot L_{measured}$.
(3) **Friction**: As mentioned in Subsection 3.2, the friction in function of the rotational speed of the motor was approximated using a quadratic function. However, in reality, the friction is also temperature dependent, with friction decreasing over time as the motor is running and heats up. The measurements shown in Figure 5 were originally performed when the motor had already been running for some time. To gain insight into the extent of the impact of the operating temperature, the no-load measurement was repeated both before a longer measurement campaign, i.e. with the motor at room temperature, and after said measurement campaign, i.e. after several hours of operation. The results of these measurements are shown in Figure 9, which shows the original measurements (warm) of the load angle in function of the stator current levels together with these new measurements (cold and hot). The simulation results using the calibrated model are also shown for reference. From these results, we determined that we could approximate the friction over the measured operating range by changing p_3 in the original fit, with an increase of ~5.1% representing the "cold" situation and a decrease of ~10.2% representing the "hot" situation. As such, we consider the following values for p_3': $[0.898, 1, 1.051] \cdot p_3$.

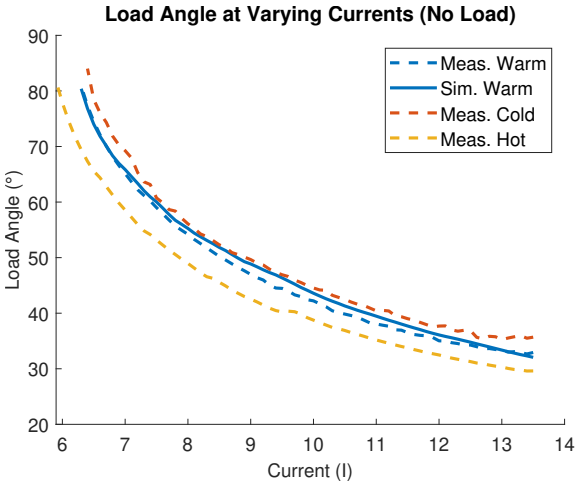

Figure 9: Load angle at varying current levels at different operating temperatures.

(4) **Additive measurement noise on the voltage measurements:** The voltage range supported by the analog inputs of the used embedded platform is rather small (0-1V). While external input conditioning circuitry does allow us to measure the phase voltages required by the load angle estimator, this limited range does make these inputs susceptible to electrical noise. During testing, we observed additive measurement noise with an amplitude of around $35mV$ at the analog inputs of the embedded platform. We expect this noise can impact the precision of the load angle estimate and thus the behavior of the system. To replicate this in simulation, we add additive white Gaussian noise (AWGN) to the voltage measurements, with a variance in the following range: $[0, 0.035^2]V$.
(5) **Additive measurement noise on the current measurements:** Similar to the voltage measurements, noise on the current measurements may affect the performance of the system. As such, we also add AWGN to the current measurements in simulation, with a variance in the range $[0, 0.035^2]V$.
(6) **Ringing on the voltage measurements:** In addition to noise on the voltage measurements, we observed some ringing at sharp transitions. As such, we have added the capability to emulate this behavior in our model so the potential impact of this effect can be evaluated in simulation. For the ringing, we consider possible values of $[0, 1]$, where 0 means ringing is disabled, and 1 means ringing is enabled.

3.5.2 Propagating Uncertainties. To propagate the previously identified uncertainties and to investigate how and to what extent they impact the simulation results, we perform Monte Carlo simulations. Here, a subset of the previously performed simulations is repeated with varying configurations for the previously mentioned parameters. We do this both by uniform sampling within the identified

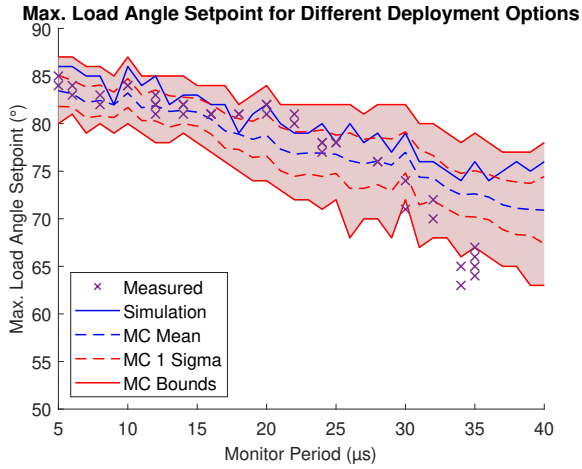

Figure 10: Confidence bounds determined using the Monte Carlo simulations together with the measurement results.

ranges for a large number of samples (16 384) to get an idea of the resulting distribution of simulation results and by simulating every combination of identified potential values (7 776 combinations) to increase the likelihood of finding the worst-case bounds on these results. To evaluate a configuration, we perform a parameter sweep over the load angle setpoint (46 possible values). As such, $(16\,384 + 7\,776) \cdot 46 = 1\,111\,360$ simulations were required. For reference, these simulations took over 80 hours when parallelized over 382 cores of a high-performance computing (HPC) cluster, i.e. requiring nearly 31 thousand CPU hours in total. Figure 10 shows the results of the Monte Carlo simulations together with the simulations and measurement results from Figure 8. For the Monte Carlo simulations, it shows the mean result (blue dashed line) with the corresponding one sigma bounds (red dashed lines), as well as the extreme values, i.e. the minima and maxima, (solid red lines).

3.5.3 Statistical Validity. Using the information gained in the previous subsection, we can further quantify the results obtained in Subsection 3.4. Indeed, this allows us to determine the *statistical validity*, i.e. *to what extent* the (measurement) results correspond to those obtained in another context (simulation). In Figure 10, we see that most of our measurements are close to the mean predicted values (within ~1-2). Indeed, for estimator periods of 16µs and below, the measurements are closer to the mean from the Monte Carlo simulations than to the prediction obtained using the initial simulations. Overall, most measurements fall within the one sigma confidence bounds. It's mostly for estimator periods of 30µs or more that we start seeing larger deviations from the mean, with measurements falling outside the one sigma bound and even outside the extreme values observed in the Monte Carlo simulations. As such, we conclude that for estimator periods smaller than 30µs , our system model represents the real system accurately enough for our purpose and can be used to make predictions of its performance. However, the model is no longer valid for estimator periods of 30µs or more. This is likely due to certain abstractions made in the model, such as the omission of certain physical or embedded platform effects.

4 DISCUSSION AND FUTURE WORK

In the previous sections, we described the experiments, measurements, and simulations we performed in an effort to (i) calibrate a plant model against its real-world counterpart and (ii) validate the performance of different design candidates determined using design space exploration techniques, which used this calibrated plant model to predict their performance. However, the order and sometimes extent of these steps were in retrospect not very efficient. In this section, we discuss some of these steps, presenting lessons learned and highlighting additional questions raised and identified challenges. In tandem with this, we present a more efficient (prescriptive) process model for model calibration and validation in the context of model-based design space exploration, which takes into account these lessons learned and other observations. This *prescriptive* process model is shown in Figure 11.

Face validation and statistical validation. Regarding validation, we saw that face validation as described in Subsection 3.4 can give some idea about the validity of a (system) model. However, it is based on expert knowledge and experience. While this can be reliable in certain situations, as demonstrated by the fact that our interpretation of the results here was later confirmed by the statistical validation, it remains subjective. Indeed, without an explicit frame of reference, the actual validity is hard to quantify and evaluate objectively. Statistical validation, as described in Subsection 3.5 can help quantify this, but this requires information about uncertainties and their impact on the (simulation) results to determine e.g. probability distributions or confidence bounds. In the *descriptive* process, the face validation and statistical validation were performed sequentially. However, this does not need to be the case. Indeed, they can be performed in parallel, as shown in the "full validation" section of the *prescriptive* process. Additionally, during the plant model calibration, we only considered face validity. However, it might be useful to also investigate uncertainties and their impact, and subsequently the statistical validity of the plant model. As such, this is also shown in the "plant model calibration" section of the *prescriptive* process.

Adaptive design space exploration and experiment definition. The probability distribution obtained during the statistical validation indicates that the DSE we performed was likely too fine-grained. Indeed, there is quite a bit of variation in the predicted performance due to uncertainties when compared to the overall trend (Figure 10). If this information had been available earlier, a larger step size could have likely been used for the monitor period instead of the current 1 µs, as the impact of such a small change in monitor period on real-world system performance would likely be indiscernible from other variations. This is illustrated in the "(adaptive) design space exploration" part of the *prescriptive* process model, where the probability distribution is used to modify the optimization problem definition. Similarly, this information might have been used to determine the number of measurements needed to validate the simulation results. Indeed, we could have determined how far apart (regarding the monitor period) the measured points should be to confirm or disprove the overall trend. This is shown as a "modify experiment" step in the "uncertainty propagation and measurement and simulation" section of the *prescriptive* process model. However,

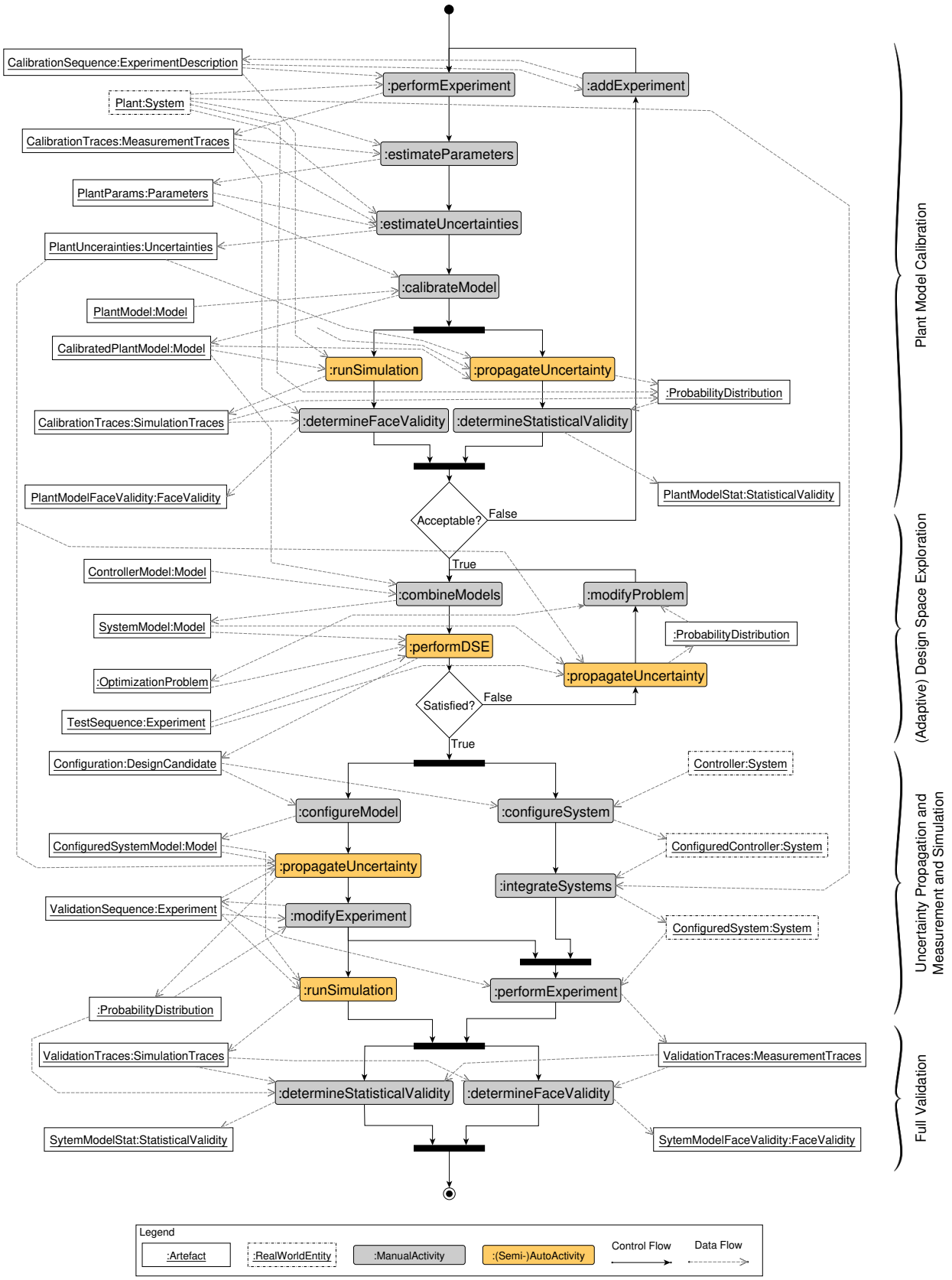

Figure 11: Possible prescriptive process model of the model calibration, design space exploration, and system validation process.

this is currently only speculation. As such, in future work, we intend to investigate to what extent this actually holds true. However, we have already identified some potential problems that would need to be addressed first:

Obtaining the probability distribution. To obtain the probability distributions required to reduce the design space, e.g. by modifying the step size, we require the full *configured* system model, as we want to know the impact of uncertainty on the closed-loop system behavior. As such, we already need to have some likely design candidates, which creates a chicken and egg problem. Indeed, we need to perform DSE to determine design candidates which we need to obtain the probability distributions to reduce the design space for the DSE. A potential way to deal with this would be to perform a kind of adaptive sampling, whereby (sampled) candidates are used to propagate the uncertainties and to obtain the probability distributions, which are then used to adjust the resolution of the DSE, leading to the loop shown in the "(adaptive) design space exploration" part of the *prescriptive* process model.

Propagating the uncertainties. Furthermore, to propagate the uncertainties, we made use of Monte Carlo simulations. However, this took a lot of computation time. In this case, the time required to propagate the uncertainties far outweighed the potential time gains resulting from reducing the resolution of the DSE. Regardless, the probability information was still relevant in the statistical validation. This highlights another problem: you may need to spend a lot of time running (Monte Carlo) simulations to save a bit of time in other steps, but this of course depends on the relative execution times of the different steps. However, there are some possible ways to deal with the computational complexity of the Monte Carlo simulations (not shown in the process model):

(1) Performing *sensitivity analysis (SA)* to determine how much each source of uncertainty contributes to observed variations on the simulation results would allow us to disregard sources with a low sensitivity index, significantly reducing the required number of simulations for the Monte Carlo analysis. However, this does create another problem: performing sensitivity analysis usually also requires many simulation runs. As such, there is a tradeoff between the time spent to perform the SA and the time saved on the Monte Carlo simulations.
(2) Another option is to use *surrogate models* which can be simulated more quickly or which can be used in other *analytical approaches* to uncertainty propagation to obtain an estimate of the required probability distributions. However, as these models are approximations of higher-order models, they introduce additional uncertainty which must be taken into account [6].

Robust optimization. Reducing the time required to obtain these probability distributions is also useful when going toward robust optimization, e.g. to take into account variability in the manufacturing process. For example, in the current paper, we considered the optimal deployment of an advanced controller for BLDC motors. However, our design candidates are to an extent tailored to our specific setup, as we calibrated our models for our setup. However, variations in motor parameters, e.g. due to manufacturing tolerances, will affect the system performance when going to series production. As such, we would need to take these variations (uncertainty) into account in the evaluation of design candidates in the design space exploration process itself. To guarantee a certain minimum performance, for example. In that case, a probability distribution on the predicted performance of design candidates is required, which would dramatically increase the computation time required to perform the DSE. Here, the potential solutions listed above would also be relevant.

Future work. The presented *prescriptive* process is at this point a hypothetical process, which assumes the challenges presented above are solved. Further research is required to actually address these challenges to make this possible and subsequently to determine how realistic this process is and the actual advantages or gains (e.g. regarding computation time) that can be obtained. Additionally, methods, tools, and techniques are required to support such a process. For example, the *validity* of models has been mentioned multiple times throughout this paper. Here, *Validity Frames (VF)* [9] can be used to explicitly capture the range of validity of each model. Similarly, the *EMF-based Simulation Specification (ESS)* [5] can be used to define and automate model validation experiments to support the validation process.

ACKNOWLEDGMENTS

This research was partially supported by Flanders Make, the strategic research centre for the manufacturing industry in Belgium. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

REFERENCES

[1] Jasper De Viaene. 2020. *Sensorless load angle detection for brushless direct current and stepping motors.* Ph. D. Dissertation. Ghent University.
[2] Jasper De Viaene, Florian Verbelen, Stijn Derammelaere, and Kurt Stockman. 2018. Energy-efficient sensorless load angle control of a BLDC motor using sinusoidal currents. *IET Electric Power Applications* 12, 9 (2018), 1378–1389.
[3] Cláudio Gomes. 2015. *Foundations for Co-simulation – IWT Proposal.* Technical Report. University of Antwerp, Antwerp.
[4] Levi Lúcio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, and Maris Jukss. 2013. FTG+ PM: An integrated framework for investigating model transformation chains. In *International SDL Forum.* Springer, 182–202.
[5] Joost Mertens and Joachim Denil. in press. ESS: EMF-based simulation specification, a domain-specific language for model validation experiments. In *2022 Annual Modeling and Simulation Conference (ANNSIM).* IEEE.
[6] Jeremy Oakley. 2004. Estimating percentiles of uncertain computer code outputs. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 53, 1 (2004), 83–93.
[7] William L Oberkampf and Christopher J Roy. 2010. *Verification and validation in scientific computing.* Cambridge University Press.
[8] Robert G Sargent. 2020. Verification and validation of simulation models: an advanced tutorial. In *2020 Winter Simulation Conference (WSC).* IEEE, 16–29.
[9] Bert Van Acker, Bentley James Oakes, Mehrdad Moradi, Paul De Meulenaere, and Joachim Denil. 2020. Validity frame concept as effort-cutting technique within the verification and validation of complex cyber-physical systems. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings.* 1–10.
[10] Yon Vanommeslaeghe, Joachim Denil, Jasper De Viaene, David Ceulemans, Stijn Derammelaere, and Paul De Meulenaere. 2019. Leveraging domain knowledge for the efficient design-space exploration of advanced cyber-physical systems. In *2019 22nd Euromicro Conference on Digital System Design (DSD).* IEEE, 351–358.
[11] Yon Vanommeslaeghe, Joachim Denil, Jasper De Viaene, David Ceulemans, Stijn Derammelaere, and Paul De Meulenaere. 2021. Ontological reasoning in the design space exploration of advanced cyber–physical systems. *Microprocessors and Microsystems* 85 (2021), 104151.
[12] Benjamin Vedder. 2017. VESC project. Retrieved July 12, 2022 from https://vesc-project.com/
[13] Bernard P Zeigler, Tag Gon Kim, and Herbert Praehofer. 2000. *Theory of modeling and simulation.* Academic press.