

## Research Article

Julien Deantoni, Paula Muñoz, Cláudio Gomes, Clark Verbrugge, Rakshit Mittal, Robert Heinrich\*, Stijn Bellis, and Antonio Vallecillo

# Quantifying and combining uncertainty for improving the behavior of Digital Twin Systems

<https://doi.org/10.1515/sample-YYYY-XXXX>

Received February DD, 2024; revised Month DD, YYYY; accepted Month DD, YYYY

**Abstract:** Uncertainty is an inherent property of any complex system, especially those that incorporate physical parts or operate in real environments. In this paper, we focus on the Digital Twins of adaptive systems, which are particularly complex to design, verify, and optimize. One of the problems of having two systems (the physical one and its digital replica) is that their behavior may not always be consistent. In addition, both twins are normally subject to different types of uncertainties, which complicates their comparison. In this paper we propose the explicit representation and treatment of the uncertainty of both twins, and show how this enables a more accurate comparison of their behaviors. Furthermore, this allows us to reduce the overall system uncertainty and improve its behavior by properly averaging the individual uncertainties of the two twins. An exemplary incubator system is used to illustrate and validate our proposal.

**Keywords:** Model-based Software Engineering, Control systems, Adaptive systems, Uncertainty

### Acronyms:

DTS: Digital Twin System

DT: Digital Twin

PT: Physical Twin

MDTS: Mitigated Digital Twin System

GT: Ground Truth

UADT: Uncertainty-Aware Digital Twin

UAAPT: Uncertainty-Aware Physical Twin

## 1 Introduction

Adaptive systems are systems that modify their behavior in response to changes in their environment or in their components. In general, adaptive systems are quite complex to design, verify, and optimize, and therefore simulations are used to analyze both their behavior and their properties of interest. In this context, models are used to represent the relevant behavioral characteristics of the system under study, whereas the simulations represent the evolution of the model over time [66]. Initially inspired by NASA's original replication approaches [25], Digital Twin Systems (DTS) have emerged that synchronize the behavior of the physical system and its models in order to improve the system's design, operation, and maintenance [14, 12].

One of the problems of having two systems (the physical one and its digital replica) is that their behavior may not always be consistent. According to Segal's law, "A man with one watch knows what time

---

**Julien Deantoni**, Universite Cote d'Azur, France, e-mail: julien.deantoni@univ-cotedazur.fr

**Paula Muñoz, Antonio Vallecillo**, ITIS Software, Universidad de Málaga, Spain, e-mail: paulam@uma.es, av@uma.es

**Cláudio Gomes**, Aarhus University, Denmark, e-mail: claudio.gomes@ece.au.dk

**Clark Verbrugge**, McGill University, Canada, e-mail: clump@cs.mcgill.ca

**Rakshit Mittal, Stijn Bellis**, University of Antwerp - Flanders Make, Belgium, e-mail: rakshit.mittal@uantwerpen.be, stijn.bellis@uantwerpen.be

\*Corresponding author: **Robert Heinrich**, Karlsruhe Institute of Technology, Germany, e-mail: robert.heinrich@kit.edu

it is. A man with two watches is never sure.” To complicate things further, such systems are not free of uncertainty; and in case of inconsistency between the two systems, it is important to understand if this is due to uncertainty or to unexpected divergence. More generally, uncertainty is an inherent property of any physical system, and can be a problem in the particular case of adaptive systems, whose decisions and control mechanisms can be affected by inaccuracies in sensor readings, looseness in mechanical parts, or inexact comparisons [28, 5, 49, 18, 27, 26]. These uncertainties can lead to erroneous behaviors.

Although in theory simulation models can help mitigate these inaccuracies, and thus correct their affected behaviors, in practice they are not free of uncertainty either. Consequently, we have to deal with both system and model uncertainties, which are of different natures. Sources of uncertainty in the case of models include input accuracy, numerical approximations, method resolution or model fidelity, among others [11, 5]. And just as the uncertainty in physical systems makes their control and operation more difficult [41], the uncertainty in the simulation models hinders their reliability and the fidelity of their predictions [6, 32].

In this paper, we address the problem of having to deal with two versions of the same system (the physical and the digital) that may diverge in their behaviors. Both systems suffer from uncertainty, but we show how this uncertainty can be combined and exploited to improve both systems. In order to address these issues in the context of DTS, in this work, we propose the following general approach which is later described in detail for our running example:

1. **Consider Uncertainty as a *first-class entity*** (c.f. Section 4.1). We explicitly represent the sources of uncertainty in the system and the model using random variables, enabling uncertainty quantification. This approach facilitates decision-making that takes explicit uncertainty into account. Instead of making decisions solely based on values’ thresholds, the controller will consider the uncertainty of the state values. These informed decisions result in more accurate control of the system’s behavior.
2. **Determine the *Model Reliability*** (c.f. Section 4.2). By explicitly quantifying uncertainty, we establish thresholds to determine when the model’s values are no longer *reliable* and unsuitable for decision-making.
3. **Mitigate Uncertainty Using the Twins’ Information** (c.f. Section 4.3). As DT and PT represent independent sources of information, we periodically combine their values to reduce their uncertainty values and control their propagation actively.
4. **Define Consistency Between Values Represented by Random Variables** (c.f. Section 4.4). To prevent the mitigation of uncertainty in the case of conflicting behaviors between the PT and DT, we introduce the concept of *consistency* to identify and address any discrepancies. This enables the triggering of self-adaptive behaviors that can modify the straying behavior of either twin.

We stress that the individual techniques for uncertainty quantification and propagation as well as reduction are already part of the state of the art (an overview is given in Section 6). Our contribution lies in the combination of these techniques in a DT context. The proposed approach will be demonstrated using the incubator case study [20], a common example of a DT of a control system. Despite its tractable size, its subtleties and intrinsic complexities permit illustrating many of the problems of adaptive systems when faced with uncertainty [19].

Based on our experiments on the case study, we show that treating uncertainty as a first-class entity enables a deeper understanding of the observed behaviors, leading to an improvement of the control decisions and hence the overall system behavior. Although the scope of Digital Twin Systems is quite varied [4], the incubator serves as a generic exemplar, and we believe our results extend to a wide range of DTS.

The organization of the paper is as follows. Section 2 describes the context and background of our work, while Section 3 describes the running example used to illustrate and evaluate our proposal. Then, Section 4 presents our proposed approach and Section 5 discusses some of its advantages and limitations. Finally, Section 6 relates our proposal to similar works, and Section 7 concludes with an outlook on future work.

## 2 Background

### 2.1 Uncertainty

Uncertainty is an inherent property of any system that operates in a real environment or interacts with physical elements or humans. Uncertainty is “the quality or state that involves imperfect and/or unknown information” [30]. Uncertainty can be due to different factors, such as imprecision in the measuring tools; lack of knowledge about the system or its environment; incorrect, incomplete, or vague information; unreliable data sources or communication networks; numerical approximations; unforeseen, emergent or unpredictable behavior; or the inability to determine whether particular events have occurred or not [30].

The purpose of explicitly representing uncertainty is twofold: a software engineer who represents or simulates a system needs to capture the relevant characteristics of uncertainty in a suitable way, while a systems engineer analyses uncertainty to try to remove it, reduce it, or mitigate its effects [44]. Methods to deal with uncertainty in its many forms (objective, subjective, epistemic, aleatory) [57], using different approaches such as mathematical and numerical models [45], probabilities [21], Fuzzy set theory [68], variability analysis [52], and risk assessment [50], have been extensively covered in literature [58].

Uncertainty can be either aleatory or epistemic [45]. *Aleatory uncertainty* refers to the inherent stochastic variability or randomness of a phenomenon. For example, measuring a physical attribute. This type of uncertainty is irreducible, in that there will always be variability in the underlying variables [30]. *Epistemic uncertainty* refers to the lack of knowledge we have about the system or its environment. For example, how the system will be used. This type of uncertainty is reducible, in that additional information may reduce it.

We now define a few types of uncertainty that are relevant for our work but we stress that our contribution is not limited to these types of uncertainty.

*Measurement uncertainty* is an aleatory uncertainty that represents the expression of the statistical dispersion of the values attributed to a measured quantity. *Numerical uncertainty* comes from operations whose outcome is only approximate. For example, using floating point arithmetic [23], using numerical solvers to approximate the solution to continuous differential equations [8], or measuring the temperature or length of any physical object, since sensors and measurement instruments are unable to distinguish values below their accuracy.

In the case of numeric values, this type of uncertainty is usually expressed in different ways, *e.g.*, by means of the parameters of a statistical distribution ( $x \pm \sigma$ , *e.g.*,  $3.5 \pm 0.01$ ); using intervals (*e.g.*,  $[a..b]$ ) according to Uniform or Triangular distributions; or by means of samples [30]. See [53] for a survey on this topic. In this study, we followed recommendations from [30] and expressed the uncertainty by using the standard deviation of a normal distribution centered in 0. In the following, uncertain reals will be considered random variables following normal distributions, otherwise they could be converted to such variables with adjusted standard deviations, as described in the GUM standard [30]. They will be compared using *equality in distribution*, *i.e.*, two variables are equal if their distributions are the same.

Note that uncertainty propagates through the operations performed on variables with uncertainty [30], affecting all related variables. Similarly, comparison between uncertain real numbers are no longer Boolean values, but become probabilities [2]. For example, consider the real values  $x = 2.0$  and  $y = 2.5$ . Using Real arithmetic,  $x < y = \text{true}$ . However, assuming some given uncertainties, namely  $x = 2.0 \pm 0.3$  and  $y = 2.5 \pm 0.25$ , then we obtain that  $x < y$  with probability 0.893 [2].

### 2.2 Control Systems

Control system engineering aims to design automated controllers for a process to keep its operating characteristics at a desired *set-point*. To drive the process to the desired step point, a control system use sensors and actuators, respectively, to monitor the state of the process and to modify its state.

A feedback-based controller utilizes an error signal which is the difference between the actual state of the system and the set-point, to make control decisions. The sensor accuracy directly impacts the quality of the control. Considering the controller of an oven, if the temperature sensor provides a biased value of  $+5^{\circ}\text{C}$  compared to the measurand, then the control will set the oven temperature  $5^{\circ}\text{C}$  higher than expected.

State-of-the-art in control engineering encompasses many levels of control loops. In embedded systems, the control is mainly implemented by using a Proportional-Integral-Derivative (PID) algorithm and its derivatives, since they provide excellent performance and simplicity in tuning. For simpler systems, where hysteresis around the step point are sufficient, the controller algorithms are referred as *bang-bang* or *on/off* controllers. They use the actuator or not, depending on a threshold over and under the desired step point. See [34] for an extensive survey on this topic.

## 2.3 Digital Twins (DTs)

While there is not yet an agreed definition of Digital Twin, in this paper we will consider that a *Digital Twin (DT)* is a virtual representation (a model) of a real-world entity or process (the *Physical Twin, PT*), synchronized at specific points in time [14]. The twinned systems (the DT and PT), the connections between them, and the set of system services compose the so-called *Digital Twin System (DTS)* [25].

The services allow exploiting the data exchanged by the two twins in different ways [54, 55, 47]. Examples of these services are dashboards to visualize and display data; Machine Learning components to support decision-making processes and predict changes in the PT over time in order to accomplish, *e.g.*, preventive maintenance; monitors to detect anomalies and trigger alerts to users; or algorithms to improve the system performance, conduct fault diagnosis or what-if analysis.

A DTS can also modify the structure or parameters of its corresponding PT based on observations. In this regard, DTs can be used for the implementation of adaptive systems as well as traditional control systems.

## 3 Running Example - The Incubator

The incubator consists of an insulated box with a heating element and a fan. The controller can read values from two temperature sensors inside the box, and actuate on a fan and the heat bed by switching them *on* or *off* (see Figure 1). The controller's goal is to maintain a stable temperature for objects inside the insulated box [20, 19].

A conceptual diagram of the incubator DTS is represented in Figure 2. The left-hand side represents elements of the PT, shown in Figure 1, with a separation between the plant and the controller. The implemented controller is a *bang-bang* controller. On the right-hand side, the DT, aimed to mimic the PT, is represented. The plant model is specified by ordinary differential equations and can be simulated (by using, *e.g.*, a Euler solver). The controller is modeled by a State Chart and can be simulated too. The data exchanged between the controller and the plant is the same in the physical and the digital twins. Typically, both twins write measured/computed data in the database and can consume reconfiguration commands. A set of services completes the architecture. The services also read and write from and to the database.

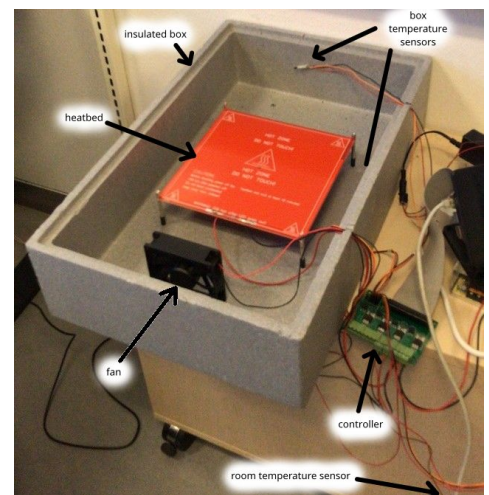


Fig. 1: The experimental incubator (lid open).

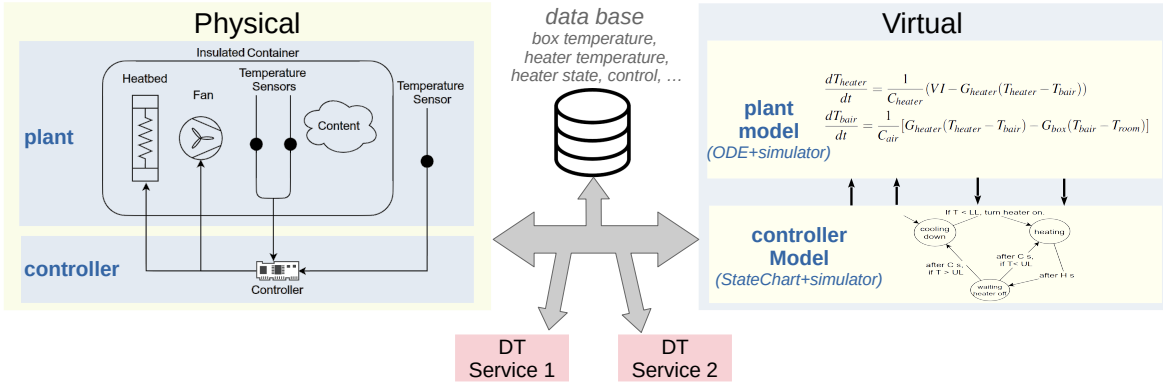


Fig. 2: A schematic overview of the incubator DT setup with the PT on the left and the DT on the right as well as the other DT services.

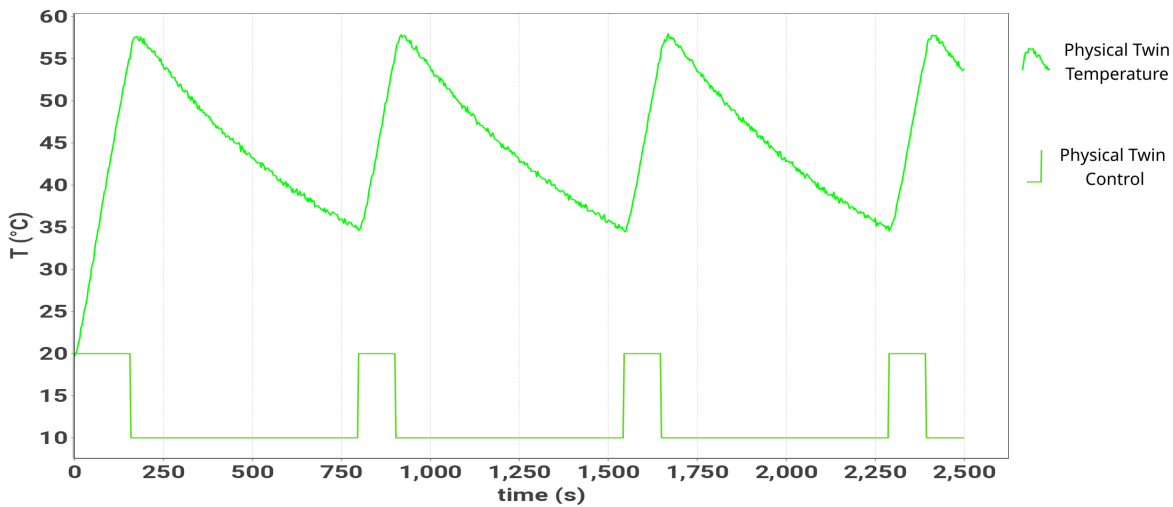


Fig. 3: Evolution of the incubator box temperature and the heater control of the classical control (PT). The non-smoothness of the temperature curve highlights the presence of stable white noise in the incubator temperature.

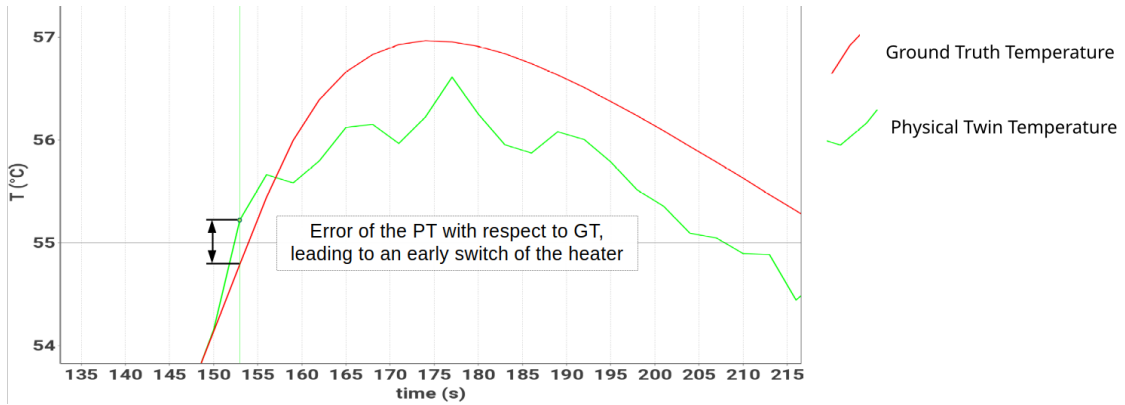
### 3.1 Physical Twin Uncertainty

The actual physical incubator uses two DS18S20 sensors<sup>1</sup>. One sensor measures the temperature inside the box, while the other measures the room temperature.<sup>2</sup> From their datasheets, these sensors have an accuracy of  $\pm 0.5^{\circ}C$  and can only be read once every two seconds. A Raspberry Pi serves as the controller, running every three seconds. It averages the temperature values from both sensors to determine the internal temperature of the box. Assuming that the accuracy depicted in the datasheet follows a uniform probability density function, the standard uncertainty  $\sigma_1$  and  $\sigma_2$  of respective sensors  $S_1$  and  $S_2$  can be approximated to  $0.5/\sqrt{3} = 0.289$ . This value was used to account for the sensor uncertainty explicitly. Although there are other sources of uncertainty in the system, such as the time delay between sensor readings and their use by the controller, for this example, we only considered the uncertainty from the sensors.

It is important to note that we assume that the uncertainty from sensors remains stable, as the noise between successive sensor readings are not correlated random variables. Noisy variables are obtained when

<sup>1</sup> <https://www.analog.com/media/en/technical-documentation/data-sheets/DS18S20.pdf>

<sup>2</sup> Note that so far the incubator has been tested under the hypothesis of constant room temperature.



**Fig. 4:** Error of PT with respect to measurand system (ground truth), leading to an early switch of the heater. This causes the behavior of the PT and the DT to drift over time, requiring periodic synchronization.

using the actual sensors, as shown in Figure 3. Additionally, due to the bang-bang controller, the timing of heater activation and deactivation may vary depending on the noise or uncertainty. This causes different executions to deviate in time from each other, see Figure 4. This deviation will be characterized in Section 5.

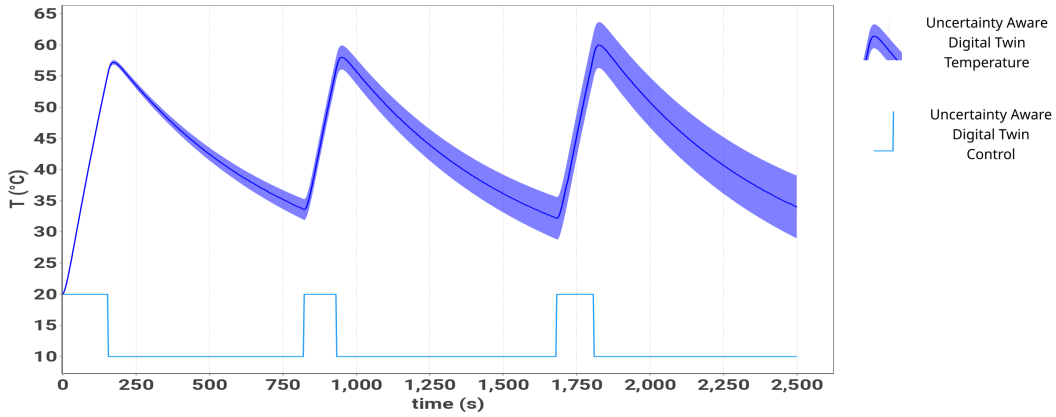
In our experiments, we compared our results with the ground truth (GT), which is partially based on measurements. To enable such comparisons, we used a synthetic measurand defined by differential equations, validated against real experiments (see experiment datasets online<sup>3</sup>). We also used a synthetic physical environment to replicate the sensor noises according to their specifications.

### 3.2 Digital Twin Uncertainty

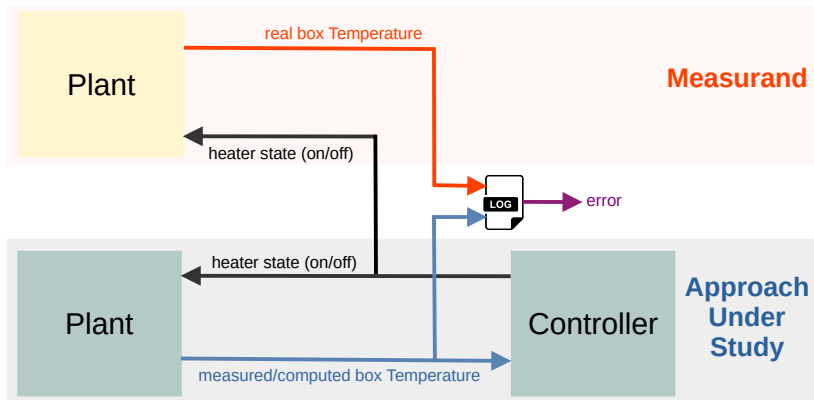
The incubator model in this study is based on ordinary differential equations according to Newton’s law of cooling. The model input parameters are the thermal conductivity of the air inside the box and the heater; the heat transfer capacity of the box and the heater; and the voltage and current input of the heater when *on*. All these parameters are calibrated to closely match the observations of the actual incubator. However, they are all subject to uncertainty. For example, the heat transfer capacity is calculated from the materials and the mass of the heater. These two values are obtained using heater data sheet knowledge and instruments, which both introduce uncertainty. To address these measurement uncertainties, the parameters were modeled using standard deviations in the simulations.

In addition, a solver approximates the incubator model using a forward Euler integration method. The discretization performed by the solver introduces numerical approximations and, consequently, uncertainties. The Euler method is a first-order method, meaning that the local error (error per step) is proportional to the square of the step size, and so is the related uncertainty. Moreover, as the solver will often rely on past estimated model quantities to calculate the future model quantities, contrary to sensor uncertainty, model uncertainty will increase with time as input uncertainties propagate through the model and equations are approximated by the solver. Figure 5 illustrates the increasing uncertainty throughout the simulation, as it propagates in the successive calculations. This emphasizes the importance of quantifying uncertainty for better decision-making, as decisions made under high uncertainty could be unsafe..

<sup>3</sup> [https://github.com/INTO-CPS-Association/example\\_digital-twin\\_incubator/blob/master/software/incubator/datasets/README.md](https://github.com/INTO-CPS-Association/example_digital-twin_incubator/blob/master/software/incubator/datasets/README.md)



**Fig. 5:** Evolution of the incubator temperature and the heater control of the uncertainty-aware model-based control (Uncertainty-Aware Digital Twin, UADT). The uncertainty is represented around the nominal temperature value. It highlights the increase of uncertainty along the simulation.



**Fig. 6:** Experimental setup to measure the error of an approach under study with respect to the measurand.

### 3.3 Experimental setup

As in any dual system, deviations between the behavior of the two twins are possible. In order to quantify this deviation, we defined a set of experiments where the measurand (*i.e.*, the PT) is run conjointly with the approach under study (the DT). More precisely, the measurand of the plant is controlled by the controller of the approach under study. This experiment allows comparing what actually happens in reality with what is perceived by the controller (see Figure 6).

With this architecture we are able to measure the difference between the perceived temperature and the actual one each time the controller decided to switch the heater *on* or *off*. We will use this setup to compare the behavior of the system We will use this setup to compare the behavior of the system when uncertainty is not considered, when the individual uncertainties of the two twins are taken into account, and when these uncertainties are properly combined to improve the overall system behavior. The results of these tests will be later discussed in Section 5.1.

## 4 Proposal

The previous section has presented the two main artifacts of our DTS, the physical and the digital twins. Both are subject to uncertainty. The uncertainty of the PT comes from sources such as the sensors' accuracy or discretization period. The uncertainty of the DT comes from numerical approximations, parameter approximations and from the model itself. Thus, the uncertainties of both twins come from different and independent sources. In this section, we explain how making the uncertainty in each artifact explicit can be used to improve the overall behavior of the DTS.

### 4.1 Considering Uncertainty as a First-class Entity

We propose to explicitly model the various sources of uncertainty in the DTS. The main idea consists in assessing systematically the sources of uncertainty to enable the conscious use of all the data in a DTS. This means that random variables must be leveraged in both the PT, the DT, and the associated services. Following the ISO Guide to Measurement Uncertainty [30], the uncertainty in each of the values can be represented by a Gaussian distribution, whose uncertainty is given by the standard deviation of the distribution. Then we can extend the definition of *execution trace* to account for uncertainty.

**Definition 1.** We consider that an **execution trace** (of either the system or the model) is no longer a set of values  $Y = \{y_i\}_{i=1}^n$  but a set of random variables, where each random variable is defined by both an actual value and its standard deviation:  $Y = \{y_i \pm \sigma_i\}_{i=1}^n$ .

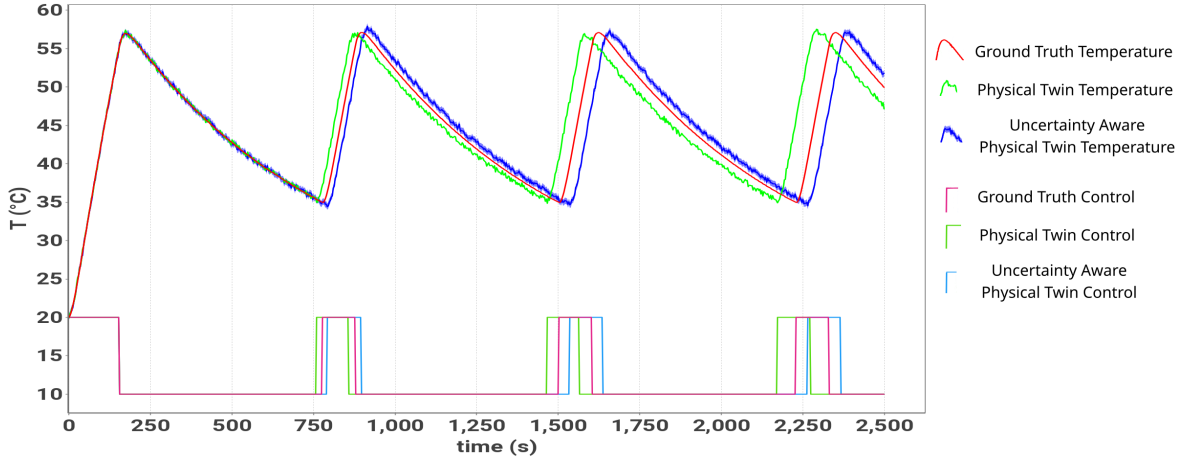
This is intuitively depicted in Figure 5, which shows not only the curve with the nominal incubator temperature but also its uncertainty. We can see how the temperature evolution is now represented by what is called a “flow pipe,” which wraps around the initial curve, enveloping it, and shows the uncertainty.

Considering the use of uncertainty in the PT, we used the knowledge about the sensors' accuracy to define the box and room temperatures as random variables with a standard deviation of 0.289 (see Section 3.1)<sup>4</sup>.

We explained that the controller of the PT averages the values from two sensors to obtain the box temperature, each of which has a standard deviation of 0.289. Since these sensors are two independent sources of evidence for the same value, the resulting uncertainty  $\sigma_T$  of the temperature used by the controller

<sup>4</sup> These adaptations can be done directly in the implementation of the sensor acquisition, or offered as a service of the DTS, which could create random variables from the raw sensor readings in real-time.





**Fig. 7:** Comparison of the evolution of the incubator temperature and the heater control between the measurand system (Ground Truth), the classical control system (Physical Twin) and the uncertainty-aware control (Uncertainty-Aware Physical Twin). Confidence = 95%. This figure also shows the drifting of the different behaviors over time.

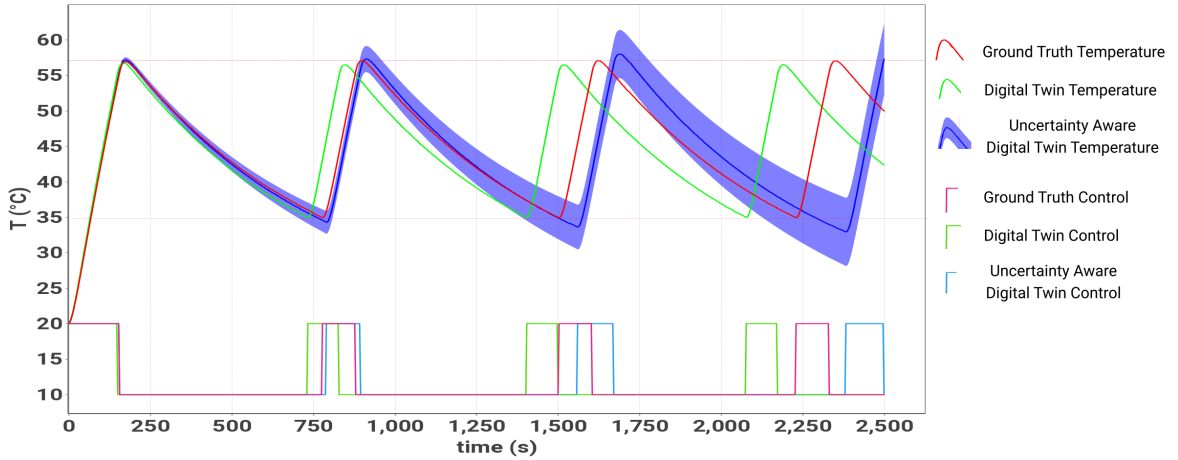
is  $\frac{1}{2}\sqrt{\sigma_1^2 + \sigma_2^2} \approx 0.204$  which is less than 0.289. This shows us that the uncertainty decreases when averaging independent sources of evidence. Sensors are noisy, and it is well-known that averaging independent sensor values reduces the effect of the noise [17]. However, it is not always obvious what the resulting uncertainty is after averaging. Making uncertainty explicit helps in the subsequent operation with its values. Here, since the box temperature is a random variable, the controller may not compare whether the temperature is above or below a specific threshold. Instead, it may compare whether the confidence of being above or below is greater than a specific confidence level (typically 95%).<sup>5</sup>

In this case, using random variables is a natural enabler for a systematic and lightweight form of stochastic control. In our experiments, such modifications provided a control closer to the ground truth (GT) than the classical system control that does not consider uncertainty.

This is illustrated in Figure 7, where the *red* time series represents the GT (*i.e.*, the measurand), the *green* time series the classical control system (*i.e.*, the PT) and the *blue* time series the control with uncertainty as a first-class entity (hereinafter referred to as *Uncertainty-aware Physical Twin*, UAPT). We observe that the red line is closer to the blue line than the green line over time. The blue line responds more accurately to the temperature thresholds, while the green line diverges further after each cycle. We will discuss this in more detail in Section 5.1.

Considering the use of uncertainty in the DT (*i.e.*, the simulation models), all the arithmetic behind the ODE and the solver should make use of the properties of random variables to understand how the uncertainty from the random variables used in the parameters and the inputs propagates to the outputs. Additionally, it motivates us to characterize the numerical error introduced by the equation solver. It is not always obvious to figure out how fast the uncertainty grows throughout the simulation, depending on the size of the discretization time step or the change of uncertainty in the sensors. By making uncertainty explicit, users are faced with the fact that computing the evolution of the box temperature from noisy sensors with an Euler solver and a discretization time step of 1ms, for 2500 seconds, leads to a box temperature with a standard deviation of 2.52 °C, *i.e.*, with an error of  $\pm 2.52$  °C (at 95% of confidence, see Figure 8). It is up to the user to consider this information relevant or not and to handle it accordingly. However, it should not be ignored.

<sup>5</sup> A *confidence level* represents the long-run proportion of confidence intervals that theoretically contain the true value of a parameter. In turn, the *confidence interval* represents a range of values providing the estimate of an unknown parameter of a population. For example, out of all intervals computed at the 95% level, 95% of them should contain the parameter's true value.



**Fig. 8:** Comparison of the evolution of the incubator temperature and the heater control between the measurand system (Ground Truth), the model-based control system (Digital Twin) and the uncertainty-aware model-based control (Uncertainty-Aware Digital Twin). Confidence = 95%.

## 4.2 Model reliability

To make the above consideration explicit, we propose to define the notion of *reliability* of the simulation based on the quantification of uncertainty. In other words, it is important, when a simulation is used in a specific context, to clearly define the conditions under which the simulation is considered sufficiently meaningful (and reliable). For example, considering Figure 8, the decisions made with lower levels of uncertainty are closer to the GT. However, as uncertainty levels increase, the control system becomes unable to make prompt decisions with such high degrees of uncertainty. Instead, it delays them until the confidence level reaches the desired threshold.

The amount of acceptable uncertainty depends on various factors, such as the characteristics of the modeled physical phenomena or the robustness of the controller. However, we advocate for an explicit definition of the acceptable amount of uncertainty on each output random variable, above which the simulation is not considered anymore as a model of the reality (in the sense of [43]), *i.e.*, the model cannot be used to confidently answer questions about the corresponding reality.

**Definition 2.** A simulation model is considered **reliable** if the uncertainty associated with its input and output variables is below the threshold defined for each one, which determines their corresponding acceptable amount of uncertainty.

For instance, for the incubator example, an error of  $\pm 2.5^\circ\text{C}$  in the temperature inside the box may be tolerated if the goal of the simulation is to dimension the incubator power supply; however, it is not acceptable if the goal is to use the incubator to hatch eggs<sup>6</sup>.

Technically speaking, the definition and handling of such random variables as first-class entities ultimately rely on technical libraries and require changes both in the physical and digital twins of the system. In the modeling and simulation context imposed by a DTS, a suitable library should (1) allow the usual arithmetic operations imposed by classical algorithmic and ODE solving, (2) allow for probabilistic comparison of random variables as required by stochastic process control, and (3) use closed-form solutions to avoid too

<sup>6</sup> Within a range of 35 to 40.5°C there is the possibility that eggs will hatch. The optimum (for hens) is 37.5 °C. Above this temperature, in addition to reduced hatching, there will be an increase in the number of crippled and deformed chicks. Above 40.5°C no embryos will survive. <https://brinsea.co.uk/latest/wp-content/uploads/2017/04/What-if-the-power-goes-off-2010.pdf>

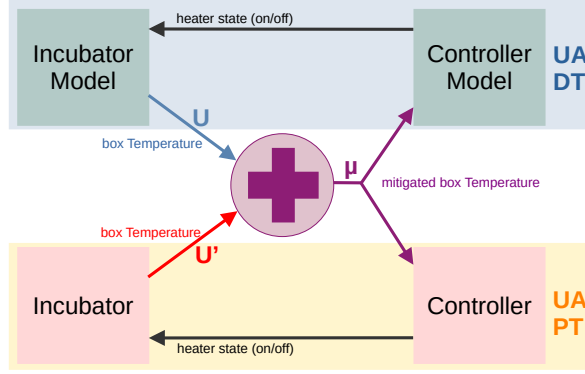


Fig. 9: Architecture to Mitigate Uncertainty ( $\mu$ ) in the incubator DTS context (MDTS).

much overhead, which would make real implementations unrealistic. In this paper, we used the Java Library described in [2], which meets such requirements.

- 
- 
- 
- 

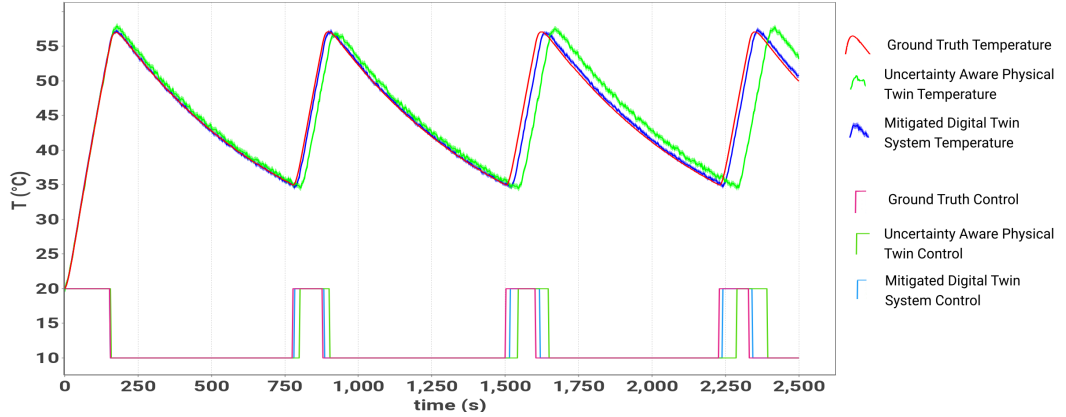
### 4.3 Uncertainty Mitigation By Using the Twins’ Uncertainties

We explained in the previous subsection that both the PT and the DT have uncertainties. In this paper, we take advantage of the fact that we have two synchronized systems executing the same behavior with different sources of uncertainty and therefore representing two independent sources of information. As we shall see, this will enable the mitigation of uncertainty. We rely on the extensive work about forecast prediction, where since the late 1960s it is well known that if you take several forecasts and average them, then the resulting aggregated forecast outperforms the individual forecasts [10]. The most accurate way to average them is to use a weighted average, in which each forecast is weighted by its inverse “error” [61]; this is in essence what the Kalman filter does [3]. In our context, that error corresponds to their uncertainty. This way, less weight is given to the least certain predictions.

In our case, the PT has uncertainty in the box temperature  $T_p$  due to its implementation characteristics; and the DT has uncertainty in the box temperature  $T_d$  due to modeling characteristics. This means that  $T_p$  and  $T_d$  can be modeled by two independent variables that follow Normal distributions with standard deviations  $\sigma_p$  and  $\sigma_d$ , respectively. Their *weighted average* is then given by another Normal distribution,  $T_A = \frac{1}{(\sigma_p^2 + \sigma_d^2)}(\sigma_d^2 T_p + \sigma_p^2 T_d)$ , whose variance is  $\sigma_A^2 = (\sigma_p^2 \sigma_d^2) / (\sigma_p^2 + \sigma_d^2)$ .

Since for any two positive numbers  $a, b$  we know that  $ab / (a + b) < \min\{a, b\}$ , we conclude that the uncertainty of the weighted average of two uncertain numbers will always be less than the uncertainty of any of the two.

To realize this uncertainty reduction, we propose a DT service that implements the architecture described in Figure 9. The box temperatures provided by both twins are averaged before being provided to the controllers of each twin. As explained, the mitigated uncertainty  $\mu$  from Figure 9 is smaller than the minimum between the uncertainties  $U$  and  $U'$  of the temperatures from the plant and from its model. Since the input uncertainty  $\mu$  is now lower in both controllers, they can perform a more accurate control of the box temperature. This is illustrated in Figure 10, where the behavior of the controller with the mitigated uncertainty ( $\mu$ ) is much closer to the intended ground truth and consequently drifts at a slower rate from it. Note that Figure 10 also represents the evolution of the uncertainty aware physical twin temperature that



**Fig. 10:** Comparison of the evolution of the incubator temperature and the heater control between the measurand *system* (ground truth), the uncertainty aware control *system* (Uncertainty Aware Physical twin) and the mitigated control *system* (mitigated digital twin system). Confidence = 95%.

would be used in a classical digital twin setup. The mitigated version is visibly closer to the ground truth than the non-mitigated one.

We refer to the uncertainty-aware DTS making use of mitigation as the *Mitigated Digital Twin System (MDTS)*. It uses the UADT and UAPT, but their controllers use the averaged uncertainty  $\mu$  (see Figure 9). By using this method, we obtain a better control since controllers are fed with a less uncertain box temperature value. However, the uncertainty of the plant model continues to grow throughout the simulation. To avoid this phenomenon, we will use the notion of *model reliability* described above. Thus, when the model simulation is too close to its reliability limit, we reset its box temperature with a more certain value, *i.e.*, the mitigated one; reducing the box temperature from  $U$  to  $\mu$  and consequently taking away the plant model from its reliability limit. The same kind of approach is used in the Kalman filter [42], but the resetting is done at each simulation step.

To avoid too many synchronizations between the PT and the DT, for instance to be used in a distributed DTS setup, we realize the mitigation only when required, *i.e.*, when the model simulation gets too close to its reliability limit. Again, we take advantage of the explicit representation of uncertainty and the possibility to consciously define the simulation reliability, to (1) avoid the increasing uncertainty in the plant model throughout the simulation, and (2) limit the number of re-settings to a minimum to maintain the reliability of the model simulation.

Note that this averaging makes sense only if applied to values that are *consistent*, *i.e.*, when different sources of information represent the same reality. To ensure that the averages yield correct results, we have defined a notion of consistency between random variables and used it to detect divergences between the PT and DT.

#### 4.4 Defining Consistency Between Values Represented by Random Variables

In the previous subsection, we explained how PT and DT uncertainties can be averaged to mitigate the overall uncertainty and, consequently, improve the behavior of the DTS. However, averaging data if conditions change in the physical environment can be dangerous as the results may become meaningless or, worse, hide the change—which could have catastrophic consequences. For instance, if someone opens the lid during incubation or the box breaks, averaging the inconsistent values may mask the actual change in the physical environment, and potentially lead to a fire.

We exploit our uncertainty quantification to define the **consistency** (denoted as  $\simeq$ ) between two random variables. For this, suppose two values  $v_p$  and  $v_d$  that represent the same physical value at time  $t$  as provided by the PT and the DT. Each value has an explicit uncertainty,  $\sigma_p$  and  $\sigma_d$ . Then, they define two

random variables  $V_p$  and  $V_d$  with Normal distributions whose means are the values  $v_p$  and  $v_d$ , and their standard deviations are the respective uncertainties.

We say that both values are *consistent* with a given confidence level of  $c$ , e.g., 95%, if the intervals  $[v_p - K\sigma_p, v_p + K\sigma_p]$  and  $[v_d - K\sigma_d, v_d + K\sigma_d]$  overlap; *i.e.*,

$$[v_p - K\sigma_p, v_p + K\sigma_p] \cap [v_d - K\sigma_d, v_d + K\sigma_d] \neq \emptyset$$

**Definition 3.** We will say that two random variables  $V_p$  and  $V_d$  that follow normal distributions  $N(v_p, \sigma_p)$  and  $N(v_d, \sigma_d)$ , respectively, are **consistent** (and write  $V_p \approx V_d$ ) with a given confidence level of  $c$ , e.g., 95%, if the intervals  $[v_p - K\sigma_p, v_p + K\sigma_p]$  and  $[v_d - K\sigma_d, v_d + K\sigma_d]$  overlap; *i.e.*,  $[v_p - K\sigma_p, v_p + K\sigma_p] \cap [v_d - K\sigma_d, v_d + K\sigma_d] \neq \emptyset$

In this equation,  $K$  is a real number that represents the factor needed to ensure that the percentage of values that lie within an interval estimate of  $[x - K\sigma, x + K\sigma]$  from a Normal distribution  $N(x, \sigma)$  is at least  $c$ . For example,  $K = 1$  for  $c = 0.68$ ,  $K = 2$  for  $c = 0.95$ , and  $K = 3$  for  $c = 0.997$ . This is known as the 68–95–99.7 rule in statistics. In the following, we will consider that  $c = 0.95$ , and therefore we will take  $K = 2$ . Note that this value  $K = 2$  corresponds to what is called *extended uncertainty* in the GUM [30], which ensures a 95% degree of confidence.

This definition of consistency between values that are represented by random variables ( $\approx$ ) can be extended to return not only a Boolean value that expresses whether two variables are consistent or not, but a degree of consistency ( $\hat{\approx}$ ), expressed using a real number between 0 and 1. First, if the uncertainty interval of one of the variables is fully contained in the other, then the degree of consistency is 1. If the two intervals do not overlap, the degree of consistency is 0. Otherwise, if the intervals overlap, the degree of consistency is simply defined by the ratio between the intersection of the uncertainty intervals and their union. More precisely,

$$V_p \hat{\approx} V_d = \frac{\min\{v_p + 2\sigma_p, v_d + 2\sigma_d\} - \max\{v_p - 2\sigma_p, v_d - 2\sigma_d\}}{\max\{v_p + 2\sigma_p, v_d + 2\sigma_d\} - \min\{v_p - 2\sigma_p, v_d - 2\sigma_d\}}$$

**Definition 4.** Given an extended uncertainty  $K$  that produces a confidence level of  $c$ , and two random variables  $V_p$  and  $V_d$  that follow normal distributions  $N(v_p, \sigma_p)$  and  $N(v_d, \sigma_d)$ , respectively, the **degree of consistency** between them ( $V_p \hat{\approx} V_d$ ) is defined as follows.

- $V_p \hat{\approx} V_d = 0$  if their uncertainty intervals do not overlap, *i.e.*,  $[v_p - K\sigma_p, v_p + K\sigma_p] \cap [v_d - K\sigma_d, v_d + K\sigma_d] = \emptyset$
- $V_p \hat{\approx} V_d = 1$  if one of the intervals is fully contained in the other, *i.e.*,  $[v_p - K\sigma_p, v_p + K\sigma_p] \subseteq [v_d - K\sigma_d, v_d + K\sigma_d]$  or  $[v_p - K\sigma_p, v_p + K\sigma_p] \supseteq [v_d - K\sigma_d, v_d + K\sigma_d]$
- $V_p \hat{\approx} V_d = \frac{\min\{v_p + K\sigma_p, v_d + K\sigma_d\} - \max\{v_p - K\sigma_p, v_d - K\sigma_d\}}{\max\{v_p + K\sigma_p, v_d + K\sigma_d\} - \min\{v_p - K\sigma_p, v_d - K\sigma_d\}}$  otherwise, *i.e.*, the intervals overlap.

The less the intervals of uncertainty of the two variables overlap, the closer to zero the degree of consistency is. Note that this consistency operator is very different from the simulation’s reliability one. Operator “ $\hat{\approx}$ ” does not represent the confidence we can have in the data, which may have a lot of uncertainty. Instead, it represents the likelihood that the two values belong to *close enough* distributions. Note also that this is different from the probability of the two variables being *equal in distribution* [2], and also different from classical statistical distance functions between random variables [48].

#### 4.4.1 Consistent Behaviors

The consistency operator ( $\hat{\approx}$ ) was defined between values that are represented by means of random variables. This operator can also be extended to uncertain behaviors, *i.e.*, sequences of uncertain values that correspond to the execution traces of either the DT or the PT.

**Definition 5.** Given two uncertain behaviors  $X = \{x(t) \pm \sigma_x(t)\}_{t=1}^n$  and  $Y = \{y(t) \pm \sigma_y(t)\}_{t=1}^n$ , we say that, given a confidence level  $c$ , *e.g.*,  $c = 0.95$ ,  $X$  and  $Y$  are **consistent behaviors** (noted as  $X \hat{\simeq} Y$ ) iff  $x(t) \hat{\simeq} y(t) \geq c, \forall t = 1..n$ .

Note that one could also define weaker versions of this operator. For example, we can request that  $x(t) \hat{\simeq} y(t) \geq c$  for only a percentage of the values of  $t$ , and not for all of them. In other words, trace consistency can be adapted to different types of more or less smooth processes, but it always defines when two traces should be considered to represent the same measurement behavior.

#### 4.4.2 Inconsistent Behaviors

In this paper, we are also interested in detecting inconsistent behaviors. More precisely, we are interested in detecting when two consistent behaviors start diverging and become **inconsistent**.

**Definition 6.** An **inconsistency** occurs at one moment in time (*e.g.*,  $t_0$ ) if  $x(t_0) \hat{\simeq} y(t_0) \leq R$ , being  $R$  a threshold that represents the minimum degree of consistency below which the two values cannot be considered consistent.

For example, we can set  $R = 0.05$ . Note that, by definition, if we set the value of  $R$  to 0, we request that the two uncertainty intervals be disjoint.

Intuitively, this means that the flow pipes enveloping the two paths do not overlap, or overlap so little that they cannot be considered similar.

From our experiments, it is important to check the consistency of the behaviors before averaging their uncertainties to avoid hiding a divergence between the twins. The detection and potential reactions to inconsistent and divergent behaviors are discussed later in Section 5.2.

## 5 Results & Discussion

In the previous section, we have defined how to capture model and physical uncertainties in the variables by means of random variables that are able to quantify and operate with their uncertainty. We have also described how to mitigate the uncertainty of the inputs of the controllers by combining the model and system uncertainties using their weighted average. Finally, we have defined and characterized what we understand by consistent behaviors and how to detect inconsistencies that represent a divergence in the behaviors of the twins.

In this section, we discuss in more detail the improvements that can be achieved with these definitions by studying a large number of experiments based on the incubator system.

### 5.1 Improving the Behavior of the System with Uncertainty-aware Controllers

Our goal was to create a control system that is more accurate to the ground truth by taking into account the uncertainty of the system, as compared to the classical approach where uncertainty is ignored. To achieve this, we used three models represented by red, green, and blue lines. The red line represents ground truth (GT), the green line represents the classical model (PT), and the blue line represents our proposal (MDTS) that considers uncertainty and its mitigation (see Figure 10).

Previous figures 7, 8 and 10 were obtained by single executions of the system and are therefore subject to change depending on the noise of each run. They provide a simple way to visualize whether a specific

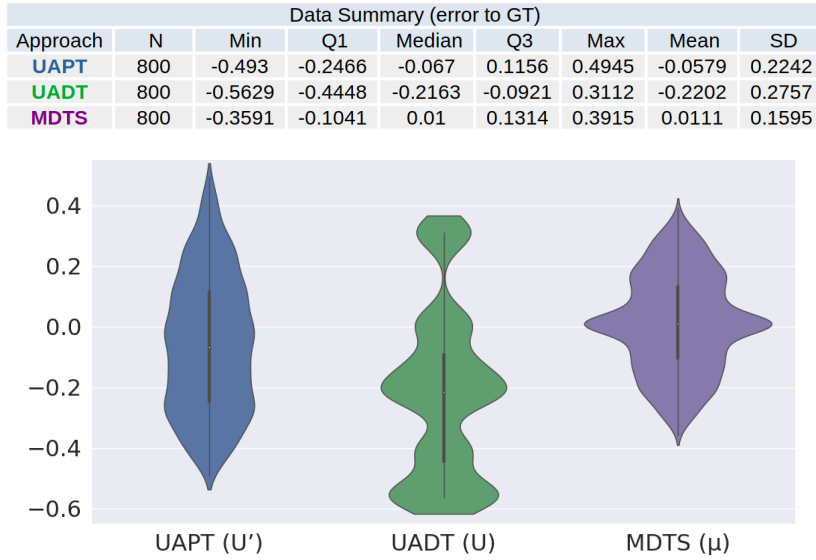


Fig. 11: Error with respect to measurand for different configurations, along 100 runs of 2500 seconds each.

approach is “closer” or not to what would be a control system based on the box temperature from the measurand. The deviation in time between the different approaches comes from the use of an erroneous box temperature by the controller, compared to the actual one in reality (see Figure 4). In order to quantify this error, we set up experiments where the measurand is run conjointly with the approach under study. More precisely, the measurand of the plant is controlled by the controller of the approach under study. This experiment allows comparing what actually happens in reality with what is perceived by the controller (see Figure 6). Based on the architecture depicted in Figure 6, we logged 100 times executions of a 2500-second incubation, and measured the difference between the perceived temperature and the actual one each time the controller decided to switch the heater *on* or *off*. We measured the error of 3 different approaches: the uncertainty-aware PT approach (UAPT), the uncertainty-aware DT (UADT), and the mitigated approach (MDTS) proposed in Section 4.3. The results are presented in Figure 11, where we compare the errors from the UAPT, UADT and the MDTS, using a violin plot [29]. These violin plots combine a box plot with a kernel density plot, providing a compact representation of the distribution and density of the errors for each approach. The results represent a significant reduction of the error in the MDTS compared to both the UAPT and the UADT, confirming the benefits of the proposed approach.

In order to better characterize the reduction of the error in the mitigated approach, we additionally measured the uncertainty of the box temperature for the 3 approaches: UAPT, UADT and MDTS. In other words, we measured  $U$ ,  $U'$  and  $\mu$  (see Figure 9). The measures were realized along 100 runs of 2500 seconds and logged at each time step. The results are represented in Figure 12 using another violin plot. The figure shows the distribution of the values obtained for  $U'$  (from UAPT),  $U$  (from UADT) and  $\mu$ , the mitigated uncertainty. We can see that, as expected, the uncertainty from the measurements  $U'$  remains constant all over the runs. The uncertainty from the model  $U$  is spread between a very small one (0.005 at the first simulation step of each run) to the maximum we defined for its reliability, *i.e.*, 0.3 in this experiment. The mitigated uncertainty  $\mu$  is significantly smaller than both  $U$  and  $U'$ . This explains the reduction of the error observed in Figure 11.

Data Summary (box temperature uncertainty)								
Approach	N	Min	Q1	Median	Q3	Max	Mean	SD
UAPT	83401	0.204	0.204	0.204	0.204	0.204	0.204	0
UADT	83401	0.006	0.1944	0.2295	0.2644	0.2998	0.2245	0.053
MDTS	83401	0.0059	0.1407	0.1525	0.1615	0.1687	0.1473	0.0237

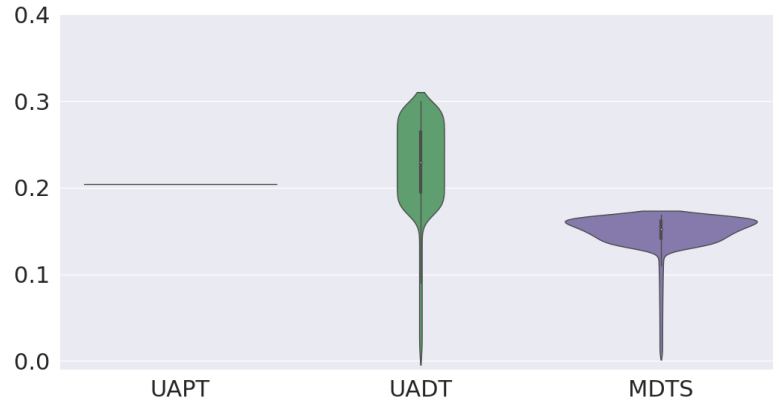


Fig. 12: Uncertainty along 100 runs of 2500 seconds each.

## 5.2 Managing Inconsistent Behaviors

Sections 4.4.1 and 4.4.2 defined the notions of consistent and inconsistent behaviors of the two twins. Inconsistent behaviors may result in divergent executions, which may require reactions to reverse them or react to their consequences.

There are several ways to understand and react to the divergent behaviors of the two twins, depending on the specific system and the application domain. At best, inconsistency can be tolerated if it is considered harmless or caused by temporary situations. Whatever the case, inconsistent behaviors should always be detected because of the damaging, or even dangerous, consequences they may entail.

Occasional inconsistency may reveal intermittent small deviations, or simply that the uncertainty of one of the twins is not correctly assessed. Detecting when these inconsistencies occur can help engineers understand the source of the problem. For instance, considering the incubator, if inconsistencies always occur around the top of the box temperature time series after the heater is *off* and before the temperature decreases, this may be due to poorly quantified sources of uncertainty in the heat restitution capacity of the heater, explaining why heater inertia is different between twins.

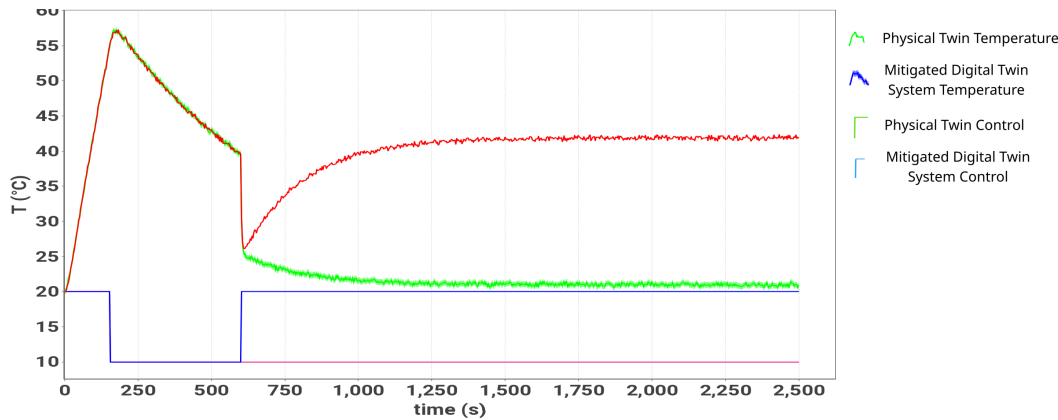
It may also be important to monitor the frequency of occasional inconsistencies, as the accuracy of some sensors decreases over time. An increase in the frequency of inconsistencies may reveal, *e.g.*, the aging of a sensor, possibly requiring an adjustment or predictive maintenance.

Finally, inconsistencies lasting for several successive values may reveal deep divergences between the twins. Such divergences may be due to a problem or failure in one of the twins. In the PT, it may be due, *e.g.*, to the breakage of a physical part or to an unexpected event in its environment. In the DT, divergences may be due to a bug in the solver, a too-abstract model, or even to a source of epistemic uncertainty that has not been considered [45].

In all cases, the user (and the services) of the DTS must be warned since a proper reaction is often required. Furthermore, if the controlled process is critical and the source of divergence is not yet understood, a safety procedure could be initiated to bring the controlled process into a safe state. This can be difficult if we consider complex processes, such as autonomous vehicles, or more straightforward if we consider the incubator, where the heater can simply be turned off.

To illustrate the use of this consistency checking, we mimicked the opening of the lid during the simulation after 600 seconds of incubation. This event drastically reduces the insulation of the incubator. Our approach promptly detects this anomaly by checking the consistency between the physical and digital





**Fig. 13:** Consistency Based Detection of a Failure introduced at time 600 seconds. By using the proposed notion of consistency, the proposed mitigated DT system successfully detects the failure and put the system in safe mode. The traditional PT continues trying to achieve the requested control, possibly leading to other failures.

twins and stops the system safely by turning off the heater. In contrast, the classical approach, which lacks any anomaly-detection mechanism, fails to detect this divergence and tries to maintain the desired temperature, resulting in potentially harmful consequences. It could be possible to add an ad-hoc anomaly detection mechanism in the classical approach. However, we believe that using our notion of inconsistency between the PT and the DT behaviors provides a natural support for this task.

Figure 13 illustrates the effectiveness of our inconsistency approach for anomaly detection. Few control cycles after the lid is opened, our approach successfully detects the divergence between the twins based on their uncertainty. The incubator heater is stopped and the temperature gradually decreases toward room temperature over the next 2000 seconds (approximately 33 minutes), *i.e.*, until the end of the simulation. Conversely, in the classical approach, the controller keeps the heater on during these 33 minutes in an attempt to maintain the desired temperature, resulting in a significant waste of energy and potential danger for the incubated object and its environment.

### 5.3 Limitations of our proposal

So far, we have discussed the features and advantages of our proposal. This section discusses some of the limitations that we have also identified.

One of the main limitations is that the exploratory phase of the proposed technique has been conducted only on the case study of the incubator. This does not ensure that the technique will be fully applicable to other contexts or systems with different characteristics. However, based on the results, we believe that the technique could be applicable to any DTS that operates by synchronizing the behavior of both twins.

Furthermore, the tests have been conducted mainly on synthetic data, which means that we do not know of other challenges that may arise when applying the technique in practice. For example, unknown uncertainties that we may have abstracted in the synthetic model, or emergent interactions between them [6], could also influence the system behavior. We however tried to use synthetic data that was as realistic as possible from the datasets available online<sup>7</sup>.

Another limitation may lie in the quantification of the initial uncertainty of the parameters. Each simulation parameter has an associated initial level of uncertainty that will then propagate. Determining the value of that uncertainty may not be trivial because it may not be directly derived from the measurement

<sup>7</sup> [https://github.com/INTO-CPS-Association/example\\_digital-twin\\_incubator/blob/master/software/incubator/datasets/README.md](https://github.com/INTO-CPS-Association/example_digital-twin_incubator/blob/master/software/incubator/datasets/README.md)

device, or it may be a value that changes over time (*e.g.*, sensor sensibility decay). Accurately calibrating these values is essential to obtain satisfactory results.

Despite these limitations, our proposed technique has shown promising results in detecting anomalies and improving the behavior of DTS. With further research and testing, we believe that this technique can be adapted and applied to a variety of systems, leading to significant benefits in terms of uncertainty mitigation.

## 6 Related work

There are three main lines of work closely related to our proposal. The first one concerns the uncertainty in simulation models of control systems. The second one is related to the explicit representation of uncertainty in software models. The third line deals with uncertainty in the twinned systems.

### 6.1 Uncertainty in control systems

Uncertainty in control systems and their simulation has been traditionally managed using different approaches.

One approach uses intervals to represent the possible values of uncertain attributes. For example, Fujimoto [22, 39] use time intervals to deal with the concepts of approximate time and event ordering in the context of DEVS [66]. Saadawi and Wainer also explored replacing time datatype in DEVS models by intervals in their RTA-DEVS formalism [51]. Other proposals provide methods to specify uncertainty in the state, input, and output variables in addition to the time variable [59, 16]. Works such as [33] make conservative decisions based on intervals to robustify the specification of controllers of cyber-physical systems so that they satisfy safety requirements under uncertain conditions. Reachability is the set of techniques for quantifying and propagating intervals [15]. Reachability analysis can be used for interval uncertainty quantification from noisy data and its propagation [64].

However, specifying and operating with intervals requires a significant effort by the modeler since there is no direct support for making computations with them, such as arithmetic operations or comparisons, which are burdensome and error-prone tasks. In addition, intervals provide a too coarse-grained and pessimistic representation of uncertainty, which results in very conservative (also called *cautious*) simulations [62].

Another set of papers studies the relationship between the uncertainty of the input parameters and that of the simulation results, in order to define measures for risk quantification under input uncertainty. In general, there are two sources of uncertainty in a typical stochastic simulation experiment: the extrinsic uncertainty on input parameters and the intrinsic uncertainty on output response (also called stochastic uncertainty) which reflects the inherent stochasticity of the system. Some authors [24, 67] propose nested Monte Carlo simulation approaches to estimate them [31]. Other works use the Kalman Filter for uncertainty mitigation and propagation [42]. However, in case of divergence, the Kalman filter will internally decide whether it believes the physical data or the model, and continue to average both values. Others [9] propose statistical methods for the calculation of confidence intervals for the mean of a simulation output. They get more accurate results than those that use interval arithmetic or very conservative (*i.e.*, robust) estimations. However, both the complexity of the calculations and their computational costs hinder their applicability. Finally, recent works such as [32] make use of random variables to represent uncertain attributes and uncertainty propagation is achieved using closed-form solutions mitigate these problems. In our work, we follow this latter approach.

## 6.2 Uncertainty in software models

The explicit representation of uncertainty is a well-known challenge. The survey [58] covers current approaches, for which significant challenges still remain. For example, there are very few programming or modeling language libraries that support measurement uncertainty, and even those that support uncertainty propagation (e.g., [36, 37, 1, 60]) do not provide the performance required to run simulations and do not support comparison between uncertain numbers. Uncertain comparisons are critical for adaptive systems since their decisions must be based on comparisons. In physical systems, logical variables representing comparisons between quantities rarely result in crisp true or false values. Instead, extensions of Boolean logic such as probability theory [21] are more appropriate. In this paper, we have used the Java library described in [2] that supports all basic primitive data types endowed with uncertainty.

## 6.3 Uncertainty in DTs

The treatment of uncertainty in the domain of DTs is gaining attention. Recent works aim at identifying the uncertainties of relevance in this domain, either in general [35, 46, 56] or focusing on particular application domains such as energy [13], industry 4.0 [65, 40] or automotive [38]. More specialized works, such as [63], discuss how uncertainty can be quantified and propagated to monitor safety properties using monte-carlo simulations. These works tend to focus on system or model uncertainties, and how the uncertainty of one can be mitigated by that of its twin counterpart. To the best of our knowledge, ours is the first work that takes into account both system and model uncertainties, recognizes that they are different in nature, and uses them in combination to reduce the overall system uncertainty.

# 7 Conclusions

In this paper, we have shown how the explicit representation of the sources of uncertainty of both the system and the models of a DTS can be used to improve its behavior. Furthermore, this enables a more accurate comparison of the behaviors of the physical and the digital twins. It also provides support to assess their validity and determine when the two behaviors are consistent or, on the contrary, diverge.

Our work can be continued along several lines. First, further experiments with different types of DTS will allow us to gain more confidence in the applicability and effectiveness of our proposal, identify possible limitations, and improve it. Second, dealing with other types of uncertainties, *e.g.*, epistemic ones, represents an interesting research challenge because it would require dealing with different logics and propagation mechanisms. Moreover, dealing with different types of uncertainties would require dealing also with their interactions, which is a problem in itself—the so-called *uncertainty interaction problem* [6, 7]. Finally, we also plan to use our approach within a validity framework for analyzing both models and experiments.

Related to our response above, . We now mention this in the paper, and refer to some recent works on to make readers aware of this problem

*Open research:* All the software, artifacts, and results described in the paper are available from <https://github.com/atenearesearchgroup/uncertainty-mitigation-dts>. The incubator case study is fully open source and available online: [https://github.com/INTO-CPS-Association/example\\_digital-twin\\_incubator](https://github.com/INTO-CPS-Association/example_digital-twin_incubator)

**Acknowledgment:** This work was initiated during the 18<sup>th</sup> CAMPaM workshop supported by Università di Corsica, Antwerpen University, and the Université Cote d’Azur.

**Funding:** It was partially funded by the DFG (German Research Foundation) – project number 499241390 (FeCoMASS); the Spanish Government (FEDER/Ministerio de Ciencia e Innovación–Agencia Estatal de Investigación) under projects PID2021-125527NB-I00 and TED2021-130523B-I00; the Poul Due Jensen Foundation, which has supported the establishment of the Center for Digital Twin Technology at Aarhus

University; The Natural Sciences and Engineering Research Council (NSERC) under grant RGPIN-2019-05213.

## References

- [1] Michaël Baudin et al. “OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation”. In: *Handbook of Uncertainty Quantification*. <https://openturns.github.io/>. Springer, 2016, pp. 1–38. ISBN: 978-3-319-11259-6. DOI: 10.1007/978-3-319-11259-6\_64-1.
- [2] Manuel F. Bertoa et al. “Incorporating measurement uncertainty into OCL/UML primitive datatypes”. In: *Softw. Syst. Model.* 19.5 (2020), pp. 1163–1189. DOI: 10.1007/s10270-019-00741-0.
- [3] Gary Bishop, Greg Welch, et al. “An Introduction to the Kalman Filter”. In: *Proc of SIGGRAPH, Course 8.27599-23175* (2001), p. 41.
- [4] Till Böttjer et al. “A review of unit level digital twin applications in the manufacturing industry”. In: *CIRP Journal of Manufacturing Science and Technology* 45 (2023), pp. 162–189. ISSN: 1755-5817. DOI: <https://doi.org/10.1016/j.cirpj.2023.06.011>. URL: <https://www.sciencedirect.com/science/article/pii/S1755581723000822>.
- [5] Radu Calinescu et al. “Understanding Uncertainty in Self-adaptive Systems”. In: *Proc. of ACSOS'20*. IEEE, 2020, pp. 242–251. DOI: 10.1109/ACSOS49614.2020.00047.
- [6] Javier Cámara et al. “The uncertainty interaction problem in self-adaptive systems”. In: *Softw. Syst. Model.* 21.4 (2022), pp. 1277–1294. DOI: 10.1007/S10270-022-01037-6.
- [7] Javier Cámara et al. “Uncertainty Flow Diagrams: Towards a Systematic Representation of Uncertainty Propagation and Interaction in Adaptive Systems”. In: *Proc. of SEAMS'24*. ACM, 2024, pp. 37–43. DOI: 10.1145/3643915.3644084.
- [8] Francois E. Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer, 2006, p. 643. ISBN: 9780387261027.
- [9] R. C. H. Cheng and W. Holland. “Calculation of Confidence Intervals for Simulation Output”. In: *ACM Trans. Model. Comput. Simul.* 14.4 (Oct. 2004), pp. 344–362. ISSN: 1049-3301. DOI: 10.1145/1029174.1029176.
- [10] Robert T. Clemen. “Combining forecasts: A review and annotated bibliography”. In: *International Journal of Forecasting* 5.4 (1989), pp. 559–583. DOI: 10.1016/0169-2070(89)90012-5.
- [11] Hugh W. Coleman. “Verification and validation in computational fluid dynamics and heat transfer using experimental uncertainty analysis concepts”. In: *Proc. of ASME Summer Heat Transfer Conference, HT'05*. Vol. 42. Springer, July 2005, pp. 1–2. DOI: 10.1115/HT2005-72800.
- [12] Manuela Dalibor et al. “A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins”. In: *J. Syst. Softw.* 193 (2022), p. 111361. DOI: 10.1016/j.jss.2022.111361.
- [13] Laya Das, Blazhe Gjorgiev, and Giovanni Sansavini. “Uncertainty-aware deep learning for digital twin-driven monitoring: Application to fault detection in power lines”. In: *CoRR* abs/2303.10954 (2023). DOI: 10.48550/arXiv.2303.10954.
- [14] Digital Twin Consortium. *Glossary of Digital Twins*. <https://www.digitaltwinconsortium.org/glossary/index.htm>. 2021.
- [15] Alexandre Donzé. “Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems”. In: *Computer Aided Verification*. Springer Berlin Heidelberg, 2010, pp. 167–170. DOI: 10.1007/978-3-642-14295-6\_17.
- [16] P. Dorato. “A historical review of robust control”. In: *IEEE Control Systems Magazine* 7.2 (Apr. 1987), pp. 44–47. DOI: 10.1109/mcs.1987.1105273.
- [17] Wilfried Elmenreich. “An Introduction to Sensor Fusion”. In: *Vienna University of Technology, Austria* 502 (2002), pp. 1–28.
- [18] Naeem Esfahani and Sam Malek. “Uncertainty in Self-Adaptive Software Systems”. In: *Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers*. Springer, 2013, pp. 214–238. ISBN: 978-3-642-35813-5. DOI: 10.1007/978-3-642-35813-5\_9.
- [19] Hao Feng et al. “Integration Of The Mape-K Loop In Digital Twins”. In: *2022 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, July 2022. DOI: 10.23919/annsim55834.2022.9859489.
- [20] Hao Feng et al. “Introduction to Digital Twin Engineering”. In: *2021 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, July 2021. DOI: 10.23919/annsim52504.2021.9552135.
- [21] Bruno de Finetti. *Theory of Probability: A critical introductory treatment*. John Wiley & Sons, 2017. ISBN: 9781119286370.
- [22] Richard Fujimoto. “Exploiting Temporal Uncertainty in Parallel and Distributed Simulations”. In: *Proc. of PADS'99*. IEEE Computer Society, 1999, pp. 46–53. DOI: 10.1109/PADS.1999.766160.
- [23] David Goldberg. “What every computer scientist should know about floating-point arithmetic”. In: *ACM Computing Surveys* 23.1 (Mar. 1991), pp. 5–48. DOI: 10.1145/103162.103163.

- [24] Michael B. Gordy and Sandeep Juneja. “Nested Simulation in Portfolio Risk Measurement”. In: *Management Science* 56 (Aug. 2010), pp. 1833–1848. DOI: 10.1287/mnsc.1100.1213.
- [25] Michael Grieves and John Vickers. “Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems”. In: *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Springer, 2017, pp. 85–113. ISBN: 978-3-319-38756-7. DOI: 10.1007/978-3-319-38756-7\_4.
- [26] Sebastian Hahner, Robert Heinrich, and Ralf Reussner. “Architecture-Based Uncertainty Impact Analysis to Ensure Confidentiality”. In: *IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. 2023, pp. 126–132. DOI: 10.1109/SEAMS59076.2023.00026.
- [27] Sebastian Hahner et al. “A Classification of Software-Architectural Uncertainty regarding Confidentiality”. In: *SECURITY*. 2022.
- [28] Randa Herzallah. “Uncertainty in control problems: A survey”. In: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, pp. 82–90. ISSN: 1474-6670. DOI: 10.3182/20050703-6-CZ-1902.01092.
- [29] Jerry L. Hintze and Ray D. Nelson. “Violin Plots: A Box Plot-Density Trace Synergism”. In: *The American Statistician* 52.2 (1998), pp. 181–184. DOI: 10.1080/00031305.1998.10480559.
- [30] JCGM 100:2008. *Evaluation of measurement data—Guide to the expression of uncertainty in measurement (GUM)*. Joint Com. for Guides in Metrology. 2008. URL: [https://www.bipm.org/documents/20126/2071204/JCGM\\_100\\_2008\\_E.pdf](https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf).
- [31] JCGM 101:2008. *Evaluation of measurement data—Supplement 1 to the “Guide to the expression of uncertainty in measurement”—Propagation of distributions using a Monte Carlo method*. Joint Com. for Guides in Metrology. 2008. URL: [http://www.bipm.org/utis/common/documents/jcgm/JCGM\\_101\\_2008\\_E.pdf](http://www.bipm.org/utis/common/documents/jcgm/JCGM_101_2008_E.pdf).
- [32] Jean-Marc Jézéquel and Antonio Vallecillo. “Uncertainty-aware Simulation of Adaptive Systems”. In: *ACM Trans. Model. Comput. Simul.* 33.3 (2023), 8:1–8:19. DOI: 10.1145/3589517.
- [33] Tsutomu Kobayashi et al. “Robustifying Controller Specifications of Cyber-Physical Systems Against Perceptual Uncertainty”. In: *Proc. of NASA Formal Methods 2021*. Vol. 12673. LNCS. Springer, 2021, pp. 198–213. DOI: 10.1007/978-3-030-76384-8\_13.
- [34] Štefan Kozák. “State-of-the-art in control engineering”. In: *Journal of Electrical Systems and Information Technology* 1.1 (2014), pp. 1–9. ISSN: 2314-7172. DOI: <https://doi.org/10.1016/j.jesit.2014.03.002>. URL: <https://www.sciencedirect.com/science/article/pii/S2314717214000038>.
- [35] Vinay Kulkarni et al. “Digital Twin as an Aid for Decision-Making in the Face of Uncertainty”. In: *Proc. of WSC’22*. IEEE, 2022, pp. 1371–1385. DOI: 10.1109/WSC57314.2022.10015528.
- [36] Eric O. Lebigot. *Uncertainties package*. <https://pythonhosted.org/uncertainties/>. Accessed: May 30, 2022. 2016.
- [37] Abraham Lee. *SOERP Uncertainties package*. <https://pypi.org/project/soerp/>. Accessed: May 30, 2022. 2013.
- [38] Linyu Lin, Han Bao, and Nam Dinh. “Uncertainty Quantification and Software Risk Analysis for Digital Twins in the Nearly Autonomous Management and Control Systems: A Review”. In: *CoRR* abs/2103.03680 (2021). arXiv: 2103.03680. URL: <https://arxiv.org/abs/2103.03680>.
- [39] Margaret L. Loper and Richard M. Fujimoto. “Pre-sampling as an approach for exploiting temporal uncertainty”. In: *Proc. of PADS’00*. IEEE Computer Society, 2000, pp. 157–164. DOI: 10.1109/PADS.2000.847159.
- [40] Dan Luo, Simon Thevenin, and Alexandre Dolgui. “A Digital Twin-Driven Methodology for Material Resource Planning Under Uncertainties”. In: *Proc. of APMS’21*. Springer, 2021, pp. 321–329. DOI: 10.1007/978-3-030-85874-2\_34.
- [41] Ali Mesbah. “Stochastic model predictive control with active uncertainty learning: A Survey on dual control”. In: *Annu. Rev. Control.* 45 (2018), pp. 107–117. DOI: 10.1016/j.arcontrol.2017.11.001.
- [42] I. Mezic. “Coupled Nonlinear Dynamical Systems: Asymptotic Behavior and Uncertainty Propagation”. In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 2. Dec. 2004, 1778–1783 Vol.2. DOI: 10.1109/CDC.2004.1430303.
- [43] Marvin Minsky. *Matter, mind and models*. MIT Press, 1965.
- [44] Gabriel A. Moreno et al. “Uncertainty reduction in self-adaptive systems”. In: *Proc. of SEAMS@ICSE’18*. Ed. by Jesper Andersson and Danny Weyns. ACM, 2018, pp. 51–57. DOI: 10.1145/3194133.3194144.
- [45] William L. Oberkamp et al. “Error and uncertainty in modeling and simulation”. In: *Reliability Engineering & System Safety* 75.3 (2002), pp. 333–357. ISSN: 0951-8320. DOI: 10.1016/S0951-8320(01)00120-X.
- [46] Flávio Oquendo. “Dealing with Uncertainty in Software Architecture on the Internet-of-Things with Digital Twins”. In: *Proc. of ICCSA’19*. Vol. 11619. LNCS. Springer, 2019, pp. 770–786. DOI: 10.1007/978-3-030-24289-3\_57.
- [47] Randy Paredis, Cláudio Gomes, and Hans Vangheluwe. “Towards a Family of Digital Model/Shadow/Twin Workflows and Architectures”. In: *Proc. of IN4PL’21*. SCITEPRESS, 2021, pp. 174–182. DOI: 10.5220/0010717600003062.
- [48] Svetlozar T. Rachev et al. “Probability Distances and Probability Metrics: Definitions”. In: *The Methods of Distances in the Theory of Probability and Statistics*. Springer, 2013, pp. 11–31. ISBN: 978-1-4614-4869-3. DOI: 10.1007/978-1-4614-4869-3\_2.

- [49] Andres J. Ramirez, Adam C. Jensen, and Betty H. C. Cheng. “A taxonomy of uncertainty for dynamically adaptive systems”. In: *7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2012*. IEEE Computer Society, 2012. ISBN: 978-1-4673-1788-7. DOI: 10.1109/SEAMS.2012.6224396.
- [50] Marvin Rausand. *Risk Assessment: Theory, Methods, and Applications*. John Wiley & Sons, 2013. ISBN: 9780470637647.
- [51] Hesham Saadawi and Gabriel A. Wainer. “Rational time-advance DEVS (RTA-DEVS)”. In: *Proc. of SpringSim’10*. SCS/ACM, Apr. 2010, 143:1–143:8. DOI: 10.1145/1878537.1878686.
- [52] Andrew JE Seely and Peter T Macklem. “Complex systems and the technology of variability analysis”. In: *Critical Care* 8 (2004), pp. 367–384. DOI: 10.1186/cc2948.
- [53] Pedro da Silva Hack and Carla Schwengber ten Caten. “Measurement Uncertainty: Literature Review and Research Trends”. In: *IEEE Trans. Instrumentation and Measurement* 61.8 (2012), pp. 2116–2124. DOI: 10.1109/TIM.2012.2193694.
- [54] Fei Tao et al. “Digital twin driven prognostics and health management for complex equipment”. In: *CIRP Annals* 67.1 (2018), pp. 169–172. ISSN: 0007-8506. DOI: <https://doi.org/10.1016/j.cirp.2018.04.055>.
- [55] Fei Tao et al. “Digital Twin in Industry: State-of-the-Art”. In: *IEEE Trans. Ind. Informatics* 15.4 (2019), pp. 2405–2415.
- [56] Adam Thelen et al. “A Comprehensive Review of Digital Twin - Part 2: Roles of Uncertainty Quantification and Optimization, a Battery Digital Twin, and Perspectives”. In: *CoRR* abs/2208.12904 (2022). DOI: 10.48550/arXiv.2208.12904. arXiv: 2208.12904. URL: <https://doi.org/10.48550/arXiv.2208.12904>.
- [57] Daniel P. Thunnissen. “Uncertainty classification for the design and development of complex systems”. In: *Proc. of the 3rd Annual Predictive Methods Conference, Veros Software*. 2003. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.128.133>.
- [58] Javier Troya et al. “Uncertainty representation in software models: a survey”. In: *Softw. Syst. Model.* 20.4 (2021), pp. 1183–1213. DOI: 10.1007/s10270-020-00842-1.
- [59] Damián Vicino, Gabriel A. Wainer, and Olivier Dalle. “Uncertainty on Discrete-Event System Simulation”. In: *ACM Trans. Model. Comput. Simul.* 32.1 (2022), 2:1–2:27. DOI: 10.1145/3466169.
- [60] Wikipedia. *List of uncertainty propagation software*. Accessed: January 2024. URL: [https://web.archive.org/web/20230326201711/https://en.wikipedia.org/wiki/List\\_of\\_uncertainty\\_propagation\\_software](https://web.archive.org/web/20230326201711/https://en.wikipedia.org/wiki/List_of_uncertainty_propagation_software).
- [61] Robert L. Winkler and Spyros Makridakis. “The Combination of Forecasts”. In: *Journal of the Royal Statistical Society* 146.2 (1983), pp. 150–157. DOI: 10.2307/2982011.
- [62] B. Wittenmark. “Stochastic adaptive control methods: a survey”. In: *International Journal of Control* 21.5 (1975), pp. 705–730. DOI: 10.1080/00207177508922026.
- [63] Jim Woodcock et al. “Uncertainty Quantification and Runtime Monitoring Using Environment-Aware Digital Twins”. In: *Leveraging Applications of Formal Methods, Verification and Validation: Tools and Trends*. Springer International Publishing, 2021, pp. 72–87. DOI: 10.1007/978-3-030-83723-5\_6.
- [64] Thomas Wright, Cláudio Gomes, and Jim Woodcock. “Formally Verified Self-adaptation of an Incubator Digital Twin”. In: *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2022, pp. 89–109. DOI: 10.1007/978-3-031-19762-8\_7.
- [65] Qinghua Xu et al. “Uncertainty-aware transfer learning to evolve digital twins for industrial elevators”. In: *Proc. of ESEC/FSE’22*. ACM, 2022, pp. 1257–1268. DOI: 10.1145/3540250.3558957.
- [66] Bernard Phillip Zeigler, Alexandre Muzy, and Ernesto Kofman. *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*. 3rd ed. Academic Press, 2018. ISBN: 978-0128133705.
- [67] Helin Zhu, Tianyi Liu, and Enlu Zhou. “Risk Quantification in Stochastic Simulation under Input Uncertainty”. In: *ACM Trans. Model. Comput. Simul.* 30.1 (Feb. 2020), 1:1–1:24. ISSN: 1049-3301. DOI: 10.1145/3329117.
- [68] Hans-Jürgen Zimmermann. *Fuzzy Set Theory – and Its Applications*. Springer Science+Business Media, 2001.