INTRODUCTION TO DIGITAL TWIN ENGINEERING

Hao Feng
Cláudio Gomes
Casper Thule
Kenneth Lausdahl
Alexandros Iosifidis
Peter Gorm Larsen

DIGIT, Department of Electrical and Computer Engineering
Aarhus University
Åbogade 34, Aarhus N, DENMARK
{haof,claudio.gomes,casper.thule,lausdahl,ai,pgl}@ece.au.dk

Abstract

Cyber-Physical Systems (CPSs) are getting increasingly complex and generate large amounts of data. Analyzing such data provides us an insight into a given system. The digital twin concept emerges as an attempt to seamlessly integrate the data and insight in order to improve system performances. It enables applications such as visualization, monitoring, state estimation, and self-adaptation. In this paper, we demonstrate the construction of a digital twin exemplified by an incubator system, including the benefits and challenges of each application. The result is a description of the building blocks of a digital twin.

Keywords: cyber-physical systems, digital shadow, digital twin, self-adaptive system, monitoring

1 INTRODUCTION

Due to performance demand and various requirements, large amounts of data can be gathered from CPSs for analysis such as smart control and monitoring. In order to seamlessly integrate the data and the insight gained from the analysis, the concept of Digital Twin (DT) (Grieves and Vickers 2017) has emerged. Various definitions of DTs (Liu et al. 2021, Rasheed et al. 2020, Tao et al. 2019) have been proposed since it was first presented by Michael Grieves in 2003 (Tao et al. 2019). In this paper, we give our summary. We view a DT as a system which incorporates different techniques to increase the values of the physical system. As summarized in Figure 1, if the communication between the physical system and digital system is bi-directional meaning that data can flow automatically from the physical system to the digital system and vice versa, we call such the digital system a Digital Twin (DT) and the physical system a Physical Twin (PT). On the other hand, we call the digital system a Digital Shadow (DS) when the data only flows automatically from the PT to the digital system. Naturally, in a DS, data may flow back as well in the broad sense of the word (as in, a human takes action), but this is not done directly.

The availability of such data and accompanying analysis potentially enable applications such as visualizations (Ward et al. 2010), monitoring (Gawand et al. 2015), state estimation (Dehghanpour et al. 2019), anomaly detection (Chandola et al. 2009), what-if (co)-simulation (Rizzi 2009), and self-adaptation (Zhou

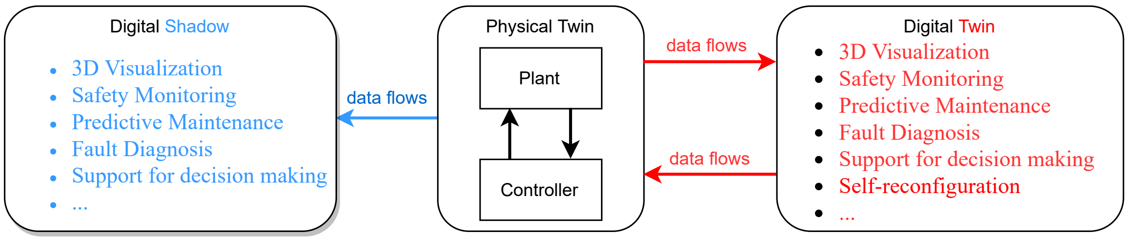


Figure 1: Communications inside a DS and DT. The blue and red arrows represent automated data flows of a DS and a DT respectively.

et al. 2019). These applications have the potential to improve the performance of a PT in many aspects. For example, if an expert desires to determine the optimal parameters for a controller in a PT, he can run what-if (co)-simulations using the DS, to obtain performance results based on candidate parameter sets. The expert can then reconfigure the parameters according to the results. However, a DS has its own limitations. One of the drawbacks of a DS is that the data only flows from the PT to the DS. This means a DS cannot automatically affect the PT. The DT does not have such limitation.

By demonstrating the benefits and challenges inside a DS and DT, we hope to better understand a DS and a DT, give an introduction to a DS and a DT, and provide a good validation case for the community. Thus we discuss some of the building blocks of a DS and DT, using a simple but representative incubator system as running example. Based on this system, we introduce the communication, analysis of data, and applications inside a DS and DT.

The rest of the paper is organized as follows: Section 2 gives an introduction to the incubator PT. Based on the PT, we discuss parts of features and challenges inside the DS. In order to overcome the drawbacks of the DS, we transform the DS to a DT in section 3 and discuss benefits and challenges of a DT. Finally, we conclude and discuss further directions in section 4.

2 INCUBATOR DIGITAL SHADOW

In order to construct a DS and a DT, we first developed an incubator system to serve as a PT. The systematic diagram of the incubator is shown in fig. 2. The incubator is a system with the ability to hold a desired temperature within an insulated container. The controller for the incubator is similar to a bang-bang controller, with a small difference described below. The controller receives data from the sensors of the incubator, turns on/off the heatbed (a heating unit) and a fan. Due to the delayed temperature propagation effect of the heatbed, the controller needs to wait after each actuation, to make sure the temperature does not rise too much. The details of the incubator system are in Feng et al. (2021).

Based on the PT, we implemented the DS by deploying a communication channel for data transmission from the PT to the DS and creating a model of the PT. Such DSs increase the value of their PT counterparts but not without challenges. In this section, we are going to discuss the features and challenges inside the DS.

2.1 Communication and Data Storage

Communication from a PT to the DS needs to tackle problems such as point to point communication, multi clients to multi clients communication, messages routing, packet dropout, service failure, load balancing, and so on. In addition, the communication delays should be minimized. Delays occur while exchanging information from a PT to the DS, due to a limited communication channel capacity. For example, real-time monitoring could fail with disastrous consequences if the delays violate the required tolerances.

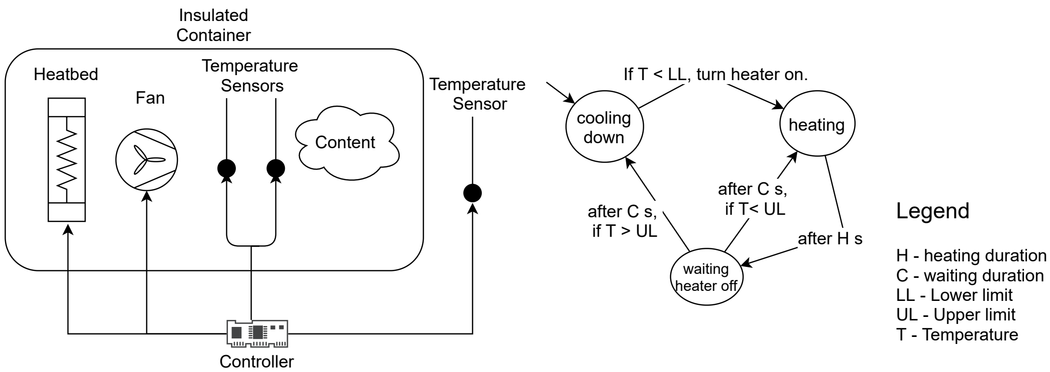


Figure 2: Schematic overview of the incubator. The left is the incubator system. The right is the statechart of the controller.

Many different technologies already exist for tackling the communication issues, i.e. RabbitMQ (https://www.rabbitmq.com/), Ditto (https://www.eclipse.org/ditto/), Apache Kafka(https://kafka.apache.org/), and https://www.rti.com/. In our case, we chose RabbitMQ to serve as a data broker because of its lightweight and easy deployment on premise.

Regarding data storage, we used InfluxDB for its ease of use, its bindings to different programming languages, and its simple visualization facilities. For other technologies and tools for data storage, we refer the reader to Padgavankar and Gupta (2014), Mazumdar et al. (2019).

2.2 Data Visualization

Humans use sight as one of a key senses for understanding information (Ward et al. 2010). Recent advances in tools for creating visual interfaces such as Unity (https://unity.com/), Qt (https://www.qt.io/), Grafana, Dash, Gazebo, and so on, have made it easier to create intuitive and visual interfaces for a DS. One of the challenges in DT engineering is the rapid construction of these interfaces from a PT. Such interfaces should, for example, allow the user to: 1) Selectively visualize the 3D PT and its state; 2) Create dashboards to plot the most important data; 3) Replay past behavior of the PT; 4) Spawn new what-if simulations; 5) Run optimizations; 6) Display predictive maintenance results;

When faced with big data either collected from simulations or a PT, data visualization needs to provide perceptual scalability, real-time scalability, and interactive scalability (Agrawal et al. 2015). More details about data visualization can be found in Chan (2006).

For the incubator case study, we used the simple visualization capability of InfluxDB to create a dashboard showing the temperature data from the sensors, the state of the fan and the heatbed, and the parameters of the controller and the control states. Figure 3 shows our visualization results.

2.3 Modelling and Calibration

Models are used to describe behaviors of interest, often involving equations combined with parameters. Such models lay the foundation for other applications such as state estimation and monitoring. Modeling might be a challenging work since it involves different knowledges in terms of the behaviors of interest. Without a better comprehension of a system, it could be difficult to build a high fidelity model.

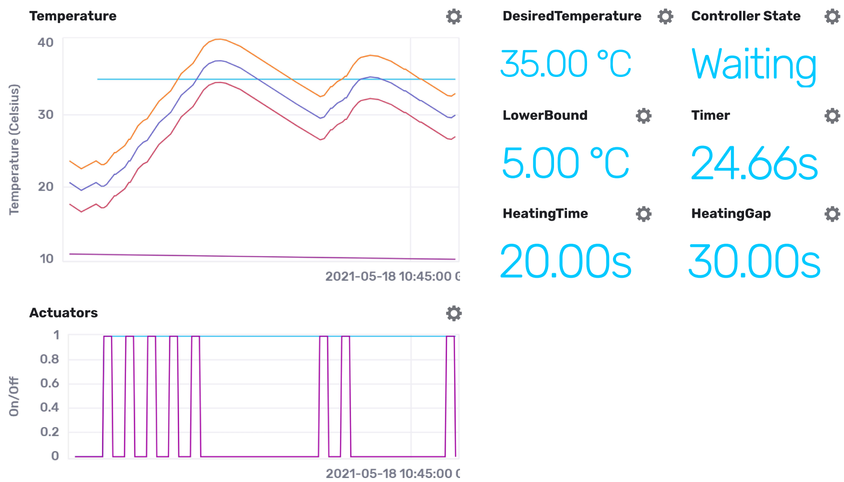


Figure 3: Visualization interface of incubator PT. The horizontal axis represents time. The plots below shows the state the heatbed (purple line) and the fan (blue line). The plots above show the sensor temperatures and the desired temperature. The right section shows the parameters and states of the controller.

An important part of employing models in a DS context is calibration. Calibration is the process of adjusting certain model parameters to make the behaviors of the model fit the behaviors of the PT (Williams and Esteves 2017). Without calibration, a model may not be able to predict the behaviors of interest correctly.

The success of calibration depends not only on the data but also on the fidelity of a model. Regarding data from a PT, we highlight the problem of noisy measurements. Because of noise, inferred parameter values have a stochastic component. Moreover, the structure of the model may be such that some parameters only have an effect on the model behavior if particular inputs are given. So it may be the case that, under certain inputs, some parameters may never properly be found by calibration. In the state of the art, techniques such as noise filtering (Rhodes 1971) and design of experiments (Telford 2007) can be of assistance.

Based on different goals of a DS, we may choose models with various fidelity. High fidelity models generally results a satisfied calibration while it may be time-consuming. Although relative low fidelity models are with less accuracy, they have the advantages of simulation speed. In addition, the accuracy of simulations also affects calibration results because some parts of the methods compare the trajectories of simulations with measured trajectories. For more details on system identification, we refer the reader to the vast literature on the topic (Isermann and Münchhof 2010, Keesman 2011, Nelles 2013).

We developed two models for the incubator PT. One contains two parameters without the assumption that the heatbed retains heat energy. Another consists of four parameters with the assumption. The calibration results showed that the model with four parameters performs better than the model with two parameters. The four parameter model is given by

$$\frac{dT_{heater}}{dt} = \frac{1}{C_{heater}}(VI\Delta t - G_{heater}(T_{heater} - T_{bair})) \tag{1a}$$

$$\frac{dT_{bair}}{dt} = \frac{1}{C_{air}}[G_{heater}(T_{heater} - T_{bair}) - G_{box}(T_{bair} - T_{room})]. \tag{1b}$$

The four unknown parameters are C_{heater}, C_{air} (unit:J K^{-1}), G_{heater}, and G_{box} (unit:J K^{-1}), where C convert the changing rate of temperature to energy and G determines the proportion of energy converted from the difference of temperatures. T_{heater} represents the temperature in the heatbed and T_{bair} the temperature of the air inside the incubator.

In order to calibrate the parameters we collected data from the incubator PT and utilized a non-linear least square solver. The details about modeling and calibration can be accessed in Feng et al. (2021). The calibration results are shown in fig. 4, which shows that our model aligns with the PT under current conditions. The purpose of the need of the alignment is to describes behaviors of the PT using the model so that we can use the model to replace the PT for some analysis. This involves the generalization of the model, which means we need to figure out on what conditions the model can predict the behaviors of interest correctly, when the model is invalid.

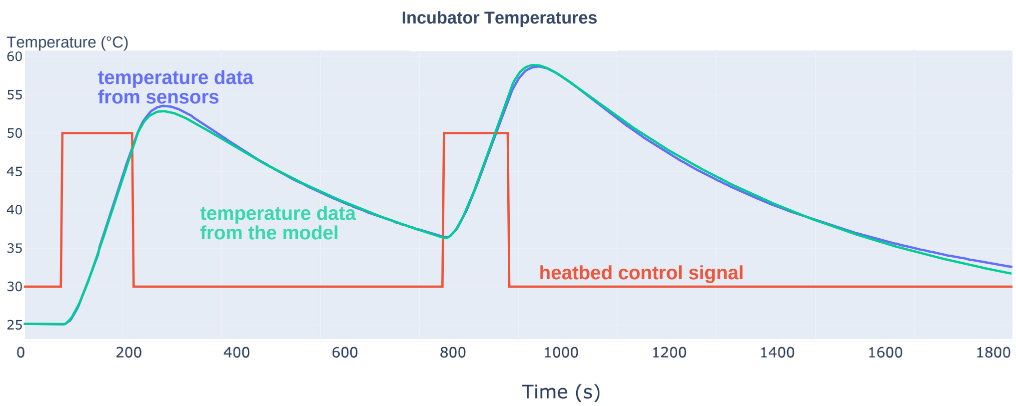


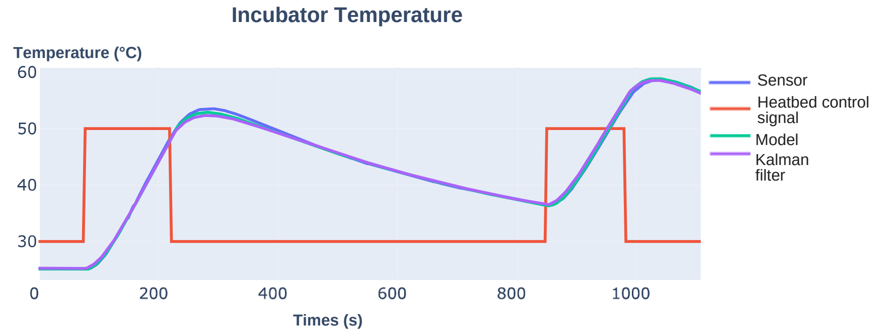
Figure 4: Calibration results of (1). The blue line is the average temperature representing the air temperature inside the incubator, while the green line represents the results from our calibrated model. The read line shows the two states of the heatbed, on and off.

2.4 State Estimation

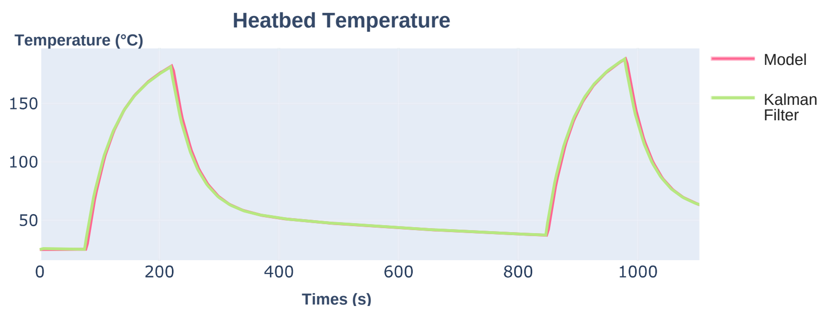
State represents the internal condition or status of a system at a given instant of time (Simon 2006). Those states can be static or changing with time. Take the model in (1) for example, the states are the temperature of the air inside the incubator and the temperature of the heatbed. If electrical power flows into the heatbed, then the heatbed heats the air. Such a power flow changes the states of the incubator system.

State estimation is the use of different methods to estimate the states of interest that cannot be obtained directly through measured data. For example, in the incubator system, we need to estimate the temperature of the heatbed. However this cannot be obtained directly through the sensors. In addition, the data from sensors is contaminated by noise.

One of common state estimation algorithms is Kalman Filter (KF). It combines the predicted behaviors produced by a dynamic model with the multiple sequential measurements from sensors, to form an estimate of the system's varying state, that is better than the estimate obtained by only using measurements. By combining (1) after calibration and the measured temperature data of the air inside the incubator, we used a KF to estimate the states of the heatbed and the air. As seen in fig. 5, the KF gives the estimated stated of the heatbed and the air temperature inside the incubator, while attempting to minimize the expected variance of the estimation error. For more details and alternative state estimators, we refer the reader to Simon (2006).



(a) State estimation results of air temperature inside the box. The blue line is the average air temperature inside the incubator from the sensory data. The green line represents the same temperature but it is generated from (1). The purple line showing the same air temperature results from the KF.



(b) State estimation results of the heatbed. The pink line is the heatbed temperature from (1) and the light green line from KF.

Figure 5: State estimation results.

2.5 Monitoring

Monitoring in context of a DS is the act of observing and evaluating the behaviors of a PT as it operates (Bartocci et al. 2018). One of the benefits of monitoring in the DS context is 3D visualization of a PT since 3D visualization might be helpful for humans to understand compared with long textual listings of data, depending on the nature of a PT.

We distinguish between online and offline monitoring. Offline monitoring stores an entire observation trace of the PT, which is periodically examined, while online monitoring is performed as data is received (Gawand et al. 2015). One important feature of offline monitoring is replaying. By replaying a situation where some kind of error has been logged using monitoring techniques, it can be easier to understand why it appeared.

In an online monitoring scenario, it can be advantageous to monitor desirable properties e.g., in relation to the safety of a PT to its environment, it can be very valuable to be able to monitor when the discrepancy between the monitored and predicted behaviors start and provide supporting decision makings from the DS side if there is an expert taking the final decision. To address some of the these challenges, we refer the reader to the state of the art in run-time monitoring (Yilin Mo et al. 2012, Paridari et al. 2018).

In order to realize online monitoring for the incubator PT, we need sensory data, models, calibration, and state estimation. The sensory data provides the information for the air temperature inside the incubator. In addition, the heatbed temperature cannot be measured directly, thus the state estimation implemented by KF gives us the state of the heatbed. These results are then plotted in the visualization interface (recall fig. 3).

In addition, monitoring techniques can be used for anomaly detection. Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior (Chandola et al. 2009). Anomaly detection is not trivial due to noise, availability of labelled data, evolution of normal behaviors, and so on. Different technologies from diverse disciplines such as machine learning, data mining, and statistics can be applied for anomaly detection. More information is available in Ariyaluran Habeeb et al. (2019), Braei and Wagner (2020), Kwon et al. (2019).

In our work, we used KF for basic anomaly detection by combining the measured behaviors from the PT and the predicted behaviors. In order to showcase an anomaly, we opened the lid of incubator system because the model we used to do the state estimation only represents the behaviors of the system with a closed lid. Opening it causes the system to violate the physical principles the model was originally built on, and therefore the KF will fail to track what is happening. The results are illustrated in fig. 6. We repeated the opening and closing of the lid twice. As seen, when we opened the lid, the purple line and the blue line shows the discrepancy, showing the anomaly. After closing the lid, the physical principles of the model comply with the situation, thus the two lines merge gradually.

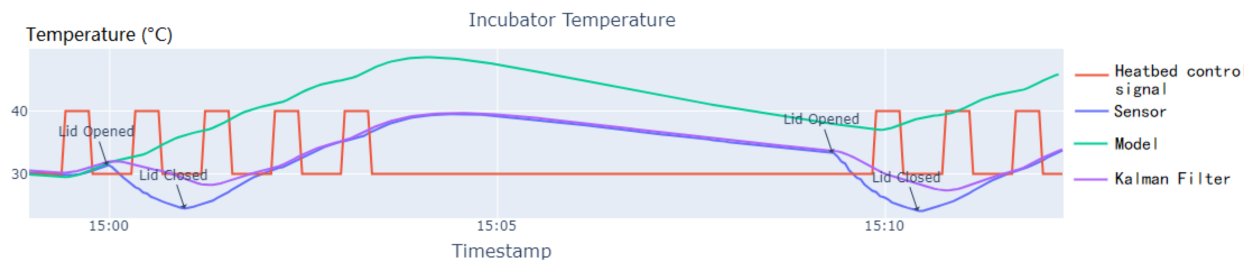


Figure 6: Results of KF for anomaly detection.

2.6 What-If Simulation

What-if simulation is a data-intensive simulation whose goal is to inspect the behavior of the PT under some given hypotheses called scenarios (Rizzi 2009). In cases where violations of desirable properties have been detected during monitoring of the PT behavior, a what-if simulation would enable to a human operator to try alternative interventions purely in a virtual setting to inspect what the consequences would be, *before* taking a final decision about what intervention would be best. This naturally requires simulation of the different alternatives to be faster than real-time. In cases where the simulations consist of multiple sub-models, a co-simulation (Gomes et al. 2018) should be used. In the context of DS, compared to an offline simulation, the what-if simulation is endowed with the ability to access the historical data directly from a DS, synchronize the simulation with the PT, and/or automatically incorporate the simulation results into other services running in the DS.

What-if simulation provides insights for optimizing the PT configurations to achieve the desired goal. However, this can in particular be vulnerable in cases where the consequences of wrong choices can be fatal, for example in relation to safety constraints.

We used what-if simulation to determine the optimal configurations for the parameter H in the controller described in fig. 2. The scenario is that we need to make an intervention in the incubator, thus it is necessary to cut off the power for a period time. We need to determine the largest available time period (the time between two extreme points) for the intervention, if we allow a slight overheat of the incubator. The result is shown in fig. 7. The red line gives us the largest available time period but it overheats excessively, thereby violating the slight overheat allowance. The green line has the largest available time period compared with the yellow and blue line and the parameter is what we need.

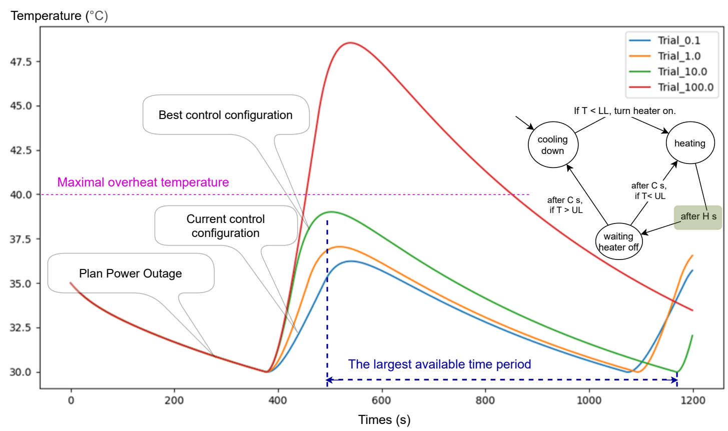


Figure 7: Results of what-if simulation for controller configuration. The horizontal and vertical axes represent time and temperature respectively.

3 DIGITAL TWIN

In section 2, we introduced features inside a DS some of which, such as anomaly detection and what-if (co)-simulation, provide insights for decision making. However, the DS cannot automatically reconfigure the system based on those results. This leads to the need of a DT. In this section, we discuss the communication and self-adaptation in a DT, including the benefits and challenges.

3.1 Bidirectional Communication

In order to transform a DS to a DT, it is necessary to enable bidirectional communication, i.e. data flow from a PT to a DT and vice versa. In addition, the PT should make some changes for receiving data e.g., we updated the controller to receive messages from the DT and reconfigure the parameters. An example of a DT is shown in fig. 8. A configurator feature is added to the PT to make it configurable, and a configuration service is added to the digital shadow, in order to turn it into a DT.

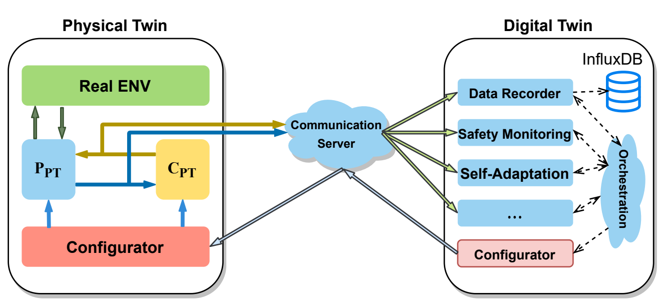


Figure 8: Schematic overview of a CPS-based Digital Twins. The data collected from the PT are sent to the DT through the communication server for further utilization such as safety monitoring, self-adaptation. In addition, the DT can send feedbacks to the PT via the communication server.

There are many technologies that can be used to realize bidirectional communication. For example, Ditto and RabbitMQ as mentioned above. Enabling bi-directional communication raises the important challenge of ensuring cyber-security (Humayed et al. 2017, Kumar et al. 2020, Zacchia Lun et al. 2019). Moreover, legacy systems cannot simply be connected to the internet because such systems have not received the necessary software patches to improve security, and doing so would mean significant down-time. Regarding the incubator, we mitigated this issue by limiting the range of the parameters for the controller of the incubator PT.

3.2 Self-Adaptation

Self-adaptation is the ability of a computer system to change parts all of its working algorithm over time (Sari and Sopuru 2021). With the rapid increasing complexity of CPSs, understanding the massive data from a PT and deciding the proper response to the PT are non trivial for human (Zhou et al. 2019).

The DT can be used to implement self-adaptation and process large amounts of data in a timely manner. However, self-adaptation procedures can be very intricate, rely heavily on domain knowledge, and are application specific (e.g., a self-adaptation loop for the incubator is very different than the self-adaptation loop for a robot manipulator). Incidentally, the people who know best how to reconfigure the system, are not necessarily the ones that can code those self-adaptation loops. Therefore, languages and frameworks (Keskisärkkä 2014) that enable domain experts to change the DT can play an important role.Moreover, it is crucial that new configuration, and the act of moving the system from the current configuration to the new one, are safe (Cámara et al. 2013, Weyns 2019).Techniques such as reachability analysis (Kurzhanski and Varaiya 2000, Asarin et al. 2000) can play a role here.

We give, as example, the following self-adaptation loop applied to the incubator. Those steps are finished automatically by the aid of the orchestration services in fig. 8.

1) Start when an anomaly is detected. This could be, e.g., due to an object (e.g., a bucket of ice) being placed in the incubator. The KF and anomaly detector can be used to detect the changes of the system.
2) Schedule an experiment to gather relevant data. The nature of this experiment is application specific, and design of experiments can be used for this (Pronzato and Pázman 2013).
3) Configure controller to schedule the new experiment.
4) Gather experiment data, using the data recorder.
5) Run parameter estimation for new experiments.
6) Re-configure KF with new parameters.
7) Run What-if simulations to optimize controller behaviors.
8) Re-configure controller.

4 CONCLUSION AND FUTURE WORK

In this paper, we introduced digital twin engineering in the context of an incubator PT. By bridging the connection from the PT to the DS, we described features such as visualizations, state estimation, monitoring, anomaly detection, and what-if (co)-simulation, and how they are implemented in the incubator. We showed that going from a DS to a DT is a complex endeavour, that raises many challenges in security and safety.

It is easy to see that the DS features are mostly application independent, requiring the use of models. This makes it possible to study DS engineering and to devise frameworks to quickly produce DSs. Engineering a DT is, however, more challenging, because many self-adaptation loops are application dependent, as we exemplified.

As limitation, we did not consider the important challenge in data transformation and correlation, from multiple sources (including cross company data sources), since such a challenge is difficult to reproduce with the incubator example. In the future, we will research the necessary abstractions in order to describe the coordination of DS services required to implement self-adaptation loops.

ACKNOWLEDGMENTS

We are grateful to the Poul Due Jensen Foundation, which has supported the establishment of a new Centre for Digital Twin Technology at Aarhus University. We are thankful for the discussions with Yuan Zhao, funded by Natural Science Foundation of Tianjin 20JCQNJC0039. Hao Feng also acknowledges support from China Scholarship Council. Finally, we thank Till Böttjer and the anonymous reviewers for their valuable review of this manuscript.

REFERENCES

Agrawal, R., A. Kadadi, X. Dai, and F. Andres. 2015. "Challenges and Opportunities with Big Data Visualization". In *Proceedings of the 7th International Conference on Management of Computational and Collective intElligence in Digital EcoSystems*, pp. 169–173. New York, NY, USA, Association for Computing Machinery.

Ariyaluran Habeeb, R. A., F. Nasaruddin, A. Gani, I. A. Targio Hashem, E. Ahmed, and M. Imran. 2019. "Real-Time Big Data Processing for Anomaly Detection: A Survey". *International Journal of Information Management* vol. 45, pp. 289–307.

Asarin, E., O. Bournez, T. Dang, and O. Maler. 2000. "Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems". In *Hybrid Systems: Computation and Control*, edited by G. Goos, J. Hartmanis, J. van Leeuwen, N. Lynch, and B. H. Krogh, Volume 1790, pp. 20–31. Berlin, Heidelberg, Springer Berlin Heidelberg.

Bartocci, E., J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, and S. Sankaranarayanan. 2018. "Specification-Based Monitoring of Cyber-Physical Systems: A Survey on Theory, Tools and Applications". In *Lectures on Runtime Verification*, edited by E. Bartocci and Y. Falcone, Volume 10457, pp. 135–175. Cham, Springer International Publishing.

Braei, M., and S. Wagner. 2020. "Anomaly Detection in Univariate Time-Series: A Survey on the State-of-the-Art". *arXiv:2004.00433 [cs, stat]*.

Cámara, J., R. de Lemos, C. Ghezzi, and A. Lopes. (Eds.) 2013. *Assurances for Self-Adaptive Systems: Principles, Models, and Techniques*, Volume 7740. Berlin, Heidelberg, Springer Berlin Heidelberg.

Chan, W. W.-Y. 2006. "A survey on multivariate data visualization". *Department of Computer Science and Engineering. Hong Kong University of Science and Technology* vol. 8 (6), pp. 1–29.

Chandola, V., A. Banerjee, and V. Kumar. 2009. "Anomaly Detection: A Survey". *ACM Computing Surveys* vol. 41 (3), pp. 15:1–15:58.

Dehghanpour, K., Z. Wang, J. Wang, Y. Yuan, and F. Bu. 2019. "A Survey on State Estimation Techniques and Challenges in Smart Distribution Systems". *IEEE Transactions on Smart Grid* vol. 10 (2), pp. 2312–2322.

Feng, H., C. Gomes, C. Thule, K. Lausdahl, M. Sandberg, and P. G. Larsen. 2021. "The Incubator Case Study for Digital Twin Engineering". *arXiv:2102.10390 [cs, eess]*.

Gawand, H. L., A. K. Bhattacharjee, and K. Roy. 2015. "Online Monitoring of a Cyber Physical System against Control Aware Cyber Attacks". In *Procedia Computer Science*, Volume 70, pp. 238–244, Elsevier B.V.

Gomes, C., C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe. 2018. "Co-Simulation: A Survey". *ACM computing surveys* vol. 51 (3), pp. 1–33.

Grieves, M., and J. Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In *Transdisciplinary Perspectives on Complex Systems: New Findings and Ap-*

proaches, edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves, pp. 85–113. Cham, Springer International Publishing.

Humayed, A., J. Lin, F. Li, and B. Luo. 2017. "Cyber-Physical Systems Security—A Survey". *IEEE Internet of Things Journal* vol. 4 (6), pp. 1802–1831.

Isermann, R., and M. Münchhof. 2010. *Identification of Dynamic Systems: An Introduction with Applications*. Springer Science & Business Media.

Keesman, K. J. 2011. *System Identification: An Introduction*. Springer Science & Business Media.

Keskisärkkä, R. 2014. "Semantic Complex Event Processing for Decision Support". In *The Semantic Web – ISWC 2014*, edited by P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, pp. 529–536. Cham, Springer International Publishing.

Kumar, C., S. Marston, and R. Sen. 2020. "Cyber-Physical Systems (CPS) Security: State of the Art and Research Opportunities for Information Systems Academics". *Communications of the Association for Information Systems* vol. 47 (1).

Kurzhanski, A. B., and P. Varaiya. 2000. "Ellipsoidal Techniques for Reachability Analysis". In *Hybrid Systems: Computation and Control*, edited by G. Goos, J. Hartmanis, J. van Leeuwen, N. Lynch, and B. H. Krogh, Volume 1790, pp. 202–214. Berlin, Heidelberg, Springer Berlin Heidelberg.

Kwon, D., H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. 2019. "A Survey of Deep Learning-Based Network Anomaly Detection". *Cluster Computing* vol. 22 (1), pp. 949–961.

Liu, M., S. Fang, H. Dong, and C. Xu. 2021. "Review of Digital Twin about Concepts, Technologies, and Industrial Applications". *Journal of Manufacturing Systems* vol. 58, pp. 346–361.

Mazumdar, S., D. Seybold, K. Kritikos, and G. Verginadis. 2019. "A Survey on Data Storage and Placement Methodologies for Cloud-Big Data Ecosystem". *Journal of Big Data*.

Nelles, O. 2013. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer Science & Business Media.

Padgavankar, M. H., and S. R. Gupta. 2014. "Big Data Storage and Challenges". *International Journal of Computer Science and Information Technologies* vol. 5 (2), pp. 2218–2223.

Paridari, K., N. O'Mahony, A. El-Din Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg. 2018. "A Framework for Attack-Resilient Industrial Control Systems: Attack Detection and Controller Reconfiguration". *Proceedings of the IEEE* vol. 106 (1), pp. 113–128.

Pronzato, L., and A. Pázman. 2013. *Design of Experiments in Nonlinear Models*, Volume 212. New York, NY, Springer New York.

Rasheed, A., O. San, and T. Kvamsdal. 2020. "Digital Twin: Values, Challenges and Enablers From a Modeling Perspective". *IEEE Access* vol. 8, pp. 21980–22012.

Rhodes, I. 1971. "A Tutorial Introduction to Estimation and Filtering". *IEEE Transactions on Automatic Control* vol. 16 (6), pp. 688–706.

Rizzi, S. 2009. "What-If Analysis". In *Encyclopedia of Database Systems*, pp. 3525–3529. Springer US.

Sari, A., and J. C. Sopuru. 2021. "Bayesian Model for Evaluating Real-World Adaptation Progress of a Cyber-Physical System". In *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*, pp. 324–343. IGI Global.

Simon, D. 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons.

Tao, F., Q. Qi, L. Wang, and A. Y. C. Nee. 2019. "Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison". *Engineering* vol. 5 (4), pp. 653–661.

Tao, F., H. Zhang, A. Liu, and A. Y. C. Nee. 2019. "Digital Twin in Industry: State-of-the-Art". *IEEE Transactions on Industrial Informatics* vol. 15 (4), pp. 2405–2415.

Telford, J. K. 2007. "A Brief Introduction to Design of Experiments". *Johns Hopkins apl technical digest* vol. 27 (3), pp. 224–232.

Ward, M. O., G. Grinstein, and D. Keim. 2010. *Interactive Data Visualization: Foundations, Techniques, and Applications*. CRC Press.

Weyns, D. 2019. "Software Engineering of Self-Adaptive Systems". In *Handbook of Software Engineering*, edited by S. Cha, R. N. Taylor, and K. Kang, pp. 399–443. Cham, Springer International Publishing.

Williams, J. J., and L. S. Esteves. 2017. "Guidance on Setup, Calibration, and Validation of Hydrodynamic, Wave, and Sediment Models for Shelf Seas and Estuaries". *Advances in civil engineering* vol. 2017.

Yilin Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, Heejo Lee, A. Perrig, and B. Sinopoli. 2012. "Cyber–Physical Security of a Smart Grid Infrastructure". *Proceedings of the IEEE* vol. 100 (1), pp. 195–209.

Zacchia Lun, Y., A. D'Innocenzo, F. Smarra, I. Malavolta, and M. D. Di Benedetto. 2019. "State of the Art of Cyber-Physical Systems Security: An Automatic Control Perspective". *Journal of Systems and Software* vol. 149, pp. 174–216.

Zhou, P., D. Zuo, K. M. Hou, Z. Zhang, J. Dong, J. Li, and H. Zhou. 2019. "A Comprehensive Technological Survey on the Dependable Self-Management CPS: From Self-Adaptive Architecture to Self-Management Strategies". *Sensors* vol. 19 (5), pp. 1033.

AUTHOR BIOGRAPHIES

HAO FENG is a PhD Student at Aarhus University specializing in the fundamental techniques that make up Digital Twins, including monitoring. His PhD combines software and robotics. His email address is haof@ece.au.dk.

CLÁUDIO GOMES is a postdoc researcher at Aarhus University. He received his PhD at the University of Antwerp, for his work on the foundations of co-simulation. His email address is claudio.gomes@ece.au.dk.

CASPER THULE is a PostDoc at Aarhus University and member of the AU DIGIT Centre of Digitalization, Big Data and Data Analytics. His area of research is hybrid co-simulation and cyber-physical systems. His email address is casper.thule@ece.au.dk.

KENNETH LAUSDAHL is a researcher at Aarhus University. He holds a PhD from AU with focus on tool automation for formal methods and co-simulation. He has been involved in research projects in the CPS and co-simulation domain during the last decade. His email address is lausdahl@ece.au.dk.

ALEXANDROS IOSIFIDIS is an Associate Professor at Aarhus University. He serves as Associate Editor in Chief for Neurocomputing journal and as Associate/Area Editor for Signal Processing: Image Communication and Frontiers on Signal Processing journals. He has (co-)authored more than 180 papers on Machine Learning, Computer Vision, and Signal Processing. His email address is ai@ece.au.dk.

PETER GORM LARSEN is a professor at Aarhus University. He has been involved with >40 research projects and he currently leads the AU DIGIT Centre, the AU Centre for Digital Twins, as well as the research group for CPSs. He has published more than 200 articles and three books on modelling of CPSs and DTs. His email address is pgl@ece.au.dk.