Under What Conditions Does a Digital Shadow Track a Periodic Linear Physical System?

Hao Feng[2](✉), Cláudio Gomes[2], Michael Sandberg[1], Hugo Daniel Macedo[2], and Peter Gorm Larsen[2]

[1] Department of Mechanical and Production Engineering, Aarhus University, Inge Lehmanns Gade 10, Aarhus C, Denmark
ms@mpe.au.dk

[2] Department of Electrical and Computer Engineering, Aarhus University, Helsingforsgade 10, Aarhus N, Denmark
{haof,claudio.gomes,hdm,pgl}@ece.au.dk

Abstract. The synchronization between a Digital Shadow (DS) and a Cyber-Physical System (CPS) is paramount to enable anomaly detection, predictive maintenance, what-if analysis, etc. Such synchronization means that a simulation reflects, as closely as possible, the states of the CPS. The simulation however, requires the complete initial state of the system to be known, which is often infeasible in real applications. In our work, we study the conditions under which knowing the initial state of the system is irrelevant for a simulation to eventually synchronize with the CPS. We apply traditional stability analysis to answer this question for linear periodic systems. We demonstrate the method using a simple but representative system, an incubator with a periodic control signal.

Keywords: Digital Shadow · Cyber-Physical System · Stability · Tracking

1 Introduction

A DS is a type of monitoring system. It contains not only the digital version of the system but also amounts of services that can improve the value of the physical system. One of the services could be real-time simulation that accurately mimics a CPS and its environment, and it is paramount to enable many of the benefits that the Digital Twin vision offers, such as the improvement of productivity [1], automation [3], system intelligence [13], and more. A DS is also key in, for example, anomaly detection that can prevent failures by triggering alarms, so new process conditions can be employed in time to increase asset lifetime. Figure 1 shows two possible architectures for enabling anomaly detection.

One architecture that is an easy way to realize the anomaly detection mainly consist of the *real-time simulation* of a plant model (P'_{PT} in Fig. 1), feeding all

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. Hao Feng also acknowledges support from the China Scholarship Council.

A. Cerone et al. (Eds.): SEFM 2021 Workshops, LNCS 13230, pp. 143–155, 2022.
https://doi.org/10.1007/978-3-031-12429-7_11

inputs from the real plant to a simulation model, and observing whether the resulting behavior matches the measured. This concept is illustrated in Fig. 1 (when omitting the Kalman filter). Figure 1 also shows a second architecture, in which the *Kalman filter* is taken into account. Here, the latter architecture requires partial state measurement from the plant.

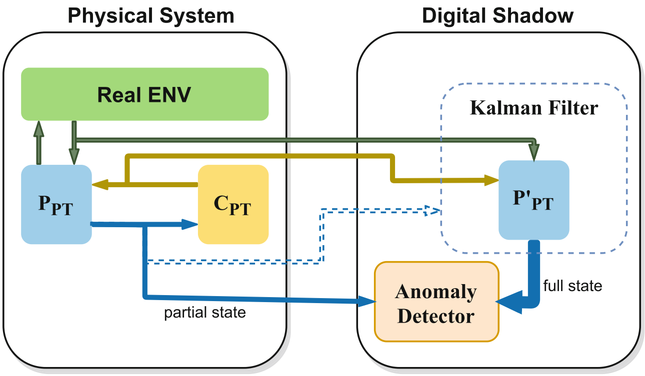

Fig. 1. Illustration of the two architectures for anomaly detection. One architecture is represented by the solid lines. Another one is the previous one combined with the Kalman filter.

When applicable, the Kalman filter architecture is superior to the real-time simulation architecture because the Kalman filter utilizes the state measurement data and the prediction data from a model, which makes it suffer less from *state-drift problems*. In contrast, the real-time simulation architecture does not take into account the state measurement data. State drifting is most commonly known in localization problems, where, for example, a moving object's acceleration and velocity are measured through an accelerometer and integrated to obtain the position. Here, assuming that the initial position and velocity are known, the process of integrating (essentially simulating a point mass system) yields the position over time, which almost always diverges from the real position of the moving object. Figure 2 shows one reference trajectory (in green), one tracking simulation trajectory that was started from different initial conditions (in blue), and two trajectories that are diverging.

The application of the Kalman filter is, however, not always possible as it requires a model to obey a certain structure. For instance, it requires the ability of updating the state of the model, which makes black-box models difficult to extend with Kalman filters. For example, the authors in [4] had to propose an extension to the Functional Mockup Interface (FMI) standard version 2.0 [2], to be able to apply Kalman filters.

In this paper, we focus on the conditions that make it possible for a real-time simulation architecture (without a Kalman filter) to avoid state drifting. Since real-time simulation models can utilize co-simulation interfaces [10], this

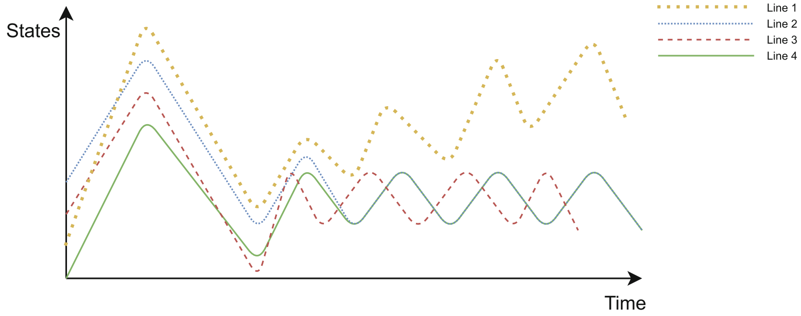

Fig. 2. A reference trajectory (line 4), and one converging tracking simulation that is started with different initial conditions (line 2). The line 3 and line 1 curves depict tracking simulations that suffer from state drifting. (Color figure online)

approach can be applied to virtually any black-box model. In the approach, we employ traditional stability analysis to demonstrate that a periodic linear system can be accurately tracked by a real-time simulation, even if the initial state of the plant is unknown. We exemplify the application of the method by using an open-loop incubator system, based on our previous work in [6,7]. The analyses in this work are limited to systems that are periodic and linear, and our incubator system fulfills these conditions.

We have empirical evidence that the closed-loop incubator system can also be tracked by real-time simulations, and we expect that more advanced stability analysis can be applied to CPSs that are non-linear or hybrid. The stability of these systems has been studied extensively (see, for example, [9,11,12,14,15]), and in future work, we seek to apply these methods to the closed-loop periodic systems as well.

The rest of the paper is organized as follows: Sect. 2 gives the background for the work, and details how to discretize continuous-time linear systems (an important step in our proposed method). Then, in Sect. 3, we introduce our contribution and apply it to the analysis of the incubator system. Next, Sect. 4 corroborate the results of the theoretical analysis with experimental results, and finally, Sect. 5 concludes the paper and discusses future work.

2 Background

This section gives background of the physical incubator system and its continuous and discrete-time models.

2.1 Incubator System

The incubator is a system that has the ability to regulate the temperature within an insulated container. A systematic diagram of the incubator is illustrated in Fig. 3 and Fig. 5 (right) shows a picture of the physical system.

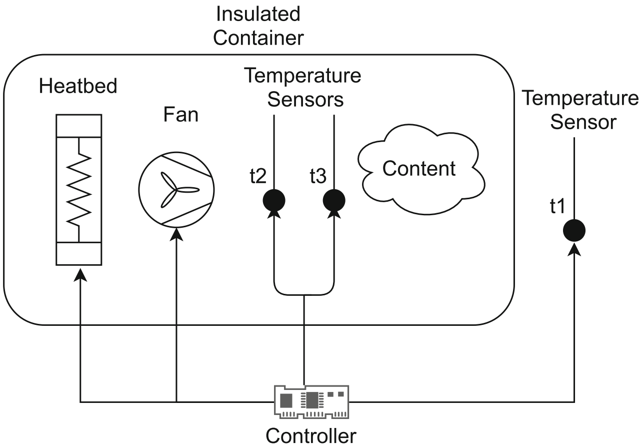

Fig. 3. Schematic overview of the incubator. The physical system can be seen in Fig. 5 (right).

The main components of our incubator system are an insulated container, a heatbed, a fan, three temperature sensors, and a controller. The heatbed is used to generate heat energy for heating the air inside the insulated container (referred to as "air temperature"), and it has two different states, on and off. The fan circulates the air, to achieve a more uniform temperature distribution. There are three sensors, two of which are used to measure the air temperature and one for the ambient room temperature. In the experiments, the fan is always on, but the controller can regulate both the heatbed and fan states. For detailed information about the incubator, see [7].

In this work, we configure the controller to turn on and off the heatbed in periodical intervals. This approach is in contrast with the closed-loop controller introduced in [7], which activates the heatbed based on the measured temperature. To keep the temperature relatively low in the experiments (defined here as $<50\,^{\circ}$C), we set the on-state to be shorter than the off-state (6 vs. 27 s).

2.2 Models

The state variables that enter our time-dependent thermal model of the incubator system are the air and heatbed temperatures, T_{bair} and T_{heater}. Here, we consider the rate of energy to establish an Ordinary Differential Equation (ODE). In reality, the temperature of the system is not completely uniformly distributed, meaning that T_{bair} and T_{heater} are lumped state variables. While T_{heater} is not directly measured from the system, we consider the average of the signal from the two internal temperature sensors to be representative of T_{bair}. The ODEs read

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

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

For more details, see [5]. In Eq. (1), C denotes the lumped thermal capacity, G is the effective heat transfer coefficient, T_{room} is the ambient room temperature, and VI represents the product of the voltage and the current representing the power of the heatbed when turned on.

To use Eq. (1) in stepwise simulations, it is necessary to convert it to a discrete-time model. In this paper, we give a brief explanation of the steps needed to accomplish this, but readers who are interested in more details can refer to [8].

In general, linear and continuous-time dynamical systems can be converted into a state-space model that has the generic form:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2a}$$

$$y(t) = Cx(t) + Du(t). \tag{2b}$$

The solution of Eq. (2a) is given as

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{3}$$

Using Eq. (3), the temperature state at any time instance can be obtained. In Eq. (3), we can see that the initial point of $x(t_0)$ is necessary when approximating the model behavior. However, in a computational unit, it is not possible to generate continuous behaviors, as it has to be sampled which is a discrete-time behavior. If an interval of the sampling process is T, then the time series is $0, T, 2T, \ldots, kT, (k+1)T, \ldots$, where k is an integer. In addition, the inputs between two sampling instants are constants, which means that $u(\tau) = u(kT)$ for $kT \leq \tau \leq (k+1)T$. In Eq. (3), let $t_0 = kT$. Then the next sample we can obtain is $x((k+1)T)$, thus we set $t = (k+1)T$. Substitute these back into Eq. (3) and we have

$$x((k+1)T) = e^{A(T)}x(kT) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)}Bd\tau u(kT). \tag{4}$$

Since the T is known, we can simplify Eq. (4) into the form of

$$x(k+1) = A'x(k) + B'u(k), \tag{5}$$

where $A' = e^{AT}$, $B' = \int_{kT}^{(k+1)T} e^{A((k+1)T-\tau)}Bd\tau$. Note that Eq. (5) is similar to Eq. (2a) but the coefficients are different and the time interval between $x(k+1)$ and $x(k)$ is T.

To summarize, Eq. (5) is the discrete-time model of the continuous-time model of Eq. (2) and it is feasible to implement Eq. (5) to generate behaviors in a computational unit. In the section below, we show how to apply this technique to prove that simulations of the incubator system will converge towards the real system behavior.

3 Stability Analysis of Periodic Linear Systems

This section gives an introduction about the stability of a dynamic system and how to utilize it to tackle the issues of tracking a periodic physical system under unknown initial conditions.

Stability is a property of a dynamic system. A pendulum, for example, is a stable system. Regardless of what the initial state of a pendulum is, unless perfectly upright, the pendulum will oscillate around the equilibrium point that is the state of vertical down if it is a perfect pendulum and has no friction. However, if there is friction, the pendulum will eventually go into the state of vertical down as its initial gravitational potential energy is consumed by friction and other energy losses. This is called asymptotic stability that is the property that a system will eventually converge to a stable state when it is started close to one.

Accordingly, a simulation of the incubator system can be started with different initial conditions that will all eventually converge to its single stable state. This also applies for the true initial conditions, and this property alleviates the issue of having unknown initial conditions.

While it does not make sense to talk about a stable periodic incubator system, because the temperature will continuously oscillate, we can transform the system into an equivalent system that represents the temperature evolution at certain periods of time. This new system is then asymptotically stable, which is illustrated in Fig. 4, where the solution to the original system is not asymptotically stable, but the equivalent sampled discrete-time system is.

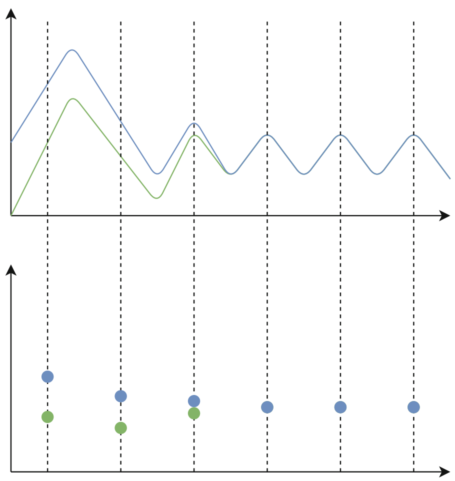

Fig. 4. This figure illustrates how the probed temperature is not asymptotically stable, but the equivalent sampled discrete-time system is.

The incubator continuous-time model is presented in Eq. (1). For simplicity, we use the generic discrete-time model of Eq. (5) to represent the incubator. Note that to convert the continuous-time model to a discrete-time model, the sampling rate that is the T in Eq. (4) is needed and we call it $T_{sampling}$. Note that the $T_{sampling}$ is normally chosen according to the system based on the hardware. In addition, we set A' and B' to A and B respectively for consistency. Then we have a linear system in the form of

$$x(k+1) = Ax(k) + Bu(k). \tag{6}$$

The controller only controls the states of the heatbed that are on (heating for short) or off (cooling for short), and we assume the room temperature is constant in between controller samples. As a result, the $u(k)$ has to be either $\left[VI\ T_{room}\right]^T$ or $\left[0\ T_{room}\right]^T$. Substitute the two $u(k)$ back to Eq. (6), then we obtain two different models that indicate the advancements of temperatures in terms of heating or cooling. These two models are given by

$$x(k+1) = Ax(k) + M \tag{7a}$$

$$x(k+1) = Ax(k) + N, \tag{7b}$$

where Eq. (7a) is the model of heating and Eq. (7b) of cooling. In each sampling step, $T_{sampling}$, either of the models in Eq. (7) is executed and therefore, the time for heating ($T_{heating}$) or cooling ($T_{cooling}$) should be equal to $n \cdot T_{sampling}$, where n is a non-negative integer.

To highlight the main idea of the method, suppose that the controller always switches from heating to cooling at every time sample, which means that $T_{heating} = T_{sampling}$ and $T_{cooling} = T_{sampling}$. Then, if we start with the heating state at time step k, the temperatures at next time step $k+1$ is

$$x(k+1) = Ax(k) + M. \tag{8}$$

Afterward, the incubator goes to the cooling state from time step $k+1$, thus the temperatures at time step $k+2$ are obtained by Eq. (7b)

$$\begin{aligned} x(k+2) &= A(Ax(k) + M) + N \\ x(k+2) &= A^2x(k) + AM + N. \end{aligned} \tag{9}$$

Equation (9) represents the temperatures during one period of heating and cooling. If the initial k is 0, we get the temperatures at time step 2 according to Eq. (9). If the periodic control signal continues, we have the equation representing the temperatures at time step $2k$

$$x(2k) = A^{2k}x(0) + (AM + N)(A^{2k-2} + A^{2k-4} + \ldots + A^0). \tag{10}$$

Note that Eq. (10) is only valid for $T_{heating} = T_{sampling}$ and $T_{cooling} = T_{sampling}$. If $T_{heating} = m * T_{sampling}$ and $T_{cooling} = n * T_{sampling}$, where m and n are integers, then in order to obtain Eq. (10), it involves m times and n times compositions of Eq. (7a) and Eq. (7b) respectively.

If the eigenvalues of the matrix A that represents the properties of the system are less than 1, which means the system is stable, then $\lim_{k\to+\infty} A^{2k} = 0$. As a result, the temperatures at time step $2k$, when k is large enough, is $x(2k) = (AM + N)(A^{2k-2} + A^{2k-4} + \ldots + A^0)$, which is independent of the initial conditions, $x(0)$. If the absolute value of all eigenvalues of the matrix A are less than 1, then $(AM + N)(A^{2k-2} + A^{2k-4} + \ldots + A^0)$ converges. This means under such a periodic control signal, the temperatures will eventually convergence toward $\lim_{k\to+\infty}(AM + N)(A^{2k-2} + A^{2k-4} + \ldots + A^0)$ no matter the initial conditions. This proves that a DS can track a periodic system even when the initial conditions are unknown.

If the absolute value of one of the eigenvalues of A is bigger than 1, a system is not stable and there are no stable states. In this case, a DS would not be able to track the periodic system without knowing the initial conditions.

Overall, if a system is stable when subject to a given control signal and if it only has one possible stable state, then a DS of the system can track the physical system without knowing exactly the initial conditions. Our incubator system is a stable system when subject to the periodic control signal and the eigenvalues of A are less than 1, which will be shown in the next section. Therefore, a DS of the incubator can track the physical incubator without knowing exactly the initial conditions.

4 Simulation and Experiments Results of Incubator

In Sect. 3, we analyzed the theoretical stability of the incubator. This section presents the simulation and experimental results of our analysis. Note that due to limitations of utilized hardware, it takes around one second to fetch data from one sensor, and three seconds for all three sensors.

As a first step before the stability analysis, we conducted a small experiment to investigate the internal temperature distribution of the air inside the insulated container. These results are shown in Fig. 5.

We used three temperature sensors to measure the temperature at different locations, marked $t1$, $t2$, and $t3$ in Fig. 5. As it can be seen, the temperature is not uniform when the incubator warms up. Here, the largest temperature difference was about 6 °C. According to this experiment, we found that the temperature at sensor $t1$ is close to the average temperature of sensors $t2$ and $t3$ (Fig. 5). This finding supports our decision of moving $t1$ outside for measuring the ambient room temperature and using the average temperature of sensors $t2$ and $t3$ as representative of the internal air temperature. We used this new setup in the following experiments.

Next, we conducted another experiment to determine the best-fit parameters in Eq. (1), and as Fig. 6 shows, the calibrated model captured the physical system very well. Nevertheless, for control conditions that deviate from the ones in the experiments, recalibration might be needed. Also, it is important to note that

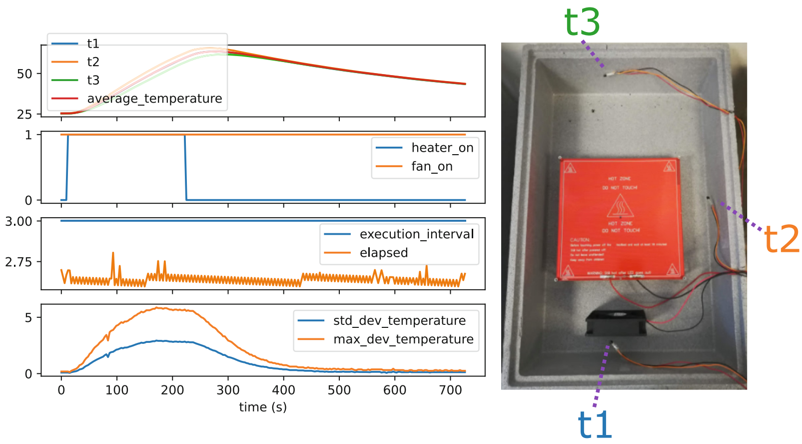

Fig. 5. Experimentally measured temperature, heater signal, together with maximum and std. deviation of the measured temperature (left). The experimental setup can be seen in the figure to the right.

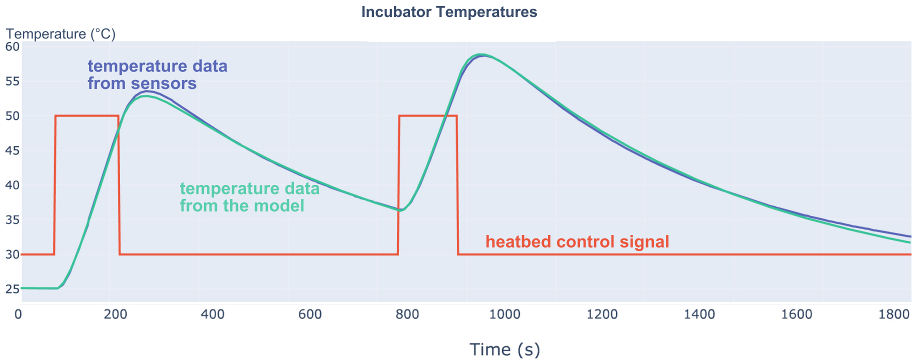

Fig. 6. Calibration results of (1). The blue line is the average temperature representing the air temperature, while the green line shows the results of our calibrated model. The red line shows the two states of the heatbed, heating and cooling. (Color figure online)

the model is incapable of capturing other complex process conditions such as opening the incubator lid as shown in Fig. 7 despite calibration.

After the calibrated model was demonstrated to be representative of the physical system, we utilized it to demonstrate that the incubator system is, in fact, stable and that it converges towards the same temperature state no matter the initial conditions following our approach described in Sects. 2 and 3. To do this, we ran five simulations with different initial conditions of the air temperature that are 40 °C, 30 °C, 25 °C, 20 °C, and 10 °C. The room temperature in the experiments was set to 22.44 °C and the voltage and the current to the heatbed were 12 V and 10.45 A. The simulation results are shown in Fig. 8.

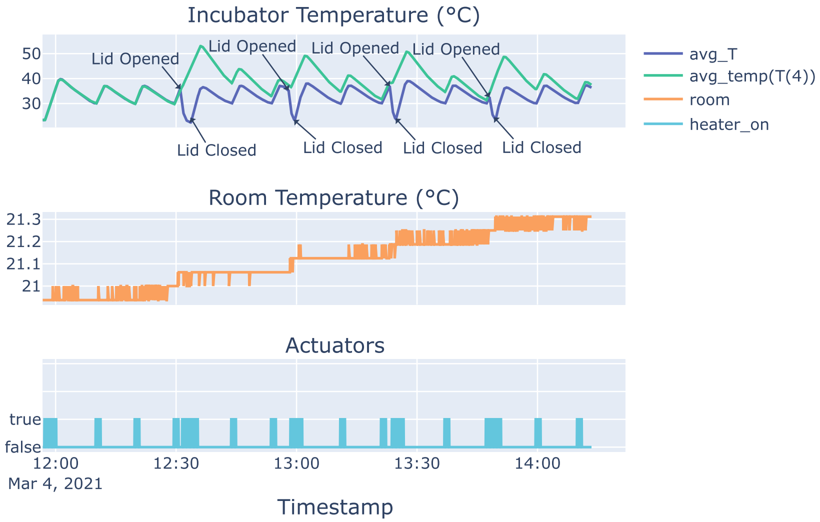

Fig. 7. Example of how the model initially captures the physical system very well, but starts to deviate when the process conditions change as the lid is opened. This cannot be resolved by recalibration, as the underlying model does not reflect the disturbance that is introduced by opening the lid.

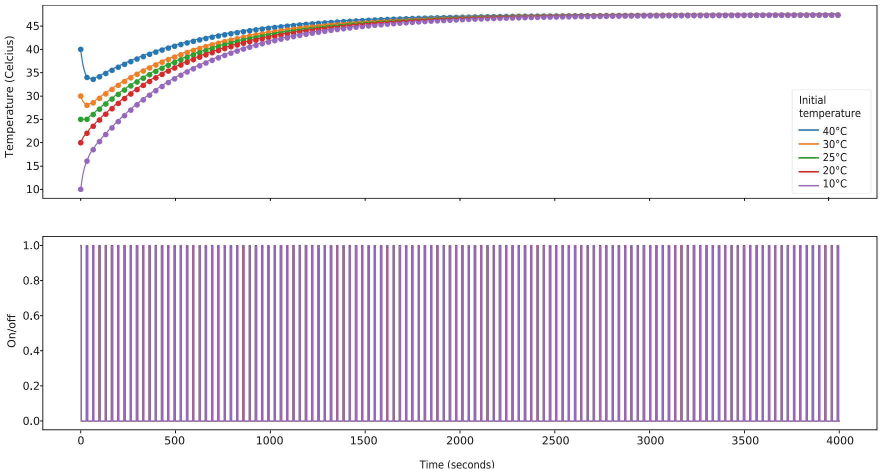

Fig. 8. Incubator simulation results. The top figure presents the simulation results of the air temperature starting with five different initial conditions, while the lower figure shows the controller signal for the heater bed (6 s on, followed by 27 s off).

As it can be seen in Fig. 8, all five initial conditions converged to the same stable state, meaning that the discrepancy between DS and the real physical system will be identical no matter the initial conditions after enough time. We also

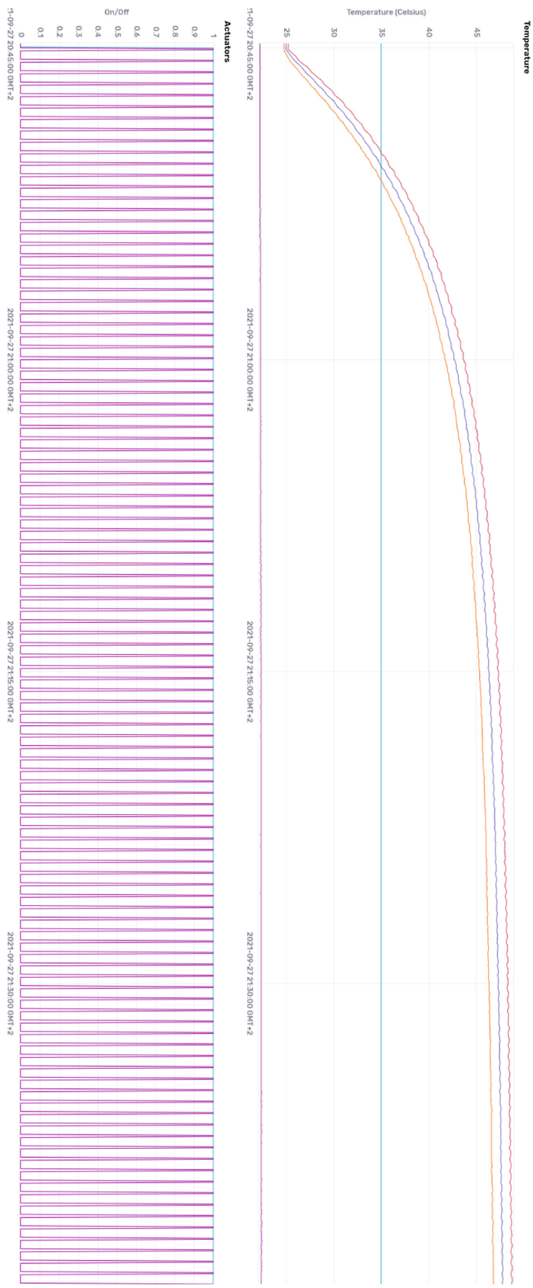

Fig. 9. Experimental results recorded and visualized using InfluxDB. The top figure shows the measured temperatures $t2$ (red) and $t3$ (orange), as well as the average temperature (purple). The lower figure shows the control signal. (Color figure online)

confirmed this by running experiments on the physical system. Here, we used InfluxDB to record and visualize the air temperature data. While it was not possible to control the initial starting temperature of the physical system, the experiment showed (Fig. 9) that the temperature increased at the early stage and gradually converged towards the same 47–48 °C when the system was started at room temperature. Since this result matches the simulation, this verifies the reliability of our model and approach. Note that, as consequence, if the lid is opened and closed again during one experiment, the temperature will also converge to the stable states because it is analog to start a simulation with a different initial condition. Finally, we note that the eigenvalues of the A matrix in our model are 0.942 and 0.448, both less than 1. This concludes that our DS can track the real physical system and that the system is stable.

5 Conclusion and Future Work

In order to tackle the problem of tracking a physical system with unknown initial conditions, we transformed it into a problem of proving the stability of the system. For a system with an open-loop controller that can enter into a stable state, we showed that a DS starting with a different initial conditions will also reach a similar stable state and track the physical system. Subsequently, we demonstrated this theory using our incubator system fitted with a periodic open-loop controller that resulted in a stable system. By conducting simulations and experiments, we verified our theory and concluded that a DS can track the periodic physical system if the system is stable and only has one stable state.

In the future, we are interested in analyzing the system with a closed-loop controller since this is more robust and general compared to an open-loop system. Also, we want further develop the DS of our incubator system, making it capable of coping with disturbances from opening the lid, etc.

References

1. Bauernhansl, T., Hartleif, S., Felix, T.: The Digital Shadow of production – a concept for the effective and efficient information supply in dynamic industrial environments. Procedia CIRP **72**, 69–74 (2018). https://doi.org/10.1016/j.procir.2018.03.188
2. Blochwitz, T.: Functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: 9th International Modelica Conference, pp. 173–184. Linköping University Electronic Press, Munich, November 2012. https://doi.org/10.3384/ecp12076173
3. Brecher, C., Buchsbaum, M., Storms, S.: Control from the cloud: edge computing, services and digital shadow for automation technologies. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 9327–9333, May 2019. https://doi.org/10.1109/ICRA.2019.8793488

4. Brembeck, J., Pfeiffer, A., Fleps-Dezasse, M., Otter, M., Wernersson, K., Elmqvist, H.: Nonlinear state estimation with an extended FMI 2.0 co-simulation interface. In: 10th International Modelica Conference, pp. 53–62. Linköping University Electronic Press; Linköpings Universitet, Lund, March 2014. https://doi.org/10.3384/ecp1409653
5. Feng, H., Gomes, C., Sandberg, M., Thule, C., Lausdahl, K., Larsen, P.G.: Developing a physical and digital twin: a process model. In: 3rd International Workshop on Multi-paradigm Modeling for Cyber-Physical Systems, Fukuoka, Japan (2021, to appear)
6. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Iosifidis, A., Larsen, P.G.: Introduction to digital twin engineering. In: 2021 Annual Modeling and Simulation Conference (ANNSIM), pp. 1–12, July 2021. https://doi.org/10.23919/ANNSIM52504.2021.9552135
7. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Sandberg, M., Larsen, P.G.: The incubator case study for digital twin engineering, February 2021. arXiv:2102.10390 [cs, eess]
8. Friedland, B.: Control System Design: An Introduction to State-Space Methods. Courier Corporation (2012)
9. Goebel, R., Sanfelice, R.G., Teel, A.R.: Hybrid Dynamical Systems: Modeling, Stability, and Robustness. Princeton University Press (2012). http://www.u.arizona.edu/~sricardo/index.php?n=Main.Books
10. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: a survey. ACM Comput. Surv. **51**(3), 49:1-49:33 (2018). https://doi.org/10.1145/3179993
11. Johansson, M.: Piecewise Linear Control Systems, vol. 284. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36801-9
12. Jungers, R.: The Joint Spectral Radius: Theory and Applications, vol. 385. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-95980-9
13. Ladj, A., Wang, Z., Meski, O., Belkadi, F., Ritou, M., Da Cunha, C.: A knowledge-based Digital Shadow for machining industry in a Digital Twin perspective. J. Manuf. Syst. **58**, 168–179 (2021). https://doi.org/10.1016/j.jmsy.2020.07.018
14. Legat, B., Gomes, C., Karalis, P., Jungers, R.M., Navarro-López, E.M., Vangheluwe, H.: Stability of planar switched systems under delayed event detection. In: 2020 59th IEEE Conference on Decision and Control (CDC), pp. 5792–5797, December 2020. https://doi.org/10.1109/CDC42340.2020.9304152
15. Liberzon, D.: Switching in Systems and Control. Springer, Boston (2012). https://doi.org/10.1007/978-1-4612-0017-8