Self-improving Models for the Intelligent Digital Twin: Towards Closing the Reality-to-Simulation Gap

Manuel S. Müller*, Nasser Jazdi*, Michael Weyrich*

**Institute of Industrial Automation and Software Engineering, University of Stuttgart, Stuttgart, DE 70550 Germany (Tel: +49-711-685-67306; e-mail: manuel.mueller@ias.uni-stuttgart.de).*

Abstract: This paper presents a novel approach to ensure the quality of the Digital Twin models that modern Cyber-Physical Manufacturing Systems (CPMS) rely on. CPMS are configurable and intelligent. Environmental and system parameters change frequently. Thus, static models are inadequate. Autonomous mobile robots and the simulation of their movement are important elements of these CPMS. Based on our reinforcement learning-based methodology, we use these robots as an example to show how the Digital Twin automatically improves models that do not perfectly represent the physical asset, making it an intelligent Digital Twin. In our scenario, the behavior of the asset deviates from the simulated prediction, i.e., a simulation gap occurs. The presented approach closes this simulation gap through a three-step mechanism. First, it makes the simulated data and the real data comparable and synchronizes it. Second, it applies reinforcement learning to find patterns in the deviations between the simulated and real data. Third, it learns to compensate for them. The evaluation of this example shows promising results.

Keywords: Reinforcement learning, Digital Twin, synchronization, reality-to-simulation gap, intelligent modelling, Cyber-Physical Manufacturing Systems, autonomous mobile robot, self-improving models

1. INTRODUCTION

Intelligent cyber-phyiscal manufacturing systems (CPMS) become flexible, (re-)configurable and intelligent. They support smaller batch sizes, shorter reconfiguration times and perform production and reconfiguration with less human intervention up to full autonomy. To achieve this flexibility, CPMS are often set up in a matrix production environment where autonomous mobile robots connect the individual workstations. One enabling technology on this way is the Digital Twin. The Digital Twin of a CPMS "is a virtual representation of a physical asset […], capable of mirroring its static and dynamic characteristics" (Ashtari Talkhestani et al., 2019). Its main capabilities is mirroring the physical asset to the cyber world. The Digital Twin therefore needs the three capabilities: "synchronization with the real asset, active data acquisition from the real environment and the ability of simulation" (Ashtari Talkhestani et al., 2019). With these capabilities, it enables applications like virtual commissioning, what-if simulations of reconfiguration variants, simulative studies of corner cases etc. However, all these applications rely on the quality of the Digital Twin's models. As West and Blackburn showed in (West and Blackburn, 2018), it is impractical or at least uneconomic to model every detail of the physical asset in advance. Another reason for model inaccuracy is the fact that in the life cycle of an asset the gap between models and reality is constantly increasing. The cause of this gap is the changes within the physical asset or in its environment. CPMS therefore must deal with imperfect models. As the models do not perfectly represent the physical asset, the behaviour of the asset is expected to deviate from the simulated prognosis, i.e. a simulation gap arises. What would humans do in this situation? They would study the deviations from their expectation and improve on the models based on the experience they gain during operation. For this reason, it is an intuitive approach to apply this strategy to the Digital Twin, as well, rendering the Digital Twin an intelligent Digital Twin. In this paper, we present a reinforcement learning based approach that follows this strategy, thus closing the simulation gap. The approach is evaluated on an autonomous mobile robot connecting the workstations of a CPMS.

The reminder of this paper is structured as follows: After the introduction, we outline the background of this paper in Section 2. Afterwards, in Section 3, we present our approach of self-improving models for the Digital Twin. Thereafter, we illustrate our prototype (Section 4) and evaluate it towards the ability of closing the simulation gap. The paper ends with a conclusion (Section 5).

2. DIGITAL TWIN AND SIMULATION-REALITY GAP

Digital Twin started as a vision of the NASA, driven by the idea of fully simulatable aerospace missions (Glaessgen and Stargel). In the early days, the NASA painted the vision that the Digital Twin would be equipped with a set of models, covering every detail of the system's behavior. Thus, it predict any behavior of the system in any case (West and Blackburn, 2018). With this property, they would enable the perfect zero-shot-transfer to the physical world. However, this approach has several drawbacks. West and Blackburn consider this unrealistic or at least uneconomic. Our survey on Digital Twin for verification and validation (Löcklin et al., 2020) supports this thesis since we hardly found approaches with full-featured Digital Twins and none of them alone was enough to certify the asset.

However, there are huge progresses in this field. Just the take strategy changed. Simulation is one core attribute of the Digital Twin. With the synchronization, there is a strong emphasize on the transfer from simulation to reality and back. Previous work towards the synchronization of the Digital Twin focus on changes on the CPMS, how to detect these and keep the (simulation) models and their relations consistently to the physical asset (Ashtari Talkhestani et al., 2018; Talkhestani et al., 2018). In more recent work, the area of AI-enhanced Digital Twin (Jazdi et al., 2021) is researched. Specifically transfer learning is connected to the Digital Twin (Maschler et al., 2021). From this perspective, the research on bringing the simulation-to-reality gap, which has been studied for over 20 years now (Mouret and Chatzilygeroudis, 2017), comes into touch with the Digital Twin research. However, the latter survey outlines, that the simulators are not good enough now. Although approaches exist that try to tune the simulator in order to bridge the simulation-to-reality gap (Collins et al., 2020), according to Mouret and Chatzilygeroudis, this may not get so far in the next couple of years. Therefore, it moves the Zero-Shot-Transfer in the future. In contrast, (W. Zhao et al., 2020) identify in their survey system identification, domain randomization, domain adaption, and learning with disturbances as subsidiary approaches in order to bridge the simulation-to-reality gap. In this area, (Bousmalis et al., 2018) study the applicability of simulation-trained models to the reality using randomized environments and domain adaption methods. Moreover, (Zhao et al., 2020) and (Rao et al., 2020) consider the area from a deep reinforcement learning perspective with the intention to train the agents in simulation environment and apply them to real systems.

However, considering the Digital Twin, there is a difference to the pure simulation-to-reality transfer: Instead of starting with a simulation and transferring the result to the reality, in the case of the Digital Twin, simulation and physical execution coexist. Therefore, not only a transfer to reality, but also a transfer back is required. One approach considering this issue is (Chang and Padir, 2020). Since the environment and the system itself change and the initial models exist in different qualities, an intelligent Digital Twin has to care dynamically about its models. As a result, it must improve them as soon as it detects that its models are no longer suitable or even valid.

The departure from the vision of the Digital Twin being able to describe the asset in any detail has the consequence that discrepancies arise between the forecast of the Digital Twin and the process data of the physical asset. In other words, a simulation-reality gap is created. Approaches to deal with this simulation gap already exist from the simulation sciences and in particular from the field of reinforcement learning. Reinforcement learning uses the simulation to safely explore the environment selecting actions from the action space and learn from the environment's feedback how to improve the behavior. However, these mainly focus on the transfer from a simulation to a real application. The Digital Twin, in contrast, requires a mutual interaction between cyber and physical world. This paper provides a contribution to this. We propose an approach that builds on the spirit of adaptive models but with the focus on closing the reality-to-simulation gap in order to create self-improving models for the intelligent Digital Twin. These self-improving models therefore get closer to the physical asset's actual behavior. This approach is presented in the following section.

3. APPROACH FOR SELF-IMPROVING MODELS

The following aspects build the basic idea of our approach: Observation, perception, analysis, and reasoning. For the reasoning, the existing operational experience and expert knowledge are used. In contrast to the general approaches, where the algorithms are pre-training in the simulation first and afterwards irrelevant when applied to reality, the Digital Twin paradigm proposes using Digital Twin' s simulation part and asses at the same time with equal priority. For this reason, the information transfer takes place in two directions: from the Digital Twin to the asset providing predictions and what-if analyses and from the asset to the Digital Twin providing not only reference data to validate the predictions but also information about changes in the environment and the system itself. This bidirectional transfer keeps the Digital Twin consistent with the asset and at the same time enriches the asset with additional background. We consider this bidirectional transfer synchronization. The synchronization builds the basis for the (automatic) model improvement. Fig. 1 illustrates the relationship.

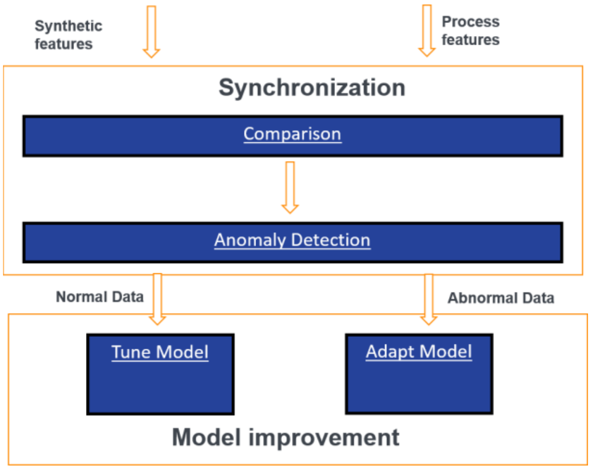


Fig. 1: Workflow of synchronization and model improvement

The synchronization uses both, the synthetic data and the process data, compares them and searches for aspects that are not well predicted or at least explainable from the simulation models. Therefore, the synchronization process consists of two parts, comparison and anomaly detection. From the comparison between the cyber data and the asset, plausibility checks validate the process data. In this way, obvious measurement errors are unveiled. In this case, the system relies on the simulation. However, especially in the beginning of the application phase, large simulation-to-reality gaps have to be taken into account. If inconsistencies occur that are not obvious measurement errors, the measurements always overrule the simulations. Therefore, the simulation adopts the system state based on the measurements, accepting the violation of physical constraints. In the example of the autonomous mobile robot, position jumps occur. The anomaly detection then differentiates normal deviations between simulation and reality from unexplainable behaviour. Clustering the process data in repeating steps and thereafter statistically evaluating the respective data inside these clusters, the anomaly detection provides evidence to the comparison part, where to put the border between obvious measurement error and unexpected but plausible data. The first way of supporting this decision is straightforward calculation of observed variations and therefore concluding on plausible deviations. However, using more sophisticated anomaly detection, this procedure may be complemented by identification of preliminary signs of leaving the validity range. The survey (Lindemann et al., 2021) sums up approaches for more sophisticated anomaly detection.

Independent on the used approach, anomaly detection provides information whether the incoming data is considered normal or abnormal. This information is crucial for model improvement since the way of changing the models differs fundamentally between these two cases. In the normal case, the basic correctness of the underlying models is assumed. Therefore, the model is tuned using methods of system identification or parameter tuning. In contrast, having abnormal data indicates fundamental problems in the model. Therefore, it is sensible to adapt the model as such by searching for a translation between current models and reality.

But how to compare the models of the Digital Twin with the physical asset? As (Mouret and Chatzilygeroudis, 2017) show, the simulators' synthetic data still differ significantly from real process data. In general, models simplify the reality and therefore by design need to be made comparable first. Our approach in this aspect is visualized in Fig. 2. We identified three steps namely data acquisition, pre-processing and transfer, which have to be executed in both domains, cyber and physical world differently in order to prepare for synchronization and model improvement. In the cyber domain, the simulation environment produces synthetic data. Normally, this data represent a subset of the total space of possibilities the system acts in. It is pure, i.e. does not contain noise or dirt effects. In order to make the data more general, the latter might be added. Moreover, to prepare the system for real-world data, the covered space has to be extended. This process is subsumed with the term Domain Randomization. Concrete approaches of how this works are proposed in (Tobin et al., 2017). The result of this Domain Randomization are synthetic features, which have to be unified in order to match the process features. In this context, it has to be noted, that the algorithms in the simulation domain might differ from the algorithms in the physical space. The Simulation-to-Reality Wrapper takes care on this task. It puts the detection layer on a higher level and therefore eases the comparison. One example is the domain of object detection. In this area, not only the identified label but also the confusion matrix to other labels should be considered. However, reducing this conversion table to the 5-10 most relevant misclassifications and comparing this between cyber world and reality is better comparable than comparing the features, which the object detectors use.

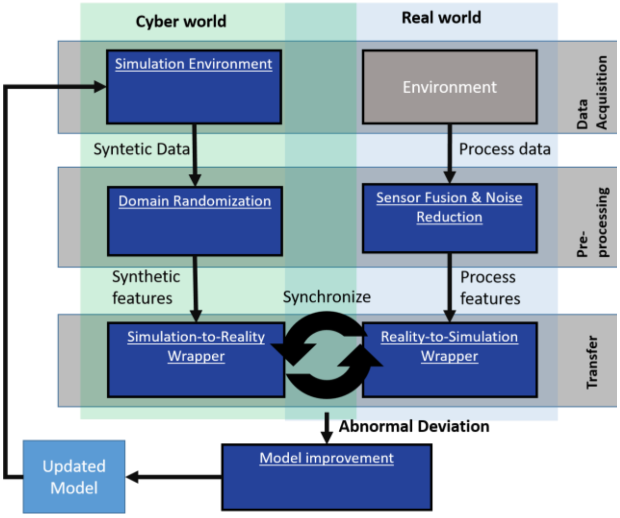


Fig. 2: Synchronization and model improvement

The physical world however acquires the process data directly from its sensory. The so acquired process data is pre-processed with exactly the opposite intention of the cyber world, namely to reduce dirt effects and noise and therefore purify the signal. Sensor fusion & noise reduction therefore are the matter of choice for pre-processing real-world data. As in the cyber space, the transfer follows. The Reality-to-Simulation Wrapper also abstracts from domain-specific features, making the information comparable to the synthetic features.

As previously announced, the model improvement follows two different approaches, depending on whether the deviation between forecast and measured values is classified as normal or abnormal. The differentiation between normal and abnormal is use case specific. In the easiest case, the signal-to-noise ratio serves as threshold. However, as the abnormal deviation can be seen as an anomaly, the Digital Twin can also exploit more sophisticated anomaly detection algorithms as described in (Lindemann et al., 2021).

While for the case of normal deviation the way to tune the model is already included in the model itself, the question of how to close the gap between simulation and reality when the model reaches its limits requires a model-independent approach. This paper focuses on this area. We propose exploiting reinforcement learning for this purpose. Table 1 summarizes the parameters of the learning algorithm. The basic idea is to use the synchronization signal as a delayed reward judging the actions of modifying the original model.

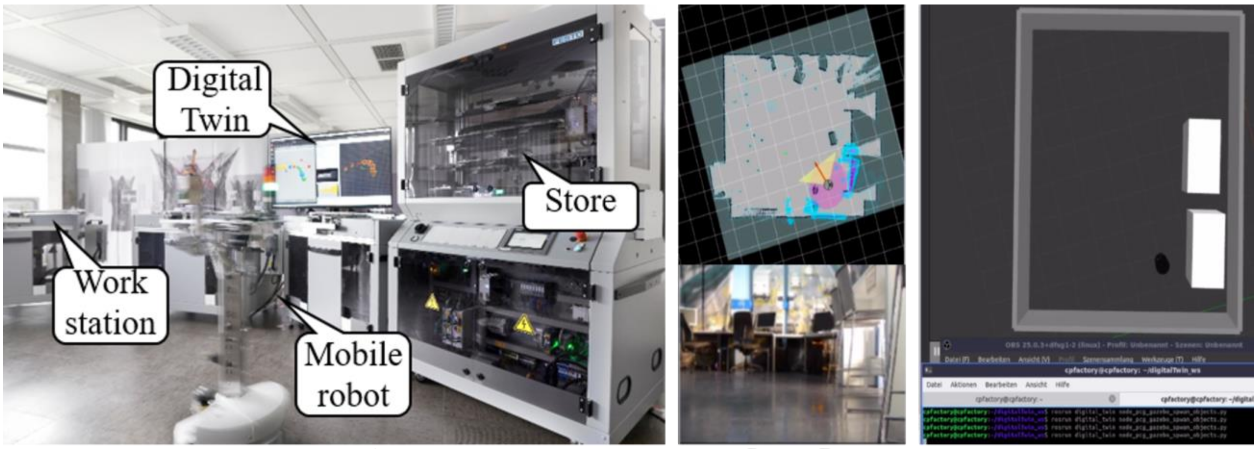
Table 1: parameters of the reinforcement learning agent

Parameters	Values
Algorithm class	State-Action.Reward-State-Action (SARSA)
Available input	Continuous value, Delayed reward, Multi-action
Assumptions	Only longitudinal control
Action space	0...20 cms^{-1}, quantization 1 cms^{-1}
Reward	$R(x) = \begin{cases} +1, & x < 1\,cm \\ -5, & x \geq 1\,cm \end{cases}$
End of an episode	x > 2 cm or rotational deviation > 0.01 rad

The reinforcement learning agent now translates the changes in the form of a discretized translation table from the original model to the process data, i.e., the measurements. This way, a new model is created. Each time the new model predicts the position of the robot with a maximum tolerance of 1 cm to the measurement, the agent receives a positive reward. This way the simulation models are adapted such that they better represent the physical asset and therefore reduce the simulation-reality gap. During the learning process, the agent's goal of minimizing the gap between cyber and real world makes the translation table converge to the actual representation of reality, i.e. a new, improved model is build. However, there are some aspects to care for in order to make this approach successful. The first aspect is the definition of the action space. It has to cover the complete observed space of the process data. Moreover, the resolution must be high enough that the quantization noise is significantly smaller than the expected model deviation. Since training data is limited, we suggest starting with very rough quantization. The strategy is just to get below the anomaly level and tune out the quantization errors in model tuning. In our experiment, we took an action space of only 20 actions leading to convergence with about 100 training samples. Of cause, larger action spaces are possible but lead to long convergence times or require for more training samples, respectively. This aspect has to be taken into account especially because synthetic data is identified to not correctly represent reality and therefore is not available. Against this background, it seems expedient to narrow down the range of deviation as much as possible. In our example, the odemetry of a mobile robot was reduced to the longitudinal control. Changes in the cross control therefore trigger a new training sequence. More details on the prototype and the experiment are descripted in the following section.

4. CASE STUDY

As previously indicated, we consider an autonomous mobile robot connecting various workstations in a matrix production. The autonomous mobile robot's Digital Twin is used to safely navigate in the manufacturing system and coordinate exchanging workpieces between the respective resources. Fig. 3 (left) illustrates the autonomous mobile robot within the CPMS. In the sketched scenario, the deviation between simulation and reality is small to enable the other factory components to use them e.g. to co-simulate docking processes. In our case, we only tolerate a simulation-to-reality gap of 2 cm. The Robotino 3 Premium by Festo is used as the autonomous mobile robot. It is equipped with a laser scanner used for simultaneous localization and mapping and a monocular camera assisting the object detection and visualizing the environment. On the Robotino, a Robot Operating System (ROS) node, is running. Through this ROS node, the Robotino is controlled wirelessly from a pc running Lubuntu 20.04. The simulation environment builds on Gazebo. However, machine learning models, especially reinforcement learning models complement it. The framework RviZ serves for visualization. Fig. 3 illustrates the perception of the intelligent Digital Twin. In this visualization, the autonomous mobile robot's process data with camera image and laser scanner data is shown on the left, where the simulation is visualized on the right. In this simulation, the two white tables on the right side of the autonomous mobile robot are already recognized and modelled as bounding boxes. The tool bar on the left hand side allows for selecting the signals to monitor. The map in the centre shows the current laser scanner measurement (cyan), the direction of movement (red arrow), and the assigned cost maps representing regions to stay out. The terminal on the bottom right shows logs of the running scripts. In this scenario, we consider the correct movement and positioning of the robot in the simulation. The position of the Digital Twin and the asset is synchronized. Laser scanner data is therefore processed to extract the position using Simultaneous Localization And Mapping (SLAM), which is then transferred to the simulation coordinate system. Moreover, features like the mentioned tables, represented as bounding boxes are transferred to simulation. In turn, the simulation model of the movement predicts the position under domain randomization using physics models. In the drive of the autonomous mobile robot, two nonlinearities exist, which are not yet modelled. Instead, the old model assumes a linear relationship between requested velocity and set velocity, i.e. the controller is always considered to be oscillated in. Therefore, in the Digital Twin, without countermeasures, the autonomous mobile robot "jumps" in the simulation



Asset

Process Data

Digital Twin

Fig. 3: Perception of the intelligent Digital Twin

environment due to frequent position updates. These position updates always occur when the tolerated deviation of 2 cm is violated. In conclusion, the number of position updates provides information about the quality of the model. It is compared before and after applying synchronization. However, the improved model is not applied directly but runs in parallel to the original model. In this way, the anomaly detection keeps triggering. Thus, the algorithms run in the model improvement branch supplying the reinforcement learning algorithm with samples. The algorithms switch to the new model once the reinforcement learning algorithm has converged.

5. EVALUATION

We evaluate the case study according two criteria. First, we check for plausibility interpreting the found translation table. Afterwards we quantify the success comparing the deviation between model's prediction and actual process data.

Our synchronization process takes the measurements from the laser scanner and processes it to position information using SLAM algorithm. This position measurement is validated through reference measurement. The synchronization module now compares this position to the simulated position. This simulated position is calculated using a simple physics model integrating over the set point value of the velocity. Since the positioning module has a much higher dudty cycle than the speed controller, the speed controller is seen as a subordinate control loop, which is assumed to be swung in. Then we run the reinforcement learning algorithm to come up with a mapping table that maps the original speed ("old action") to the better fitting velocity ("new action") to apply in simulation. The mapping from old action to new action is visualized in Fig. 4. By this experiment, it shows off that the naive assumption and therefore the resulting model is too simple. We can observe a start saturation up to 6 cms^{-1}. Obviously, the physical controller is not able to set movements below this level. Moreover, the plot indicates a speed saturation at 16 cms^{-1}. The physical controller seems to have a limit at this level instead of the assumed 20 cms^{-1}. Reasons for this behavior could be friction etc. consuming the control reserves. Both nonlinearities are plausible and were validated on the asset. Of cause, it was not a hard challenge to correct the models or build a better controller not showing these deficits manually. However, the amazing aspect of this approach is that the system automatically identified the problem in the model (i.e. identified the limits of the model) and improved it in order to compensate for the inadequacy. Using this model in the Intelligent Digital Twin for subsequent processes like trajectory prediction etc. running in an open loop simulation now provides more accurate results. Nevertheless, the compensated model is not yet ideal as well. In the linear area (7 cms^{-1}...15 cms^{-1}) the quantization error clearly shows off. Its severity depends on the trade-off between complexity and precision. The subsequent model tuning can be exploited here to decide in favor of complexity.

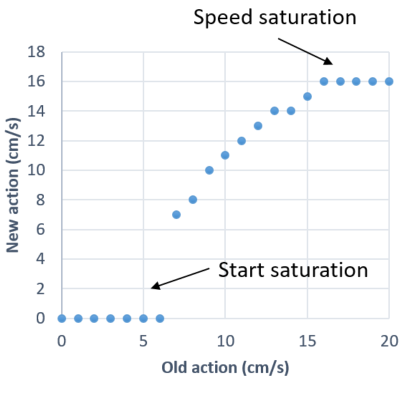


Fig. 4: Revealed pattern between old and new model

Applying this compensation shows a significant improvement of the model as visualized in Fig. 5 and Fig. 6. Fig. 5 shows the results without compensation. In average, every third position value has to be corrected (marked red), because it exceeds the limit of 2 cm deviation. Although the updates were this frequent, still deviations above 4 cm occur. The model quality is therefore rather poor. The new model applies the improvement using reinforcement learning. The results are visualized in Fig. 6. As can be seen, not only the number of deviations was reduced from 33 to 10, approximately a third compared to the old model, the magnitude of the deviation was also reduced. In other words, the model got three times closer to the actual position, an improvement by 300%.

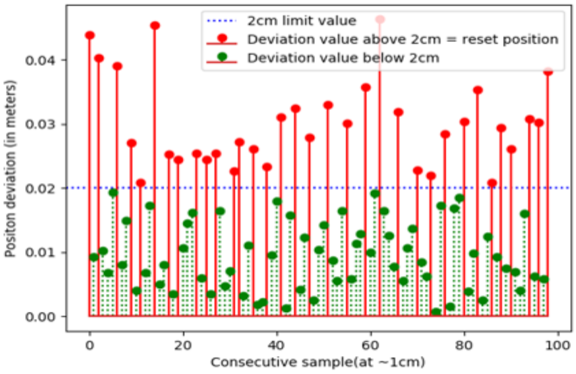


Fig. 5: Results without compensation

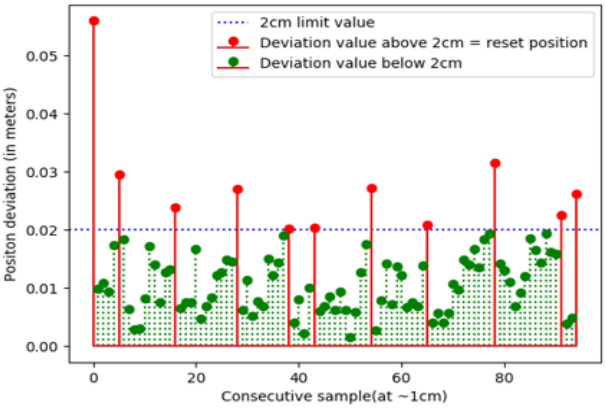


Fig. 6: Results with compensation

6. CONCLUSION

Modern Cyber-Physical Manufacturing Systems (CPMS) become flexible, (re-)configurable and intelligent. The intelligent Digital Twin supports this transformation process. In order to do so, a core challenge, a core challenge of the Digital Twin is to close the simulation-to-reality gap. The proposed approach thereby shows how self-improving models can be created. This synchronization reveals the limitations of the models, and the subsequent step of self-improvement causes the models to get closer to the actual behavior of the physical asset. A distinction must be made between normal deviations and anomalies, since the approach differs in the individual cases. While the improvement of models, which are in the normal range, is already inherent in the models, the model-free reinforcement learning method SARSA is suitable for improvement in the insufficiently modeled areas. In our case we reached:

- The synchronization mechanism automatically found out the nonlinearities in the system.
- The system learned autonomous how to compensate for the shortcomings of the movement model.
- The accuracy of the model increased by 300%, i.e. the position is closer to the actual position of the asset.

However, keeping the complexity low remains a critical aspect. Dealing with this aspect is interesting for future work. Moreover, it would be interesting to research more models applying the presented methodology. In addition, further approaches to adapt the models of the Digital Twin during runtime should be considered. Furthermore, the benefit to the transfer-learning model to reality using the synchronized models should be researched in future work.

REFERENCES

Ashtari Talkhestani B, Jazdi N, Schlögl W, Weyrich M. A concept in synchronization of virtual production system with real factory based on anchor-point method. Procedia CIRP 2018; 67: 13–7.

Ashtari Talkhestani B, Jung T, Lindemann B, Sahlab N, Jazdi N, Schloegl W, et al. An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System. at - Automatisierungstechnik 2019; 67(9): 762–82.

Bousmalis K, Irpan A, Wohlhart P, Bai Y, Kelcey M, Kalakrishnan M, et al. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2018. p. 4243–4250.

Chang P, Padir T. Sim2Real2Sim: Bridging the Gap Between Simulation and Real-World in Flexible Object Manipulation. In: 2020 Fourth IEEE International Conference on Robotic Computing (IRC). IEEE; 2020. p. 56–62.

Collins J, Brown R, Leitner J, Howard D. Traversing the Reality Gap via Simulator Tuning, 2020.

Glaessgen E, Stargel D. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures.

Jazdi N, Ashtari Talkhestani B, Maschler B, Weyrich M. Realization of AI-enhanced industrial automation systems using intelligent Digital Twins. Procedia CIRP 2021; 97: 396–400.

Lindemann B, Maschler B, Sahlab N, Weyrich M. A Survey on Anomaly Detection for Technical Systems using LSTM Networks. Computers in Industry 2021; 131(3): 103498.

Löcklin A, Müller M, Jung T, Jazdi N, White D, Weyrich M. Digital Twin for Verification and Validation of Industrial Automation Systems – a Survey. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). 1; 2020. p. 851–858.

Maschler B, Müller T, Löcklin A, Weyrich M. Transfer Learning as an Enhancement for Reconfiguration Management of Cyber-Physical Production Systems. Unpublished, 2021.

Mouret J-B, Chatzilygeroudis K. 20 years of reality gap: a few thoughts about simulators in evolutionary robotics. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2017. p. 1121–1124.

Rao K, Harris C, Irpan A, Levine S, Ibarz J, Khansari M. RL-CycleGAN: Reinforcement Learning Aware Simulation-to-Real. In: 2020 IEEE/CVF Conference on Computer; 2020. p. 11154–11163.

Talkhestani BA, Jazdi N, Schloegl W, Weyrich M. Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points. Procedia CIRP 2018; 72: 159–64.

Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2017.

W. Zhao, J. P. Queralta, T. Westerlund. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI); 2020. p. 737–744.

West TD, Blackburn M. DEMONSTRATED BENEFITS OF A NASCENT DIGITAL TWIN. INSIGHT 2018; 21(1): 43–7.

Zhao W, Queralta JP, Qingqing L, Westerlund T. Towards Closing the Sim-to-Real Gap in Collaborative Multi-Robot Deep Reinforcement Learning. In: 2020 5th International Conference on Robotics and Automation Engineering (ICRAE). IEEE; 2020. p. 7–12.