Digital twins for flexible linking of live sensor data with real-time models

Dr.-Ing. Reiner Jedermann[1], Kunal Singh[1], Prof. Walter Lang[1], Dr. Pramod Mahajan[2]
[1]University Bremen, Institute for Microsensors, -actuators and –systems (IMSAS), Germany
[2]Leibniz Institute for Agricultural Engineering and Bioeconomy (ATB), Potsdam, Germany

Author's version, please cite as: Jedermann, R.; Singh, K.; Lang, W.; Mahajan, P.: Digital twins for flexible linking of live sensor data with real-time models. In: Reindl, L.; Wöllenstein, J.,(eds.): Sensoren und Messsysteme, Beiträge der 21. ITG/GMA-Fachtagung, 10. – 11. Mai 2022, Nürnberg, Informationstechnische Gesellschaft im VDE (VDE ITG), 2022, Vol. 303, pp. 316-322.
Check https://www.ama-science.org/proceedings/listing/666 for final version and updates.

Abstract

For taking most advantage of real-time sensor data, data has to be processed by a single or even by a chain of models. Digital Twins (DTs) provide software platforms to perform the processing also in real-time in contrast to earlier simulation studies. The step from IoT to a full exploitation of DTs solutions entails new challenges, such as the transformation of models into an updateable format, but also new features, such as the easier and flexible linking of different models through a streaming platform. We discuss this key challenges and features based on our example application for remote monitoring of ocean fruit transportation.

1 Scope and motivation

Digital twins (DT) have gained much attention in the recent years, both in industry and research community. According to [1], a DT is a replica of a physical object, representing its properties as close as possible, but now in the digital world, and preferably in real-time. The digital replica enables to predict the future behavior of the real object, or to test possible intervention before applying them in the real world.

Most DT applications are currently found in logistics, manufacturing, and operational research. Sensors play only a limited role in such applications. The connection from the real to the digital world is mostly implemented by RFID readers. Information consists of discrete values, e.g., the object ID number, or it can be reduced to a two-stage output, giving whether the object has arrived at the expected location or not.

However, in general, sensors are giving continuous output signals, such as temperature, humidity, or concentration of a chemical substance. In this article, we ask and review how such continuous sensor output can be integrated into DT.

Continuous sensors require mathematical processing, mostly far beyond simple threshold checking, including prediction of future values, filling gaps due to reduced set of sensors, and estimation of product life cycle or quality data. For example, when the core temperature of an object is not possible to measure directly, this gap can be filled by deducing it from the surface temperature, where the sensor is easier to install. Deviations from recommended operation and storage conditions cause a certain amount of stress to the object, such as wearing of a mechanical object or quality loss of a food product, leading to another class of models.

These models must be updated after each new measurement. This real-time feature of DTs is often stressed in literature, but practical recommendations how models can be made updatable for processing real-time or live sensor data is almost missing [2]. In this article, we consider this question in detail.

The paper is organized around an example from our own research. Since more than 10 years, we have worked on the remote monitoring of refrigerated ocean containers with food products. Five field tests were carried out with our Intelligent Container (IC) for the transport of bananas from Central America to Europe. The last three field tests also included artificial ripening inside the container [3]. During these tests, we already evaluated required IoT technologies for DTs such as wireless sensors, communication gateways, and data transfer via Wi-Fi, satellite, and cellular networks.

In cooperation with experts from agricultural science, we developed various models to predict biological and thermal processes inside the container. The green-life model predicts the expected timespan until an unwanted ripening process commences and the bananas can no longer be used commercially. A second model estimates a factor to describe the thermal coupling between circulating cooling air and a banana box. A third model estimates the current heat production by biological processes as indicator for the progress of ripening.

In summary, Sensors and IoT technologies are available, as well as prediction and analysis models, and now, the transformation into a DT seems to be the next logical step, leading us to the question:

What are the essential new features, challenges, and concepts of DTs, helping us to include live sensor data in process models?

2 Definitions and related projects

Modelling the relation between temperature deviations and resulting quality changes in agricultural products has a long tradition in biological research. A common scale to describe the fruit quality is the shelf-life, giving the remaining number of days until the quality falls below an acceptance threshold and the product must be disposed. A list with model parameters for 60 different fruits and vegetables can be found in [4]. Online tools to calculate the shelf-life for 6 products were provided by the FRISBEE project [5]. Some temperature data loggers provide an integrated shelf-life model. The Verigo Bluetooth logger warns by a LED on predicted quality problems [6].

These solutions are limited to manual data transfer to the mathematical simulation, or to the online platform. In other data logger based solutions, data exchange is only possible at certain checkpoints via Bluetooth or RFID readers.

IoT solutions, such as the remote container monitoring system by Maersk [7], focus on the machinery state of the cooling unit and container location and less on the product temperature in the cargo hold, and therefore lack a detailed analysis of product temperature data.

Recent research combines computational fluid dynamics (CFD) modelling for the influence of packing and airflow with biological models [8] but is also limited to simulation. However, the importance to link models with individual sensor data was emphasized by the same group of authors [1]: *"each shipment is subject to a unique and unpredictable set of temperature and gas atmosphere conditions"*.

The prevalence of simulation models led [9] to the question: *"Is it a DT or just a model?"*

Instead of excluding some applications from being a DT, Uhlenkamp et al. [10] suggested a taxonomy to evaluate and compare different DT applications, saying that: "*each DT is legitimate in its corresponding context, which makes an overarching definition of DT more abstract and thus difficult to comprehend and to imagine"*.

Their taxonomy includes seven dimensions or scales. We highlight only two dimensions, which have the most relevance for the question of sensor integration. The first one is the data integration level: Basic digital twins offer only offline simulation with manually feeding recorded data into models. More advanced solutions offer automated data transfer from the real object to the twin. The most advanced stage is achieved by two-way communication to send back information or control commands to the real object.

The 'Goal Type' of the DT application is the second scale with high relevance for sensor integration:

- Basic twins only acquire and display sensor information about the object by using IoT technologies.
- Second stage is information analysis by evaluating internal properties of the objects and prediction of the future behavior of the object.
- In virtual experiments, as third stage, the outcome of corrective actions can be tested in simulation of the DT platform before applying the intervention on the real object. A what-if scenario, for example, can test the effect of different temperature setpoint values on the temperature and product quality at the end of the transport.
- The fourth stage can be implemented by an automated system for testing different possible interventions and selecting the most beneficial one. The feedback loop is closed by sending a control command back to the real object to trigger the corrective action.

The above-mentioned biological applications have mostly a low data integration level. This is not only due to lacking communication technology but also to lack of knowledge on how a complex simulation model can be converted into an updatable format for integration of live sensor data. In the next section, we will focus on this question.

3 Hidden states and integral models

In general, a physical object, such as a box of fruits inside the container, has more quantities of interest than the number of actual measurable properties. The first ones are the system states $\boldsymbol{x}(t)$, and the latter ones the system outputs $\boldsymbol{y}(t)$. Additional known control inputs, such as the setpoint temperature, are denoted as $\boldsymbol{u}(t)$.

In our reefer container example, it might only be possible to install a sensor at the box surface. The current surface temperature is the first type of system state, which can be measured directly, although sensor noise must be considered.

The box core temperature is the first 'hidden' state, which cannot be directly measured but deduced from the measurements by adequate modelling. Bananas can produce tremendous amounts of heat by converting starch to sugar. The current biological activity or heat production can be considered a second 'hidden' state, although it requires more advanced filtering for estimation.

Other system states might be completely non-observable, for example the concentration of an enzyme contributing to quality loss, but without causing temperature changes.

The first question when planning sensor integration into a DT should be to ask, which states are directly measurable, hidden, or non-observable [11].

In some cases, the estimation of a hidden state can be straight forward, especially for accumulated changes of a quality attribute, e.g., mechanical wearing of a component or the shelf-life. The object starts with an initial budget of quality. According to the length and magnitude of deviations from the optimal transport and handling conditions, a certain amount is subtracted from the budget for each time interval. In mathematical terms, the state must stand in a direct integral relation to a measurable property.

Unfortunately, most hidden states require a more elaborated approach for estimation, as introduced in the next section.

4 Systems theory and state observers

The principles behind estimating hidden states can be best understood in the light of systems theory. The mathematical description of the system behavior is given in the so-called space-state form (**Figure 1**). In this, the changes of the states $\boldsymbol{x(t)}$ over time are described by ordinary differential equations with $\boldsymbol{x_0}$ as the unknown initial state and the control variables $\boldsymbol{u(t)}$ as inputs. In the linear case, the differential equations can be written in matrix form. Unknown stochastic influences are added as system noise $\boldsymbol{w_I(t)}$ to the changes of the states. The measurable outputs are a linear combination of the states with added measurement noise $\boldsymbol{w_M(t)}$.
The Kalman filter [12] is a common approach to estimate the internal system states based on the known or measured system inputs and outputs. In principle, such a state observer can estimate multiple states based only on a single measured variable, but a poor relation between the number of input and target variables makes the filter more noise sensitive.
For the non-linear case, more elaborated methods can be applied such as Hidden Markov Processes [2].

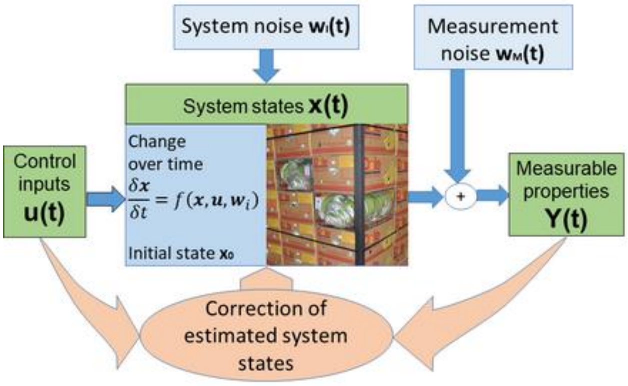

Figure 1 System description in state-space form and principle of state observance

The prediction of object properties, which cannot be directly measured, is the first key challenge for sensor-based DTs. Beside the above-introduced state-observers, machine-learning techniques can be applied. Therein, it must be kept in mind, that problems of non-observability remain, independent of the estimator approach.
The ability to provide estimations for such 'hidden' properties is one of the key features, in which DTs go beyond previous IoT solutions and offline modelling.

5 From single to multiple models

An extensive and concise description of a physical object has to contain several models. For various reasons, it is recommendable to handle the models separately on the DT platform. Models can originate from different research groups. They might be available in different mathematical description formats and programming languages. They might even only run on a dedicated server, e.g., for CFD airflow simulations.
The models can often be arranged in a processing chain. A sensor provides the temperature data. A first model calculates the expected future development of temperature for the remaining transport duration. Based on this prediction, a second model predicts the quality state for the expected time of arrival. A third process decides about possible interventions and sends back control commands to the physical object.
In our recent research, we linked the models in a single software program. However, for an increasing number of models and transport scenarios, such proprietary solutions are less adequate.
This leads us to the second key feature, which drives DTs beyond IoT and isolated modelling approaches. DTs provide software platforms for linking models of different sources.
One common approach to implement such platforms is the software pattern of "event-drive architecture". Sensors and models communicate through topics. A sensor publishes its measurement to a certain topic, which holds, for example, the temperature data for one container. All models are considered as event-driven process.

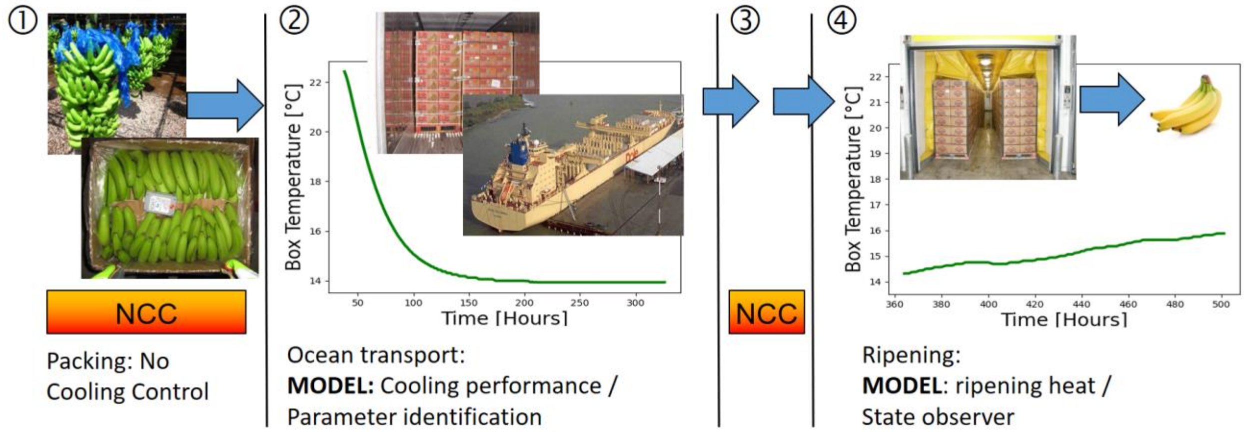

Figure 2 The banana cool chain and processing chain. Related models (NCC = no cooling control)

After subscription to a topic, they are notified, when new data is available. Each sensor measurement is processed as soon as it becomes available. Model results are published to another topic, to which in turn other subsequent models subscribe.

In this way, models can be programmed and tested individually, even in different programing languages. Publish and subscribe interfaces have been provided for Java and Matlab [13].

Topics and subscriber information is handled by a special data base, the so-called streaming platform. The open-source Apache Kafka platform is dedicated to event processing [14]. More details about event processing, our platform implementation and tests of its performance can be found in a separate contribution [13].

6 Models for the banana chain

Bananas are transported in a green, unripe state from Central America to Europe. After unloading, ripening is initiated with the help of ethylene gas in special chambers. The first four phases of the banana chain (**Figure 2**) were considered in our project: 1) The bananas are washed and packed after harvest without cooling. 2) The bananas are loaded 'warm' to a reefer container and cooling starts. Cooling continues during ocean transportation. 3) Between arrival at the port in Europe and the ripening chamber, cooling is interrupted for few hours or even days. 4) The ripening process is carried out either in a special chamber, or−as in our project−directly inside the Intelligent Container.

The green-life model should be applied until phase 3. After initialization of artificial ripening, it is no longer valid. Other models should only be applied during an individual phase, e.g., the temperature prediction model for container cooling in phase 2, and the ripening model in phase 4. Models should be assigned to certain timeslots marked by transport events such as transport start, and -arrival, ripening start, and its completion.

In the following, we summarize two models, which we selected as examples for DT integration. More details about these models can be found in our earlier publications [15], [16].

The bananas are cooled down from typically 25°C to 14°C during the first few days in the reefer container in phase 2. The speed of the cooling process varies inside the container, depending on position, the diameter of gaps between pallets and their packing. The efficiency of cooling, or the coupling of the banana boxes to the airstream, is described by the first time-constant model parameter k_M. Even in the green state, bananas produce a lower quantity of thermal energy, given by the second parameter k_P.

By skilled formulation of the model equations [15], the estimation of the two parameters can be reduced to a linear system identification problem (**Figure 3**). Moreover, it is possible to formulate the parameter identification in incremental form, with the estimates becoming more accurate with each additional temperature measurement.

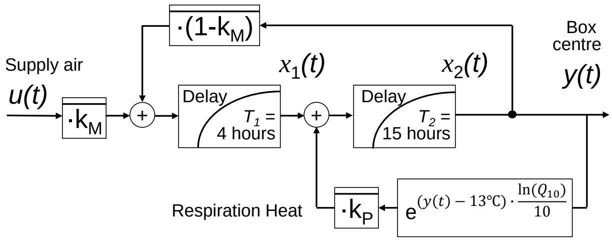

Figure 3 Parameter identification model in phase 2. For details see [15]

During ripening in phase 4, the heat production of bananas largely increases, and the heat production can no longer be considered as a constant parameter k_P, but as an additional unknown time-variable system state. However, the estimation of k_M can be considered as completed. The heat removal from the box by the cooling unit can be calculated based on k_M, thus enabling the estimation of the biological heat production from the model. The model structure is modified according to **Figure 4**. The model states can be estimated by the Kalman filter [16]. By observing changes of the heat production, it can be monitored whether the ripening process has fully started, or heat production raising beyond a critical value and ripening must be stopped by forced cooling and ventilation.

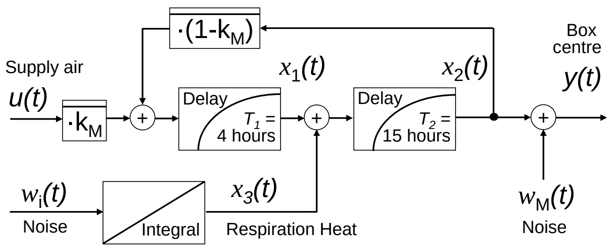

Figure 4 Modified model for estimation of ripening heat in phase 4. For details, see [16]

7 Model management

The event-driven architecture largely simplifies the assignment of the models to certain timeslots, and in- and outputs. A configuration file can be defined for each cargo type. It contains a list of steps that must be carried out at each transport event. Models can be loaded, started, or stopped. The configuration file also contains lists with input and output topics for each model.

The sensors send their measurements as JSON (JavaScript Object Notation) format. Transport events can be sent separately or combined with the sensor data. When the DT platform receives a transport event, it translates the event with help of the configuration file into a command for a certain model. The commands are published to a 'configuration' topic.

The models can be programmed independently from each other. All models subscribe to the configuration topic. If a model detects a command marked with its related name, it starts or stops processing the data from the input topics.

For example, the ripening model receives the start command after the ‘start-ripening’ event. It subscribes to two input topics. From the output of the cooling parameter identification, it reads the last estimated k_M value. Only sensor data with a timestamp later then the ‘start-ripening’ event are processed from the second input topic and the Kalman filter is updated after each new sensor reading. The estimated heat production is written to an output topic.

Models can run directly on the server, or, if they require another operating system, they can run on a separate workstation. Models can be exchanged with a newer version without stopping the whole framework. They only require a network connection to the Kafka streaming platform.

8 Digital twin prediction results

In order to generate reproducible results, the DT platform was not tested with live sensor data from a real transport, but with a set of recorded data from previous tests. Measurements, recorded with an interval of one hours, were played back with accelerated speed, i.e., one measurement per second, or 10 measurements per second. The required time to test a full data set was reduced to few minutes. The DT platform receives either real sensor data or playback data over the same interface. In contrast to the example in section 3, we were able to install sensors directly in the center of the boxes.

The first model updates the estimation of the k_M value after each measurement (**Figure 5**). After initial fluctuations, the estimation converts to a stable value when temperature data for a period of 4 days becomes available.

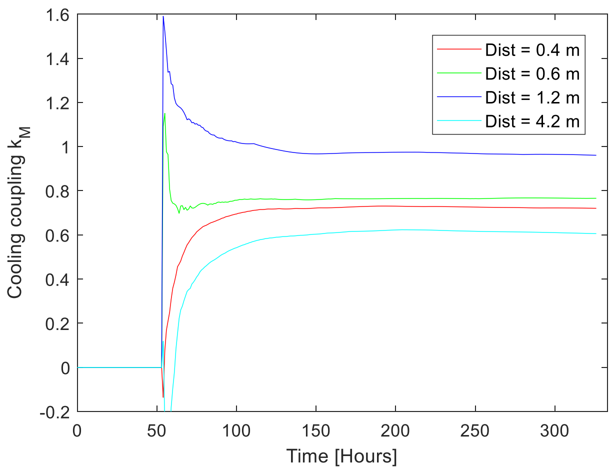

Figure 5 Change of estimated k_M values over time for banana boxes in different pallet positions inside the container. The legend gives the distances from the boxes to the cooling unit.

The last estimated k_M values were fed into the ripening model. The values were used to calculate an initial estimate for the state x_3. After start of ripening, the heat production could no longer be described with such a constant. The increase of the biological heat production during the ripening process is shown in **Figure 6**.

After 5.7 days, the ripening process was completed. The bananas were removed from the container and placed in a chamber with higher ventilation and cooling power to stop the process.

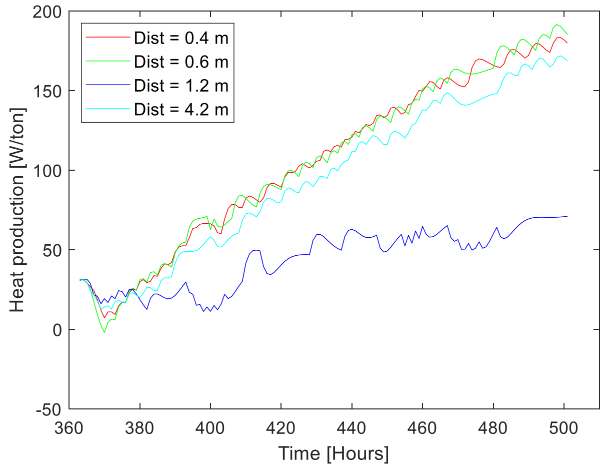

Figure 6 Change of heat production during ripening. Same pallet positions as in Figure 3.

The test of what-if scenarios is one of the advanced features of DTs. A container can have a poor cooling performance caused by wrong packing or old machinery equipment. The problem might become obvious only after the ship has left the harbor. In this case, the operator can test the effect of setpoint changes on the cargo temperature. **Figure 7** demonstrates such a what-if scenario. The container had left the packing station with the setpoint adjusted to 13°C. After 100 hours the what-if scenario was started. In a first step k_M, k_P and the initial state x_0 are estimated. Afterwards the same model in as Figure 3 is applied with the new setpoint value as *u(t)*.

Similar to transport events, a what-if scenario is triggered by a query event with the setpoint under test and the remaining transport duration as additional parameters, e.g., via a graphical user interface. The query can be repeated at a later point of time to achieve a more accurate estimation of k_M and temperature prediction.

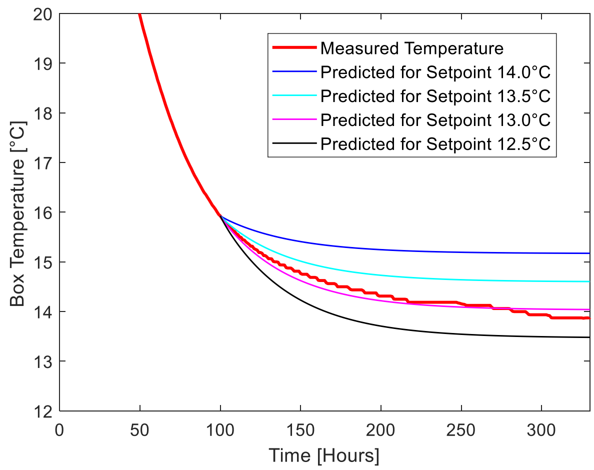

Figure 7 Setpoint scenarios, effect on predicted box temperature. Comparison with the actual measured temperature for a setpoint of 13°C.

Nevertheless, there is only little freedom to adjust the setpoint in the banana cooling chain. The scenario above was rather programmed for demonstration purposes.
A more practical what-if scenario must include the fruit processing and delivery planning, e.g., containers, which are predicted to have a low green-life at arrival, can be prioritized for faster handling. The effect on other containers, which must be postponed in turn, should be tested in advance.

9 Summary and Conclusions

We had already provided a full functioning prototype for remote quality monitoring in 2013. Two major drivers made us re-think the concept and software structure:
a) Technology drivers: Our initial concept was based on local data processing. The Intelligent Container was able to decide itself about possible quality risks. Communication was reduced to warning messages on detected risks and occasionally status messages. Due to the dramatic decrease of cellular communication costs, there is no longer a reason to withhold the data locally. Instead cloud computing has become the new paradigm.
Modern streaming platforms can forward data to multiple processing instances within few milli seconds [13]. There is no longer a need to combine multiple models in a single software unit due to performance requirements.

b) Conceptual drivers: DTs have become a new focus in research on transport and production logistics. This actually gave us the motivation to reconsider the concept of our Intelligent Container. In summary, we identified and applied the following new features, challenges, and concepts of DTs in our project:

1) Real-time data processing: DTs are not merely displaying real-time sensor data, but they also process the data to provide additional information. Although this feature is stressed in most articles about DTs, there are few guidelines, how it can be applied to a concrete model, e.g., to estimate non-measurable properties from the sensor data. The conversion from offline simulation models to an updateable model for real-time data is still one of the big challenges in DTs. The theory of state observers is now more than 50 years old, nevertheless, each individual model requires to re-write its mathematical description.

2) Model linking: DTs provide flexible platforms to link multiple models and other processing instances. There are several solutions available beside our Kafka based platform, both open-source and commercial [11]. The platform requires only minor changes to host new model types. Most work is required to write and adapt a wrapper function for the models to communicate with the DT platform. The actual linking can be done by assigning certain input and output topics to the models.
Our models had been tested individually with real-time sensor data, but we were lacking a solution to forward data among multiple models. With the DT streaming platform, we can take advantage of several model chains. The estimated k_M parameter for cooling performance from the first model provides the necessary input for subsequent models, e.g.:

- Prediction of future temperature development, followed by a third model to estimate the green-life at arrival.
- Estimation of the current heat, produced by the ripening process.
- Predict the effect of setpoint changes on the future temperature development in what-if scenarios.

In summary, the concept of DTs turned out to be motivating and useful for the inclusion of live sensor data in process models. On the example of the Intelligent Container we showed, how a remote sensing application can be adapted to take more advantages of DT concepts. Although there is no general solution to make a model updateable, most of the described steps can be adapted to other application for extending IoT solutions with advanced DT features.

10 Literature

[1] Defraeye, T., Shrivastava, C., Berry, T., Verboven, P., Onwude, D., Schudel, et al.: Digital twins are coming: Will we need them in supply chains of fresh horticultural produce? Trends in Food Science & Technology, 109, 245-258 (2021). doi:10.1016/j.tifs.2021.01.025

[2] Cronrath, C., Ekström, L., Lennartson, B. Formal Properties of the Digital Twin – Implications for Learning, optimization, and Control. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), 679-684 (2020) doi:10.1109/CASE48305.2020.9216822

[3] Jedermann, R., Lang, W.: 15 Years of Intelligent Container Research. In M. Freitag, H. Kotzab, & N. Megow (Eds.), Dynamics in Logistics: Twenty-Five Years of Interdisciplinary Logistics Research in Bremen, Germany (pp. 227-247). Cham, Springer International Publishing, 227-247 (2021). doi:10.1007/978-3-030-88662-2_11

[4] Tijskens, L. M. M. (2004). Discovering the Future: Modelling Quality Matters. (Ph.D. Thesis), University of Wageningen. Retrieved from http://library.wur.nl/WebQuery/wurpubs/lang/334193

[5] Gwanpua, S. G., Verboven, P., et. al.: The FRISBEE tool, a software for optimising the trade-off between food quality, energy use, and global warming impact of cold chains. Journal of Food Engineering, 148, 2-12 (2015). doi:10.1016/j.jfoodeng.2014.06.021

[6] Jedermann, R., Praeger, U., Lang, W.: Challenges and opportunities in remote monitoring of perishable products. Food Packaging and Shelf Life, 14(A), 18-25 (2017). doi:10.1016/j.fpsl.2017.08.006

[7] Zarkani, S., Rasmussen, C. H. Remote reefer monitoring looking back and looking forward. In: Cool Logistics Global, 8th global conference, Bremen, Germany (2016)

[8] Defraeye, T., Tagliavini, G., Wu, W., Prawiranto, K., Schudel, S., Assefa Kerisima, M., et al.: Digital twins probe into food cooling and biochemical quality changes for reducing losses in refrigerated supply chains. Resources, Conservation and Recycling, 149, 778-794 (2019). doi:10.1016/j.resconrec.2019.06.002

[9] Wright, L., Davidson, S.: How to tell the difference between a model and a digital twin. Advanced Modeling and Simulation in Engineering Sciences, 7(1), 13 (2020). doi:10.1186/s40323-020-00147-4

[10] Uhlenkamp, J. F., Hribernik, K., Wellsandt, S., Thoben, K. D. Digital Twin Applications: A first systemization of their dimensions. In: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 1-8 (2019) doi:10.1109/ICE.2019.8792579

[11] Jedermann, R., Lang, W., Geyer, M., Mahajan, P. (2022). Digital Twin features for the Intelligent Container. In: Freitag, Kinra, Kotzab, Megow (eds.): Dynamics in Logistics. Proceedings of the 8th International Conference LDIC 2022, Bremen, Germany. Springer, International Publishing, 2022, pp. 217-228. doi: 10.1007/978-3-031-05359-7_18

[12] Brown, R. G., Hwang, P. Y. C. (2012). Introduction to random signals and applied Kalman filtering: with MATLAB exercises (4. ed ed.). Hoboken, NJ: Wiley.

[13] Singh, K., Jedermann, R. Performance of a Digital Twin platform for tracing quality changes in fruits. In: 21 ITG/GMA-Fachtagung Sensoren und Messsysteme 2022, Nürnberg (2022)

[14] López, C. E. B.: Real-time event-based platform for the development of digital twin applications. The International Journal of Advanced Manufacturing Technology, 116(3), 835-845 (2021). doi:10.1007/s00170-021-07490-9

[15] Jedermann, R., Geyer, M., Praeger, U., Lang, W.: Sea transport of bananas in containers - Parameter identification for a temperature model. Journal of Food Engineering, 115(3), 330-338 (2013). doi:10.1016/j.jfoodeng.2012.10.039

[16] Jedermann, R., Lang, W. Model based estimation of biological heat generation during cold-chain transport and processing. In: 3rd IIR International Conference on Sustainability and the Cold Chain, St Mary's University, Twickenham, London, UK(2014)