Digital Twin features for the Intelligent Container

Reiner Jedermann[1][0000-0002-0390-9143], Walter Lang[1][0000-0002-9813-5231], Martin Geyer[2][0000-0001-5185-4880] and Pramod Mahajan[2][0000-0002-6597-9841]

[1] University Bremen, Institute for Microsensors, -actuators and –systems (IMSAS), Germany
[2] Leibniz Institute for Agricultural Engineering Potsdam-Bornim, Germany
rjedermann@imsas.uni-bremen.de

Abstract. The "Intelligent Container" for remote monitoring of refrigerated transports of fresh fruits already implements typical features of digital twins, including remote sensing and modeling of physical and biological objects. This article asks how the Intelligent Container can be extended to make the best use of digital twin concepts. Existing applications in agricultural science focus on offline simulation models that can predict shelf life and the effects of packaging on cooling but cannot integrate real-time data or correct their current estimates according to those data. This update feature is considered a key component of digital twins. The related challenges and algorithms can be best understood from the viewpoint of systems theory and state-space description. Internal properties of real objects can be either directly measurable, hidden, or unobservable, and implementation of the update process should be adapted accordingly. Using ocean transport and banana processing as an example, this paper demonstrates how models can be made "updateable", in addition to discussing the necessary steps for linking different sub-models over a standardized platform according to the "publish/subscribe" pattern.

Keywords: Digital twin, Intelligent Container, state observer, updatable model, food logistics, cold chain.

This is a preprint of the following chapter:

Jedermann, R.; Lang, W.; Geyer, M.; Mahajan, P.: *Digital Twin Features for the Intelligent Container*. In: Dynamics in Logistics. LDIC 2022, Bremen, Germany, Springer International Publishing, 2022, pp. 217-228. (doi: 10.1007/978-3-031-05359-7_18)

The final authenticated version is available online at **https://dx.doi.org/10.1007/978-3-031-05359-7_18**

1 Introduction

The "Intelligent Container" (**IC**) was developed over 10 years ago to monitor temperature conditions and consequent changes in the quality of fruit during ocean transportation to improve logistical process planning [1]. Although IC development began long before the term digital twins (**DTs**) was used publicly, DTs now represent a popular concept for remote management of various complex systems, including shop floors, chemical plants, and wind farms [2]. Agricultural applications of DTs have been reported since 2017 [3].

The IC already implements some typical DT features, including real-time remote sensing and modeling of physical objects. This article's major focus is using DT concepts and applications to guide future extensions of the IC, which can be considered a proxy for other remote sensing and modeling applications in food logistics that contend with similar issues. Furthermore, we consider special challenges associated with modeling biological objects.

The scientific literature includes several definitions of DTs. According to [4], *"a digital twin is a virtual, dynamic model in the virtual world that is fully consistent with its corresponding physical entity in the real world and can simulate its physical counterpart's characteristics, behavior, life, and performance in a timely fashion."* Alternate definitions are listed in [5], and a more detailed five-dimensional definition can be found in [2].

However, there is still no binding definition of DT, with [6] writing 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"*. They further suggest using a taxonomy of seven dimensions or features to classify different DT applications. Their approach indicates that low implementation levels for some features can be sufficient for certain applications and do not provide a basis for rejecting classification as a DT. Nonetheless, it is worth considering whether additional benefits can be achieved by increasing the implementation level.

Although [7] provides a more ambitious definition of DTs in agricultural science, this iteration is almost impossible to achieve, even with current state-of-the-art modeling: *"As such, this digital replica evolves and reacts hygrothermally and metabolically in a similar way as its physical counterpart—a real fruit or vegetable—but now in-silico and preferably in real time."*

The similarity of the twins can be maintained only if changes to the physical object are continuously measured, and the digital counterpart is updated accordingly. [8] raised the question, *"how to tell the difference between a model and a digital twin"*. Key point is the 'digital thread' [8] to update the model based on real-time sensor data. Given the literature mostly neglects the complexity of this update process, its principal difficulties and solutions comprise this paper's second special focus.

The sensor system is mostly realized by Internet-of-Things (IoT) technologies, and the complexity of the DT's data processing varies largely depending on the target application. According to the taxonomy of [6], the processing can be classified into four stages or goal types. The first three are acquiring data, predicting future changes to the physical object, and conducting virtual experiments. The latter provides the option of testing possible interventions on the digital platform before applying them to the real object. The fourth action implementation stage involves the DT automatically adjusting controls for the physical object, thus closing the feedback loop.

Biological objects differ from work-pieces in the manufacturing scenarios in which DTs have traditionally been deployed. Related models often entail high numbers of "hidden" states that cannot be directly measured. Section 2 discusses the consequent challenges and the update process involved in realizing the digital thread. Section 3 considers DT software realization and demonstrates how simulation models can be converted to an updateable form. Having applied this concept to the IC, Section 4 summarizes the steps involved in converting a monitoring solution to a DT-compatible platform.

1.1 The Intelligent Container

The temperature history during cold chain operations substantially impacts the quality of agricultural products, although this is often not visible from the outside. A green banana remains a green banana, whether it was transported at perfect temperature and atmospheric conditions or whether the temperature was a few degrees too warm. However, in the latter case, an unwanted ripening process can commence shortly after the transport's arrival, making the fruit useless for further commercial processing. By revealing hidden changes in quality to the operator, fruit in a critical state can be prioritized for harbor handling and further processing; in the worst-case scenario, a timely replacement delivery can be organized.

The IC system comprises a network of wireless sensors inside the cargo hold that measure temperature and other environmental conditions. The sensor data is processed by a gateway mounted in the container, with the effects of temperature deviations evaluated by a so-called shelf-life or green-life model. The gateway sends either full sensor data or warning messages alone to a cloud server via cellular or satellite networks [9], allowing the user to make decisions based on forward projection of the predicted shelf-life, i.e., to assess whether the consignment will arrive in an acceptable quality state after the expected transport duration. Making the actual quality of agricultural products visible enables stock rotation optimization. Assigning deliveries according to the first-expires first-out approach can reduce losses of highly perishable products by 8% to 14% of the total volume [10].

1.2 Digital twins in the food supply chain

Modeling has a long tradition in agricultural science, especially for predicting the effect of temperature deviations on shelf-life. [11] provides model parameters for the shelf-life of 60 different fruits and vegetables. Elsewhere, an online tool developed by the FRISBEE project [12] provides shelf-life models for six food products and enables simulation of the effects of different transport modes on total energy consumption and shelf-life, with one model for apples including the effect of air humidity on moisture loss. The FRISBEE tool also enables virtual experiments that can compare the effects of different cooling and transport modes on product quality and carbon footprints. Meanwhile, [13] provides a range of mathematical models for simulating accumulated gas concentrations in the headspace above fresh produce and related shelf-life changes.

Other models based on computational fluid dynamics (CFD) simulation analyze spatial airflow and temperature profiles in containers, trucks [14], and inside boxes according to different packing designs [15]. [16] combined CFD modeling for temperature conditions and biological models to create a DT for mango fruits, and [17] used data loggers to collect temperature data from various transports, allowing detection of weak spots in the cooling chain using detailed thermal and quality modeling of the fruit.

Although purely offline modeling has value, automated data integration arguably generates more benefits. For [7], the supply chain for agricultural produce needs not merely models but DTs because *"each shipment is subject to a unique and unpredictable set of temperature and gas atmosphere conditions"*.

Other DT applications in agriculture have IoT origins. According to [3], although 28 applications have emerged since 2017, most are at a conceptual or prototype stage. Most deployed applications focus on remote sensing, such as monitoring the filling height of feed silos for livestock. Only two other deployed applications include detailed modeling; these use deep learning to monitor animal behavior or proactively prevent tractor malfunctions.

Remote sensing solutions for refrigerated containers can be split into "remote container monitoring" systems, which focus on the maintenance and operation state of the cooling engine, and wireless data loggers, which can be placed inside boxes or pallets. The missing link between these two solution types [9] remains an obstacle for implementing DTs.

2 Making models updateable

Applying multidisciplinary knowledge is among the key enablers of DTs [2]. DTs should provide a holistic model of a real object. Models for fluid dynamics, heat transfer and biochemical processes must be selected, programmed, and linked, with the complexity of modeling representing a challenge of its own. However, even if a complex simulation model is available, one DT-specific challenge remains; the set of models must be continuously "updateable" using real-time sensor data. Despite the importance of this key feature of DTs [8], few articles on DTs describe how the update process can be realized. We suggest three levels for categorizing solutions to implement the update process.

First, the update is implemented by simply overwriting obsolete sensor data in the model. This is especially possible if the system state can be described using discrete values, which are measurable without noise, such as the location of a workpiece at a certain machine on a shop floor. A model tracking the workload of each machine simply has to overwrite information at each new RFID scanning event.

The second level includes incremental models. Crucial system properties often have an accumulative character, such as a component's accumulated mechanical stress. In the food logistics context, agricultural products have a high initial shelf-life, with a certain amount of quality lost each day depending on the current deviations from the optimal transport conditions. For computational efficiency, instead of recalculating the whole model, a decrement is recalculated according to the incoming real-time sensor data.

Such incremental models are only applicable if the critical indicator—i.e., quality—has a direct integral relation to a measurable quantity. Otherwise, a more complex update method is necessary. This third level entails methods that estimate system states that are not directly measurable. The general approach can be best understood within the framework of systems theory as discussed in the following section. The update process can be realized as a state-space observer. An overview of related mathematical methods regarding DTs can be found in [5]. Nonetheless, this paper focuses on the consequences from systems theory rather than on mathematical details.

2.1 State-space description

The systems theory perspective indicates that each dynamic system can be described by a set of time-dependent variables $\mathbf{x}(t)$, or, in short, the state vector $\mathbf{x}$. Vectors are demarcated using bold font in the following. The transitions of $\mathbf{x}$ can be described using a set of first-order ordinary differential equations [18] with the initial value $\mathbf{x_0}$. The change of states over time $\frac{\partial x}{\partial t}$ is given by a function of current state $\mathbf{x}$, the control vector $\mathbf{u}$, and system noise $\mathbf{w}_\mathrm{i}$. The measurable quantities of the physical object $\mathbf{y}$ are given by a linear combination of the system states $\mathbf{x}$ and measurement noise $\mathbf{w}_\mathrm{M}$ (Fig. 1).

Except for laboratory experiments, typical biological objects have fewer measurable outputs than internal states. Certain internal states are of less interest, and some can be excluded to produce a simplified model. However, others give crucial information regarding the object, including, for example, the degeneration of a biochemical substance, which indicates the loss of fruit quality over time. A simple example model for a fruit box might comprise three states, namely, the box core, surface temperatures, and the remaining concentration of the biochemical substance; the latter is not directly measurable. Vector $\mathbf{u}$ contains the system inputs or control variables; in this case, the cooling setpoint and the ambient temperature. The measurement noise $\mathbf{w}_\mathrm{M}$ includes the intrinsic noise of the sensor element, amplifier noise, and quantization noise of the analog-digital converter. The system or process noise $\mathbf{w}_\mathrm{i}$ describes unpredictable time-varying fluctuations in system states with known and unknown causes, including direct thermal noise, time-varying airflow due to turbulence, and random variations in the speed of biochemical reactions.

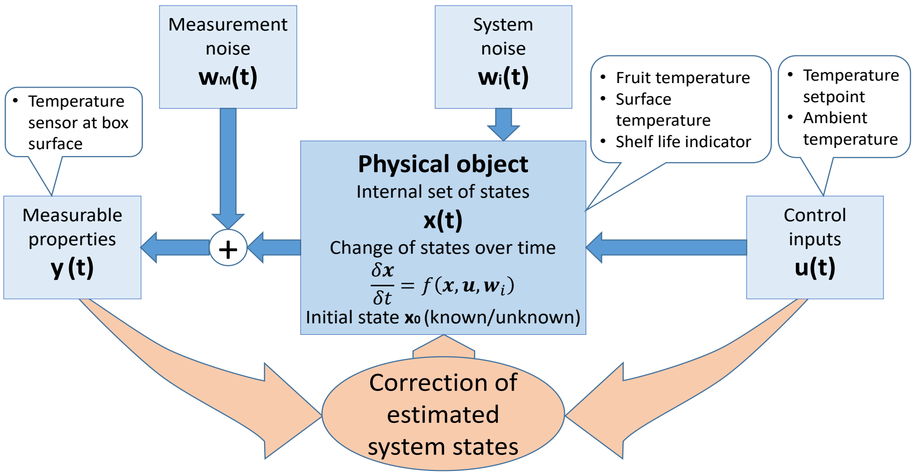

Fig. 1. State-space description of dynamic systems. Example properties are given in the white legends, and the principle of the update process is in orange.

In the case of a linear system, the differential equations simplify to (1), with matrix **A** describing the time behavior of the states and their mutual influences. Matrix **B** quantifies the effects of the inputs on the states.

$$\frac{\partial \dot{x}}{\partial t} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{w_i} \tag{1}$$

The DT model's first task involves estimating "hidden" states that cannot be directly measured, such as the fruit quality indicator. Next, the model predicts the system's future behavior. If the current state is at least roughly known, the model can predict future changes to **x** and the expected measurement values **y**. Due to noise, errors in the estimation of the model parameters **A, B** and the initial state $\mathbf{x_0}$, predictions increasingly deviate from the real object over time. The update process must correct the estimated system states based on the available information, namely, the measured values **y** and the known control inputs **u** (see lower part of Fig. 1).

A so-called state observer provides methods for correcting the states based on the error between actual measurement **y** and prediction $\mathbf{y_p}$, with the Kalman filter [18] representing the most common realization of such. A state can be either

- *Directly measurable*, for example, surface temperatures, in which case, the Kalman filter can be deployed to reduce measurement noise); or
- *Hidden*, for example, core temperatures, which are not directly accessible by a sensor but impact surface temperature, making it feasible to estimate core temperature via long-term observation of the surface. Shelf-life cannot be directly measured. Destructive laboratory methods cannot be applied by real-time surveillance. However, if shelf-life losses relate to increased respiration activity or heat production, shelf-life is at least theoretically observable. However, in practice, noise reduces the accuracy of such estimations.
- *Not observable*, if quality changes do not cause temperature changes or other measurable outputs. In such cases, future projections based on the assumed initial value $\mathbf{x_0}$ are possible, but no later corrections can be made based on the measurements. Improvements compared with an incremental model are mostly unfeasible.

Practical systems often feature non-linear relations, such as the relationship between temperature and shelf-life loss. In such cases, a Hidden Markow model can realize state observation [5]. Nonetheless, the general problem of observability remains, independent of non-linearity and of the model structure, even for black-box or machine-learning models. The DT literature mostly neglects such observability problems [5].

2.2 Complexity of biological systems

In moving from initial applications of DTs with pure mechanical objects to agricultural products involving biological systems, model complexity largely increases. Numerous chemical and enzymatic reactions must be considered. Although some of them can be excluded for simplified modeling, real-time measurements cannot access most of these states.

If modeling begins at the point of harvest, the initial state $\mathbf{x_0}$ underlies variations due to influences such as nutrition, weather conditions during the growing period, mechanical damage during harvest, and the position of the fruit on the tree or plant. However, even fruits with identical harvest conditions demonstrate substantial differences in ripeness state. This biological variance creates a certain degree of modeling and state estimation uncertainty that must be considered during model deployment and networking with subsequent models. For example, our tests identified typical deviation of ±5 days in the shelf-life of bananas despite these bananas being harvested at the same time and transported and stored at the same temperature [19].

Furthermore, values of model parameters, such as the matrices **A**, **B**, can deviate or even drift over time. The diameter of gaps between pallets in a container affects airflow and, thus, temperature. The distribution of gaps is not known in advance due to careless stowage and non-cuboid pallet shape. Pallets might even move slightly during transportation, changing gap diameters.

Our IC project developed a method [20] to describe the heat transfer from supply air to box temperature using a surrogate model featuring only two free time-constant parameters. During the two-week ocean transportation, these two parameters were identified from the temperature data: cooling efficiency and heat generation by ripening processes. Upon arrival, the fruit was exposed to ethylene to initiate the ripening process. Thereafter, the first model parameter was fixed to the identified value, and the second parameter was replaced with a time-dependent state variable that could be estimated using a Kalman filter.

3 Software structure of digital twins

The updateable models must be interlinked and fed the correct data. Values for past and current real-time readings and predictions for the future must be combined on the timeline as model inputs. Past model inputs can be logged control inputs to the physical object, direct sensor readings, Kalman filtered sensor data, or simulated outputs of preceding models. Future model inputs must be generated differently, with control inputs mostly assumed to remain at the last known value; furthermore, instead of sensor data, the related quantities should be predicted using the preceding models.

The complexity of the required software platform increases with goal type. The action implementation stage includes bi-directional data exchange over the digital thread and complex decision-making based on a tool that tests and compares different possible interventions using virtual experiments.

3.1 Software components

The numerous possible configurations for connecting different sub-models with data streams for control inputs, sensor readings, model outputs and predicted values drives the need for standardized software solutions. Such platforms provide modularity for easily swapping or adding sensors and models. Furthermore, direct point-to-point linking of the models using proprietary interfaces becomes almost unmanageable for complex systems. Event-drive architectures using the "publish/subscribe" pattern are a common method for providing generic interfaces for DT platforms, with [21] suggesting an architecture comprising three layers:

- A streaming platform or **persistence layer** acts as a mediator between sensors and models. Sensors publish their data to so-called topics or message queues. Models subscribe to certain topics to receive notifications of new data becoming available and retrieve information from other models.
- The DT is linked to the physical object by the **communication and integration layer**. Basic components are sensors that capture the process state and mostly wireless communication that provides a secure and reliable channel for transmitting process information.
- The sub-models are programmed as **event-driven services** that react to published events such as the availability of new sensor data or sensed distortions of the physical process.

Although more complex descriptions of DTs exist, e.g. [2], the basic software components can be described in terms of these three layers. Numerous software platforms for DTs are available, with Source Forge listing more than 40 entries on its "Best Digital Twin Software" list [22] and Dashdevs including 20 commercial solutions on their list [23]. Eclipse IoT projects provide several components associated with implementing a DT [24]. The development of a specific DT can be based on such solutions.

For example, communication and integration can mostly be handled by existing IoT solutions and enhanced using project-specific sensor types. Notably, the Message Queuing Telemetry Transport (**MQTT**) protocol has become one of the most common standards for transmitting real-time sensor data (Fig. 2). Several research projects [21, 25] use Apache Kafka as an open-source streaming platform for the persistence layer that is combined with the Eclipse Hono protocol adapter for MQTT.

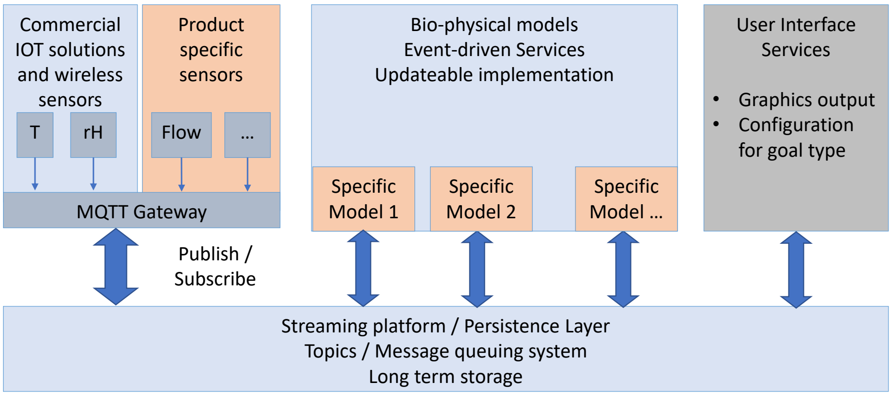

Fig. 2. Typical DT software architecture. Necessary project-specific extensions to standard software and available IoT solutions are marked in orange.

Although some platforms entail general models, such as time-series analysis and deep learning, agricultural applications require the development of fruit and transport-chain-specific models that depends on the user.

The system is completed with the introduction of user interfaces for graphical output and control.

3.2 Converting simulation models into event-driven services

Describing physical objects using a set of sub-models should be as concise and extensive as possible. The first step involves collecting available models and then fine-tuning them in laboratory experiments for the specific product before completing the set of sub-models with the missing elements. In the second step, the observability of system properties should be verified. Models that are only available in an offline form for simulation purposes must be converted into an updateable form in the third step, enabling the model to react to events such as the availability of new real-time data. If crucial system states are unobservable, an incremental model can represent the best solution. Otherwise, it is worthwhile pursuing optimal use of available sensor data using a state observer or a statistical- or machine-learning-based approach.

Step four considers the uncertainties of model prediction, especially where states are unobservable. Error propagation for the uncertainty of model inputs must be included and forwarded through the model chain. Notably, sensors are a cost factor, requiring consideration of whether some of the sensors could be replaced by modeling without significantly increasing the prediction's overall uncertainty.

Finally, step five programs interfaces for communication with the streaming platform, which typically follows the "publish/subscribe" pattern.

3.3 From the Intelligent Container to a general digital twin

Given our IC prototype was designed for demonstration purposes, it was focused on one specific product in one transport chain with a specific stowage scheme. Two example models were tested to predict the effect of temperature variations on the green-life of bananas and to predict box temperature as a function of a coefficient for cooling efficiency and a proportional factor for ripening heat.

The aforementioned application of a Kalman filter as observer demonstrates the complexity of linking models in different configurations for the processing of real-time sensor data. Thus far, we have only demonstrated that our embedded processing platform has sufficient resources to run the Kalman filter. The model was only tested by manually combining recorded and predicted data from transport and ripening processes.

For real-time deployment, numerous services must be linked in different configurations, including biological models, modules for parameter estimation, and data streams. Data streams must switch between past measurements from the physical object and future predictions from the DT models, with the latter becoming obsolete upon the arrival of new real-time data.

A more generalized solution is preferable to the earlier point-to-point linking of modules. Streaming platforms provide more flexibility to exchange modules. For example, if a new model provides a more accurate estimation of the coefficient for cooling efficiency by direct airflow measurement, it can directly publish its prediction to the related topic after disabling a less accurate estimator, and subsequent models do not have to know where the data derives from.

The IC demonstrator was mostly based on temperature sensors. The accuracy of prediction can be improved by integrating new sensor types such as 2-dimension airflow sensors and nondispersive infrared ethylene sensors [1]. Such sensors should implement DT-compatible protocols (e.g., MQTT). Accordingly, commercially available and project-specific sensors can be easily incorporated into the IC.

Although other models—such as the influence of humidity on shelf-life and water condensation on fruit [26] and the headspace model for the accumulation of humidity and gases [13]—are available in principle, these have not been integrated into the IC.

The most important task is converting available models into event-driven services as the previous section described. Notably, compliance with DT standard solutions enables the IC to be integrated into a system of systems. A DT for order and transport management can include several sub-systems representing each refrigerated container.

4 Summary

The success of DTs in other applications fields drives the motivation to represent perishable products in the cool chain by detailed digital counterparts. Modeling the related biochemical and heat-transfer processes is critical to taking advantage of DT concepts.

Furthermore, different research groups have established a focus on either fruit modeling or IoT technologies for real-time monitoring. Neither a binding definition of DT nor practical considerations preclude such applications from being considered a DT because each DT is legitimate in its corresponding context [6]. Instead, it is important to consider how certain applications can benefit from implementing additional DT features. The necessary combination of models with real-time data leads to the question, how the model can be updated by new incoming sensor measurements. Incremental models or simply overwriting obsolete values are often insufficient for updating the DTs of food products.

Adopting systems theory with state-space descriptions provides a framework for understanding the problems and opportunities associated with a more elaborate update process. Although unmeasurable quantities of a physical object might be hidden, they are often observable in principle by their effects on other measurable values.

The steps to making better use of DT features in the IC entail implementing state-observers for hidden quantities, converting models to event-driven services, and implementing streaming platforms. These steps are the same for other agricultural processes. The streaming platform can be implemented using open-source solutions or commercial cloud services, with IoT sensor solutions broadly available. DT platforms provide adapters to handle various standard protocols, and only product-specific sensors remain to be added for special measurement tasks. Notably, the lack of models is the key obstacle to increasing the use of DTs in agriculture.

Agricultural models and remote sensing applications are generally programmed as proprietary solutions. The models of several research groups have been published as mathematical algorithms or as online tools with a graphical user interface. Digital twins provide the most useful platform for combining and integrating such models following reprogramming as event-driven services. By applying the principles this article describes, available simulation models can be prepared to be part of a DT solution that offers better real-time monitoring and control in the food logistics field.

References

1. Jedermann, R., Lang, W.: 15 Years of Intelligent Container Research In: Freitag, M., H. Kotzab, H., Megow, N. (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
2. Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., et al..: Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58, 3–21 (2021). doi:10.1016/j.jmsy.2019.10.001
3. Pylianidis, C., Osinga, S., Athanasiadis, I. N.: Introducing digital twins to agriculture. *Computers and Electronics in Agriculture,* 184, 105942 (2021). doi:10.1016/j.compag.2020.105942
4. Zhuang, C., Liu, J., Xiong, H.: Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *The International Journal of Advanced Manufacturing Technology,* 96(1–4), 1149–1163 (2018). doi:10.1007/s00170-018-1617-6
5. 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), pp. 679–684. (2020). doi:10.1109/CASE48305.2020.9216822
6. 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
7. Defraeye, 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
8. 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
9. 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
10. Lang, W., Jedermann, R.: What can MEMS do for logistics of food? Intelligent container technologies: a review. *IEEE Sensors Journal,* 16(18), 6810–6818 (2016). doi:10.1109/JSEN.2016.2576287
11. Tijskens, L. M. M.: Discovering the future: modelling quality matters. (Ph.D. Thesis), University of Wageningen. (2004). Retrieved from http://library.wur.nl/ WebQuery/wurpubs/lang/334193
12. Gwanpua, 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
13. Jalali, A., Linke, M., Geyer, M., Mahajan, P.: Integrative programming for simulation of packaging headspace and shelf life of fresh produce. *MethodsX,* 8, 101514 (2021). doi:10.1016/j.mex.2021.101514
14. Moureh, J., Flick, D.: Airflow pattern and temperature distribution in a typical refrigerated truck configuration loaded with pallets. *International Journal of Refrigeration,* 27(5), 464–474 (2004). doi:10.1016/j.ijrefrig.2004.03.003
15. Ambaw, A., Mukama, M., Opara, U. L.: Analysis of the effects of package design on the rate and uniformity of cooling of stacked pomegranates: numerical and experimental studies. *Computers and Electronics in Agriculture,* 136, 13–24 (2017). doi:10.1016/j.compag.2017.02.015
16. 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
17. Shoji, K., Schudel, S., Onwude, D., Shrivastava, C., Defraeye, T.: Mapping the postharvest life of imported fruits from packhouse to retail stores using physics-based digital twins. *Resources, Conservation and Recycling,* 176, 105914 (2022). doi:10.1016/j.resconrec.2021.105914
18. Brown, R. G., Hwang, P. Y. C. (2012). Introduction to random signals and applied Kalman filtering: with MATLAB exercises (4th edn.). Wiley, Hoboken, NJ.
19. Jedermann, R., Lloyd, C., Poetsch, T.: Communication techniques and challenges for wireless food quality monitoring. *Philosophical Transactions* of *the Royal Society A,* 372(2017), 20130304 (2014). doi:10.1098/rsta.2013.0304
20. 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, International Institute of Refrigeration (IIR), London, UK (2014)
21. 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
22. SourceForge: Slashdot Media: best digital twin software. https://sourceforge.net/software/digital-twin/, last accessed 2021/10/26.
23. dashdevs: Product owner talks: 20 digital twins solution providers. https://dashdevs.com/blog/product-owner-talks-20-digital-twins-service-companies/, last accessed 2021/10/26.
24. Eclipse Foundation Europe GmbH: Eclipse IoT open source projects. https://iot.eclipse.org/projects/, last accessed 2021/10/26.

25. Kamath, V., Morgan, J., Ali, M. I. Industrial IoT and digital twins for a smart factory: an open source toolkit for application design and benchmarking. In: 2020 Global Internet of Things Summit (GIoTS), pp. 1–6 (2020). doi:10.1109/GIOTS49054.2020.9119497
26. Linke, M., Praeger, U., Mahajan, P. V., Geyer, M.: Water vapour condensation on the surface of bulky fruit: Some basics and a simple measurement method. *Journal of Food Engineering*, 307, 110661 (2021). doi:10.1016/j.jfoodeng.2021.110661