Enhancing Manufacturing Agility: The Role of Digital Twins and AI in Reconfigurable Production Systems

Enhancing Manufacturing Agility: The Role of Digital Twins and AI in Reconfigurable Production Systems

Matteo Iacovanelli1*, Giovanni Mazzuto1, Marco Ortenzi1, Filippo Emanuele Ciarapica1, Maurizio Bevilacqua1, Bjorn Vanderwal 2

1 Departement of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, Ancona, Italy

2 Software Departement

Marelec Food Technologies Nieuwpoort, Belgium

Abstract — This paper highlights the development and implementation of a digital twin for a pilot within the AGILEHAND project, aimed at improving planning and resource management in food processing. By integrating real-time data, the digital twin enhances forecast accuracy, optimizes raw material usage, and ensures efficient scheduling. The system models the physical production line, including multiple processing stations and conveyors, using an open source discrete-event simulation. Additionally, key elements such as products and operators are modelled to reflect real-world interactions, allowing for detailed analysis of production line behaviour across various scenarios. Moreover, Artificial intelligence plays a critical role in optimizing operations, managing order processing, and addressing inventory overlap to ensure smooth transitions between production phases. This comprehensive approach facilitates precise production tracking and adaptive scheduling, ultimately improving operational efficiency and resource utilization.

Keywords — Digital Twin, Data-driven, Digital Modelling, Food industry, Simulation, Logistics, Industry 4.0

INTRODUCTION

Managing the production of soft, deformable, and perishable food products presents significant challenges due to the dynamic nature of production systems, especially for the production manager that continuously must shift the production. Digital Twins offer a data-driven framework for automating the creation of simulation models, which support real-time and near-real-time planning, synchronization, and reconfiguration of production and logistics systems. This approach helps to minimize deviations in Estimated Time of Arrival (ETA).

The European Project AGILEHAND addresses these challenges to supports the production manager by leveraging Digital Twin technology to enhance resource utilization and align more effectively with demand. In this context, the European AGILEHAND project focuses on developing a Digital Twin for a production plant in the food industry, specifically aimed at the autonomous grading, handling, and packaging of soft and deformable products. The goal is to enhance the flexibility, agility, and reconfigurability of production and logistics systems within European manufacturing companies. Additionally, the project seeks to implement a traceability system integrated with data-driven Digital Twin technology, coupled with Artificial Intelligence (AI) based solutions. This will enable the monitoring, adaptive control, and synchronization of production and logistics flows within factories, even when faced with variability in products, production mix, or fresh market demands. The objective is to ensure high performance in customer response times and optimize resource usage. Moreover, the project aims to develop advanced algorithms for order management, warehouse management, and production scheduling, further optimizing the overall industrial process.

Building on previous work that developed a DT framework using machine learning and process mining techniques to continuously improve the model and optimize logistics projects [1], this paper presents a modelling approach that utilizes empirical observations (real data). It also develops Python-based software that communicates with a real production plant through an MQTT protocol. One of the key pilot applications of the DT is Marelec, a leading manufacturer of intelligent portioning, customized weighing, and grading solutions for the poultry, meat, seafood, and marine industries.

In a competitive environment, companies have been pushed to invest in advanced technologies[2]. A considerable challenge arises in the production of products with a flexible or deformable structure, particularly in sectors such as fresh food[3]. Here, companies have to balance the need to produce different, high-quality products at low cost and within short lead times, while ensuring environmentally sustainable practices and meeting customer specifications[4]. These dynamic conditions require rapid reallocation of resources, flexible reconfiguration of systems, and frequent rescheduling of orders[5]. Reconfigurable Manufacturing Systems (RMS), introduced by Koren, have emerged as a solution[6]. While originally designed to minimise production costs and reduce lead times, these systems often paid limited attention to customer needs[7].

The objectives of the digital twin for the company Marelec are outlined, along with an analysis of the production system.

In Section 2, Section 3 explains how, by using SimPy and discrete event modelling, the digital twin simulates the system's behaviour and optimises processes. Finally, Section 4 examines how artificial intelligence and linear programming assist in configuring production lines based on customer orders and warehouse availability.

PREVIOUS WORK & EXISTING THEORIES

However, modern markets, characterized by fluctuating orders, demand variability, quality concerns, and unexpected disruptions, require a high degree of responsiveness.[8]. RMS achieves this through reconfigurability, which enables systems to be adjusted and reorganised repeatedly and efficiently[9]. Core characteristics like modularity, flexibility, scalability, and diagnosability allow these systems to swiftly adapt to changing production demands and market conditions, ensuring businesses remain agile and competitive[10]. Moreover, challenges include understanding the role of layout, interaction with other systems, and the use of simulations to address uncertainties in design. There is a clear need for a combined optimization and simulation approach to tackle variable and uncertain environments[7].

This growing need to optimize production processes has driven the adoption of Industry 4.0 technologies, which are fundamentally based on digitalization and connectivity [11]. This paradigm shift has introduced innovations such as the Internet of Things (IoT), Big Data, Artificial Intelligence (AI), and Digital Twins (DT). Among these, DT stands out as a transformative technology, as it integrates multiple advancements to provide a comprehensive understanding of system behaviour. By analysing vast amounts of critical variables, DT enables the optimization of key parameters, enhancing efficiency and decision-making[12]. Originally conceptualized by Grieves[13], Digital Twins extend human capabilities by facilitating system visualization, real-time monitoring, and predictive analysis. These elements form the foundation for a new era of intelligent problem-solving and continuous innovation[14].

Nowadays, the Digital Twin is evolving rapidly, becoming a cornerstone for improving the performance of physical entities through advanced computational techniques[15]. It consists of a physical entity, its virtual counterpart, and the data connections between them, creating a continuous feedback loop between the physical and digital worlds. With integration into the IoT, data from physical systems is transmitted to the digital counterpart for simulations and monitoring, optimising performance and preventing failures[14]. The Digital Twin has expanded across various industrial sectors, but a unified vision of what exactly DT represents and how it can be adapted to different applications is still lacking, with the risk of diluting its concept. DT offers immense potential to address business challenges, such as process planning and management, through advanced analysis and modelling techniques[16].

In the food processing sector, a Digital Food Twin could optimise traceability and supply chain management, integrating analysis and simulations to respond in real-time to critical events[17]. The use of advanced simulation models, such as Discrete Event Simulation (DES), is already common, but in more complex contexts, a dynamic approach is required that leverages technologies such as IoT and cyber-physical systems. With advanced simulation models and IoT technologies, the Digital Twin can tackle the complexities of production processes in real-time, improving operational efficiency and reducing errors[18].

At the same time, Warehouse and order management algorithms in the food industry are still evolving and require further attention. Existing models reveal both strengths and limitations, highlighting the need for stronger collaboration between businesses and research institutions while addressing challenges such as intellectual property protection and internal organizational resistance[19]. To improve adaptability, advanced technologies like artificial intelligence and digital twins provide innovative solutions. Reinforcement learning-based approaches have proven highly effective in handling complex, dynamic environments with infinite state spaces, enabling companies to identify optimal solutions faster than traditional scheduling methods[20]. Businesses that adopt AI-driven data processing through adaptive algorithms, predictive analytics, and machine learning can enhance responsiveness, anticipate future disruptions, and foster real-time collaboration with supply chain partners. Moreover, AI enhances post-disruption recovery by strengthening resilience and coordination within the supply chain network [21].

METHODS

Studying the Physical Asset

To begin the development of the digital twin for the Marelec pilot, it's necessary to follow a structured approach that starts with an understanding of the system characteristics, requirements, and potential complexities is essential[22]. This initial step provides a clear and comprehensive understanding of the real system, which is essential for creating a digital model that accurately reflects reality [23]. Once the production logic is clearly defined and a full understanding of the physical system is achieved, the modelling and simulation phase, advanced models will be developed to replicate the behaviour of the real system[24].

A digital model serves as a simplified representation of a real system and process. It highlights essential characteristics that define its properties and behaviour. Nevertheless, as digital model is designed for a specific purpose, it provides only a partial depiction of the real system and does not include every detail of the actual production process[25].

The production line initially consists of four input lines. ("Line1") and ("Line2") are similar and allow for the processing of chicken, either keeping the fillets and inner together or separating them. On the other hand, (“Line3”) and (“Line4”) only produce fillets and inner as a single piece. In each line there is one operator in a station (“Uploading”) who places the chickens onto the line (“Enter line”). The next station (“Fillet Disassembly”) involves manual fillet harvesting, where operators separate the two fillets from the bones. A visual pre-quality check is performed by the operators at the filleting line. If a B/C fillet is detected, operators remove it from the line to prevent contamination of the others. The fillets are then placed onto the fillet conveyor (“Fillet Conveyor”). (“Line1”) and (“Line2”) are for manual inner fillet harvesting, where operators separate the two inner fillets from the bones. The inner fillets are placed onto the inner conveyor (“Inner Conveyor”).

(“Inner Conveyor”) carrying the inner from the first two lines direct to a station where operators manually place the inner into trays ("Inner Assembly"). Once the desired weight is reached, the trays are placed on the conveyor ("Inner Final Conveyor"), which moves them to the packaging stations.

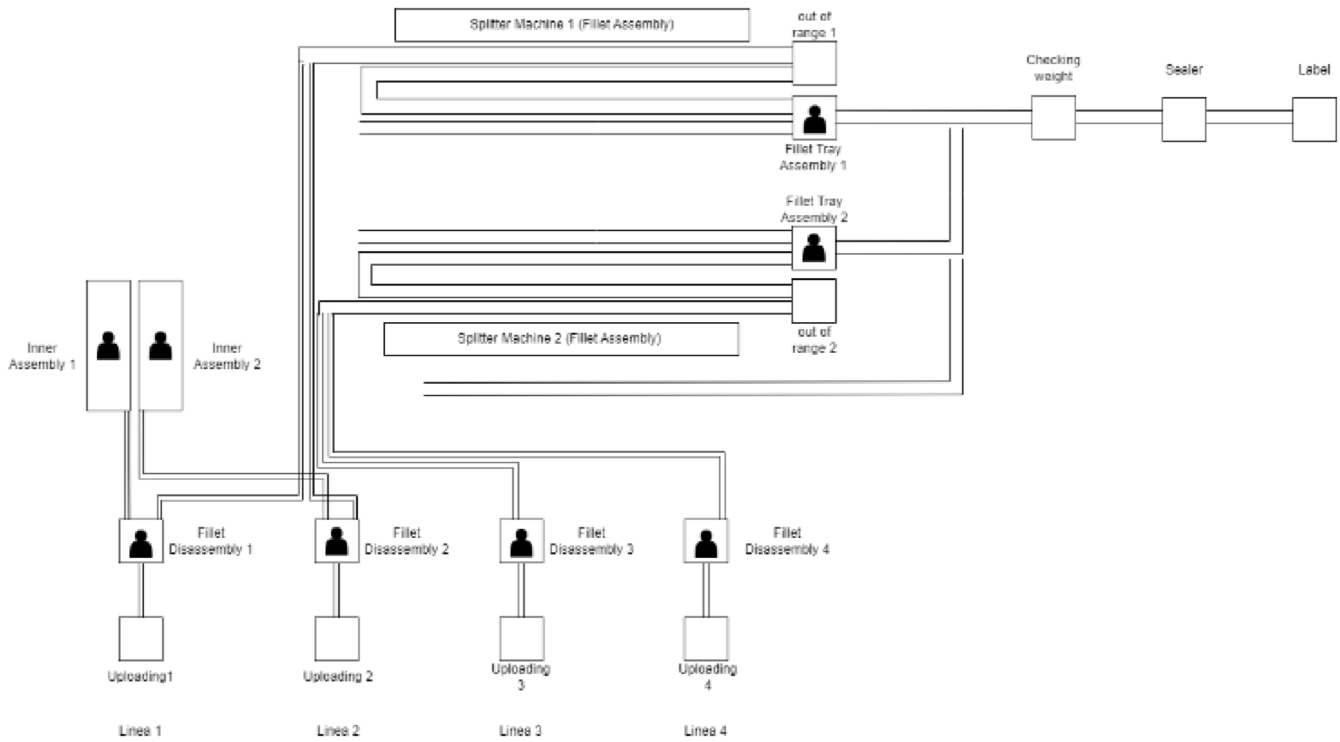

Figure 1 Marelec flowchart

The splitter machine is the most important station, where (“Fillet Conveyor”) from the first two lines feed into an automated station consisting of 30 splitters. At this station, a flow scale weighs and grades each product individually, ensuring a continuous throughput of raw materials. The material is divided and weighed into 30 sub-batches of 1, 2, 4.5, or 5 kilograms. Each batch consists of fillets within a narrow weight range, depending on the specifications of the order. Fillets that fall outside the required range are sent to the end of the line (“Out of Range”), where a floating operator removes the full box. This station is the core of the company, as it allows the production manager to schedule the production.

For this reason, several sensors are designed along the conveyor belt (“Fillet Conveyor2”), each with a weight range based on customer order specifications. If the weight of the fillet matches the conditions set by the sensor, it enters a station. Here, the station continuously monitors and accumulate the weight of incoming fillet. Each time a new fillet arrives; its weight is added to an ongoing total. When the total accumulated weight reaches the batch weight, it is sent to the (“Fillet Assembly”). Once the assembly has been done, it is transferred onto the ("Fillet Full Conveyor").

The ("Fillet assembly") plays a crucial role, as it is used by the production manager to monitor production and, in case of delays, to readdress the production. Subsequently, the ("Fillet Full Conveyor") and another conveyor carrying empty trays ("Conveyor Tray"), will proceed side by side to a manual station. At this station, operators will place the fillets into the corresponding trays ("Fillet Tray Assembly"). In this case, the fillets and trays moved alongside each other on two separate conveyors. This was achieved by keeping the station where the trays are stored and only allowing one tray to pass when the fillets were in the corresponding position. So, the fillets and the tray arrive at the “Fillet Tray assembly” at the same time.

Trays filled with fillets or inner from all lines are placed on the ("Final conveyor"). Once on this conveyor, the trays are reweighed, and if necessary, a floating operator adjusts the weight ("Checking Weight").

Finally, the trays are sealed with film ("Sealer") and labelled with the customer-specific label ("Label"). This last station provides the production manager with the status of the order completion.

All the production line, as explained before, are represented in Figure 1, the blocks represent the stations and the rows represent the conveyors.

Digital Twin scope

For Marelec, the digital twin is a new tool that provides immediate solutions to improve planning and resource management.

They need a reliable system for next-day planning, accurately forecasting raw material requirements and identifying the best suppliers using historical quality and yield data. Using real-time data, matching forecasts with Estimated Time of Arrival (ETA) calculations is essential to enable fast and accurate operational adjustments as conditions change and to maintain smooth workflows and effective scheduling. Marelec also seeks to optimise resources, recommending switching to alternative raw materials when this improves efficiency.

Modeling and Simulation

Model development within the digital twin ecosystem focuses on replicating real plant behaviour with high accuracy. To achieve this, SimPy, a process-based discrete-event simulation framework, was used. By exploiting SimPy capabilities, the dynamic processes of the plant can be simulated accurately, ensuring that the model aligns closely with the characteristics of the system under investigation[26].

In this context, moving objects like chicken, fillets, inners, and trays are treated as "entities" in the DES framework. Each entity is defined by attributes such as batch weight and quality. Operators are considered renewable resources in DES, switching between idle and active states based on events, while labels and plastic rolls for the sealer are classified as non-renewable resources.

Given that SimPy operates within a closed environment, customization or addition of variables is not possible. Therefore, a sub-environment was created with specific properties to capture detailed simulation data. An empty list within this sub-environment collects information such as clock time, entry and exit times at stations, and the start and end times of processes to measure both waiting and processing durations. Additionally, it includes details about the entities (chickens, fillets, or inners), their IDs, and weights customer-defined thresholds. The next step involves modelling the components of the system, including stations, resources, and entities.

In Figure 2, the fundamental role of a station can be likened to a buffer that stores entities. The hold() method within the class dictates how the station retains these entities, which are sourced from other stations identified in the "entrance" class attribute. Similarly, the "exit" attribute specifies the addresses of subsequent stations. While this does not constitute a fully functional station yet, its enhancement leads to the creation of the Base_station, which builds upon the Buffer concept. The Base_station can perform tasks with the help of operators (defined in the renewable resources attribute) and/or non-renewable resources (such as raw materials, defined in the non-renewable resources attribute). The priority attribute is essential for managing concurrency issues, ensuring that delays are avoided when an operator is needed by multiple stations simultaneously.

A station can autonomously manage entities by retrieving them from the previous station using the take_from() method, moving them to the next stations via the move_to() method, or waiting for other connected stations to complete their movements through the wait_for() method.

The system also requires an "Input Station", which acts as the generator of parts entering the processing line. Defined by the class name "Input_station", this extends the Base_station with the generate_entity() method, facilitating the movement of entities within the line. As a generator, the Input_station does not have an input station, so its "entrance" attribute remains empty.

Similarly to the Input Station, the Output_station class is designed to handle the disposal of entities once they exit the processing line. This is achieved through the destroy_entity() method. In this case, the "exit" attribute is left empty, as there are no further stations to route entities to.

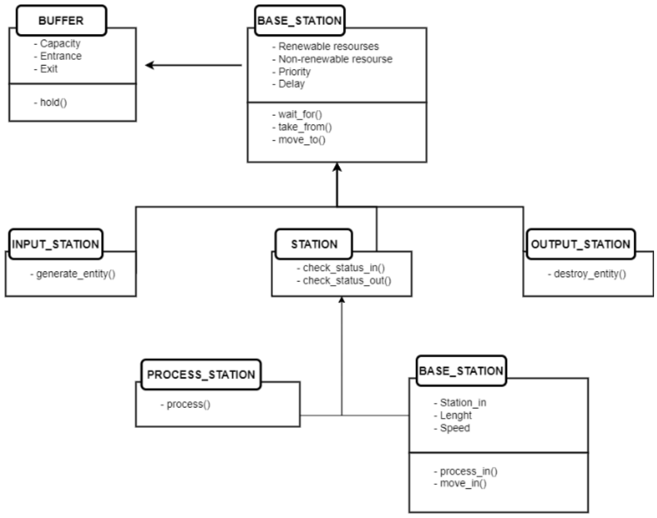

Figure 2 Station class diagram

Another extension of the Base_station class is the Station class, which introduces additional functionalities beyond those provided by the Input and Output stations. Specifically, the Station class includes methods such as check_status_in() and check_status_out() to assess whether the station has adequate capacity for incoming entities and to verify the availability of entities for processing. This foundation supports the development of more specialized stations like Process_station and Conveyor_station.

The Process_station class represents a basic processing station within the line, defined by the process() method. This class includes methods associated with the operations performed at the station. It can be extended with additional attributes or methods to accommodate more complex requirements, such as those needed for Fillet Assembly. For example, Fillet Assembly requires an attribute to specify the weight threshold to determine if a fillet batch meets the required specifications.

The Conveyor_station class is characterized by technical attributes such as length and speed. It supports various tasks along the conveyor, each described by the process_in() method. Additionally, the move_in() method commands the movement of entities along the conveyor.

The standard operator is represented by the Operator class, which includes attributes related to skills and assigned workstations. The Team Leader class extends the standard Operator class and incorporates additional responsibilities. This role involves defining and managing the tasks assigned to the other team members. By treating these modelled objects as modules, we can use them to simulate the physical processing line. This allows the digital twin to conduct simulations on a digital model that accurately mirrors the real system.

In the Marelec case study, there are three types of entities: chickens, fillets, and inners. These entities share common attributes defined in the Entity class. Each entity must have attributes for type identification, an ID, quality, weight, and dimensions, which are specified by dedicated methods within the class. The Chicken, Fillet and Inner classes are specialized extension of the Entity class, tailored to handle the specific characteristics of chickens.

As is possible to see in Figure 3, the Chicken class includes several attributes that identify the entity, along with three methods for assigning these attributes to entities. The methods chicken_base(), chicken_copy() and lines(), are responsible for generating only the "chicken" entity with generic attribute and assigns the number of chickens to be processed for each line.

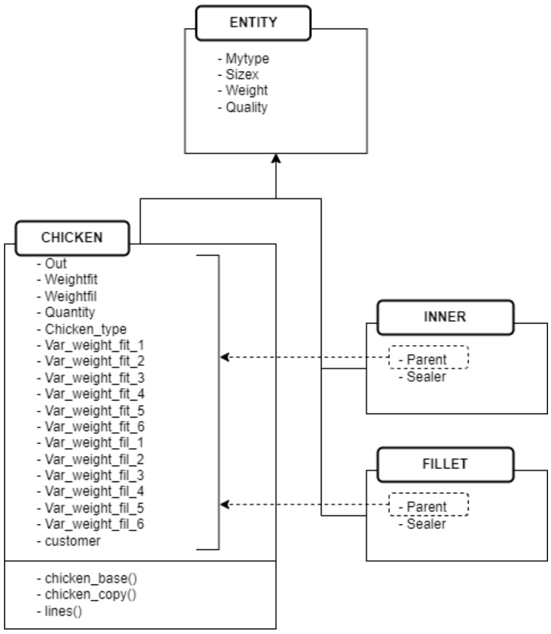

Figure 3 Entities class diagram

The definition of Fillet and Inner class are different, in fact they have only two attributes. The attribute sealer indicates whether the entity has a sealer or not, while the attribute parent is an extension of the Chicken class. This attribute allows both classes to inherit the attributes of the "Chicken" class. This inheritance is essential because the three entities are interconnected. In fact, at the beginning of the simulation, a single object "Chicken" is created, which encompasses all other entities. When this object enters in the disassembly station, the station divides "Chicken" in five objects: "Chicken", "Fillet1", "Fillet2", "Inner1" and "Inner2". To ensure the inheritance, the entities are organised in a group. The chicken serves as the parent entity, containing the other four entities: two fillets and two inners.

Artificial Intelligence

Production planning and control defines what, how much and when to produce, purchase and deliver to align production performance with customer demand[27].

For this, AI is a decision-support tool for companies that need to quickly identify the optimal parameter configuration to enhance production performance and planning. Digital twins for warehouse management can improve efficiency by providing decision-making support and advanced performance analysis[28]. Baruffaldi et al. [29] introduced an innovative decision-support tool for warehouse management, incorporating customer data, costs, and uncertainties related to return on investment.

As mentioned before, in Marelec the most important station is the splitter machine. It allows the production manager to decide which type of chicken to select for entry into the four lines, whether to separate the fillet from the inner and how to configure the splitters.

This process ensures that the chickens selected correspond to the weight range required by the customer, allowing orders to be processed on time.

For this reason, a Python code was developed which, depending on the quantity of chickens in warehouse and customer orders, determines the following to fulfil each order:

- the type of chicken to be used.
- whether to separate the fillet from the inner.
- the quantity of chicken required.
- the parameters for configuring the splitter.

First, two classes, Warehouse and Order, were created. The “Warehouse” is a simple class defined by attributes that allow to identify the chickens in stock

The ”Order” class, in addition to having attributes to identify customer orders, includes three functions: splitter_line_1_2(), splitter_line_3_4() and replace_order(). The first two functions configure the settings of the splitter station to be configured for lines 1 and 2 and lines 3 and 4, respectively. The third function updates the order to be processed once an order is completed. This allows the splitter set up to be reconfigure with the new parameters for the next order, ensuring a seamless transition between orders.

From the classes described above, another class was created. The “Overlap” class calculates the overlap percentage between customer orders and the chickens in stock, determining the best chicken for each order. It processes a list of orders, comparing the weight ranges of the orders with those of the chickens in the warehouse. The overlap_percentage_calculation() method calculates the overlap percentages for the “Fillet W” (fillet + inner) and “Fillet W/O” (only fillet) of each type of chicken. For each order, it identifies the category of chicken with the greatest overlap and calculates the average weight for that category. The results are stored in a list that includes details such as customer, total quantity, most suitable chicken categories and overlap percentages for “Fillet W” and “Fillet W/O”. This helps determine whether to use chickens with or without fillets.

Now that the best chicken type has been determined, it is necessary to create the chicken object using the “Chicken” class, as showed in 3. The Chicken_base() method generates a list of chicken objects based on the results provided by class “Overlap”. It randomly assigns weights to the different parts of the chicken (fillets and inners), ensuring that each chicken object reflects the correct category and size. Additionally, it determines the total quantity and whether the chicken should be processed as "Fillet W/O" or "Fillet W". The Chicken_copy() method is similar to the previous method but allows a single chicken object to be created.

The Lines() method is responsible for distributing the quantity of chicken between the different production lines (Lines 1-2 and Lines 3-4). This division is necessary only Lines 1-2 allow for the selection between "Fillet W/O" and "Fillet W", while Line 3-4 can only produce "Fillet W". The method splits the total quantity into two smaller iterations, calculates the weight of each part and creates the required chicken parts (fillets and inner) for each line. It then keeps track of the number of chickens produced per line and updates the count for each size category. In this way, the system can efficiently manage the production process, optimising the distribution of chickens according to their category and weight, enabling the production manager to know how many chickens and which types need to be allocated at the beginning of each line for each order. Figure 4 represents all the connections between the models to obtain the results.

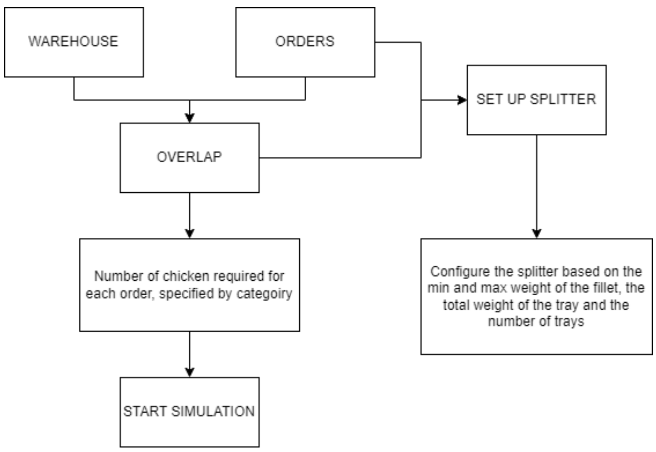

Figure 4 Algorithm diagram

The research follows a discrete event-based approach, setting it apart from physics-based methodologies. Although the data used in this work are hypothetical due to privacy concerns, the approach effectively demonstrates the potential of the technology in optimising warehouse management and order fulfilment. The simulation, based on realistic scenarios, successfully replicates the real system's behaviour, providing valuable insights for production planning, resource allocation, and efficient order completion. This highlights the system's ability to streamline operations and support decision-making, showcasing its potential for real-world applications in production environments.

RESULTS

Due to privacy concerns related to the Marelec company, the data presented in this section are not real. However, the primary aim of this work is to demonstrate the technological potential and the results it can achieve. The data used in the simulation are hypothetical, with an initial warehouse storage condition and hypothetical orders created for the purpose of this demonstration.

WAREHOUSE		
Supplier	**Category**	**Chicken Quantity**
A	CK440-550	15000
A	CK550-750	8500
A	CK>750	7500
B	CK440-550	9000
B	CK550-750	5000
B	CK>750	6500
C	CK440-550	9000
C	CK550-750	10000
C	CK>750	7000
D	CK440-550	8000
D	CK550-750	9000
D	CK>750	7000

Table 1 Hypothetical warehouse

In the warehouse (Table 1), it is possible to view the storage of raw materials, classified by supplier, chicken type (category the minimum and maximum chicken weight obtainable), along with the total quantity of chickens for each classification.

ORDERS				
Customer	**Trays number**	**Tot tray weight**	**Min unit weight**	**Max unit weight**
Customer A	4000	5000	150	180
Customer B	5500	4500	180	210
Customer C	5000	2500	210	240
Customer D	3000	1000	240	270

Table 2 Hypothetical orders

In the Table 2, orders are classified by customer, total number of trays, total batch weight, minimum and maximum weight of each raw material inside, and the delivery deadline.

ALGORITHM RESULTS						
Custom er	**Complet ed tray**	**Total tray**	**Missing tray**	**Used chicken**	**Typology used**	**check**
Customer A	2000	2000		34482	400-550 WITH	OK
Customer A	2000	2000		31250	550-750 WITHOU T	OK
Customer B	113	5500	5387	12550	*550-750 WITH	NOT OK
Customer C	0	2500	2500	0	*550-750 WITH	NOT OK

Customer C	2500	2500		14534	>750 WITHOUT	OK
Customer D	3000	3000		5554	>750 WITH	OK

Table 3 Algorithm results

The algorithm generates, the first result, Table 3 showing the number of chickens used for each order, specifying the category.

From this table, the production manager can gain valuable insights, this approach highlights the effectiveness of the technology in streamlining warehouse management, order fulfilment, and restocking. It demonstrates the system's potential for real-world applications in production environments.

This table highlights several key aspects. Firstly, only the orders for customers A and D are fully satisfied, while those for customers B and C have been only partially fulfilled. It is possible to observe that customer A's order utilises two types of chicken (CK 400-550 WITH and CK 550-750 WITHOUT). This is because the algorithm suggests using higher-category chicken, separating the inners from the fillets to achieve greater consistency between the chosen chicken weight and the weight required by the order. However, this process is only feasible on line 1 and line 2. As a result, two different types of chicken are used for both customer A and customer C. Another important detail is the list of chickens used and the missing trays, allowing for better stock replenishment planning. Once this data is obtained, such as the number of chickens to be loaded onto the four lines and the splitter settings, the simulation can be initiated based on the SimPy framework. This allows for the successful replication of the real system's behaviour.

SIMULATION RESULTS			
Time	**Category**	**Total weight**	**Customer**
285.99	FIL	5014	Unclassified order
289.50	FIL	4960	Customer A
294.26	FIL	4427	Customer B
299.02	FIL	5040	Customer A
306.52	FIL	5028	Unclassified order
313.31	FIL	4986	Customer A

318.07	FIL	4445	Customer B
433.43	FIT	2426	Customer C
437.69	FIT	4932	Customer A
441.94	FIT	4469	Customer B
446.20	FIT	4913	Customer A
450.86	FIT	2430	Customer C
457.24	FIT	5052	Unclassified order
461.51	FIT	2480	Customer C
1290.31	FIT	1004	Customer D
1294.56	FIT	1032	Customer D
1298.82	FIL	4414	Customer B
1303.08	FIT	1070	Customer D
1307.33	FIT	1038	Customer D
1309.46	FIT	2468	Customer C
1313.71	FIT	2473	Customer C

Table 4 Simulation results

In particular, the simulation enables the viewing of output data and order completion times (Table 4). For each product, both the exit time and additional information, such as the total batch weight and the associated customer, are displayed. It is evident that orders for customers 1 and 2 are initially processed, as they fall within a similar weight range. As previously mentioned, two types of chickens are selected for customer 1. Moreover, since weight is a variable factor, it is possible to produce batches of similar orders using the same type of chicken.

We also observe that order 4 is only processed at the end, once customer 1’s order has been fully satisfied. At that point, the splitter updates the parameters to allow order 4 to be processed. Additionally, we note the presence of some remaining orders for customers 2 and 3, as they have not yet been completed, as shown in the resulting table. The "unclassified order" are those batches obtained from fillets that do not fall within any of the weight ranges requested by customers.

This makes it possible to calculate the estimated time of arrival (ETA).

Discussion

The results obtained from the simulation effectively address Marelec initial questions and needs. The proposed system demonstrates its ability to support next-day planning by accurately forecasting raw material requirements and identifying the most suitable suppliers based on historical yield and quality data. The algorithm optimizes resource usage by recommending alternative chicken categories when this improves efficiency, as shown in the fulfilment of Customer A’s order using two different types of chicken. The integration of real-time data allows the system to compare forecasts with Estimated Time of Arrival (ETA) calculations. This facilitates swift operational adjustments and ensures smooth workflows and efficient scheduling. Furthermore, the simulation highlights the system’s ability to detect partially fulfilled orders and unclassified batches, which improves stock replenishment planning and resource management. By simulating realistic production conditions, the DT approach not only answers Marelec operational challenges but also proves to be a technologically advanced and practical solution for streamlining production processes and enhancing strategic decision-making.

CONCLUSION

This study aims to introduce a digital modelling approach for a food production line, with a particular focus on soft and deformable products. The process spans from outlining the key development stages to the implementation of the model, starting with an artificial intelligence algorithm that assesses the availability of raw materials in stock and daily orders. This allows the production manager to determine the necessary quantities to order and configure the simulation.

However, using the Digital Twin (DT) with batch data rather than real-time data may yield accurate results but could prove inadequate given the dynamic nature of production lines. Factors such as handling, storage, and transportation conditions can change rapidly due to environmental influences, product maturity, and logistical variables, affecting the quality of products that are highly susceptible to damage and degradation.

Future improvements could involve the development of more intuitive interfaces to enhance the usability of this advanced technology, such as voice-recognition-based interaction systems. Additionally, establishing a direct connection with the real industrial plant and conducting final validation in the field using real-time data would be essential. In line with the objectives of the European project AGILEHAND, integrating real-time data into the DT framework represents a significant step towards more effective management of soft and deformable food products.

MATCH & CONTRIBUTION

This contribution aligns strongly with the theme of the ICE IEEE 2025 conference on "AI-driven Industrial Transformation: Digital Leadership in Technology, Engineering, Innovation & Entrepreneurship." The paper presents an AI-powered digital twin approach for production planning and optimization in the food industry, particularly focusing on soft and deformable products. By integrating artificial intelligence, simulation frameworks (SimPy), and discrete-event modelling, the research demonstrates how data-driven engineering can enhance real-time decision-making and resource allocation. The proposed system supports agile, next-day planning, enabling production managers to dynamically respond to order variability, supply chain constraints, and product characteristics. Furthermore, the work highlights the potential for real-world deployment by emphasizing the integration of real-time data streams and adaptive system behaviour. This contribution reflects the conference’s emphasis on digital transformation and innovation through AI, offering a practical and scalable solution for industrial challenges and advancing the field of intelligent manufacturing systems.

ACKNOWLEDGMENT

This paper is supported by European Union's Horizon Europe research and innovation programme under grant agreement No 101092043, project AGILEHAND (Smart Grading, Handling and Packaging Solutions for Soft and Deformable Products in Agile and Reconfigurable Lines).

References

[1] M. Iacovanelli, F. E. Ciarapica, G. Mazzuto, M. Bevilacqua, M. Ortenzi, and M. Rossi, “Digital twin framework for improved handling of soft and deformable products in food manufacturing,” *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*, pp. 1–7, doi: 10.1109/ICE/ITMC61926.2024.10794319.

[2] J. Lima, J. F. P. Moreira, and R. M. Sousa, “Remote supervision of production processes in the food industry,” *Volume 2016-January, Pages 1123 - 1127*, vol. 2016-January, p. 2015, doi: 10.1109/IEEM.2015.7385823.

[3] S. E. P. Robert L. Shewfelt, *Postharvest Handling* , Fourth Edition. 2022.

[4] S. Singh, A. Barde, B. Mahanty, and M. K. Tiwari, “Digital twin driven inclusive manufacturing using emerging technologies,” in *IFAC-PapersOnLine*, Elsevier B.V., Sep. 2019, pp. 2225–2230. doi: 10.1016/j.ifacol.2019.11.536.

[5] L. Ribeiro and L. Gomes, “Describing Structure and Complex Interactions in Multi-Agent-Based Industrial Cyber-Physical Systems,” *Volume 9, Pages 153126 - 153141*, vol. 9, pp. 153126–153141, 2021, doi: 10.1109/ACCESS.2021.3127344.

[6] Y. Kor&, “Reconfigurable Manufacturing Systems.”

[7] I. Maganha, C. Silva, and L. M. D. F. Ferreira, “The layout design in reconfigurable manufacturing systems: a literature review,” *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 1, pp. 683–700, Nov. 2019, doi: 10.1007/s00170-019-04190-3.

[8] L. Ribeiro and L. Gomes, “Describing Structure and Complex Interactions in Multi-Agent-Based Industrial Cyber-Physical Systems,” *IEEE Access*, vol. 9, pp. 153126–153141, 2021, doi: 10.1109/ACCESS.2021.3127344.

[9] R. M. Setchi and N. Lagos, “Reconfigurability and reconfigurable manufacturing systems - State-of-the-art review,” *2nd IEEE International Conference on Industrial Informatics, INDIN’04*, pp. 529–535, 2004, doi: 10.1109/indin.2004.1417401.

[10] Y. Koren and M. Shpitalni, “Design of reconfigurable manufacturing systems,” *J Manuf Syst*, vol. 29, no. 4, pp. 130–141, 2010, doi: 10.1016/j.jmsy.2011.01.001.

[11] N. Nikolakis, G. Siaterlis, X. Bampoula, I. Papadopoulos, T. Tsoukaladelis, and K. Alexopoulos, “A Digital Twin-Enabled Cyber-Physical System Approach for Mixed Packaging,” in *Advances in Transdisciplinary Engineering*, IOS Press BV, Apr. 2022, pp. 485–496. doi: 10.3233/ATDE220167.

[12] H. Alimam, G. Mazzuto, N. Tozzi, F. Emanuele Ciarapica, and M. Bevilacqua, “The resurrection of digital triplet: A cognitive pillar of human-machine integration at the dawn of industry 5.0,” Dec. 01, 2023, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2023.101846.

[13] Michael Grieves, “Digital Twin: Manufacturing Excellence through Virtual Factory Replication,” *White paper*, no. March, pp. 1–7, 2014, [Online]. Available: https://www.researchgate.net/publication/275211047%0Ahttps://www.3ds.com/fileadmin/PRODUCTS-SERVICES/DELMIA/PDF/Whitepaper/DELMIA-APRISO-Digital-Twin-Whitepaper.pdf

[14] P. K. R. Maddikunta *et al.*, “Industry 5.0: A survey on enabling technologies and potential applications,” *J Ind Inf Integr*, vol. 26, no. June 2021, p. 100257, 2022, doi: 10.1016/j.jii.2021.100257.

[15] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the Digital Twin: A systematic literature review,” *CIRP J Manuf Sci Technol*, vol. 29, pp. 36–52, 2020, doi: 10.1016/j.cirpj.2020.02.002.

[16] K. Y. H. Lim, P. Zheng, and C. H. Chen, “A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives,” *J Intell Manuf*, vol. 31, no. 6, pp. 1313–1337, 2020, doi: 10.1007/s10845-019-01512-w.

[17] C. Krupitzer, T. Noack, and C. Borsum, “Digital Food Twins Combining Data Science and Food Science: System Model, Applications, and Challenges †,” *Processes*, vol. 10, no. 9, 2022, doi: 10.3390/pr10091781.
[18] V. Fani, B. Bindi, and R. Bandinelli, “Designing and optimizing production in a high variety / low volume environment through data-driven simulation,” *Proceedings - European Council for Modelling and Simulation, ECMS*, vol. 35, no. 1, pp. 10–15, 2021, doi: 10.7148/2021-0010.
[19] B. Bigliardi and F. Galati, “Models of adoption of open innovation within the food industry,” *Volume 30, Issue 1, Pages 16 - 26*, vol. 30, no. 1, pp. 16–26, Mar. 2013, doi: 10.1016/j.tifs.2012.11.001.
[20] A. D. Workneh and M. Gmira, “Scheduling Algorithms: Challenges towards Smart Manufacturing,” *International Journal of Electrical and Computer Engineering Systems*, vol. 13, no. 7, pp. 587–600, 2022, doi: 10.32985/ijeces.13.7.11.
[21] A. Belhadi, V. Mani, S. S. Kamble, S. A. R. Khan, and S. Verma, “Artificial intelligence-driven innovation for enhancing supply chain resilience and performance under the effect of supply chain dynamism: an empirical investigation,” *Ann Oper Res*, vol. 333, no. 2–3, pp. 627–652, 2024, doi: 10.1007/s10479-021-03956-x.
[22] G. Mazzuto, S. Antomarioni, F. E. Ciarapica, and M. Bevilacqua, “Health Indicator for Predictive Maintenance Based on Fuzzy Cognitive Maps, Grey Wolf, and K-Nearest Neighbors Algorithms,” *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/8832011.
[23] T. Y. Melesse, M. Bollo, V. Di Pasquale, and S. Riemma, “Digital Twin for Inventory Planning of Fresh Produce,” in *IFAC-PapersOnLine*, Elsevier B.V., 2022, pp. 2743–2748. doi: 10.1016/j.ifacol.2022.10.134.
[24] B. E. Vignali G., “A tube-in-tube food pasteurizer modelling for a digital twin application.,” *Proceedings of the 6th International Food Operations and Processing Simulation Workshop (FoodOPS 2020)*, 2020.
[25] M. Resman, J. Protner, M. Simic, and N. Herakovic, “A five-step approach to planning data-driven digital twins for discrete manufacturing systems,” *Applied Sciences (Switzerland)*, vol. 11, no. 8, 2021, doi: 10.3390/app11083639.
[26] S. Croci, F. E. Ciarapica, G. Mazzuto, M. Bevilacqua, M. Ortenzi, and G. Osler, “A digital twin modeling for the production line optimized management in the soft and deformable food sector,” *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*, pp. 1–8, doi: 10.1109/ICE/ITMC61926.2024.10794247.
[27] A. Bueno, M. Godinho Filho, and A. G. Frank, “Smart production planning and control in the Industry 4.0 context: A systematic literature review,” *Comput Ind Eng*, vol. 149, no. August, p. 106774, 2020, doi: 10.1016/j.cie.2020.106774.
[28] “Bottani, E., Cammardella, A., Murino, T., & Vespoli, S. (2017). From the cyber-physical system to the digital twin: the process development for behaviour modelling of a cyber guided vehicle in M2M logic. XXII Summer School Francesco TurcoIndustrial System,” p. 2017, 2017.
[29] G. Baruffaldi, R. Accorsi, and R. Manzini, “Warehouse management system customization and information availability in 3pl companies: A decision-support tool,” *Industrial Management and Data Systems*, vol. 119, no. 2, pp. 251–273, 2019, doi: 10.1108/IMDS-01-2018-0033.