BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 72(3), 2024, Article number: e149816
DOI: 10.24425/bpasts.2024.149816

CONTROL, INFORMATICS AND ROBOTICS

Time parameters verification of a numerical simulator of an automated store warehouse

Sebastian RZYDZIK [1]* and **Piotr KROCZEK** [2]

[1] Faculty of Mechanical Engineering, Silesian University of Technology, ul. Konarskiego 18A, 44-100 Gliwice, Poland
[2] HemiTech sp. z o.o., ul. Piastowska 2, 44-122 Gliwice, Poland

Abstract. The intelligent automated store warehouse (iZMS) research and development project was created to meet the expectations of a modern automatic store. The project concerns the development of the concept and pilot implementation of an automated store warehouse adapted to the autonomous and automatic sales of goods selected by retail chains. One of the aims of the iZMS project is to develop a scalable solution that allows for the simple adaptation of the iZMS to the needs of a potential customer, considering their requirements in terms of the quantity and variety of assortment offered within the iZMS. An important requirement in the use of the iZMS system is minimizing the customer waiting time for purchased products. This problem is related, among others, to the placement of products on the shelves of racks and will be solved in the optimizing process. Running optimization tasks requires a simulator that will mimic the features of a physical device faster than in real time to generate many proposals of the allocation of goods on storage racks in the shortest possible time and choose the best one, guaranteeing the shortest picking time of a representative basket of goods. A numerical simulator was developed to model the physical structures of food storage equipment and then simulate the sales process. Among the results obtained, the most important are the time parameters of individual operations, which will be used to optimize the placement of goods on storage racks. After analyzing the needs resulting from the usage of the iZMS system, we decided to develop a dynamic, deterministic simulator with discrete objects and perform the simulation with a controlled time increment and, in some cases, utilize elements of event-driven simulation, in which the flow of goods is simulated with first-in, first-out (FIFO) queues. Finally, verification of the numerical simulator with a physical model confirmed that it could be employed in optimization processes.

Keywords: automated store warehouse; vending machine; numerical simulator; digital twin.

1. INTRODUCTION

In 2017, it was predicted that the value of the entire robot market would reach $87 billion by 2025, more than half of which will be in the retail sector [1]. Currently, estimates of the robot market suggest that it will exceed $200 billion in 2030, with more than 40% relating to service robots [2]. The epidemiological threat of COVID-19 helped to look positively at the development of robots in the e-commerce and retail industries. The COVID-19 pandemic facilitated a change in the perspective of potential users/customers regarding the use of service robots [3] and turned out to be a stimulus that increased interest in the implementation of robotic solutions [4, 5].

The widest use of robotic systems occurs in distribution centers and robotic warehouses [6–10]. Amazon [11, 12], Walmart [13, 14], and the British brand Ocado [15] are leading in the adaptation and implementation of robotic solutions in their distribution centers. These solutions are dedicated to large-scale chains of retailers using groups of mobile robots that fulfill orders for a large group of customers simultaneously. At the same time, retail chains are constantly looking for solutions that bridge the existing shopping experience with modern technologies that meet the expectations of the retail industry. One of the first approaches to this topic is small self-service stores such as Auchan Minute [16], or extended versions of vending machines such as Carrefour Express [17] and Lewiatan Go [18]. There are also fully robotic solutions available on the market such as automated pharmacies [19, 20], parcel machines [21], or robots that distribute DVDs [22]. Existing systems, despite the high technological advancement, are not comprehensive solutions combining a large capacity and a wide range of supported products, such as distribution centers, with small dimensions and high flexibility in the configuration of the internal structure, which allows them to be utilized as small independent stores operating 24/7. There is still a market gap for a system dedicated to medium and small chain stores that, in a fully automatic way, facilitates selling a wide range of food products and flexibly adjusting the size of the system depending on the target location.

The intelligent automated store warehouse (iZMS) research and development project was created to meet the expectations of a modern automatic store. The iZMS is a project implemented by a consortium of HemiTech Sp. z o.o., MGL Sp. z o.o., and the Silesian University of Technology, financed by the National Centre for Research and Development. The project concerns the development of the concept and pilot implementation of an automated store warehouse adapted to the autonomous and automatic sales of goods selected by retail chains. The iZMS concept is based on the need to solve significant problems in the retail industry related to store staff shortages, restrictions

*e-mail: sebastian.rzydzik@polsl.pl

Manuscript submitted 2023-04-27, revised 2024-03-11, initially accepted for publication 2023-03-12, published in May 2024.

on opening times (trade bans on Sundays and holidays or night sales), or the need to isolate workers from customers due to an epidemiological threat. Due to its construction that allows for the simultaneous storage and automatic release of products, it can be expected that the iZMS system will significantly improve sales efficiency by reducing the need for staff, decreasing the energy needed to store goods, and lowering supply chain costs. One of the aims of the iZMS project is to develop a scalable solution that allows for the simple adaptation of the iZMS to the needs of a potential customer, considering their requirements for the quantity and variety of assortment served and offered within the iZMS. The scaling of the iZMS was achieved by using a modular structure, combining specialized elements – iZMS modules – within a cluster. Modules are the basic structural element of the entire system, performing tasks related to the storage and transport of groups of products that have certain common features (e.g., size and type of packaging or storage conditions).

An important parameter in the use of the iZMS system is minimizing the customer waiting time for purchased products. This problem is related, among others, to the placement of products on the shelves and will be solved in the process of optimizing the structure of the location of products in the warehouse, depending on the customers' shopping preferences. Running optimization tasks requires a simulator [23, 24] that will mimic the features of a physical device faster than real time [25, 26] to generate many proposals for the allocation of goods on storage racks [27–29] in the shortest possible time and choose the best one, guaranteeing the shortest picking time of a representative basket of goods [30]. According to the assumptions, the warehouse will be shut down for the duration of the replenishment/replacement of products, which will also involve checking the technical condition of the systems, among other things. Therefore, a numerical simulator was developed to model the physical structures of food storage equipment and then simulate the sales process. Among the results obtained, the most important are the time parameters of individual operations, which will be utilized to optimize the placement of products on storage racks. The planned optimization algorithm will run the sales simulation many times for different product placement configurations, which requires short simulation times from the simulator. An example is optimization using a genetic algorithm, which necessitates searching the state space for many chromosomes. Optimization is time-limited because the distribution of goods on the shelves will be the basis for generating an order to the supplier of goods, e.g., once a week. Therefore, the minimum time needed to perform the sales is important, because then there will be more potential optimal solutions. For this purpose, the following criteria that the simulator must meet have been defined:

1. The minimum number of different types of goods for which it will be possible to create a simulation model ≥ 1000. This limitation is the result of an analysis of the diversity of goods expected by future recipients of the iZMS system.
2. The maximum calculation time using the simulation model for a single sales operation < 50 ms (with at least 1000 different types of goods). This limitation results from the estimation of the computing power available in the designed computer control system.
3. The average relative error of the model in the context of the time of product release operations during test transactions $\leq 10\%$. In discussions with the ordering party, it was found that an error at this level is acceptable for predicting warehouse performance for the sales department.

This goal will be achieved based on the comparison of the simulator results and data related to the time of performing individual operations obtained from the experiment using a physical system.

2. SIMULATOR DESCRIPTION

2.1. Types of simulators

Each simulator is a simplification of the phenomenon and, under certain assumptions, reflects the most prominent features that result from its purpose and omits others [31]. For known analytical solutions, one can create a mathematical equation or a system of equations and solve it. In the case of simulating logistic processes, we must model system components [32], that is, units in the system, the relations between them, and events that change the state of the system.

Diverse types of simulations were discussed [33–35]. Discrete simulations have entities that take many states finitely, but continuous ones allow their entities to include infinite values. Dynamic simulations have time as a variable, while static ones do not. Deterministic simulations apply the rules in the same way every time, but stochastic ones have a random component in employing some rules.

If there is no guarantee that events will occur at regular time intervals and we do not know what value of the time step to take (too small will increase the duration of the simulation, too high raises the risk of accumulating supported events, including mutually exclusive ones), we can use event-driven simulation [35–38]. This approach utilizes a list of events that occur at various times and are handled in ascending order in time [39,40]. A feature of this type of simulation is the time spent "jumping" between events.

2.2. Warehouse object model

After analyzing the needs resulting from the use of the iZMS system, we decided to develop a dynamic, deterministic simulator with discrete objects, perform the simulation with a controlled time increment, and in some cases use elements of event-driven simulation. In time-controlled simulations, there is a variable that registers the current time, which is incremented in fixed steps. In the loop, after each increment of the time quantum, events that may happen are checked and handled, e.g., the possibility of picking up the product or putting it in the required place by the robot. Speeding up the simulation can be achieved by predicting the moment of occurrence of the next event and performing a time shift [41]. In the developed simulator, this was achieved by analyzing the time first-in, first-out (FIFO) queues assigned to selected system components and shortening them to the events that would occur the fastest.

It was assumed that the criterion for stopping the simulation would be the completion of all products ordered by the customer in the basket. Products that are not in stock are noted in the shopping list as unavailable.

In practical implementation, the numerical simulator is a set of program classes that represent:
- The physical structure of the automated warehouse.
- The operating logic of the iZMS cluster and the flow of products from storage to the goods issuing zone (GIZ).

Figure 1 shows an exemplary diagram of the physical structure of the warehouse and its object representation. As a result of the analysis of the physical layout, several classes of objects were distinguished for which their object representations were created in the MATLAB program. Using the example structure of the iZMS cluster, classes can be distinguished among the objects:
- The Cell class represents the container where the goods (products) are.
- The Cells class represents storage racks and shelves that contain bins.
- The Product class represents goods for sale.
- The Products class represents a list of purchased goods.
- The Basket class represents a shopping basket for goods.
- The Baskets class represents a set of shopping baskets for goods.
- The Robot class represents a robot that moves goods (products) or baskets of goods.
- The Robots class represents a set of all robots in the iZMS cluster.
- The ReloadingPoint class represents a place where robots can transfer goods or baskets of goods to each other.
- The ReloadingPoints class represents a collection of reloading points.
- The Node class represents a single point on the goods flow path: a cell, a robot, a reloading point, or a shopping basket.
- The Nodes class represents the product flow path, that is, from the shelf to the shopping basket.
- The Stage class represents a timed FIFO queue. This can be of the StageDist type, where the time is calculated based on speed and distance, or the StageTime type, where the time is entered directly (equivalent to the delay).
- The Stages class represents a set of timed FIFO queues assigned to a robot or reloading point.
- The Tools class contains various methods needed to run the simulator.
- The Simulator class represents an algorithm of relations and events between objects.

2.3. Graph of the flow of goods

The basic operation of the simulator focuses on mapping the path of the goods flow from the cell to the issuing zone. An example graph of the flow of goods for a fixed physical structure of the warehouse is shown in Fig. 2. The number before the object type is its identifier, e.g., 1:Cells, 2:Cells, and 3:Cells: these are physically different storage racks (see Fig. 1). The Node and Nodes classes were developed for modeling flow paths, a container for a set of objects. The nodes of the graph correspond to the objects involved in the transport of goods, and the branches are the relationships between these objects. The basic relationships are picking up or putting away an item by the manipulator from/to the cell, basket, or ReloadingPoint. Depending on the location in the warehouse, each item is assigned one transport path. Transport paths are combined and thus identify nodes that are used repeatedly, such as a ReloadingPoint or robot.

The developed numerical simulator facilitates defining the following combinations of goods flow path nodes (Fig. 3):
- *The beginning of the path* – a node representing a container with goods in cells.
- *Path* – any combination of allowed connections of selected objects: robots, reloading points, and baskets.
- *End of the path* – always the shopping basket in the goods issuing zone.

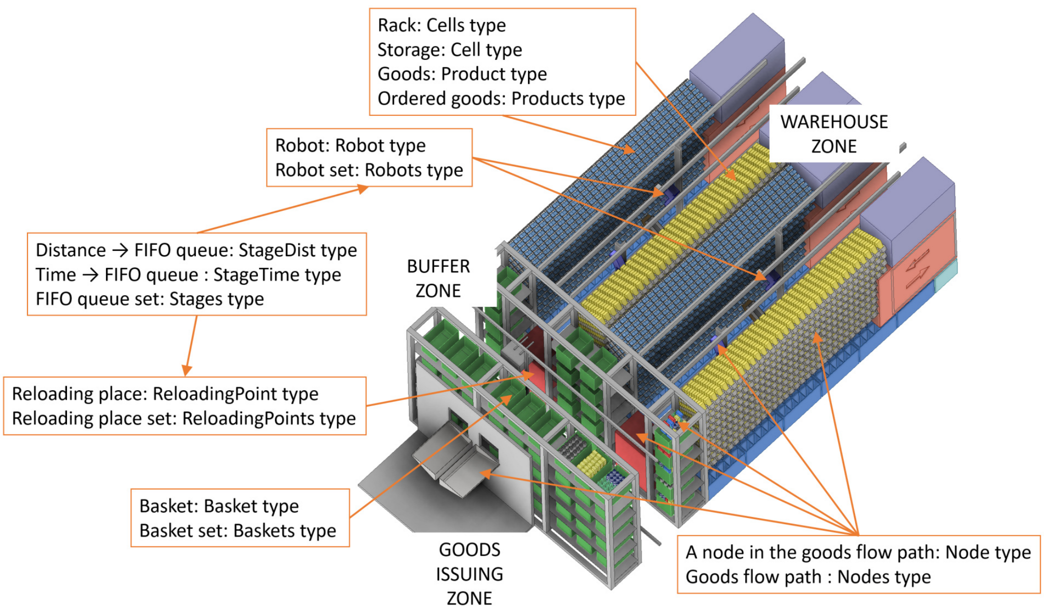


Fig. 1. Object representation of an exemplary iZMS cluster structure

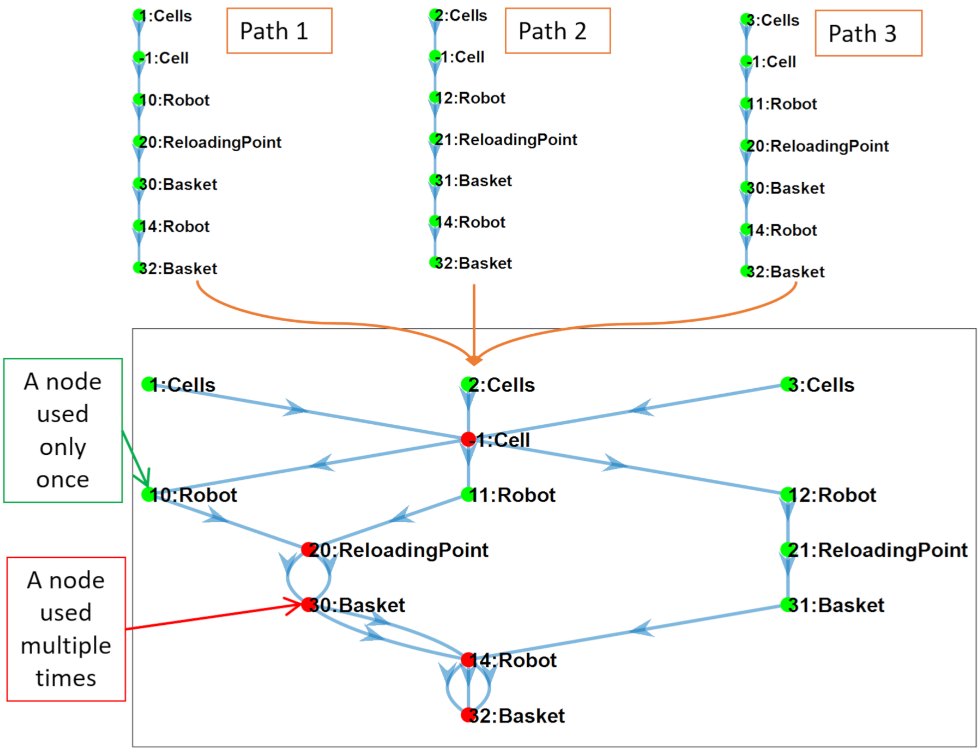


Fig. 2. An example graph of the flow of goods for a fixed physical structure of the warehouse

The set of dependencies between object classes presented in Fig. 3 makes it possible to define goods flow paths for various, even overly complex, warehouse structures. These structures can include multiple racks with variously arranged shelves. Products can be picked up and transferred by multiple robots and end up in a shopping basket.

The simulator only considers many-to-one conflicts, e.g., many robots want to put an item into the basket at the same time. Such a conflict is resolved according to the "first come, first served" rule. For this purpose, the simulator analyzes the inputs of FIFO queues at each step. If the input is already occupied by the goods ID, other robots must wait until the input is released.

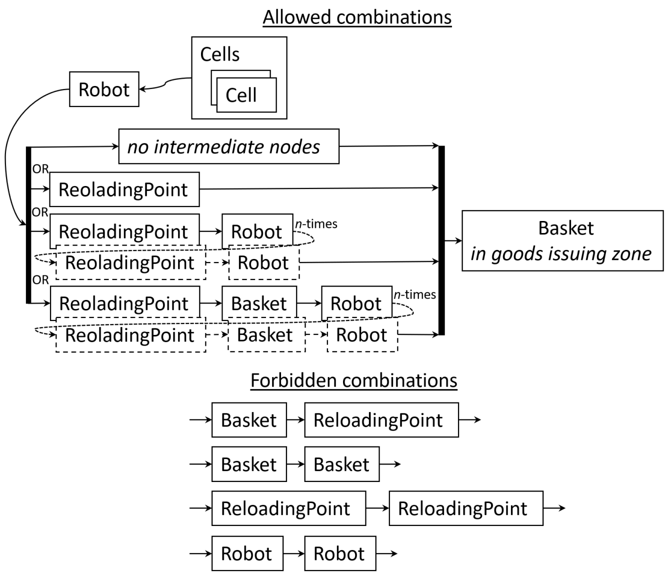


Fig. 3. Rules for creating a goods flow graph

It was decided to map the flow of goods by way of a time FIFO queue, due to the purpose of building this simulator, which is among other things designed to optimize the timing for completing the required list of products. The transport time is calculated for the known values of the path on which the goods are transported and the speed of this movement. The following classes were created for modeling FIFO queues: Stage and Stages, a container for these objects, and inherited classes StageDist (the class facilitates the calculation of transport time based on speed and distance parameters) and StageTime (the class allows for simulating the transport time of products by explicitly defining this time).

The StageDist object facilitates defining the parameters of propulsion systems by providing a speed model in the form of a trapezoid shape (Fig. 4). The parameters of the model include the maximum allowable speed $v_{\max}$, the maximum allowable acceleration $a_{+\max}$, deceleration $a_{-\max}$, and the displacement distance s (rotation or translation are available). The numerical model has a function to minimize the time t needed to perform the displacement s, considering the constraints. This function implements the following strategies: "accelerate to maximum speed with maximum allowable acceleration" and "stop drive with maximum deceleration".

The StageTime object is represented by a FIFO queue and has parameters: time increment dT and set displacement time T_{Lim}. During the simulation, an active object of this class moves an element in the FIFO queue with step dT until the specified timeout T_{Lim} is reached.

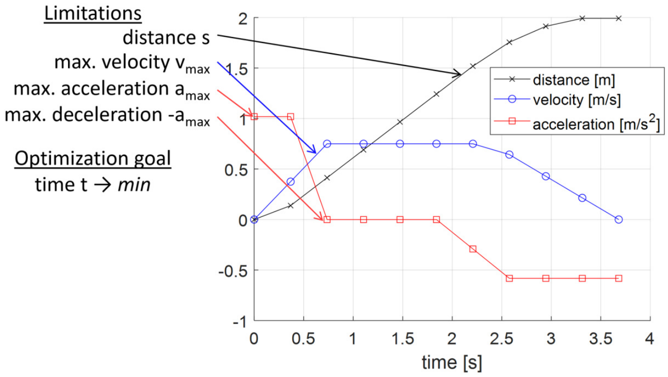


Fig. 4. An example of modeled drive system parameters

2.4. Simulator operation

Figure 5 shows the diagram of the simulator operation. During initialization, simulation parameters are entered into the simulator (see Table 3), and the mechanical structure of the warehouse (number of racks and shelves, distances between them, etc.), object parameters (e.g., drive characteristics), goods flow paths, and a list of necessary goods that the simulator must put in the basket. The main loop contains a condition that checks the completeness of the basket with goods. Depending on availability, items are marked as existing or out of stock. The goods in the warehouse are transported to the basket according to the path assigned to them, the choice of which depends on the location of the goods on the given rack. Each rack should be assigned a

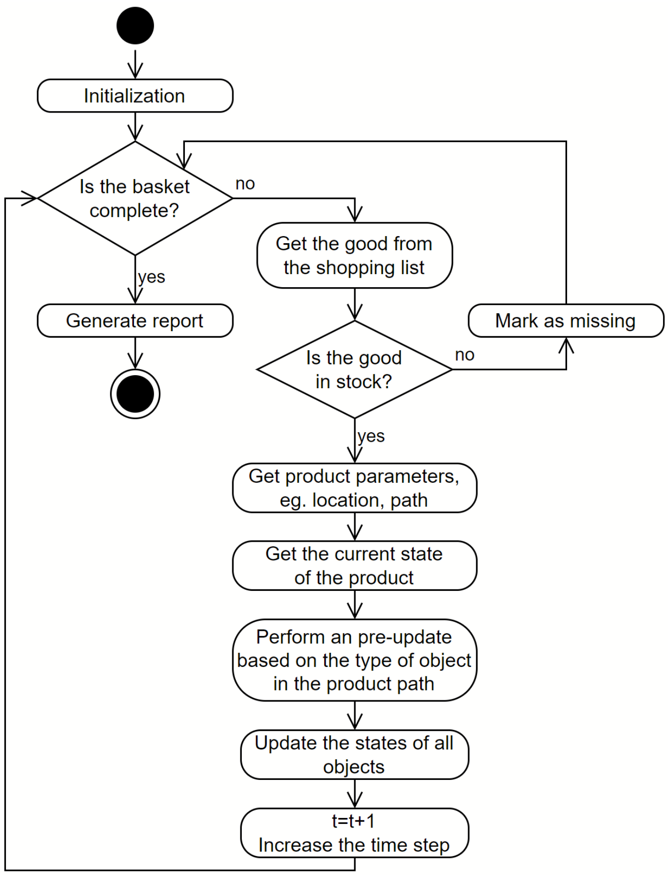


Fig. 5. Simulator activity diagram

robot that will transport the goods to the next nodes on this path, up to the basket made available to the customer. Optionally, a simple rule for the order of picking goods can be enabled in the simulation, according to the strategy "an item closest to the basket is transported first" or vice versa "an item that is furthest from the basket is transported first". It should be noted that research on optimizing the distribution of goods in the warehouse and the type of their picking strategy will be conducted in the next stage of the project. The simulator then prepares a state update by making changes to all active objects, considering their relationships, including conflicts. If the robot transporting the item reaches the ReloadingPoint, which is currently busy servicing another robot, it must wait in the queue. If two robots with goods reach the ReloadingPoint at the same time, the conflict is resolved randomly. Conflicts regarding similar situations, such as placing or taking goods to/from the basket by the robot, are resolved in the same way.

In the next step, the state of all objects is updated and then the simulation step is increased. During the simulation, various information is saved, such as a prediction of the real time of picking all goods and each one separately, the filling level of the shopping basket, and simulation time and efficiency.

Due to the protection of intellectual property, detailed algorithms of interactions between objects cannot be disclosed in any publication.

3. PHYSICAL WAREHOUSE PARAMETERS

The developed simulator is intended to reproduce the features of a physical sales warehouse. For this purpose, the simulator has certain characteristic numerical values with which it can be tuned so that the time of picking goods is comparable in both systems – physical and numerical.

Research using a physical system consisted of completing a set of goods placed in cells that were randomly placed on the shelves. Completion of the list of ordered goods was conducted with a properly programmed control system and a physical manipulator. The completed set of goods was placed in the shopping basket in the issuing zone. During the process of completing the goods, the control system recorded data, including data related to the time required to perform individual operations.

3.1. View of the test stand

Figure 6 shows the racks, shelves, and cells as well as the location and orientation of the global coordinate system, known as the GCS.

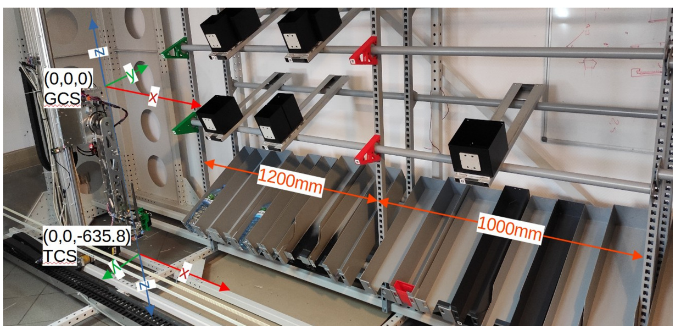


Fig. 6. An example of modeled drive system parameters

The cells and goods issuing zone were described with identifiers. The reference cell coordinates were read by the control system. The measurement results are presented in Fig. 7. The reference position of the issuing zone (id: GIZ) was set as the coordinates in the GCS system with the following values: $x = 2660$ mm, $y = -450$ mm, $z = 260$ mm.

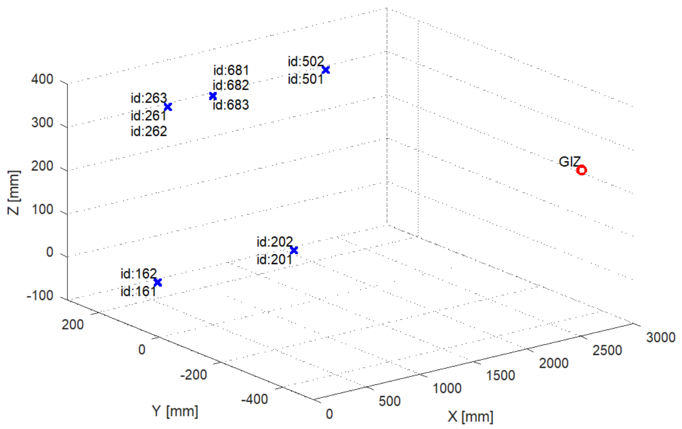


Fig. 7. Location of containers and goods issuing zone (GIZ) in three-dimensional space in the global GCS coordinate system

3.2. Selection of numerical simulator parameters

The data obtained from the control system were in the form of cyclically repeated records, examples of which are shown in Table 1. The following values may appear in the "Task type" column:

- Arrival at cell – Arrival at the pick-up position.
- Identification – Identification of the cell with the item.
- Grab – The sequence of grabbing the item from the cell.
- Arrival at the issuing zone – Arrival at the position of the issuing zone.
- Placement – The sequence of putting down the goods in the issuing zone.

Table 1
An example set of measurement time for a goods release cycle

Timestamp	Task type	Time duration, s
1641816618	Arrival at cell	0.860
1641816619	Identification	2.008
1641816623	Grab	2.877
1641816623	Placement	0.089
1641816624	Arrival at the issuing zone	1.018
1641816630	Placement	6.124

The cycle of completing a single item consisted of sequentially performed operations:

Go to the cell → identification of the item (product) → capture the item → go to the shopping basket → put the item into the shopping basket (the goods issuing zone).

Only two of these operations, related to the arrival of the manipulator in the cell or the goods issuing zone, took place with variable time. Other operations were performed in constant time within the limits of statistical repeatability.

In the next stage, a statistical analysis of constant time parameters measured in the physical system related to the following operations: identification of the item, grabbing it, and placing it in the shopping basket. The analysis showed a high repeatability of these operations for each of the 12 goods issued (Table 2). Based on this analysis, a time step was defined in the simulator with a constant value of 11.024 s, which was added to each issued item to obtain simulated goods picking time consistent with real time.

Table 2
Standard deviation analysis of constant time parameters

Task	Average task time duration, s	Standard deviation
Identification	1.999	0.022
Grab	2.841	0.024
Placement	6.184	0.022
Sum	11.024	

The measurement data and information about the parameters of the physical system made it possible to create and configure a numerical simulator. The parameters of the simulator with their description are presented in Table 3.

Table 3
Selected parameters of the numeric stimulator

Parameter	Description
SimPar.dT = 0.02	s, time increment (accuracy with which the simulation time is measured)
SimPar.dS = 0.01	m, path increment (accuracy with which the manipulator displacement increment is determined)
SimPar.vMax = 3.75	m/s, maximum drive speed
imPar.aMax = 3.75	m/s^2, maximum acceleration of the drives
SimPar.posCorr1a = 1.31	correction factor "a" in linear regression for the time to reach the cell
SimPar.posCorr1b = −0.43	correction factor "b" in linear regression for the time to reach the cell
SimPar.posCorr2a = 1.16	correction factor "a" in linear regression for the time to reach the issuing zone
SimPar.posCorr2b = −0.09	correction factor "b" in linear regression for the time to reach the issuing zone

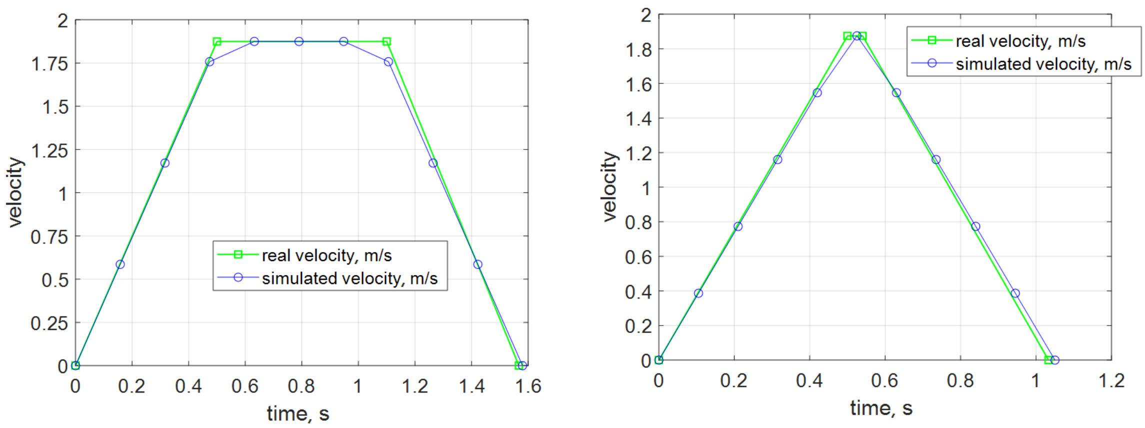
3.3. Drive parameters

The movements performed by the physical manipulator were conducted with the following limitations: maximum speed: 1.875 m/s and maximum acceleration/deceleration: 3.750 m/s^2. As a result of the manipulator displacement limitations adopted in this way, they were implemented in accordance with the given speed variability characteristics, which took the shape of a trapezoid (the so-called "ramps"). With the basic configuration of the simulator, the next step was to check the accuracy of obtaining the time parameters of the drives. For this purpose, the speed characteristics of the drive were compared with its simulated equivalent. The example results of these comparisons (green lines – real velocity and blue lines – simulated velocity) are shown in Fig. 8. The results obtained were considered exceptionally good, as the time-matching error did not exceed 3% for each drive.

3.4. Arrangement of goods on shelves

The next step was to define the geometric representation of the cell rack and the issuing zone. Figure 7 shows the spatial arrangement of the characteristic points assigned to the cells and the issuing zone according to the orientation of the global coordinate system.

Data on the location of characteristic points assigned to cells and the issuing zone in a three-dimensional space made it possible to determine the distances covered by the manipulator during the transport of the goods and to compare them with the measured times during the transport from the cell and to the issuing zone. The results are shown in Table 4. According to the parameters adopted by the device, the manipulator covered the same distances at various times, with arrival at the issuing zone always taking longer.



Limitations: s=2.0 m, v_{max}=1.875 m/s, a_{max}=3.750 m/s^2
Real time: t_{max}=1.567 s, Simulated time: t_{max}=1.580 s
Absolute relative time error: $|e_t|$=0.83%

Limitations: s=1.0 m, v_{max}=1.875 m/s, a_{max}=3.750 m/s^2
Real time: t_{max}=1.033 s, Simulated time: t_{max}=1.060 s
Absolute relative time error: $|e_t|$=2.61%

Fig. 8. Comparison of selected setpoints of physical and simulated drives

Table 4
List of the distances and times of displacement measured and simulated by the manipulator

Goods id	Distance from the measuring system, mm		Measured time, s		Simulated time, s	
	Arrival at the cell	Arrival at the issuing zone	Arrival at the cell	Arrival at the issuing zone	Arrival at the cell	Arrival at the issuing zone
161	2243.3	2243.3	0.777	1.743	1.56	1.56
162	2244.1	2244.1	1.684	1.746	1.56	1.56
501	852.5	852.5	0.860	1.018	0.98	0.98
502	852.5	852.5	0.963	1.063	0.98	0.98
681	1736.0	1736.0	1.361	1.481	1.38	1.38
261	2131.0	2131.0	1.559	1.675	1.54	1.54
262	2130.8	2130.8	1.614	1.680	1.54	1.54
201	1114.0	1114.0	0.940	1.198	1.10	1.10
202	1113.5	1113.5	0.920	1.190	1.10	1.10
263	2130.7	2130.7	1.544	1.707	1.54	1.54
682	1736.1	1736.1	1.326	1.523	1.38	1.38
683	1735.9	1735.9	1.316	1.490	1.38	1.38

4. VERIFICATION RESULTS

4.1. Initial simulation results

Considering the data and parameters described in the previous points in the simulator facilitated the simulation of the picking of 12 goods (the number of goods was determined arbitrarily by the design team).

The summary of the results obtained for variable time parameters is shown in Table 4. For constant distances covered by the manipulator, the simulated time of arrivals to the cell and the issuing zone is the same, which is contradictory to the times measured on the physical object. To modify the simulated arrival times, a linear regression analysis was performed. The results of these calculations for the time of arrival to the cell are presented in Fig. 9 (left) and in the issuing zone in Fig. 9 (right). Due to the significant difference in the time of arrival to the cell of the first item on the list (item id = 161), in the linear regression analysis performed, this measurement was rejected as too much of an outlier (an explanation of this problem is included later in the article).

4.2. Simulation results

The summary of the results obtained for variable time parameters, considering the correction, is shown in Table 5. As can be seen, in the case of the first item release cycle (id = 161),

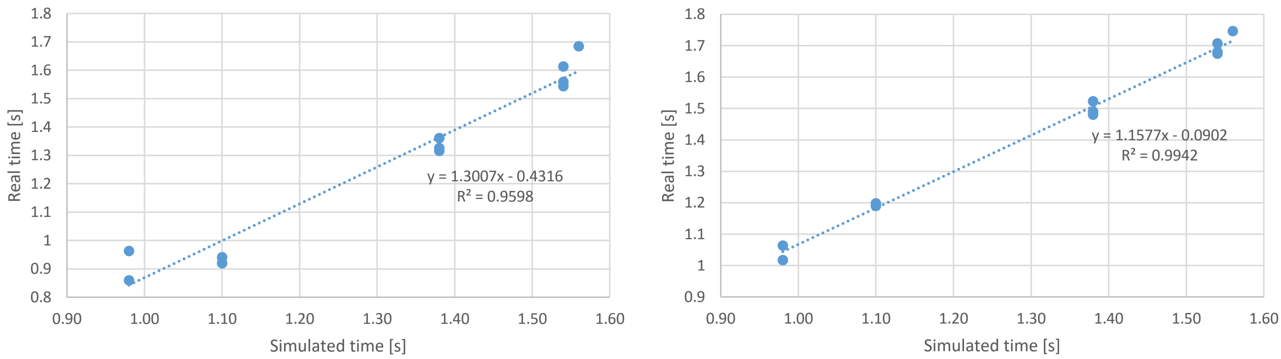


Fig. 9. The result of the regression analysis for the time of arrival to the cell (left) and for the arrival time to the issuing zone (right)

Table 5
List of variable time parameters

Goods id	Task (real time, s)		Task (simulated time, s)		Relative error, %		Average relative error, %
	Arrival at the cell	Arrival at the issuing zone	Arrival at the cell	Arrival at the issuing zone	Arrival at the cell	Arrival at the issuing zone	
161	0.777	1.743	1.60	1.72	–105.87	1.32	–52.28
162	1.684	1.746	1.60	1.72	5.01	1.51	3.26
501	0.860	1.018	0.87	1.08	–1.20	–6.13	–3.66
502	0.963	1.063	0.87	1.07	9.67	–0.62	4.52
681	1.361	1.481	1.36	1.51	0.10	–1.99	–0.94
261	1.559	1.675	1.57	1.69	–0.72	–0.91	–0.82
262	1.614	1.680	1.57	1.69	2.71	–0.60	1.06
201	0.940	1.198	1.00	1.18	–6.33	1.52	–2.40
202	0.920	1.190	1.00	1.18	–8.75	0.83	–3.96
263	1.544	1.707	1.57	1.69	–1.68	1.01	–0.34
682	1.326	1.523	1.36	1.51	–2.58	0.87	–0.86
683	1.316	1.490	1.36	1.51	–3.35	–1.33	–2.34
				Mean relative errors, %	–9.42	–0.38	–4.90
				Total relative errors, %	–112.99	–4.52	–58.76
				Standard deviation	29.45	2.08	14.49
				Mean relative errors, %*)	–0.65	–0.53	–0.59
				Total relative errors, %*)	–7.12	–5.84	–6.48
				Standard deviation*)	4.87	2.11	2.54

*) without considering the release of the first item

an exceptionally large relative error appeared when the manipulator arrived in the cell with the good, which was greater than 105%. This was related to the initial position of the manipulator. By default, in the simulator, the manipulator starts from the coordinates of the issuing zone (default parking position). The initial position of the physical manipulator was different, and information about this position was not known. However, it was decided not to reject this measurement and to include it in the calculations as an excessive deviation resulting from the initial state. In the case of other goods, this error is not greater than 10%, and on average for the property, it is not larger than 4%.

Table 6
Summary of goods issuing times

Goods id	Real time, s	Simulated time, s	Relative error, %
161	13.54	14.40	–6.35
162	14.43	14.36	0.49
501	12.97	13.00	–0.23
502	13.04	13.00	0.31
681	13.91	13.94	–0.22
261	14.27	14.32	–0.35
262	14.31	14.32	–0.07
201	13.15	13.24	–0.68
202	13.14	13.24	–0.76
263	14.22	14.32	–0.70
682	13.82	13.94	–0.87
683	13.82	13.94	–0.87
Sum	164.62	166.02	
		Mean relative error, %	–0.86
		Total relative error, %	–10.3
		Standard deviation	1.71
		Mean relative error, %*)	–0.36
		Total relative errors, %*)	–3.95
		Standard deviation*)	0.45

*) without considering the release of the first item

Table 6 shows the time taken to issue each of the 12 goods, broken down into the time measured by the physical system and the time calculated in the simulator. Considering the situation described above, the error resulting from the discrepancy in the starting coordinates of the physical and simulated manipulators also affects the relative error of issuing a single item and completing a set of 12 goods. As can be assumed, the highest value of this error occurs for the first item with id = 161 and is greater than 6%. In the case of other goods, it is not greater than 1%. The total relative error is greater than 10%, but without considering the release of the first item is much lower and is less than 4%.

5. RESULTS OF THE PERFORMANCE TEST

The performance tests were conducted in two stages:

1. Determination of the maximum number of various goods for which the time of a single sales transaction is less than 50 ms, for the selection of 12 different goods. The intermediate goal was to determine the nature of changes in the number of various goods in random-access memory (RAM) occupancy and the increase in computation time.
2. Determination of the average time of a single sale transaction for a fixed number of various goods, specified in Stage I, for the selection of 12 goods.

Simulation tests were performed on a computer with a computing power of 57 MOps/s (in the floating-point math test) with 16 GB RAM, using the MS Windows 10 Pro operating system and in the MATLAB R2020b environment.

During tests in the simulator, the number of goods in the warehouse was increased by one, checking whether the condition < 50 ms for issuing one item was met. It was established that the limit on the number of different goods in the warehouse must be less than 1536. The simulation time was always < 600 ms, which gives < 50 ms for issuing one item. Memory usage was 172 kB, which gives 112 bytes per one program object. The simulated goods completion time was approximately 166.02 s (see Table 6) and was independent of the number of simulation runs and the number of various goods.

The size of the possible volume of goods created as software objects depends primarily on the amount of RAM of the computer on which the simulation is run. More than 1.5 million goods were defined in the tests, which were objects of the good-in-cell type. In successive simulations, no influence of the number of good-in-cell objects was observed on the simulation of the goods picking time. It was observed that the RAM occupancy and computation time depend linearly on the number of various goods represented by the objects.

6. CONCLUSIONS

Measurements of the time parameters performed on the laboratory version of the iZMS system allowed for verification of the time parameters of the simulator and design requirements:

1. The simulator should enable the handling of over 1000 different types of goods – the expected functionality by future recipients of the iZMS system.
2. The maximum simulation time of a single product sales operation should be less than 50 ms (with at least 1000 different types of goods) – the limitation results from the need to select the optimal combination from among many thousands of proposals for arranging goods on the warehouse shelves generated by the optimization algorithm (e.g., the genetic algorithm) and calculating the time for picking representative baskets.
3. The average relative error in predicting the picking time of an item should be less than 10%.

Considering the results obtained, it can be concluded that the parameters of the numerical simulator were correctly selected and that they can be used in the process of optimizing the arrangement of goods in the warehouse zone of the iZMS cluster.

The analysis of the conducted research, including both the results and the computational environment, determines the path of further proceedings. The MATLAB environment is perfect for rapid prototyping, but unfortunately, it is not computationally efficient. Switching to another programming language would speed up the simulation. Preliminary tests conducted on the alpha version of the simulator written in C/C++ allowed for a 10-fold acceleration and, at the same time, generated a larger space in which there can be more optimally good solutions to the task of arranging goods on warehouse shelves. Alternatively, we can assume that the space of good solutions to the optimization task is sufficient and consider using a processor with less computing power but greater energy efficiency.

Due to the purpose of building the simulator, which was primarily to represent the time parameters related to the movement of goods from cells to the basket at an elevated level, its memory requirements are exceedingly small. Therefore, there is no risk that the target computing computer will need to be equipped with a large amount of RAM.

It should be emphasized that the reproduction of the operation of a specific physical prototype of the warehouse and the maintenance of time regimes in this case is only laboratory confirmation. At the current stage of project implementation, a verified method of procedure that should be used in the target solution is important.

7. FURTHER WORK

The hardware (warehouse) and software (simulator) solutions presented in this article are their laboratory versions, intended for preliminary tests performed in the first stages of the project. The next stages concern the development of further versions and the development of a device that will be ready for implementation in the leading retail chain in Poland. Therefore, intensive work is currently underway, the foundation of which is the conclusions and experience resulting from testing the laboratory versions. Finally, a family of warehouses will be created for the automatic sale of goods with a distributed control system and integrated with a calculation module that allows for the optimization of the arrangement of goods on shelves to obtain the minimum sales time for the most frequently purchased goods.

At the time of drafting this paper, a recent version of the iZMS system was created, which allowed us to revise the adopted assumptions regarding mechanical systems that were simplified by building prototypes, among others. The variety of modules was also reduced. Thus, the next version of the simulator will be slightly simplified, leading to faster calculations.

ACKNOWLEDGEMENTS

The research described in this paper was conducted within the “Intelligent Cell Cluster of the Automated Store Warehouse (iZMS)” project (POIR.01.01.01-00-0104/20). The project is co-financed by the European Union from the European Regional Development Fund under the Intelligent Development Program and is coordinated by the National Centre for Research and Development in Poland.

Part of the research was also conducted within the grant no. RJO15/SDW/006-19 financed by the Ministry of Education and Science of Poland.

REFERENCES

[1] M. Wolfgang, V. Lukic, A. Sander, J. Martin, and D. Küpper. “Gaining Robotics Advantage.” [Online]. Available: https://www.bcg.com/publications/2017/strategy-technology-digital-gaining-robotics-advantage. [Accessed: 11 Dec. 2023].

[2] R. Lässig, M. Lorenz, E. Sissimatos, I. Wicker, and T. Buchner, “Robotics Outlook 2030: How Intelligence and Mobility Will Shape the Future.” [Online]. Available: https://www.bcg.com/publications/2021/how-intelligence-and-mobility-will-shape-the-future-of-the-robotics-industry?linkId=123960002. [Accessed: 11 Dec. 2023].

[3] E. Merdin-Uygur and S. Ozturkcan, “Consumers and Service Robots: Power Relationships amid COVID-19 Pandemic,” *J. Retail. Consum. Serv.*, vol. 70, p. 103174, 2023, doi: 10.1016/j.jretconser.2022.103174.

[4] B. Graf. “The Service Robotics Market Is Opening up Completely New Opportunities.” [Online]. Available: https://ifr.org/post/the-service-robotics-market-is-opening-up-completely-new-opportunities. [Accessed: 11 Dec. 2023].

[5] M. Bill, C. Müller, W. Kraus, and S. Bieller, “World Robotics 2022.” [Online]. Available: https://ifr.org/downloads/press2018/2022_WR_extended_version.pdf. [Accessed: 11 Dec. 2023].

[6] E.A. Oyekanlu *et al.*, “A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications,” *IEEE Access*, vol. 8, pp. 202312–202353, 2020, doi: 10.1109/ACCESS.2020.3035729.

[7] BionicHIVE. “SQUID”. [Online]. Available: https://www.bionichive.com. [Accessed: 11 Dec. 2023].

[8] Via Robotics. “InVIA.” [Online]. Available: https://inviarobotics.com/our-system/invia-picker-robots. [Accessed: 11 Dec. 2023].

[9] Hikrobot. [Online]. Available: https://www.hikrobotics.com/en. [Accessed: 11 Dec. 2023].

[10] Exotec. “Skypod.” [Online]. Available: https://www.exotec.com/skypod-system. [Accessed: 11 Dec. 2023].

[11] Amazon. “10 Years of Amazon Robotics: How Robots Help Sort Packages, Move Product, and Improve Safety.” [Online]. Available: https://www.aboutamazon.com/news/operations/10-years-of-amazon-robotics-how-robots-help-sort-packages-move-product-and-improve-safety. [Accessed: 11 Dec. 2023].

[12] A. Yudiansyah, “Can the Mobile Robot Be a Future Order-Picking Solution?: A Case Study at Amazon Fulfillment Center”. *Adv. Transp. Logist. Res.*, vol. 3, p. 7, 2020.

[13] T. Stallbaumer. “How Walmart’s Alphabot Is Helping to Revolutionize Online Grocery Pickup and Delivery.” [Online]. Available: https://corporate.walmart.com/newsroom/2020/01/08/how-walmarts-alphabot-is-helping-to-revolutionize-online-grocery-pickup-and-delivery. [Accessed: 11 Dec. 2023].

[14] Alert Innovation. “Alphabot.” [Online]. Available: https://www.walmartasr.com/technologies/alphabot-asrs-system. [Accessed: 11 Dec. 2023].

[15] Ocado Group. [Online]. Available: https://www.ocadogroup.com/osp/our-technology. [Accessed: 11 Dec. 2023].

[16] Wiadomości Handlowe. “Kieszonkowy Koncept Sklepu Auchan Minute Coraz Bliżej Polski.” [Online]. Available: https://www.wiadomoscihandlowe.pl/artykul/kieszonkowy-koncept-sklepu-auchan-minute-coraz-blizej-polski. [Accessed: 11 Dec. 2023].

[17] Wiadomości Handlowe. “Sklepy Bezobsługowe to Cenna Lekcja Dla Sieci Handlowych.” [Online]. Available: https://www.wiadomoscihandlowe.pl/pomysl-na-sklep/sklepy-bezobslugowe-to-cenna-lekcja-dla-sieci-handlowych-eksperci-2408303. [Accessed: 11 Dec. 2023].

[18] Gazeta Krakowska. “Kraków. Lewiatan w Automacie. Sieć Uruchomiła w Krakowie Swój Pierwszy Całodobowy Sklep Bezobsługowy.” [Online]. Available: https://gazetakrakowska.pl/krakow-lewiatan-w-automacie-siec-uruchomila-w-krakowie-swoj-pierwszy-calodobowy-sklep-bezobslugowy-zdjecia-1102/ar/c3-14761614. [Accessed: 11 Dec. 2023].

[19] UnitDose One. [Online]. Available: https://unitdoseone.com. [Accessed: 11 Dec. 2023].

[20] Fablox. [Online]. Available: https://fablox.pl. [Accessed: 11 Dec. 2023].

[21] PickupHero. [Online]. Available: https://www.rrobotics.co/pl/product/pickuphero. [Accessed: 11 Dec. 2023].

[22] P. Rehkamp. “2016 Eureka! awards winner: Best Buy Co. Inc.” [Online]. Available: https://www.bizjournals.com/twincities/news/2016/06/17/2016-eureka-awards-winner-best-buy-co-inc.html. [Accessed: 11 Dec. 2023].

[23] J.P. Gagliardi, J. Renaud, and A. Ruiz, “A Simulation Model to Improve Warehouse Operations,” in *Proc. 2007 Winter Simulation Conference*, 2007, pp. 2012–2018, doi: 10.1145/1351542.1351899.

[24] O. Ganbold, K. Kundu, H. Li, and W. Zhang, “A Simulation-Based Optimization Method for Warehouse Worker Assignment,“ *Algorithms*, vol. 13, no. 12:326, 2016, doi: 10.3390/a13120326.

[25] X. Liu, J. Ospina, I. Zografopoulos, A. Russel and C. Konstantinou, “Faster Than Real-Time Simulation: Methods, Tools, and Applications,” in Proc. *9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, 2021, pp. 1–7, doi: 10.1145/3470481.3472703.

[26] X. Xu, Y. Li and R. Tang, “Simulation Optimization of Discrete Logistics Processes: A Case Study on Logistics of an E-Commerce Enterprise in Shanghai,” *Discrete Dyn. Nat. Soc.*, vol. 2019, p. 2493638, 2019 doi: 10.1155/2019/2493638.

[27] A. Ratkiewicz and K. Lewczuk, “Rack cell configuration problem: a mathematical model and effective combined heuristic,” *Bull. Acad. Sci. Tech. Sc.*, vol. 69, no. 1, p. e135836, 2021, doi: 10.24425/bpasts.2021.135836.

[28] Y. Kuo and H.-C. Jiang, “Inventory Classification with Limitations in the Number of Changeovers and Space for Inventory,” *Manag. Prod. Eng. Rev.*, vol. 14, no. 4, pp. 92–99, 2023, doi: 10.24425/mper.2023.147206.

[29] K. Czerniachowska, R. Wichniarek, and K. Żywicki, “Industry Expertise Heuristics for Dimensioning Shelf Space of Rack Storage Location in a Distribution Centre with Zone Picking,” *Manag. Prod. Eng. Rev.*, vol. 14, no. 1, pp. 43–60, 2023, doi: 10.24425/mper.2023.145365.

[30] T. Markowski and P. Bilski, “Optimization of Autonomous Agent Routes in Logistics Warehouse,” *Int. J. Electron. Telecommun.*, vol. 67, no. 4, pp. 559–564, 2021, doi: 10.24425/ijet.2021.137846.

[31] T. Altiok and B. Melamed. *Simulation Modeling and Analysis with Arena*, Academic Press, 2007.

[32] R. Ivanov, Y. Sherstennikov, V. Porokhnya, and T. Grynko, “Mathematical Model for Imitation of Management of the Enterprise’s Logistical System,” in Proc. *9th International Conference on Monitoring, Modeling & Management of Emergent Economy (M3E2 2021)*, 2021, vol. 107, p. 9, doi: 10.1051/SHSCONF/202110710004.

[33] L. Leemis, “Input Modeling Techniques for Discrete-Event Simulations,” in *Proc. 2001 Winter Simulation Conference*, USA, 2001, pp. 62–73, doi: 10.1109/WSC.2001.977247.

[34] N. Saad, *Modelling, Simulation, and Analysis of Supply Chain Systems Using Discrete-Event Simulation*. PhD thesis, University of Sheffield, 2003.

[35] P.K. Damarapurapu, D. Kotte, K. Basha, and R. Venkatesh, “Discrete Event Simulation of Production and Logistics Processes for the Industrial Environment Using R Programming,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, pp. 6760–6764, 2020, doi: 10.30534/ijeter/2020/238102020.

[36] D. Kuzmin, V. Baginova, and A. Ageikin, “Discrete Event Simulation Model of the Railway Station,” *Transp. Res. Procedia*, vol. 63, pp. 929–937, 2022, doi: 10.1016/j.trpro.2022.06.091.

[37] K. Kumar, “Design of Vending Machine through Implementation of Visual Automata Simulator and Finite State Machine,” *Int. J. Res. Circuits Dev. Syst.*, vol. 2, pp. 60–64, 2021.

[38] A. Ashrafian *et al.*, “Full-Scale Discrete Event Simulation of an Automated Modular Conveyor System for Warehouse Logistics,” in *Advances in Production Management Systems. Towards Smart Production Management Systems*, F. Ameri, K.E. Stecke, G. von Cieminski and D. Kiritsis, Eds., *IFIP Advances in Information and Communication Technology*; Springer International Publishing: Cham, 2019, vol. 567, pp. 35–42, doi: 10.1007/978-3-030-29996-5_4.

[39] C.A.R. Hoare, “Communicating Sequential Processes,” *Prentice-Hall International Series in Computer Science*, Reprinted, Prentice Hall: New York, 2000.

[40] C. Janius and S. Mir. *Using Discrete Event Simulation: Improving Efficiency and Eliminating Nonvalue Added Work*, Master thesis, Mälardalen University, 2016.

[41] A. Almech and E. Roanes-Lozano, “An Accelerated-Time Simulation of Queues at Ticket Offices at Railway Stations,” *Math. Probl. Eng.*, vol. 2021, p. 9313174, 2021, doi: 10.1155/2021/9313174.