Digital Twin of a Flexible Manufacturing System for Solution Preparation

Paulo Francisco Carreira Faria

paulo.faria@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

December 2021

Abstract

In the last decades, there has been a necessity for systems capable of handling market changes and various personalized customer needs, with near mass production efficiency, defined as mass customization. Industry 5.0 further exposed this need for robust flexible systems, as well as a necessity of manufacturing systems that work in close cooperation with workers, taking advantage of the problem-solving capabilities and knowledge of the manufacturing process. A solution for this necessity is to develop a flexible manufacturing system, capable of handling different customer requests and real-time decisions from operators. Which this thesis tackles by proposing a Digital Twin focused on the simulation of a Robot System for Solution Preparation, capable of making real-time scheduling decisions and forecasts by using a simulation model to test different resource configurations and customer requests, while allowing an operator to make changes in the processing time and order of some operations in real-time. A mixed event simulation model was utilized to do resource forecasts, where real-time decisions were performed by recurring to completely reactive scheduling with parallel tasks. Resource forecasts were utilized to know where the manufacturing system can be improved. It was shown that combining parallel tasking with parallel machines to key processes, utilizing heuristics that emphasize the shortest transportation time and increasing the robot's speed, best impact the performance of the system, reducing overall completion time by 82%, when comparing single tasking and single machines. The simulation model also has an animated visualization window, for a deeper understanding of the system.

Keywords: Digital Twin, Mass Customization, Industry 5.0, Flexible Manufacturing System, Simulation

1. Introduction

During the last decades, there have been remarkable leaps in the areas of information technologies and digitalization, that lead to the creation of autonomous, self-regulated systems.

These developments increased the output of the industry by a large margin, but the growing variety and rising costumer demand for individual or custom-made products at lower costs, calls for the design and operation of systems capable of handling this increasing variety in products. This need to deliver products and services that best meet individual customers' needs with near mass production efficiency, which is defined as mass customization is therefore necessary [1]. A way to deal with this issue is the use of Flexible Manufacturing Systems (FMS), in which, the main idea is, a system built a priori to deal with changes in the market demands, able to yield a wide range of products from a single base unit [2].

The need to increase production flexibility is part of the ideals of Industry 5.0, which recurs to data and AI as well as having closer cooperation between man and autonomous machines, taking advantage of the human problem-solving capabilities to increase the production flexibility [3].

This paper addresses the need for mass customization and flexible robust systems, by developing a Digital Twin of a flexible manufacturing system, capable of handling different requests and real-time changes from operators and clients, by making use of the new technological trends.

T. Coito et al. [4] first presented this manufacturing system as a case study, in the quality control laboratory of the pharmaceutical industry. The authors propose a platform that allows for the autonomous acquisition and management of personalized data in real-time for mass customization manufacturing environments, that supports the integration of dynamic Decision Support Systems.

The idea of a Digital Twin (DT) refers to a comprehensive physical and functional description of a component, product or system, which includes more or less all information which is considered useful, so that a digital entity of its own could be created, considered as a "twin", that allows the exchanging of information between the real system and its digital counterpart. The Digital Twin when coupled with data analytics allows for real-time monitoring, rapid analysis, and real-time decisions, allows stakeholders to quickly detect problems in physical systems, increase the accuracy of their results and more efficiently produce better products [5].

To make real-time scheduling decisions, the Digital Twin must be coupled with a scheduling algorithm. Scheduling in this context has the goal of assigning a set of jobs, each having a set of operations that need to be scheduled in machines with the goal of reducing the total time to process all the jobs (makespan) and increase machine utilization. As in the real world real, machines can breakdown, orders can be late, operators might be unavailable, new urgent orders might arrive, there can be variations in processing time, especially in flexible systems, making the scheduling plan obsolete very quickly. In completely reactive scheduling, no firm schedule is made in advance and all decisions are made in real-time, the decisions are made using a dispatching rule to select the next job with the highest priority from a set of available jobs waiting to be processed [6].

The Robot System for Solution Preparation is an industrial prototype, its purpose is to create liquid preparations of products in bottles, which is done by having an anthropomorphic robot with 8 different workstations (Ws), with unique functions such as mixing, labelling, stirring that are within the robot's range, with the robot being the resource responsible for the movement of bottles. The already existing system processes a single bottle or job at a time, it's expected to reduce the makespan by recurring to parallel task scheduling, which means, having multiple entities being processed at a time, combined with testing different resource allocations, robot parameters, and the right scheduling algorithm, to create a system capable of efficiently handling multiple bottles simultaneously and different costumer requests while allowing stakeholders to manipulate process and scheduling parameters, and also to make real-time decisions, creating the desired flexible environment with Industry 5.0 ideals. This thesis focuses on the simulation part of the DT, which is part of the environment comprised of Real System and Stakeholders and DT, (figure 1):

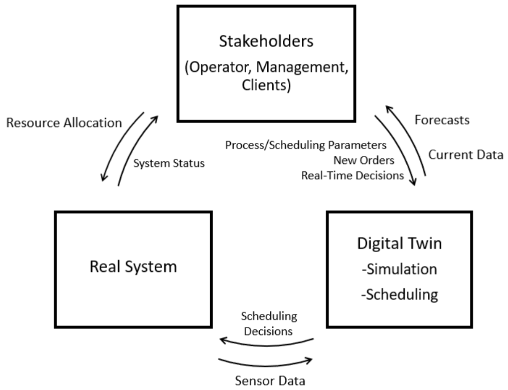

Figure 1: Diagram of the of Real System, Stakeholders and Digital Twin environment

The Digital Twin receives both historical and real-time data, associated with the process flow of the manufacturing system, that is, processing times, transportation times and decisions that relate to the scheduling process, and with it, makes scheduling decisions, based on a chosen algorithm. It works as a tool to help Stakeholders monitor the asset and do simulation runs to better understand the system and decide scheduling parameters, where resources should be allocated and to make real-time scheduling decisions with the constant flow of information from its real counterpart.

2. Discrete Event Systems Modelling

Computer simulation is a powerful tool used to analyze the performance of existing or newly designed systems by making use of mathematical or numerical techniques. This works by creating a model of the conceptual framework of the system, which then can be used for a wide variety of experiments with the system, and by analyzing the results, conclusions can be taken in order to help with the decision-making processes of the stakeholders [7].

2.1 Time Advance Mechanisms

One type of simulation is discrete event modelling, where changes in the system occur at discrete times, these changes then affect the system depending on the chosen Time Advance Mechanism (TAM). The two most common TAMs currently in use are the “next event” method, implemented in Discrete Event Simulations (DES) and the “time step” or “fixed increment” method implemented in Discrete Time Simulations (DTS).

Real-time is continuous but a digital computer would take an infinite amount of time to represent it, so in DTS a “Simulation Clock” is introduced in which the time is discretized in order to be finite, this works by having constant time increments of Δt (that can be a second, an hour and so on), effectively “skipping time” between increments, this way, continuous time can be simulated on digital computers [8].

Contrary to DTS, in Discrete Event Systems (DES), the state transitions are driven by the combination of asynchronous and concurrent events, this way, events are the cause of change in the system, so the concept of time is not the driving force of the simulation and is instead a variable dependent on the state transitions, so the simulation effectively skips time between events [9].

2.2 Related Work

T. Coito et al. [10] first presented the prototype as a case study, where the implementation of a middleware platform is discussed, with the goal of facilitating the information exchange between real assets and the Decision Support Systems for a Quality Control laboratory, where the prototype is included. The authors propose a platform that allows for the autonomous acquisition and management of personalized data in real-time mass customization manufacturing environments, that supports the integration of dynamic Decision Support Systems.

Yu et al. [11] Fused the concept of Digital Twin with job shop scheduling of a flexible job shop of a manufacturing system. The authors created a Scheduling cloud platform that takes input from the sensors from the physical workshop which fuels data to its respective Digital Twin and also originates a fault prediction and diagnosis curve, the Digital Twin then gives to the Scheduling cloud platform simulation data and energy consumption diagrams to help planning the process steps. Yu et al. found advantageous to have access to a lot of data from the whole manufacturing process, helping to monitor the whole life cycle of the products, reduce energy consumption and predict failures in the processes.

Wladimir Hofmann et al. [12] presents a Digital Twin with real-time decision support for port operations, to deal with the issue of truck congestion in arrival gates. The Digital Twin assists the dispatching operator in the decision-making process of releasing trucks whenever the port terminal is free, with the goal of reducing the probability of deadline violations due to low utilization of bottleneck resources. The Digital Twin receives both present and past information, using IoT, from the registered trucks, and by using a dispatching algorithm, which enables different dispatching policies, to be evaluated and presented to the dispatching operator, he can then discuss with the involved stakeholders the best combination of parameters.

Karagiannis et al. [13] addressed the issue of how hard automation solutions that increase productivity, end up not allowing industries to adapt to market changes and system malfunction, by utilizing an existing consumer goods industrial production line. The authors developed a DES model which offers the possibility to test all the probable occurrences in the assembly line, by the manipulation of parameters, in a risk-free virtual environment.

To the authors knowledge, there is not a lot of research topic that bring together the concept of Digital Twin working with a human counterpart in a flexible environment, the literature mostly present somewhat stochastic conditions, machine breakdowns, new orders, late arrivals and workers unavailable as proprieties that make the system dynamic, but rarely the inclusion of a human counterpart, capable of changing processing times and workflow, also, the flexibility of the manufacturing systems, tends to come from supply chain flexibility, not from the manufacturing process itself [14], [15].

3. Developed Digital Twin

The first step to make the necessary changes to adapt this prototype to the necessities presented earlier, is to know the workings of the already existing system, which will also help with the validation of the simulation model.

In this system, a bottle is considered as the entity that goes through the processes with time and workflow defined by each client request, with the workflow shown in figure 3, it first starts in an Entrance Storage (ES), then it's transported by the robot manipulator through each workstation to be processed, until it ends in the Final Storage (FS), both the ES and the FS use a rotating storing device, the workstations (Ws) and their positions relative to the robot are displayed bellow:

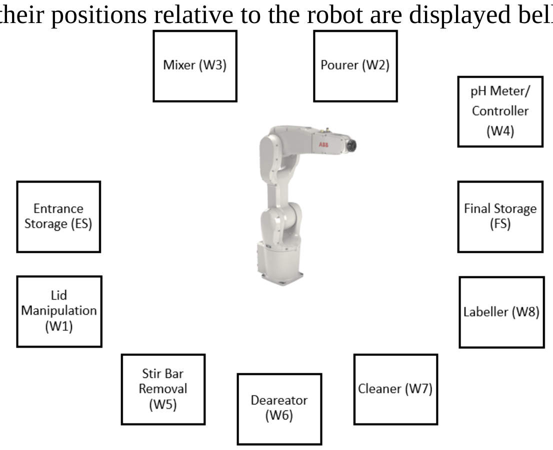

Figure 2: Position of Workstations and Robot of the prototype

Each movement of the robot when it is carrying a bottle is also considered as a process $M_{ii'}$, the first one M_{12} describing the movement from ES to workstation 1 (W1), M_{23} from W1 to W2 and so on, in total there are 13 movements, and each bottle has its own order of movement, and may repeat or skip some, depending on the client's request and real time decisions, these are shown in figure 3.

As its intended to create a model with parallel tasks, every time the robot finishes placing a bottle in a Ws, a chosen dispatching rule determines what bottle should the robot attend to next, and so, the robot moves without carrying a container when changing between tasks, this motion time is defined as pre movement time Mp_{pi}, from the robot's current position (p) to the workstation of the next task (i).

To better understand how the processes interact with each other, the Process Model and Notation (BPMN) was employed, a visual modeling language for defining enterprise process workflows [16]. Which can be consulted in figure 3.

The industrial prototype makes liquid solutions in bottles, which can be described as jobs $J_1, J_2, \dots, J_n$ to be scheduled in the machines W1 to W8 with the order represented in figure 3, where each job has a specific route through the machines depending on customer requests and real-time decisions.

There are real-time events and decisions, such as: number of iterations between W3 and W4 until the pH is correct either automatically or by the workers decision, real-time changes in the process time in W3 and W6, preparation orders that can arrive at any moment for processing, and the stochasticity of processes. These dynamic proprieties make the schedule building a difficult task, as it quickly gets obsolete, making rescheduling too frequent and ineffective, this makes a completely reactive scheduling an adequate option to apply in this case study.

To address this issue, traditional dispatching rules such as, the Shortest Processing Time (SPT), Longest Processing Time (LPT), and Least Work Remaining (LWR) were employed, which define the priority of jobs currently not being processed . The job j with highest priority according to the chosen dispatching rule, is the next one to be processed. This procedure occurs according to Figure 4, and the job's priority is set to zero if the next workstation in the workflow is full, or if the movement might stop the flow of tasks and result in a standstill for the system, the dispatching rules have the following equations:

Shortest Processing Time (SPT), each job has an associated priority P_j with j as the job number and t_{ij} as the processing time of operation i defined as:

$$P_j = \frac{1}{t_{ij} + \varepsilon} \tag{1.a}$$

If the next movement will be towards the Final Storage (FS), t_{ij} will be zero and the priority will be infinite, to prevent this, a small number (ε) is utilized in the priority equation.

Longest Processing Time (LPT):

$$P_j = t_{ij} + \varepsilon \tag{1.b}$$

In the case of LPT, the small number (ε) is used to distinguish the priority from zero, as this means, that the job is not available.

Least Work Remaining (LWR), sums all the processing times of a job j, from the current operation c, until the last one r:

$$\sum_{i=c}^{r} (t_{ij}) + \varepsilon \tag{1.c}$$

Since the transportation time is long, the robot's movement can be relevant in scheduling decisions, so two variation algorithms based on the movement time were employed, to analyze how they fare against traditional scheduling algorithms:

Shortest Movement Time (SMT): prioritizes jobs that are closer to the robot's current position, based on the time it would take for the robot to reach the desired Ws, the priority of a job is defined by the (eq. 1.d), represents the decision process.

Current Shortest Movement Time (CSMT): when the robot places a bottle in a workstation, it might be beneficial to wait for the process to finish, and transport that same bottle, CSMT employs the SMT idea for jobs currently not being processed or free jobs, and it compares the priority of these jobs (eq 1.d), with a priority of the job that the robot just transported, defined as the current priority (eq 1.e), which is based on the processing time.

Free Jobs: the priority equation P_j with j as the job number, Mp_{pi} being the pre movement time from current location of the robot p to the workstation i:

$$Pi_j = \frac{1}{Mp_{pi} + \varepsilon} \tag{1.d}$$

Current Job: Current job is defined as the job in which the robot is about to finish transporting, its priority equation is defined as:

$$Pc_j = \frac{1}{t_{ij} + \varepsilon} \tag{1.e}$$

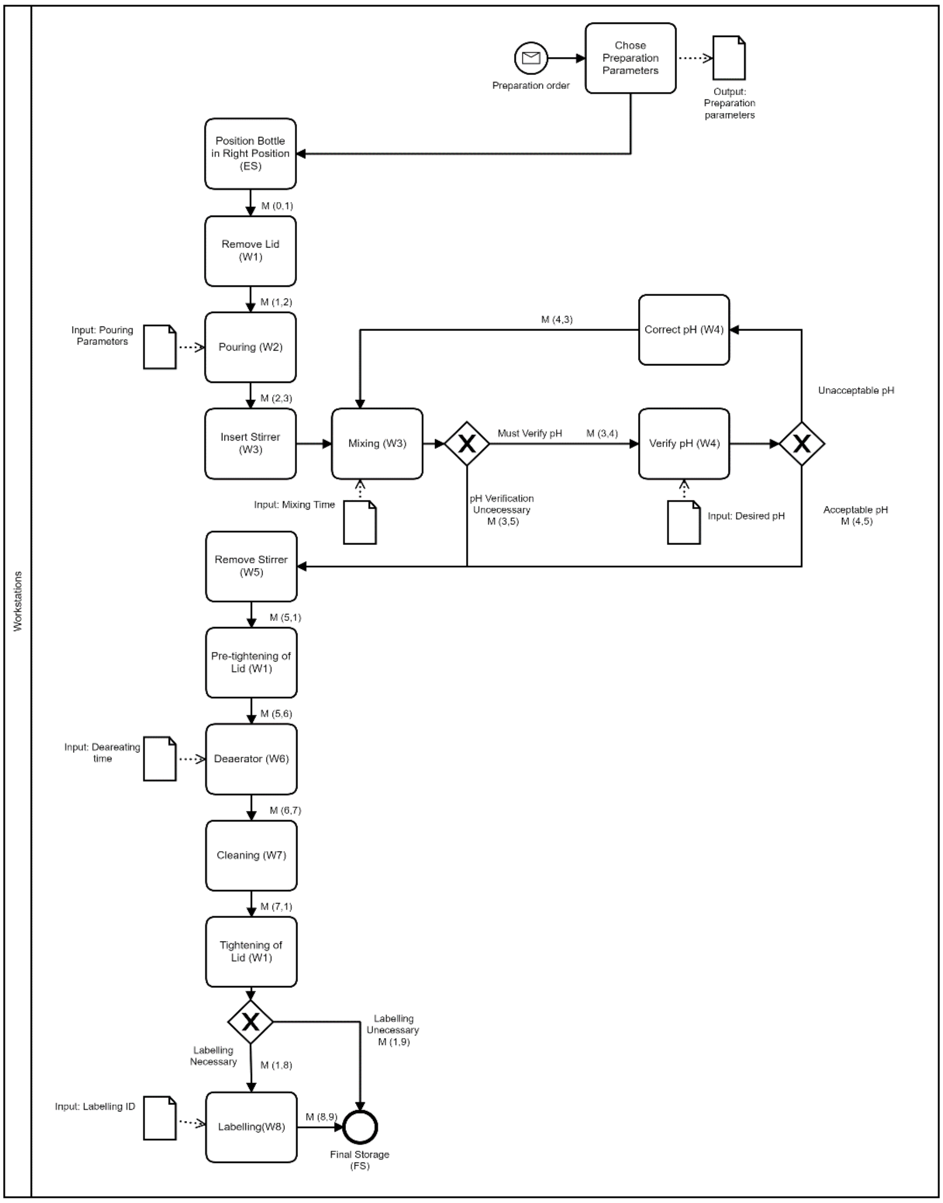

Figure 3: Workflow of each client request through the workstations in BPMN, the movements are represented only by the M letter

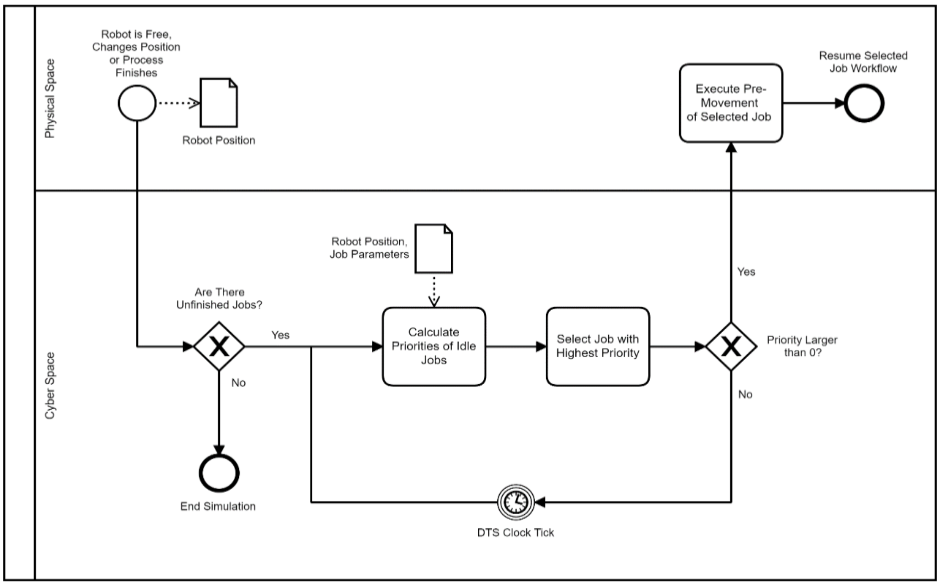

Figure 4: Decision workflow for the dispatching rules in BPMN

3.1 Model Translation

To develop the simulation model, the Anylogic software was utilized, three model is divided in workstations, each one with a set of Anylogic blocks such as Seize (to seize a workstation), Resource Pool (available Ws), Move by Crane (crane movement), Release (release a bottle from the workstation) and Delay (processing time). These activities can be described in Anylogic recurring to the process modeling library and the material handling library, the robot is described in Anylogic as a jib-crane, as the software does not have an animated robot, it emulates the movements and interaction with the bottles well enough in a visualization setting, for this the block (MovebyCrane) was utilized, which allows the modeler to define the trajectory and transportation time of the entities [17].

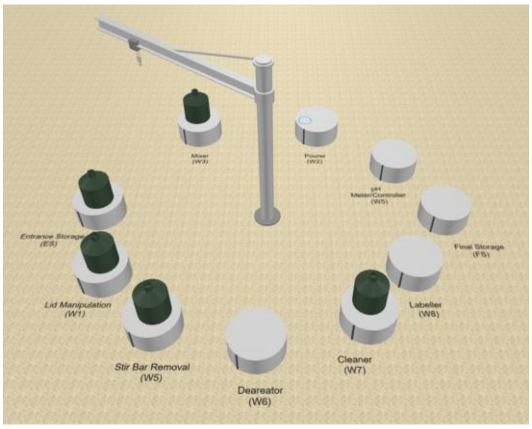

Figure 5: Visualization window in Anylogic during a simulation run

Anylogic also supports both 2D and 3D animation which helps the modeler in simulation, as it provides another angle to understand the model, and make sure its functioning correctly, as visualizing the robot movement and interactions with the bottles, is an easier way of understanding the simulation model while its running.

3.2 Data Processing

To collect data related with the robot's movement time, it's necessary to understand its motion, since the already existing system only provides a limited set of movements, shown in figure 3, some extrapolations need to be done to obtain all the possible movements. Since it is not relevant the movement of all joints, the robot's movement can be simplified, through the division of a single movement in three components, the approach motion time (ma), the rotational motion time (mr) and the exit motion time (me), these are to approach the workstation and place the bottle, to do the rotation between workstations, and to take a bottle from a workstation respectively.

When switching between jobs, the pre movement time (Mp_{pi}) is employed, it consists of the rotation motion without a bottle, from its current position (p), to the workstation where the next movement starts (i), then the robot performs the exit motion for the Ws (i), grabbing the bottle (figure 6)

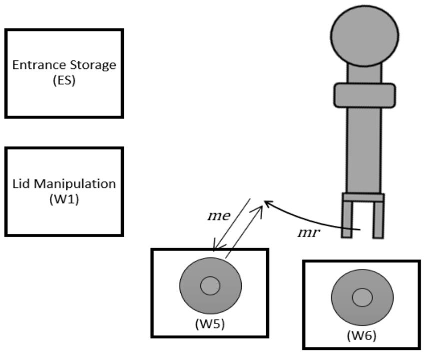

Figure 6: Pre movement from robot's current position W6 to W5

The movement time ($M_{ii'}$) consists of the rotation motion time (mr) with a bottle, from the workstation where the movement starts (i), to the next workstation (i'), where the robot leave the bottle, with the approach motion time (ma), this can be shown in figure 7.

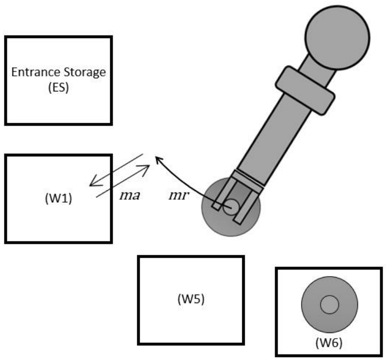

Figure 7: Movement time between W5 and W1

The rotation motion between several workstations was observed, the average speed v of the rotation time, varies slightly depending on the movement, since the complex robot movement was simplified in three parts, and the rotation speed is not constant, an acceptable rotation motion speed of the robot (v), for the rotation time (mr), must be chosen for calibration of the model, discussed in section 4.1.

Most processes are either stochastic, or can change in real-time, but there is the exception in some of them, in W2, in which the bottles are filled with different solutions, the quantity of each and different products depends on the customer's request. The time it takes to fill a recipient naturally depends on the amount of liquid solution, and since the quantity of liquid is measured with a scale, the liquid flow diminishes when the weight is close, taking longer to fill to the desired amount which adds variability to the process. The time to fill a bottle was approximated using the following linear regression:

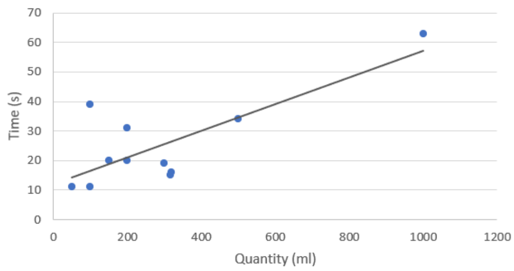

Figure 8: Plot of time to fill a bottle with solution, depending on the quantity, real data is represented by the blue dots, and the gray line is the approximation curve

As to emulate the high variability in the process, probability density function $f(x)$, utilized for normal distribution, was applied to the resulting data, and a triangular fitting was applied:

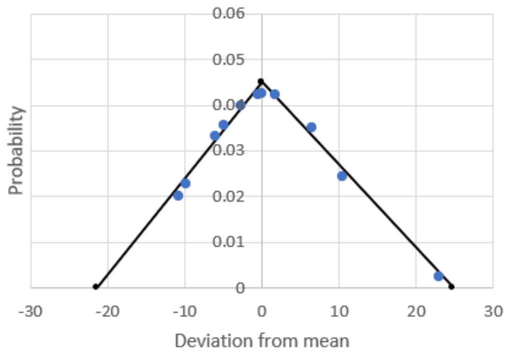

Figure 9: Probability density function of the time in W2, blue dots are the percentiles of the real data, and the black line is the chosen fitting distribution

The time in W3 depends on the amount of liquid filled in W2, and can be changed in real-time by a worker, a curve was created as to emulate the dependency on W2, to represent the variation, a triangular distribution was employed as well.

Linear approximations and triangular probability density functions were also utilized for describing the process time in W6, which can also be changed in real-time, with the help of individuals involved in the development of the system.

As for the rest of the workstations, the workflow is linear and does not depend on multiple parameters, the time data was obtained either by observation or by consulting entities involved in the development of the prototype.

Decisions such as the need for a bottle to check pH and labelling come from recipe in preparation requests, although real costumer requests are not available, individuals involved in the development can be consulted. Regarding the pH check, about 50% of requests need to go through this process, and out of the ones that do, 40% need to do it again. Regarding labelling, since the labelling machine can run out of labels, the system must still be able finish the jobs, and so 70% of recipes need a label, even though every bottle should have one.

With this information in respect to the workstations, a list of 500 different costumer requests was created, using uniform random distributions of linear approximations from the W2, W3 and W6 as well as the need to go to W4 and W8 for pH check and labelling respectively.

The variation in the processes, emulated by the triangular distributions, were imbedded in the model, as to simulate the stochastic conditions. regarding the W4, when a job must go to do a pH check from recipe, the probability of it going again follows a poisson distribution.

4. Simulation Study

4.1 Verification, Validation and Calibration

The verification was done by analyzing the simulation runs, step by step, through visual validation of the crane movement, checking if the parameters and variables are correct, and manually doing the math related with scheduling to check if the right decisions are being made by the model.

To calibrate and validate the model as a whole, the system can be seen as input-output transformations, where the model receives input parameters and transforms it in output measures of performance, testing the model's capability of predicting the future behavior (output) of the real system, when fed the same input data as in the real counterpart. For calibration, different values of rotation motion speed of the robot (v) where tested, to see which value more closely emulates the real counterpart as discussed in section 3.2.

Rotation speeds (v) from 3 rad/s to 4 rad/s were experimented, and the relative approximation error E_{iv} at the end of the operations (i) and accumulated relative approximation error E_v of all operations (r) of the model time relative to the actual time, defined by equation 2.1a and 2.1b respectively were used as measure of performance, with the results in figure 10 [18].

$$E_{iv} = \frac{|Ts_{iv} - Tr_{iv}|}{Tr_{iv}} * 100 \qquad (2.a)$$

$$E_v = \sum_{i=1}^{r} \left(\frac{|Ts_{iv} - Tr_{iv}|}{Tr_{iv}} * 100 \right) \qquad (2.b)$$

Tr_i: Real-time at which process i finishes for rotation velocity of v.

Ts_i: Simulation time at which process i finishes for rotation velocity of v.

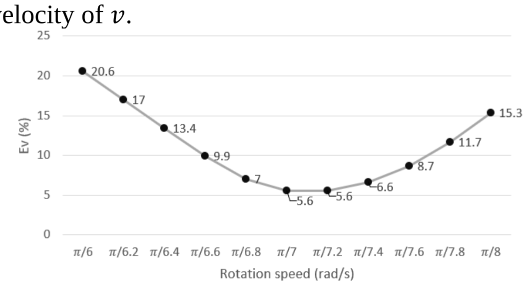

Figure 10: Plot of Accumulated relative approximation error, depending on the rotation speed

With testing results from different rotation speed values, it was shown that rad/s was the value that ensures the smaller accumulated error of 5.6%, and so, it was the chosen speed for calibration and validation.

Although this approximation is not as valuable as having real data regarding all the movements, it can describe the movement times for the existing system with enough accuracy, with this information, tables containing the rotation $(mr_{ii'})$, pre movement (Mp_{pi}) and movement times $(M_{ii'})$ with all the possible movements, where then defined, and fed to the simulation model for simulation runs.

4.2 Results and Analysis

With the model validated, the next step is to do simulation runs with multiple bottles and a variety of different parameters, which are called iterations, and discover what decisions reduce the makespan and increase the utilization of the equipment.

In this step, different model parameters are chosen to better understand where the system can be improved, for those, different measures of performance are utilized:

Makespan (T): time to process all the Jobs, or total time.

Resource Utilization (U_w): measures the utilization of a workstation or the robot, is the relation between the total working time of a resource (Tu_W) and the makespan (eq. 7.a):

$$U_w = \frac{Tu_w}{T} \quad (7.a)$$

Resource Occupation (Ro): measures the occupation of a workstation, that is, the percentage of time, these have bottles, independent of the working time, is the relation between the occupation time of a resource (To_w) and the makespan (eq. 7.b):

$$O_w = \frac{To_w}{T} \quad (7.b)$$

Performance Improvement (I): also called makespan reduction or reduction in total completion time, is to compare the makespan of different iterations, and measure the effect of parameter changes, with T_p as the makespan of the iteration used as a comparison term, (eq. 7.c).

$$I = \left(1 - \frac{T}{T_p}\right) * 100 \quad (7.c)$$

4.2.1 Parallel Tasks

The prototype currently only supports single tasking, where a new job is only processed when the last one is completed, as said in the introduction, one of the goals of this thesis is to see the effect of parallel tasking in the completion time, with different heuristics. For this, an iteration with single tasks was compared with the parallel tasking with the dispatching rules referred in section 3, although only the best performing dispatching rules are shown, 50 client requests were used as inputs to the model, with the following results:

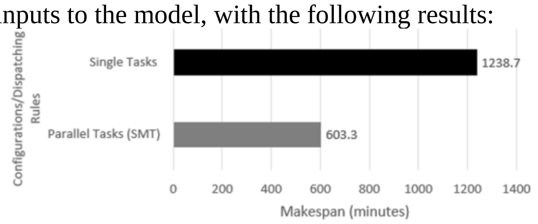

Figure 11: Makespan comparison of single tasking with parallel tasking with the best performing SMT rule

Combining parallel tasking with the SMT dispatching rule originates the best results, reducing the overall completion time by 51.3%, when comparing with single tasking, making this change imperative to improve the system, as well as allowing the system to be improved by adding identical parallel machines in key processes.

4.2.2 Home Position

Individuals involved in the development of the prototype claimed the robot goes to the home position for safety reasons, as it's a prototype, but they claim that in the future its intended for the need to go to the home position in not necessary, and so, the model was put to test to determine the effect of the home position in the performance.

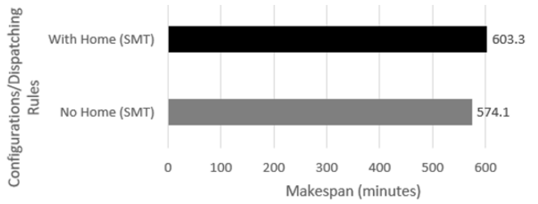

Figure 12: Makespan comparison of having to go to home position and not going to the home position (gray)

Going to the home position does not affect the process time dramatically, with a reduction in makespan of 4.8%.

This is because, most of the time, no more than one bottle is available, which is proven by the low utilization of the robot, as its idle almost half the time.

4.2.3 Resource Allocation

Tables with resource utilization, resource occupation and performance improvement were used to better understand which workstations should have parallel machines. The W3 has the largest utilization and occupation rate, and its predecessors in the workflow, as seen in figure 3, W2 and W4, have a large difference between its occupation and utilization rates, implying that the bottles at these Ws spend most of the time waiting, which indicates a possible bottleneck at W3, so the first step was adding an extra W3.

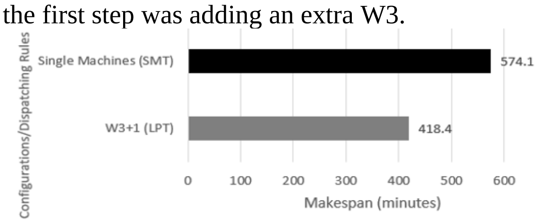

Figure 13: Makespan comparison of having single machines, and having an extra W3

As seen on figure 13, adding a parallel W3 yields an increase in performance of about 27.1% when comparing with the best performing dispatching rule with single machines, meaning the W3 is a bottleneck, applying this change in the real counterpart proves effective, with the LPT algorithm being the most effective for the configuration. With this new configuration, the W3 might still be a bottleneck as its utilization/occupation ratio is still high, W6 might also be another possible bottleneck, as it has both a large utilization and occupation, W1, with a large occupation and having each job going through it three times (figure 3), makes considering an extra parallel W1 a possible improvement to the system, the next iterations where made based on these comments, with the best ones shown below:

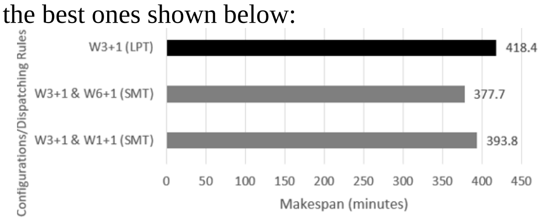

Figure 14: Makespan comparison of having two W3s, with having an extra W3 with an W6 or W1

According to the simulation model, adding an extra machine to W6 when already having two W3s, further decreases makespan by about 9.7%. Adding an extra W1, when already having two W3s, results in a performance increase of 5.9%, it might not be as effective as adding a W3 and a W6, depending on the cost of the machines it might be a viable option.

W3 still possesses the highest utilization, so its still restricting the workflow and W6 has still the 2nd overall highest utilization. W1 appears to restrict the flow of tasks, as W5 has a low utilization/occupation ratio, since each job need to go through W1, having a single machine might create a choke point. As the combination of extra parallel W3 and W6 has the lowest makespan for most dispatching rules, it is utilized as a comparison term to other configurations:

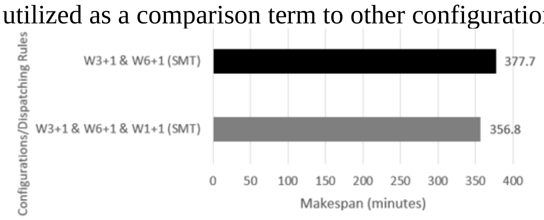

Figure 15: Makespan comparison of having two W3s, W6s and W1s, with having three W3s and two W6s and W1s (gray)

Including an extra W1 to the configuration that already possesses two W3 and W6, yields the largest reduction in makespan of 5.5%.

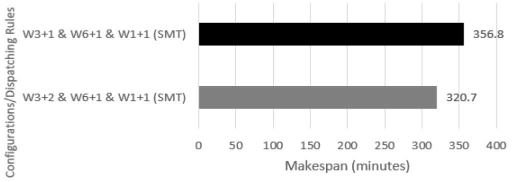

Figure 16: Makespan comparison of having two W3s, W6s and W1s, with having three W3s and two W6s and W1s (gray)

Adding an extra W3, relative to previous configuration, further reduces the makespan by 10.1%, other additions were tested, using this configuration as a reference, but the makespan does not decrease more than 2%, so adding more Ws does not yield much better results.

The robot is now utilized most of the time (92.6%), with its high utilization, the movement time might significatively be hindering the workflow, as the workstations still have an overall low utilization, and increasing the number of parallel machines no longer yields reductions in makespan larger than 2%, also the fact that the SMT rule, which relates with the robot's speed, is performing better than other dispatching rules, makes a study on the robot movement speed a relevant topic.

4.2.4 Robot Speed

The robot currently works at a slow speed as it is only a prototype, but it can be increased further down the line.

The configurations that have the have the largest impact on the makespan as well as the single machines configuration, were employed and a factor of 1.25 and 1.5, was multiplied by both the pre movement and movement time for this experiment, with the following results:

Standard configuration:

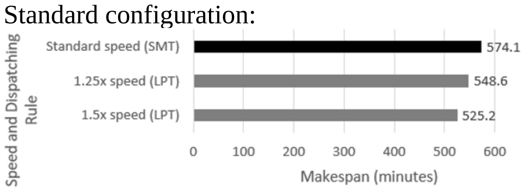

Figure 17: Makespan comparison of increasing the robot speed for the single machines configuration

Two W3s:

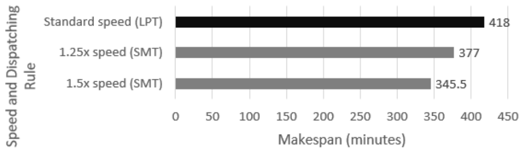

Figure 18: Makespan comparison of increasing the robot speed for the configuration with an extra W3

Two W3s and W6s:

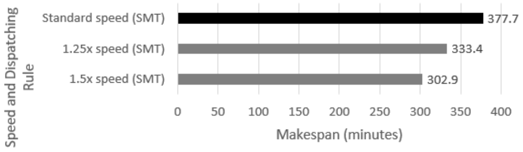

Figure 19: Makespan comparison of increasing the robot speed for the configuration with an extra W3 and W6

Two W3s, W6s and W1s:

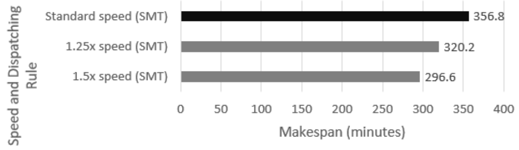

Figure 20: Makespan comparison of increasing the robot speed for the configuration with an extra W3, W6 and W1

Three W3s and two W6s and W1s:

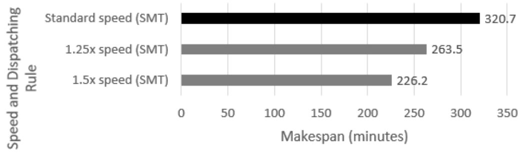

Figure 21: Makespan comparison of increasing the robot speed for the configuration with two extra W3s, and one extra W6 and W1

Increasing the velocity of the robot does not reduce the single machines configuration's completion time by a large margin, but for the other configurations, if the velocity of the robot can be increased by 25% and 50%, the makespan is reduced by at least 9.8% and 20.5% respectively.

Combining an increase in robot speed, with adding key workstations, can be considered as the best option, choosing the right combination depends on the cost and feasibility of the changes, information that the author does not possess.

5. Conclusions and Future Work

The premise of a manufacturing system capable of handling different customer requests while dealing with real-time changes, was achieved, by developing a Digital Twin focused on simulation recurring to scheduling of parallel tasks. A mixed event simulation model was then developed, capable of handling various customer requests, robot movement speeds, as well as different combinations of machines, it recurs to five different heuristics: SPT, LPT, LWR, SMT and CSMT, to make scheduling decisions. Simulation runs with different parameters were experimented, it was proven that applying parallel task scheduling with a SMT dispatching rule, reduces the makespan by 51.3%, when comparing with single task scheduling. Experiments with additional parallel identical machines were also conducted, the single addition that proves most effective is adding one W3, which decreases the overall completion time by 27%. The robot's speed was also a topic of analysis, and it proves beneficial when parallel machines are added, in the configuration with two extra W3 and one extra W6 and W1, when increasing its velocity by 50%, yields a reduction in makespan of 29.5%, compared to standard speed. Between all heuristics, SMT proved to be the most effective in reducing makespan, with the LPT outperforming the SMT in a few configurations.

Overall, the objective of developing a Digital Twin capable of making real-time decisions, was achieved by utilizing completely reactive scheduling. With the ability do forecasts with different dispatching rules, various machine configurations and robot speed, with access to a visualization window, for a more intuitive understanding of the model, gives stakeholders more information about where the system can be improved.

The next step is to apply this concept to the real asset, where the input of the Digital Twin wouldn't be historical data, but instead, real-time data from the asset, recurring to sensors and the already existing software. For this, an interface that allows processing times and decisions to be altered in real-time, when an operator decides, instead of probability distributions, that represent the decision, would need to be integrated into the DT. The operator should also have the option to makes scheduling decisions, as told in the introduction, people have problem-solving skills and knowledge of the system, therefore, they should have the ability to override scheduling decisions of the DT, when they see fit, which is a topic that is included in the goals of industry 5.0.

References

[1] M. M. Tseng, Y. Wang, and R. J. Jiao, "Mass Customization," *CIRP Encyclopedia of Production Engineering*, pp. 1–8, 2017, doi: 10.1007/978-3-642-35950-7_16701-3.

[2] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International Journal of Flexible Manufacturing Systems 2006 17:4*, vol. 17, no. 4, pp. 261–276, Oct. 2006, doi: 10.1007/S10696-006-9028-7.

[3] Maija. Breque, Lars. de Nul, Athanasios. Petridis, and European Commission. Directorate-General for Research and Innovation., *Industry 5.0 : towards a sustainable, human-centric and resilient European industry*. doi: 10.2777/308407.

[4] T. Coito *et al.*, "A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation," *Computers in Industry*, vol. 123, p. 103329, Dec. 2020, doi: 10.1016/J.COMPIND.2020.103329.

[5] R. Boschert Stefan and Rosen, "Digital Twin—The Simulation Aspect," in *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*, D. Hehenberger Peter and Bradley, Ed. Cham: Springer International Publishing, 2016, pp. 59–74. doi: 10.1007/978-3-319-32156-1_5.

[6] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling 2008 12:4*, vol. 12, no. 4, pp. 417–431, Oct. 2008, doi: 10.1007/S10951-008-0090-8.

[7] Averill M. Law and David Kelton, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.

[8] A. Alrowaie Ahmed, "The effect of time-advance mechanism in modeling and simulation," M.S. thesis, Naval Postgraduate School, Monterey, USA, 2011. Accessed: Aug. 22, 2021. [Online]. Available: https://calhoun.nps.edu/handle/10945/10798

[9] Jerry Banks, John Carson, Barry Nelson, and David Nicol, *Discrete-event System Simulation. 5th ed.* 2016.

[10] T. Coito et al., "A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation," Computers in Industry, vol. 123, p. 103329, Dec. 2020, doi: 10.1016/J.COMPIND.2020.103329.

[11] H. Yu, S. Han, D. Yang, Z. Wang, and W. Feng, "Job Shop Scheduling Based on Digital Twin Technology: A Survey and an Intelligent Platform," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/8823273.

[12] W. Hofmann and F. Branding, "Implementation of an IoT- and Cloud-based Digital Twin for Real-Time Decision Support in Port Operations," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2104–2109, Jan. 2019, doi: 10.1016/J.IFACOL.2019.11.516.

[13] P. Karagiannis, N. C. Zacharaki, G. Michalos, and S. Makris, "Increasing flexibility in consumer goods industry with the help of robotized systems," *Procedia CIRP*, vol. 86, pp. 192–197, Jan. 2019, doi: 10.1016/J.PROCIR.2020.01.039.

[14] A. Fahmy, T. Hassan, and H. Bassioni, "What is Dynamic Scheduling?," *PM World Journal*, vol. 3, Aug. 2014.

[15] K. Muhamadin, M. Shukri, and O. Khayal, "A Review for Dynamic Scheduling in Manufacturing," *The Global Journal of Researches in Engineering*, vol. 18, no. 5-J, pp. 25–37, Aug. 2018, doi: 10.13140/RG.2.2.15345.33129.

[16] T. Allweyer, *BPMN 2.0*. Norderstedt, Germany: BoD, 2010.

[17] Anylogic, "Help Section." https://anylogic.help/ (accessed Sep. 09, 2021).

[18] P. Thompson and Y. Liu, "Understandings of Margin of Error" Sep. 2005, doi: 10.21037/jtd.2017.09.14.