Journal of Manufacturing Systems 65 (2022) 538–549

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Development of a digital model and metamodel to improve the performance of an automated manufacturing line

Patrick Ruane [a,*], Patrick Walsh [b], John Cosgrove [b]

[a] *Johnson & Johnson Vision Care and Technological University of the Shannon, Limerick, Ireland*
[b] *Technological University of the Shannon, Limerick, Ireland*

ARTICLE INFO

Keywords:
Simulation
Metamodel
Digitalization
OEE
MTBF
MTTR
Reliability

ABSTRACT

Digitalization in manufacturing is the conversion of information into digital format, the integration of this digital data and technologies into the manufacturing process and the use of those technologies (eg: simulation) to change a business model to provide new revenue and value-producing opportunities. Digitalization may be seen as the increased generation, analysis, and use of data to improve the efficiency of the overall manufacturing system. Simulation in manufacturing is often applied in situations where conducting experiments on a real system is impossible or very difficult due to cost or time to carry out the experiment is too long. A key input to the simulation model of automated equipment is the acquisition of valid data in relation to cycle time and reliability of various workstations on this line. As a consequence of being able to simulate equipment processes and interact with this validated simulation model, both the understanding of how the production system will perform under varying reliability and cycle time conditions is achieved. The simulation model then enables the experimentation of 'what if scenarios' that can be tested easily, while also providing a valuable tool to inform the maintenance personnel what station reliabilities they need to target in order to sustain a high performing manufacturing line. Simulation metamodeling is an approach to line design which is of great interest to design engineers and research experts. However, its application in automated medical devices manufacturing line design has never been well explored. The author has adopted an open-source simulation tool (JaamSim) to develop a digital model of an automated medical devices manufacturing line in the Johnson & Johnson Vision Care (JJVC) manufacturing facility. This paper demonstrates with a high level of rigour, fidelity and overall system design/approach, how a digital model along with the use of a metamodel can be used for the development of an automated manufacturing line in the medical devices industry. The digital model and metamodel can be used by manufacturing engineering teams to perform scenario testing during the design and development phase of the line or as part of the continuous improvement stage when the line is in full operation. The overall average absolute error when comparing the simulation model outputs to the metamodel outputs was 0.87% was achieved with the metamodel for the actual industrial application used by the author.

1. Introduction

Digital manufacturing technologies such as simulation models, have been considered an essential part of the continuous effort towards improving the Overall Equipment Efficiency (OEE) of automated manufacturing equipment and processes. These digital technologies such as a Digital Twin (DT)/Digital Model (DM) have proven to be a powerful tool to support the design and evaluation of a manufacturing system due to its low cost, quick analysis, low risk and meaningful insight that it can provide, thus improving the understanding of the influence of each component/system on the automated manufacturing line [1].

1.1. Digital twin technology

A Digital Twin is a digital model of real-world physical system. The digital twin concept allows manufacturing companies to create models of their production systems and processes using real-time data collected from smart sensors [2]. The digital twin and the physical system are connected through IoT or smart sensors and actuators. A digital twin of a

* Corresponding author.
E-mail addresses: patrick.ruane@tus.ie (P. Ruane), patrick.ruane@tus.ie (P. Walsh).

https://doi.org/10.1016/j.jmsy.2022.10.011
Received 27 May 2022; Received in revised form 5 August 2022; Accepted 13 October 2022
Available online 26 October 2022

physical system is utilized to increase the performance of the product throughout its life cycle. This is done through the employment of data driven simulation during the design and planning phase as well as while the system is in operation. This process enables the user to optimize the performance during the whole lifetime of the product. Given the definition of a digital twin manufacturing system, two main functionalities are needed, a virtual representation of the system and an interface to the real system allowing synchronization, re-planning and the capability to allow process owners to make changes to the digital twin. A Digital Twin (DT) enables in-depth analyses to detect potential issues, preventing downtime, and testing new manufacturing line opportunities, and customize production based on customer requirements. It is recognized that computer simulations are an integral part of the digital twin concept [3].

A recent comprehensive work on digital twins [4] for production systems, defined a digital twin as, "a digital representation of an active unique product or service or production system that is characterized by certain properties or conditions used in order to analyse, understand and improve the product, product service system or production". Such models can be parameterized, and they are able to consider several influences including stochastic behaviour [2]. According to [2] a digital twin can be a simulation model, but a simulation model may not necessarily be a digital twin. Simulations models may have the same type of sensor information and controls of a digital twin, but the information may be generated and manipulated within the simulation. The simulation model may replicate what could happen in the real world, but not necessarily what is currently happening [5]. Digital models can be classified into three (3) subcategories based on their level of data integration between the physical and digital counterparts [6]:

1. Digital model: A digital representation of an existing or planned physical object without any form of automated data exchange between the physical and digital objects. Most of the current offline simulation models are this kind of digital model.
2. Digital shadow: A digital model with an automated one-way data flow between the physical and digital objects, e.g., a simulation model using real-time sensor data as inputs.
3. Digital twin: A digital model with bi-directional data flow between the physical and digital objects, e.g., a simulation model that uses real-time sensor data as inputs and updates some of the parameters of a manufacturing process or equipment.

This paper reviews the development of a digital model and metamodel of an industrial use case (Tray Loader System) that can be used during both the equipment design stage and continuous improvement/ optimization of the actual system. A digital model/simulation model was built that simulates product loaded into plastic trays, then processed through the various stations on the line. The simulation model is used to determine the impact of changing cycle time and reliability of the process stations whilst at the same time altering the size of various buffers in the system. A metamodel was also developed and was also used to identify the impact of changes to the system very quickly without the need to continuously update the actual simulation model and then running that model each time to determine the impact those changes have on the line performance. The Metamodel along with the Digital/ Simulation model can be used by equipment designers to test different equipment designs and determining the resultant response performance metrics. Metamodels, commonly known as surrogate models, response surfaces, approximate models or emulators are used to approximate the input-output behaviour of simulation models [7]. The metamodel is a model of the I/O function (or 'response function') implied by the underlying simulation model.

The aim of this paper is to propose a mathematical model (formulation) and a methodology as a solution to maximize production rate (throughput of the Tray Loader system). Many of the previous works on this subject are limited as they focus mainly on one line factor only, are limited to a small number of workstations/process steps, not based on an actual industrial use case, whereas this work focuses on a complete section of an actual manufacturing line and also on the combination of line factors that drive overall system throughput/performance of this automated manufacturing line, these being: reliability, cycle time and buffer capacities. Although significant challenges are apparent in developing Digital Twin applications they offer the promise of extending the use of simulation from traditional stand-alone system design applications to simulation as a core functionality of systems by means of seamless assistance across the entire lifecycle from design, engineering, operations to service [8].

1.2. Overall equipment efficiency (OEE)

The concept of OEE, introduced by [9], is being used increasingly in industry. It looks at the wider manufacturing aspects, not only the equipment availability and performance, but the efficiency losses that result from rework and yield losses. According to [10] the relationship between OEE and losses depends on equipment availability, their performance rates, and the quality of the product. OEE monitors the actual performance of a machine relative to its performance capabilities under optimal manufacturing conditions. According to [11] Manufacturing Line efficiency can be expressed, using the OEE metric that depends on three factors: availability, performance, and quality.

$$\mathrm{OEE} = (\mathrm{Availability})\mathrm{x}(\mathrm{Performance})\mathrm{x}(\mathrm{Quality})$$

Availability is the ratio of the time spent on the realization of a task to the scheduled time. Availability is reduced by disruptions at work and machine failures. According to [11] the term of availability contains planned work time and unplanned events e.g., the disturbances at work and random machine failures. Any unplanned event that causes the equipment to be unavailable results in reduced efficiency. The reliability of systems or devices such as sensors, robots, conveyors are defined as the probability that they will work correctly for a given time under defined conditions of work. The most popular method for estimating reliability parameters uses the theory of probability to forecast a value of MTTF (Mean Time to Failure), MTBF (Mean Time Between Failures) and MTTR (Mean Time to Repair). The use of normal, exponential, triangular distributions to describe both failure and repair times are often used. In practice, for description of reliability, the parameter MTTF is normally used, which is the expected value of exponentially distributed random variable with failure rate λ. In the case of repairable objects, the parameter MTBF and the MTTR are used.

$$\mathrm{MTBF} = \mathrm{MTTF} + \mathrm{MTTR}$$

Machinery failures affect the availability of means of production and may cause severe disturbances in production processes. The average availability is given by:

$$\mathrm{Availability} = \mathrm{MTTF}/\mathrm{MTBF}$$

1.3. Equipment reliability & maintenance of manufacturing lines

Manufacturing equipment reliability is a significant factor that plays an important role in the ability of equipment to perform to the required levels in operation. Reliability is a measure of an equipment's ability to operate efficiently within set limits and confines of time [12]. Optimizing reliability is paramount for the successful operation of equipment and the minimization of costs associated with downtime and unexpected breakdowns [13]. Equipment that spends most of its time running efficiently under continuous operations, indicates that reliability is being achieved due to regular maintenance activities being done thus delivering smooth operation. Maintenance is an important consideration in the enhancement of equipment reliability. Measures such as MTTF, MTBF and MTTR help define the reliability of a given station/line [14,15]. Consistent maintenance ensures that

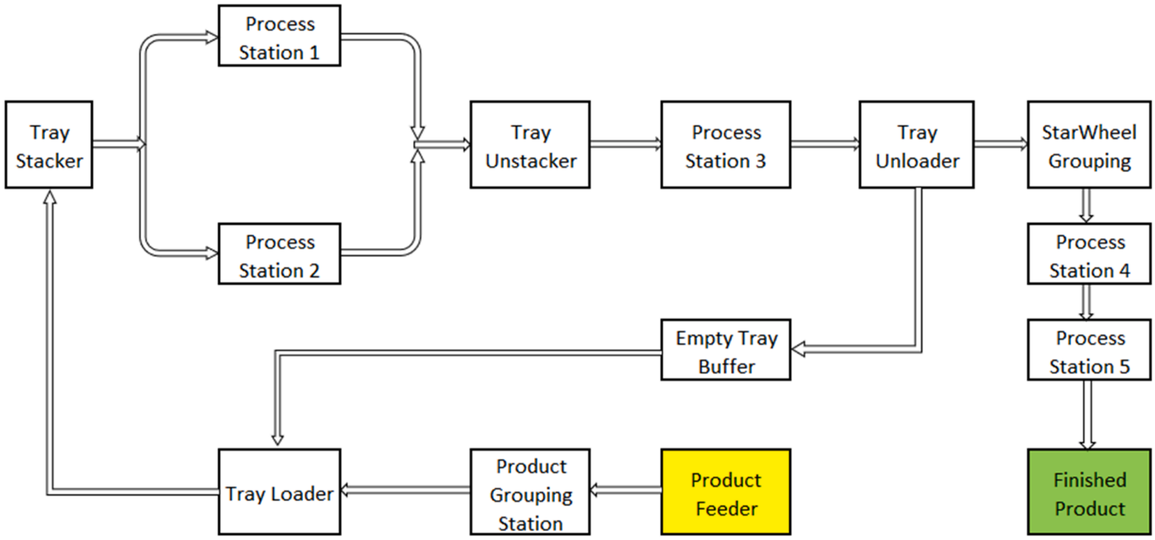

Fig. 1. Automated Tray Loading System Industrial Case.

manufacturing lines operate to their optimum performance and achieve their business targets [16]. MTTR is easily determined using the number of repair hours divide by the total number of repairs within that specified period. The higher the MTTR the greater it negatively affects line throughput. Another measure of line reliability is MTTF and is determined by dividing the total number of operation hours within a predetermined period by the number of failures [17]. Mean Time Between Failure (MTBF) is the sum of MTTF and MTTR. Maintenance is a set of organized activities that are carried out to keep an item in its best operational condition with minimum cost acquired. The cost at times may appear high in the beginning, but they are intended to keep the overall condition of the equipment better and its operating and other expenses low (considered over its life span) [18]. It is also important to improve equipment reliability throughout an equipment's life to meet the business goals and objectives [13].

A goal of high performing manufacturing companies including Johnson & Johnson Vision Care (JJVC) are as follows:

1. Eliminate/reduce the unplanned maintenance activities (station set-up after component changeout, minor station adjustments/tweaking to improve station performance) by designing out these non-value add and sometimes repetitive activities during the original equipment design.
2. Reduce the planned maintenance activities (component changeout) to a minimum, and then design the line to require the minimum time to replace this component/system with little/no set-up thereafter.

The equipment design has a significant influence on all these factors. More reliable components are selected requiring less frequent maintenance and the equipment design is such that it allows components to be replaced quickly with minimal/no set-ups post change-out. Simulation is a key technology for the development of planning and exploratory models to optimize decision making including the design and operation of these complex and smart production systems [19]. Simulation and Metamodelling was a tool that was used by the author to verify and aid in confirming the performance capability of the equipment design before it is built. Reducing both the planned and unplanned maintenance activities has the effect of increasing the available time that the manufacturing line is available to run, thus increasing line efficiency. The author has shown how simulation and metamodelling can be used as an approach to predict the performance capability of new manufacturing lines during the design stage. If the new line is not already built/running, the simulation model can use data (station cycle times and reliability) from pervious similar equipment designs or actual component data supplied by the Original Equipment Supplier (OEM).

2. Development, verification and validation of the tray loader digital model

2.1. Industrial use case overview

A digital model of an industrial system (Fig. 1) known as a Tray Loading System was developed using JaamSim software.

This system consists of individual product (p) that arrives from an upstream line to a product feeder at defined arrival times. These are then grouped into multiples of 10. The group of products are then loaded into empty plastic trays that can hold up to 660 parts. Once filled the plastic tray moves at a defined cycle time to a tray stacker. The tray stacker accumulates the filled trays into groups of 30. This group of 30 trays then undergoes a batch process in either Process station 1 or 2 under defined conditions. Upon completion of this batch process, the trays of product leave Process Station 1 or 2, where a tray unstacking operation takes place. Each individual tray of product undergoes a further process step (Process Station 3), again under defined conditions. Once a tray is finished at Process Station 3, the product is removed from the tray at the Tray Unloading station and is then passed to the Star Wheel grouping station, where the product is now grouped into batches of 30. These groups are then passed to Process Station 4 and 5 for the final finishing process. The empty trays from the tray unloading station, are returned to the empty tray buffer and finally back to the tray loader operation, to repeat the overall process. The digital model developed, will simulate this whole operation, considering the following 5 points:

1. Entities (units of Product) per arrival.
2. Service times for process stations, travel times forconveyors
3. Probability distributions for reliability and repair ofstations.
4. Conditions for process stations to process and pass productto the next station.
5. Queue size and location.

2.2. Verification of the tray loader digital model

A detailed verification process was undertaken on the Tray Loader digital model following the Logical/mathematical verification, program/code verification steps outlined by [20] and the detailed knowledge of the author of the actual tray loading system. All the Tray Loader

Table 1
JaamSim Downtime Entity Input Parameters.

Keyword	Description
Description	A free form string describing the Entity
FirstDowntime	The calendar or working time for the first planned or unplanned maintenance event. If an input is not provided, the first maintenance event is determined by the input for the Interval keyword. A number, an object that returns a number, or an expression can be entered.
IntervalWorkingEntity	The object whose working time determines the occurrence of the planned or unplanned maintenance events. Calendar time is used if the input is left blank.
DurationWorkingEntity	The object whose working time determines the completion of the planned or unplanned maintenance activity. Calendar time is used if the input is left blank.
Interval	The calendar or working time between the start of the last planned or unplanned maintenance activity and the start of the next maintenance activity. A number, an expression, or an object that returns a number can be entered.
Duration	The calendar or working time required to complete the planned or unplanned maintenance activity. A number, an expression, or an object that returns a number can be entered.
MaxDowntimesPending	The maximum number of downtimes pending for the downtime event

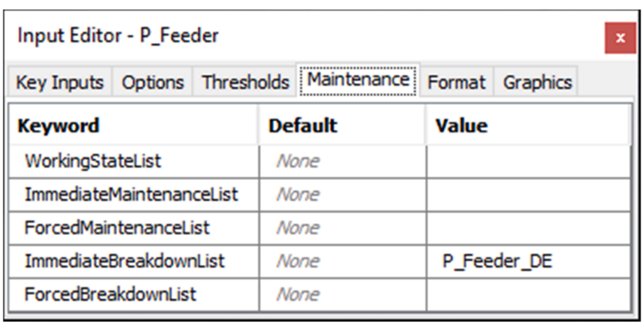

Fig. 2. P_Feeder Object Maintenance Configuration.

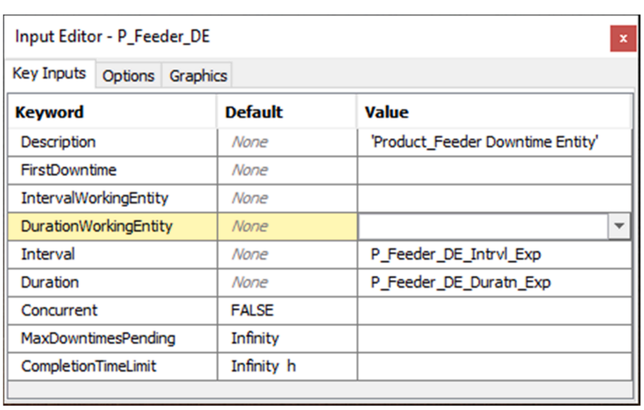

Fig. 3. P_Feeder_DE Parameter Configuration.

Objects, Service Times, Steps, Thresholds, Maintenance conditions and Threshold condition logic were all verified and confirmed to be correct to how the actual line operates. A detailed verification checklist was completed on the Tray Loader digital model. As part of the digital model verification process it was important to verify that the product flow into and out of the various simulation objects (as seen from the JaamSim GUI) are identical to what occurs on the tray loader line. This verification process allowed any additions or changes to the simulation logic to be corrected, verified, and visualized immediately. It was through the ongoing and iterative model verification and the testing process during model development, that a realistic model of the actual dynamic interactions was developed and fine-tuned. During this phase of model verification, the weak points of the system were discovered and corrected. It is extremely advantageous to find these early-stage simulation bugs, thus allowing a well-tested and robust system to be developed.

2.3. Validation of the tray loader digital model

The approach taken for developing the Tray Loader digital model followed the steps described by [21]. Step 5 of this approach deals with confirming that the programmed model is valid. The model is run using the standard basic settings from the actual tray loader system. The simulation model output data for the system was compared with the comparable output data collected from the actual system. This is called results validation. If the results are consistent with how the system should operate, then the simulation model is said to have face validity. Sensitivity analyses is performed on the programmed model to see which factors have the greatest impact on the performance measures and, thus, must be modelled carefully [21]. According to [22], validation is concerned with determining whether the conceptual digital model (as opposed to the computer program) is an accurate representation of the system under study. [22] outlines the following three (3) steps to validate a simulation model.

1. Obtaining real-world data from the actual system.
2. Tests for comparing simulated and real data (namely graphical, Schruben-Turing or t tests).
3. Sensitivity analysis (using statistical design of experiments with associated regression analysis).

The above approach was used to validate the Tray Loader digital model, see section 3 for more detail. Actual Tray Loader system data was collected from the historian database for all the relevant process stations used in the digital model. The data collected included input feed rate, yield, throughput and uptime per minute for each process station. Excel macros were then developed to calculate the equipment reliability metrics namely: Mean Time Between Failures (MTBF) and Mean Time to Repair (MTTR) for each of the process stations using the uptime/minute data. The Input feed rate, yield, output data and the MTBF/MTTR for each process station was analysed, outliers removed, and distributions determined along with the distribution parameters. Minitab is used to analyse all the data obtained. Minitab is a statistical analysis software that assists in the analysis of data collected from any process and provides a simple, effective way to input the data, manipulate that data and statistically analyse it. The methodology used to determine the MTTR and MTBF for the actual Tray Loader System will be described in section 3.

3. Determination of simulation input data

3.1. JaamSim downtime entity

Planned maintenance and breakdowns are modelled using the DowntimeEntity, which generates random or scheduled events based on either working time or calendar time. Normally, a maintenance activity is scheduled to occur at regular intervals based on calendar time. Breakdowns are normally modelled to occur randomly based on the working time for the object. The '*DowntimeEntity*' object in JaamSim is used to generate planned and unplanned maintenance events for various types of objects (workstations, buffers, conveyors). The *DowntimeEntity* generates the downtime events and their durations, but the objects that use one or more *DowntimeEntities* must provide their own logic for halting. See Table 1 for the list of *DowntimeEntity* Input parameters used in the Tray Loader application.

In JaamSim a DowntimeEntity object is then dragged and dropped

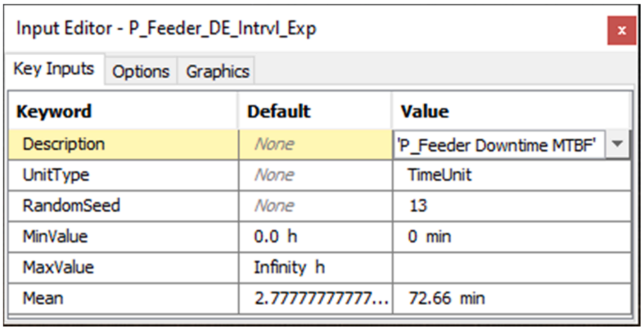

Fig. 4. P_Feeder Station MTBF Configuration.

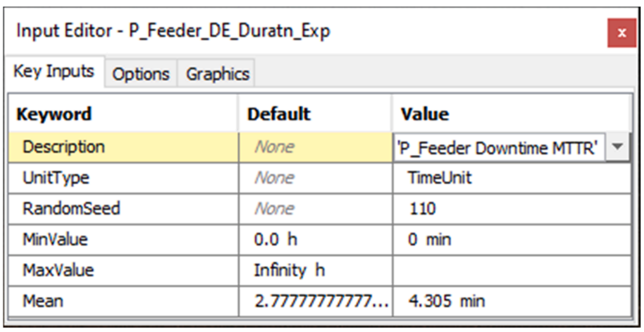

Fig. 5. P_Feeder Station MTTR Configuration.

from the JaamSim Model Builder to the application view window. On the Tray Loader digital model, one station called *P_Feeder*. One of the parameter settings to be configured is the *ImmediateBreakdownList* located within the maintenance table *ImmediateBreakdownList* is modelled in the Tray Loader application as it is unplanned, takes the production line immediately out of operation and is not accounted for in the business plan schedules. A *DowntimeEntity* object called *P_Feeder_DE* is referenced within the corresponding value cell for the *P_Feeder* object, see Fig. 2.

ImmediateBreakdownList is a list of DowntimeEntities representing unplanned maintenance performed immediately, interrupting any work underway at present. Other maintenance types such as *ForcedMaintenanceList*, *ForcedBreakdownList* or *ImmediateMaintenanceList* can also be configured depending on the user application to be modelled. A DowntimeEntity object is then dragged and dropped from the JaamSim Model Builder to the view window. The object is called *P_Feeder_DE* with input parameters configured as shown in Fig. 3.

The *Interval* is the working time between the start of the last unplanned downtime activity and the start of the next downtime activity. A number, an expression, or an object that returns a number can be entered. The *Duration* is the working time required to complete the downtime activity. A number, an expression, or an object that returns a number can be entered. In this research, *Interval* is used to replicate the Mean Time Between Failures (MTBF), whereas *Duration* is used to replicate Mean Time to Repair (MTTR) for each object of the model. Two Exponential Probability Distribution objects are then dragged and dropped from the JaamSim Model Builder to the view window. The objects are called *P_Feeder_DE_Intrvl_Exp* and *P_Feeder_DE_Duratn_Exp* to simulate the MTBF and MTTR respectively for the P_Feeder station. Input parameters are configured for both distributions as shown in Figs. 4 and 5.

The *UnitType* is selected as *TimeUnit*, *RandomSeed* is automatically selected by JaamSim and *MinValue* is set to 0 s. A python program was developed to automatically take settings/parameters from an excel file and populate the values for means into both distributions automatically.

3.2. Determining the Actual values for MTBF and MTTR

It is important that the downtime distribution for both Interval (MTBF) and duration (MTTR) are correctly set-up for each station and are reflective of the reliability from the actual Tray loader system. If these distributions, their value's and/or parameter settings are not representative of the actual system, then all results obtained from the simulation model will be of little use. In order to determine the actual distributions and parameter values for the MTBF and MTTR for the Tray loader system, data was collected from the historian database over a period of 186 shifts (1 shift = 12hrs). Total uptime for each station was recorded every minute over a period of 129,000 min. The cumulative total uptime (sec) was recorded for each station of interest from the database. Briefly outlining how the data was captured and analysed, refer to Table 2 for example of how Mean Time Between Failures (Interval) and Mean Time to Repair (Duration) is determined for P_Feeder station. The cumulative uptime for a particular station is recorded at 1 min intervals. The uptime/min is then determined for each minute over the whole duration. From Table 2, the uptime for 7:05:00 is 155 – 115 = 40 s. The % uptime between 7:04:00 and 07:05:00 = $\frac{40}{60} \times 100$ = 66.6%. This process is repeated for each minute. For the purpose of calculating the MTTF and MTTR, the total uptime is calculated by adding the uptime for each minute until the station goes stops (< 60 s uptime for that minute). This give the duration that the station was running until a stoppage occurs, thus Mean Time To Failure can then be calculated. Likewise, the number of seconds that the station was down is recorded which is then used to calculate the Mean Time to Repair. Hence from Table 2, we can see that between 07:00:00 and 07:10:00, the

Table 2
Sample MTBF and MTTR for P_Feeder Station.

Station #	Time	Cumulative Uptime (Sec)	Uptime/Min (Sec)	Interval -MTTF (Min)	Duration -MTTR (Min)	Interval -MTBF (Min)
1	7:01:00	55	55		5	
1	7:02:00	115	60	115		120
1	7:03:00	115	0			
1	7:04:00	115	0			
1	7:05:00	155	40		140	
1	7:06:00	215	60	100		240
1	7:07:00	215	0			
1	7:08:00	255	40		80	
1	7:09:00	315	60			
1	7:10:00	375	60			
1	7:11:00	400	25	185		265
1	7:12:00	400	0			
1	7:13:00	400	0			
1	7:14:00	450	50		165	
1	7:15:00	490	40	90		255
1	7:16:00	490	0			

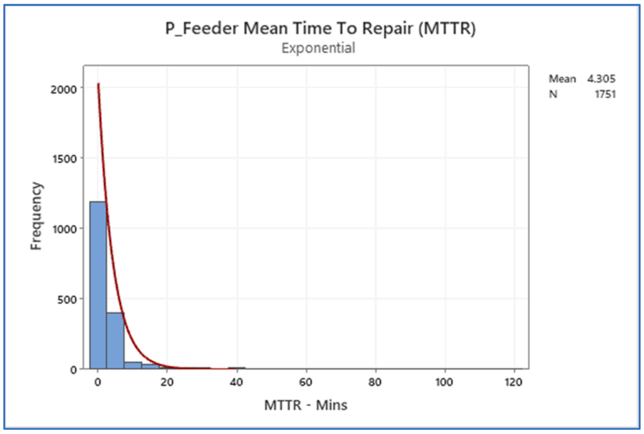

Fig. 6. P_Feeder MTBF Grubbs Outlier Test.

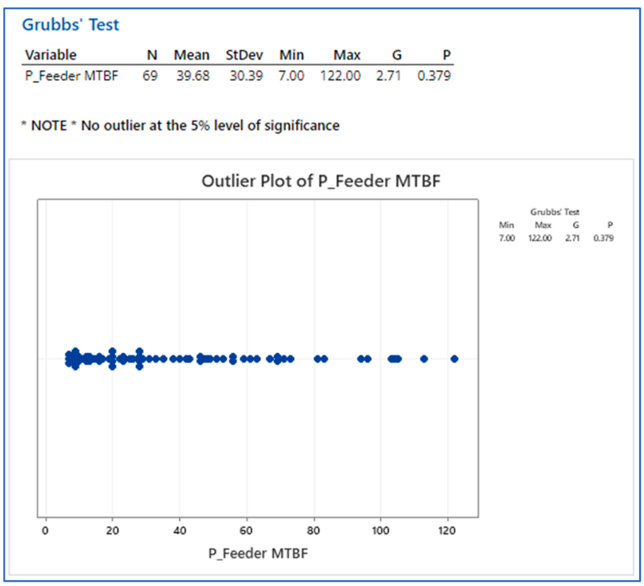

Fig. 7. P_Feeder MTBF Distribution Plot.

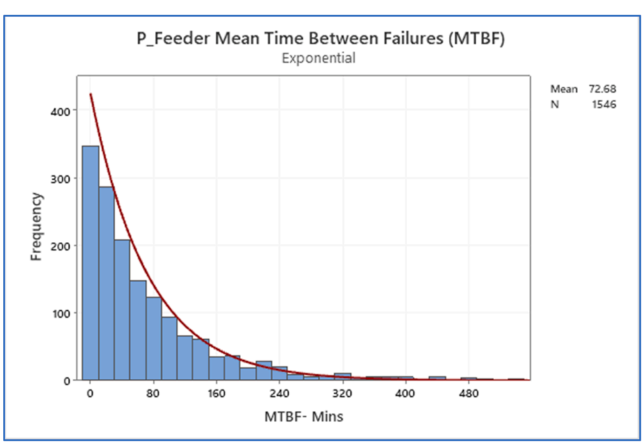

Fig. 8. P_Feeder MTTR Distribution Plot.

station was down for 5 s, running for 115 s, down for 140 s, running for 100 s and down for 80 s. MTBF is calculated as the sum of MTTF and MTTR.

The durations/frequencies that this station was running will be used to calculate MTBF distribution, similarly the durations/frequencies that this station was down will be used to calculate MTTR distribution. This whole process is repeated for every station used in the Tray Loader JaamSim model over a period of 129,000 min. The Interval (MTBF) and Duration (MTTR) data just described above is determined. Excel macros were developed to automatically determine the MTTF, MTTR and MTBF data in Table 2. This data is then entered into Minitab where outliers are removed (See Fig. 6).

A Minitab outlier test is performed on the data using the Grubbs Outlier method (See Fig. 6), where statistical outliers are removed first, then followed by any special cause outliers such as planned downtime activities. Once outliers were removed, the Grubbs outlier test indicated that there was no outlier MTBF data at the 5% level of significance. The same process was repeated for the P_Feeder MTTR data. A best fitting distribution identification process is then completed on both the MTBF and MTTR data again using Minitab. This analysis indicates that an Exponential distribution was the most appropriate fit for the *P_Feeder* MTBF data, with a mean (μ) of 72.68 Mins. See Fig. 7 for Exponential distribution plot of the *P_Feeder* MTBF data.

Likewise, a similar process is repeated for the *P_Feeder* MTTR data, where the data is entered into Minitab, outliers removed, and the most appropriate distribution selected. As can be seen in Fig. 8, the Exponential distribution with mean (μ) of 4.305 min is the best fit to the *P_Feeder* MTTR data.

The above process outlined for calculating the actual distributions and parameters for *P_Feeder* MTBF and MTTR is repeated for the remaining 7 stations used in the Tray Loader simulation model.

3.3. Running the tray loader digital model

The digital model of the Tray Loader was set-up with an initial warm-up period of 8hrs. This enables the Tray Loader model to completely fill with product before the actual simulation run starts. Each simulation run is set to 12 hrs, to replicate the 12hr shift that is used to operate the actual Tray Loader system. A simulation test run of 5000 replications was executed. The throughput data from each workstation along with the reliability data and tray buffer data from the 5000 simulation replications was written to a.csv file. The results from the simulation run were then compared with data from the actual Tray Loader system over an extended time period. The mean(μ) and standard deviation(σ) from the simulation results and actual line data are statistically compared to each other to confirm that the simulation model is a true representation of the actual Tray Loader system. This statistical analysis is completed for the P_Feeder workstation (both throughput and reliability) using a hypothesis tests known as a two-sample t-test, See Table 3 for results.

The summary statistics taken from the comparison of the actual *P_Feeder* throughputs and the simulation results are provided in Table 3 along with all the station reliability data for actual and simulated results. The maximum prediction error for all the mean data is 0.2% and the p-values for all the data are significantly greater than 0.05. The prediction error for the standard deviation on *P_Feeder* throughput is 7.72% which is expected, as the digital model has less variability/noise. This is not a significant factor in the model development and can be explained that some additional variability/noise occurs on the actual line due to human interactions, external factors such as materials availability, facility/utility system downtimes which have not been included in the model generation. Based on the analysis of the simulation model and the high level of accuracy with the empirical data gathered from the actual production line, the approach gives a high degree of confidence that the Tray Loader digital model is valid, accurately represents the real physical and operational production environment and provides a solid basis for the further use of this digital model.

4. Metamodel development

Metamodeling is the name commonly given to the practice of using a

Table 3
Verification data of Tray Loader Digital Model.

Station Name	Data Type	Data Source	Mean (μ)	Prediction Error	2-Sample t test P Value	Null Hypothesis	Null Hypothesis Result
P_Feeder	Throughput	Simulation (μ)	322,733	-0.20%	0.609	H_o: $pf_{savg} - pf_{aavg} = 0$	Accept
		Actual (μ)	322,096				
		Simulation (σ)	12,401	7.72%	0.281	H_o: $pf_{sstd} - pf_{astd} = 0$	Accept
		Actual (σ)	13,438				
P_Feeder	Reliability	Simulation	0.93862	-0.10%	0.352	H_o: $pf_{savg} - pf_{aavg} = 0$	Accept
		Actual	0.93769				

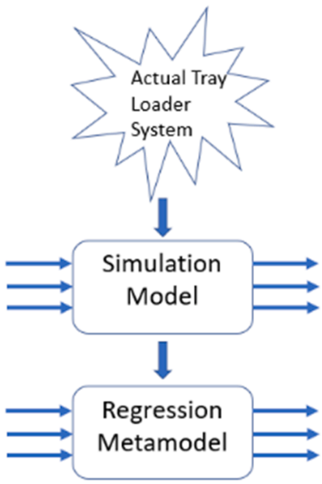

Fig. 9. Metamodel, Simulation Model and Problem Entity [24].

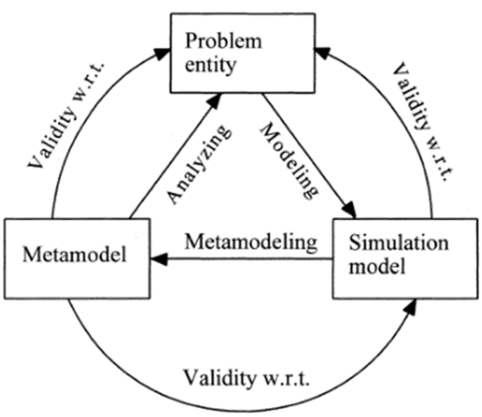

Fig. 10. Regression Metamodeling.

model to describe another model, which in this instance, is the JaamSim Tray Loader model. Metamodeling (literally, "beyond Modelling") is the Modelling of models [23]. A simulation model is a representation of a real word system, whereas the term meta-model referred to herein is a mathematical approximation of a simulation model. [24]. The relationships among a metamodel, simulation model, and problem entity are shown in Fig. 9 [24], where w.r.t. means `with respect to'. In this research the problem entity is the actual Tray Loader system that is being modelled. A simulation model can be a Discrete Event Simulation model of the problem entity, this model may be deterministic or stochastic. A metamodel approximates the input/output (I/O) transformation that is implied by the simulation model, the resulting black-box model is known as a response surface. There are different types of metamodels, e.g.: polynomial regression models (which are a type of linear regression) and neural networks (a type of non-linear regression) [25].

According to [26] the four (4) reasons to develop and use a regression metamodel are as follows:

1. Highlight the significant design factors.
2. Show the interactions between the various factors.
3. Determine their relative importance.
4. Quantify their effects on the output performance.

As outlined a metamodel is a regression model of another model. Thus, a regression metamodel is a model of the simulation model itself built using regression analysis [27]. As can be seen in Fig. 10, the user develops a well-structured simulation model and the relationship between the inputs and the outputs of this simulation model is in turn modelled in an analytical form by the regression metamodel [28].

Typically, if we assume that the regression metamodel is linear, it belongs to one of the following three classes:

1. First-order polynomial, which consists of main effects only, besides an overall or grand mean.
2. First-order polynomial augmented with interactions between pairs of factors,
3. Second-order polynomial, which includes purely quadratic effects.

If the response (P_Feeder Output) is well modelled by a linear function of the independent variables, then the approximating function is the first-order model as per Eq. (1).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + e \tag{1}$$

First Order Regression Metamodel.

If the statistical tests on the first-order meta-model are not acceptable or satisfactory, a second-order meta-model is then chosen, the approximating function is shown by (Eq. 2).

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum_{i<j} \sum \beta_{ij} x_i x_j + e \tag{2}$$

Second Order Regression Metamodelwhere β_o = regression intercept. β_i =main or first-order effect of factor i. x_i =value of the factor i. β_{ii}=quadratic effect of factor i. βij = interaction between the factor i and j (i ‡ j), e=fitting error of the regression model. n = the number of factors.

The Equipment Design/Manufacturing Engineer are very interested in answering the following questions when designing new equipment or deciding on where to make improvements to existing equipment to maximize the throughput. These four (4) questions include the following:

1. How sensitive is the line output to changes in the design factors?

2. Which of these factors and which of the potential interactions between them explain the variability in the line output?

3. What could be expected if a factor is increased/decreased by one unit or percent?

4. Where should their efforts be concentrated first, in-order to improve line output performance?

To address these questions, it is proposed to generate a Metamodel of the simulation model, then use sensitivity analysis with this model to provide useful insights and help the engineers to gain a better

Table 4
Metamodel DOE Factors and Settings.

Factor #	Factor Name	Factor Type	# of Levels	High Setting	Centre Setting	Low Setting
1	P_Feeder	Reliability	2	98	94	90
2	T_Pack	Reliability	2	98	94	90
3	Tray Stack	Reliability	2	98	94	90
4	Process3	Reliability	2	95	90	85
5	Process4	Reliability	2	95	90	85
6	Process 5	Reliability	2	99	95	91
7	P_Feeder	Cycle Time	2	1.10	0.95	0.80
8	Process4 Process5	Cycle Time	2	3.20	2.70	2.20
9	Tray Count	Reliability	2	120	100	80
10	P_Feeder_Yield	Yield	2	95	90	85

Table 5
Linear Model Analysis of Variance for Response.

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares	F-Value	P-Value
Regression	10	1.03E+ 11	1.03E+ 10	44.21	0
P_Feeder Rel	1	3.02E+ 09	3.02E+ 09	12.91	0.001
T_Pack Rel	1	7.58E+ 08	7.58E+ 08	3.24	0.078
T_Stack Rel	1	6.59E+ 08	6.59E+ 08	2.82	0.099
Process 3 Rel	1	2.34E+ 08	2.34E+ 08	1	0.321
Process 4 Rel	1	2.5E+ 08	2.5E+ 08	1.07	0.306
P_Feeder CT	1	7.65E+ 10	7.65E+ 10	327.36	0
Process 4/5 CT	1	8.21E+ 08	8.21E+ 08	3.51	0.067
Tray Count	1	2.7E+ 09	2.7E+ 09	11.56	0.001
P_Feeder Yield	1	6.64E+ 09	6.64E+ 09	28.41	0
Process 5 Rel	1	5.12E+ 08	5.12E+ 08	2.19	0.145
Error	51	1.19E+ 10	2.34E+ 08		
Lack-of-Fit	45	1.19E+ 10	2.65E+ 08	939.88	0
Pure Error	6	1,690,653	2,81,776		
Total	61	1.15E+ 11			
R^2	89.60%				

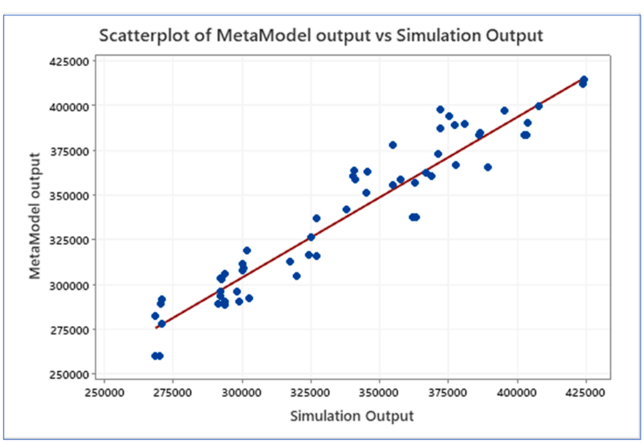

Fig. 11. Simulation Output vs Metamodel Output (1st Order).

understanding on what the impact is of changing the various factor settings on the overall behaviour of the Tray Loader system. Sensitivity analysis is a method to estimate (quantify) how a change in an input factor of a system affects an output performance measure, i.e., how sensitive an output is to change in an input [29].

4.1. Tray loader metamodel development methodology

[24] describes an approach to develop a metamodel. A refined 8 step methodology to build a Tray loader metamodel is described as follows:

1. Define the problem. What is the issue to be solved and/or response variable to be modelled (*P_Feeder* Output), factors and buffers to be considered?
2. Define the range of factors (factor settings used and buffer sizes).
3. Build the simulation model and complete verification and validation of the model.
4. Specify the form of the metamodel. This is either a first order (Linear) or second order regression model.
5. Develop the Design of Experiment base design where response values are recorded for the various factor settings. The simulation model is used to generate the response variable data based on the factor settings.
6. Develop the regression metamodel. The metamodel to predict the response variable is developed using the factor settings and output data from the simulation model.
7. Verify the metamodel is a valid model to predict the response (*P_Feeder* Output) based on the various factor settings.
8. If metamodel is not valid, then proceed back to step 4. Otherwise metamodel developed is sufficient to use.

A lack of fit test is used to determine if the model adequately fits the simulation model [30] and as such the actual tray loader system. The validity of the metamodel with respect to the simulation model is determined by examining the model fit diagnostics. A check of the distribution of the residuals leads to the determination of the validity of some of the model assumptions. Comparing the simulation output data and the metamodel output data is another method to validate the validity of the metamodel [26].

A metamodel is modified until it is valid for all experimental conditions tested, using the specified validity measures and their required values. To decide whether to accept a specific metamodel, a useful criterion is to determine the absolute error between the simulation model and the metamodel. The absolute error (AE) is given by the Eq. 3.

$$AE = \left| \frac{Metamodel \quad Output - Simulation \quad Output}{Simulation \quad Output} \right| \quad x \quad 100 \tag{3}$$

Metamodel Absolute Error.

4.2. Tray loader metamodel development

The purpose of developing a metamodel is to allow the design/manufacturing engineer to quickly predict what impact a minor or major change to the Tray Loader system will have on the throughput from the line (P_Feeder Output). As an example, if an engineer has a new design for Process Station 3, that will improve the reliability from 93% to 95%, while at the same time reduce the cycle time by 10%, whilst increasing the Empty Tray buffer storage by 10 trays. They need to quickly determine what the impact of these changes individually and all together has on the overall output from the line. The use of a Metamodel will allow them to quickly predict the overall improvement from the line because of implementing those changes. A critical review of the Actual Tray loader system and the simulation model by subject matter experts was completed to determine the most critical factors and levels that were important in developing a metamodel of overall Tray Loader system. The review resulted in 9 factors each with 2 levels being selected and to be used for the Metamodel DOE. The factors and levels are shown in Table 4. The range in the high and low setting levels for each factor were selected to allow the designer to test for significant changes to the reliability, cycle time and buffer size for the various stations on the Tray loader when using the metamodel. The resultant metamodel could then more accurately predict the *P_Feeder* output over a larger range of factor settings. As such if a design engineer needs to test the impact of improving the reliability of a particular station from 92% to 94%, then he/she could easily determine that impact using this metamodel, as those settings are within the range of factor settings used for the

Table 6
Tray Loader 2nd Order Regression Metamodel.

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares	F-Value	P-Value
Regression	51	1.25E+ 11	2.45E+ 09	1422.01	0
P_Feeder Rel	1	8.4E+ 08	8.4E+ 08	487.26	0
T_Pack Rel	1	63,901,092	63,901,092	37.06	0
T_Stack Rel	1	1.45E+ 08	1.45E+ 08	84.15	0
Process 3 Rel	1	7.19E+ 08	7.19E+ 08	416.76	0
Process 4 Rel	1	7.08E+ 08	7.08E+ 08	410.69	0
P_Feeder CT	1	8.05E+ 08	8.05E+ 08	466.96	0
Process 4/5 CT	1	2.61E+ 08	2.61E+ 08	151.33	0
Tray Count	1	5.05E+ 08	5.05E+ 08	293.14	0
P_Feeder Yield	1	1.22E+ 08	1.22E+ 08	70.56	0
Process 5 Rel	1	2.42E+ 08	2.42E+ 08	140.46	0
P_Feeder CT*P_Feeder Rel	1	5.22E+ 08	5.22E+ 08	302.8	0
P_Feeder CT*Tray Count	1	1.11E+ 09	1.11E+ 09	643.7	0
P_Feeder CT*P_Feeder CT	1	1.48E+ 09	1.48E+ 09	858.61	0
P_Feeder CT*T_Pack Rel	1	2.47E+ 08	2.47E+ 08	143.12	0
P_Feeder CT*T_Stack Rel	1	3.59E+ 08	3.59E+ 08	208.04	0
P_Feeder CT*Process 3 Rel	1	3.82E+ 08	3.82E+ 08	221.71	0
P_Feeder CT*Process 5 Rel	1	5.65E+ 08	5.65E+ 08	327.91	0
P_Feeder Yield*P_Feeder Rel	1	2.82E+ 08	2.82E+ 08	163.43	0
P_Feeder Yield*Tray Count	1	2.95E+ 08	2.95E+ 08	170.88	0
P_Feeder Yield*T_Stack Rel	1	3.25E+ 08	3.25E+ 08	188.47	0
P_Feeder Yield*Process 3 Rel	1	8.98E+ 08	8.98E+ 08	520.71	0
P_Feeder Yield*Process 4/5 CT	1	3.49E+ 08	3.49E+ 08	202.14	0
P_Feeder Yield*Process 5 Rel	1	4.6E+ 08	4.6E+ 08	266.88	0
P_Feeder Rel*T_Pack Rel	1	4.18E+ 08	4.18E+ 08	242.64	0
P_Feeder Rel*T_Stack Rel	1	2.98E+ 08	2.98E+ 08	172.69	0
P_Feeder Rel*Process 4 Rel	1	4.66E+ 08	4.66E+ 08	269.99	0
P_Feeder Rel*Process 3 Rel	1	7.54E+ 08	7.54E+ 08	437.55	0
P_Feeder Rel*Process 4/5 CT	1	4.63E+ 08	4.63E+ 08	268.41	0
P_Feeder Rel*Process 5 Rel	1	73,57,013	73,657,013	42.72	0
Tray Count*T_Pack Rel	1	1.15E+ 08	1.15E+ 08	66.55	0
Tray Count*T_Stack Rel	1	6.4E+ 08	6.4E+ 08	370.91	0
Tray Count*Process 3 Rel	1	5.2E+ 08	5.2E+ 08	301.34	0
Tray Count*Process 4 Rel	1	5.45E+ 08	5.45E+ 08	315.89	0
Tray Count*Process 5 Rel	1	1.15E+ 09	1.15E+ 09	666.52	0
Tray Count*Process 4/5 CT	1	43,715,524	43,715,524	25.35	0
T_Pack Rel*T_Stack Rel	1	4.5E+ 08	4.5E+ 08	261.07	0
T_Pack Rel*Process 4 Rel	1	7.57E+ 08	7.57E+ 08	439.25	0
T_Pack Rel*Process 3 Rel	1	51,762,376	51,762,376	30.02	0
T_Pack Rel*Process 4/5 CT	1	1.29E+ 08	1.29E+ 08	74.82	0
T_Pack Rel*Process 5 Rel	1	1.25E+ 08	1.25E+ 08	72.39	0
T_Stack Rel*Process 4 Rel	1	1.47E+ 08	1.47E+ 08	85.3	0
T_Stack Rel*Process 3 Rel	1	16,297,646	16,297,646	9.45	0.006
T_Stack Rel*Process 4/5 CT	1	5.52E+ 08	5.52E+ 08	320.29	0
T_Stack Rel*Process 5 Rel	1	3.36E+ 08	3.36E+ 08	195.02	0
Process 4 Rel*Process 3 Rel	1	5.29E+ 08	5.29E+ 08	306.56	0
Process 4 Rel*Process 4/5 CT	1	6.52E+ 08	6.52E+ 08	377.98	0
Process 4 Rel*Process 5 Rel	1	7.2E+ 08	7.2E+ 08	417.57	0
Process 3 Rel*Process 4/5 CT	1	5.61E+ 08	5.61E+ 08	325.5	0
Process 3 Rel*Process 5 Rel	1	2.46E+ 08	2.46E+ 08	142.42	0
Process 4/5 CT*Process 5 Rel	1	4.68E+ 08	4.68E+ 08	271.47	0
T_Stack Rel*T_Stack Rel	1	1.06E+ 08	1.06E+ 08	61.21	0
Error	20	34,484,462	1,724,223		
Lack-of-Fit	14	32,793,809	2,342,415	8.31	0.008
Pure Error	6	1,690,653	281,776		
Total	71	1.25E+ 11			
$\mathbf{R^2}$	99.97 %				

metamodel development.

A Design of Experiments (DOE) was generated (Minitab), using the Plackett-Burman design resulting in a 62-run factorial design. This design consisted of 48 runs as the base design, 5 runs as centre points and an additional 9 runs to replicate a push and pull type production system. The base design explored the factor setting at the extremes (High and Low Settings), 5 runs with all the factor setting at approximately the midpoint of the upper and lower setting and the remaining 9 runs with different combination of settings. The Plackett-Burman Design is used to create a designed experiment to identify the most important factors early in the experimentation process. Plackett-Burman designs can fit 2 – 47 factors with each having 2 levels. Normally a Plackett-Burman design is used when there are 8 or more factors and where the critical factors need to be identified [31]. Each of the experiments was run using the validated Tray Loader Jaam Sim (TLJSim) digital model. Each experiment (62 in total) is executed with 300 replications being completed, average is taken over the 300 replications and this value then used as the response for that experiment. The simulation runs are performed with a simulated 8 hr warm up period followed by a 12 hr production time-period on a PC with an I5 Core CPU (1.60 GHz) and 16 GB of RAM, each run taking approximately 20 min to execute. As per the methodology presented in Section 4.1 (Step 4), a regression meta-model considering only the main effects of the factors is developed using Minitab. The 1st order linear regression model developed is given in Eq. 4:

$$y = -68560 + 1885x_1 + 940x_2 + 881x_3 + 417x_4 - 429x_5 + 768x_6 - 247167x_7 - 7730x_8 + 364x_9 + 2280x_{10} \quad (4)$$

Linear Regression MetamodelWhere: y = P_Feeder Output (Response variable). x_1 =P_Feeder Reliability. x_2 =T_Pack Reliability. x_3 =T_Stack Reliability. x_4 =Process3 Reliability. x_5 =Process4 Reliability. x_6 =Process5 Reliability. x_7 =P_Feeder Cycle Time. x_8 =Process 4/5 Cycle Time. x_9 =Tray Count. x_{10} =P_Feeder Yield.

Analysis of Variance (ANOVA) for the response (considering the main effects) is presented in Table 5. As can be seen from the R2 value, 89.6% of the variation can be explained by this linear model that was developed.

In order to test the validity of this linear regression metamodel, we use the approach suggested by [26]. The factor settings used in the 62 experimental runs were used as inputs to the linear metamodel. The simulation response output (P_Feeder Output) was then compared to the predicted values for these combinations using Eq. 4. A scatterplot comparing the simulation output to the metamodel output are shown in Fig. 11. The absolute error between the simulated output and the metamodel output was calculated using Eq. 3. The average absolute error between the simulation output and the metamodel output over these 62 runs was calculated as 3.42%.

Due to the R2 value of 89.6% (the variation not explained by the model being > 10%) along with the higher than absolute error of 3.42% (Simulation output vs Metamodel output), it was determined that this linear metamodel was not satisfactory to use as a tool to predict the *P_Feeder* output based on varying the input factor settings. As such, based on our objectives, the results of using this 1st order linear metamodel as shown in Eq. 4 is not appropriate for use. Consequently, a new 2nd order regression metamodel is evaluated considering both the main effects of the factors and their two-way interactions. See Eq. 5 for the new 2nd order regression metamodel developed for the tray loader system.

2nd Order Tray Loader Regression MetamodelWhere: y = P_Feeder Output (Response variable). x_1 = P_Feeder Reliability. x_2 = T_Pack Reliability. x_3 = T_Stack Reliability. x_4 = Process3 Reliability. x_5 = Process4 Reliability. x_6 = Process5 Reliability. x_7 = P_Feeder Cycle Time. x_8 = Process 4/5 Cycle Time. x_9 = Tray Count. x_{10} = P_Feeder Yield.

Again, an Analysis of Variance (ANOVA) for the response (considering the main effects and interactions) is presented in table 7.4. As can be seen from the R2 value, 97.91% of the variation can be explained by this 2nd order model that was developed. During development of the 2nd order model all factor interactions with a P value > 0.10 were removed from the regression model. Fisher's F value in Table 6 for regression demonstrates a high significance for the regression model (F =1422 and P = 0.0). An F-test is a statistical test in which the test statistic has an F-distribution under the null hypothesis. The F statistic is a value that is obtained when an ANOVA test or a regression analysis is carried out to determine if the means between two populations are significantly different. It is most often used when comparing statistical models that have been fitted to a data set, to identify the model that best fits the population from which the data were sampled. The larger the F value, the more important it is, that this factor influenced the response variable.

The validity of this 2nd order regression metamodel is again confirmed. The factor settings used in the 62 experimental runs are used as inputs to the 2nd order metamodel. The simulation response output (*P_Feeder* Output) is then compared to the predicted values for these combinations using Eq. 5. The absolute error between the simulated output and the metamodel output is calculated using Eq. 3. A scatterplot comparing the simulation output to the metamodel output, see Fig. 12

$$\begin{aligned} y = & -512101 + 196672x_1 - 29360x_2 - 154089x_3 + 197641x_4 - 152691x_5 - 75371x_6 + 6031262x_7 - 807672x_8 - 24026x_9 - 25751x_{10} - 21556x_1x_7 - 3006x_7x_9 \\ & + 287.9x_6x_9 - 18827x_2x_7 + 11440x_3x_7 + 14610x_4x_7 - 14058x_6x_7 - 502.6x_1x_{10} + 57.63x_9x_{10} + 386.7x_3x_{10} - 851.9x_4x_{10} + 3495x_8x_{10} \\ & + 521x_6x_{10} - 398.1x_1x_2 - 393.8x_1x_3 + 439.5x_1x_5 - 737.9x_1x_4 - 3225x_1x_8 - 218.8x_1x_6 + 48.44x_2x_9 + 223.7x_3x_9 - 137.83x_5x_9 - 208.7x_4x_9 \\ & + 196.3x_8x_9 + 384.5x_2x_3 + 879.5x_2x_5 - 131.8x_2x_4 + 1575x_2x_8 - 272.2x_2x_6 + 268.9x_3x_5 - 52.0x_3x_4 + 10939x_3x_8 - 776.8x_3x_6 - 521.4x_4x_5 - 8398x_5x_8 \\ & + 891.6x_5x_6 - 4261x_4x_8 + 359.8x_4x_6 + 7633x_6x_8 + 579.4x_3^2 - 1709868x_7^2 \end{aligned} \quad (5)$$

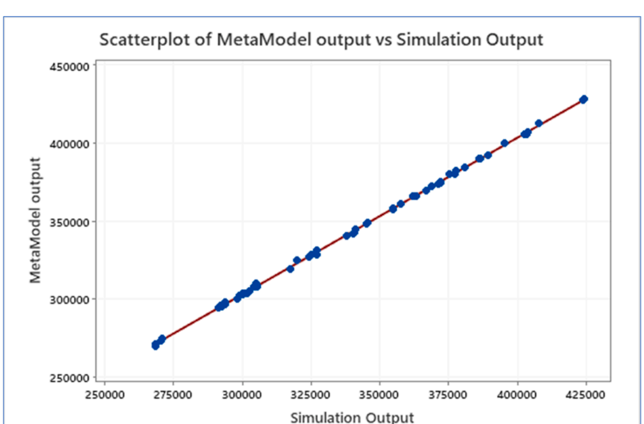

Fig. 12. Simulation Output vs Metamodel Output (2nd Order).

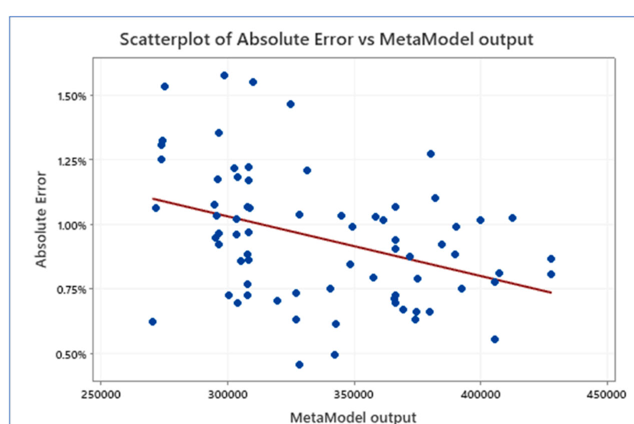

Fig. 13. Absolute Error (2nd Order).

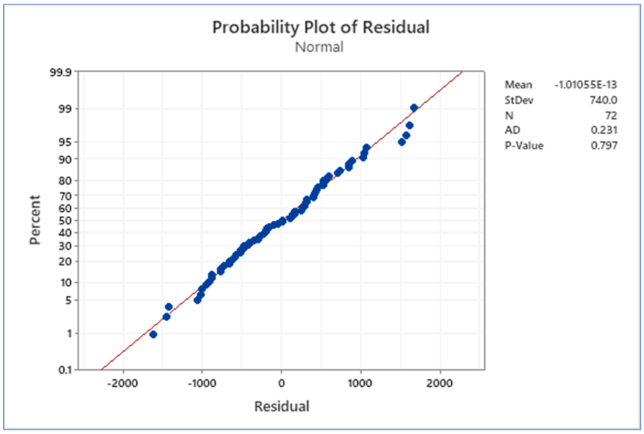

Fig. 14. Residual Probability Plot.

along with the absolute error are shown in Fig. 13. The average absolute error between the simulation output and the metamodel output over these 62 runs is calculated as 0.93%. This 2nd order regression metamodel should mainly be used to predict *P_Feeder* Output (y response) for values of the factor settings (x variables) that fall within the range of the x variables in the experimental data set. Any predictions of the *P_Feeder* Output (y response) based on values of the x variables that are significantly outside these values may lead to inaccurate response values.

Based on the high R^2 value of 99.97%, average absolute error of 0.93% and F test for the regression model at 1422, we can conclusively conclude that this metamodel is a very good representation of the data generated by the simulation model and as such the actual tray loader system. It can then be recommended that this metamodel be used to predict *P_Feeder* Output based on the associated factor settings. A normality test for the residuals (difference between simulation model and metamodel) was performed using the Anderson-Darling normality test, with results shown in Fig. 14. As can be seen from Fig. 14, the residual data set has a P = 0.797 and AD = 0.231 indicating a very good fit of the residual data set to a normal distribution (> 95% confidence level). The P and AD values indicates that the residual normality about the mean is another good model fit diagnostic.

As per step 7 of the 8 step metamodel development process (section 4.1) a sensitivity analysis was then completed, whereby a set of 10 experiments with different input factor settings are used as inputs to both the simulation model and metamodel equation. Each experiment (10 in total) is run using the Tray Loader JaamSim model with 300 replications being completed, *P_Feeder* output averages taken over those 300 replications and this result being used as the simulation model output response for that experiment. The factor settings are used as inputs to the regression metamodel developed, Eq. 5 and the resultant Metamodel response for *P_Feeder* output for each experiment is shown in Table 7. The absolute error between the simulation model and the metamodel is calculated for each experiment using Eq. 3, see Table 7. The overall average absolute error across the 10 runs is 0.87% when using the Average simulation Output (over 300 replications per experiment) compared to the Metamodel Output.

A scatterplot comparing the simulation output to the metamodel output (Fig. 15) along with the absolute error is shown in Fig. 16. As can be seen from these graphs, the metamodel is a close representation of the simulation model and as such the actual tray loader system, provided the factor setting are within the ranges used for the metamodel development.

5. Conclusions

As manufacturing capital equipment is expensive, it is necessary that the equipment once in operation is reliable and delivers to the business plan targets. Simulation along with Metamodeling is an invaluable tool to confirm that an automated manufacturing line can produce to the required business objectives before and after it goes into operation. Implementing the actual changes to equipment to improve reliability can be both time consuming and expensive. This paper has shown with a systematic approach, analysis and rigour, how a digital/simulation model of an actual industrial use case (Tray Loader system) was created, verified and validated for use with a high degree of confidence. The

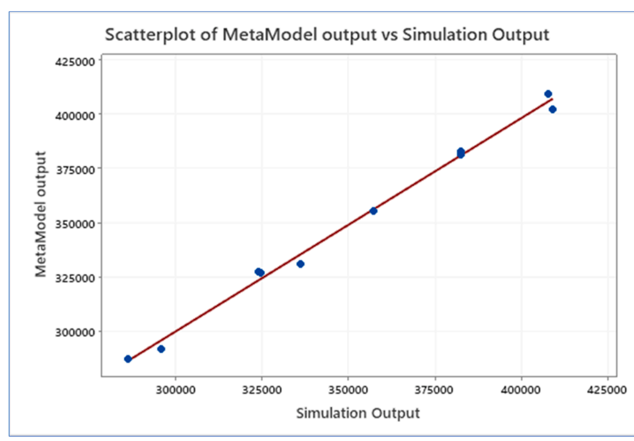

Fig. 15. Sensitivity Analysis of Simulation Model vs Metamodel.

Table 7
Tray Loader 2nd Order Regression Metamodel.

	Input Factor Settings										Output Responses		
Run #	P_Feeder Rel%	T_Pack Rel %	T_Stack Rel %	Process 3 Rel %	Process 4 Rel %	Process 5 Rel %	P_Feeder CT (Sec)	Process 4/5 CT (Sec)	Tray Count	P_Feeder Yield %	Simulation Model Avg Output	MetaModel output	Absolute Error
1	93	94	94	86	93	94	0.9	2.9	105	90	3,57,211	3,55,555	0.46%
2	96	97	97	92	95	97	0.8	2.5	110	88	4,07,740	4,03,299	1.09%
3	95	96	98	92	89	95	1.1	2.6	100	86	2,86,017	2,87,433	0.50%
4	95	91	94	88	93	95	1	2.5	105	87	3,24,666	3,27,119	0.76%
5	92	92	94	86	87	90	0.8	2.2	120	95	3,82,515	3,82,708	0.05%
6	94	93	93	90	94	96	1	2.7	100	85	3,23,913	3,27,613	1.14%
7	92	93	92	84	88	95	1	2.3	120	88	3,36,134	3,31,223	1.46%
8	92	90	95	92	95	97.53	1.1	2.5	110	91	2,95,793	2,91,998	1.28%
9	98	97	98	92	90	96	0.9	2.3	110	91	3,82,459	3,81,280	0.31%
10	95	96	95	89	91	94	0.8	2.2	120	90	4,09,094	4,02,498	1.61%
												Average Error	0.87%
												Standard Dev	0.0053

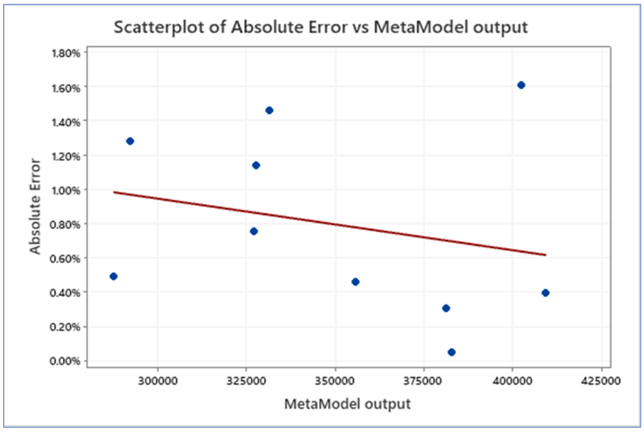

Fig. 16. Absolute Error from Sensitivity Analysis Study.

author has shown how a metamodel can be used to accurately predict the throughput, including OEE (using cycle time, reliability and yield data) from an automated medical devices lines that operates in a stochastic behaviour. The author has also shown how a 2nd order metamodel was created that accurately represents the behaviour of both the digital model and the actual system. This metamodel can be used to rapidly determine the performance impact of the line while changing one or more line design/factor parameters. As these medical device automated manufacturing lines can cost in excess of $50 mm each, it is critical that the line is designed to be capable of achieving these business performance objectives once in operation. Simulation and metamodeling can thus be used to verify the line design before it is built to ensure these business objectives will be achieved. The detailed methodology taken by the author in developing, verifying/validating the digital model along with the development and testing of the metamodel of an automated medical devices manufacturing line can be fully used for other types of manufacturing systems. These technologies can be a subset of an overall digital manufacturing system that enables the optimization of a manufacturing line during the line design stage or when the line is put into operation. The use of this technology gives a deeper understanding of what can occur on the manufacturing line when it is running. A simulation model and/or Metamodel when combined with optimization engine, can be used to identify problems before they occur and aid in the selection of optimum parameters to run the line before it is fully designed or built. Digital model and Metamodeling technologies supports other Industry 4.0 technologies such as predictive maintenance, OEE improvement, waste reduction, improve batch changeover times and to improve product quality [32]. It allows for efficient design and development, linking 3D models with simulation and emulation of equipment control code. In addition, having a digital model enables virtual line analysis, removing the physical restraints of expert engineers having to be on your location [33]. The author has demonstrated how the development of digital model can be validated and subsequently used for the development of a Metamodel which is then used for the study of equipment design, maintenance and reliability of an automated manufacturing line in the medical devices industry. This paper contributes to the body of knowledge by providing a framework that assists other practitioners in the development of a digital model and metamodel of an automated manufacturing line with a view to maximizing the performance (OEE) of the actual system. Further research can be done in integrating the metamodel to a Digital Twin and an optimization system to aid the user in determining the optimum line conditions to maximize (OEE) and/or minimize line downtime. There is a need for an overall architecture to enable fast simulation execution speed when enabling a Digital Twin and this could be achieved by the use of a metamodel that is rigourly developed and tested.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Klos S, Patalas J, Trebuna y P. Improving manufacturing processes using simulation methods. Appl Comput Sci 2016;12(nº 4):7–17.
[2] G. Shao, C. Laroque, P. Lendermann, S. Jain, L. Hay Lee, Y.O. Rose, “Digital Twin for Smart Manufacturing: The Simulation Aspect”, 2019.
[3] Malik A, A YB. Digital twins for collaborative robots. Robot Comput-Integr Manuf, nº 2020;68.
[4] R.F.C. Stark, Y.K. Lindow, “Development and Operation of Digital Twins for Technical Systems and Services”, 2019.
[5] W. Wong, “What’s the Difference Between a Simulation and a Digital Twin?”, 2018. [En línea]. Available: ⟨https://www.electronicdesign.com/technologies/embedded-revolution/article/21806550/whats-the-difference-between-a-simulation-and-a-digital-twin⟩.
[6] W. Kritzinger, M. Karner, G. Traar, J. Henjes, Y.W. Sihn, “Digital Twin in Manufacturing: A Categorical Literature Review and Classification”, de *IFAC Conference Papers*, 2018.
[7] Van Beers W, Kleijnen YJP. Customized sequential designs for random simulation experiments: kriging metamodelling and bootstrapping. Eur J Oper Res 2008;186:1099–113.
[8] S. Boschert, Y.R. Rosen, Digital Twin—The Simulation Aspect, 2016.
[9] Nakajima S. Introduction to TPM. *Cambridge, MA*: Productivity Press; 1988.
[10] Tajiri M a GF. TPM implementation-a Japanese approach. McGraw-Hil,; 1992.
[11] Grzegorz Gołda AKIP. Modeling and Simulation of Manufacturing Line Improvement. Int J Comput Eng Res (IJCER) 2016:26–30.
[12] Thoma K, Scharte B, Hiller y D. Resilience engineering as part of security research: definitions, concepts and science approaches. Eur J Secur Res 2016:3–19.
[13] K. Rosita, Y.M. Rada, “Equipment Reliability Optimization Using Predictive Reliability Centered Maintenance”, *IEEE 8th International Conference on Industrial Engineering and Applications*, pp. 348 - 354, 2021.
[14] Forsthoffer M. Reliability optimization. Forsthoffer’s More Best Practices for Rotating Equipment. Butterworth-Heinemann,; 2017.
[15] Y. Prasetyo, Y.K. Rosita, “Equipment Reliability Optimization Using Predictive Reliability Centered Maintenance”, 2020.
[16] Roy R, Stark R, Tracht K, Takata S, Mori YM. Continuous maintenance and the future – foundations and technological challenges. CIRP Ann - Manuf Technol 2016;65:667–88.
[17] Pancholi N, Bhatt YD. Performance reliability improvement by optimising maintenance practices through failure analysis in process industry. IOSR J Mech Civ Eng 2016;13(nº 6):66–73.
[18] Jawalkar CS. Introduction to Basic Manufacturing. Alpha Science International Ltd; 2016.
[19] Ferreira W, Armellini F, Eulalia YL. Simulation in Industry 4.0: A state of the art review. Comput Ind Eng 2021.
[20] Davis PK. Generalizing concepts and methods of verification, validation and accreditation (VV&A) for military simulations. Natl Def Res Inst 1992:5–6.
[21] A.M. Law, “How to Build Valid and Credible Simulation Models”, 2019.
[22] Kleijnen JP. Theory and methodology of verification and validation of simulation models. Eur J Oper Res 1995;82:145–62.
[23] J. Sprinkle, B. Rumpe, H. Vangheluwe, Y.G. Karsai, “Metamodelling State of the Art and Research Challenges”, ⟨https://www.researchgate.net/publication/265469303⟩, pp. 57 - 76, 2014.
[24] Kleijnen J, Sargent YR. A methodology for fitting and validating meta-models in simulation. Eur J Oper Res 2000;120:14–29.
[25] R.R. Barton, “Metamodelling: A State of the Art Review”, 1994.
[26] Durieux S, Pierreval YH. Regression metamodelling for the design of automated manufacturing system composed of parallel machines sharing a material handling resource. Int J Prod Econ 2004;89:21–30.
[27] Kleijnen J. An overview of the design and analysis of simulation experiments for sensitivity analysis. Eur J Oper Res 2005;164:287–300.
[28] Friedman L, Friedman YH. Statistical considerations in computer simulation: the state of the art. J Stat Comput Simul 1984;19:237–63.
[29] D. Kelton, “Designing Simulation Experiments”, 1999.
[30] Panis R, Tyers R, Houck YE. Combining regression diagnostics with simulation metamodels. Eur J Oper Res 1994;73:85–94.
[31] Plackett R, Burmann YJ. Design of optimal multifactorial experiments. Biometrika 1946;33:305–25.
[32] G. Shao, C. Laroque, S. Jain, L. Hay Lee, P. Lendermann, y O. Rose, “Digital Twin for Smart Manufacturing: The Simulation Aspect”, 2019.
[33] Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, Wang y L. Enabling technologies and tools for digital twin. J Manuf Syst 2021;58:3–21.