DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

Conceptual Platform for Basic Waterfall Manufacturing System Using Simulink with Lean-Agile Measurement Criteria

Maryam Khalid Abdul Wahid[1], Lamyaa Mohammad Dawood[2]
[1,2]Department of production Engineering & Metallurgy, University of Technology, Baghdad, Iraq
[1]pme.20.52@grade.uotechnology.edu.iq , [2]Lamyaa.m.dawood@uotechnology.edu.iq

Abstract—*This paper forms part I of two-part series that present the full path for constructing a Lean-Agile platform using Simulink and test the different criteria of such manufacturing systems. management of manufacturing systems is one of the main rules in engineering in order to reduce the waste (lean) and to fulfill the changing requirements of the market (agile). This paper describes the process of construction of the basic waterfall platform using Matlab/Simulink, and to develop the Matlab program that facilitates the input process of constants and variables into the model and then to show the results. this platform will mimic the manufacturing system of motorcycle engine that takes six parts as input and outputs four different models, Matlab/Simulink is used to construct the platform, and the library of Simscape used to model the discrete (not continuous) behavior of the system. Waterfall platform has been constructed. and simulated with the specific constants and variables from an excel file. Showing the results of workers-usage, over-processing, machine-usage,queue-usage. It shows the orders done with time of simulation, illustrating and indication of lean and agility of the proposed system. Different figures were generated and introduced to show various measuring criteria that evaluate the performance of the proposed system. The different measurement criteria drawn using a second Matlab program developed specially for this purpose.*

Index Terms— *Lean, agile, Leagile, manufacturing system, workshop*.

I. INTRODUCTION

The manufacturing companies faced intense competition, prompting them to implement changes like shortening product life cycles, cutting production costs, and addressing diverse customer demands. Numerous enterprises have adopted process enhancement approaches, including lean manufacturing (LM) and agile manufacturing (AM) systems, to meet customer requirements and enhance overall satisfaction. LM and AM, as production concepts, are increasingly becoming prevalent in the contemporary market [1] [2].

Lean and agile methodologies are also used to deal with technical dept (TD).[3], as the application of agile project management on the principles of library IT [4]. As A Framework Integrating Social and Technical Elements for the Application of Lean Project Management to Achieve Sustainable Value in the Context of Digital Transformation [5]. The management paradigms of lean and agility have become relevant across diverse industrial sectors and processes. While both LM and AM are distinctive, LM arises as a

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

reaction to resource constraints and competitive pressures, whereas AM addresses the challenges brought about by continuous change [6].

Numerous organizations are adopting lean production systems to enhance efficiency by identifying and eliminating time and cost-consuming activities that do not contribute to the value of a product. This implementation aims to expedite processes, minimize waste, and enhance overall quality and productivity. In contrast, agile manufacturing provides businesses with the capability to adjust swiftly to dynamic market conditions. Agility not only involves adapting to change but also leveraging market volatility to introduce new products, gaining a competitive advantage. Consequently, a flexible manufacturing system, accommodating changes in volume and product variety, becomes essential to achieve these production objectives [7] [8] [9].

The LM and AM paradigms, while distinct, complement each other. The term 'leagile' signifies a significant overlap in the content of both paradigms, as observed by Naylor et al. (1999) and other scholars. Lean-agile involves the simultaneous implementation of both lean and agile paradigms within an organization or company, aiming to enhance efficiency, boost productivity, maximize profits, and instill confidence in clients by promptly adapting to their evolving requirements[10].

Lean Manufacturing (LM) is a systematic approach that focuses on identifying and removing operational inefficiencies through continuous improvement, cost reduction, and meeting customer expectations for maximum value at a competitive price point. Japanese manufacturers, exemplified by the "Toyota Production System" (TPS), employ the Genba Kanri base operating technique as a key element of their lean manufacturing production system. Over the years, several Japanese companies have dedicated significant efforts to developing and incorporating these foundational operating standards, reaching a stage where they are ingrained in manufacturing processes. The concept of "waste elimination" pertains to the reduction or elimination of supply chain activities that do not contribute value [11].

Agile Manufacturing (AM) is characterized as the ability to excel in a competitive environment characterized by ongoing and unpredictable changes, driven by the creation of products and services tailored to meet specific client needs. The concept of agility, encompassing both flexible and rapid-response production, aims to minimize the time required to deliver market-ready products or services to consumers. While leanness primarily focuses on eliminating non-value-added operations, agility emphasizes a proactive response to market dynamics [12] [13].

The integration of LM and AM systems enables the assimilation of new information, is seen as a commitment to optimizing quality, and enhances the benefits offered by the manufacturing system. A prevalent trend in the corporate sphere in recent times involves amalgamating the professions of LM and AM into a cohesive and unified practice [14] [15] [16].

The incorporating of the principles of both lean and agile is introduced in [17], the objective of this study is to put forward a comprehensive method for supplier selection, incorporating the principles of lean, agile, resilience, green, and sustainable (LARGS) practices. This approach underwent validation through structural equation modeling (SEM) and the intuitionistic fuzzy TOPSIS method. The study adds to the current body of literature by introducing an integrated methodology designed to tackle the decision-making complexities associated with supplier selection. This approach serves as a valuable tool for managers aiming to improve sustainable supply chain performance, particularly from the perspective of LARGS principles.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

The contribution in this research effort is to develop a conceptual platform using Simulink/Matlab that mimics a manufacturing system that describe the machining and assembly of a motorcycle engine, this platform will be the base for developing the future Lean+Agile system. Also in this paper the measuring performance criteria will be introduced for the basic waterfall system.

II. METHODS AND MATERIALS

This paper includes the representation or modelling of the workshop proposed to present the conceptual platform using Matlab/Simulink.

The modelling of the manufacturing system (workshop) using the SimScape library inside Simulink. Which build up a block diagram that presents the manufacturing system (workshop) stages. Starting from the raw material yard, then to the raw material packaging system, to the transportation system used to transport the raw material from the raw material resource pool to the warehouse, in the warehouse the un-packaging happens, then to the manufacturing area where the parts first enters a queue that used as an intermediate waiting line for the parts before they enter the manufacturing machines, which are bending, milling, drilling, lathe, grinding, and boring. Then to the redistribution platform that used to deliver the six different and manufactured parts to four distinctive assembly lines. Each assembly line is responsible for deliver a different model. Then to the inspection machines that inspects the assembled model then each model gets delivered to the customer. The model also has the resource pools which mimic the rooms for the workers.

Also it includes the sub-systems used in this conceptual manufacturing system. In which some of the blocks of the system are gathered in groups called sub-system, the reason behind this, is to simplify the visualization of the model, or the block diagram.

Also the auxiliary Matlab programs, these Matlab programs used for two main purposes, the first one is the given parameters extraction program, and as the name indicates, it is used to extract data from and Excel file which include all the data required from this model, as the number of workers for each station and the service time for each machine in this model, and the queue capacity for all of the queues in the model, with the order sheet for the model. The second Matlab program is the results drawing program, which is used to draw all of the results obtained from the Simulink program, and save it automatically in a predefined directory on the PC.

The parameters matrices and the excel sheet which contains all of the data required for the Simulink model to operate. This excel sheet is made in a dynamic matter which facilitates the establishing of the parameters used in Simulink model as the conceptual manufacturing system. The reason of using Matlab/Simulink that it has many features that match the purposes of this study, such as:
Efficiency: Simulink's graphical modeling approach allows for faster system design and prototyping. It reduces the need for manual coding and debugging. Interdisciplinary Modeling: Simulink supports multidomain modeling, making it suitable for complex systems that involve multiple physical domains. Validation and Testing: Simulink enables thorough testing and validation of system designs, reducing the risk of errors and failures in real-world implementations. Code Generation: The ability to generate code from Simulink models streamlines the transition from simulation to real-time implementation. Collaboration: Simulink's automatic documentation and compatibility with MATLAB facilitate collaboration among team members and stakeholders [18] [19] [20] [21] [22].

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

In this study Simscape will be used to model the manufacturing system or the workshop under consideration, due to its high ability to work with discrete (not continuous) products and methodologies, with other important features such as:

Cross-Domain Modeling: Simscape enables the modeling and simulation of systems that involve multiple physical domains, providing a holistic view of system behavior. Time Savings: The automatic equation generation in Simscape eliminates the need for manual derivation of equations, saving valuable time in the modeling process. Customization: Users can customize existing components or create their own, allowing for precise modeling of specific systems and components [23] [24] [25] [26] [27].

A. Problem Formulation

Fig. 1, shows the manufacturing system of this paper modelled in Simulink. This model mimics the manufacturing system for manufacturing motorcycle engine. The manufacturing system is used to manufacture four different models of the motorcycle engine, these models have the names (model_1, model_2, model_3, model_4).

The manufacturing system uses six different primary parts, these parts have the names (part_a, part_b, part_c, part_d, part_e, part_f). The manufacturing system takes its orders as a matrix contains four rows which presents the four models (model_1, model_2, model_3, model_4), and one hundred columns that present the days starting from day one until day 100. The manufacturing system takes one order per day. The order contains the model wanted (which will be model_1, or model_2, or, model_3, or model_4) and the number of copies wanted of that model. The manufacturing system is shown in *Fig. 1* below, it consists of a group of sub-systems and areas.

Each of them has a specific goals and functions to perform inside the manufacturing system. Starting from the left, the first area is the 'Raw Material Inventory', this area has the job of producing the raw parts of (part_a, part_b, part_c, part_d, part_e, part_f). the raw part is the part produced from the casting operation without any further manufacturing operation, it should be noted that the casting operation and any other upstream operations are not included in this manufacturing system model.

In this manufacturing system, the raw parts are generated almost instantaneously at the start of the simulation, the reason behind this is to evaluate the manufacturing system itself, by excluding the effect of shortage in the raw parts.

This area has no dependencies, i.e. it does not depend on any upstream blocks, hence it starts to operate as soon as the simulation starts.

The second area in this manufacturing system is the 'Raw Material Packaging System' or (RMPS) this area consists also from six groups; each group is responsible for one part of the six parts. This area is responsible for mimic the operation of packaging the raw parts into dozens (consists of 12 parts each) or any other number determined by the given parameter from the excel sheet. This area is also takes.

The raw Material Inventory or (RMI). This area, has twelve blocks, divided into six groups. Each group has two blocks, one of these two blocks is the 'entity generator' block, and the other block is the 'entity store' block. Since this area has six groups of blocks, each group is responsible for producing the raw parts of part_a, part_b, part_c, part_d, part_e, or part_f. 'Raw Material Packaging Worker' or (RMPW) for each packaging operation. It takes the (RMPW)'s from the (RMPW) resource pool, this will provide a measuring criterion related to the total usage of the workers in the system.

This area also provides the ability to determine the service time for each of the six lines of packaging, i.e. the raw material packaging time for part_a, through part_f. the 'Raw

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

Material Packaging Time' or (RMPT) is determined using the given parameter in the excel sheet, each of the six parts has its unique (RMPT).

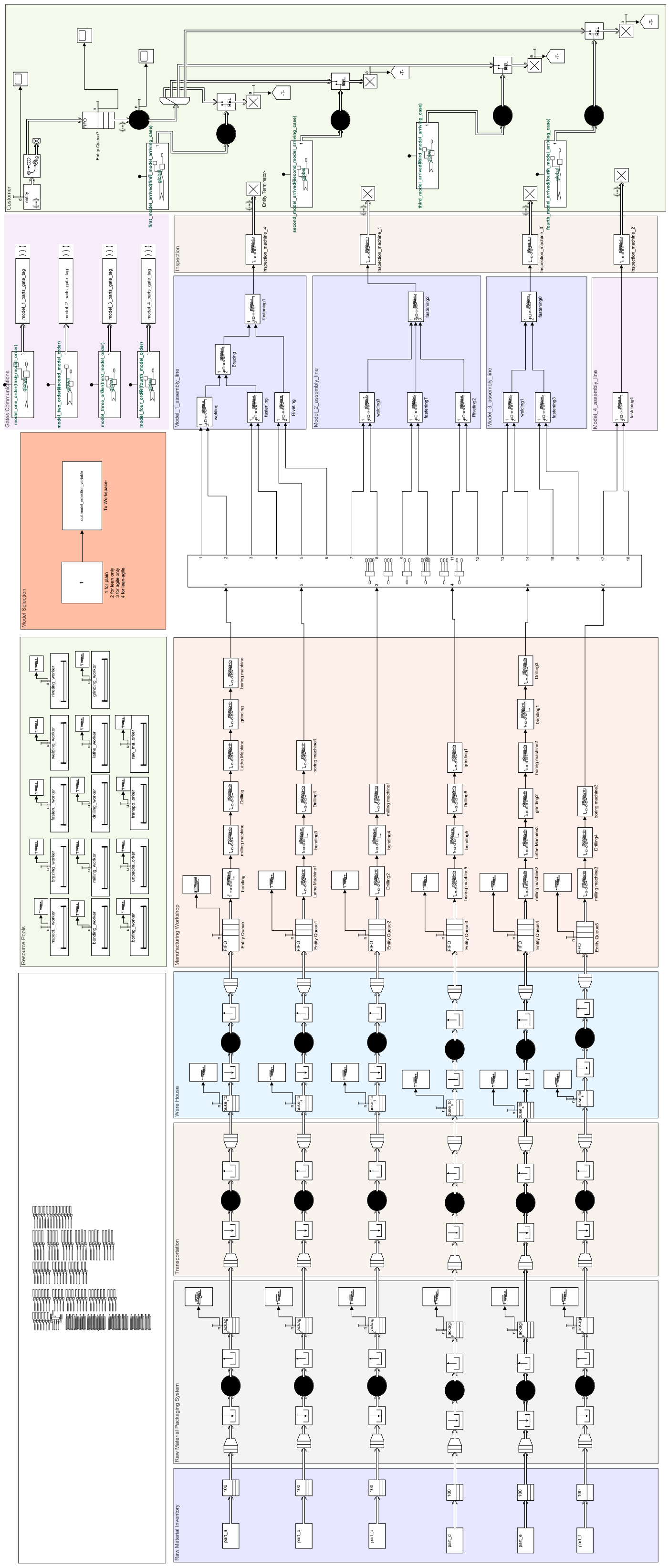

FIG. 1. THE MODELLED MANUFACTURING SYSTEM USING SIMULINK.

Received 03/February/2024; Accepted 28/June/2024

© 2017 University of Technology, Iraq *ISSN (Print) 1811-9212* *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

This area has its ‘entity store’ blocks which is used as a waiting line for the parts before the next area.

The third area of this manufacturing system is the transportation area. Also it consists of six different paths, each of the paths is dedicated with one part. In this area, the raw parts are also packed again, to be transported to next stage. This packaging happens in a tens manner, i.e. each ten groups of each raw parts are packaged together before get transported to the next stage. This are also uses blocks called ‘resource acquire’ used to call for a ‘Transportation Worker’ or (TW). Each (TW) is used to transport one package.

The fourth area is the warehouse. This area presents the inventory for the manufacturing area. And has the job of un-packaging for the raw parts and make them ready to be manufactured individually. This area has also six paths, each of them for one of the six parts. In this area the acquisition for the ‘Un-Packaging Worker’ or (UPW) is done. The service time for the un-packaging is determined using a variable named part-unloading-time which is also determined as a parameter from the excel sheet, this actually six variables that takes different values each of them for a specific part. Also, each of these six paths has a sub-system above the queue block, used to calculate the usage of this queue in the manufacturing system.

The fifth area in this system is the manufacturing area which contains all of the manufacturing operations.

It consists of six paths, each of the paths is related to a specific part, these paths are distinct, i.e. each of the parts has its own manufacturing operations. As shown below, the Table I, illustrates the manufacturing operations for each of the six parts respectively to their order.

TABLE I. MANUFACTURING OPERATIONS FOR EACH PARTS

	Part a	**Part b**	**Part c**	**Part d**	**Part e**	**Part f**
Op.1	Bending	Lathe	Drilling	Boring	Milling	Milling
Op.2	Milling	Bending	Bending	Bending	Lathe	Drilling
Op.3	Drilling	Drilling	milling	Drilling	Grinding	Boring
Op.4	Lathe	Boring		Grinding	Boring	
Op.5	Grinding				Bending	
Op.6	Boring				Drilling	

Each of this operations are presented in the form of sub-systems, each of these sub-system has its unique block diagram that will be explained in the upcoming sections.

The sixth area in this manufacturing system is the assembly lines area.

As the name indicates this area is used to assemble the four models of the motorcycle engine, this area is actually four smaller areas, each of them is responsible for one engine model. It consists of four different assembly operations, welding, fastening, brazing, and riveting. The four models are divided as the next Table II:

TABLE II. DIFFERENT ASSEMBLY OPERATIONS FOR EACH MODELS

	Model 1	**Model 2**	**Model 3**	**Model 4**
Op.1	Welding	Welding	Welding	Fastening
Op.2	Fastening	Fastening	Fastening	
Op.3	Riveting	Riveting	Fastening	
Op.4	Brazing	fastening		
Op.5	Fastening			

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

Each of these four assembly operations has its unique block diagram which will be discussed in details in the upcoming sections.

in consists of four inspection machines, each engine model has its own inspection machine. This area calls for inspection workers, and each model has its own inspection time, i.e. the time taken by the inspection machine connected to the first model is different from the time taken by the inspection machine of the second model.

These inspection time is determined by the user through the excel file. Then these parameters are drawn back to the Matlab workspace using the first Matlab program. Finally, the Simulink model starts to extract these values and use them in the simulation.

The customer area this area is the most technically-sophisticated part of the Simulink model. It mimics the customer's orders in the plant or the manufacturing system. Starting from generating the orders depending on the orders matrix that is extracted from the excel file. After generating the orders, it puts the repeats of each order, and starts to send them to the 'entity terminator' blocks. The 'entity terminator' block mimics the satisfaction of a customer by receiving its related order.

This area has blocks called 'Simulink functions', these blocks used to multicast the required order (or the require model) to the gates of the manufactured parts. For example, of the required model was model_3 which in turn requires part_A, part_c, and part_f to be assembled. The multicast works to open the gates of only the required parts (part_A, part_c, and part_f). This will cause the assembly lines to work only for-demand, i.e. the assembly machines will work only for the current order (the order in-hand), this will reduce the over processing but also will cause a latency in case of changing the model wanted from the order of the customer, i.e. if the wanted model was model_1 with 4 repeats, then the next order was model_3, the gates for the parts of model_3 will open and send the parts to assembly, and the customer of the current model will wait till the assembly operation and the inspection operation is done.

Resource pools area this area as the pools for all of the workers of the system. The word 'pool' in the SimScape library of the Simulink is referring for a block that contains resources (workers) that may be called at any stage of the simulation, in this manufacturing system, each of the machines (lathe, grinding, welding, …etc) has its unique kind of workers, for example the welding worker is different than the grinding worker, and different than the transportation worker.

The total number of workers pools are 14. Each of these pools has its subsystem that calculate its usage over time.

At the start of the Simulink model, these blocks are shown in red color, indicating that a variable is missing, this is due to the fact that each of these pools has it 'number of workers' variable that needs to be extracted from the Matlab workspace. For example, the block 'milling machine workers' requires the variable 'number_of_milling_workers', this variable is extracted from the Matlab workspace, so the first Matlab program must run before running the simulation.

B. The Proposed Algorithm

The proposed algorithm is to develop two Matlab algorithms (scripts) that governs the work or performance of the platform. Also to build up two different Simulink sub-systems that will control the basic outcomes of the Simulink platform.

These Matlab programs are:

C. Extraction program, or the first program:

Received 03/February/2024; Accepted 28/June/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

This program is responsible of extracting all the data from the excel sheet and transferring them to the Matlab workspace.

It starts by the clc command that will clear the command window of the Matlab. Then with the command clear all that will erase all of the variables stored in the Matlab workspace.

It should be noted that this program is the first step in running the manufacturing system, that's why the clear all command is used, to clear all possible values from previous runs.

Then, using the xlread() command to read from the excel sheet, this program extract eight different matrices, each matrix contains a number of constants that will be used in the Simulink model.

Then, each matrix is decomposed alone to extract each of its constants, using the ordinary indentation of the matrix by referring the constant row and column, that's why the excel sheet must keep its general structure, and only the values it contains may be changed. As shown below the Algorithm (1), for first Matlab.

ALGORITH 1. THE GENERATING CODES USING MATLAB PROGRAM IN THIS STUDY

```
clc
clear all
excelFile   = 'MARIAM_1_PLAIN_MODEL_EXCEL_SHEET.xlsx';
service_times    = xlsread(excelFile, 'service_times')
number_of_workers    = xlsread(excelFile, 'number_of_workers')
customer_orders  = xlsread(excelFile, 'orders')
manufacturing_times = xlsread(excelFile, 'manufacturig_times')
thresh_holds= xlsread(excelFile, 'thresh_holds')
machine_capacity= xlsread(excelFile, 'capacity_sheet')
queue_capacity= xlsread(excelFile, 'queue_capacity')
storage_AHP_matrix= xlsread(excelFile, 'AHP_MATRIX')
agile_alone_model_1_final_model_AHP_value=storage_AHP_matrix(5,8);
agile_alone_model_2_final_model_AHP_value=storage_AHP_matrix(5,9);
agile_alone_model_3_final_model_AHP_value=storage_AHP_matrix(5,10);
agile_alone_model_4_final_model_AHP_value=storage_AHP_matrix(5,11);
lean_agile_model_1_final_model_AHP_value=agile_alone_model_1_final_model_AHP_value;
lean_agile_model_2_final_model_AHP_value=agile_alone_model_2_final_model_AHP_value;
lean_agile_model_3_final_model_AHP_value=agile_alone_model_3_final_model_AHP_value;
lean_agile_model_4_final_model_AHP_value=agile_alone_model_4_final_model_AHP_value;
raw_material_packaging_storage_lean_alone_AHP_value=storage_AHP_matrix(1,7);
ware_house_storage_lean_alone_AHP_value=storage_AHP_matrix(2,7);
lean_agile_raw_material_packaging_storage_AHP_value=raw_material_packaging_storage_lean_alone_AHP_
value;
lean_agile_ware_house_storage_AHP_value=ware_house_storage_lean_alone_AHP_value;
raw_material_packaging_storage_part_a_AHP_value=storage_AHP_matrix(1,1);
raw_material_packaging_storage_part_b_AHP_value=storage_AHP_matrix(1,2);
raw_material_packaging_storage_part_c_AHP_value=storage_AHP_matrix(1,3);
raw_material_packaging_storage_part_d_AHP_value=storage_AHP_matrix(1,4);
raw_material_packaging_storage_part_e_AHP_value=storage_AHP_matrix(1,5);
raw_material_packaging_storage_part_f_AHP_value=storage_AHP_matrix(1,6);
ware_house_storage_part_a_AHP_value=storage_AHP_matrix(2,1);
ware_house_storage_part_b_AHP_value=storage_AHP_matrix(2,2);
ware_house_storage_part_c_AHP_value=storage_AHP_matrix(2,3);
ware_house_storage_part_d_AHP_value=storage_AHP_matrix(2,4);
ware_house_storage_part_e_AHP_value=storage_AHP_matrix(2,5);
ware_house_storage_part_f_AHP_value=storage_AHP_matrix(2,6);
manufacturing_workshop_storage_part_a_AHP_value=storage_AHP_matrix(3,1);
manufacturing_workshop_storage_part_b_AHP_value=storage_AHP_matrix(3,2);
manufacturing_workshop_storage_part_c_AHP_value=storage_AHP_matrix(3,3);
manufacturing_workshop_storage_part_d_AHP_value=storage_AHP_matrix(3,4);
manufacturing_workshop_storage_part_e_AHP_value=storage_AHP_matrix(3,5);
manufacturing_workshop_storage_part_f_AHP_value=storage_AHP_matrix(3,6);
```

This algorithim using Matlab program is composed of (329) lines of coding.

Received 03/February/2024; Accepted 28/June/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

By referring to constant position inside the matrix, and give it a distinct name, its value will be stored in the Matlab workspace automatically.

The Matlab workspace after running this first Matlab program (Algorithm1) has all the variables wanted and needed for the further steps.

As stored in the workspace. all of the values are set to their variable names and stored accordingly in the Matlab workspace.

The names of the matrices extracted are:

$$\text{service\_times} - \text{number\_of\_workers} - \text{customer\_orders} - \text{manufacturing\_times} - \text{thresh\_holds} - \text{machine\_capacity}, - \text{queue\_capacity} - \text{storage\_AHP\_matrix}$$

D. Drawing Program, or the Second Program:

This program has the purpose of drawing the results available in the Matlab workspace.

This program should be implemented at the final step of the run, i.e. after the simulation is done.

The drawing program is composed of (11626) lines of coding. The reason behind this size of extensive coding, that this program is responsible of drawing the results of five different configurations of the manufacturing system, the first configuration is the water fall configuration, the second configuration is the lean, the third configuration is the agile configuration, the fourth configuration is the lean-agile configuration, and the fifth configuration is the fuzzy configuration.

On the other hand, the proposed two sub-systems are:

E. Usage-Percentage Subsystem

This subsystem presents the backbone of the results. Due to the fact that this subsystem is used extensively in this model of the manufacturing system. The subsystem is shown below in *Fig. 2*.

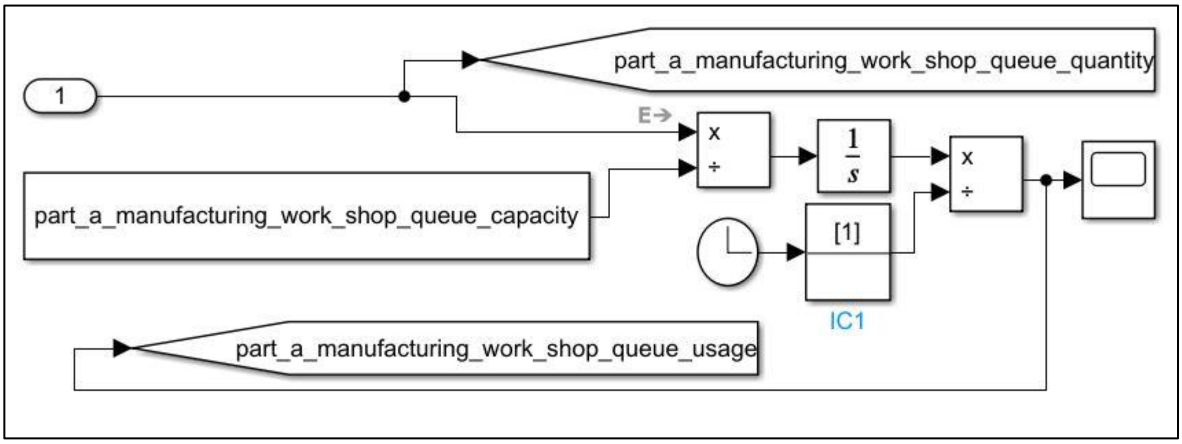

FIG. 2. THE USAGE-PERCENTAGE SUBSYSTEM.

This subsystem calculates the usage-percentage of each block that is connected to. For example, it calculates the usage of this queue over time (as a percentage).

Starting from the upper left corner, the first block is the ‘input port’ block which takes its input as the number of entities in the queue (the queue of the manufacturing area for part_A).

This input has given the name of (part_a_manufacturing_work_shop_queue_quantity), the first step after the input port is the ‘go to’ block which takes this variable and send its

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

values over time to the Matlab workspace using a 'to workspace' block (that will be taken later by the second Matlab program to be drawn).

After the 'go to' block, a 'divide' block is used to divide the incoming variable over the constant 'part_a_manufacturing_work_shop_queue_capacity' (this variable is extracted from the excel file to the Matlab workspace by the first Matlab program). The resulted signal from this 'divide' block is the percentage occupation of the queue under consideration, takes its maximum value "1" in the case where the two variables are equal:

$$\text{part\_a\_manufacturing\_work\_shop\_queue\_quantity} = \text{part\_a\_manufacturing\_work\_shop\_queue\_capacity}$$

This value of '1' indicates that the queue is used at its maximum ability. The value of '0' indicates that the queue is empty.

After the divide block, an 'integration' block is used, this integration block will calculate the usage over time. After this integration block, there is another 'divide' block, the incoming signal will be divided by the time passed since the start of the simulation (the current time of the simulation). This divide operation will give the percentage usage of this queue block (or any other inventory blocks that is connected to).

The 'clock' block is the block that gives the current time (the time passed since the start of the simulation).

It should be noted that the block that is connected to the clock block is the 'initial value' block, which provides a value taken only once (at the start of the simulation, i.e. at the first step of the simulation) which gives the value of unity 'one'. This block is presented to prevent the case of 'divide by zero' at the first step of the simulation, because of the fact that at the first step of the simulation, the clock block gives the value of zero, due to the fact that at the start of the simulation the time passed since the start of the simulation is zero.

After the last 'divide' block, these is a 'scope' block which is used to monitor the performance of this subsystem as the time of the simulation goes by.

Finally, the output signal is sent to a 'to workspace' block by using a 'go to' block, that will send this results to the Matlab workspace, that will be used later by the second Matlab program (Algorithm2).

F. Path Selection Subsystem

This subsystem is presented six times in this manufacturing system, each of the parts (part-a to part-f) has its unique 'path selection subsystem'.

This subsystem guides the manufactured part to the designated assembly line (model-1 or model-2 …etc). The subsystem is shown below in *Fig. 3*.

Starting from the left side, the subsystem starts with an input block that takes part_A as the incoming entity, after the input port, an 'entity queue' is used to store the incoming entities in case of blocking the downstream path for part_A.

The 'entity queue' has its output port connected to the 'enable gate' block. in the same time the 'entity queue' block has a signal out called 'n' which is the 'number of entities in the block' that mimic the number of part_A that are in the queue. This signal line goes first (to the left) to a 'usage percentage' subsystem to calculate the usage of this queue. Another route (upward) goes to a 'go to' block that will lead this signal to the Matlab workspace. Another route (to the right), goes to a 'switch' block that uses the criteria of 2 (2 parts as minimum) to send a message to the enable gate to be opened and let part_A passes. This 'switch' block is used to make sure that parts don't pass only if they achieve a certain number of parts available at the queue, which in this case is 2. If the number of parts in the

Received 03/February/2024; Accepted 28/June/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

queue is more than 2, the enable gate will receive a message of unity (positive number) and hence it will open letting only one part to pass to the 'entity output switch'.

The 'entity output switch' outputs the incoming entity into four different possibilities; the first possibility is the first path which mimics the first assembly line (first engine model). And the second possibility is the second path, which mimics the second assembly line (second engine model). The third possibility is the third path which mimics the third assembly line (third engine model). And the fourth possibility is the fourth path which mimics the fourth assembly line (fourth engine model).

It should be noted that this 'path selection sub system' is for part_A, part_A enters in all of the four models, that's why it has four possibilities. In case of part_F, it has only two possibilities, due to the fact it enters only in two engine models.

The outputs of the 'entity output switch' have 'release gate' on it, this 'release gate' operates on entities that come from the multicast receiver block shown above the 'release gate' block. This multicast receiver receives it message from a multicast sender blocks located at the customer area.

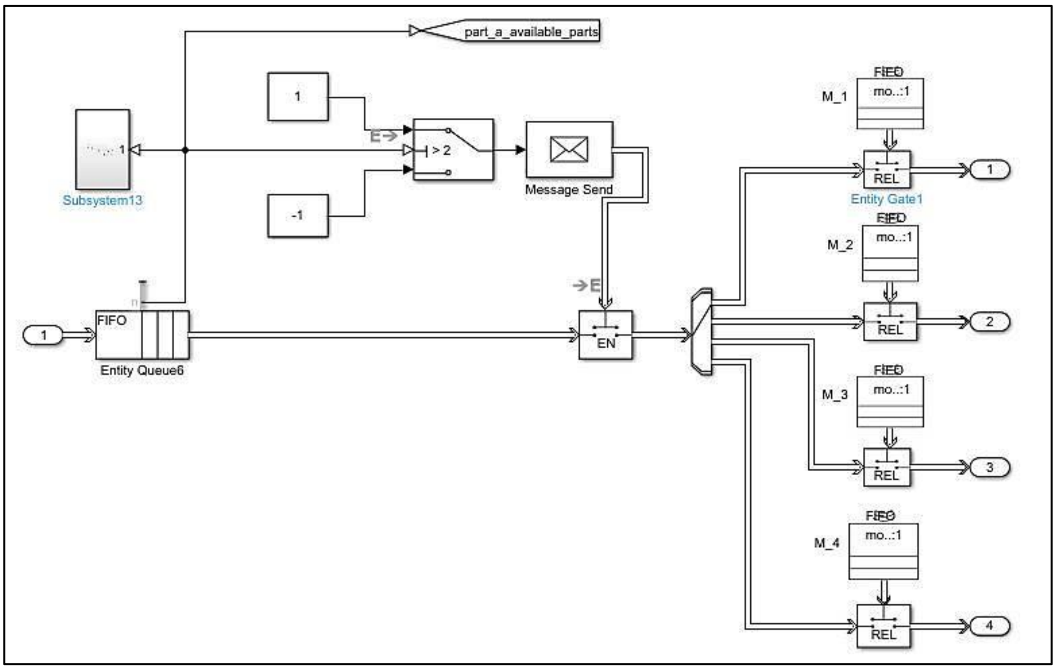

FIG. 3. THE PATH SELECTION SUBSYSTEM.

G. Parameters

One of the main parts of the manufacturing system is the constant parameters that are used during the simulation, for example: the orders, total number of milling workers, or the welding workers, service times for the machines…etc.

All of the constant parameters are gathered in a single excel file, this file is used in the `xlread()` function from Matlab in the first Matlab program.

All of these values could be changed before executing any run of the simulation, and should be the same for all of the five configurations of the manufacturing system. The matrices are the next:

H. The "orders" matrix: As shown below in Table III, the order matrix.

Received 03/February/2024; Accepted 28/June/2024

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

TABLE III. THE ORDERS MATRIX

	day 1	day 2	day 3	day 4	~~	Day 100
model_1	1				~~	2
model_2		1			~~	
model_3			20		~~	
model_4				1	~~	

The orders matrix contains all of the orders, it has four rows which present the four engine models (first model, second model, third model, and the fourth model).

It has (100) columns, each column presents a day (or an order), the simulation has the capacity to run the manufacturing system for (100) days (or orders).

Each cell of the matrix presents the repetition of the order, which mimics the number of copies of that engine model.

I. The "service_times" matrix: This matrix is shown below in the following table (4).

TABLE 4. THE SERVICE TIMES MATRIX

(all times in minutes)	A	B	C	D	E	F	LEAN ALONE	LEAN AGILE and FUZZY
Raw Material Packaging time	2	1	3	4	2	1	2	2
Transportation Time	1	1	1	1	2	1	3	3
Ware House Unloading Time	1	1	1	1	1	1	4	4
Bending Time	1	1	1	3	1	1	0	
Milling Time	1	2	1	1	1	1	0	
Drilling Time	1	1	1	1	1	1	0	
lathe Time	1	2	1	1	1	1	0	
Grinding Time	1	1	1	6	1	1	0	
Boring time	1	2	1	1	7	1	0	
sum	10	12	11	19	17	9	9	9

This matrix is used to store the service times of all of the processes inside the manufacturing system. Starting from the raw material packaging time for all of the parts (A, B, C, D, E, and F). Then the transportation time for all of the parts.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

It should be noted that the last two columns are related to the different configurations of the manufacturing system that will be explained later.

The rest of the service times are shown in the table, which are: warehouse unloading time, which is the time required for the workers to unload each of the parts at the warehouse inventory. Bending time, which is the time taken by the bending machine for each of the parts. The same for all of the other machines.

J. Proposed Workflow

The general workflow is shown below in *Fig. 4*. The work starts by wetting the values at the excel sheet. Then running the first Matlab program that will extract these values and transferring them to the Matlab workspace. After that the first model (waterfall) is opened and simulated. Then the second Matlab program get to run to show the results of the first model (waterfall).

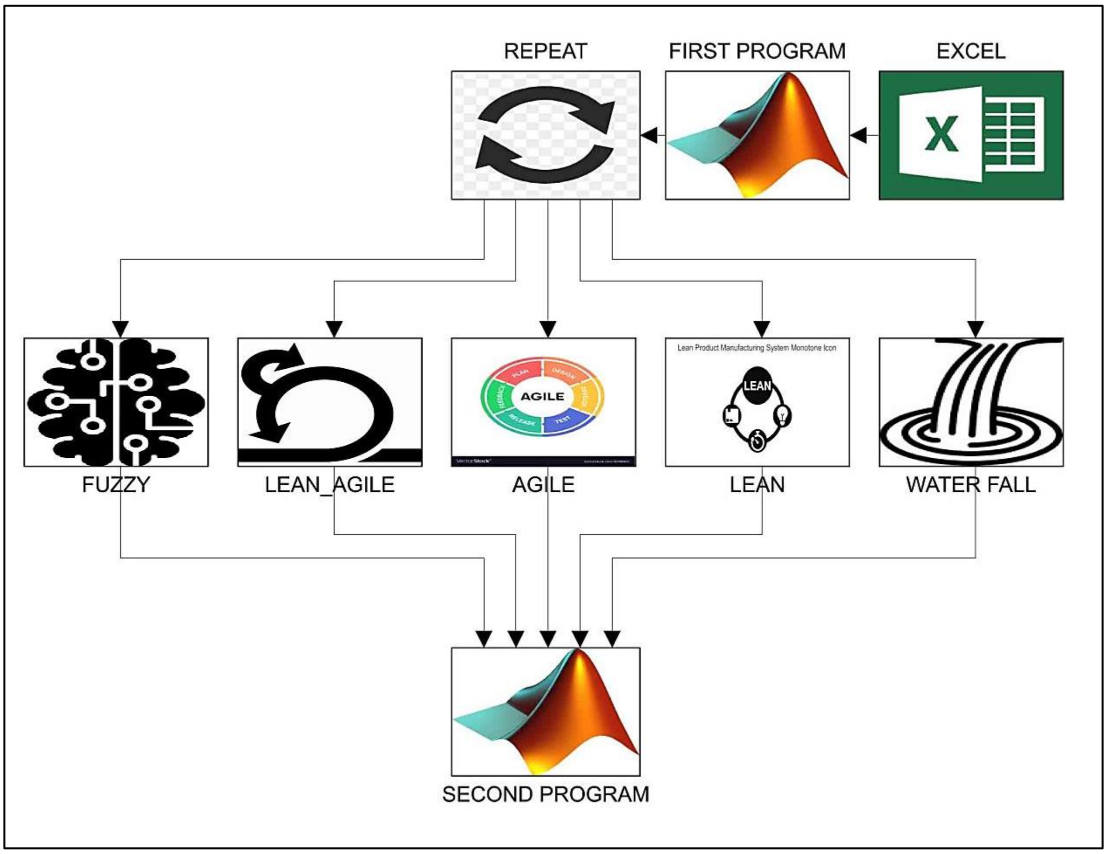

FIG. 4. THE GENERAL WORKFLOW FOR THE CONCEPTUAL MANUFACTURING SYSTEM.

After that the second model (lean) is opened and get to be simulated, then the second Matlab program is used again to the results of the second model (lean).

This process is repeated for all of the remaining three models (agile, lean-agile, and fuzzy models).

III. RESULTS AND DISCUSSIONS

In order to measure the performance of the proposed manufacturing system, the second Matlab program is run and the results are presented through multiple figures that are generated automatically and saved in a user-defined directory.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

The first result obtained from the second Matlab program is the (WaterFall Raw Material Packaging store usage). This plot is shown in *Fig. 5*.

The horizontal line shows the time in (ten minutes), and the vertical axis shows the store usage as a Percentage of the time.

The legend at the upper right corner of the plot shows the colors of six curves each of them presents a part, starting from part a to part f.

It can be noted that the curves for all of the parts keeps a value of zero from the beginning of the simulation till the time of 75, this is due to the fact that there is not a single part kept at the store during this time and this is because the parts during this time are in the raw material packaging system and none of them reached its queue yet.

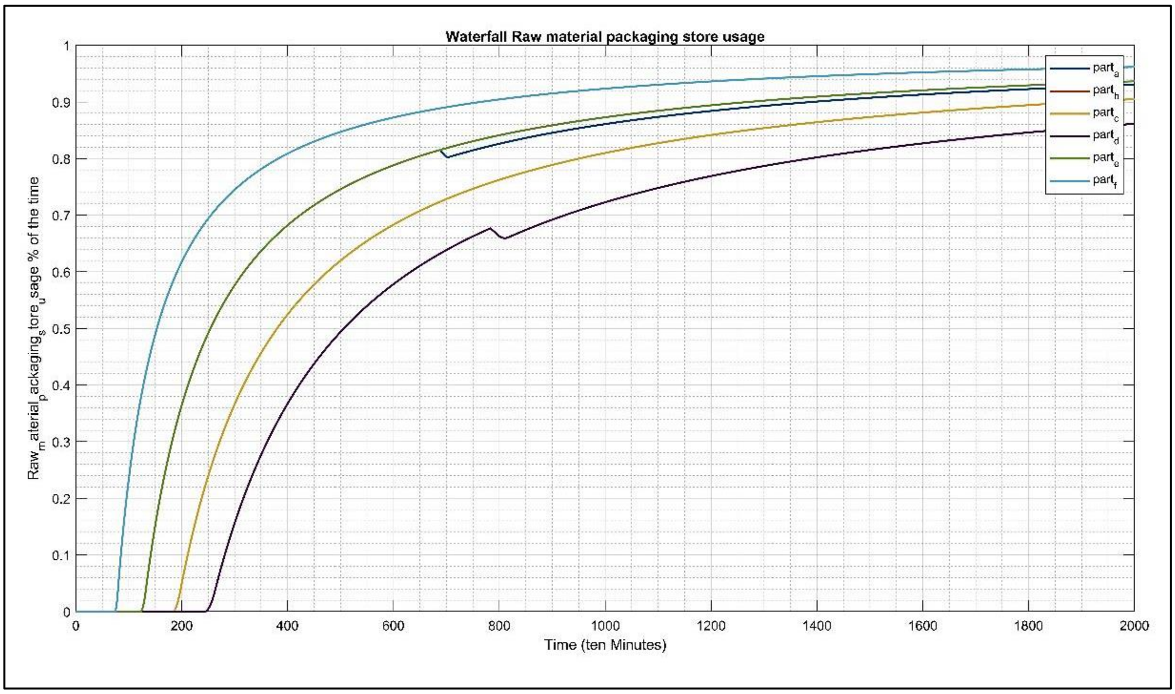

FIG. 5. WATERFALL MATERIAL PACKAGING STORE USAGE.

In order to find the area under the six curves of the previous figure, i.e. *Fig. 5* another figure is generated from the same second Matlab program, in which the areas are shown in colored bars, these bars are shown in *Fig. 6*.

The horizontal axis has the indications for the different areas starting from part-a to part-f and the vertical axis has the units of (time cross percentage of usage).

It can be noted from *Fig. 6* that the least usage-over-time is for part-d, this is because that all part-d has the longest time for the raw material packaging time from the excel file and its value is shown in the previous chapter. So that its raw material store kept empty the most time.

On the other hand, it can be noted that both of part-b and part-f has the same vale of usage-over-time, from *Fig. 5* it can be seen that both of these parts has the same and identical curves.

It should be noted that in all of the usage-over-time figures that will be presented in this chapter the maximum value that a bar can reach is 2000 (ten-minutes cross percentage of usage), to illustrate this feature, back to *Fig. 5*, the horizontal axis has a maximum value of (2000) ten minutes, and the vertical axis has the value of 1 (indicating a 100% usage). So

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

that by multiplication it will give the value of 2000 (ten-minutes cross percentage of usage). One of the main criteria in comparing different manufacturing systems is the over processing, that indicates the waste effort done by the manufacturing system.

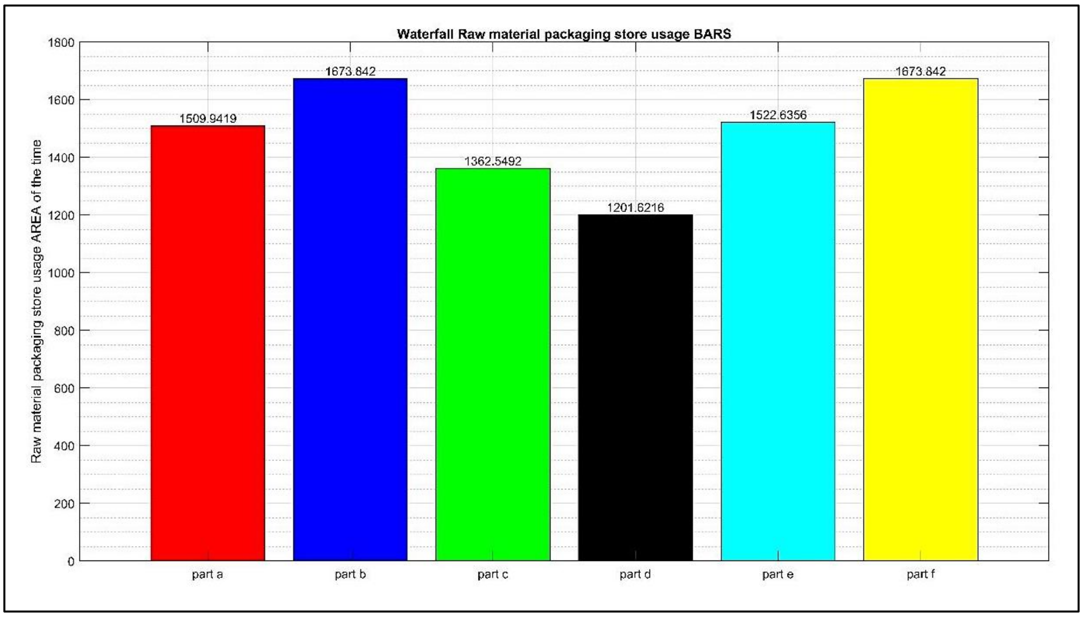

FIG. 6. WATERFALL RAW MATERIAL PACKAGING STORE USAGE.

The over processing for all of the storage queues is shown below in *Fig. 7*.

Fig. 7 shows in its vertical axis the inventory usage impact, this factor presents the impact for all of the inventory places available in all of the queues one area (raw material packaging area, ware house area, machining area, path selection area). And the horizontal axis shows two adjacent columns for each of the areas that contains queues.

For each of the four areas, the first column presents total storage impact. And the second column shows the storage usage impact.

It can be noted that the maximum impact (or the total storage impact) is different between the area. The equation for the storage impact is:

$$\text{total storage impact} = \sum_{i=a}^{i=f} \text{part (i) sorage capacity} \ldots\ldots\ldots\ldots\ldots\ (1)$$

These values are taken from the excel file, and for the second column, the over processing for the storage is the remaining parts in the same storage after the simulation is finished. With the following equation:

$$\text{over processoing for the storage} = \sum_{i=a}^{i=f} \text{part (i)remaining quantity at the finish} \ldots\ldots\ldots\ldots\ldots\ (2)$$

In order to put another main factor under consideration which is the impact of the components under study, more figures are generated which are the usage impact bar charts, as a start, *Fig. 7* shows the inventory usage impact.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

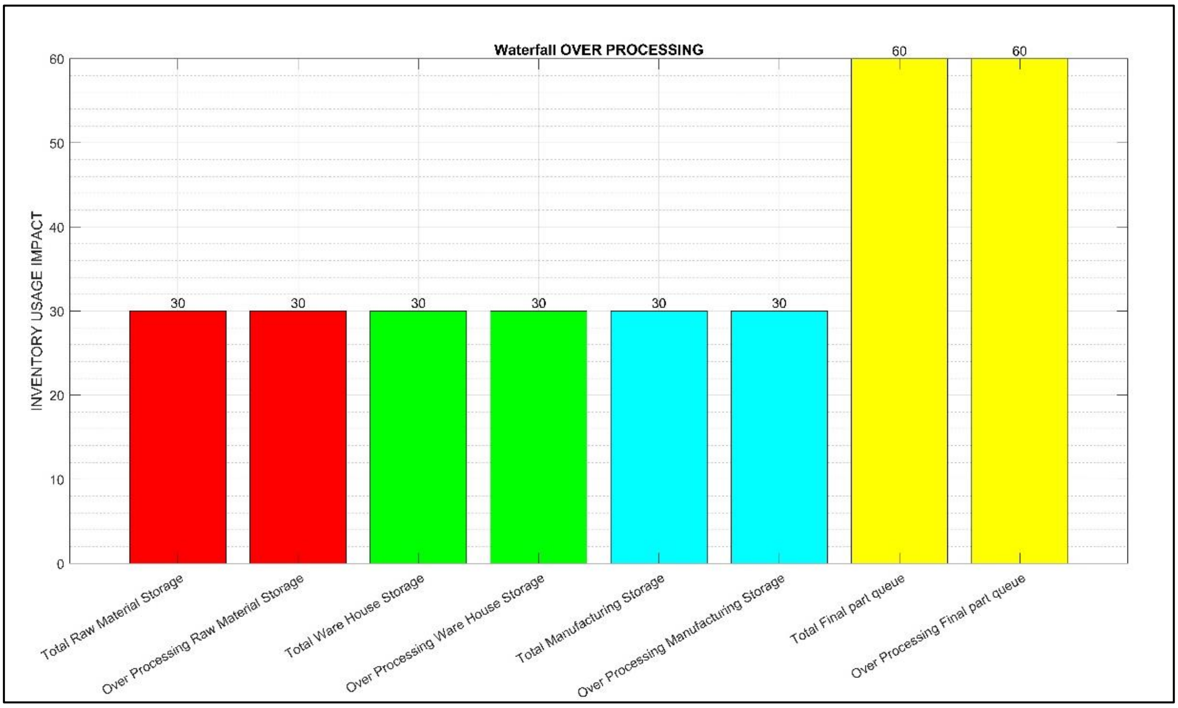

FIG. 7. WATERFALL OVER PROCESSING (BAR CHART).

This factor (the impact factor depends on the AHP matrix that resented earlier in the previous chapter, which is in the excel file.) for example, at the warehouse storage area, the different parts require different spaces for storage, for example, part-d has bigger size that the other five parts, so that, it should be dealt with its impact to storage area in different manner that the other (smaller) five parts. Or this reason, the impact charts are generated to add more feasible and sensible measuring criterion to understand the different aspects of the manufacturing system, also to facilitate the comparison between the different platforms in this study.

For example, in *Fig. 8* as shown below, each storage area has been presented in two columns, the first column, is the total impact and the second one is the actual impact. For the total impact, the equation is:

$$\text{Total impact} = \sum_{i=a}^{i=f} \text{part(i)total impact} \sum_{i=a}^{i=f} (\text{part(i)storage capacity} * \text{part(i)AHP matrix} * \text{total time value}) \ldots\ldots\ldots\ldots (3)$$

It should be noted that the total time value is the same for all of the study which is the total simulation time (2000 ten minutes of simulation).

For the second column the equation is:

$$\text{Actual impact} = \sum_{i=a}^{i=f} (\text{part(i)total impact} * \frac{\text{part (i)store usage (from bar chart)}}{\text{total time value}}) \ldots\ldots (4)$$

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

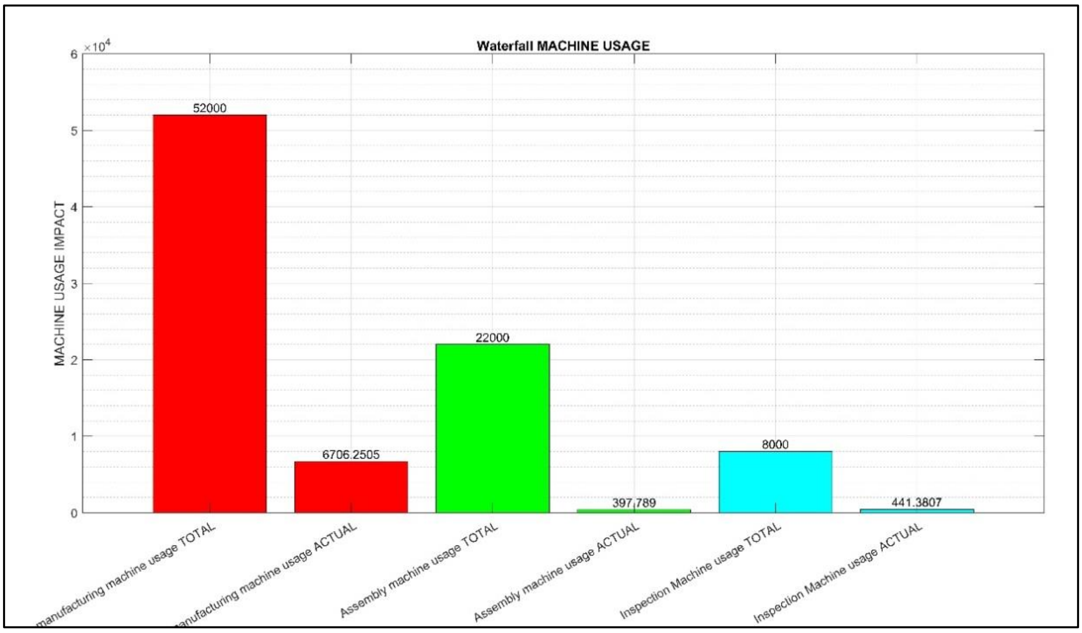

FIG. 8. WATERFALL MACHINE USAGE (BAR CHART).

In *Fig. 9* as shown below, the machine usage impact is shown; it is also shown in a bar chart mode. The reason that all of the impact figures are in a bar chart mode, and none of them is in a curve mode, is that (an according the previous two equation) the impact is a single value found at the end of the simulation.

From *Fig. 9*, assembly machine usage has a relatively the smallest factor from its total impact value, the reason behind this, is that all of the upcoming (downstream) path is blocked, due to the fact that the models are assembled but there is no order to hand these models for, so the assembled models stay and wait at the simulation lines, this phenomenon should be settled at the lean-alone platform.

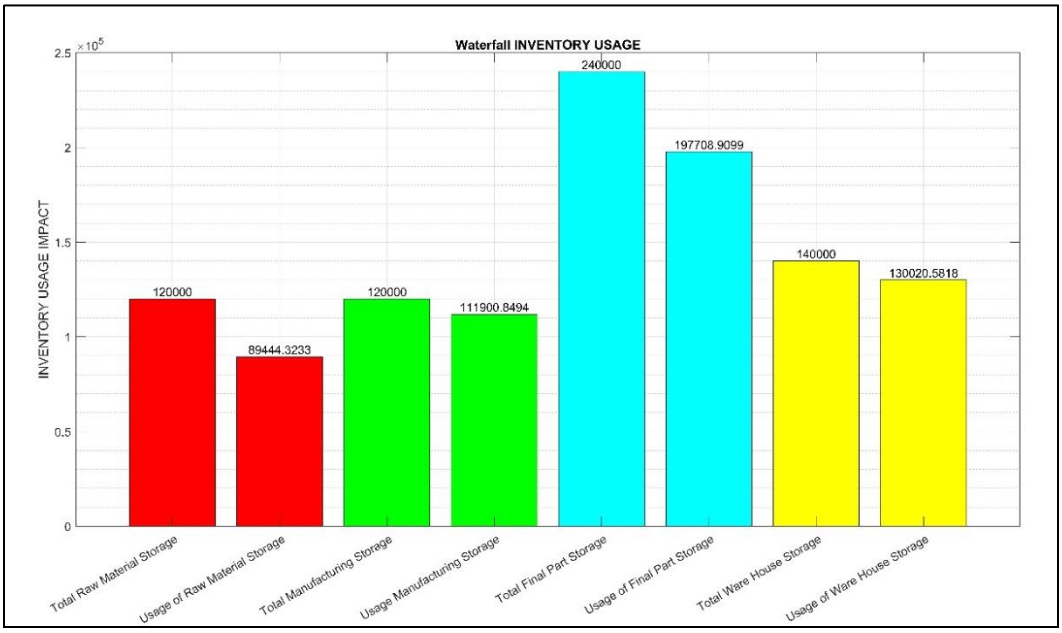

FIG. 9. WATERFALL INVENTORY USAGE (BAR CHART).

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

Fig. 10 shown below, the another main output of the second Matlab program (results for second Algorithm), which is the total orders done.

As shown in *Fig. 10*, the total orders output is shown in blue line (stepped line or line segments), this segmented line is composed of adding all of the orders of the four models together, which are also shown in the same figure with different color codes.

It should be noted, that all of the orders are produces instantly as the simulation starts. i.e. the orders matrix from the excel file has no sense of time, it provides all of its orders instantly, but with the same arrangement and cascading provided by the excel file. This point was very important in order to test the maximum ability of the platform under consideration to satisfy the orders. for example, if the times (or dates) of the orders were taken into consideration, it may happen in one or more of the platforms that the platform is faster than the orders flow, this will cause the platform (or the manufacturing system) to deal with no-order case, causing the loose of the comparison sense for the platform if it was fast or slow in satisfying the orders.

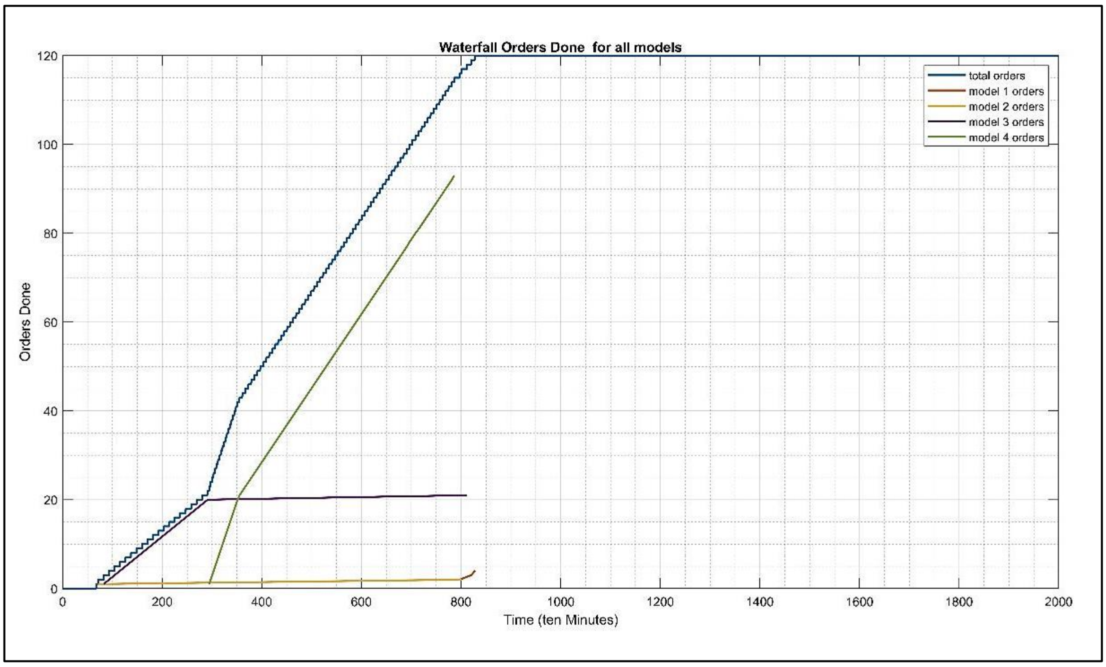

FIG. 10. WATERFALL TOTAL ORDERS DONE.

IV. CONCLUSIONS

A motorcycle engine manufacturing system has been modelled using Matlab Simulink. The SimScape Library of Simulink has been used to mimic the different stages of the manufacturing process. Different areas were developed to present the various aspects of a manufacturing system, like the raw material area, the transportation system, the warehouse area, the manufacturing processes, the assembly lines, the inspection area, and the customer's order area. And in order to calculate the usage of the machines and the inventory queues, a special subsystem were developed to produce the percentage usage of these components. Another subsystem was developed to determine the right path for the manufactured part. Two Matlab programs were developed separately to perform auxiliary roles, the first Matlab program used to extract the parameters required for the simulation from an excel File, and the second Matlab program used to draw the results obtained from the simulation after running the program. An excel file was developed to contain all of the

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

matrices required for the simulation that will grant an ease of change for the parameters that are used in the simulation. A general workflow for the whole model is developed to explain the process of the model. Multiple results were shown for the waterfall platform showing its performance. The proposed waterfall system shows (as expected) 100% over processing for all of the queues and storage phases along the manufacturing line. It shows a 0.35 (or 35%) of workers usage which is around 17 workers from the total number of workers 50 workers. It shows the inventory usage to be ($\frac{197708}{240000} = 82\%$) for the final part storage queues. It shows that it done all the orders (120 orders) at the time of (828.2 ten minutes). Which is relatively short time but with high losses and over processing.

V. DECLARATIONS

Availability of data and material: Supplementary materials such as the Matlab scripts and the Simulink platform are available upon contacting the authors email.

Conflicts of Interest: The authors declare no conflict of interest.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable

VI. LIST OF ABBREVIATION

Abbreviation	Meaning
RMI	The raw Material Inventory
RMPW	Raw Material Packaging Worker
RMPT	Raw Material Packaging Time
RMPS	Raw Material Packaging System
LM	Lean Manufacturing
AM	Agile Manufacturing

REFERENCES

[1] Ghobakhloo, Morteza, and A. Azar., "Business excellence via advanced manufacturing technology and lean-agile manufacturing," Journal of Manufacturing Technology Management 29.1, p.p.: 2-24., 2017.

[2] P. Nieuwenhuis, E. Katsifou, "More Sustainable Automotive Production through Understanding Decoupling Points in Leagile Manufacturing," Journal of Cleaner Production 95, p.p.: 232-241. 2015.

[3] Simangunsong, Surya Seven Y., Teguh Raharjo, and Anita Nur Fitriani, "Lean and Agile Software Development for Managing Technical Debt on A Largescale Software: A Systematic Literature Review," Indonesian Journal of Computer Science, vol. 12, no. 6, p.p.: 3526, 2023.

[4] R. Uzwyshyn, "The Application of Agile Project Management Principles for Library IT," International Federation of Library Associations and Institutions (IFLA), Erasmus University, Rotterdam Netherlands, 2023.

[5] Lima, B. F., Neto, J. V., Santos, R. S., & Caiado, R. G. G., "A Socio-Technical Framework for Lean Project Management Implementation towards Sustainable Value in the Digital Transformation Context," Sustainability, vol. 15, p.p.: 1756, 2023.

[6] Elmoselhy, Salah A., "Implementing the Hybrid Lean-Agile Manufacturing System Strategically in Automotive Sector," SAE International Journal of Materials and Manufacturing 8.2, p.p.: 592-601, 2015.

[7] Ito, J. Y., Silveira, F. F., Munhoz, I. P., & Akkari, A. C. S., "International publication trends in Lean Agile Management research: a bibliometric analysis," Procedia Computer Science, vol. 219, pp.: 666-673, 2023.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq

DOI: https://doi.org/10.33103/uot.ijccce.24.4.1

[8] Lee, Khai Loon, and Teh Xing Qi., "The Effect of Lean and Agile Practices on Supply Chain Operational Performance in Malaysia Manufacturing Industry," International Journal of Industrial Management 12, p.p.: 319-340. 2021.

[9] Budianto, B., Surachman, S., Hadiwidjojo, D., & Rofiaty, R., "The effect of manufacturing agility competencies on lean manufacturing in increasing operational performance," growing science / Uncertain Supply Chain Management 9.1 p.p.: 195-204, 2021.

[10] Leite, Marco, A. J. Baptista, and A. M. R. Ribeiro, "A road map for implementing lean and agile techniques in SMEs product development teams," International Journal of Product Development 21.1, p.p.: 20-40., 2016.

[11] Khalfallah, M., & Lakhal, L., "The impact of lean manufacturing practices on operational and financial performance: the mediating role of agile manufacturing," International Journal of Quality & Reliability Management 38.1, p.p.: 147-168, 2020.

[12] Udokporo, C., Anosike, A., & Lim, M., "A decision-support framework for Lean, Agile and Green practices in product life cycle stages," Production Planning & Control 32.10, p.p.: 789-810, 2020.

[13] Mathiyazhagan, K., Agarwal, V., Appolloni, A., Saikouk, T., & Gnanavelbabu, A, "Integrating lean and agile practices for achieving global sustainability goals in Indian manufacturing industries," Technological Forecasting and Social Change 171, 2021.

[14] Abdelhadi, Abdelhakim, and Fares Abus Khamis., "Investigating the Application of Agile and Lean Methodologies to Nonprofit Organizations (NPOs) Using Clustering Methodologies," the International Conference on Industrial Engineering and Operations Management Manila, Philippines, p.p.: 166-171, 2023.

[15] Al Samman, Thaeir Ahmed Saadoon, "Modeling Lean, Agile, Leagile Manufacturing Strategies: An Fuzzy Analytical Hierarchy Process Approach For Ready-made ware (Clothing) Industry in Mosul, Iraq," International Journal of Advances in Engineering & Technology 7.3, p.p.: 1091-1108, 2014.

[16] Lotfi, Maryam, and Soroosh Saghiri, "Disentangling resilience, agility and leanness," Journal of Manufacturing Technology Management 29.1, p.p.: 168-197, 2018.

[17] A. Ghazvinian, B. Feng, H. Talebzadeh, & M. Dzikuć, "Lean, Agile, Resilient, Green, and Sustainable (LARGS) Supplier Selection Using Multi-Criteria Structural Equation Modeling under Fuzzy Environments," Sustainability 16.4, p.p.: 1594., 2024.

[18] K. Ogata, “Modern Control Engineering”, Pearson Education, 2010.

[19] Lathi, Bruce, Modern Digital and Analog Communication Systems, Oxford University Press, 2019.

[20] Nise, Norman S., Control Systems Engineering, Wiley, 2015.

[21] Ashrafi, M., "Application of MATLAB/Simulink in Simulation and Modeling of Energy Systems," the International MultiConference of Engineers and Computer Scientists (IMECS).

[22] M. Tavana, "Modeling and Simulation in MATLAB/Simulink of Quadcopter Dynamics," 2018.

[23] T. Nakamura &, Yoda, "Real-time simulation of electric vehicle based on multi-domain modeling in Simscape," in Conference on Human System Interactions (HSI), Warsaw, Poland,, 2015.

[24] R. Cisneros, A. Melendez & D. Sosa, "Simulation of electrical circuits using MATLAB/Simulink," in Southeastern Symposium on System Theory, New Orleans, LA, USA, 2008.

[25] A. Zayko and P. Jokinen, "Vehicle thermal management model for early system design," in International Electric Vehicle Conference, Florence, Italy, 2014.

[26] S. J. Orsila, S. K. Kauhaniemi, and S. K. Särkkä, "Power control of a wind turbine with a doubly fed induction generator using direct torque control," in European Wind Energy Conference & Exhibition, Athens, Greece, 2006.

[27] S. H. Ali, M. G. Ben Haj Fraj, and M. S. Ben Romdhane, "Electric vehicle modeling and simulation," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013.

Received 03/February/2024; Accepted 28/June/2024
© 2017 University of Technology, Iraq