Proceedings of the
4th International Conference on Information Systems, Logistics and Supply Chain
CREATIVE LOGISTICS FOR AN UNCERTAIN WORLD
ILS 2012 – Quebec (Canada), August 26-29

A comparative study of simulation models for the production control of unreliable manufacturing systems

H. Rivera-Gomez[1,*], A. Gharbi[1], J.P. Kenne[2]

[1]Automated Production Engineering Department, École de Technologie Supérieure,
Production System Design and Control Laboratory, Université du Québec
1100 Notre Dame Street West, Montreal, QC, Canada, H3C 1K3
hriver06@hotmail.com
ali.gharbi@etsmtl.ca

[2] Mechanical Engineering Department, École de Technologie Supérieure
Laboratory of Integrated Production Technologies, Université du Québec
1100 Notre Dame Street West, Montreal, QC, Canada, H3C 1K3
jean-pierre.kenne@etsmtl.ca

Abstract: *This paper deals with the production control policy of an unreliable manufacturing system producing one part type, and subject to random failures and repairs. The applied production control policy is based on the so-called hedging point policy (HPP), which consists in building and maintaining a safety stock of product in order to fulfill the demand, and protect the production system against shortages during maintenance actions. The main objective of the study is to determine the most efficient option of the ARENA simulation software that simulates properly the production systems under consideration. To this end, four simulation models mimicking the dynamics and the stochastic behavior of the proposed manufacturing system were developed. Concepts of discrete and continuous simulation and modules from the ARENA flow process template, are applied to develop the models. The hedging point policy is used as input parameter of the simulation models, where we seek to determine the optimal production threshold that minimizes the inventory and backlog cost. Based on simulation results, the performance of the models is evaluated, in terms of accuracy and time economy. The obtained results shows that the continuous simulation model that uses C++ inserts outperforms the other models.*

Keywords: *Modeling methodologies, Manufacturing system, Discrete simulation, Continuous simulation, Optimal control.*

1 Introduction

In this paper we study the optimal control problem of an unreliable manufacturing system consisting of one machine producing one part type. The unreliable nature of the manufacturing system is due to the fact that the machine is subject to random breakdowns and repairs. The decision variable, denoted by the production rate of the machine, influences the number of parts in the stock level. Additionally, the state variable of the system is denoted by the current stock. We can state that we are facing a state constrain control problem where we have to choose an admissible production rate to minimize the inventory and backlog cost over an infinite horizon. In practical terms, this kind of manufacturing system is considered as a complex optimal control model, because it leads in general to intractable problems, and analytical solutions for this type of systems are only known for relatively few simple models. This difficulty promotes the development of alternative approaches, for instances based in computer simulation, to determine optimal control policies.

* Corresponding Author. Address: 1100 Notre Dame Street West, Production System Design and Control Laboratory, Montreal, QC, Canada, H3C 1K3.
Email address: hriver06@hotmail.com

Faced with such a control problem, it should be first be ascertained that in this paper, we are interested in developing simulation models using the software ARENA, this allow us to pinpoint its capabilities at reproducing accurately the dynamics and the stochastic behavior of the manufacturing systems under consideration. Generally, in simulation it exists different ways to tackle a given problem, and in this respect, the ARENA simulation software offers different options to model our production system, such as discrete or continuous models, and flow process templates. Hence, we intend in this research to examine these possibilities and determine the most efficient option to simulate our production system, in terms of accuracy and time economy.

Many researchers have studied the area of optimal control of manufacturing systems, for example Akella and Kumar (1986) proposed a complete analytical solution for the problem of controlling the production rate of a failure prone manufacturing system with constant demand. We have other analytical solution with the work of Bielecki and Kumar (1988), where they presented a solution of the previous model but without the discounted cost criterion. These works helped to consolidate the so-called Hedging Point Policy. Later Sharifnia (1988) established that the HPP is susceptible to generalizations such as multi state, in particular, he described a method for solving a short term production control problem of a failure prone manufacturing system. At first sight, finding an analytical solution to this type of productions systems is a difficult task, fortunately, alternative approaches based in discrete event simulation have been developed. For example, Kenné et al. (1997) combined a discrete-event simulation model with an analytical approach to include stochastic demand and lot sizes production. Other application of discrete simulation modeling was done by Kenné and Gharbi (1999), where they developed a simulation model considering the age of the machine and preventive maintenance, also they proposed a simulation- based control-approach that analyze the simulation results with design of experiment and optimize the control parameters with response surface methodology. Gharbi and Kenné (2003) deals with the production control problem of a system which involves multiple machines producing different part types. Several other factors have been included during the time to the area, for example Hajji et al. (2004) studied the impact of set-ups in the control policy, and they proposed a corridor policy that outperforms the Hedging point policy. A more complex model is found in the work of Gharbi and Kenné (2006), who studied a production and set-up policy for an unreliable multiple-machine, multiple-part type manufacturing system. Transfer lines are treated by Lavoie et al. (2007), where they achieved to optimize the parameters of the control policy with a discrete and a continuous simulation model. Other interesting simulation model was done by Lavoie et al. (2009), who compared different control mechanism such as Kanban, CONWIP and Hybrid for a tandem production line via simulation. Randomness has been extended to a more complex case by Hajji et al. (2009), where they studied the context of a supply chain, with unreliable producer and supplier.

As we can realize in the literature review, this area is very active, and there are a lot of interesting works analyzing different factors. Some of these works consisted in mathematical formulations based on the optimal control theory. It has to be emphasized, however, that analytical solutions are known only for a few simple cases, and so applications based in simulation, have proved to be an efficient alternative to circumvent the difficulty of finding analytical solutions. This amounts to the necessity of determining the best way to simulate this kind of manufacturing systems. Thus, in this paper we use the manufacturing system consisting of one machine producing one part type to compare four simulation models developed with the simulation software ARENA. We select this computational program because it is one of the most applied and extended simulation software in engineering. The simulation results will provide us some insight about the implementation of this software to simulate manufacturing systems that applies the Hedging Point Policy. The conclusions obtained will serve us to address more complex issues in future contributions, since simulation-based approaches are an effective alternative to examine this type of manufacturing systems, as discussed in the literature review. Applying different modules of the ARENA software, four simulation models were developed; the first model is based in discrete simulation concepts, two models were based in the concept of continuous simulation, where ARENA offers the possibility to complement the model with C++ or VBA files. The fourth model consisted in the implementation of Flow Process modules available in the ARENA software. It is important to note that, we are interested in two performance indices: the accuracy in the results and the velocity of the models. These indicators are used as parameters to determine the most efficient simulation model, since the next stage in our research methodology, implies the study of much more complex aspects that require several simulation runs.

The rest of the paper is organized as follow: Section 2 states the notation required by the model. The formulation of the production planning problem is then presented in Section 3. The proposed approach is described in Section 4 and Section 5 describes the logic applied for the simulation models. Subsequently, a comparison of the four simulation models with regards to the accuracy of the results and time economy is presented in Section 6. Finally, Section 7 concludes the paper and summarizes the main results.

2 Notations

The following notation are used in this paper:

$x(t)$ Inventory level at time t

u(t)	Production rate of the manufacturing system at time *t*
d	Constant demand rate
U_{max}	Maximum production rate
c^+	Incurred cost per unit of produced parts for positive inventory
c^-	Incurred cost per unit of produced parts for backlog
MTTF	Mean Time to Fail
MTTR	Mean Time to Repair
*Z**	Optimal Hedging point
$C(Z)$	Total cost obtained by simulation
$\hat{C}(Z)$	Cost function estimated by regression
C_{DISC}	Cost function of the Discrete model
C_{C++}	Cost function of the Continuous C++ model
C_{VBA}	Cost function of the Continuous VBA model
C_{FLOW}	Cost function of the Flow process model
β_n	Regression coefficients
$\dot{x}(t)$	Differential equation for the stock dynamics
θ	Binary variable that indicates failures and repairs

3 Problem statement

The present section introduces the manufacturing system used for the comparison of the simulation models. The system consists of an unreliable single machine producing one part type. The machine can work at maximum capacity U_{max} to satisfy a constant demand rate *d* of products. It is assumed that $U_{max} > d$. The machine is subject to failures, which are defined by an exponential distribution, and its respective mean is denoted by *MTTF*. After a failure the machine is repaired during a random amount of time given by an exponential distribution with mean *MTTR*. At failure the machine stops producing parts, at this moment, the inventory stock starts decreasing until the value of zero. If the machine is still not available, then the backlog stock will increase. We assume that the demand of finished parts is only satisfied by the inventory stock. We can understand the role of the inventory stock as a safety stock, used to protect the machine from the effect of failures. Therefore, the systems dynamics are defined by the following differential equation:

$$\dot{x}(t) = u(t) - d \qquad (1)$$

where $u(t)$ is the production rate at the instant of time t and d in the demand rate. The inventory and backlog stock implies a cost denoted by c^+ and c^- respectively. Once the production is restarted the machine is able to increase the inventory stock until a defined threshold or hedging point *Z*. If there is some backlogged stock the machine first will satisfy this backlogged demand before increasing the inventory stock. The objective is to determine the production rate *u(t)* that minimizes the total cost given by the inventory and backlog cost. Figure 1 presents the block diagram of the manufacturing system under study. Also we want to stress that the production system considers the following assumptions:

1) The machine is flexible
2) There is an infinite supply of raw parts
3) The machine is controlled by the Hedging Point Policy
4) Production planning is determined in a stochastic environment
5) The machine does not deteriorate after every failure
6) The machine produces only conforming parts
7) The failure and repair rate are constant in time

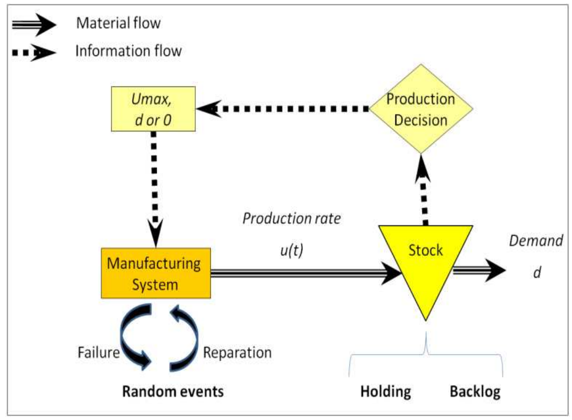

Figure 1: Block diagram of the manufacturing system

An important remark has to be made at this point, the production policy of this model can be characterized by a single threshold level *Z** called hedging point, as suggested by Kimemia and Gershwin (1983). In this case, the production rate is defined in function of the stock level, as denoted in equation (2):

$$u(x)^* = \begin{cases} u_{max} & if\ x(t) < Z^* \\ d & x(t) = Z^* \\ 0 & x(t) > Z^* \end{cases} \qquad (2)$$

More precisely equation (2) defines the Hedging Point Policy, which suggests that when the machine is in the operational state it should produce at the maximum possible rate U_{max}, if the inventory level *x(t)* is inferior than the hedging point Z^*. Furthermore, the machine should produce at demand rate d, if the inventory level is exactly equal to the threshold Z^*, and not produce at all if the inventory level exceeds the hedging point Z^*.

4 Proposed approach

Before presenting our proposed approach, let us point out that is well known in this area, that normally, the type of manufacturing systems such as the one presented in section 3, yields in general to intractable problems, due to its complex structure. Finding an analytical solution to this type of production systems is quite difficult. To cope with this difficulty, we adopt a simulation-based approach to determine the control parameter Z^*, identified in equation (2). The performance measure of interest is the expected total cost C*(Z)*, that consists of the sum of the inventory and backlog cost for a given capacity vector *Z*. This total cost is denoted in equation (3) as follows:

$$C(Z) = c^+ x^+ + c^- x^- \tag{3}$$

where c^+ and c^- are the unit cost for inventory and backlog, respectively. We can interpret the expressions x^+ and x^- are the time persistent statistics of the positive and negative stock, and as a matter of fact they are two of the outputs of our simulation models.

We are now in position to define that the first stage of our proposed approach, uses as base, the manufacturing system presented in section 3, and the production policy described by equation (2), to develop four simulation models using different modules of the ARENA software. It has to be noted that the objective is to determine the most efficient option, available in this software to simulate this type of manufacturing systems. Subsequently, in the second stage of the approach, several simulation replications are conducted, given by a defined test plan, to fit a regression equation. This process is repeated for the four simulation models. At this point we identified from the control policy, defined in equation (2), that there is one independent variable denoted by the hedging point *Z*, and one dependent variable denoted by the total cost *C*. The idea is to use a quadratic regression model to optimize the simulation results and estimate the total cost *C* as a function of the control parameter *Z*.

To this end, a statistical analysis of the simulation results is carried out. More precisely, a multifactor analysis of variance (ANOVA) is used to indicate if the regression model gives a good fit for the simulation data. This analysis also provides the proportion of the observed variability explained by the model that is denoted by the coefficient of determination. The most important is that we identify the parameters of the regression model that will allow us to estimate the optimal value of the hedging point $\hat{Z}*$, and the optimal cost $\hat{C}$. The regression model proposed is given by equation (4) as follows:

$$\hat{C}(Z) = \beta_0 + \beta_1 Z + \beta_2 Z^2 + \epsilon \tag{4}$$

where $(\beta_0, \beta_1, \beta_2)$, are unknown parameters and ϵ is the residual error. Non significant effects are ignored or added to the residual error. Using the software STATGRAPHICS we determine the unknown parameters based on the data collected from the simulation runs. Also we can verify the homogeneity of the variances and the residual normality condition. After that, we compare the results in terms of accuracy and time economy to identify the best simulation model.

5 Simulation models

We would like to emphasize that the main concern of this paper, is the development of four simulation models to identify the most accurate and fastest option, available in the ARENA software, for modeling systems such as the one presented in Section 2. Concepts as discrete and continuous simulation are used to develop our models. Let us remark that our first simulation model is based in discrete simulation. Typically, in the ARENA software, it exists two forms to build continuous models: one uses a file with C++ code to perform calculations for the state variable of the system, and the other form uses a VBA file for the calculations. We note that both forms are explored in this paper. Besides, there is other option in the ARENA software to model continuous flow of material, called Flow process, which also is examined in this research. The developed simulation models are composed by several modules and routines that follow a network structure. The models seek the same objective, as discussed in section 2, but they differ in their internal logic. The four simulation models developed are:

A. The Discrete model: where changes in the state variable, the level of the total stock, happen at discrete instants of time.

B. The Continuous VBA model: combines modules of the ARENA software with a Visual Basic routine (VBA) that calculates the value of the state equation (1).

C. The Continuous C++ model: combines ARENA modules with a C++ routine for the calculation of the state equation (1).

D. The Flow Process model: uses modules from the ARENA Flow Process template. It applies modules such as tanks, sensors and regulates to model the production system.

We close this section with a brief remark concerning our simulation models. We adopt a block diagram structure to facilitate the understanding of the logic of the different models. In the next section, is explained in detailed the logic applied in every model.

5.1 Discrete simulation model

We shall begin the explanation of the discrete model by defining that it was developed based in the block diagram presented in Figure 2. The distinctive feature of the discrete model is that all the random events and actualizations of the state variable, denoted by the stock level, are done in discrete points in time. Moreover, in this model, the simulated machine produces discrete parts. The respective description of the blocks used for this model is as follows:

1. The INITIALIZATION block read some inputs from an Excel file, such as the control parameter *Z*. Also it initializes the values of the rest of parameters required by the model, such as: the demand rate *d*, the means *MTTF* and *MTTR* for the random distributions of failures and repairs, and the maximum production rate u_{max}.
2. The DEMAND RATE block uses an expression derived from the demand *d* to simulate a delay of time that models the constant demand of products.
   The UNRELIABLE MACHINE block, simulates the production of discrete parts, it defines as resource of the model the machine, and it uses the production time defined by the Control Policy block. The machine is unreliable due to the random failures and repairs defined by the MTTF and MTTR.
3. The CONTROL POLICY block, is based on equation (2), and it defines the production time required to produce a part. To implement this block it is needed the value of the current stock. Also a FLAG is used to restart the production if the current level stock is inferior to the production threshold *Z*.
4. The UPDATE INVENTORY LEVELS block actualizes the level of the positive and negative stock after every discrete event. In this case, after the production of a part, at the arrival of a demand or at failures.
5. The TIME ADVANCE block actualizes the current time based on a schedule of discrete events, which are given by the exponential distribution used to simulate failures and repairs.
6. The OUTPUT block sends to an external file three outputs of interest, such as; time persistent statistic of the positive and negative stock, and the durations expressed in seconds required to run a replication of the simulation model.

The simulation runs until the simulation time *Tsim* reaches the stopping time *Tend*. This time is defined as the amount of time needed to ensure steady state conditions.

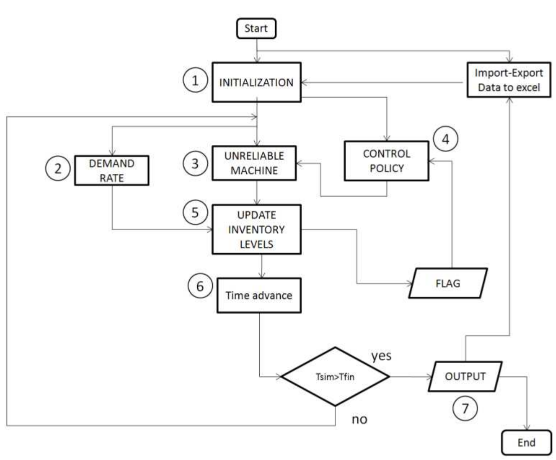

Figure 2: Simulation block diagram for the Discrete model

5.2 Continuous simulation models

With regard to the continuous models, the block diagram applied for the C++ and VBA models is presented in Figure 3. The distinction of the continuous simulation models is that they use the differential equation (1) to describe changes in the stock level. The difference between our two continuous simulation models, is based in the type of programming language used for the calculations of the differential equation (1). The name of the model indicates if we use a C++ or a VBA file for the calculations. We can understand the logic of these models as the continuous flow of material that is stopped by random failures of the machine. The description of the different block used in these models is as follows:

1. The INITIALIZATION block reads from an Excel file the required input and initializes the values of the rest of parameters needed for the simulation, such as; *Z*, *d*, *MTTF*,*MTTR* and u_{max}.

2. The FAILURES AND REPAIRS block changes the value of a binary variable θ to indicate the presence of failures and repairs, in the systems dynamics denoted by equation (1).

3. The CONTROL POLICY block defines the production rate that is implemented through the use of observation networks that raise a flag whenever the production threshold *Z* is crossed, as denoted in equation (2).

4. The STATE EQUATIONS block is defined in a C++ or VBA file attached to the continuous models. It describes the dynamics of the inventory using the differential equation (1), it needs for its operation; the production rate set by the Control Policy block, the constant demand *d*, and the value of the binary variable from the Failures and Repairs block.

5. The TIME ADVANCE blocks changes the current time based on a schedule of discrete events, the value of continuous variables, threshold crossing events and time step specifications.

6. The UPDATE INVENTORY LEVELS block traces the variations of the positive and negative stock for the chosen step size. The cumulative variables are integrated using the Runge-Kutta-Fehlberg method.

7. The OUTPUT block sends to an external file three outputs of interest, where the total cost is calculated using equation (3).

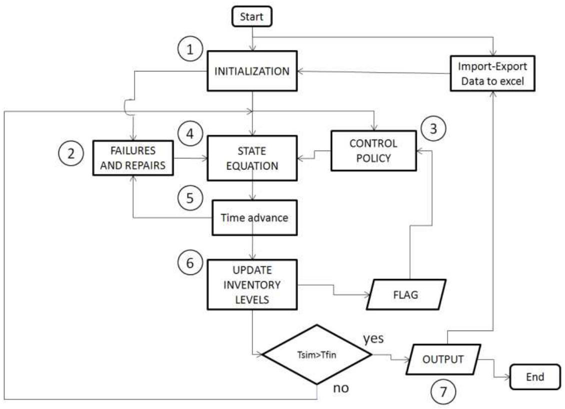

Figure 3: Simulation block diagram for the Continuous models

5.3 Flow process simulation model

In essence, the Flow Process model simulates the production systems as a continuous flow of material, and the difference with respect to the continuous models is that the Flow Process model does not use the differential equation (1) to simulate changes in the stock level. Instead, the Flow Process model uses a graphical approach, based on tanks and regulators modules that represent in a visual form the production system. The logic for this model divides the total stock in two parts; we use a tank to simulate the positive stock and other tank for the negative stock. The flow between tanks is controlled by a series of sensors and regulators according to the hedging point policy. The simulation block diagram for the Flow Process model is presented in Figure 4, and the description for its blocks is as follows:

1. The INITIALIZATION block reads from an Excel file the required inputs and initializes the value of the variables that are used to control the flow of material to the tanks.

2. The CONROL POLYCY block uses the values of current stock of the tanks, detected by the Sensors 1 block, to implement the control policy. This block changes, opens or closes the flow of the valve defined in the Production Regulator 1 block according to equation (2).

3. The PRODUCTION REGULATOR 1 block simulates the production of material, it is a valve located in the tank 1 of positive stock that adds material to this tank. It has a flow capacity defined by the maximum production rate u_{max}, this regulator is opened or closed by the random failures or other events detected by the sensors block.

4. The SENSORS TANK 1 block, includes tree sensors located in the tank of positive stock and they are used to detect the three conditions required by the production policy of equation (2). Also it indicates the moment to change the flow of material to tank 2.

5. The TANK 1 block, is the tank devoted to the positive stock, it has installed sensors to implement the control policy and regulators to simulate the production and demand of products.

6. The DEMAND REGULATOR 1 block, simulates the constant demand of product, it consist of a valve that subtracts material from the tank of positive stock at a flow rate equal to the demand *d*. This activity will stop until the FLAG 1 is raised, which indicates the moment when it is needed to redirect the flow of material to tank 2 of negative stock.

7. The UPDATE INVENTORY LEVELS 1 block, actualizes the levels of the stock after every failure, reparation or any other event detected by the sensors.

8. The FAILURES block simulates the random failures and repairs that influence the flow of material in the Production Regulator 1. These events close or open this valve depending of the events experienced.

9. The TIME ADVANCE block, actualizes the current time according to a schedule of discrete events.

10. The LOGIC FOR TANK 2 block, represents the series of modules required to simulate the negative stock. The logic for this tank is very similar to the one of the positive stock. The difference with this tank 2 is that we use only one sensor to detect the moment to change the flow to tank 1. The rest of module works in the same way as explained for tank 1.

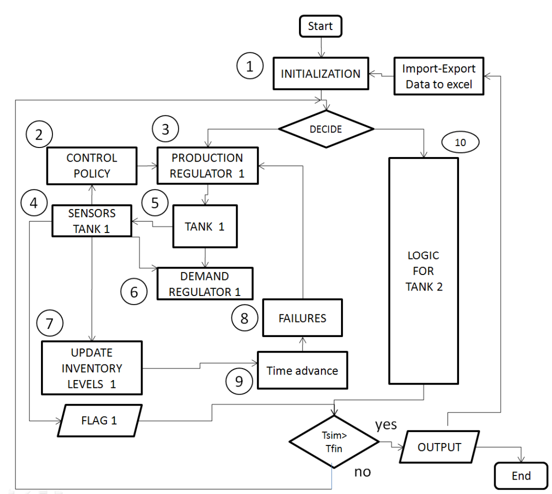

Figure 4: Simulation block diagram for the Flow Process model

At this point we have completed the explanation of the block diagrams of our four simulation models, it remains to determine which of these models is the best option to simulate the production system under consideration. The next section provides numerical examples to address this issue.

6 Simulation results analysis

Our primary concern in this section is to examine our four simulation models and eventually, determine the best model, based in their performance assessment. In this section, we have two objectives: 1) verify the accuracy of the results obtained, this is done by calculating the error in the total cost for every model. 2) measure the computational time required to complete a simulation run and detect any time economy with respect to the discrete model. We shall proceed by defining that from a series of off-line replications, the simulation run length was defined as 100,000 time units. This period ensures steady state condition, and the warm-up period is defined as 10,000 time units. The input parameters to be applied in the simulation models are presented in Table 1. According to the given values of the MTTF and MTTR, the system has an availability of 91%, this means that the machine is be capable to fulfill the demand of product, since the condition of feasibility is satisfied, ($U_{max}\,x\;0.91 > d$).

MTTF	*MTTR*	*Avail.*	*d*	U_{max}	C^-	C^+
1	0.1	0.91	100	120	1	10

Table 1. Simulation data

We concentrate next in the determination of the unknown coefficients of the regression model (4), To resolve this problem, an interval for the production threshold *Z* was defined. Once again from off-line simulations we establish that *Z* varies from 3 to 60. Then, we use an increment of 3 units in this interval to have a total of 20 observations. We run four replications for these observations, using for every replication a different random number stream. This is done with the purpose to have enough data to determine the unknown coefficients, and ensure independence in the results. In total, every simulation model, was run (20x4) =80 replications. Moreover, as discussed in section 4, ANOVA analyses are carried out to the simulation results, applying the software STATGRAPHICS. For instances, Table 2 illustrates the ANOVA table for the data collected from the continuous C++ simulation model. The statistic analysis shows that the quadratic regression model is a good fit for the data since the P-value satisfies the condition (P-value < 0.05). Furthermore, we note that the variability of the cost as a function of the production threshold *Z*, is well represented by the regression model, since the R^2 adjusted value is 96.32%. This implies that about 97% of the total variability is explained by the model. The ANOVA table for the Discrete, Continuous VBA and Flow Process model lead to similar conclusions, with R^2 adjusted values of 96.6%, 96.32%, 96.36% respectively.

Source	*SS*	*Df*	*MS*	*F-Ratio*	*P-Value*
Model	14841,2	2	7420,61	1010,09	0,0000
Residual	565,679	77	7,34649		
Total (Corr.)	15406,9	79			

Table 2. ANOVA table for the Continuous C++ model

It is worth repeating that we use the software STATGRAPHICS to define the unknown parameter of the regression model. This statistical software has been successfully applied in similar context as in Berthaut et al. (2010). Once determined the coefficients, the corresponding cost functions are defined as follows:

$$C_{DISC} = 103.201 - 3.07156Z + 0.0388909Z^2 \qquad (5)$$

$$C_{C++} = 98.461 - 2.86223Z + 0.0367528Z^2 \qquad (6)$$

$$C_{VBA} = 98.4602 - 2.8622Z + 0.0367522Z^2 \qquad (7)$$

$$C_{FLOW} = 95.5453 - 2.71367Z + 0.035007Z^2 \qquad (8)$$

where C_{DISC}, C_{C++}, C_{VBA} and C_{FLOW} denote the cost functions of the discrete, continuous model with C++ inserts, continuous model that applies VBA and the Flow process model, respectively. It follows from equation (6) and equation (7), that their coefficients are very close. This indicates that both models provide similar results, since the code in the C++ and VBA files is the same. Conversely, it is observed a difference in the coefficients of the Discrete and Flow process model. Examining Figure 5 we observe a well fit for the regression model by the data of the continuous C++ model. The optimal production threshold $\hat{Z}*$ is calculated with a numeric resolution method that consist in applying the second derivative to the cost function $\hat{C}(Z)$, and then solve the expression obtained with respect to *Z*. Following this resolution, we estimate the optimal threshold as $\hat{Z}*$=38.93 for the Continuous C++ model. This value is the optimal control parameter that ensures an optimal cost and that should be applied to the manufacturing system.

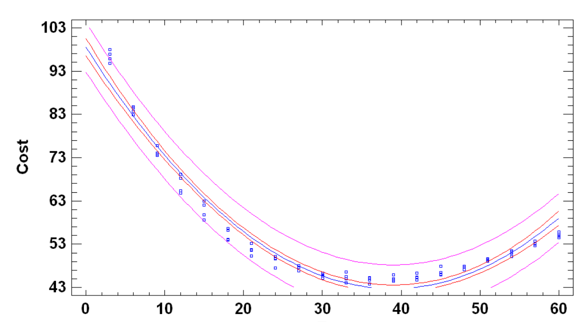

Figure 5: Cost graphic for the Continuous C++ model

We summarized the obtained optimal control parameters and optimal cost of the regression models (5-8) in Table 3.

Model	*Discrete.*	*Cont.C++*	*Cont.VBA*	*Flow Pr.*
Estimated threshold, $\hat{Z}*$	39.48	38.93	38.93	38.75
R^2 (%)	96.6	96.32	96.32	96.36

Table 3. Results from the regression models

For the purpose of testing the accuracy of the results, we compare the simulation models with a reliable pattern, denoted by an analytical solution. Fortunately, the manufacturing systems consisting of one machine producing one part type, has been studied previously by Bielecki and Kumar (1988), where they found an analytical solution for the optimal threshold $Z*$ and optimal total cost $C*$. Based on this, we determine the accuracy of our approach by calculating the relative error between the total cost given by the regression models (5-8) and the cost reported by the analytical solution. For this purpose, we use the data of table 1 as input for the analytical model to calculate the optimal cost $C*$ and its respective optimal threshold $Z*$. Then, we run 30 extra replications using as input for the simulation models, the threshold $\hat{Z}*$, estimated by the regression models. With the obtained data from these extra replications, we proceed to calculate the average total cost $\bar{C}^*$. This process is repeated for the four simulation models. After that, we calculate the relative error between the optimal cost $C*$ given by the analytical model and the cost $\bar{C}^*$ obtained by the regression of simulation results.

The next step in our analysis involves the discussion about the results of table 4, where we present the value of the optimal threshold $Z*$ and the optimal cost $C*$ calculated with the analytical solution., also we find the estimated threshold $\hat{Z}*$ given by the regression models and their average total cost $\bar{C}*$. By analyzing table 4, it is therefore evident that the Discrete model presents the biggest relative error of 0.69%. The Continuous C++ and Continuous VBA model reported a small error of 0.20%; we think that this value is similar, because in both models it is applied the same code in the routine that calculates the dynamics of the state equation. The Flow Process model presents a very small error of -0.28%. We infer that the discrete model presents the biggest error because the analytical solution was originally conceived to model a continuous system and our discrete simulation models is only an approximation to the continuous case. This observation explains the difference in the total cost reported by the discrete simulation model. Examining the results of table 4, we note that the Continuous simulation models (C++ and VBA) reports small errors compared with the analytical solution, because both, the continuous models and the analytical solution address the same system, under similar assumptions.

We next turn our attention to the fact that from the results of table 4, we observe a difference between the value of the production threshold obtained by simulation with respect to the threshold reported by the analytical solution. The intuition behind this difference is that if we observe Figure 5, it is readily apparent that the zone, where is located the optimum threshold $\hat{Z}*$, indicated that the graph of the cost function is a bit flat. Consequently, this condition implies that in the optimal zone, even big changes in the production threshold Z, report small changes in the total cost, and so is quite difficult for the regression model, to capture this variations. In general, this phenomenon explains the difference found in the production thresholds of table 4. Nevertheless, after comparing the optimal cost and calculating the relative error, we notice that our proposed approach based in the combination of the regression model and simulation works well. A prove of that is that the cost difference between our approach and the analytical solution is negligible, with a relative error below 1%.

Model	$\hat{Z}*$ *(Sim.)*	$Z*$ *(Ana)*	$\bar{C}*$ *(Sim.)*	$C*$ *(Ana)*	*Relative error (%)*
Discrete.	39.48	35.85	45.23	44.92	0.69 %
Cont. C++	38.93	35.85	45.01	44.92	0.20 %
Cont.VBA	38.93	35.85	45.01	44.92	0.20 %
Flow Pr.	38.75	35.85	44.79	44.92	-0.28 %

Table 4. Comparison of the simulation based approach and analytical results

As a matter of interest, we present in table 5 some complementary indices of the simulation models, such as: the time persistent statistics of the inventory and backlog stock and their respective costs. The difference, observed in these indices for all the models, is very small. In future applications, we can modify the simulation models to report other indices of interest such as; the availability of the system or the number of failures. Based on the obtained results, we can conclude that in terms of accuracy, the Continuous C++ , Continuous VBA and Flow Process models performs better than the Discrete model, if we compare their results with the analytical solution.

Model	X^+	X^-	*Inventory cost ($/day)*	*Backlog cost ($/day)*
Discrete.	29.76	1.54	29.76	15.47
Cont. C++	29.57	1.54	29.57	15.44
Cont.VBA	29.57	1.54	29.57	15.44
Flow Pr.	29.23	1.55	29.23	15.55

Table 5. Complementary performance indices measured by simulation

The second parameter of our interest deals with the computational time. We want to determine not only the most accurate but also the fastest option to simulate our manufacturing system. To accomplish this, we included a VBA routine to all the simulation models, that calculates the required computational time, defined in seconds, to simulate a replication run. The times reported in table 6 indicate for every simulation model, the average computational time of 80 replications. The time economy was calculated using as base for the indicator, the time of the discrete model. From the results of table 6, we notice that the slowest model of the group is the Continuous VBA model, since requires in average 782.95 seconds to run a replication. This result is even seven times slower than the time obtained with the discrete model. In contrast, it is a remarkable fact that the computational time of the Continuous C++ model reports in average an outstanding time of 1.18 seconds, this means that compared with the discrete model it reports a time economy of 98.92%. This average is an excellent result, because it means that we obtain output data in almost 1% of the time required by the Discrete model. Besides, the benefit of the Continuous C++ models is double because the error reported in the total cost is much smaller than the discrete model.

Model	*Discrete*	*Cont. C++*	*Cont.VBA*	*Flow Pr.*
Computational Time (sec.)	110.55	1.18	782.95	1.34
Time Economy (%)	-	98.92%	none	98.78%

Table 6. Time economy

Following the discussion about the results of table 6, we notice that the Flow Process model also runs extremely fast, it requires in average 1.34 seconds to simulate a replication run. This result implies also an excellent time economy, that in this cases is 98.78%. Therefore, the advantage of the Flow process model is that is extremely fast and the difference in the total cost is as small as the Continuous C++ model. Based on these results we can conclude that in terms of time economy the Continuous C++ and the Flow Process models are the fastest options, meanwhile the worse performance of the analysis was obtained by the Continuous VBA model.

Let us now discus about the remarkable difference between the computational time of the simulation models. We state that this difference in time, is based on the particular characteristic features of the programming language, applied in the models. For example, In general, Visual Basic for Applications (VBA) was originally designed to be easily learned and used by beginner programmers. VBA facilitates a friendly interaction between the user and the application by means of a graphical user interface. Traditionally, this VBA interface is based on images, buttons, text boxes, etc., rather than text commands. The compilation of VBA files with this graphical interface generates more code lines at the machine language level than a simple C++ file. Thus, this increment in code line has as a consequence that the computational time increases considerably for the Continuous VBA model.

With respect to the Flow Process model, in practical terms, the Flow process template provides a graphical interface to model continuous applications. It uses a visual approach to simulate the flow of material, instead of using differential equations as the continuous models. Furthermore, the time economy of the Flow process model is important and this model runs extremely fast because internally its modules are constructed using C++ code. Normally, C++ code takes only seconds to run. The technical advantage of the Flow process template is that is relatively simple to build a simulation model and it gives an acceptable accuracy on the results at validating with the analytical solution. From what has been said regarding the C++ lines, we can state that the models which uses C++ code in its operation, such as; the continuous C++ model and the Flow process model, reported the fastest computational time, because of the reduced number of code lines generated at the machine language level.

One comment about the importance of time economy is based on the total of 320 simulation replications needed in this research. The Continuous VBA model required a total of 17.4 hours to complete 80 replications, the Discrete model needed 2.5 hours and the Continuous C++ and Flow Process models needed only 2 minutes to complete the same work. The benefit of the Continuous C++ model is evident, as seen in this section, because in a fraction of time we can complete the same amount of simulation runs with even a better accuracy than the other models. In the future this time economy will be very useful to model more complex manufacturing systems involving other factors and phenomenon. For instance, a possible future scenario would be to apply a Factorial Design 3^4 for a given simulation model to optimize for instance four control parameters. If we replicate 5 times the 3^4 design, this means a total of (3^4x5)= 405 simulation runs. Besides, if we assume that the Discrete model would take 300 seconds per run, we will need in total 121,500 seconds or 1.4 days, to complete the replications. For the same case, the Continuous VBA would finish the simulations after 9.84 days of continuous work, but if we instead develop a Continuous or a Flow Process model, we will need only 20.25 minutes to find the same results and with even a better accuracy. Therefore, it is clear that the economy is huge not only in time but also in the resources implied in the simulations. We terminate our discussion of the models, by pointing out that the practical advantage of the Continuous C++ model is remarkable and the results obtained in this research are encouraging and satisfactory for the development of future and more complex applications in the area.

7 Conclusion

In this paper, the production control problem of a manufacturing system consisting on one machine producing one part type subject to random failures, was used to test the capabilities of the Arena simulation software. Due to the mathematical complexity of this kind of manufacturing systems alternative solutions based in computer simulation are justified. We used the production threshold as input to develop different simulation models using the software ARENA. The idea was to pinpoint the best option available in this software to simulate such manufacturing system. We analyzed four simulation models and we compared their performance with analytical results. Two indicators were analyzed, the computational time and the difference in the total cost. We found that the Continuous C++ and Flow Process models reported the biggest time economy by a ratio of hundreds compared with the Discrete and Continuous VBA models. We believe that this time economy will be improved even more as the computers evolve over time, and this condition will enable the optimization of long lines in feasible time frames, problem that remains complicated at the moment. The accuracy of the results, tells us that the simulation models provide solutions close to the analytical solution, and the difference observed in the results is less than 1%. Future work will include the improvement of the Flow process model, since we experienced some problems at interacting with some of its modules and the Excel file. We found that for this model a single replication run is completed rapidly, but manual work is needed to run several replications. The results of this paper is the base for the development of future applications considering more complex manufacturing systems, for instance, in future contributions we can examine factors such as; subcontracting, preventive maintenance, corrective maintenance, ageing, etc., just to mention some of them. To examine this kind of systems, normally, it is needed a great number of simulation runs, thus the time economy and the accuracy of the simulation model applied, plays a key role for us. So far, based on the experience gained with this research, we can conclude that the Continuous C++ is the most efficient option to simulate the type of manufacturing systems presented, and that the Flow Process model is also a good alternative but needs to be improved more at performing several replications runs.

References

Akella, R., Kumar, P.R., (1986). Optimal control of production rate in a failure-prone manufacturing system. *IEEE Trans*. Automat. Contr., AC-31, 116–126.

Belecki, T., Kumar, P. R., (1988). Optimality of zero inventory policies for unreliable manufacturing systems, *Operations Research,* vol. 36, pp. 532-541, Jul.-Aug. 1988.

Berthaut, F., Gharbi, A., Kenne, J.P., Boulet, J.K., (2010). Improved joint preventive maintenance and hedging point policy. *International Journal of Production Economics, Elsevier,* 127, pp 60-72.

Berthaut, F., Gharbi, A., Dhouib K. (2011). Joint modified block replacement and production/inventory control policy for a failure-prone manufacturing cell. *Omega, Elsevier,* 39, 642-654.

Gharbi, A., Kenne, J. P., (2003). Optimal production control problem in stochastic multiple-product multiple-machine manufacturing systems, *IEEE Transactions*, vol. 35, n. 10, pp. 941-52.

Hajji, A., Gharbi, A., Kenne, J.P. (2009). Joint replenishment and manufacturing activities control in a two stage unreliable supply chain. *International Journal of Production Research*, 47:12, 3231-3251.

Kimemia, J.G., Gershwin, S.B., (1983). An algorithm for the computer control production in flexible manufacturing systems. *IEEE Trans*., AC-15, 353–362.

Kenne, J.P., Gharbi, A., Boukas, E. K. (1997). Control policy simulation based on machine age in a failure prone one-machine, one-product manufacturing system. *International Journal of Production Research* 35:5, 1431-1445.

Kenne, J.P., Gharbi, A. (1999). Experimental design in production and maintenance control problem of a single machine, single product manufacturing system. *International Journal of Production Res.*, vol. 37, no. 3, 621-637.

Kenne, J.P., Gharbi, A. (2000). Production planning problem in manufacturing systems with general failure and repair time distributions. *Production Planning & Control*, vol. 11, no. 6, 581-588.

Lavoie, P., Kenne, J.P., Gharbi, A. (2007). Production control and combined discrete/continuous simulation modeling in failure-prone transfer lines. *International Journal of Production Research,* 45 :24, 5667-5685.

Lavoie, P., Gharbi, A., Kenne, J.P. (2009) A comparative study of pull control mechanisms for unreliable homogenous transfer lines. *International Journal of Production Economic,* doi: 10.1016.

Rossetti, M. (2010). *Simulation modeling and Arena*. John Wiley & Sons, Inc. United States of America.

Sethi, S.P., Zhang, Q., (1994) Hierarchical Decision Making in Stochastic Manufacturing Systems. Birkhauser, Boston, MA.