

# Winning Strategy for the Littlefield Simulation Game

---

*A System Dynamics Approach*

**A Major Qualifying Project /Interactive Qualifying Project Report**

**Submitted to the Faculty**

**Of the**

**WORCESTER POLYTECHNIC INSTITUTE**

**In partial fulfillment of the requirements for the**

**Degree of Bachelor of Science**

**In Economic Sciences**

**By**

---

**Qiyang Fan**

---

**Mengjie Liu**

---

**Murtaza Turab Thahirally**

---

**Siqi Wang**

**Submitted 10/14/2011**

**Approved:**

---

**Professor Khalid Saeed, Advisor**

---

**Professor Oleg Pavlov, Advisor**

## **Abstract**

Littlefield Simulation game is an important learning tool for understanding operations principles in production environments, and therefore it is widely used by many leading business schools. This project attempts to model this game using system dynamics approach, which allows realistic representation of the production system of Littlefield Simulation. It also creates an environment for comparison of the consequences of various policies.

## TABLE OF CONTENTS

<b>ABSTRACT .....</b>	<b>1</b>
<b>1.0 INTRODUCTION .....</b>	<b>5</b>
<b>2.0 BACKGROUND.....</b>	<b>7</b>
2.1 LITTLEFIELD TECHNOLOGIES AS A TEACHING TOOL .....	7
2.2 STUDENT RESEARCH AND EXPERIENCE .....	7
2.3 LIMITATIONS OF PREVIOUS RESEARCH AND STRATEGIES.....	8
2.4 SYSTEM DYNAMICS METHOD AND ITS BENEFITS .....	8
2.5 OPERATIONS MANAGEMENT TECHNIQUES AS POLICIES .....	9
<b>3.0 PROBLEM EXAMINATION .....</b>	<b>11</b>
3.1 GAME-PLAY EXPERIENCE AND PROBLEM DESCRIPTION .....	11
3.2 DATA REVIEW .....	12
3.2.1 <i>Station Queues</i> .....	12
3.2.2 <i>Station Utilization</i> .....	13
3.2.3 <i>Inventory</i> .....	14
3.2.4 <i>Completed Jobs</i> .....	15
3.2.5 <i>Lead Time</i> .....	16
3.2.6 <i>Revenue</i> .....	17
3.3 DYNAMIC HYPOTHESIS .....	18
<b>4.0 SYSTEM DYNAMICS MODEL DESCRIPTION AND TESTING .....</b>	<b>21</b>
4.1 OVERVIEW OF THE SECTORS .....	21
4.2 SECTOR DESCRIPTIONS .....	22
4.2.1 <i>Input to the system</i> .....	22
4.2.2 <i>Cash Constraints</i> .....	25
4.2.3 <i>Financials</i> .....	29
4.2.4 <i>Contracts</i> .....	33
4.2.5 <i>Average Lead time</i> .....	36
4.2.6 <i>Machines</i> .....	37
4.2.7 <i>Stations</i> .....	40
4.3 MODEL TESTING .....	46
4.3.1 <i>Extreme Value Testing</i> .....	46
4.3.1.1 <i>Incoming Customer Order</i> .....	46
4.3.1.2 <i>Desired Number of Machines</i> .....	49

4.3.1.3 Reorder Level and Reorder Quantity.....	50
4.3.2 <i>Input Response Testing</i> .....	53
4.3.2.1 Constant Incoming orders .....	53
4.3.2.2 Step Response.....	54
4.3.2.3 Sine Wave Input .....	55
4.3.2.4 Random Input Response .....	56
<b>5.0 GAME-PLAYING STRATEGIES FOR MAXIMIZING PROFIT .....</b>	<b>58</b>
5.1 ANALYSIS OF CURRENT STRATEGY .....	58
5.1.1 <i>Initial Condition and Input to the System</i> .....	58
5.1.2 <i>Contract Selection</i> .....	59
5.1.3 <i>Ideal Number of Machines to Purchase</i> .....	60
5.1.4 <i>Re-order Level</i> .....	61
5.1.5 <i>Re-order Quantity</i> .....	62
5.1.6 <i>Machine Priority</i> .....	63
5.2 PREDICTIONS OF MODEL BEHAVIOR .....	63
5.3 PROFIT-MAXIMIZING STRATEGY IMPLEMENTATION AND TESTING .....	65
5.3.1 <i>Initial Run with Reactive Strategy</i> .....	66
5.3.2 <i>Switching to Contract 3</i> .....	67
5.3.3 <i>Implementing optimal re-order level and re-order quantity</i> .....	70
5.3.3.1 Optimum Reorder Level .....	71
5.3.3.2 Optimum Reorder Quantity .....	72
5.3.3.3 Refining reorder level and reorder quantity for the last 50 days....	73
5.3.4 <i>The Number of Machines</i> .....	75
5.3.5 <i>Verifying Optimal Strategy with the Random Input</i> .....	77
<b>6.0 LIMITATIONS AND FUTURE WORK .....</b>	<b>78</b>
6.1 LIMITATIONS .....	78
6.2 FUTURE WORK .....	78
<b>7.0 CONCLUSION .....</b>	<b>80</b>
<b>8.0 WORK CITED .....</b>	<b>81</b>
<b>9.0 APPENDIX .....</b>	<b>83</b>
9.1 EQUATIONS .....	83
9.2 USER INTERFACE .....	90
9.3 LITTLEFIELD GAME PARAMETERS .....	91

## TABLE OF FIGURES

FIGURE 1-DAILY AVERAGE NUMBER OF KITS FOR STATIONS.....	12
FIGURE 2-UTILIZATION FOR STATION1 .....	13
FIGURE 3 INVENTORY LEVEL.....	14
FIGURE 4 NUMBER OF JOBS COMPLETED FOR EACH CONTRACT.....	15
FIGURE 5 NUMBER OF JOBS COMPLETED .....	15
FIGURE 6 DAILY AVERAGE JOB LEAD TIME BY CONTRACT TYPE .....	16
FIGURE 7 DAILY AVERAGE REVENUE BY CONTRACT TYPE .....	17
FIGURE 8 DAILY AVERAGE REVENUE.....	17
FIGURE 9 DYNAMIC HYPOTHESIS MODEL .....	19
FIGURE 10 OVERALL SECTOR DIAGRAM .....	21
FIGURE 11 INPUT OF THE SYSTEM.....	23
FIGURE 12 EFFECT OF RAW MATERIAL RATIO ON JOB RELEASE .....	25
FIGURE 13 CASH CONSTRAINTS.....	26
FIGURE 14 CASH CONSTRAINT ON RAW MATERIALS.....	28
FIGURE 15 CASH CONSTRAINTS ON MACHINES .....	29
FIGURE 16 FINANCIAL SECTOR .....	30
FIGURE 17 CONTRACTS SECTOR .....	33
FIGURE 18 -CONTRACT TYPE 1 AND LEAD TIME.....	35
FIGURE 19- CONTRACT TYPE 2 AND LEAD TIME.....	35
FIGURE 20 -CONTRACT TYPE 3 AND LEAD TIME.....	36
FIGURE 21 LEAD TIME SECTOR.....	36
FIGURE 22 MACHINES SECTOR.....	38
FIGURE 23 STATIONS SECTOR .....	41
FIGURE 24 CASH LEVEL FOR EXTREME VALUES OF CUSTOMER ORDERS .....	48
FIGURE 25 -LEAD TIME FOR EXTREME VALUES OF CUSTOMER ORDERS .....	48
FIGURE 26 CASH LEVEL FOR EXTREME VALUES OF DESIRED MACHINES .....	50
FIGURE 27 LEAD TIME FOR EXTREME VALUES OF DESIRED MACHINES .....	50
FIGURE 28 CASH LEVEL FOR ZERO REORDER LEVEL AND REORDER QUANTITY.....	52
FIGURE 29 LEAD TIME FOR ZERO REORDER LEVEL AND REORDER QUANTITY .....	52
FIGURE 30 CASH LEVEL WITH CONSTANT INCOMING CUSTOMER ORDERS.....	54
FIGURE 31 CASH LEVEL STEP RESPONSE.....	55
FIGURE 32 CASH LEVEL SINE WAVE RESPONSE .....	56
FIGURE 33 CASH RESPONSE TO A RANDOM INPUT .....	57
FIGURE 34 INCOMING CUSTOMER ORDERS PER DAY .....	58
FIGURE 35-MEAN, STANDARD DEVIATION AND RANGE OF INCOMING CUSTOMER ORDER PER DAY .....	59
FIGURE 36- ORDER DISTRIBUTION .....	59
FIGURE 37 CASH LEVEL WITH A REACTIVE STRATEGY .....	66
FIGURE 38 CONTRACT TYPE CHOSEN WHEN POLICY IS NOT USED .....	67
FIGURE 39 POLICY SECTOR.....	68
FIGURE 40 CONTRACT TYPE 1 REVENUE GRAPH.....	68
FIGURE 41 CONTRACT TYPE 3 REVENUE GRAPH.....	69
FIGURE 42 CASH LEVEL WHEN CONTRACT 3 IS CHOSEN .....	70
FIGURE 43- IMPLEMENTING ECONOMIC REORDER QUANTITY AND ECONOMIC REORDER LEVEL.....	71
FIGURE 44- CHANGING THE REORDER LEVEL.....	72
FIGURE 45 CHOOSING THE OPTIMUM REORDER QUANTITY.....	73
FIGURE 46 REFINING REORDER LEVEL AND REORDER QUANTITY FOR THE LAST 50 DAYS.....	74
FIGURE 47- AVERAGE LEAD TIME BEFORE CHANGING DESIRED NUMBER OF MACHINES.....	75
FIGURE 48 LEAD TIME BEFORE AND AFTER CHANGING THE NUMBER OF MACHINES .....	76
FIGURE 49 CASH LEVEL BEFORE AND AFTER CHANGING THE DESIRED NUMBER OF MACHINES .....	76
FIGURE 50 CASH LEVEL WHEN CUSTOMER ORDERS ARE RANDOM.....	77

## 1.0 Introduction

Littlefield Simulation is a game widely used in management courses that replicates a manufacturer's decision making mechanism. In the game, teams are challenged to optimize the system and maximize cash flow for Littlefield Technologies, a factory that assembles Digital Satellite System Receivers from electronic kits. The electronic kits are acquired from the supplier. The assembly process consists of four steps carried out at three stations. The last step is inspection. The configurations that participants could adjust include the contract type, inventory reorder point, reorder quantity, number of machines on each of the three stations, and the machine strategies, which is the priority of the workflow. The system receives a random number of customer orders each day, which ranges between 3 and 24. The revenue is affected by the average lead time to process the customer order, contract type the player chooses, and the inventory reordering settings.

A System Dynamics method is used to analyze the game, where the data retrieved from the game is used to create a reference mode. The next step is to create a dynamic hypothesis, which is a set of casual feedback loops, to represent the production system and the players' decision mechanisms. A System Dynamics model is built to incorporate the dynamics hypothesis and to replicate the reference mode. The last step is to design policies to achieve optimized sustainable profit growth through analyzing trends in the game and patterns generated by the model.

The paper elaborates on the System Dynamics model and the steps we went through to design the winning policy. Considering the scale and level of complexity of our model, certain statistical tools were utilized to generate optimized input parameters, which helps us efficiently

test and experiment with the model, and also to produce more realistic patterns of various policies.

The paper first analyzes the data retrieved from the game and discusses the chosen methodology with justifications. Afterwards, the model is described in detail, along with the tests performed to verify the validity. Thereafter the policy is described and tested again. The final section of the paper outlines future work that can be done and the limitations of Littlefield Simulation and the model we constructed.

## 2.0 Background

### 2.1 Littlefield Technologies as a Teaching Tool

In 1996, Stanford University adopted Littlefield Technologies in its MBA courses. "Littlefield Technologies has it all --- the good, the bad, and the lessons most managers can only learn the hard way," according to the article "Factory Simulation Program Wins Wide Recognition" on the Stanford Graduate School of Business' website. Compared to the traditional teaching methods, Littlefield Technologies is more affordable, accessible and effective.

“...the simulation games are strongly linked to class concepts- the students are forced to think logically about the problems that they are facing and they learn from iterative experimentation. “ said Julia Miyaoka, professor at San Francisco State University, who used Littlefield Technologies in both her undergraduate and MBA classes.

Littlefield Technologies draws knowledge on topics such as capacity, utilization, queuing, scheduling and inventory. It is currently adopted by institutions globally and students often compete as groups to run the factory in the most efficient and profitable manner.

### 2.2 Student Research and Experience

Many students enjoyed the experience of playing Littlefield Technologies. Winner of the 2007 MIT Sloan Operations Simulation Competition, the “yalewillwin” team built a demand model to forecast the demand trend and a queuing model to experiment with different settings, such as increasing the number of machines, changing some of the lot sizes. Other winning strategies mentioned by the team members include re-evaluating final inventory level to avoid waste or shortage of inventory, regular attention to the daily operations of manufacturing facility and team dynamics.



Another team shared their lessons learned from Littlefield Technologies. They performed detailed analysis of demand, contract types, capacity settings and reorder point using basic operations and mathematical formulas. The four major mistakes they concluded are: the incorrect documentation, which leads to over-purchasing of machines, the waste of order fees due to frequent orders, the lack of inventories in some of the days due to missing reorder time and the failure to evaluate the inventory in the last few days.

### **2.3 Limitations of Previous Research and Strategies**

After taking a closer look at the existing reports on strategies done by other students, we concluded that these strategies, regardless of results, are experience-based, reached through trial and error, or in other words “reactive” instead of “proactive”. Although many operations management concepts and mathematical formulas can provide guidelines for decision making, it is difficult to apply a rigid algorithm to respond to changes in the complex system.

During the research process, our team tried to understand the mechanisms contained in the game by dividing the mechanism or model into sectors. After studying the reactions among the sectors, each sector is analyzed individually.

### **2.4 System Dynamics Method and Its Benefits**

System dynamics is a methodology and a mathematical modeling technique for framing, understanding, and discussing complex issues and problems. It was created in the mid-1950s by Professor Jay Forrester of MIT during his research on how science and engineering can contribute in determining the success or failure of a corporation. Casual loop diagrams, stock and flow diagrams and equations are the basic components of a system dynamic model.

System dynamics has a wide range of applications as it can usually be applied whenever a problem can be expressed as variable behaviors through time. Its applications can be expanded

into many fields such as medicine, urban studies, global studies, environmental studies, information science, history, economic behavior, physics and many more to help with policy analysis and design.

Applying System Dynamics approach doesn't mean abandoning all traditional operations management techniques in this research project. In fact, we have used certain operations management techniques to design policies, which work as the rules that govern our decisions and strategies. In other words, system dynamics is tool we use to understand how policies affect behaviors of a complex system.

Jay Forrester states in *Designing the Future*, "The model generates streams of decisions controlled by policies built into the model. The policies make all the decisions step-by-step in time as the simulation unfolds. Then, if the resulting behavior is undesirable, one searches for a better set of policies that yield improved resulting."

## **2.5 Operations Management Techniques as Policies**

There are two major operations management techniques we have used: queuing theory and inventory management theories, which include Just-In-Time and Economic Order Quantity.

Queuing theory is the mathematical study developed to provide models to predict the behavior of systems that attempt to provide service for randomly arising demands. In Littlefield Technologies, queuing theory is used to assess the waiting time of arriving jobs, frequency with which the queue length exceeds a given limit, the proportion of the machine's idle time and the duration of a period during which a machine is continuously occupied. The limitation of using classical queuing theory is that it is based on data provided in this game. So it may be too restrictive to be able to model real-world situations exactly. The complexity of production lines with product-specific characteristics cannot be handled with those models.

Inventory management includes inventory control and lead-time management. In Littlefield Technologies, it is used to minimize costs and therefore maximize profit associated with different contract types. In this project, our team also adopts Just-In-Time, an inventory strategy that strives to improve the return on investment by reducing in-process inventory and associated carrying costs.

To control certain inventory level, Economic Order Quantity is used as the optimal order quantity that minimizes total inventory holding costs and ordering costs. We calculate the economic order quantity with the assumption that the ordering cost is constant, the rate of demand is constant, lead time is fixed for each contract type and only ordering and holding cost are associated with ordering. Because there's no holding cost in this Littlefield Technologies simulation game, the EOQ number we get can be as large as infinity. So, it is the best choice to order only once and order as many as possible to fulfill the need to save the ordering cost.

We also use other operations management techniques such as process analysis, bottleneck analysis, lean manufacturing, sensitivity analysis, but only as supplements in evaluating the process, policies and strategies.

### **3.0 Problem Examination**

This section examines the problem in detail. First, an overview of our game-play experience and the problem is presented. Then, the data acquired from the game-run is described and explained. Lastly, a dynamic hypothesis of the problem is proposed based on the patterns observed from the data.

#### **3.1 Game-play Experience and Problem Description**

Littlefield Simulation is an open-ended interactive game. The game reveals limited data to players, cash level and inventory levels are two indicators provided by system. The game challenges the players to react to changes in the system spontaneously and process new data as they make decisions, instead of executing a rigid decision model throughout the simulation. In our game-play, winning strategies were often achieved through trial and error.

The variation of indicators imposes great complexity, which makes it difficult for players to quickly respond to the changes in the system. The reaction of players is more reactive than proactive. For instance, when an increase of lead time is observed, we would directly purchase more machines, instead of tracing the cause of the fluctuation.

The total time span of the game-play was almost five whole days. The time requirement makes it very difficult to repeat the game.

### 3.2 Data Review

The data is extracted from the two rounds of Littlefield Technology Simulation we participated in fall 2010. Three project members played the game in two teams; therefore we have two sets of data retrieved from the simulation.

We chose to use the data from Team 2's game-play in our analysis and model calibration, because the data gives better insight into the difference between contract 1's and contract 2's impact on cash, since Team 2 changed contract type to contract 3 earlier in the game. Also, Team 2 purchased sufficient amount of machines, while the utilization rates of Team 1 peak at 100% due to lack of capacity. Therefore, Team 2's data gives better insight into the ideal number of machines.

#### 3.2.1 Station Queues

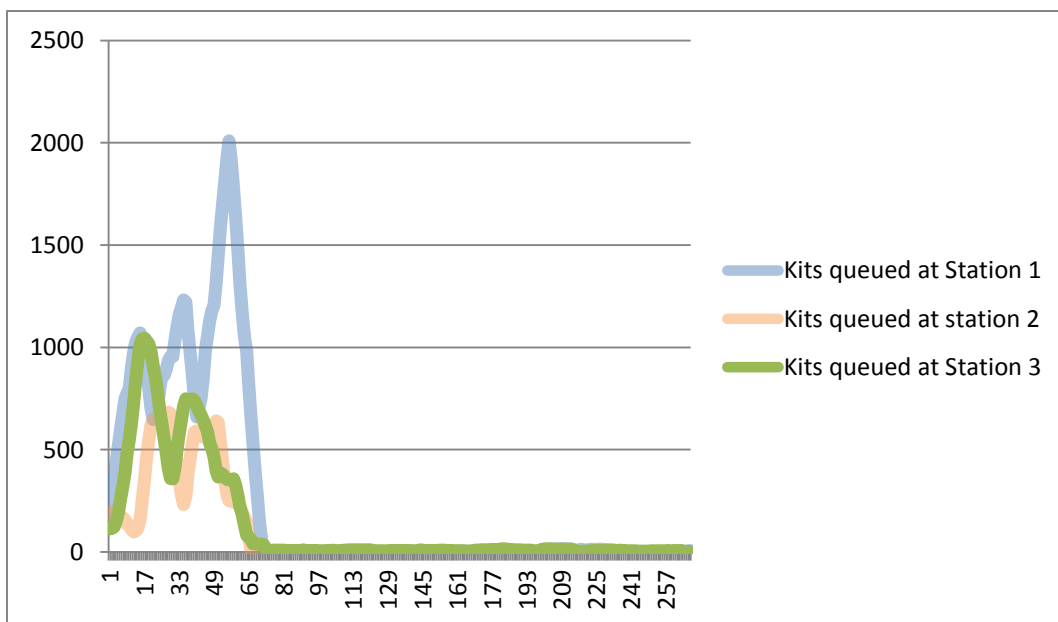


Figure 1-Daily Average Number of Kits for Stations

The graph in Figure 1 shows the kits queued for stations. The queues for three stations go up steadily as long as we have enough kits. The queues decrease if we run out of inventory.

However the queues slope down dramatically after Day 67 since we purchased more machines at that point in time. We use this data to study the ideal number of machines.

### 3.2.2 Station Utilization

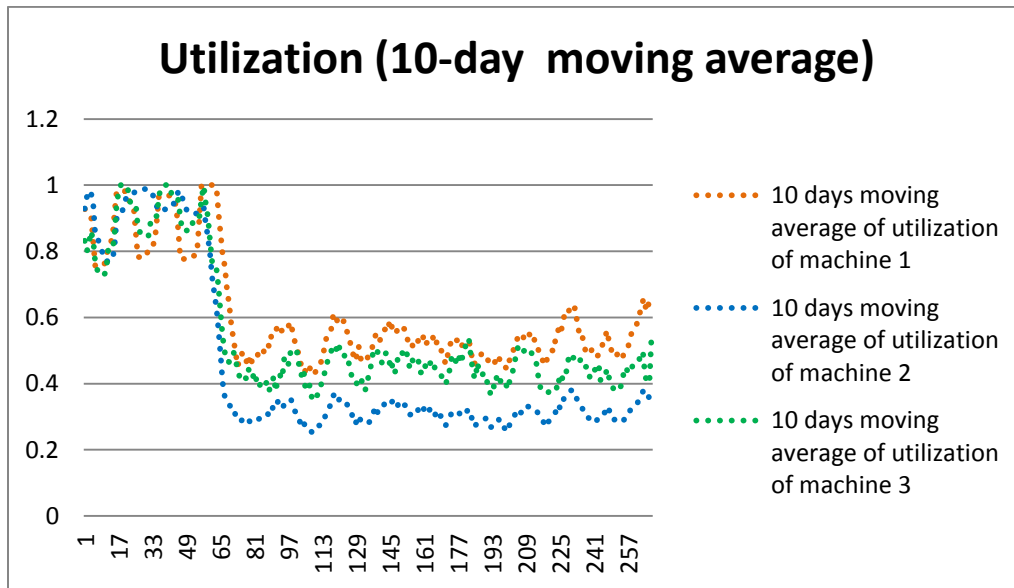


Figure 2-Utilization for Station1

As shown in Figure 2, utilization of stations fluctuates near 100% before day 69. The main reason is the shortage of machine capacity. The fluctuations in utilization are due to lack of inventory (before day 69) and the randomness of order (after day 69). After day 69 we purchased more machines and the utilization fluctuated at a much more reasonable level than before. We use 10 day-moving average so we can have a better picture of how the data looks from a general perspective. This data is used to study the ideal number of machines, discussed later in the strategy section.

### 3.2.3 Inventory

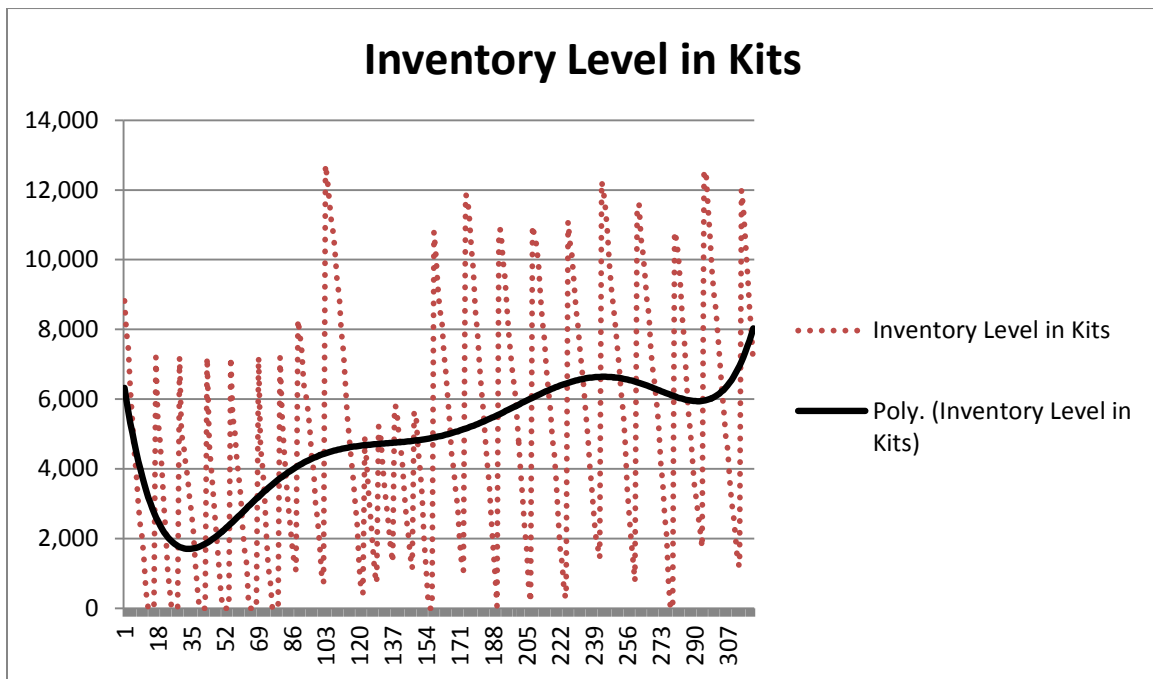


Figure 3 Inventory Level

The red dotted line in Figure 3 describes the process of inventory depletion and re-ordering.

Due to inadequate setting of inventory re-order level; there is a consistent shortage of inventory before day 69. The re-order level of inventory was reset at day 69, so the shortage of kits was alleviated. The re-order quantity was raised at day 150, inventory level increased. Generally speaking the graph is running smoothly after day 69 re-order level set. This dataset is used to study the re-order point and re-order quantity.

### 3.2.4 Completed Jobs

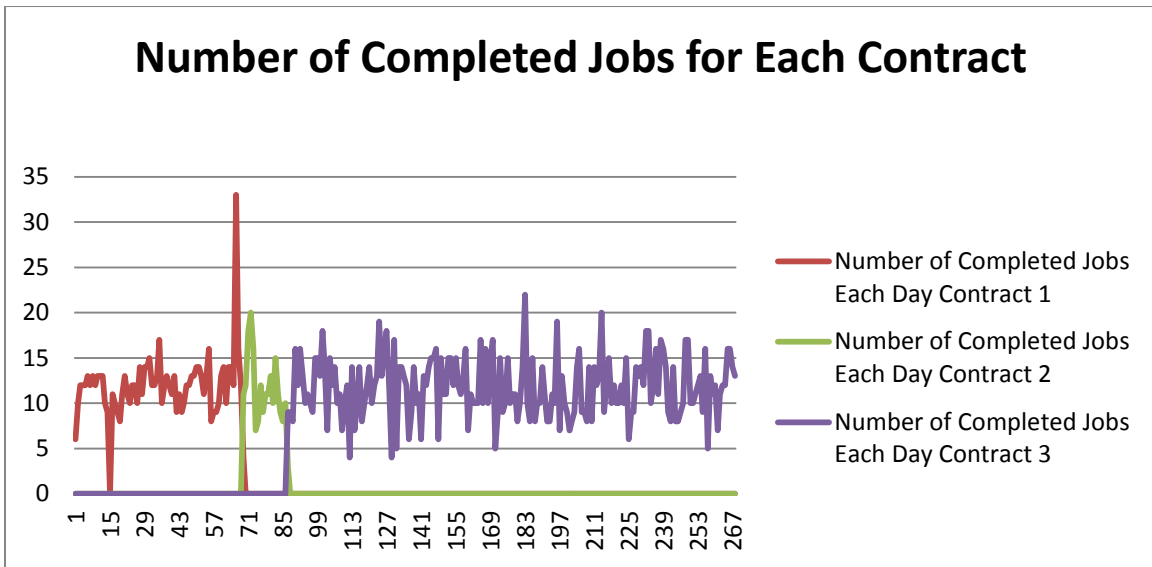


Figure 4 Number of Jobs Completed For Each Contract

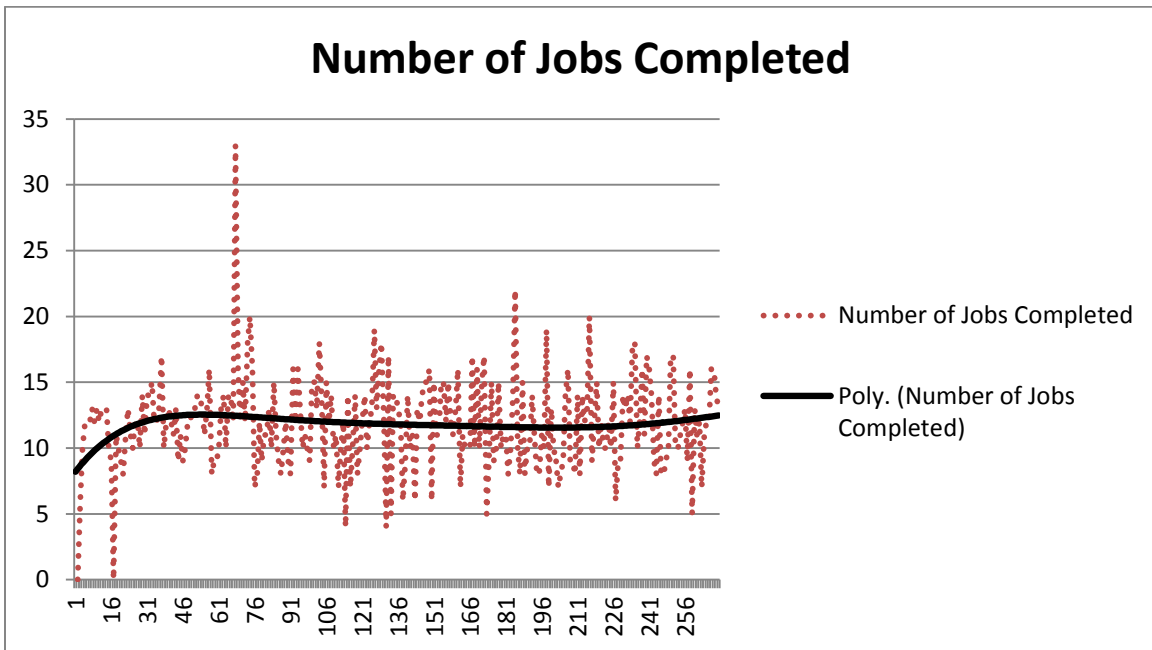


Figure 5 Number of jobs completed

Figure 4 and Figure 5 show the number of jobs completed. Due to inventory depletion, number of completed job at day 17 reached zero. Number of completed job reached peak at day 68 as a result of purchasing new machines, which increased the productivity. The normal



fluctuation of completed job reflects the random number of job arrivals. We use this data to study the number of jobs arrived each day.

### 3.2.5 Lead Time

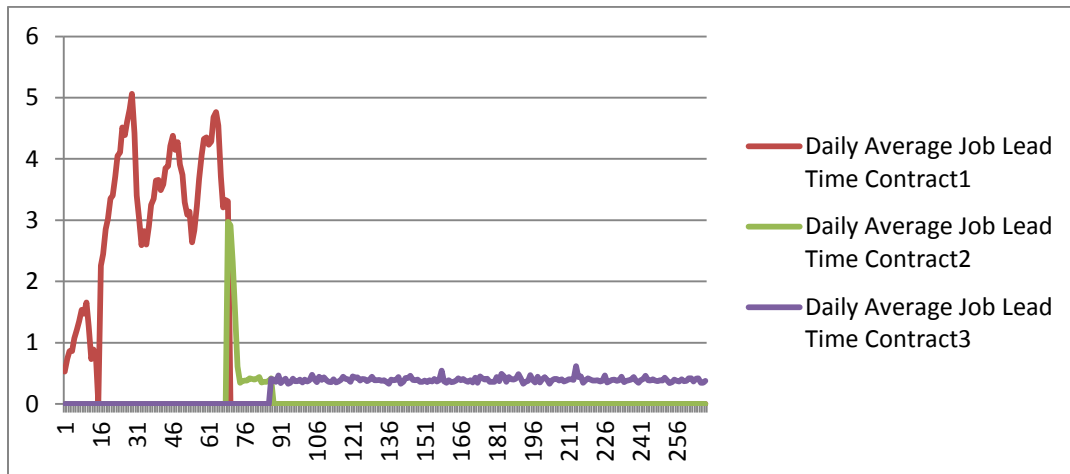


Figure 6 Daily Average Job Lead Time by Contract Type

Lead time increases when any of the machines reaches 100% utilization. We can see the lead time has an increasing trend at first due to the lack of machines. The work flow becomes much smoother after day 69 because of the purchases of new machines. The decrease in the lead time corresponds to the decreases in machine utilization. We use this data to analyze the required lead time for each contract and to confirm our estimates of ideal number of machines.

### 3.2.6 Revenue

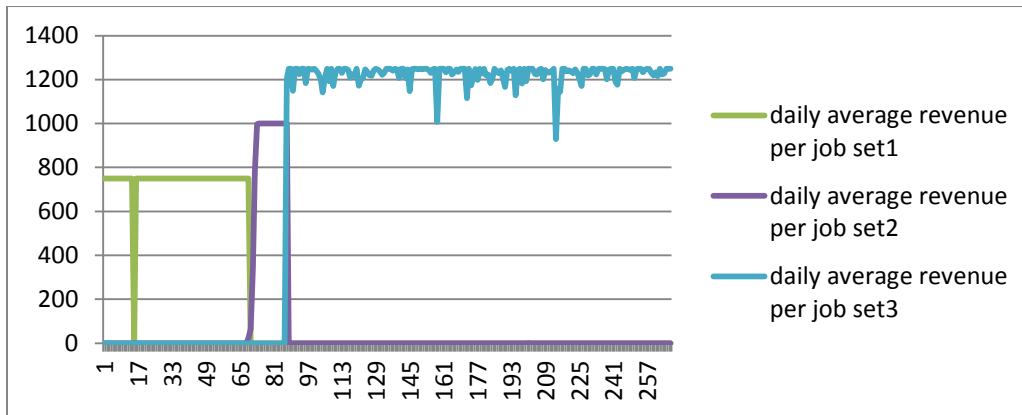


Figure 7 Daily Average Revenue by Contract Type

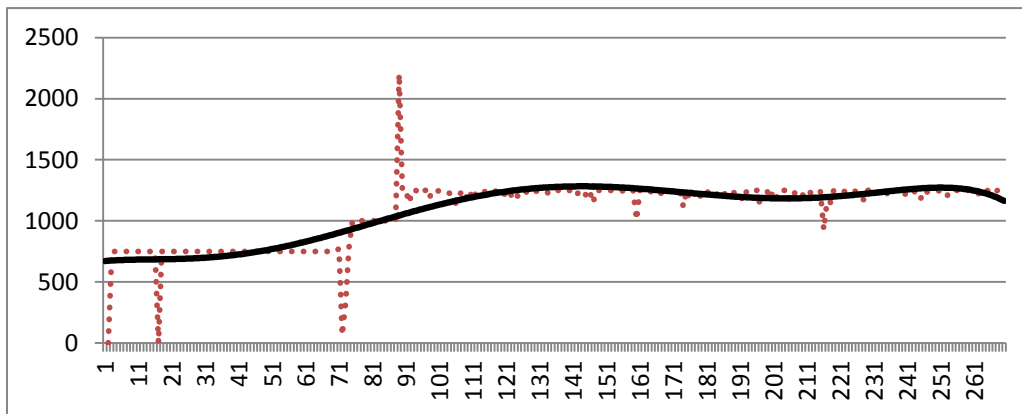


Figure 8 Daily Average Revenue

Figure 7 and Figure 8 show the daily average revenue. The dash line in Figure 8 is the actual number of daily average revenue generated. It is sum of daily average revenue per job set1, daily average revenue per job set2 and daily average revenue per job set3. The solid line in Figure 8 is the 10 day-moving average of the daily average revenue. The revenue reached 0 on day 17 and day 72. It peaked at day 89 because of the change in contract. The revenue chart is steadier than most other graphs we have presented because revenue generated per job is fixed, which implies that when we have enough kits in the inventory and enough productivity at each station, the daily revenue would not change abnormally. Revenue

generated by each job completion is a built-in parameter and cannot be changed. We use this data to study the revenue generated from each contract type.

### 3.3 Dynamic Hypothesis

Figure 9 shows a Causal Loop Diagram (CLD) that will be used as the basis for modeling the Littlefield simulation. A CLD helps in visualizing how interrelated variables affect one another. The CLD consists of a set of nodes representing the variables connected together. The relationships between these variables, denoted by arrows, can be labeled as either positive or negative.

The Littlefield Simulation CLD consists of seven feedback loops, including six negative feedback loops, or balancing loops (labeled as “B”) and one positive feedback loop, or reinforcing loops (labeled as “R”). In a reinforcing loop, each action produces a result which promotes more of the same action. In a balancing loop, each action attempts to bring two things to agreement.

The loops are described in detail after Figure 9. The CLD contains the aggregation of two series of stocks: the *Machines* stocks and *Raw Materials* stocks.

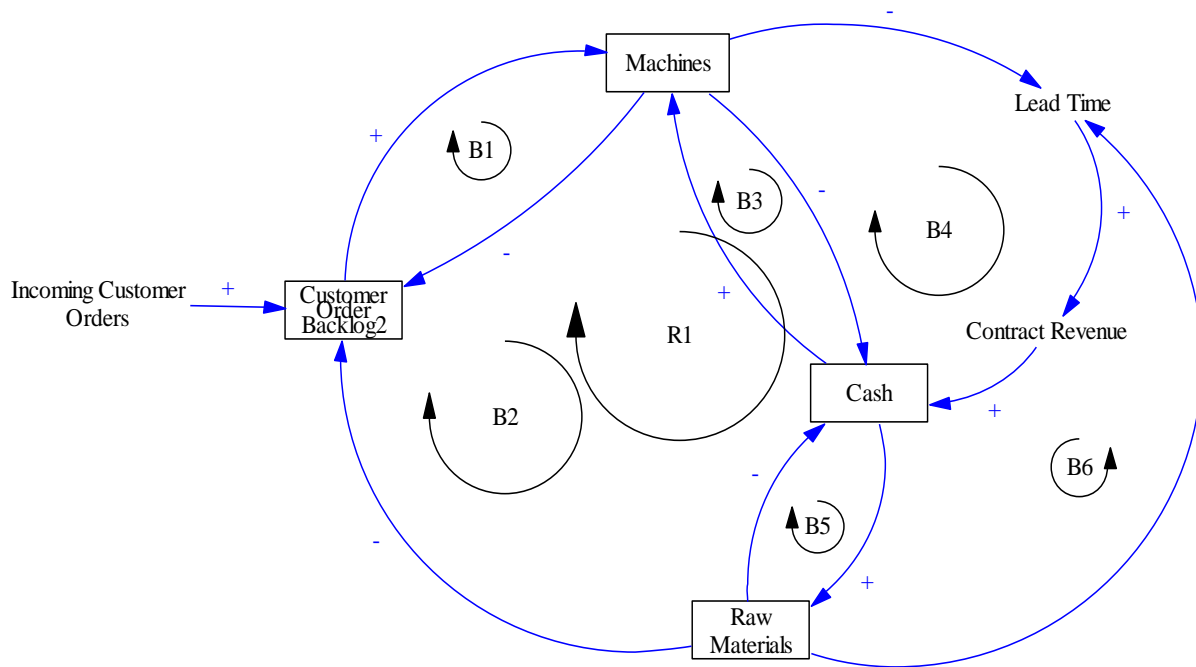


Figure 9 Dynamic Hypothesis Model

### Explanation of the Feedback Loops

**Loop Number B1:** *Customer Order Backlog* → *Machines* → *Customer Order Backlog*

Increase in *Machines* accelerates the depletion of *Customer Order Backlog*. Increase in *Customer Order Backlog* motivates the decision maker to purchase more *Machines*.

**Loop Number B2:** *Cash* → *Raw Materials* → *Customer Order Backlog2* → *Machines* → *Cash*

Increase in *Cash* enables the decision maker to purchase more *Raw Materials*. Sufficient *Raw Materials* enables the *Customer Order Backlog* to deplete at a predictable rate. Increase in *Customer Order Backlog* motivates the decision maker to purchase more *Machines*. Payment for *Machines* depletes *Cash*.

**Loop Number B3:** *Cash →Machines →Cash*

Increase in *Cash* enables the decision maker to purchase more *Machines*. Payment for new *Machines* depletes *Cash*.

**Loop Number B4:** *Cash →Machines →Lead Time →Contract Revenue →Cash*

Increase in *Cash* enables the decision maker to purchase more *Machines*. Increase in *Machines* leads to decrease in *Lead Time*. Less *Lead Time* induces higher *Contract Revenue*. Higher *Contract Revenue* means higher rate of increase for *Cash*.

**Loop Number B5:** *Cash →Raw Materials →Cash*

Increase in *Cash* enables the decision maker to purchase more *Raw Materials*. Payment for *Raw Materials* depletes *Cash*.

**Loop Number B6:** *Cash →Raw Materials →Lead Time →Contract Revenue →Cash*

Increase in *Cash* enables the decision maker to purchase more *Raw Materials*. Sufficient *Raw Materials* ensures a certain *Lead Time*. Less *Lead Time* induces higher *Contract Revenue*. Higher *Contract Revenue* means higher rate of increase for *Cash*.

**Loop Number R1:** *Cash →Raw Materials →Customer Order Backlog2 →Machines →Lead Time →Contract Revenue →Cash*

Increase in *Cash* enables the decision maker to purchase more *Raw Materials*. Sufficient *Raw Materials* ensures a certain *Lead Time*. Less *Lead Time* induces higher *Contract Revenue*. Higher *Contract Revenue* means higher rate of increase for *Cash*.

## 4.0 System Dynamics Model Description and Testing

This section describes the Littlefield model as it was built using a System Dynamics approach. The section starts with an overview of the sector diagram. Thereafter, each sector is described in detail, with justification for the approach chosen to model that sector. The sector description ends with explanation of all the equations used in the sector.

### 4.1 Overview of the Sectors

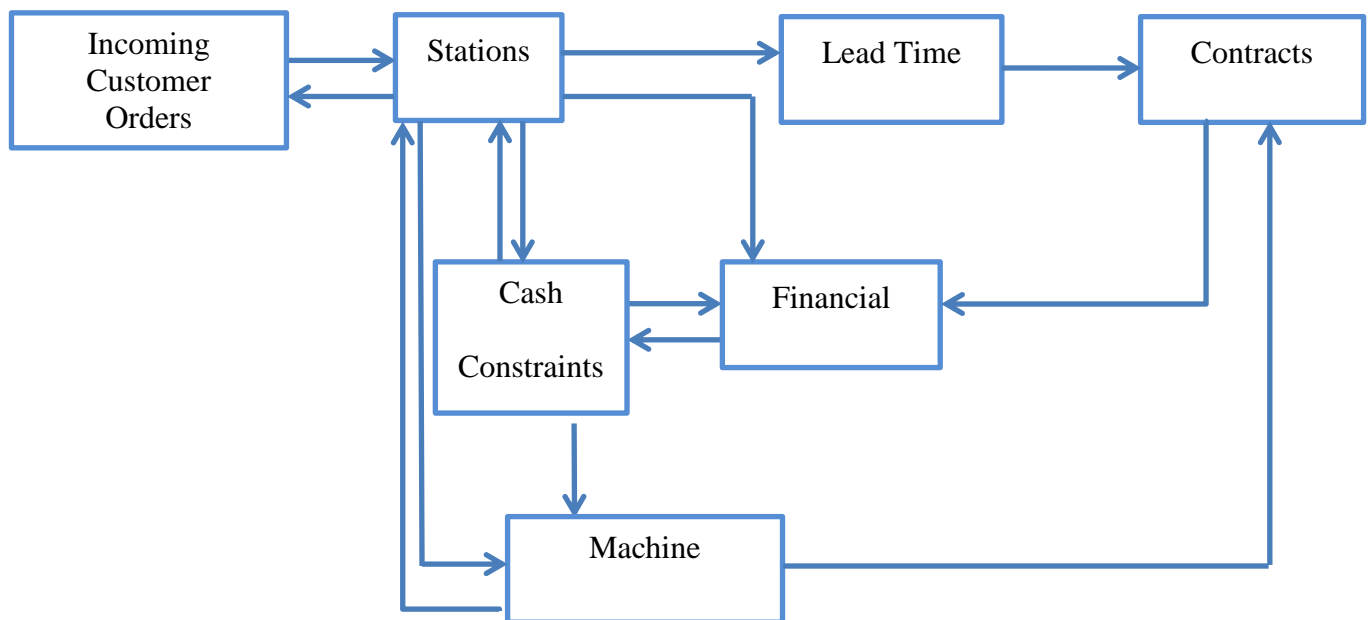


Figure 10 Overall Sector Diagram

Figure 10 is an overview of the model for the Littlefield Simulation. The diagram shows the top level relationships between the different sectors in the model. The input to the game is in the incoming customer orders sector. The job orders are released to the station sector. The station sector keeps track of the all the inventory from the raw material to finished goods. When the order is finished, the number of orders in the system is decreased and the appropriate lead time is calculated in the lead time sector. Based upon the perceived lead time the contracts sector chooses the contract which will give the highest possible revenue. Thereafter, the revenue from

the sale of the good is calculated in the financial sector based upon the lead time and the contract chosen.

The cash constraints sector places a restriction on the number of machines and raw materials that can be bought based upon the amount of cash available. Whenever raw materials and inventory are purchased there is a decrease in the amount of cash.

The game player has the ability to control 3 aspects of the game in order to maximize profit. First, the game player can change the reorder level and reorder quantity of the raw materials. Second, the game player can increase or decrease the desired lead time of the different machines by increasing or decreasing the number of machines. Last, the game player can change the priority between stage 2 and stage 4 of the production process.

## 4.2 Sector Descriptions

Each sector shown in Figure 10 is described in this section. The incoming customer orders are the input to the system. Therefore, this is the section described first.

### 4.2.1 Input to the system

Figure 11 shows the *Input of the System sector* which consists of two stocks. *Customer Order Queue* represents the *Customer Order* backlog, i.e. the *Customer Orders* waiting to be processed throughout the system. A *Customer Order* is processed when a *Job* is released to station 1 queue. *Customer Order in the System* represents the total amount of *Jobs* in the system. When a *Job* is released, the backlog increases. One constraint setting in the Littlefield Simulation is that the total amount of customer orders being processed in the system, i.e. *Customer Order in the System*, cannot exceed 100.

The *Customer Order Queue* and *Raw Material Inventory* both affect *Job Release*. A low *Raw Material Inventory* to *Customer Orders Queue* ratio would slow down *Job Release*. The impact of the *ratio* on *Job Release* is defined with a graphical function, which slopes linearly from 0 to 1 for ratios less than 1, and stays at 1 afterwards. The graphical function is then multiplied to *Job Release* in the *Stations* sector.

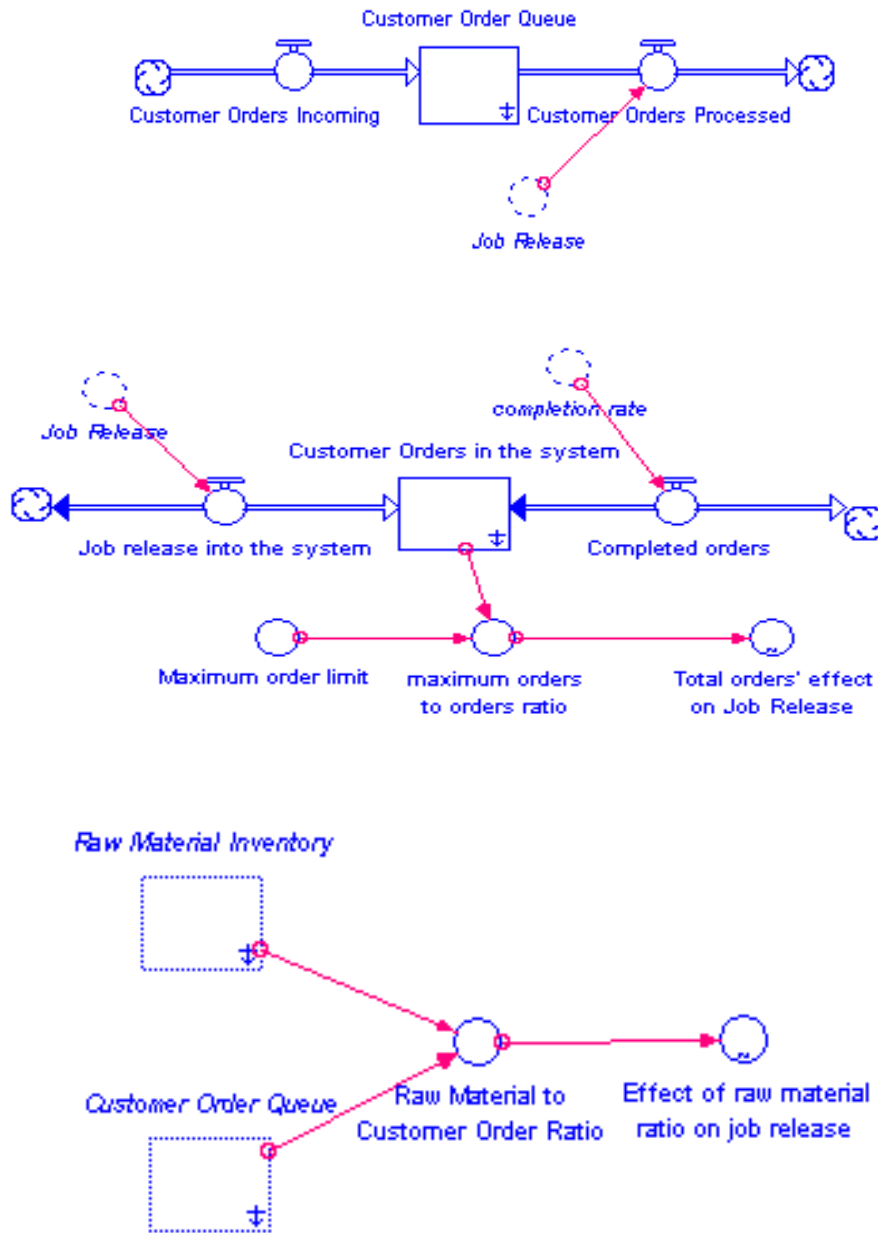


Figure 11 Input of the System



## Equations

### Stock 1: Customer Order Queue

This stock accumulates all the customer orders waiting to be released to the factory.

$$\mathbf{Customer\_Order\_Queue}(t) = \mathbf{Customer\_Order\_Queue}(t - dt) + (\mathbf{Customer\_Orders\_Incoming} - \mathbf{Customer\_Orders\_Processed}) * dt$$

INITIAL VALUE of  $\mathbf{Customer\_Order\_Queue} = 0$

**Inflows:  $\mathbf{Customer\_Orders\_Incoming} = 12$**

**Outflows:  $\mathbf{Customer\_Orders\_Processed} = \mathbf{Job\_Release}$**

### Stock 2: Customer Orders in the system

This stock measures the total orders being processed in the factory.

$$\mathbf{Customer\_Orders\_in\_the\_system}(t) = \mathbf{Customer\_Orders\_in\_the\_system}(t - dt) + (\mathbf{Job\_release\_into\_the\_system} - \mathbf{Completed\_orders}) * dt$$

INITIAL VALUE of  $\mathbf{Customer\_Orders\_in\_the\_system} = 0$

**Inflows:  $\mathbf{Job\_release\_into\_the\_system} = \mathbf{Job\_Release}$**

**Outflows:  $\mathbf{Completed\_orders} = \mathbf{completion\_rate}$**

### Variable: Raw Material to Customer Order Ratio

This variable measures whether there are enough raw materials to process customer orders

**Raw\_Material\_to\_Customer\_Order\_Ratio =**

Raw\_Material\_Inventory/Customer\_Order\_Queue)

Graphical Function: Effect of Raw Material on Job Release

Figure 12 shows the graphical function which shows the effect the ratio of raw materials to customer orders has on job release. If there is not enough raw materials the customer orders are released based upon the amount of raw materials available. If the ratio is greater than 1 all the customer orders are released into the factory.

**Effect\_of\_raw\_material\_ratio\_on\_job\_release =**

GRAPH(Raw\_Material\_to\_Customer\_Order\_Ratio)

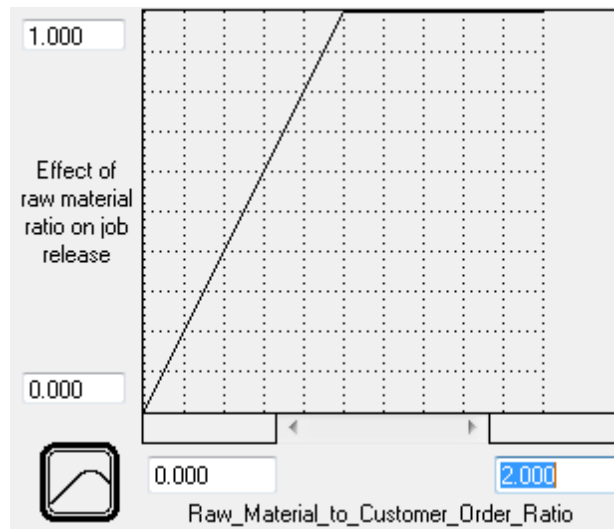


Figure 12 Effect of Raw Material Ratio on Job Release

#### 4.2.2 Cash Constraints

Figure 13 depicts the *Cash Constraints* sector. It shows the different expenses that will be encountered in the system, which includes *Total Cost of Order* and expenses to purchase *Machinery*. The constraints affect the *Raw Materials* and *Machines* separately. If the *Cash Constraints* limit purchase of machines, the system still allows the maximum number of kits to

be bought based upon the present *Cash*. The consideration is that when raw material is insufficient, the whole production process stops, and backlog will keep increasing, which would greatly affect lead time. The purchase of *Machinery* is only allowed when the *Cash Constraint* is not invoked by the algorithm, which has an implicit preference for *Raw Materials*. This decision is rather discrete. All *Cash Constraints* are implemented by calculating the *Ratio of Purchase vs. Cash*, and *Ratio of Machine Cost vs. Cash*. Then, the calculated values are used in graphical functions. The slopes of the graphical functions describe the *Effect of Cash on Purchasing Raw Materials* and *Effect of Cash Ratio on Machine*, these two have different values.

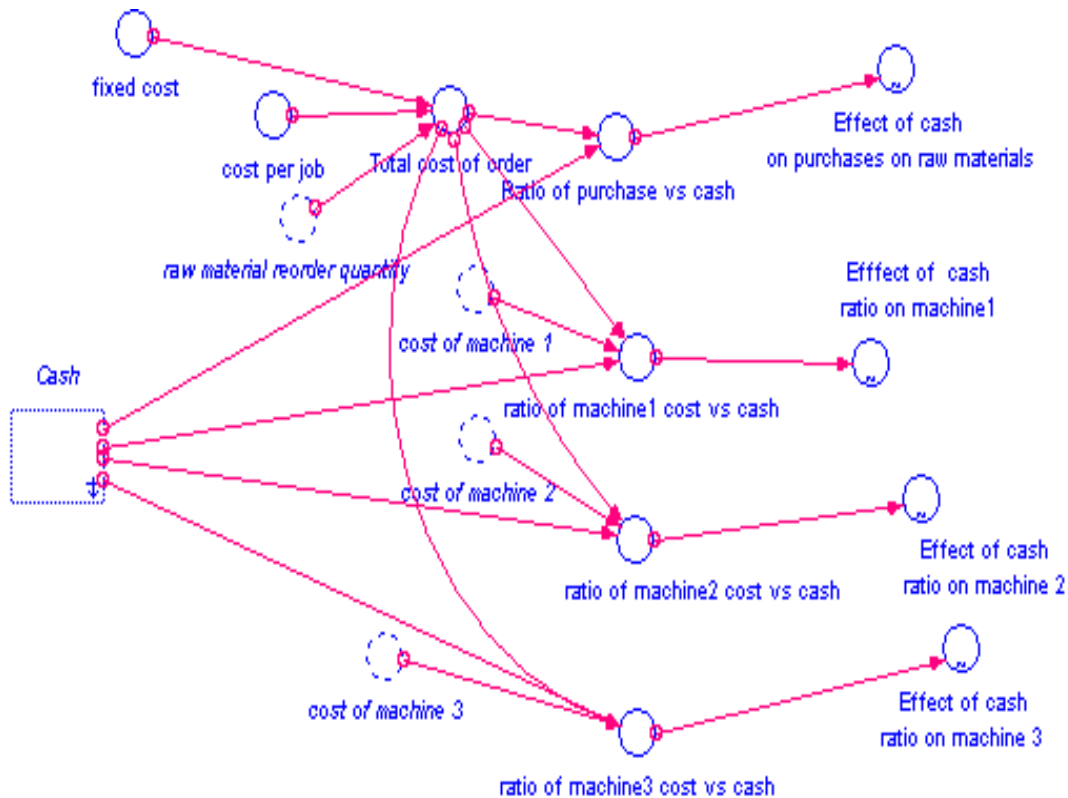


Figure 13 Cash Constraints

### Equations

The two main equations in this sector calculate the ratio of the cost of machines and the cost of raw materials with the amount of cash available. The machine purchase constraint also has the

cost of raw materials included in it because a machine can be bought only if there is enough cash for raw materials. This places a priority for raw materials because if there are no raw materials production will stop.

a) Raw Material Purchase Constraints

Variables

$$\mathbf{Ratio\_of\_purchase\_vs\_cash} = \text{Cash}/\text{Total\_cost\_of\_order}$$

$$\mathbf{Total\_cost\_of\_order} = \text{fixed\_cost} + \text{cost\_per\_job} * \text{raw\_material\_reorder\_quantity}$$

Constants

$$\mathbf{fixed\_cost} = 1000$$

$$\mathbf{cost\_per\_job} = 600$$

b) Machine Purchase Constraints

Variable

$$\mathbf{ratio\_of\_machine1\_cost\_vs\_cash} = \text{Cash} / (\text{cost\_of\_machine\_1} + \text{Total\_cost\_of\_order})$$

### Graphical Functions

a) Raw Material Purchase Constraints:

$$\mathbf{Effect\_of\_cash\_on\_purchases\_on\_raw\_materials} =$$

$$\text{GRAPH}(\text{Ratio\_of\_purchase\_vs\_cash})$$

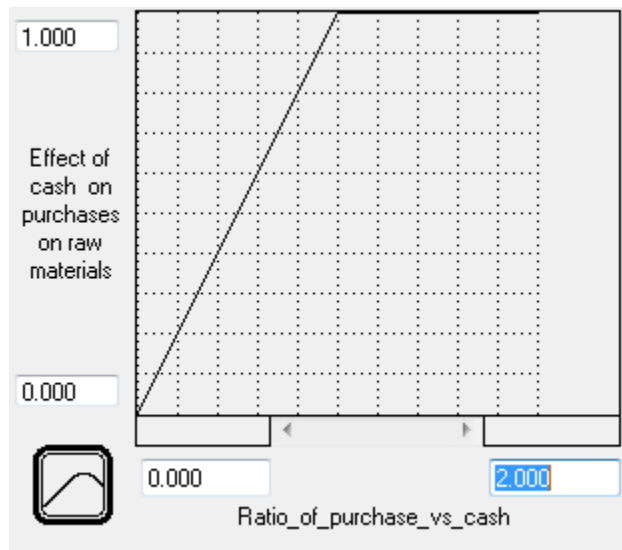


Figure 14 Cash Constraint on Raw Materials

Figure 14 shows the linear function that models how the purchase of raw materials is affected as the ratio of the cost of raw materials to cash changes. The function is a straight line from 0 to 1 for a ratio of less than 1 and 1 for higher ratios. This is because raw materials are a high priority investment because without it production will stop. Therefore if there is not enough cash to buy all the raw materials required, the cash will be used to buy as many as possible.

b) Effect of Cash Ratio on Machine

$$\text{Effect\_of\_cash\_ratio\_on\_machine1} = \text{GRAPH}(\text{ratio\_of\_machine1\_cost\_vs\_cash})$$

Figure 15 shows the graphical function that models the purchase of machine based upon the ratio of the cost of buying the machines to cash available. Since machines are lower priority than raw materials the machine will be bought only if there is enough cash. Therefore the graph is 0 for ratios less than 1 and 1 for ratios equal to or greater than 1.

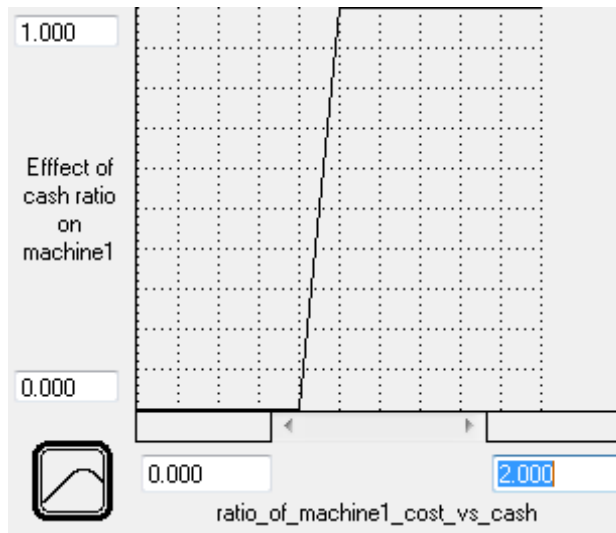


Figure 15 Cash Constraints on Machines

### 4.2.3 Financials

The *Financial Sector*, shown in Figure 16, has one stock called *Cash*, which stands for the total amount of *Cash* at any given time. The inflow to *Cash*, *Revenues per Day*, includes *Total Machine Sales*, *Revenue from Completing Orders*, and *Revenue from Interest*. *Interest* is earned from present *Cash*.

The *Total Machine Sales* is calculated by multiplying *Prices of Machines Sold* with the *Total Machine Sales*.

The *Interest Rate* on present *Cash* is 10%. This *Interest* is compounded per day. In order to calculate the *Interest Rate per Day* the following equation is used:

$$(1+x)^{365}=1.0026.$$

This gives the *Interest Rate per Day* of 0.0026%. This is multiplied to present *Cash* and added to the *Cash Value* every day.

The *Revenue from Completing Orders* is calculated by multiplying the *revenue per batch* with *completion rate*, which is explained in the contract sector.

The Outflows of Cash include expenses from *Ordering Raw Material* and *Total Machine Expenses*. An if-then-else is used in the *Constant Change in Machine* to check whether a machine is being bought or sold.

The *Total machine expenses* are calculated by multiplying the different machine prices with the number of machines bought. The variable, *Expenses from Ordering Raw Materials*, is calculated by multiplying the raw material purchase rate with the fixed cost over the raw material reorder quantity and adding cost per job.

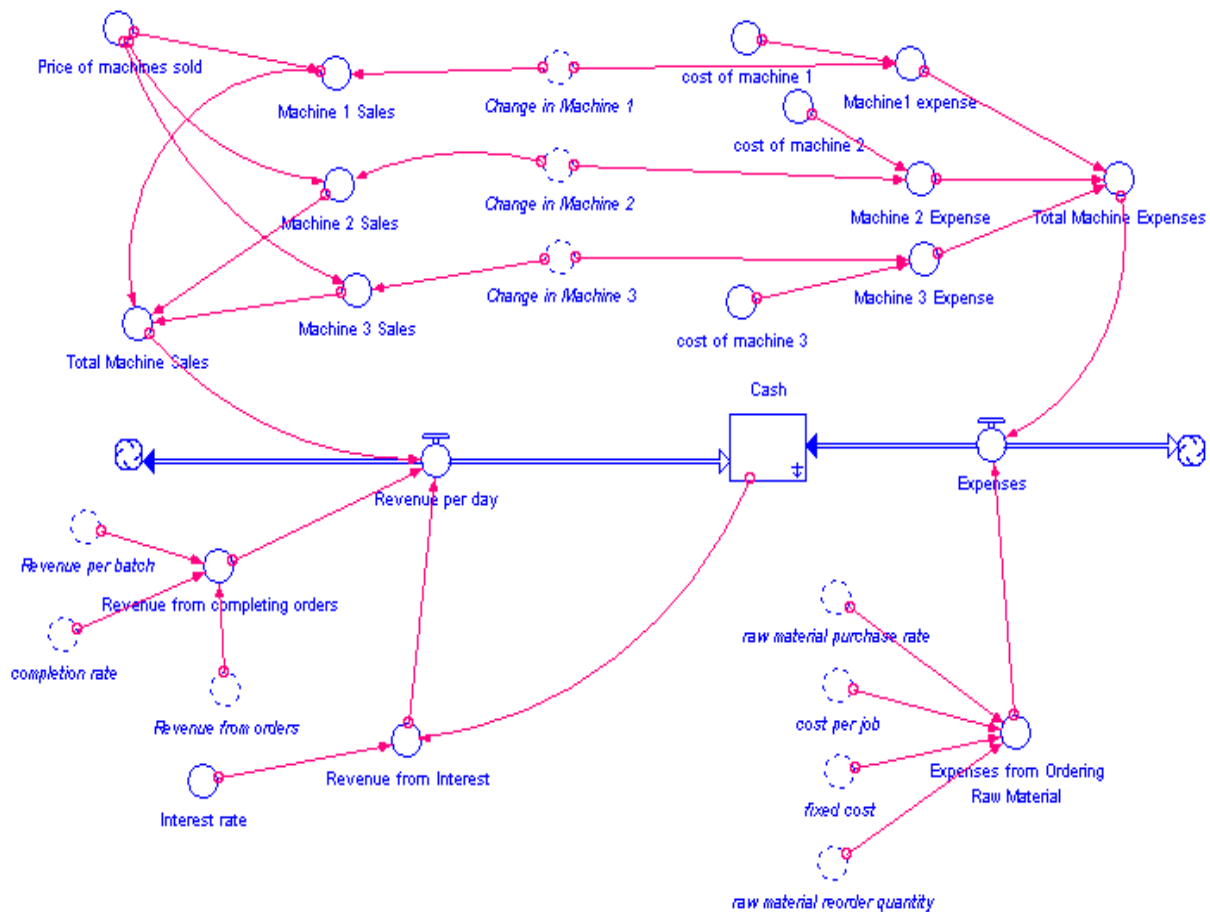


Figure 16 Financial Sector

## Equations

### Stock: Cash

The financial sector has 1 stock which is for the amount of cash. The two flows are revenues and expenses. The main revenues are from completing orders and sale of machines. Cash can also increase from interest earned on it. The expenses are from the purchase of raw materials and machines.

$$\mathbf{Cash(t)} = \text{Cash}(t - dt) + (\text{Revenue\_per\_day} - \text{Expenses}) * dt$$

$$\text{INITIAL VALUE of Cash} = \$1,000,000$$

$$\mathbf{Inflow: Revenue\_per\_day} =$$

$$\text{Revenue\_from\_completing\_orders} + \text{Revenue\_from\_Interest} + \text{Total\_Machine\_Sale}$$

$$\mathbf{Outflow: Expenses} =$$

$$\text{Expenses\_from\_Ordering\_Raw\_Material} + \text{Total\_Machine\_Expenses}$$

### Variables

$$\mathbf{Revenue\_from\_completing\_orders} = \text{completion\_rate} * \text{Revenue\_per\_batch}$$

$$\mathbf{Revenue\_from\_Interest} = \text{Cash} * \text{Interest\_rate}$$

$$\mathbf{Total\_Machine\_Sales} = \text{Machine\_1\_Sales} + \text{Machine\_2\_Sales} + \text{Machine\_3\_Sales}$$

$$\mathbf{Total\_Machine\_Expenses} = \text{Machine1\_expense} + \text{Machine\_2\_Expense} + \\ \text{Machine\_3\_Expense}$$

$$\mathbf{Expenses\_from\_Ordering\_Raw\_Material} = \text{raw\_material\_purchase\_rate} * \\ (\text{fixed\_cost}/\text{raw\_material\_reorder\_quantity} + \text{cost\_per\_job})$$



## Constants

**Interest\_rate = .0026**

**Price\_of\_machines\_sold = 10000**

**cost\_of\_machine\_1 = 90000**

**cost\_of\_machine\_2 = 80000**

**cost\_of\_machine\_3 = 100000**

## Functions

**Machine\_1\_Sales = IF(Change\_in\_Machine\_1 < 0) THEN**

**SQRT((Price\_of\_machines\_sold \* Change\_in\_Machine\_1)^2) ELSE**

**0**  
**Machine1\_expense = IF(Change\_in\_Machine\_1 > 0) THEN cost\_of\_machine\_1**

**\* Change\_in\_Machine\_1 ELSE 0**

**Machine\_2\_Expense = IF(Change\_in\_Machine\_2 > 0) THEN cost\_of\_machine\_2**

**\* Change\_in\_Machine\_2 ELSE 0**

**Machine\_2\_Sales = IF(Change\_in\_Machine\_2 < 0) THEN**

**SQRT((Price\_of\_machines\_sold \* Change\_in\_Machine\_2)^2) ELSE 0**

**Machine\_3\_Expense = IF(Change\_in\_Machine\_3 > 0) THEN cost\_of\_machine\_3**

**\* Change\_in\_Machine\_3 ELSE 0**

**Machine\_3\_Sales = IF(Change\_in\_Machine\_3 < 0) THEN**

**SQRT((Price\_of\_machines\_sold \* Change\_in\_Machine\_3)^2) ELSE 0**

These discrete functions above change from 0 to 1 based upon whether a machine is being sold or bought.

#### 4.2.4 Contracts

Figure 17 shows the contracts sector. This sector is used to select the contract type based upon the present *Average Lead Time*. *Current Average Lead Time* affects *Perceived Lead Time*, which affects the contract type chosen. The process utilizes if-then-else functions to determine a contract type based on the range of *Perceived Lead Time*. *Perceived Lead Time* is a smooth function using *Average Lead Time* and is basically used to signal the delay it takes for people to perceive the actual *Average Lead Time*. The contracts types have either a value of 1 or 0, depending on if it is chose or not.

The *Average Lead Time* also affects the *Revenue per Batch* as the lead time effects the revenue on the order based upon the contract chosen.

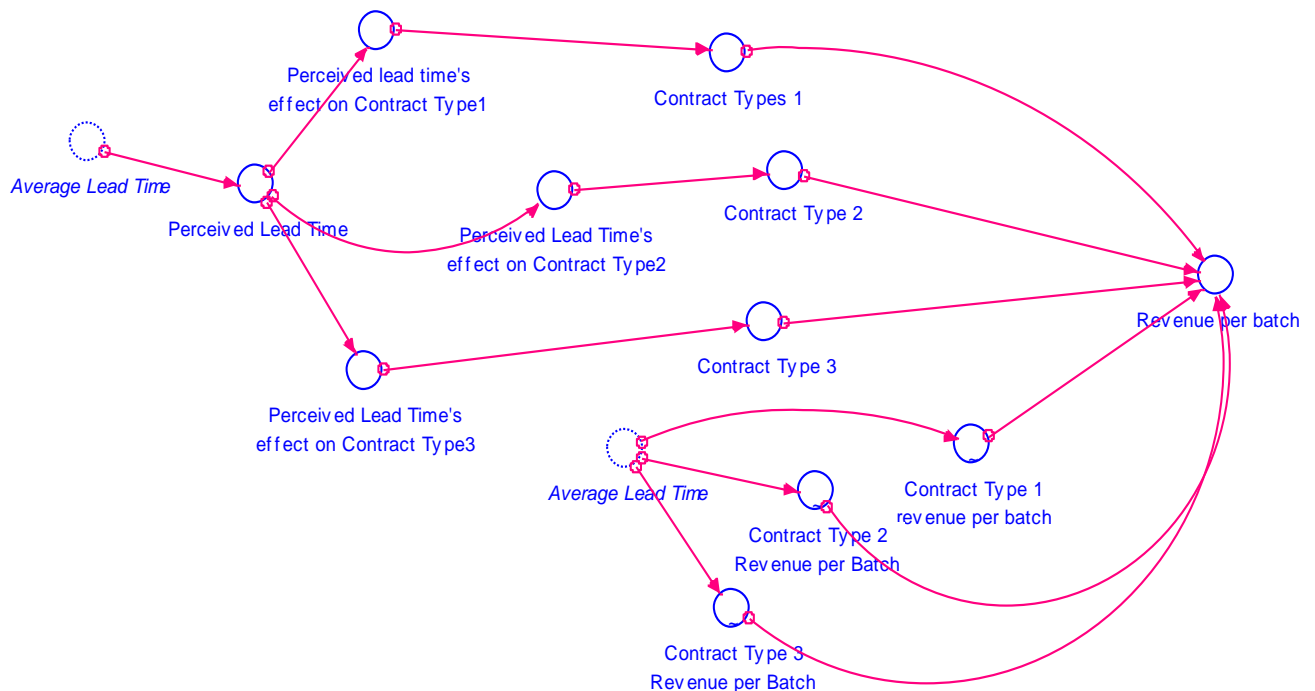


Figure 17 Contracts Sector

## Equations

This sector models the human reactions on changing contracts based upon lead time.

**Revenue\_per\_batch =**

$$\text{Contract\_Types\_1} * \text{Contract\_Type\_1\_revenue\_per\_batch} + \text{Contract\_Type\_2} * \text{Contract\_Type\_2\_Revenue\_per\_Batch} + \text{Contract\_Type\_3} * \text{Contract\_Type\_3\_Revenue\_per\_Batch}$$

The revenue per batch calculates the revenue on each order based upon the contract quoted and its lead time and the actual lead time the contract was completed in.

**Perceived\_Lead\_Time = SMTH3(average\_lead\_time,3.5,1.5)**

The perceived lead time variable is modeled as a delay of average lead time using the Smooth function.

### Perceived Lead Time's Effect on Contract Type

The discrete functions of the perceived lead time's effect on the contract decide which contract to choose based on the perceived lead time.

**Perceived\_lead\_time's\_effect\_on\_Contract\_Type1 = IF (Perceived\_Lead\_Time > 1.2)  
THEN 1 ELSE 0**

**Perceived\_Lead\_Time's\_effect\_on\_Contract\_Type2 = IF(Perceived\_Lead\_Time > .6)  
AND (Perceived\_Lead\_Time <= 1.2) THEN 1 ELSE 0**

**Perceived\_Lead\_Time's\_effect\_on\_Contract\_Type3 = IF(Perceived\_Lead\_Time<=.6)  
THEN 1 ELSE 0**

Revenue per Batch

The three graphical functions (Figure 18, Figure 19 and Figure 20) help to calculate the revenue earned for each order based upon the actual lead time and the contract selected.

**Contract\_Type\_1\_revenue\_per\_batch = GRAPH(average\_lead\_time)**

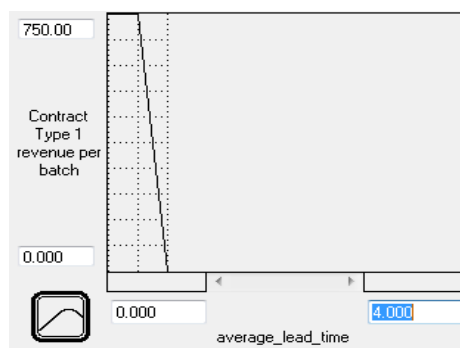


Figure 18 -Contract Type 1 and Lead Time

**Contract\_Type\_2\_Revenue\_per\_Batch = GRAPH(average\_lead\_time)**

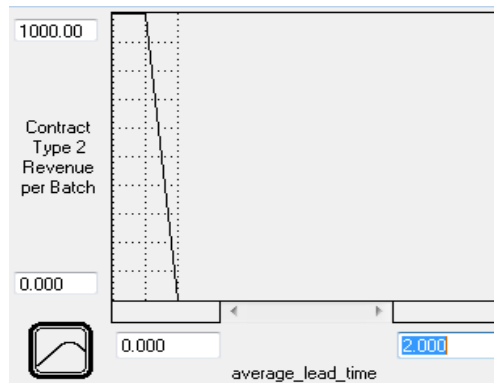


Figure 19- Contract Type 2 and Lead Time

**Contract\_Type\_3\_Revenue\_per\_Batch = GRAPH(average\_lead\_time)**

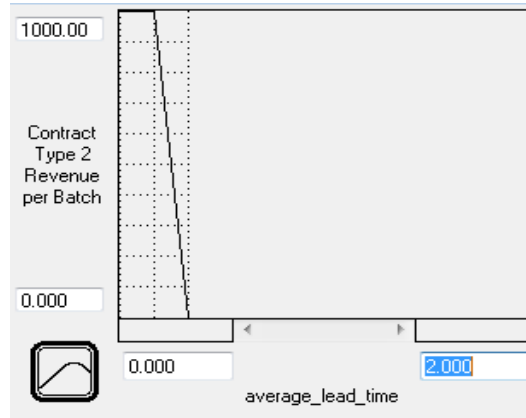


Figure 20 -Contract Type 3 and Lead Time

#### 4.2.5 Average Lead time

Figure 21 shows the *Average Lead Time* sector. This sector shows the calculation of *Average Lead Time*. It adds the actual delays in stations 1 to 4, or *Actual Time 1 to 4* and the time spent in customer backlog, *Job Release Lead Time*. The sum of all these is the *Average Lead Time*.

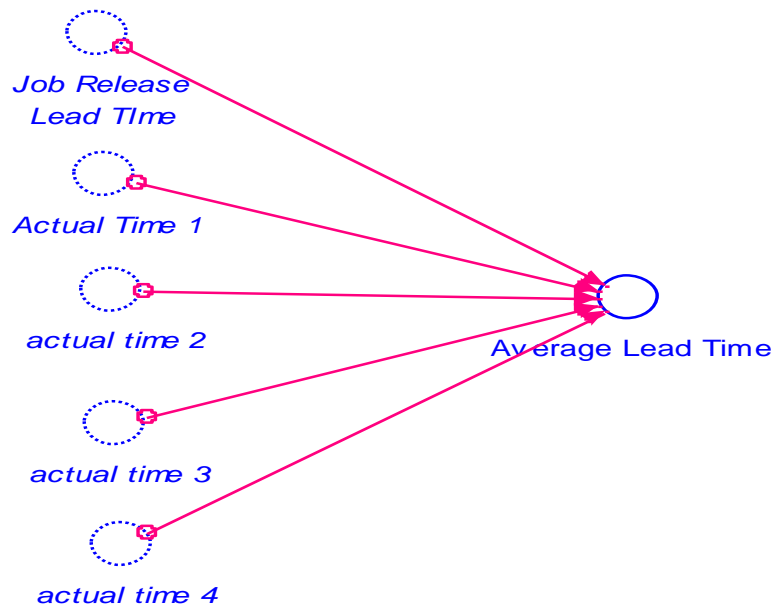


Figure 21 Lead Time Sector

Variable:

This sector calculates the actual average lead time based upon the lead time from all the stations and machines.

$$\text{Average\_Lead\_Time} = \text{Job\_Release\_Lead\_Time} + \text{Actual\_Time\_1} + \text{actual\_time\_2} \\ + \text{actual\_time\_3} + \text{actual\_time\_4}$$

#### 4.2.6 Machines

The machine sector is shown in Figure 22. This sector has 3 stocks; one stock for each machine type. The machine stocks are named *Machine 1*, *Machine 2* and *Machine 3*

The number of *Machines* will change based upon the *Desired Number of Machines* over a constant *Adjustment Time*. The *Adjustment Time* is an arbitrarily chosen constant through game play experience.

Positive *Change in Machine* can only happen if there is enough *Cash*. This is done through the *Cash Constraints* for machines, which is explained in the *Cash Constraints* sector.

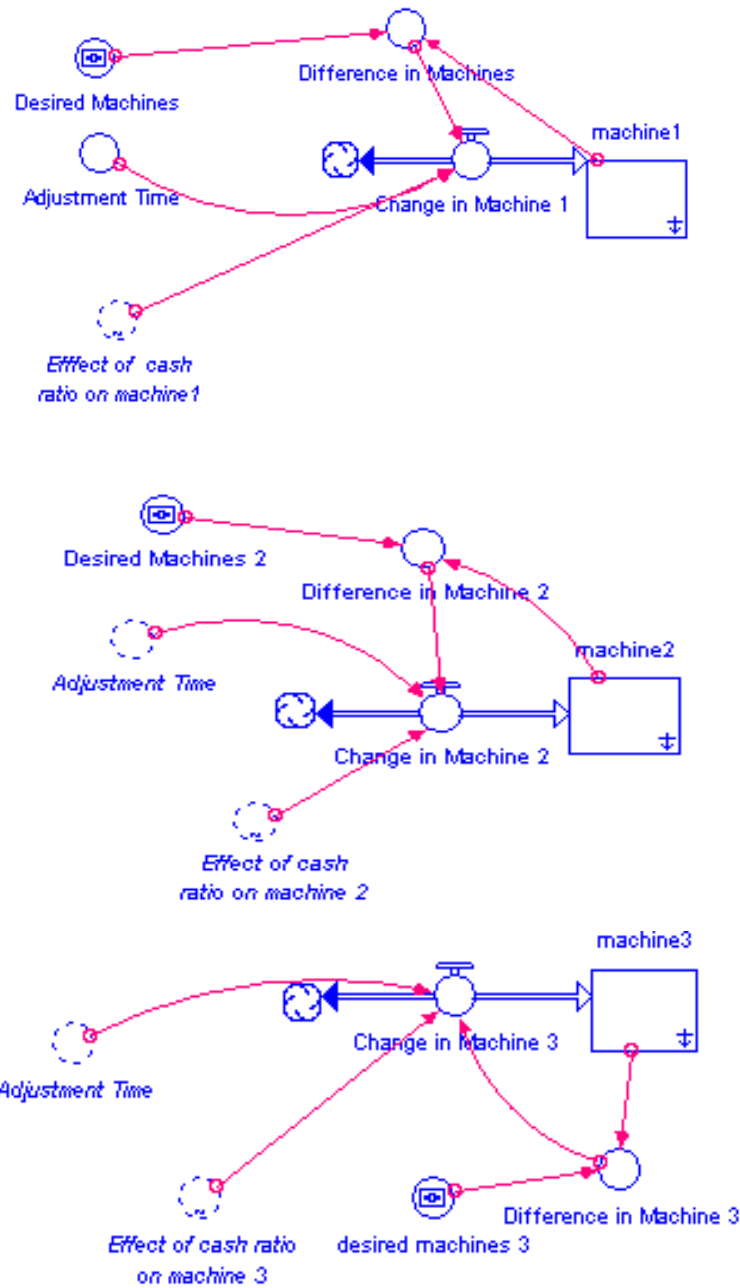


Figure 22 Machines Sector

## Equations

This sector models the machines and purchase and sale of them. The numbers of machines are modeled as stocks with their initial values based upon the parameters played in the actual game. The change in machines is based upon the difference between desired machines and actual number of machines and based upon cash constraints.

### Stock: Machine

$$\mathbf{machine1(t)} = machine1(t - dt) + (Change\_in\_Machine\_1) * dt$$

INITIAL VALUE of machine1 = 3

INITIAL VALUE of machine 2=1

INITIAL VALUE of machine 3=1

### Constant

**Desired Machines= (User Chosen Value)**

**Adjustment\_Time = 1**

### Variable

**Difference in Machines= Desired\_Machines-machine1**

### Function

**Change\_in\_Machine\_1 = Change\_in\_Machine\_1 = IF (TIME <51) OR (TIME >218)**  
THEN 0  
ELSE(Difference\_in\_Machines/Adjustment\_Time)\*Effect\_of\_\_cash\_ratio\_on\_  
machine1

The above function shows the change in machine which represent whether a machine is bought or sold. A positive value means a machine is bought and vice-versa. It is an if-then-else function as the Littlefield game has constraints on when machines can be bought and sold: No machines can be bought before day 50 and after day 218. Furthermore the machines can be



bought only if there is enough cash in the cash sector. This effect is shown by the multiplication of the variable *Effect of cash ratio on machine*

#### 4.2.7 Stations

The stations sector is shown in Figure 23. It consists of 6 stocks. The *Stations* sector is the main sector in the model. The stocks show the pipeline of the production process from the *Order of Raw Material*, *Raw Material Inventory*, and the *Queues* at each of the 4 steps.

The order for raw material is affected by *Raw Material Purchasing Rate* and *raw material arriving rate*. As long as there is enough *Cash* to support the *Customer Order*, *The Raw Material Purchase Rate* equals to *Reorder Quantity* when the sum of *Raw Material Inventory* and *Raw Material Order* is less than *Reorder Level*. And the stock of *Order for Raw Material* depletes over the time of *Order Delay*.

Arriving raw material orders add to *Raw Material Inventory*. A *Job* is *Released* when there is *Customer Order* in the queue, the *Raw Material Inventory* is sufficient and the *Total Orders In The System* do not exceed the maximum level. The time it takes to release a job is dependent on *Job Process Time*, which is a constant value, and the *Queue's Effect On Job Process Lead Time*.

Thus, the *Job* is *Released* into *Queue for Step 1*. The *Processing Rate* for each step depends on the *Queue* length for that particular queue, and the *Actual Time* needed to process a *Job*, which is affected by the *Queue* length and a *Normal Processing Time*. The queue will not go negative. A *Normal Processing Time* is determined by each *Machine's Capacity* and the number of that particular type of *Machines*. The *Machine Capacity* fluctuates around an *Average Service Rate* with a certain *Standard Deviation*.

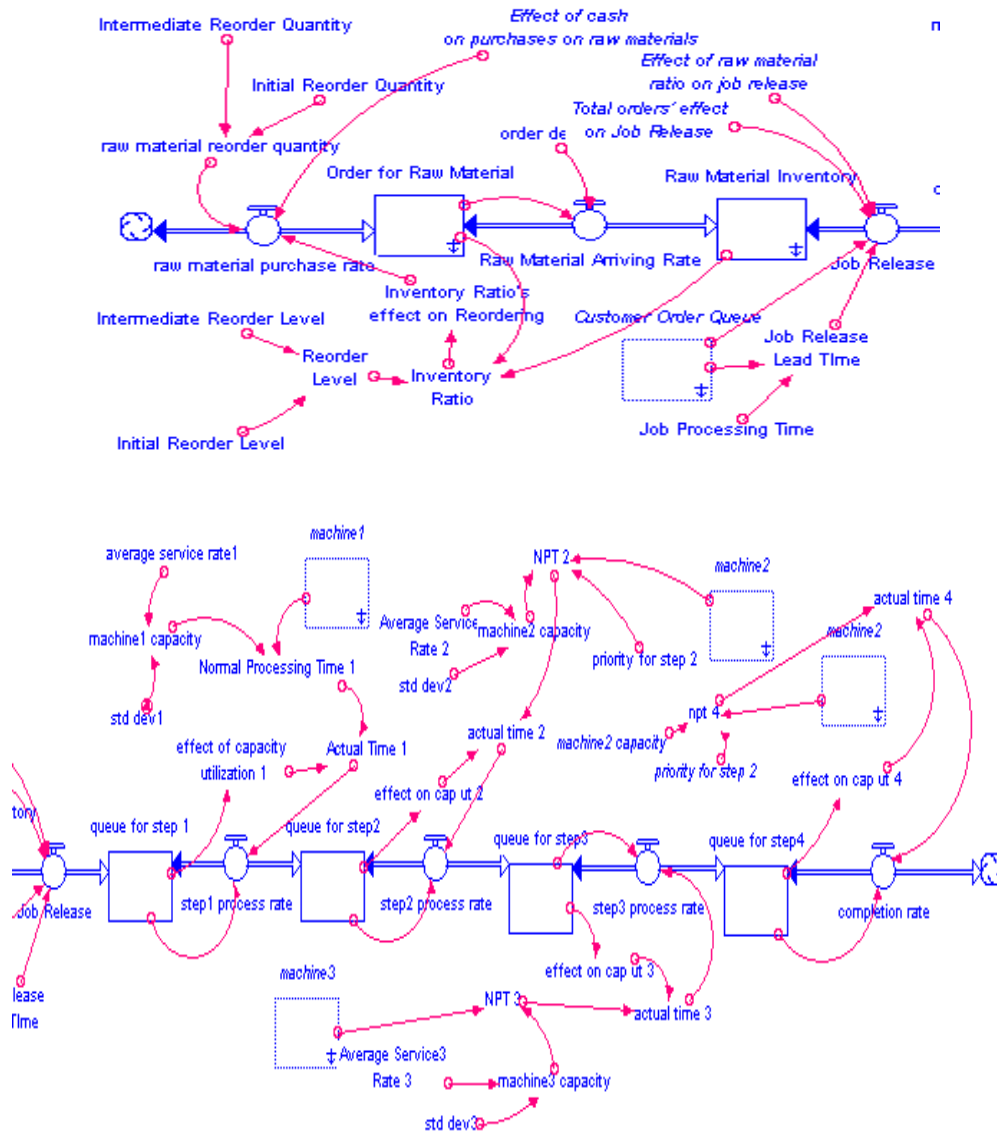


Figure 23 Stations Sector

**Equations**

**Stock 1: Order for Raw Material**

The stock measures the number of raw materials ordered and in the pipeline to be delivered to the factory.

$$\begin{aligned}
 \text{Order\_for\_Raw\_Material}(t) &= \text{Order\_for\_Raw\_Material}(t - dt) \\
 &+ (\text{raw\_material\_purchase\_rate} - \text{Raw\_Material\_Arriving\_Rate}) * dt
 \end{aligned}$$

INITIAL VALUE of Order\_for\_Raw\_Material = 0

**Inflows: raw\_material\_purchase\_rate** = raw\_material\_reorder\_quantity \*

Inventory\_Ratio's\_effect\_on\_Reordering \* Effect\_of\_cash\_on\_purchases

**Outflows: Raw\_Material\_Arriving\_Rate** = Order\_for\_Raw\_Material/order\_delay

Variable: Inventory Ratio

The inventory ratio measures whether the current stock of raw materials and the raw materials waiting to be delivered are greater than or below the reorder level.

**Inventory\_\_Ratio** =

(Raw\_Material\_Inventory+Order\_for\_Raw\_Material)/Reorder\_\_Level

Constants

**Reorder\_\_Level** = 24

**Raw\_Material\_Reorder\_Quantity** = 120

Stock 2: Raw Material Inventory

This stock measures the total number of raw materials in the system. Jobs are released into station 1 if the total number of orders in the factory is less than 100 and if there are enough raw materials.

**Raw\_Material\_Inventory(t)** = Raw\_Material\_Inventory(t - dt) +

(Raw\_Material\_Arriving\_Rate - Job\_Release) \* dt

INITIAL VALUE of Raw\_Material\_Inventory = 160

**Inflows: Raw\_Material\_Arriving\_Rate** = Order\_for\_Raw\_Material/order\_delay

**Outflows: Job\_Release =**

$$\text{Customer\_Order\_Queue} * \text{Total\_orders\_effect\_on\_Job\_Release} \\ * \text{Effect\_of\_raw\_material\_ratio\_on\_job\_release} / \text{Job\_Release\_Lead\_Time}$$

Variable

$$\text{Job\_Release\_Lead\_Time} = (\text{Customer\_Order\_Queue} + 1) * \text{Job\_Processing\_Time}$$

Constants

The Job Processing Time and Job Release Lead Time are used to calculate the total time an order has to wait before it is released.

The order delay is set to 3 as it takes 3 days for the raw material orders to get delivered.

$$\text{Job\_Processing\_Time} = 1/48$$

$$\text{order\_delay} = 3$$

Stock 3: Queue for Step 2

This stock shows the material from station 1 waiting to be processed at station 2. Once jobs are processed they are released to station 3.

$$\text{queue\_for\_step2}(t) = \text{queue\_for\_step2}(t - dt) + (\text{step1\_process\_rate} - \text{step2\_process\_rate}) * dt$$

$$\text{INITIAL VALUE of queue\_for\_step2} = 0$$

$$\text{Inflows: step1\_process\_rate} = \text{queue\_for\_step\_1} / \text{Actual\_Time\_1}$$

$$\text{Outflows: step2\_process\_rate} = \text{queue\_for\_step2} / \text{actual\_time\_2}$$

## Variables

The normal processing time (NPT) measures the time it takes the station to process an order. The actual time measures the time taken for the order waiting in queue and for the time it takes to be processed. The priority for step 2 variable shows how much of the capacity of machine 2 is set for queue 2 as opposed to queue 4. Below are formulations for station 1 and station 2. Variables machine1 and machine2 are the number of machines in each station.

$$\mathbf{Actual\_Time\_1} = \mathbf{Normal\_Processing\_Time\_1} * \mathbf{effect\_of\_capacity\_utilization\_1}$$

$$\mathbf{effect\_of\_capacity\_utilization\_1} = 1 + \mathbf{queue\_for\_step\_1}$$

$$\mathbf{Normal\_Processing\_Time\_1} = 1 / (\mathbf{machine1} * \mathbf{machine1\_capacity})$$

$$\mathbf{actual\_time\_2} = \mathbf{effect\_on\_cap\_ut\_2} * \mathbf{NPT\_2}$$

$$\mathbf{effect\_on\_cap\_ut\_2} = 1 + \mathbf{queue\_for\_step2}$$

$$\mathbf{NPT\_2} = 1 / (\mathbf{machine2} * \mathbf{machine2\_capacity} * \mathbf{priority\_for\_step\_2})$$

The machine1\_capacity variable is a variable used to calculate the capacity for machine 1 and is used similarly for other machine. The NORMAL function is a statistical function which outputs random numbers of the mean specified (average service rate) and specified standard deviation.

$$\mathbf{machine1\_capacity} = \mathbf{NORMAL}(\mathbf{average\_service\_rate1}, \mathbf{std\_dev1}, 5)$$

## Constants

$$\mathbf{std\_dev1} = 4$$

$$\mathbf{average\_service\_rate1} = 12$$

**std\_dev2 = 10**

**Average\_Service\_\_Rate\_2 = 30.6**

**priority\_for\_step\_2 = 0.5**

Function: Inventory Ratio's Effect on Re-ordering

The inventory ratio's effect on reordering variable is a discrete function which checks whether reorder level is less than the number of raw materials in the inventory and order pipeline.

**Inventory\_Ratio's\_effect\_on\_Reordering=IF(Inventory\_\_Ratio>1) THEN(0) ELSE (1)**

## **4.3 Model Testing**

We use extreme value testing and input response testing to verify the robustness of the model. We use sensitivity analysis to evaluate the model's accuracy. A well-constructed model should yield sensible and rational results under extreme conditions.

Most model behaviors are predictable. For example, if the machine purchase dramatically increases, the cash level should drop significantly, and the average lead time should decrease as well. Extreme value testing gives confidence of the model's validity.

All tests in this section follow the rule of the simulation that all settings could not be changed until 50 days in the game.

### **4.3.1 Extreme Value Testing**

Extreme value testing establishes confidence of a model's validity. The parameters chosen for extreme value testing include: incoming customer orders, desired number of machines, re-order level and quantity. All four parameters are exogenous variables. In Littlefield Simulation game, the number of incoming customer orders per day is generated by the system. Players can change the number of machines bought and sold, the re-order level, and quantity.

#### **4.3.1.1 Incoming Customer Order**

The average number of incoming customer orders per day is around 12. The range for the customer orders is from 3 to 24. Extreme values for incoming customer orders are used to test the model. If the model generates plausible behavior when tested with extreme values for incoming customer orders, then this builds confidence in the model's corresponding elements.

The values tested are 0 and 100. All other parameters are kept to their initial values at the start of gameplay. In Figure 24, the blue line shows the cash level over time when the number of incoming customer orders per day equals to 0. The cash level only increased slightly, which is due to the revenue earned from interest. The red line shows the model behavior when the number of incoming customer orders per day equals 100.

The cash level increases quickly at first when the orders are fulfilled within the lead time required by then contract. As customer order backlog increases and the lead time increases as a result, the cash level starts to increase at a lower rate which is only slightly higher than the slope of the blue line. The reason for the difference in slope is that in the second case, the cash level is higher than in the first case, which brings in more interest.

Figure 25 shows the trends of lead time when the number of incoming customer order equals zero and 100. When the value equals zero, lead time is constantly zero, since there are no jobs being processed. When incoming customer order is 100, the lead time steadily increases as the customer order backlog increases.



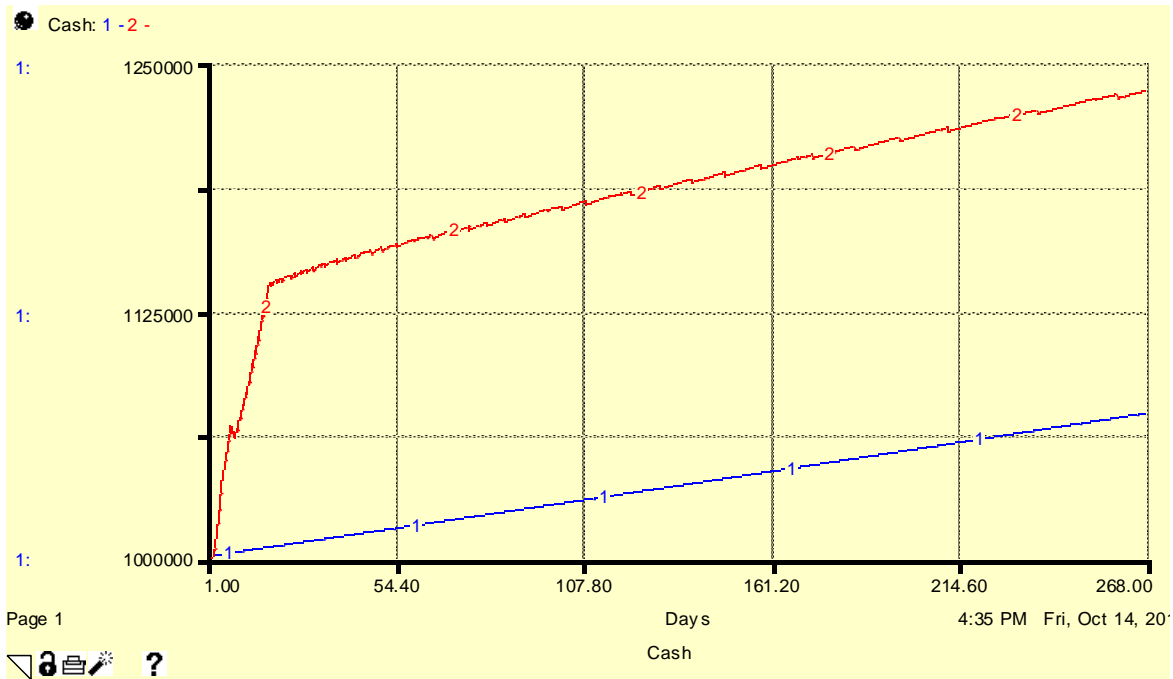


Figure 24 Cash Level for Extreme Values of Customer Orders

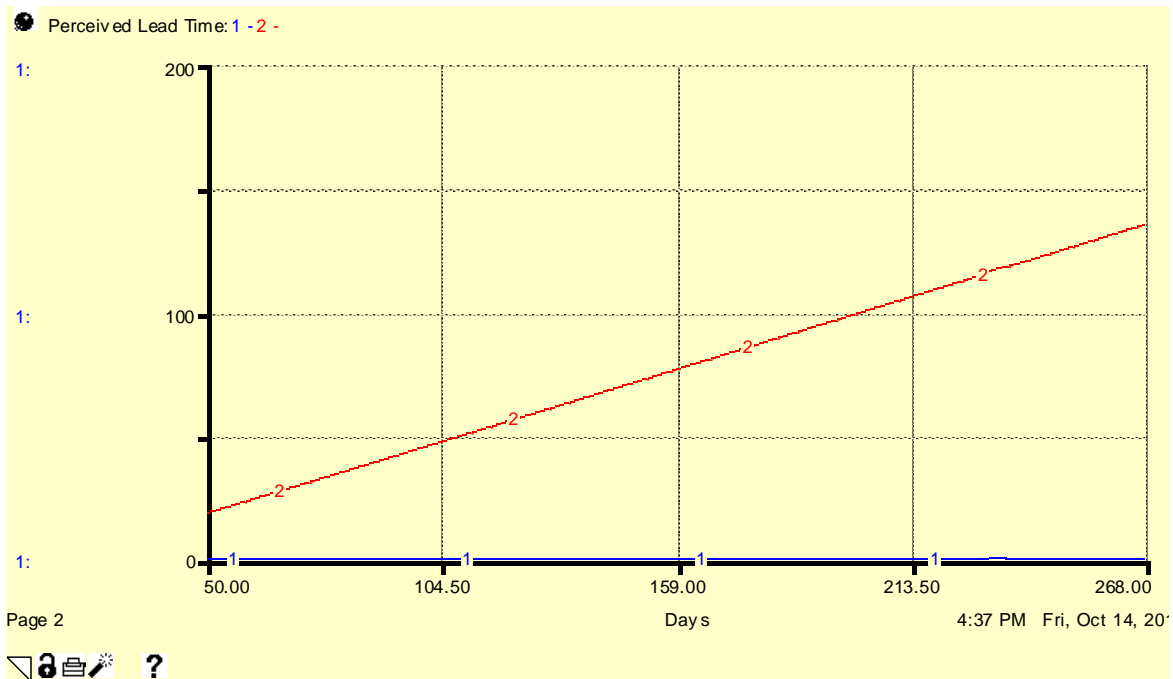


Figure 25 -Lead Time for Extreme Values of Customer Orders

#### 4.3.1.2 Desired Number of Machines

The initial numbers of machines for station 1, station 2 and station 3 respectively are three, one, and one. Extreme value testing for desired machine levels sets these values at greater and lower levels than the initial values. If in response to the test the model still generates reasonable behavior, then we build confidence in the model's corresponding elements.

The values selected for testing are zero and 20, for each machine. All other variables are kept to the same initial values at the start of the gameplay. The blue line in Figure 25 shows the cash level when desired number of machines equals zero; the red line shows cash level when desired number of machines equals 20.

When desired number of machines is zero, cash level keeps on gradually increasing because of the revenue from machines sales in first 30 days and the interest earned throughout the game. When the desired number of machines is set to 20 at day 50, the cash level immediately drops, and stays constant at a low value throughout the game-run. The reason is that purchasing machines takes considerable amount of cash. The value that cash level stays constant at is affected by cash constraint settings, which limit investment in machines based on current cash level.

Figure 27 shows the lead time's responsive behavior for extreme values of desired number of machines, the blue line correspond to zero and the red line correspond to 20.

When there is not a single machine in the system, jobs cannot be processed due to the lack of capacity. As a result, lead time keeps increasing linearly as the customer order backlog increases. When the machines are being purchased too quickly, the lead time still increases linearly as the aggressive investment in machines hinders the purchase of raw material, and thus stops production. This increases both customer order backlog and in the lead time.

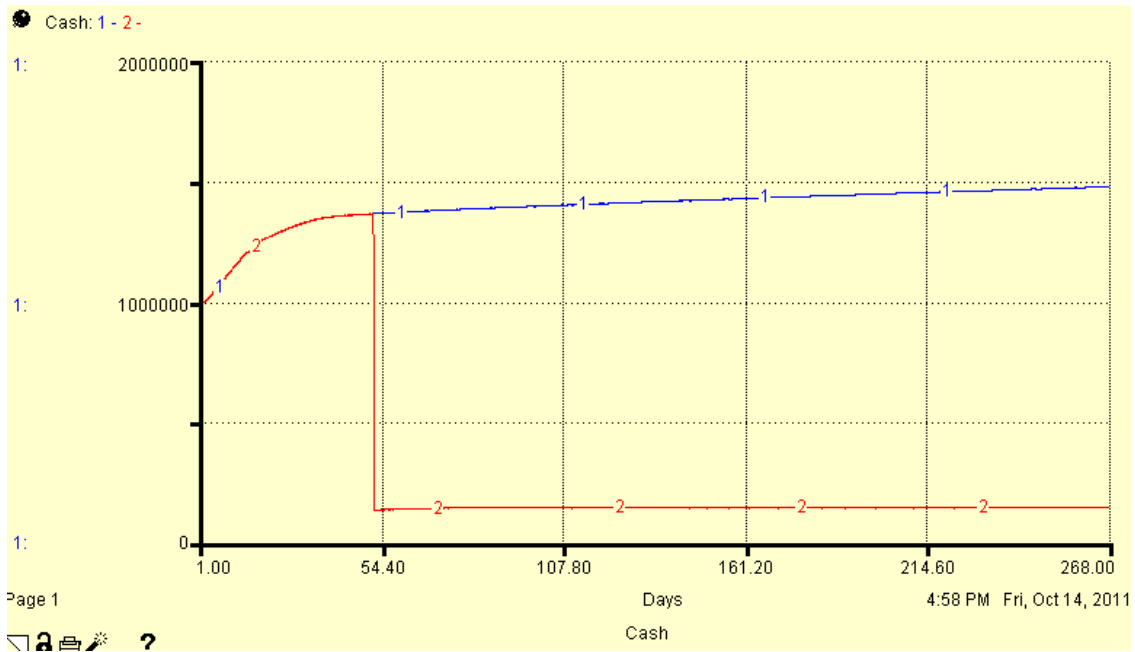


Figure 26 Cash Level for Extreme Values of Desired Machines

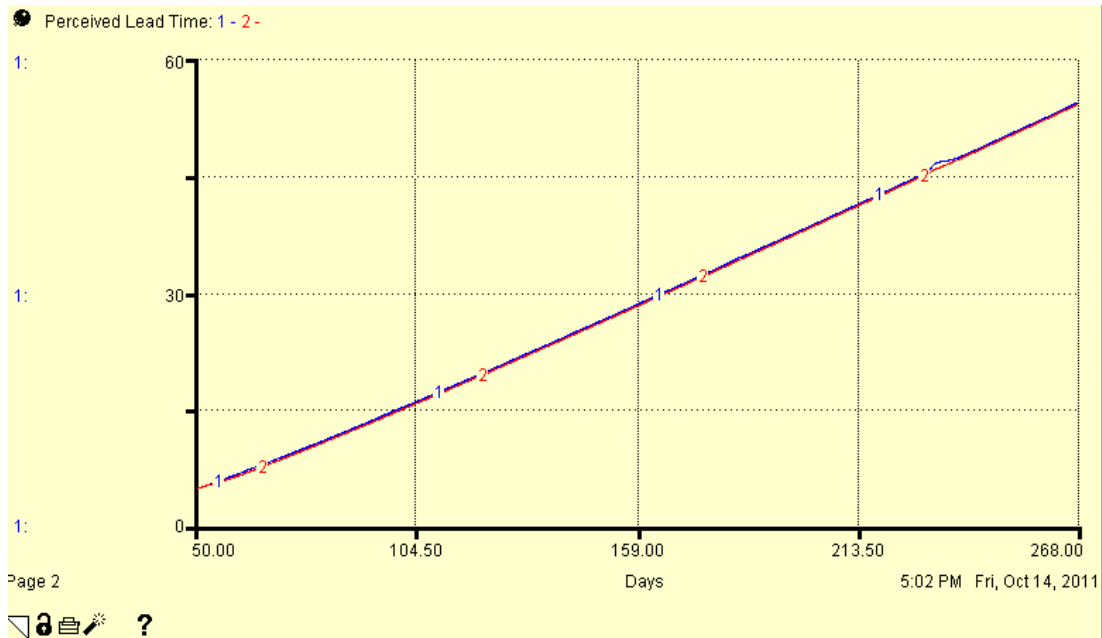


Figure 27 Lead Time for Extreme Values of Desired Machines

### 4.3.1.3 Reorder Level and Reorder Quantity

The initial values for re-order level and re-order quantity, in the game, were 24 and 160 correspondingly. Extreme value testing for re-order level and re-order quantity requires values much greater than and much less than the initial values. However, high values for re-order

quantity are part of the profit maximizing strategy as discussed in section 5.1.5. High values for re-order level do not have significant impact on the system. Therefore, only extremely low values for re-order level and re-order quantity need to be tested in extreme value testing.

Zero is selected for both tests for re-orders level and re-order quantity. All other variables are kept to the same initial values at the start of the simulation, while the two parameters tested are set to zero at separate times. Figure 28 and Figure 29 show cash level and lead time's response behavior. The blue lines show the model's reaction when re-order level equals zero, and the red lines show the model's reaction when re-order quantity equals zero.

As it takes four days for the ordered raw materials to arrive, lead time increases linearly due to the insufficient raw materials when re-order level is set to zero. When re-order quantity is set to zero, production would stop due to the lack of raw material. In both cases, customer order backlog will keep increasing and lead time will keep increasing linearly. The cash earned through interest contributes to cash level in both cases. The cash level increases slowly when re-order level equals zero since only a certain amount of jobs can be processed. The cash decreases

slowly since there are no jobs.

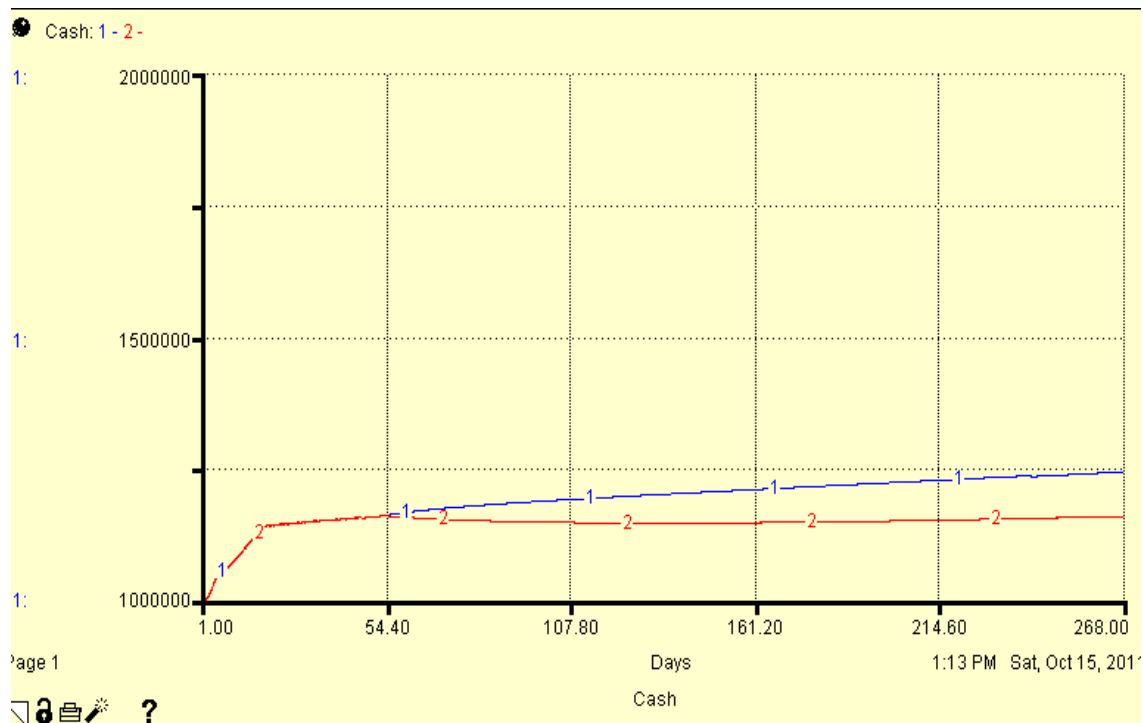


Figure 28 Cash Level for Zero Reorder Level and Reorder Quantity

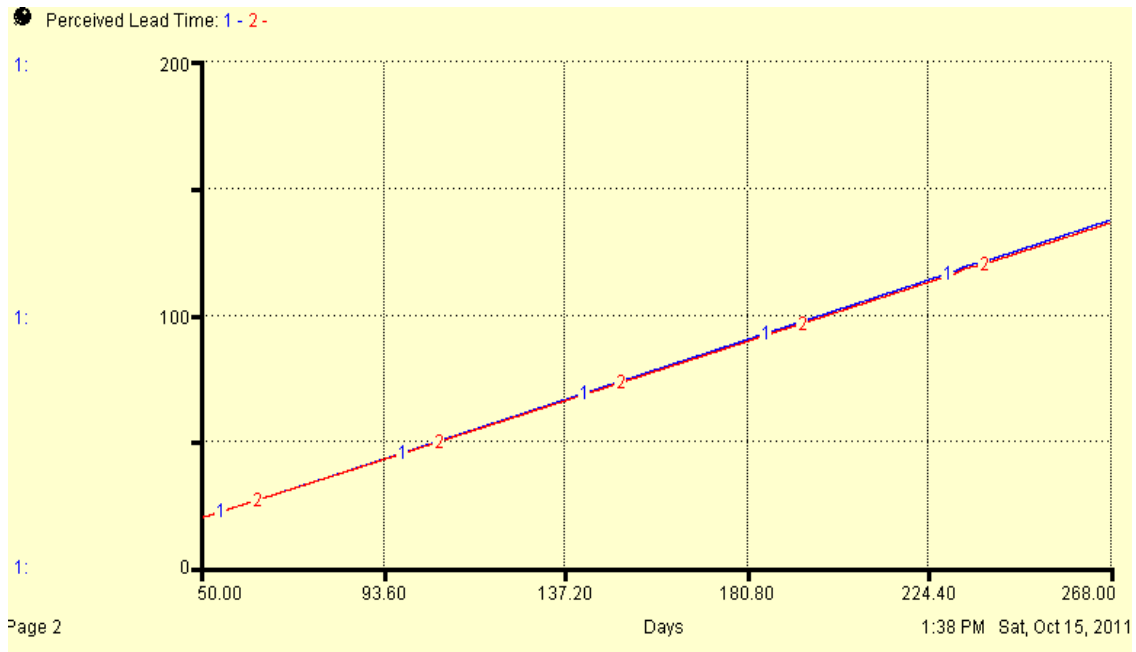


Figure 29 Lead Time for Zero Reorder Level and Reorder Quantity

### **4.3.2 Input Response Testing**

This section tests the model's responsive behavior to different input patterns. In the Littlefield game the incoming customer orders are random values that follow a normal distribution, with a mean of approximately 12 and a standard deviation of 3. As the standard deviation is of statistical significance it is prudent to test the model with different inputs. The rationale is that the intrinsic behavior of the model should not be affected by variation in inputs. The testing starts with constant incoming customer orders. Afterwards, we test the step input, sine wave input, and random input. In each case, the response of cash level is compared to the parameters' response to constant input. The similarity of the responses establishes confidence in the model, and serves as a basis for strategy design.

#### **4.3.2.1 Constant Incoming orders**

Figure 30 shows the cash response with constant incoming customer orders. This is the baseline run and the other input responses are compared to it. Figure 30 shows that not much cash is earned throughout the simulation.

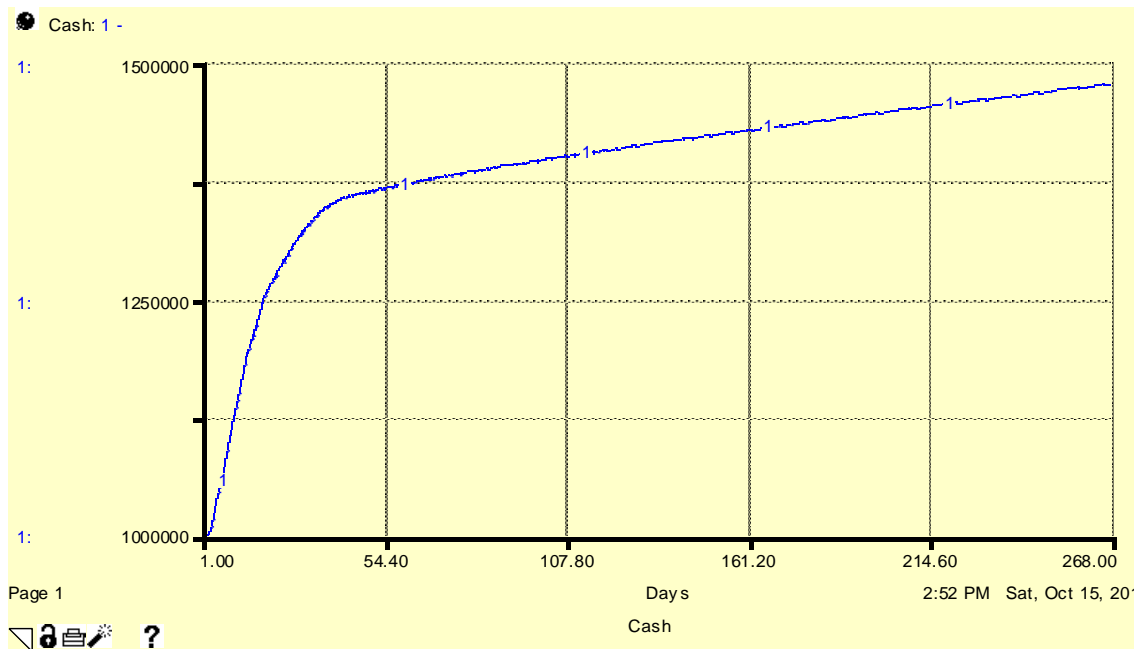


Figure 30 Cash Level With Constant Incoming Customer Orders

#### 4.3.2.2 Step Response

The step response shows how quickly the system returns to steady state, and if there is unwanted endogenous relationships in the model which may stop the system from returning to a steady state.

The red line in Figure 31 shows the cash level's response to a step input. In the base run, the cash level increases at a steady but slow rate after 50 days. Cash level's response of the step input of an equal magnitude as the base run, is to start rising at the same steady slow rate after the adjustment period.

In the testing, incoming customer orders are changed to the following equation:

$$0+\text{STEP} (\text{AMPLITUDE}= 12, \text{TIME} = 52)$$

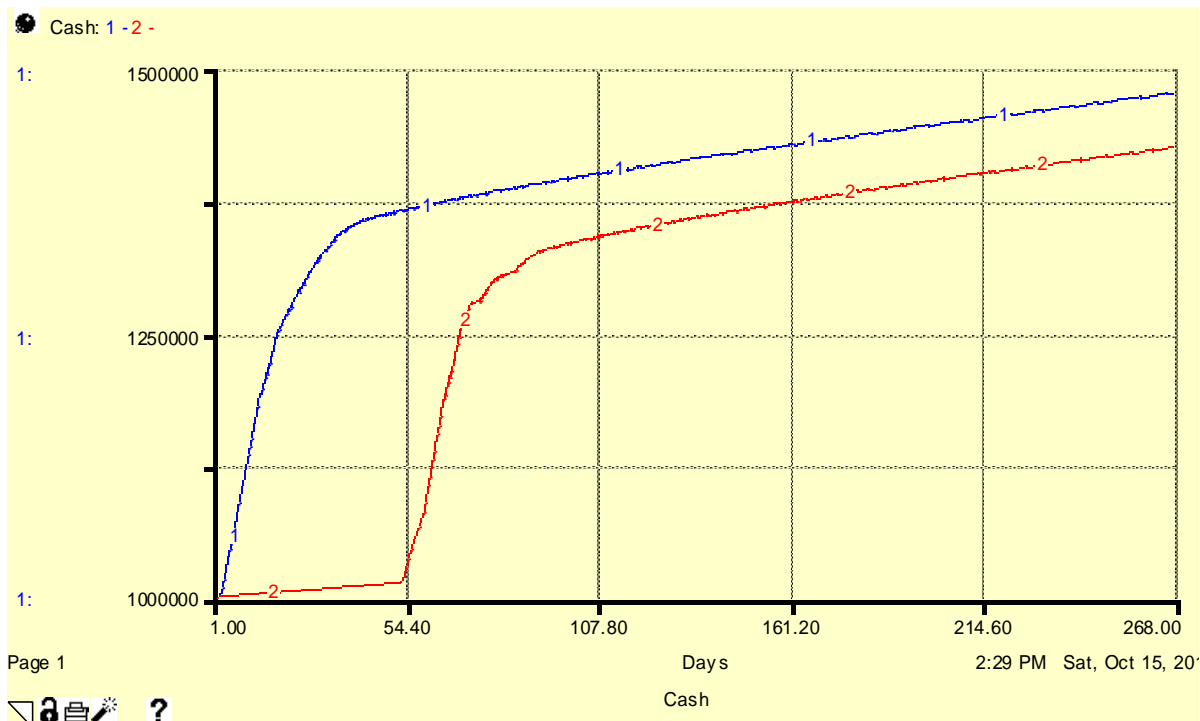


Figure 31 Cash Level Step Response

#### 4.3.2.3 Sine Wave Input

The red line in Figure 32 shows the system's response to a sine wave input. The incoming customer orders equation is changed to:

$$12+12*\text{SIN}(2*\pi*\text{TIME}/5)$$

The sine wave has the amplitude of 12 and a period of 5. The value varies between 0 and 24.

The sine wave response deviates only slightly from the constant input response; therefore, we conclude that the system is very stable.



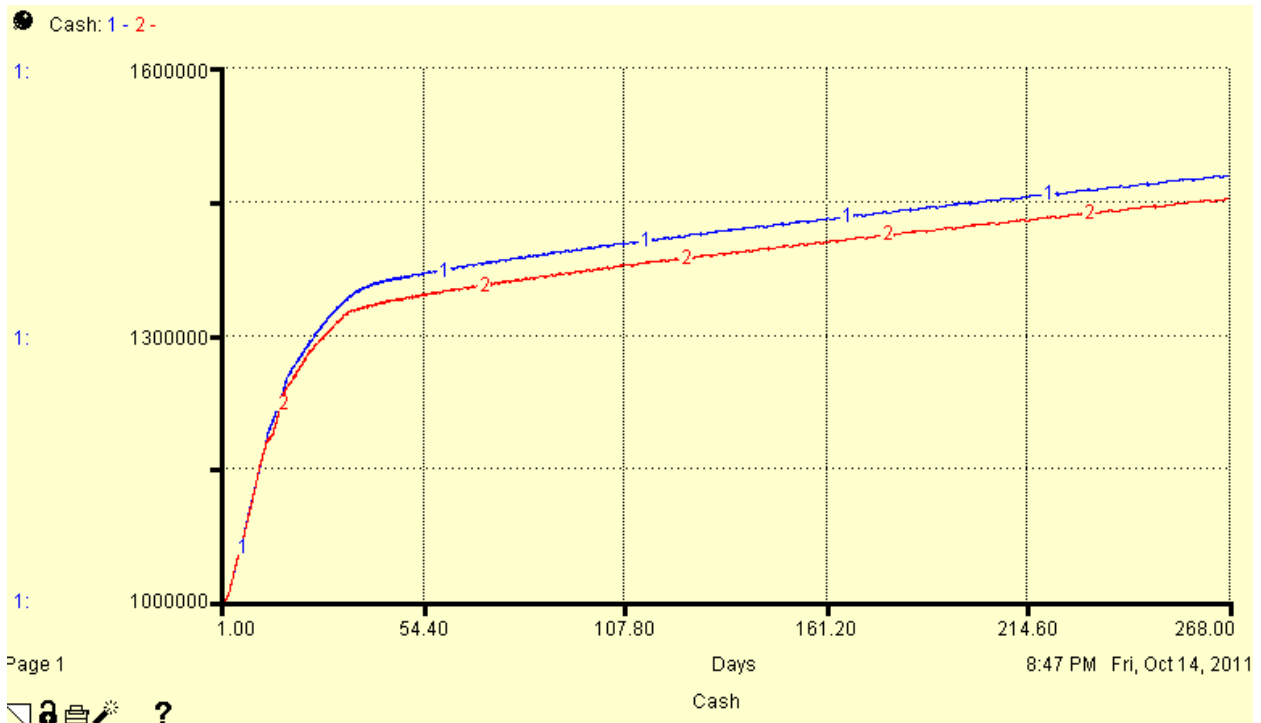


Figure 32 Cash Level Sine Wave Response

#### 4.3.2.4 Random Input Response

The input response testing culminates with the random input response testing. The random input response is important as it resembles the actual situation in Littlefield simulation, where the input is random. The red line in Figure 33 shows the response of the system to a random input. The incoming customer order was changed to:

$$(NORMAL (12, 3) + RANDOM (-6, 9))$$

The red line shows little deviation from the constant input response; therefore, we conclude that the cash level is almost independent from the type of input as long as the mean of the input is constant. This conclusion simplifies later tests as the incoming customer orders can be kept constant.

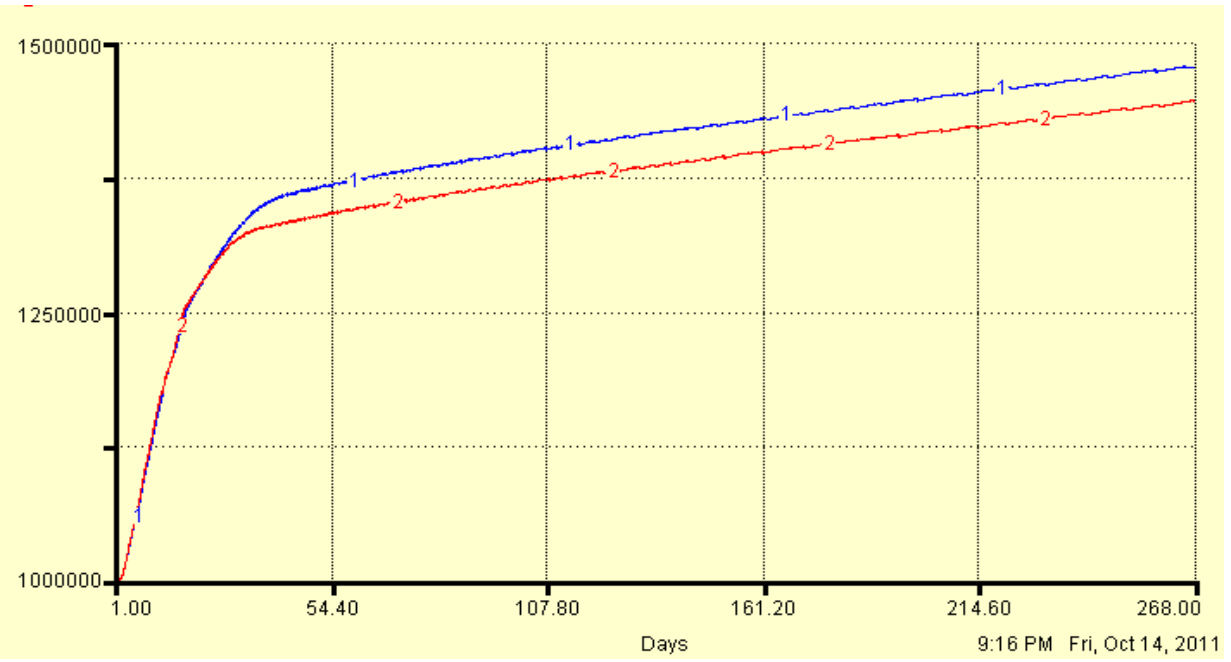


Figure 33 Cash Response to a Random Input

## 5.0 Game-playing Strategies for Maximizing Profit

The game player has control over 3 aspects of the game to maximize profit. Firstly, the reorder level and reorder quantity of the raw materials can be changed. Secondly, desired lead time of the different machines could be changed; this decision directly affects purchasing and selling of machines. Lastly, priority between stage 2 and stage 4 of the production process can be changed.

### 5.1 Analysis of Current Strategy

#### 5.1.1 Initial Condition and Input to the System

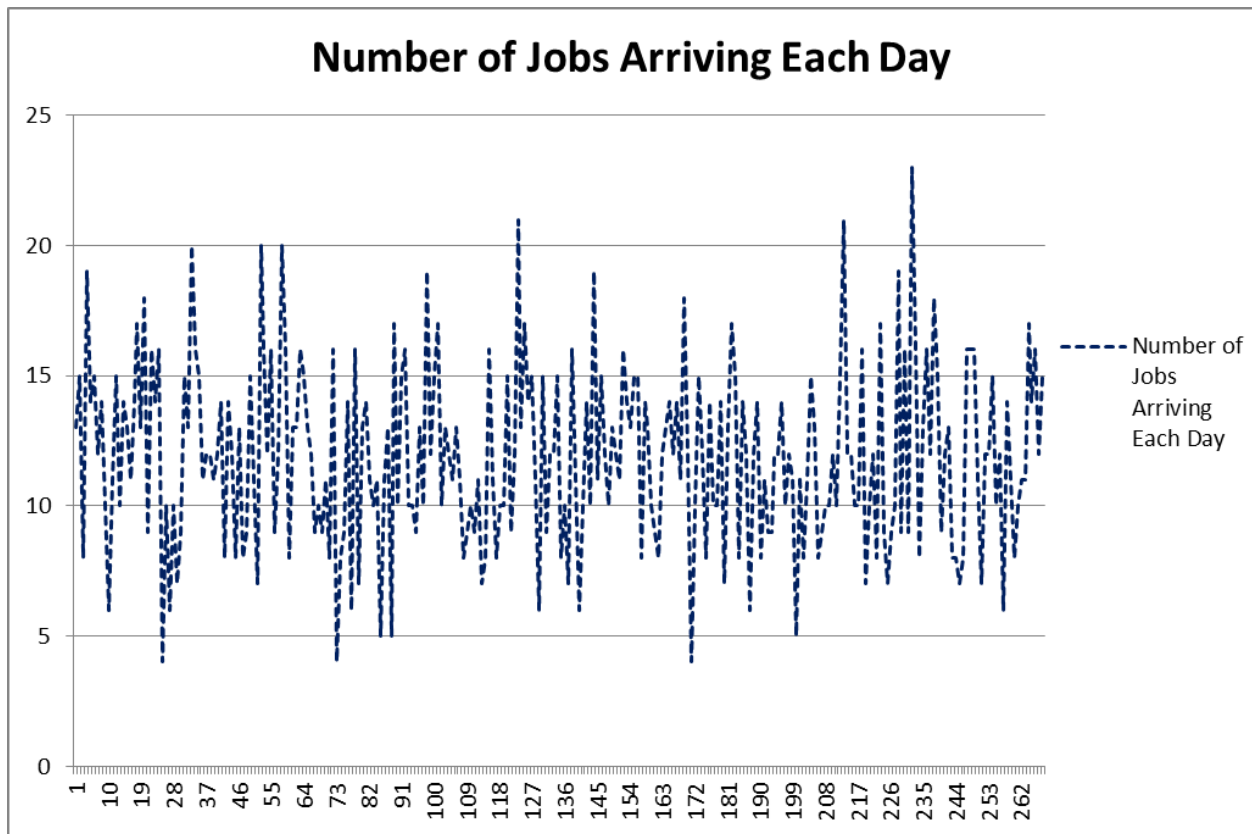


Figure 34 Incoming Customer Orders per Day

Understanding the trend of incoming customer orders (Figure 34) is essential to decision-making. In the simulation, the volume of incoming customer orders greatly affects decision-

making on items such as machines purchases, re-order level and quantity settings, and the priority of stations. These decisions are often based on the forecast of incoming customer order in the model. As listed in Figure 35, means, standard deviations, and range are calculated. The mean shows that the volume of incoming customer orders fluctuates around 12 per day.

Parameter	Value
Average	11.91
Standard Deviation	3.28
Range	3~24

Figure 35-Mean, Standard Deviation and Range of Incoming Customer Order per Day

Range	Frequency	Weight
15 < orders	59	22.01%
8<=orders<=15	149	55.60%
orders < 8	60	22.39%

Figure 36- Order Distribution

Figure 36 shows order distribution. The bulk of orders are between 8 and 15 units. The distribution itself appears to be symmetrical.

### 5.1.2 Contract Selection

There are three contracts available in the Littlefield game; for each contract selected, revenue per job decreases as the lead time increases. There are two feasible strategies: alternating contract during the gameplay, and selecting one contract throughout the gameplay. If the alternate contract strategy is chosen, purchasing additional machines would be unnecessary, but the latest trend of incoming customer orders needs to be tracked closely: when the incoming customer orders is low, and machine utilization rates are low as a result, contract 3 should be selected; when the incoming customer orders is high and the machine utilization rates are high,

contract 1 or contract 2 should be selected, based on the current number of machines. It is difficult to determine the timing of contract change as well.

If a fixed contract is chosen, more cash would be spent on machine purchasing. Aggressive machine-purchasing in the initial stage leads to relatively low utilization rates. As a result, fewer adjustments of the number of machines are required during the gameplay. It is noticeable that contract 3 grants the highest profitability and contract 1 offers the lowest. The daily revenue from contract 1, 2 and 3 are \$750, \$1000, and \$1250 respectively, when the orders are completed within required lead time.

Based on the analysis above, we recommend choosing a fixed contract strategy with contract 3.

### **5.1.3 Ideal Number of Machines to Purchase**

The maximum lead time without losing revenue for contract 3 is one day. The loss is in both in cash and opportunities to generate profit. Therefore, it is recommended to keep the lead time below one day.

According to the forecast mentioned in the section 5.1.1, the possible highest mean (+1 standard deviation) of the number of incoming customer orders is 15 per day, there are 22% of the days when the number of incoming customer orders is greater than 15, and the maximum value is 20 per day.

Therefore, when selecting fixed contract strategy with contract 3, the ideal capacity should be around 17 to 18 jobs per day (+2 standard deviation). The utilization rates will be around 70% under this setting, according to the data from game-run.

A high process rate of station 1 is desired to guarantee sufficient time for stations 2 and 3 to process their respective queues.

In the game-run, we had five machines and the peak utilization level was around 70% (Refer to the graph Station Utilization Station 1 from Sec. 3.2). Reducing the number of machines leads to an increase in utilization rate. Specifically, when reducing the number of machines at station 1 to four, the utilization starts to fluctuate around 90%, and peaks at 100%. It increases lead time, and therefore affects revenue. We conclude that the ideal number of machines at station 1 is five.

Similarly, we analyzed station 2 and station 3 utilization rates from our Team 2 data. The conclusion is that the ideal number of machines for station 1, station 2 and station 3 is five, three and two, respectively.

#### **5.1.4 Re-order Level**

Since fixed contract strategy with Contract 3 is selected, raw material shortage would lead to a loss in profit. Safety stock is a level of extra stock that is maintained to lessen the danger of this raw material shortage due to uncertainties in customer orders. A commonly used approach for the calculation of the safety stock is based on the average customer orders, the lead time it takes for raw materials to arrive after the order is placed, the service level-the desired probability that the chosen level of safety stock will not lead to a raw material shortage- and the standard deviation of customer orders.

The safety stock is calculated below:

Z=1.64 is selected to achieve 95% service level.

*Safety Stock* =

$$Z\sqrt{\text{Lead Time} \times \text{Stddev of Demand}^2 + \text{Average Demand}^2 \times \text{Stddev of Lead Time}^2} = 1.64 \times \sqrt{4 \times 3.28^2 + 11.91^2 \times 0} = 10.76 \text{ jobs}$$

The factors that determine the appropriate re-order point are the safety stock, average incoming customers and the lead time. The re-order point calculation is shown below:

$$\text{Re - order Point} = \text{Average Lead Time} \times \text{Average Demand} + \text{Safety Stock} = 4 \times 11.91 + 10.76 = 58.6 \text{ jobs}$$

We chose 60 jobs as re-order level.

### 5.1.5 Re-order Quantity

In the game, cash earns 10% per year. Accordingly, the daily interest rate is:

$$\sqrt[365]{1.1} - 1 = 0.00026 = 0.026\%$$

Therefore, the break-even point for ordering raw material each day is:

$$\frac{\$1000}{.00026} = \$3,846,153$$

When there is \$3.85 million cash, the revenue from interest equals to the cost of an order. It means if cash is over \$3.85 million, there will be an opportunity cost resulting from ordering raw material. Considering actual situations of the game-play, the opportunity cost is not significant.

The Economic Order Quantity (EOQ) is calculated below:

$$Q = \sqrt{\frac{2 \times \text{Annual Demand} \times \text{Fixed Cost per Order}}{\text{Holding Cost}}}$$

Since there is no holding cost in this game, the theoretical value for EOQ is infinitely large. Therefore, it is recommended to set a large re-ordering quantity to reduce cost.

### 5.1.6 Machine Priority

Experiment result with the model indicates that the priority of station 2 machines will only affect the lead time when capacity is insufficient. Since both the utilization of station 2 and of station 3 stay steadily below 80% in our gameplay, we concluded that the priority of station 2 won't affect our lead time and revenue. Therefore it's recommended to use default setting of priority, which is first-in-first-out.

## 5.2 Predictions of Model Behavior

Calculation indicates that the maximum ending cash is \$3.5 million. The calculation process is described in the following paragraph.

Daily job arrival has an average of 12 and a standard deviation of 3. Revenue per job for contract 3 is \$1,250. Cost for kits required per job is \$600. Daily profit is calculated as:

$$(\text{Revenue per Job} - \text{Cost per Job}) \times \text{Daily Job Arrival} = (\$1250 - \$600) \times 12 = \$7,800$$

The maximum profit that can be generated is:

$$\$7,800 \times 218 = \$1,700,400$$

The initial capital is \$1 million, and the revenue generated from first 50 days is around \$0.8 million. Adding all three values up, we conclude that the maximum ending cash is around \$3.5 million.



Due to the complexity in calculations and model structure, small discrepancies between expected value and actual value are inevitable.

### 5.3 Profit-Maximizing Strategy Implementation and Testing

This section tests the strategy designed in the previous section. First, the model is changed appropriately to include a policy sector. Thereafter, incremental changes are made until all the policy changes take effect. For each change made, the model is simulated so that effects of the change can be seen. The model testing section showed that the system response is almost independent of the type of input; therefore, in order to simplify analysis only constant orders will be used. When one parameter is changed, all the other parameters are kept constant. Unless stated, the parameter values are always the same as during the initial Littlefield game. These parameters are shown in the Appendix Section 9.3 The following strategies are tested:

- 1) Initial run with reactive strategy
- 2) Changing contract type to contract 3
- 3) Implementing optimum re-order level and re-order quantity
- 4) Changing the number of machines for maximum utilization

In each case the effectiveness of the change in the overall policy design is verified by seeing the increase in cash level at the end of the game. In the end, the strategy is tested with random input for verification. The random input is similar to the input in Littlefield Simulation.

Note that the settings cannot be changed until day 50 according to the rules of Littlefield Simulation.

### 5.3.1 Initial Run with Reactive Strategy

As shown in Figure 37, when the model runs with the initial settings given in the game, the ending cash level on day 268 is \$1.478 million so the profit earned is \$0.478 million

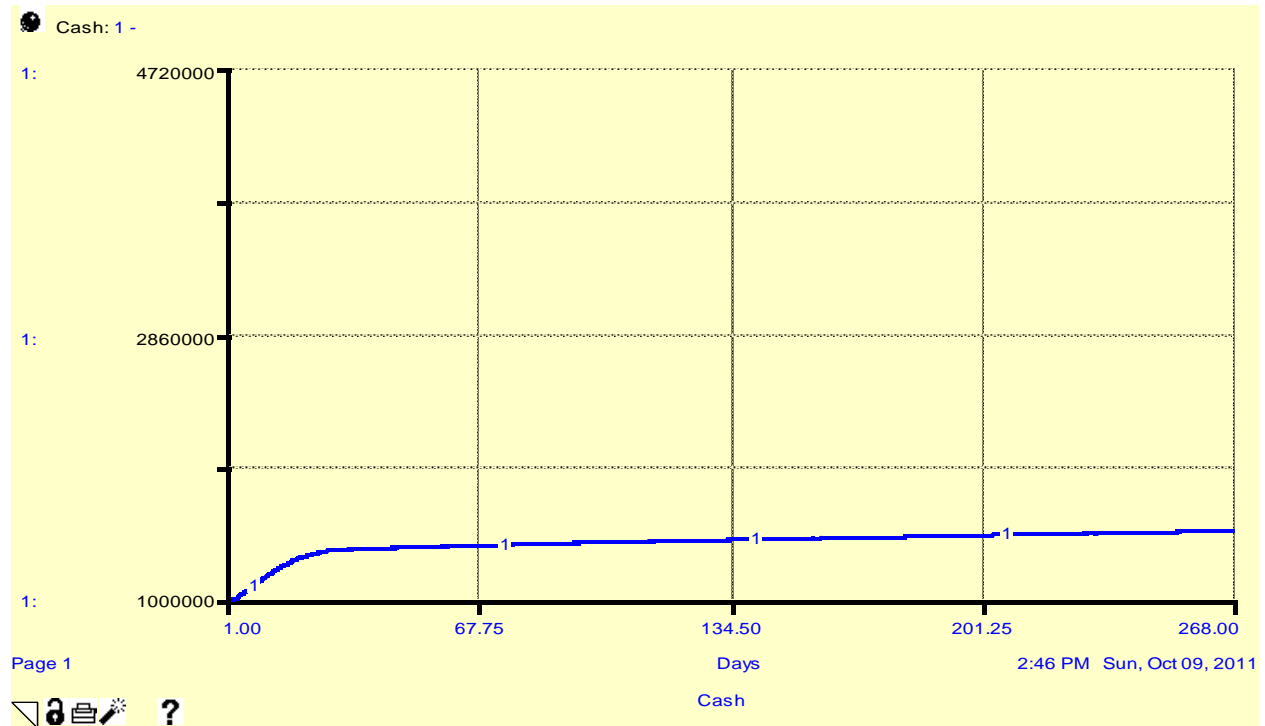


Figure 37 Cash level with a reactive strategy

During the test-run, contract 1 is selected for the first 50 days and after 50 days. Selecting contract 1 is one of the initial settings of the game. Figure 38 below shows the 3 different contract types. When the value for the contract type is 1 that particular contract has been chosen. When the value for the contract type is 0; that contract has not been chosen. After 50 days, the model automatically selects the contract 1, which is a result from the built-in algorithm that selects the most profitable contract type based on perceived lead time.

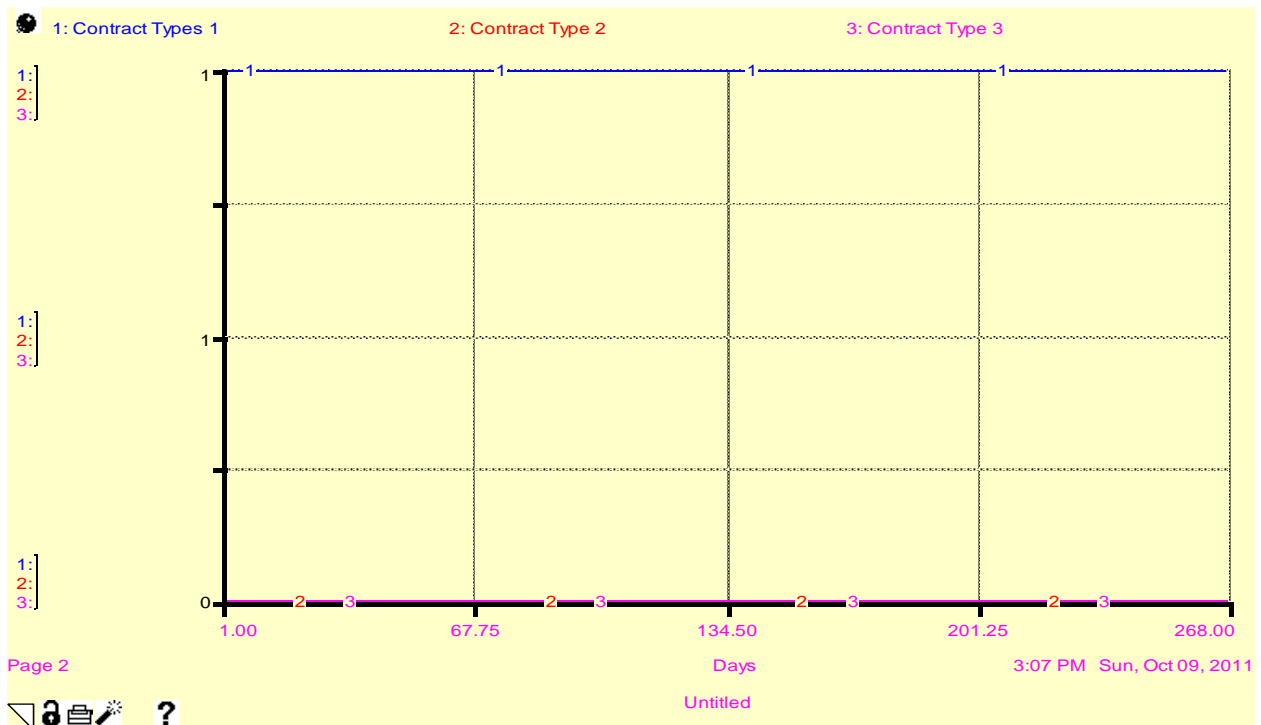


Figure 38 Contract Type Chosen When Policy Is Not Used

Since the objective is to achieve the maximum cash level at the end of the game, it is recommended to choose contract 3 from day 51 to day 268.

### 5.3.2 Switching to Contract 3

The analysis of revenue per contract, suggests that contract 3 brings in the highest profit. In this section, we are testing selecting contract 3 as a strategy.

The model is modified to include a policy sector. The goal of the sector is to choose one contract type after day 50. The sector is used to select contract 3 throughout the game. The policy sector is shown in Figure 39.

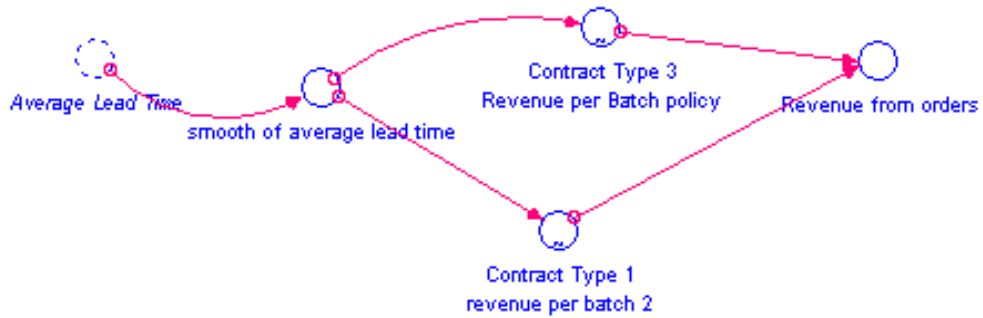


Figure 39 Policy Sector

### Policy Sector Description

The lead time is smoothed for a 1 day period since the fluctuations in the lead time cause errors in the calculations of the order revenue:

$$smooth\_of\_average\_lead\_time = SMTH1(Average\_Lead\_Time,1)$$

### Contract type 1 revenue per batch

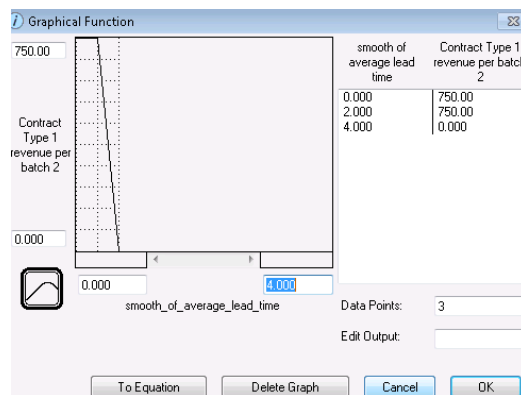


Figure 40 Contract Type 1 Revenue Graph

A lookup function in Figure 40 is used to calculate the revenue for the first 50 days when contract type 1 is selected

## Contract type 3 revenue per batch

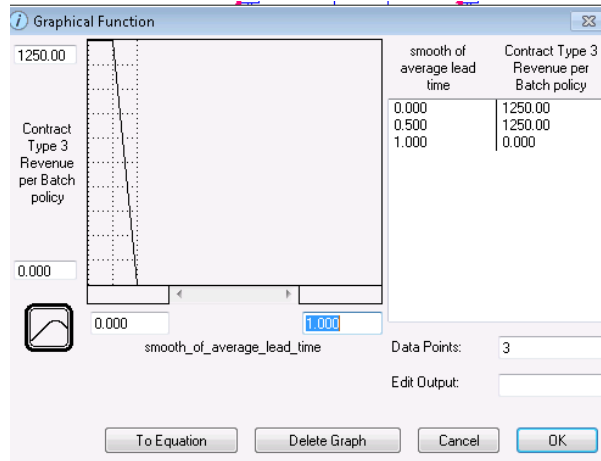


Figure 41 Contract Type 3 Revenue Graph

A lookup function in Figure 41 is used to calculate the revenue from day 51 to day 268.

The following formula calculates the revenue per order:

**Revenue from orders**= IF (TIME<50) THEN Contract\_Type\_1\_revenue\_per\_batch\_2  
ELSE Contract\_Type\_3\_\_Revenue\_per\_Batch\_policy.

## Results:

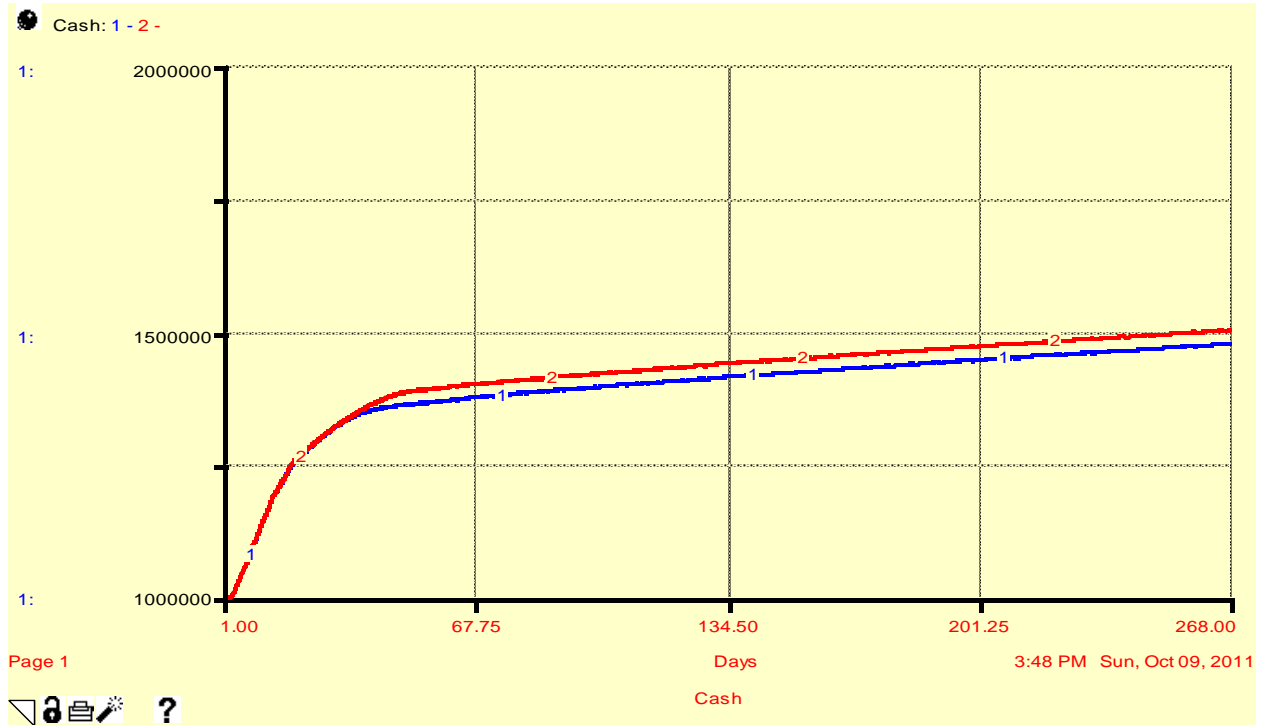


Figure 42 Cash Level when Contract 3 is Chosen

Figure 42 shows that proactively choosing contact 3 increases the cash level to \$1,504,409 (red line). It is \$26,845 higher than reactively choosing contracts, the strategy presented in last sector.

### 5.3.3 Implementing optimal re-order level and re-order quantity

As discussed in section 5.1.5, the optimal values for the re-order level and re-order quantity can be determined. The theoretical optimal value for the re-order level is 60 jobs, or 3600 kits. That calculation, however, did not include purchasing's effect on the revenue generated from interest. The theoretical optimal value for the re-order quantity is infinitely large. However, since the game has a limited number of days and the goal is to maximize cash at the end of the simulation, over-purchasing raw materials is not recommended. Therefore, the optimal re-orders level and re-order quantity should be determined through experimentations with the

model. The experiment results showed that the optimal value for the re-order level is 100 and the optimum value for re-order quantity is 2500.

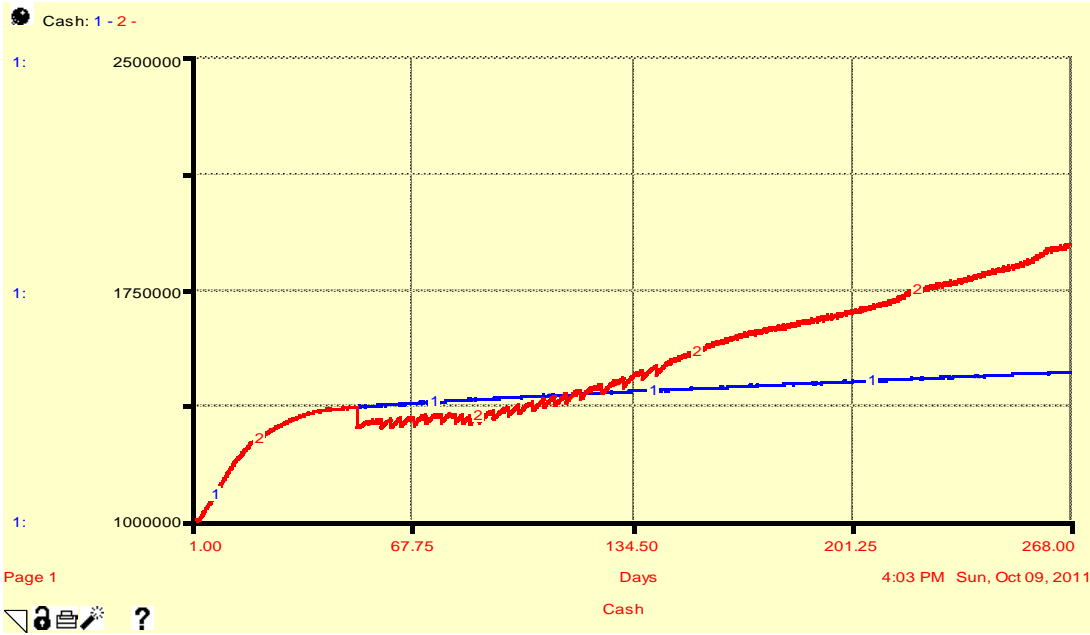


Figure 43- Implementing Economic Reorder Quantity and Economic Reorder Level

In Figure 43, the blue line shows the level of cash when choosing contract 3, and using the initial settings with 24 as re-order level and 160 as re-order quantity, while the red line shows the level of cash over time when the optimum values are implemented: 100 as re-order level and 2500 as re-order quantity. The optimum value’s ending cash level is \$1.896 million which is \$0.391 million higher than the result of strategy in 5.2.2. There is a dip in cash level on the red line because of a large amount of raw material is bought

### 5.3.3.1 Optimum Reorder Level

This experiment shows that increasing the re-order level above 100 does not lead to an increase in cash and decreasing it leads to decreases in cash. This is shown in Figure 44. All four lines are generated when contract 3 is chosen and re-order quantity is 2500 jobs. Only re-order



level is changed. The blue line shows cash over time when re-order level is 100 jobs, which leads to an ending cash level of \$1.895 million (blue line). Decreasing the value to 60 jobs leads to a decrease of cash level to \$1.565 million (red line). Decreasing the value to 90 jobs leads to a decrease of cash level to \$1.894 million (pink line). Increasing the value to 110 jobs also decreases cash level to \$1,847,550 (green line). Therefore, 100 jobs is the optimum value for re-order level.

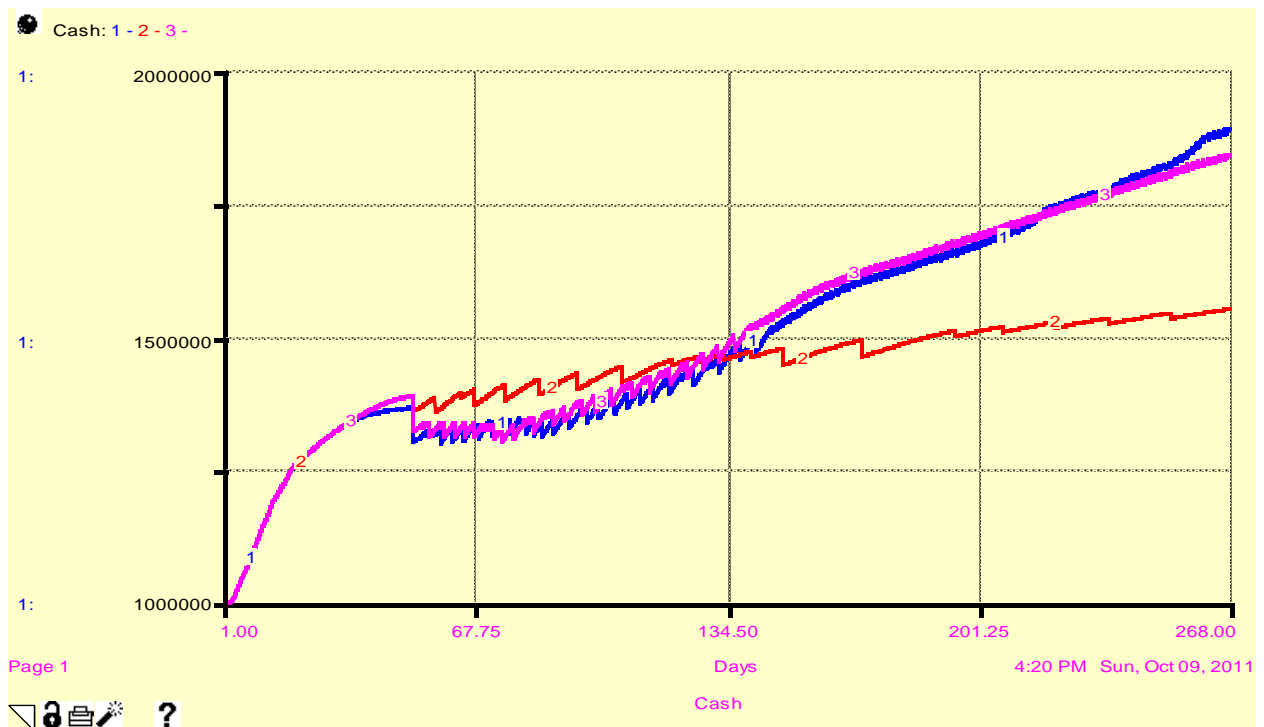


Figure 44- Changing the Reorder Level

### 5.3.3.2 Optimum Reorder Quantity

Proving that 2,520 kits is the optimum reorder quantity is more challenging than proving that 100 is the optimum reorder level. This is because the deviation in the graph when changing the reorder level is less noticeable. Therefore, Figure 45 is zoomed into the last few days of the game. All three lines are run when contract 3 is chosen and when the re-order level is set to 100 jobs. The blue line shows cash over time when re-order quantity is set to 2,500 kits; and it gives

the highest cash level. The red line and pink line are cash levels over time when the re-order quantity is set to 3,000 kits and 2,000 kits respectively. Both of them have lower cash level by around \$50,000.

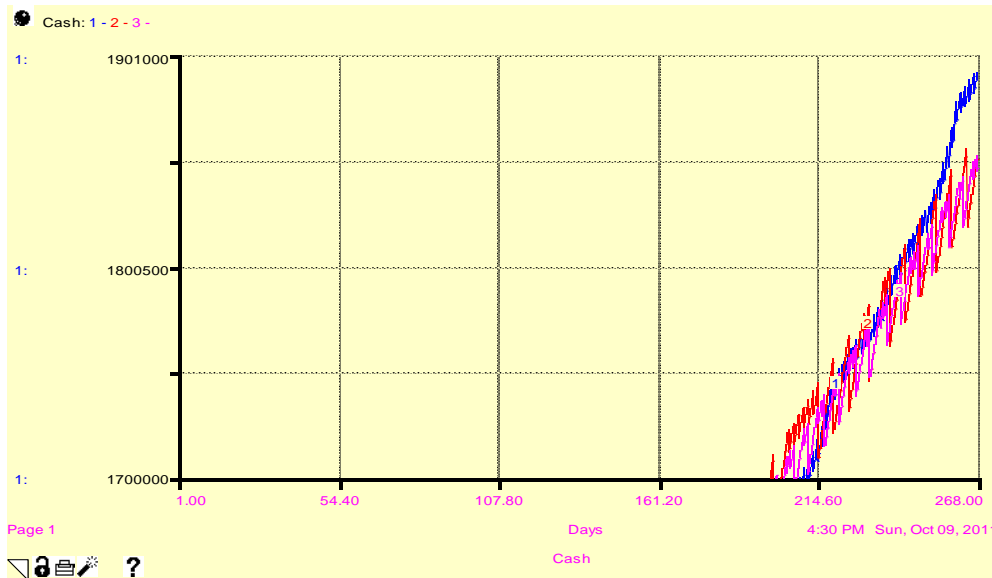
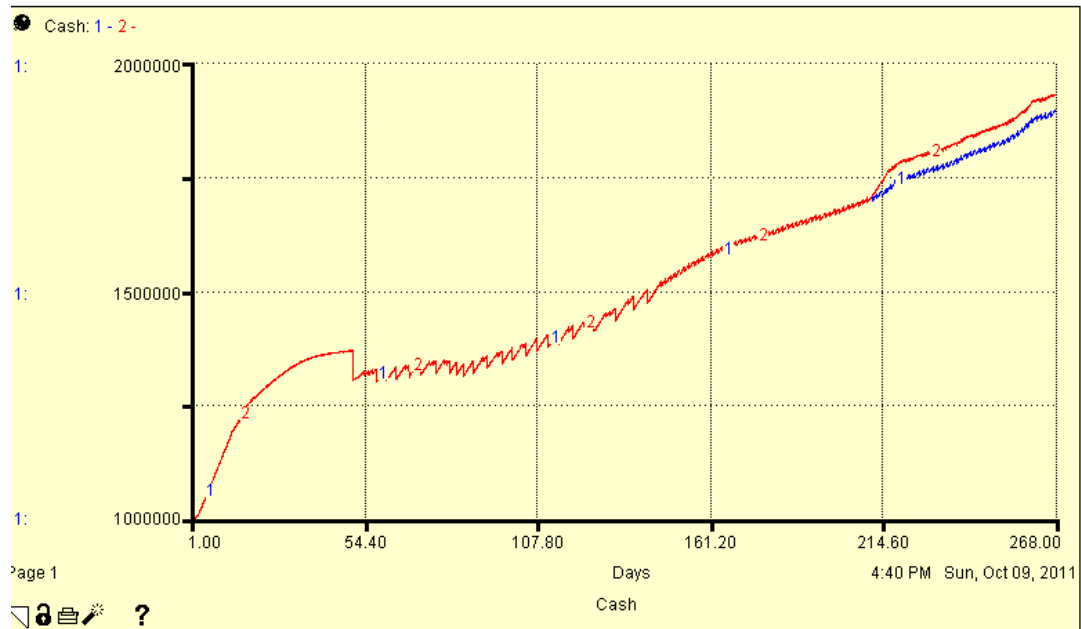


Figure 45 Choosing the optimum reorder quantity

### 5.3.3.3 Refining reorder level and reorder quantity for the last 50 days

The re-order level and re-order quantity chosen works well, but do not grant the maximum cash level in the end. Specifically, buying 2,500 kits during last 50 days is over-purchasing. So the value should be reduced for the last 50 days. The optimum values for the re-order level and re-order quantity for the last 50 days are 40 jobs and 500 jobs respectively. Figure 46 shows the graph for the refinement.



**Figure 46 Refining reorder level and reorder quantity for the last 50 days**

Both lines are generated when contract 3 is chosen and re-order level and re-order quantity is set to 100 jobs and 2,500 jobs for days 50 to 218. For the blue line, the values remained constant throughout the run. For the red line, the values changed to 40 for re-order level and 500 for re-order quantity respectively by day 218. The latter setting grants \$1.932 million of ending cash, which is \$36,165 higher than its alternative.

### 5.3.4 The Number of Machines

Besides contract type, lead time is a main determining factor for revenue per job. Figure 47 shows how the average lead time changes from day 50 to day 268.

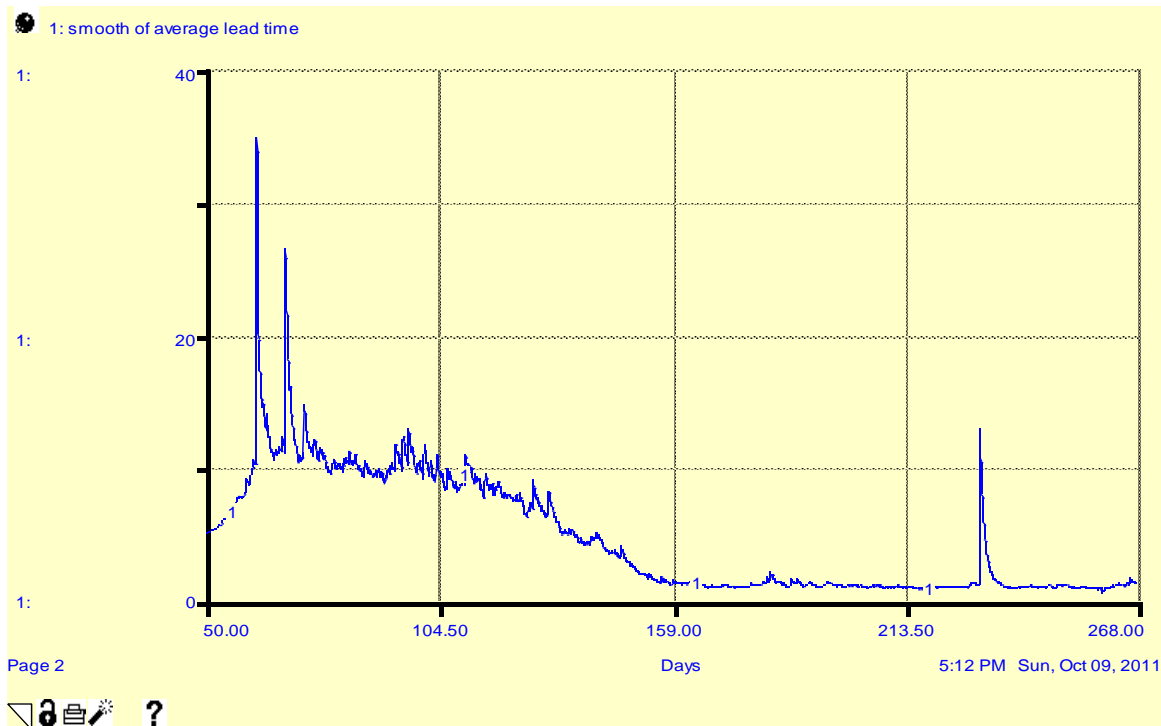


Figure 47- Average Lead Time before changing desired number of machines

Figure 47 shows that between days 50 and 150 the lead time is very high. Therefore, the revenue earned from completing orders is minimal. After day 150, the lead time decreases and the revenue earned from orders increases.

In order to reduce lead time, new machines are purchased. As discussed in section 5.1.3, the desired numbers of machines are 5-3-2 for stations 1, 2 and 3 respectively. Figure 48 shows the average lead time for the base run (blue line) and the result of implementing the desired number of machines (red line). Note that the lead time for the red line is less than for the blue line throughout the test-run.

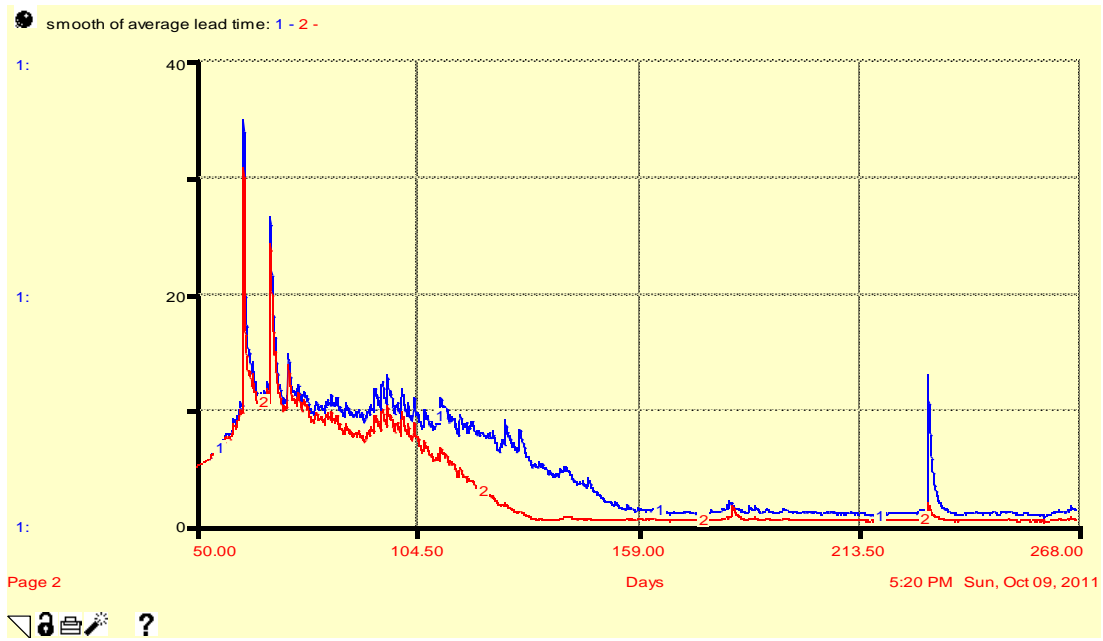


Figure 48 Lead time before and after changing the number of machines

The ending cash for the proposed strategy is \$3.511 million, which is \$1.579 million higher than in the case of the base strategy. The cash levels are shown in Figure 49.

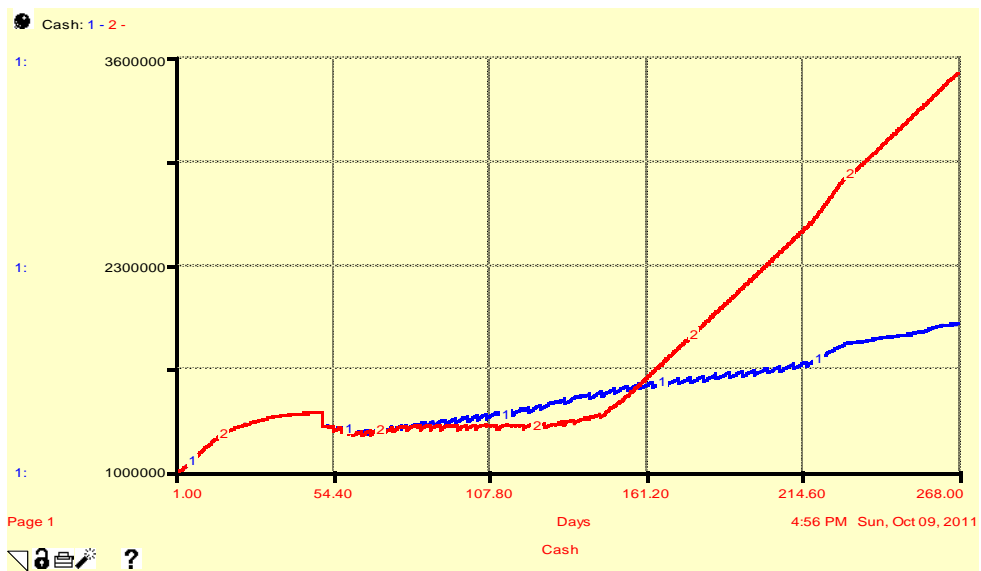


Figure 49 Cash Level before and after changing the desired number of machines

### 5.3.5 Verifying Optimal Strategy with the Random Input

In Littlefield Simulation, the number of incoming customer orders per day is random with a mean of 12 and a standard deviation of 3. When inputted with the more realistic data, the strategy proposed in previous section should grant the same result, since the system's intrinsic behavior should be independent of deviations in inputs. Figure 50 shows the cash level when the number of incoming customer orders per day is random with a mean of 12 and a standard deviation of 3.

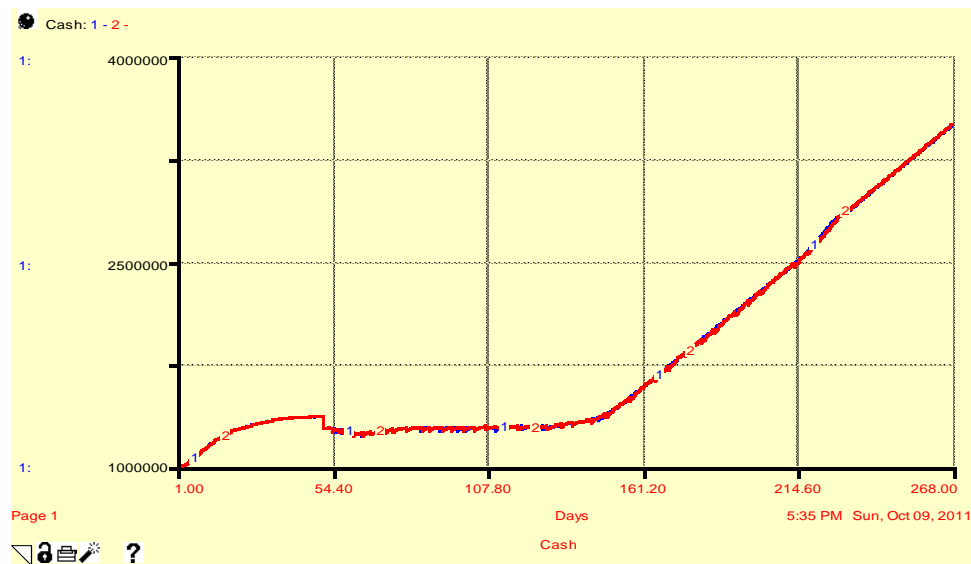


Figure 50 Cash Level when customer orders are random

In Figure 50, the blue line shows the cash level over time when customer orders are constant. The red line shows the cash level over time when the orders are random. The difference between the ending cash for the modes is only \$1,000, which is relatively small in comparison to the ending cash level.

## **6.0 Limitations and Future Work**

### **6.1 Limitations**

A significant limitation of the winning strategy devised is that it could not be tested with the Littlefield game due to time constraints. However, we are confident that the strategy would perform well because it consistently gives optimal cash in the iThink model. The model's robustness has been tested as well.

The model we constructed for this report is a continuous time system dynamics model. Using continuous time removes some of the distortions seen in the Littlefield discrete time model. For most cases, the removal of such distortions would be desirable. However, using the same policy devised on the continuous time system may not be as robust on the Littlefield discrete time system. For the most part this can be overlooked since modeling a process in continuous time corresponds better with reality.

One of the main shortcomings of the Littlefield Simulation is the lack of significant holding cost for raw materials, which deviates from reality. Furthermore, completed orders do not affect future orders in the game, since incoming customer orders are treated as an exogenous variable. While in reality, if a company performs well in terms of the lead time, it will certainly receive more orders as its relationship with customers strengthens and reputation grows.

### **6.2 Future Work**

In future work, our optimal strategy should be tested in the Littlefield Simulation game. The results can be used to amend the strategy and the model.

Future work should also include the development of a more dynamic policy where the re-order level, re-order quantity and the number of machines to be bought changes automatically

when the parameters of the game change. The future model should also encompass holding costs for raw materials in the inventory. This is also a consideration that Littlefield Technology can take into account.

Lastly, our system dynamics model can be used to simulate an industry with many competing firms.



## 7.0 Conclusion

The Littlefield game is a discrete time simulation. In order to design a winning strategy for Littlefield game, a continuous System Dynamics model was constructed to represent the discrete time simulation.

The initial model building phase of the project involved replicating the trends seen in the original game. In order to model these initial trends we used the same decision rules that were used in the Littlefield game, such as changing contracts based upon lead time. The decision rules used in the original game were reactionary rather than proactive. Such decision rules do not help in maximizing profit at the end of the game. Therefore, a more proactive strategy was needed to maximize profit.

The main finding of the project is that to maximize the cash flow the most profitable contract type should be chosen (contract 3). The player needs also to balance two strategic investments: investments in raw materials and machines. The reorder level and reorder quantity chosen was the economic reorder quantity and economic reorder point. The desired number of machines selected based upon the number of machines which would give the maximum utilization of machines.

Our ability to identify a winning strategy suggests that a system dynamics model can be used to identify and test strategies for the Littlefield simulation. Our system dynamics model can also be used in advanced system dynamics classes to teach elements of manufacturing.

## 8.0 Work Cited

Stanford Graduate School of Business website

[http://www.gsb.stanford.edu/news/research/mfg\\_kumar\\_factorygame.shtml](http://www.gsb.stanford.edu/news/research/mfg_kumar_factorygame.shtml)

Miyaoka, J. (2005), "Making Operations Management Fun: Littlefield Technologies," *INFORMS Transactions on Education*, Vol. 5, No 2, <http://ite.pubs.informs.org/Vol5No2/Miyaoka/>

“Yale SOM Team Wins International Operations Competition” News of Yale school of Management.

[http://mba.yale.edu/news\\_events/CMS/Articles/5952.shtml](http://mba.yale.edu/news_events/CMS/Articles/5952.shtml) retrieved 08.27.2011

Littlefield Simulation 2 Report by Ge Yin, Joshua Stephens, Michael Buchanan, Mingjing Chen and Tian Bai.

<http://www.scribd.com/doc/51139499/Littlefield-Simulation-2-Report> Retrieved 08.27.2011

Michael J. Radzicki and Robert A. Taylor (2008). "Origin of System Dynamics: Jay W. Forrester and the History of System Dynamics” In: *U.S. Department of Energy's Introduction to System Dynamics*

<http://www.systemdynamics.org/DL-IntroSysDyn/start.htm> Retrieved 08.27.2011

Forrester, Jay (1998) *Designing the Future*. Page 5.

[http://clexchange.org/ftp/documents/whyk12sd/Y\\_1999-03DesigningTheFuture.pdf](http://clexchange.org/ftp/documents/whyk12sd/Y_1999-03DesigningTheFuture.pdf) Retrieved 08.27.2010

Gross, Donald, John F.. Shortle, and James M. Thompson. *Fundamentals of queuing theory*. 4e éd. ed. Hoboken, NJ: J. Wiley & Sons, 2008. Print. Chapter 1, Page 3

Krajewski, Lee J., Larry P. Ritzman, and Manoj K. Malhotra. *Operations management: Processes and supply chains*. 9. ed. Upper Saddle River: Pearson Education, 2010. Print. Page 437

"Causal Loop Diagram." *Wikipedia, the Free Encyclopedia*. Web. 13 Oct. 2011. <[http://en.wikipedia.org/wiki/Causal\\_loop\\_diagram](http://en.wikipedia.org/wiki/Causal_loop_diagram)>.

"Safety Stock Calculations | Lean Sigma Supply Chain." *Resource Systems Consulting*. Web. 23 Oct. 2011. <<http://www.resourcesystemsconsulting.com/blog/archives/90>>.

"Re-order Level Calculation." *Managerial Accounting - Free Cost and Management Accounting Articles*. Web. 23 Oct. 2011. <[http://www.accountingformanagement.com/reordering\\_level.htm](http://www.accountingformanagement.com/reordering_level.htm)>.

## 9.0 Appendix

### 9.1 Equations

$$\text{Cash}(t) = \text{Cash}(t - dt) + (\text{Revenue\_per\_day} - \text{Expenses}) * dt$$

$$\text{INIT Cash} = 1000000$$

INFLOWS:

$$\text{Revenue\_per\_day} =$$

$$\text{Revenue\_from\_completing\_orders} + \text{Revenue\_from\_Interest} + \text{Total\_Machine\_Sales}$$

OUTFLOWS:

$$\text{Expenses} = \text{Expenses\_from\_Ordering\_Raw\_Material} + \text{Total\_Machine\_Expenses}$$

$$\text{Customer\_Orders\_in\_the\_system}(t) = \text{Customer\_Orders\_in\_the\_system}(t - dt) + (\text{Job\_release\_into\_the\_system} - \text{Completed\_orders}) * dt$$

$$\text{INIT Customer\_Orders\_in\_the\_system} = 0$$

INFLOWS:

$$\text{Job\_release\_into\_the\_system} = \text{Job\_Release}$$

OUTFLOWS:

$$\text{Completed\_orders} = \text{completion\_rate}$$

$$\text{Customer\_Order\_Queue}(t) = \text{Customer\_Order\_Queue}(t - dt) + (\text{Customer\_Orders\_Incoming} - \text{Customer\_Orders\_Processed}) * dt$$

$$\text{INIT Customer\_Order\_Queue} = 1$$

INFLOWS:

$$\text{Customer\_Orders\_Incoming} = 26$$

OUTFLOWS:

$$\text{Customer\_Orders\_Processed} = \text{Job\_Release}$$

$$\text{machine1}(t) = \text{machine1}(t - dt) + (\text{Change\_in\_Machine\_1}) * dt$$

$$\text{INIT machine1} = 3$$

INFLOWS:

$$\text{Change\_in\_Machine\_1} = \text{IF} (\text{TIME} < 51) \text{ OR} (\text{TIME} > 236) \text{ THEN} 0 \text{ ELSE IF} (\text{Difference\_in\_Machines} < 0) \text{ THEN} 0 \text{ ELSE}$$

$$\text{Effect\_of\_cash\_ratio\_on\_machine1} * \text{Difference\_in\_Machines} / \text{Adjustment\_Time}$$

machine2(t) = machine2(t - dt) + (Change\_in\_Machine\_2) \* dt

INIT machine2 = 1

INFLOWS:

Change\_in\_Machine\_2 = IF (TIME <51) OR (TIME >236) THEN 0 ELSE IF  
(Difference\_in\_Machine\_2 < 0) THEN 0 ELSE  
Effect\_of\_cash\_ratio\_on\_machine\_2 \* Difference\_in\_Machine\_2 / Adjustment\_Time

machine3(t) = machine3(t - dt) + (Change\_in\_Machine\_3) \* dt

INIT machine3 = 1

INFLOWS:

Change\_in\_Machine\_3 = IF (TIME <51) OR (TIME >236) THEN 0 ELSE IF  
(Difference\_in\_Machine\_3 < 0) THEN 0 ELSE  
Effect\_of\_cash\_ratio\_on\_machine\_3 \* Difference\_in\_Machine\_3 / Adjustment\_Time

Order\_for\_Raw\_Material(t) = Order\_for\_Raw\_Material(t - dt) + (raw\_material\_purchase\_rate -  
Raw\_Material\_Arriving\_Rate) \* dt

INIT Order\_for\_Raw\_Material = 0

INFLOWS:

raw\_material\_purchase\_rate =  
raw\_material\_reorder\_quantity \* Inventory\_Ratio's\_effect\_on\_Reordering \* Effect\_of\_cash\_\_on\_p  
urchases\_on\_raw\_materials

OUTFLOWS:

Raw\_Material\_Arriving\_Rate = Order\_for\_Raw\_Material / order\_delay

queue\_for\_step2(t) = queue\_for\_step2(t - dt) + (step1\_process\_rate - step2\_process\_rate) \* dt

INIT queue\_for\_step2 = 0

INFLOWS:

step1\_process\_rate = queue\_for\_step\_1 / Actual\_Time\_1

OUTFLOWS:

step2\_process\_rate = queue\_for\_step2 / actual\_time\_2

queue\_for\_step3(t) = queue\_for\_step3(t - dt) + (step2\_process\_rate - step3\_process\_rate) \* dt

INIT queue\_for\_step3 = 0

INFLOWS:

step2\_process\_rate = queue\_for\_step2 / actual\_time\_2

OUTFLOWS:

$$\text{step3\_process\_rate} = \text{queue\_for\_step3}/\text{actual\_time\_3}$$

$$\text{queue\_for\_step4}(t) = \text{queue\_for\_step4}(t - dt) + (\text{step3\_process\_rate} - \text{completion\_rate}) * dt$$

$$\text{INIT queue\_for\_step4} = 0$$

INFLOWS:

$$\text{step3\_process\_rate} = \text{queue\_for\_step3}/\text{actual\_time\_3}$$

OUTFLOWS:

$$\text{completion\_rate} = \text{queue\_for\_step4}/\text{actual\_time\_4}$$

$$\text{queue\_for\_step\_1}(t) = \text{queue\_for\_step\_1}(t - dt) + (\text{Job\_Release} - \text{step1\_process\_rate}) * dt$$

$$\text{INIT queue\_for\_step\_1} = 0$$

INFLOWS:

$$\text{Job\_Release} =$$

$$\text{Customer\_Order\_Queue} * \text{Total\_orders}'\_effect\_on\_Job\_Release * \text{Effect\_of\_raw\_material\_ratio\_o} \\ n\_job\_release / \text{Job\_Release\_Lead\_Time}$$

OUTFLOWS:

$$\text{step1\_process\_rate} = \text{queue\_for\_step\_1}/\text{Actual\_Time\_1}$$

$$\text{Raw\_Material\_Inventory}(t) = \text{Raw\_Material\_Inventory}(t - dt) + (\text{Raw\_Material\_Arriving\_Rate} - \\ \text{Job\_Release}) * dt$$

$$\text{INIT Raw\_Material\_Inventory} = 160$$

INFLOWS:

$$\text{Raw\_Material\_Arriving\_Rate} = \text{Order\_for\_Raw\_Material}/\text{order\_delay}$$

OUTFLOWS:

$$\text{Job\_Release} =$$

$$\text{Customer\_Order\_Queue} * \text{Total\_orders}'\_effect\_on\_Job\_Release * \text{Effect\_of\_raw\_material\_ratio\_o} \\ n\_job\_release / \text{Job\_Release\_Lead\_Time}$$

$$\text{Actual\_Time\_1} = \text{Normal\_Processing\_Time\_1} * \text{effect\_of\_capacity\_utilization\_1}$$

$$\text{actual\_time\_2} = \text{effect\_on\_cap\_ut\_2} * \text{NPT\_2}$$

$$\text{actual\_time\_3} = \text{NPT\_3} * \text{effect\_on\_cap\_ut\_3}$$

$$\text{actual\_time\_4} = \text{npt\_4} * \text{effect\_on\_cap\_ut\_4}$$

$$\text{Adjustment\_Time} = 1$$

Average\_Lead\_Time =  
 Job\_Release\_\_Lead\_Tlme+Actual\_Time\_1+actual\_time\_2+actual\_time\_3+actual\_time\_4  
 Average\_Service3\_Rate\_3 = 15.91  
 average\_service\_rate1 = 12  
 Average\_Service\_\_Rate\_2 = 30.6  
 Contract\_Types\_1 = Perceived\_lead\_time's\_effect\_on\_Contract\_Type1  
 Contract\_Type\_2 = Perceived\_Lead\_Time's\_\_effect\_on\_Contract\_Type2  
 Contract\_Type\_3 = Perceived\_Lead\_Time's\_effect\_on\_Contract\_Type3  
 cost\_of\_machine\_1 = 90000  
 cost\_of\_machine\_2 = 80000  
 cost\_of\_machine\_3 = 100000  
 cost\_per\_job = 600  
 Desired\_Machines = .33  
 Desired\_Machines\_2 = .33  
 desired\_machines\_3 = .33  
 Difference\_in\_Machines = Desired\_Machines-machine1  
 Difference\_in\_Machine\_2 = Desired\_Machines\_2-machine2  
 Difference\_in\_Machine\_3 = desired\_machines\_3-machine3  
 effect\_of\_capacity\_utilization\_1 = 1+queue\_for\_step\_1  
 effect\_on\_cap\_ut\_2 = 1+queue\_for\_step2  
 effect\_on\_cap\_ut\_3 = 1+queue\_for\_step3  
 effect\_on\_cap\_ut\_4 = 1+queue\_for\_step4  
 Expenses\_from\_Ordering\_\_Raw\_Material = IF (raw\_material\_reorder\_quantity= 0) THEN 0  
 ELSE raw\_material\_purchase\_rate\*(fixed\_cost/raw\_material\_reorder\_quantity+cost\_per\_job)  
 fixed\_cost = 1000  
 Initial\_Reorder\_Level = 24  
 Initial\_Reorder\_Quantity = 160  
 Interest\_rate = 0.00026  
 Intermediate\_Reorder\_Level = 24

Intermediate\_Reorder\_Quantity = 160

Inventory\_Ratio's\_effect\_on\_Reordering = IF(Inventory\_\_Ratio>1) THEN (0) ELSE (1)

Inventory\_\_Ratio = IF (Reorder\_\_Level=0) THEN 0 ELSE  
(Raw\_Material\_Inventory+Order\_for\_Raw\_Material)/Reorder\_\_Level

Job\_Processing\_Time = 1/48

Job\_Release\_\_Lead\_TTime = (Customer\_Order\_Queue+1) \*Job\_Processing\_Time

machine1\_capacity = NORMAL(average\_service\_rate1,std\_dev1, 5)

Machine1\_expense = IF(Change\_in\_Machine\_1>0) THEN  
SQRT((cost\_of\_machine\_1\*Change\_in\_Machine\_1)^2) ELSE 0

machine2\_capacity = NORMAL(Average\_Service\_\_Rate\_2,std\_dev2, 5)

machine3\_capacity = NORMAL(Average\_Service3\_Rate\_3,std\_dev3, 5)

Machine\_1\_Sales = IF(Change\_in\_Machine\_1<0) THEN  
SQRT((Price\_of\_machines\_sold\*Change\_in\_Machine\_1)^2) ELSE 0

Machine\_2\_Expense = IF(Change\_in\_Machine\_2>0) THEN  
SQRT((cost\_of\_machine\_2\*Change\_in\_Machine\_2)^2) ELSE 0

Machine\_2\_Sales = IF(Change\_in\_Machine\_2<0) THEN  
SQRT((Price\_of\_machines\_sold\*Change\_in\_Machine\_2)^2) ELSE 0

Machine\_3\_Expense = IF(Change\_in\_Machine\_3>0) THEN  
SQRT((cost\_of\_machine\_3\*Change\_in\_Machine\_3)^2) ELSE 0

Machine\_3\_Sales = IF(Change\_in\_Machine\_3<0) THEN  
SQRT((Price\_of\_machines\_sold\*Change\_in\_Machine\_3)^2) ELSE 0

maximum\_orders\_\_to\_orders\_ratio = Customer\_Orders\_in\_the\_system/Maximum\_order\_limit

Maximum\_order\_limit = 100

Normal\_Processing\_Time\_1 = 1/(machine1\*machine1\_capacity)

NPT\_2 = 1/(machine2\*machine2\_capacity\*priority\_for\_step\_2)

NPT\_3 = 1/(machine3\*machine3\_capacity)

npt\_4 = 1/(machine2\*machine2\_capacity\*(1-priority\_for\_step\_2))

order\_delay = 3

Perceived\_Lead\_Time = SMTH3(average\_lead\_time,3.5,1.5)



Perceived\_lead\_time's\_effect\_on\_Contract\_Type1 = IF (TIME< 50) THEN 1 ELSE IF (Perceived\_Lead\_Time>2) THEN 1 ELSE 0

Perceived\_Lead\_Time's\_effect\_on\_Contract\_Type3 = IF (TIME <50) THEN 0 ELSE IF(Perceived\_Lead\_Time<=.6) THEN 1 ELSE 0

Perceived\_Lead\_Time's\_\_effect\_on\_Contract\_Type2 = IF (TIME <50) THEN 0 ELSE IF(Perceived\_Lead\_Time>1) AND (Perceived\_Lead\_Time<=2) THEN 1 ELSE 0

Price\_of\_machines\_sold = 10000

priority\_for\_step\_2 = 0.5

ratio\_of\_machine1\_cost\_vs\_cash = Cash/(cost\_of\_machine\_1+Total\_cost\_of\_order)

ratio\_of\_machine2\_cost\_vs\_cash = Cash/(cost\_of\_machine\_2+Total\_cost\_of\_order)

ratio\_of\_machine3\_cost\_vs\_cash = Cash/(cost\_of\_machine\_3+Total\_cost\_of\_order)

Ratio\_of\_purchase\_vs\_cash = IF( Total\_cost\_of\_order =0) THEN 0 ELSE Cash/Total\_cost\_of\_order

raw\_material\_reorder\_quantity = IF (TIME <50) THEN Initial\_Reorder\_Quantity ELSE Intermediate\_Reorder\_Quantity

Raw\_Material\_to\_Customer\_Order\_Ratio = IF (Customer\_Order\_Queue=0) THEN 0 ELSE Raw\_Material\_Inventory/Customer\_Order\_Queue

Reorder\_Level = IF (TIME <50) THEN Initial\_Reorder\_Level ELSE Intermediate\_Reorder\_Level

Revenue\_from\_completing\_orders = completion\_rate\*Revenue\_per\_batch+Revenue\_from\_orders\*completion\_rate

Revenue\_from\_Interest = Cash\*Interest\_rate

Revenue\_from\_orders = IF (TIME < 50) THEN Contract\_Type\_1\_revenue\_per\_batch\_2 ELSE Contract\_Type\_3\_\_Revenue\_per\_Batch\_policy

Revenue\_per\_batch = Contract\_Types\_1\*Contract\_Type\_1\_revenue\_per\_batch+Contract\_Type\_2\*Contract\_Type\_2\_\_Revenue\_per\_Batch+Contract\_Type\_3\*Contract\_Type\_3\_\_Revenue\_per\_Batch

smooth\_of\_average\_lead\_time = SMTH1(Average\_Lead\_Time,1)

std\_dev1 = 4

std\_dev2 = 10

std\_dev3 = 5

Total\_cost\_of\_order = fixed\_cost+cost\_per\_job\*raw\_material\_reorder\_quantity

Total\_Machine\_Expenses = Machine1\_expense+Machine\_2\_Expense+Machine\_3\_Expense

Total\_Machine\_Sales = Machine\_1\_Sales+Machine\_2\_Sales+Machine\_3\_Sales

Contract\_Type\_1\_revenue\_per\_batch = GRAPH(average\_lead\_time)

(0.00, 750), (2.00, 750), (4.00, 0.00)

Contract\_Type\_1\_revenue\_per\_batch\_2 = GRAPH(smooth\_of\_average\_lead\_time)

(0.00, 750), (2.00, 750), (4.00, 0.00)

Contract\_Type\_2\_Revenue\_per\_Batch = GRAPH(average\_lead\_time)

(0.00, 1000), (1.00, 1000), (2.00, 0.00)

Contract\_Type\_3\_\_Revenue\_per\_Batch = GRAPH(average\_lead\_time)

(0.00, 1250), (0.5, 1250), (1.00, 0.00)

Contract\_Type\_3\_\_Revenue\_per\_Batch\_policy = GRAPH(smooth\_of\_average\_lead\_time)

(0.00, 1250), (0.5, 1250), (1.00, 0.00)

Effect\_of\_cash\_ratio\_on\_machine\_2 = GRAPH(ratio\_of\_machine2\_cost\_vs\_cash)

(0.00, 0.00), (0.2, 0.00), (0.4, 0.00), (0.6, 0.00), (0.8, 0.00), (1.00, 1.00), (1.20, 1.00), (1.40, 1.00), (1.60, 1.00), (1.80, 1.00), (2.00, 1.00)

Effect\_of\_cash\_ratio\_on\_machine\_3 = GRAPH(ratio\_of\_machine3\_cost\_vs\_cash)

(0.00, 0.00), (0.2, 0.00), (0.4, 0.00), (0.6, 0.00), (0.8, 0.00), (1.00, 1.00), (1.20, 1.00), (1.40, 1.00), (1.60, 1.00), (1.80, 1.00), (2.00, 1.00)

Effect\_of\_cash\_\_on\_purchases\_on\_raw\_materials = GRAPH(Ratio\_of\_purchase\_vs\_cash)

(0.00, 0.00), (0.2, 0.2), (0.4, 0.4), (0.6, 0.6), (0.8, 0.8), (1.00, 1.00), (1.20, 1.00), (1.40, 1.00), (1.60, 1.00), (1.80, 1.00), (2.00, 1.00)

Effect\_of\_raw\_material\_ratio\_on\_job\_release =  
GRAPH(Raw\_Material\_to\_Customer\_Order\_Ratio)

(0.00, 0.00), (0.2, 0.2), (0.4, 0.4), (0.6, 0.6), (0.8, 0.8), (1.00, 1.00), (1.20, 1.00), (1.40, 1.00), (1.60, 1.00), (1.80, 1.00), (2.00, 1.00)

Effect\_of\_\_cash\_ratio\_on\_machine1 = GRAPH(ratio\_of\_machine1\_cost\_vs\_cash)

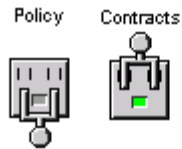
(0.00, 0.00), (0.2, 0.00), (0.4, 0.00), (0.6, 0.00), (0.8, 0.00), (1.00, 1.00), (1.20, 1.00), (1.40, 1.00), (1.60, 1.00), (1.80, 1.00), (2.00, 1.00)

Total\_orders'\_effect\_on\_Job\_Release = GRAPH(maximum\_orders\_\_to\_orders\_ratio)

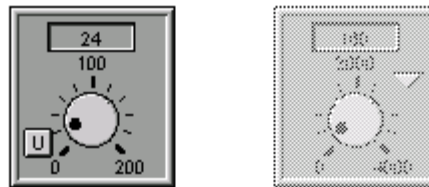
(0.00, 1.00), (0.1, 1.00), (0.2, 1.00), (0.3, 1.00), (0.4, 1.00), (0.5, 1.00), (0.6, 1.00), (0.7, 1.00), (0.8, 1.00), (0.9, 1.00), (1, 1.00), (1.10, 0.00), (1.20, 0.00), (1.30, 0.00), (1.40, 0.00), (1.50, 0.00), (1.60, 0.00), (1.70, 0.00), (1.80, 0.00), (1.90, 0.00), (2.00, 0.00)

## 9.2 User Interface

Note: Only one of the switches (policy and contract) has to be on. Both switches cannot be on at the same time.



Intermediate Reorder Level Intermediate Reorder Quantity



Desired Machines



Desired Machines 2



desired machines 3



priority for step 2



## 9.3 Littlefield Game Parameters



# Edit Scenario: default

### *Time and Administration*

No of simulated days per real day:

Random number generator seed: (0 means system clock)

Total number of simulated days:

Student access begins on day

Student access ends on day

After completing screened period, simulator should: run  suspend

### *Financial*

Initial cash per team: (no commas)

Interest rate per year:

Are students ever allowed to increase debt?  yes  no

If allowed, debt becomes available on day

If allowed, initial cost of debt as a fraction of debt

Debt rate per year:

(Interest is compounded daily)

### *Jobs and Orders*

Kits per order:

Initial lots per order:

Students enabled to change lots per order?  yes  no

Options for lots per order (separate with commas)

Students enabled to change maximum job inventory?  yes  no

Starting maximum jobs in factory:

### *Process Stations*

Initial count	Purchase Price (\$)	Retire Price (\$)	Daily Cost (\$)
<input type="text" value="3"/>	<input type="text" value="90000"/>	<input type="text" value="10000"/>	<input type="text" value="0.0"/>
<input type="text" value="1"/>	<input type="text" value="80000"/>	<input type="text" value="10000"/>	<input type="text" value="0.0"/>
<input type="text" value="1"/>	<input type="text" value="100000"/>	<input type="text" value="10000"/>	<input type="text" value="0.0"/>

1	<input type="text" value="1.4"/>	<input type="text" value="0.065"/>	<input checked="" type="radio"/>	<input type="radio"/>
2	<input type="text" value="0.2"/>	<input type="text" value="0.0050"/>	<input checked="" type="radio"/>	<input type="radio"/>
3	<input type="text" value="0.0"/>	<input type="text" value="0.03"/>	<input type="radio"/>	<input checked="" type="radio"/>
2	<input type="text" value="0.2"/>	<input type="text" value="0.02"/>	<input checked="" type="radio"/>	<input type="radio"/>

*Materials*

Students enabled to change inventory parameters?  yes  no

Starting number of kits:

Reorder point:

Reorder quantity:

Incremental cost per kit:

Fixed cost per order:

Lead time per order (days):

*Contracts*

Students enabled to change contracts?  yes  no

(Factory will start with the first contract)

**Quoted Lead Time (days)    Max Lead Time (days)    Max Order Revenue**

<input type="text" value="7"/>	<input type="text" value="14"/>	<input type="text" value="750"/>
<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="1000"/>
<input type="text" value="0.5"/>	<input type="text" value="1"/>	<input type="text" value="1250"/>

*Maximum of 3 contracts allowed*

*Demand Schedule*

Inter-arrival time distribution:  deterministic  exponential  uniform

All jobs per day values must be greater than zero:

**Day    Jobs per Day**

0