LOGISTICS CIM MODELING & SIMULATION

by

Qingxue Li
Reg No: 200007681

submitted in part fulfilment of the requirements for the degree of Master of Technology in the Department of Mechanical & Industrial Engineering in the Faculty of Engineering at Durban Institute of Technology

This dissertation represents my own work

APPROVED FOR FINAL SUBMISSION

16 - 01 - 03

Supervisor & Co-supervisor | Date

Durban 8th of November, 2002

DURBAN INSTITUTE OF TECHNOLOGY

SCHOOL OF POSTGRADUATE STUDIES

ANNEXURE E1

DECLARATION BY THE CANDIDATE IN RESPECT OF THE CONTENTS OF THE DISSERTATION/THESIS

The Administration Officer
School of Postgraduate Studies
Durban Institute of Technology

Dear Sir/Madam

I, ...QINGXUE LI..
(Full Name of Student)

Reg. No.: ....200007681..

hereby declare that the dissertation/thesis entitled

LOGISTICS CIM MODELING & SIMULATION

....

....

....

submitted for a ...Master... Degree in the Department of ....Mechanical..... ...., Faculty of ....Engineering....... is the result of my own investigation and research and that it has not been submitted in part or in full for any other degree or to any other Tertiary Institution.

16-01-2003

Signature of Student **Date**

ABSTRACT

This study presents a development of a method for Logistic Computer Integrated Manufacturing (LCIM) modelling. Based on this new model, a small LCIM cell simulation system was developed.

System integration is an important criterion for computer-integrated manufacturing (CIM). CIM modeling requires integration of systems to more accurate results. It can make CIM more flexible, which can meet the requirement for Logistics Manufacturing (LM) & Agile Manufacturing (AM). LCIM is proposing to meet this requirement starting with the basic CIM cell.

This paper proposes a method for LCIM modelling, and introduces the design criteria and its development. Using global design methods an integrated LCIM model, which includes Physical module, Process module and Functional module is presented. Through the CIM integrating case study, it is proved that based on these three models a system view could be provided.

The key approach for LCIM is to develop the CIM cell at system level, which is accessible to external sources whilst still providing support to internal functions. Main system design scheme, key technologies and development processes for the integrated system is also investigated.

Based on the new platform, both LCIM Simulation model and it's a interface is implemented. The result gives some indications of effective operations for a factory's CIM cell. It provides an integrated method, which can be a tool to improve the efficiency of production operations.

ACKNOWLEDGEMENTS

I express my sincere gratitude to my promoter, Professor G. Bright, Mr. J. Paramanund, and Mr. S. Pillay, who support the topic and gave me ideas for commencement of this project. This served as a basis for full development of the method presented in the study. They were a permanent source of support and encouragement during my research work. I am especially grateful for their patience with me – Chinese national, a foreigner.

I also wish to thank a friend and a teacher – Dr. Daohang Sha who was always there when I encountered problems. I also appreciate his sharing of knowledge with me. Miss Wei Wang also is supporting me all the time.

I would also like to recognize what my former classmates Xingjun Li and Weiqiao Wang. They help me to obtain the necessary materials and provided support to me. I would like to thank the staff of the department who always gave me help, especially Miss Shireen Singh, Mr. Chris Govinder, Mr. Ranil Singh, Mr. Wong Wingfong and Mr. Kuben Pather who have spent a lot of time to help me to correct the text.

Due to the English problem, this dissertation uses illustrations to explain the concepts. I would like to thank examiners for their kind considerations.

Finally, I am very grateful to the Research Department of M L Sultan Technikon / Durban Institute of Technology who funded this project and provided guidance.

Table of Contents

Table of Figures

List of Tables

1. Introduction

Modern manufacturing is now facing more frequent changes, such as new technologies, innovations etc. It demands that the relevant manufacturing system have to be flexible so that can quickly respond to these changes [20]. In the meantime, it also needs to cooperate with the changing and updating of other fields, such as control and management. Logistics management is one of the main elements, which influences production greatly. Manufacturing systems such as Kanban, Just In Time (JIT), Material Requirements Planning (MRP), and Distribution Resource Planning (DRP) require logistics and manufacturing activities of a firm to work closely together. Without a cooperative effort, the full advantages of systems can never be realized. It is therefore the challenge for current researchers to effectively adapt the manufacturing system to the logistic requirements.

The emergence of Computer Integrated Manufacturing (CIM) has resulted in an increased interest and appreciation for system design, real-time planning, scheduling and control [19]. CIM has included many aspects from these fields and it has already become one basic cell for manufacturing systems. To achieve CIM, all aspects of the manufacturing establishment must be integrated so that they can share information, communicate with each other, and provide a holistic picture as to the state of the entire manufacturing facility at any time. Such cooperation characteristic is a good starting point to analyze the logistics impact for manufacturing on the production side.

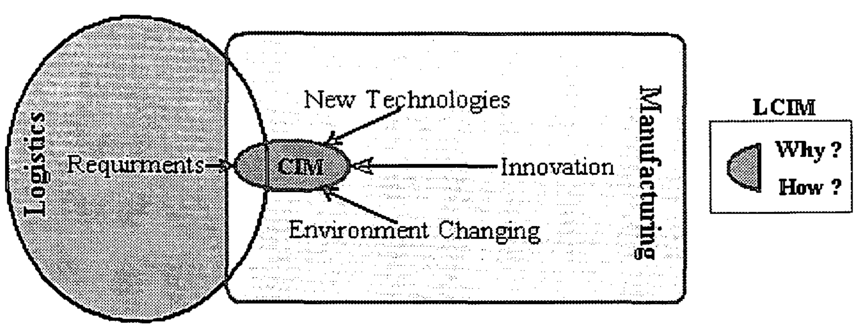


FIGURE 1-1: RELATED FIELDS

In this project a Logistic CIM (LCIM) model is proposed. The goal of the integrated model is to provide a systematic method for integrating and interfacing with the logistical requirements. LCIM tries to answer the following aspects:

Table 1-1: Research Cover Points

<table>
<tr><td rowspan="2">Why is LCIM needed?</td><td colspan="2">Problem Analysis</td></tr>
<tr><td colspan="2">Literature Review (Related Theories)</td></tr>
<tr><td rowspan="4">Development LCIM</td><td rowspan="2">CIM Integrating Platform</td><td>Investigation</td></tr>
<tr><td>Case Illustration</td></tr>
<tr><td rowspan="2">CIM's Logistic Support</td><td>Investigation</td></tr>
<tr><td>Case Illustration</td></tr>
<tr><td rowspan="4">Implementation of LCIM</td><td>Simulation Model</td><td>New Technologies</td></tr>
<tr><td>Graphic User Interface (GUI) Design</td><td>Testing</td></tr>
<tr><td rowspan="2">Real Time Implementation</td><td>Physical Execution</td></tr>
<tr><td>Database</td></tr>
<tr><td>Research Evaluation</td><td colspan="2">Discussion</td></tr>
</table>

This is a cross-field research, which is different with traditional manufacturing system design. Therefore this research will combine the system control, management knowledge and manufacturing system knowledge. Also this project proposes a three-function module's model. By using Artificial Intelligent (AI), it links these functions, and initializes a logistics-supporting scheme for the CIM system. The LCIM concept is illustrated through the related case analysis.

2. Problem Definition

2.1 Introduction

This is a working experience related project that is arisen from Singapore working experience from 1996 to 2000. During this period, Singapore's industry had been suffering economic crisis and had to respond to logistics requirements (Generally, it can be classified into Macro and Micro levels. Macro is more concern about the supply chain co-operation and the Micro level is more talk about inventory and lead-time control). This change in thinking has resulted in Singapore developing world class infrastructure such as first class airport and harbor facilities. Industry and commercial business interests have also capitalized from this new way of thinking that resulted in improved economic revival. How the small and medium enterprises could survive in this stiff environment? There was no clear answer for them, but they did spend more time on planning, scheduling, and production arrangement. They invest more money on training, adopt new technologies, and worked more closely between management and production line. In fact, what they have done is in line with logistics requirement. But the problem is that they did not get a clear guidance in mind.

All the problems raised due to logistics requirement is beyond the scope of this research dissertation. Therefore this dissertation will focus on the following aspects:

- Integrated system structure design;
- How structuring for logistics influencing; and
- New technologies, especially artificial intelligent integrating.

With consideration of the above aspects a simple CIM cell is discussed in context of logistics requirements. Consequently, building up a suitable CIM model is the basic requirement for such investigation.

The CIM cell in the Advanced Manufacturing Technologies Laboratory (AMT) at the ML SULTAN Technikon in Durban, South Africa (now known as the Durban Institute of Technology, DIT) has been used for the design for the LCIM model.

2.2 Background Analysis

Manufacturing has been facing increased pressure due to economic changes, new technologies and innovation, globalisation etc. To response to these changes, there are emerging of the new manufacturing concepts such as AM, Autonomous & Distributed Manufacturing System and Biological Manufacturing System that are trying to provide a framework for integrating new trends [4,17]. For example, AM is a concept that requires spiral development, and provides a framework for integrating People, Technology, and Organisation.

On the other hand, manufacturing has to solve the updating problems. It requires re-implementations for old concepts. For example, investigations have shown that systems such as Kanban, JIT, MRP, and DRP require logistics and manufacturing activities of a firm working together closely. Without a cooperative effort, the full advantages of systems can never be realized. Therefore Manufacturing Logistics has been implemented to deal with this field, which has shown that integrating logistics can reduce replenishment lead times to increase manufacturing flexibility and reduce order fulfillment lead times [36].

CIM is defined as the use of computer systems and automation systems to operate and control production. Now CIM cells have been used widely in the manufacturing environment, but the question is how such a basic manufacturing cell can be shaped for these changes as faced by most enterprises.

2.3 Problem Formulation

In general, the problems that are faced by CIM system are:

a. The need for frequent restructuring;
b. The need for more regular operations decisions;
c. The need for faster new-product-to-market cycles; and
d. The need for more Human- System communication [4].

All of these problems have shaped CIM into two kinds of activities. One is the current problem solving. For example, CIM flexibility is dependent upon software, but most factory engineers have encountered difficulties implementing new production functions on existing software systems so they have to restructure the system [33]. The other is the new requirement that occur from new technologies and integration demands, for example, the AM and ML's trend influence [19].

This research more concentrates on new technology integration than problem solving. LCIM concept tries to solve the logistics integration problem for CIM.

2.3.1 Logistic CIM

The start point for LCIM is LCIM modeling. To achieve CIM, all aspects of the manufacturing establishment must be integrated so that they can share information, communicate with one another and provide a holistic picture as to the state of the entire system for the manufacturing facility at any time. This is in line with the logistic requirements, but their focuses are different. To shape CIM for logistics requirement, it requires three steps:

1. Identify the activities and functions in the CIM cell;
2. Find the method for integrating logistics; and
3. Evaluation and feedback for spiral development.

2.3.2 Research sub-problems

There is a need for methods and tools to support an interdisciplinary approach, with special emphasis on organisational simulation, appropriate and selective use of technology, increased user involvement and rapid prototyping.

- There exists a need to address research in a system as a whole, not just research into new manufacturing technology.
- More emphasis on developing decision support systems that expand the range of possible decision alternatives.
- Technologies to support the learning companies or organisations.

To make this research valuable, all the factors mentioned above must be taken into account.

2.2.3 Limitation

Currently, in the AMT laboratory, there is just one DENFORD milling center, one EMCO PC TURN 120 CNC lathe, and one GRYPHON robot.

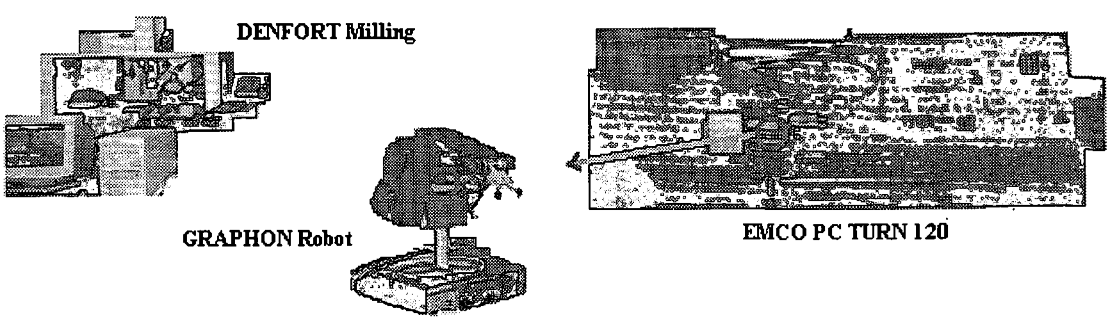


FIGURE 2-1: CURRENT LAB'S FACILITIES

The current physic connection is:

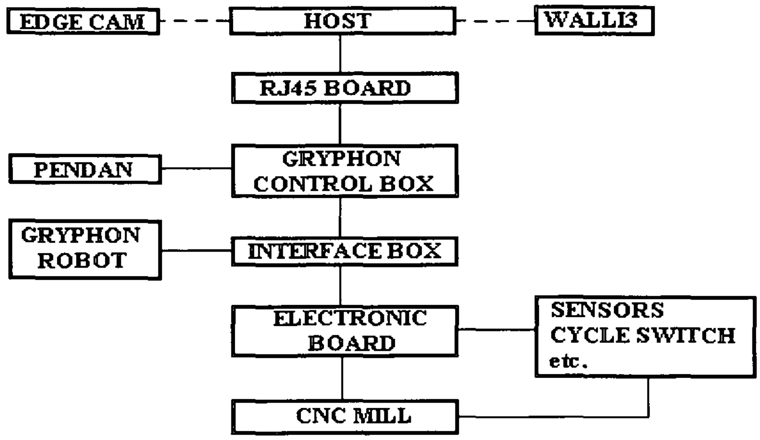


FIGURE 2-2: CURRENT CONNECTION

All the discussions will be based on the available of facilities, equipments and software at present in the Laboratory.

2.4 Research Methodology

This research is using Modelling and Simulation techniques. Modelling is used to consolidate the investigation with innovation. Based on the model, it tries to develop a simulation model that can be used for various analyses. This provides an advantage against real systems development, and the use of simulation also has appeared favourable as compared to purely analytical methods that often fail to capture complex interactions of a particular manufacturing system [2]. The advantages of using such techniques are:

- It can provide general insight into the nature of a process;
- Identify specific problems or problem areas within a system;
- Develop specific policies or plans for a process;
- Test new concepts and/or systems prior to implementation;
- Improve the effectiveness of a system; and
- Provide an "insurance policy" for system performance.

Artificial Intelligent has been used successfully in manufacturing system. It not only helps for management decision-making, but also applies production control very well. Such cross-field characteristic makes it a perfect tool for this project. This research therefore requires an in-depth knowledge of AI and its proper application capabilities.

For cross-field research, it demands lots of investigations. All the work must be based on the related fields' investigations thus the new integration opportunities can be explored.

Artificial Intelligent, Modeling and Simulation can provide the basic tools for this research. It uses manufacturing system design and analysis method to investigate ML, AM and CIM system. Based on these, investigations exploration into the LCIM concept now can be done. The development map is as follows:

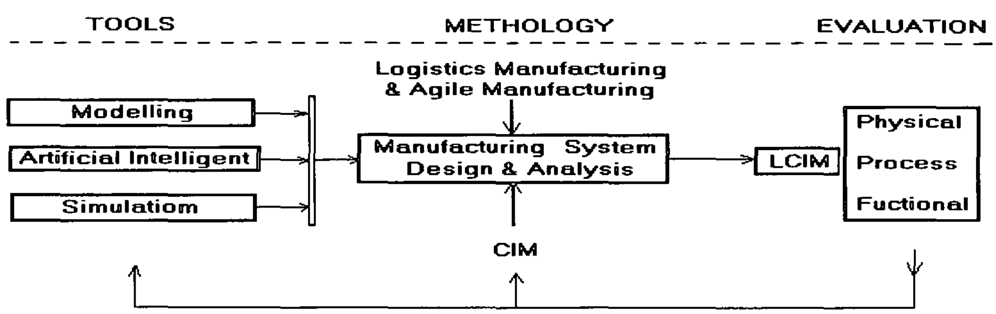


FIGURE 2-3: DEVELOPMENT METHODOLOGY

From CIM to LCIM requires investigation from the former and current activities. It needs suitable modeling methods to design the framework. Generally, it requires the effort as listed as below.

Table 2-1: List of Activity

CIM	Modeling	Simulation
Structuring	*	*
Logistics	*	*
Artificial Intelligent		*

3. Literature Review

CIM systems are being developed in response to the business trends toward shortened product life cycles and the need for reductions in the design-to-production transition period. CIM can be defined as the use of computer systems and automation systems to operate and control production. According to this definition, all aspects of the manufacturing establishment must be integrated.

To achieve this goal, high level co-operate skills and sophisticated management ability is needed in practice. The way to achieve these requirements is by using modelling and simulation techniques to test run all possible situations. These can produce shorter response times that are not allowed for experimentation and iteration with the real implementation. This makes the system working at its optimum from the start (all decisions will be made on the basis of modeling and simulation, rather than build-and-test methods) [2].

Many methods are available for manufacturing systems modeling [9, 16]. Mathematical modeling gives a good foundation for manufacturing systems modeling. Modeling with differential equations and finite-state approach is also used for manufacturing systems modeling, and along with queuing theories developing, it gives potential for more accuracy and practicable models for manufacturing modeling [8, 23]. Some researchers also use Petri net and IDEF as their tools for system modeling [10, 22].

System modeling takes advantage over real testing as one can save on time and also on cost to run the simulation on computer before the real test is done. Recent report applications of simulation for real-time operational control include adaptive scheduling and planning, performance forecasting and real-time displays of system status [15, 24, 28, 29]. The use of simulation has appeared favorable to purely analytical methods that often fail to capture complex interactions of a particular manufacturing system [2].

For LCIM modeling, it is important to treat LCIM as a whole conceptual item to suit new technologies demands and management requirements. This makes systems integrating a key point for LCIM. New technologies applied in manufacturing, enhances the approach to manufacturing systems such as Fuzzy Logic, Expert Systems and Neutral Networks [13, 28]. These techniques not only make manufacturing system flexible, but also contribute for modeling and simulation.

3.1 Manufacturing Systems Design and Analysis

There are many literatures that discuss systems design and analysis [12, 16]. In general, manufacturing system must be kept as simple as possible. The main attribute of systems theory in the area of manufacturing systems design and analysis is its simplicity. This is achieved by dividing the system as a whole into sub-systems:

(i) by clearly identifying the input, process and output elements of each sub-system;

(ii) by determining the appropriate controls for monitoring achievement against predetermined standards; and

(iii) by formulating and initiating the necessary corrective action.

A manufacturing system usually employs a series of value-adding manufacturing processes to convert the raw materials into more useful forms and eventually into finished products. It can usually be represented as system Input-Output (I/O) models [6]. Raw materials and labor can be treated as the system inputs, through certain mechanism and control processes, the system can provided the required products, which is the system output. It can be represented by the diagram below:

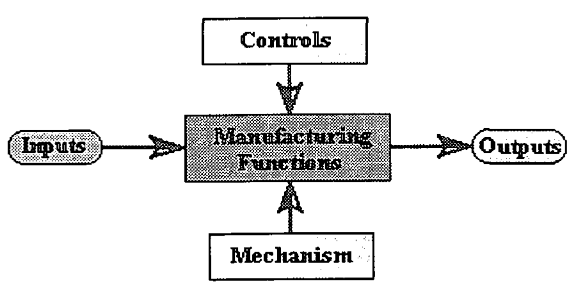


FIGURE 3-1: IDEFO MODEL

In contrast to the system I/O model, differential or difference equation, transfer function and frequency response commonly used in classical control, which can capture system dynamic numerical relations between system inputs, outputs and state variables in the time domain. These can be included in the state space representation, and make state space representation plays a key role in systems design and analysis with modern control technologies. (Interested reader can refer to **Modern Control Theory** for more details)

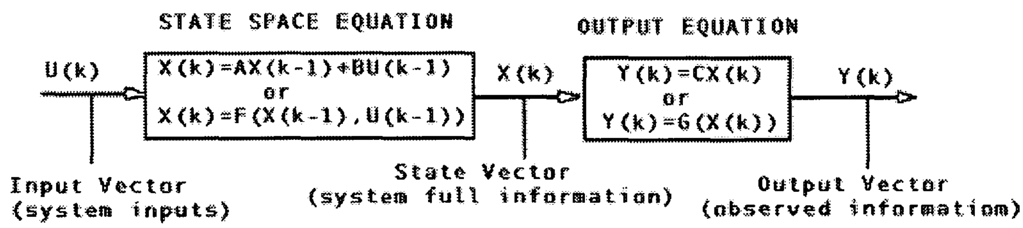


FIGURE 3-2: SYSTEM STATE SPACE REPRESENTATION

It is important to follow some methodologies when developing a manufacturing system. There does not seem to be a single approach for methodology. An important observation about design methodologies is that the situation should always be approached from the viewpoint of a responsible manager who is actually involved in the operation of the system concerned, rather than from that of a technical specialist. The design process can be interpreted as problem solving process. It can be illustrated as follow:

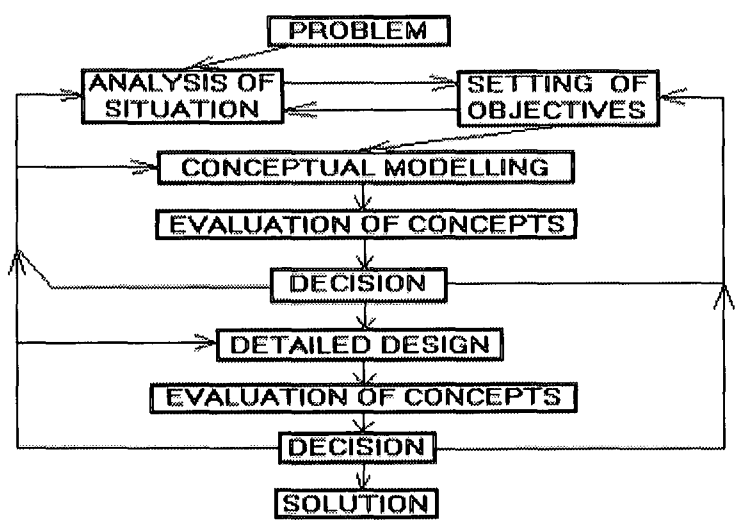


FIGURE 3-3: DESIGN STRUCTURE

3.2 Computer Integrated Manufacturing (CIM)

CIM systems are being developed in response to the business trends, towards shortened product life cycles and the need for reductions in the design-to-production transition period. CIM is defined as the use of computer and automation systems to operate and control production. Generally, CIM includes two separate activities: the information processing performed by computer systems and the physical activities performed by automation systems.

From an entity flow perspective, CIM can be viewed as a computer based control on the flow of decision, information, products and resources in any manufacturing system. Some of these flows may be physical flows while the others may be logical flows from one state to another. For instance in a flexible assembly system, the jobs may be assemblies, subassemblies, components, etc. which are physically flowing entities, the resources may be robots, automated stations, transports, etc., the information may be the status of jobs and resources. While the robots and transports resource flow physically, the stations are stationary and are viewed as flowing between busy and idle states. Each entity flow requires a certain time. The design and control of a system determines the nature of this flow and hence the performance of the system.

In this project, CIM system is a production process. Management and intelligent decision merges with a basic product cell. The CIM cell core is information analysis and integration. Build up CIM cells need many fields of co-operate integration. Intelligent management, intelligent control, and intelligent information analysis are the key technologies for CIM cells.

3.2.1 CIM Hardware

CIM includes various physical activities and information processes. It needs different supporting components. The different level components for CIM are presented as the picture shown below.

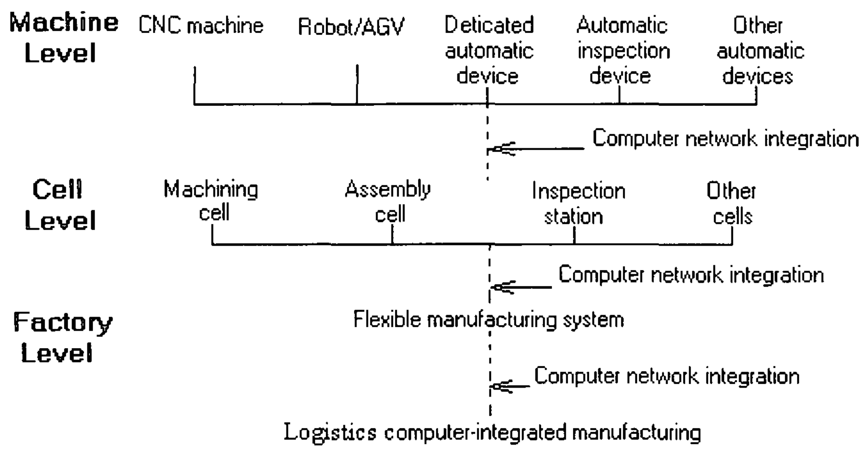


FIGURE 3-4: COMPONENTS OF CIM

For a CIM cell, the physical activities are performed by a wide range of devices, which includes machine tools, assembly stations, robotics, materials transfer systems, automated material handling, storage systems and inspection systems for quality control. CIM counts on the hardware to alter and move material, take measurements and provide feedback information to the human operators.

3.2.2 CIM Software

CIM software here refers to the support packages for information processing. CIM software can be daftly divided by functions into Host level software and Cell level software. The detailed functions can be summarized as follow:

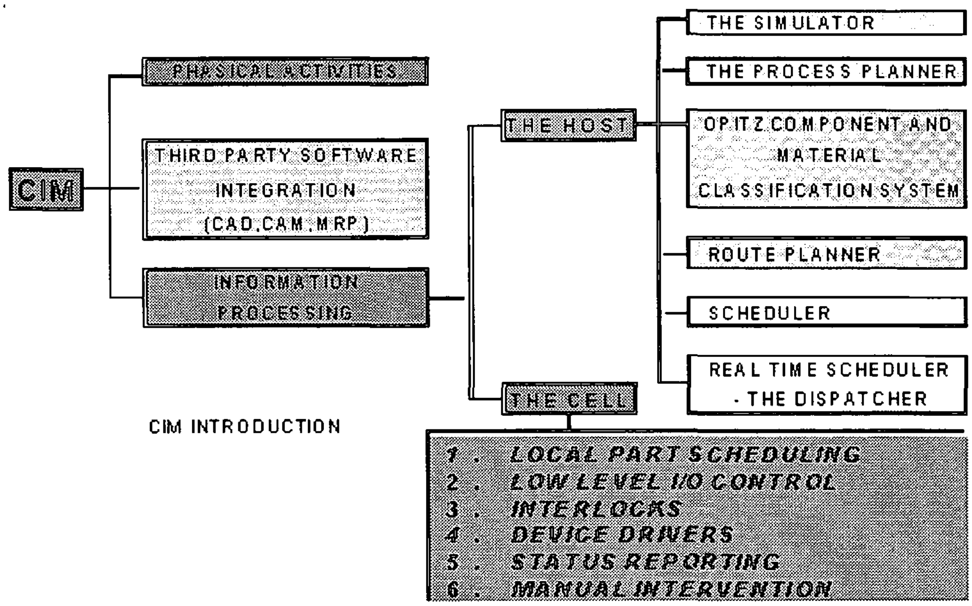


FIGURE 3-5: CIM FUNCTION SUMMARY

Host level information processing tasks include: the design of components; planning the production of the components [25], controlling the operations in production [22, 26] and performing various business related functions necessary for a manufacturing establishment, especially for planning and scheduling [10]. The Cell level software normally handles the low level signal processing, such as I/O controlling, status reporting etc.

3.3 Manufacturing Logistics & Agile Manufacturing

There is a growing need to adopt a broader approach to technology development and deployment. Development projects should be expected to employ a holistic approach, taking into account all the factors necessary to bring about change. These include skill and knowledge enhancement issues, technological requirements, management and organization, training and motivation of staff and technology deployment, the whole system supporting and cooperation. Those are the content for ML & AM.

3.3.1 Manufacturing Logistics (ML)

ML refers to all planning, co-ordination, and support functions, which are required to carry out manufacturing activities. It starts at the point where end-item customer demands are determined, and the point they are fulfilled. A narrow and a broad view of manufacturing logistics were identified. The narrow view included the planning, scheduling and control of all activities resulting in the processing, movement and storage of inventory. The broad view included integration across multiple of manufacturing facilities, integration between manufacturing and other corporate functional areas such as sales, marketing and engineering design, and integration with logistical functions such as transportation, warehousing and distribution [35]. The following figure gives the possible dimensions of ML.

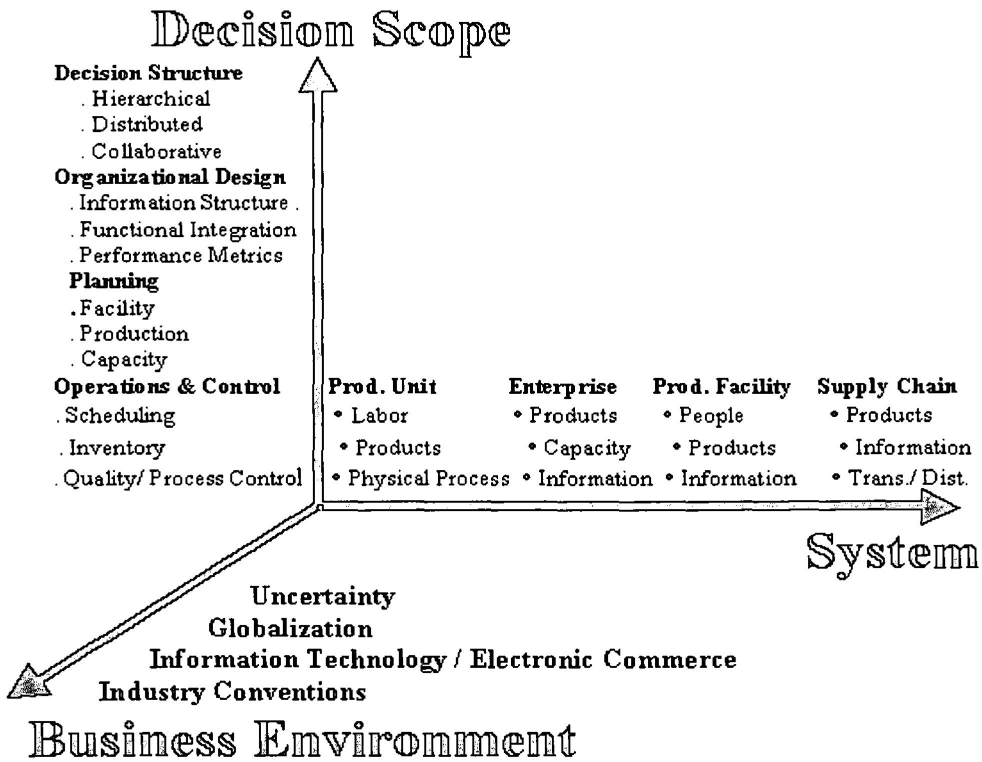


FIGURE 3-6: DIMENSION OF MANUFACTURING LOGISTICS

The major problem classes of ML were identified as the design, planning and control of production, systems, operations, information and people [34]. The ML activities are trying to provide tools for various integrating with Decision making system integration and environment responding.

ML requires that the logistics and manufacturing activities work together closely. The key to successful implementation of ML is an effective integrated of the information system. It requires that:

- Logistics must reduce replenishment lead times to increase manufacturing flexibility and reduce order fulfilment lead times.
- Manufacturing and logistics must work together in the production scheduling area to reduce production planning cycle time. Logistics can provide input into production scheduling and system requirements.
- Manufacturing and Logistics strategies, such as shortening of lead times, set-up times, and production run size must be used to minimise average inventory levels and stock outs.
- Logistics must develop strategies to reduce supplier lead times for parts and supplies.
- Logistics must adopt the philosophy that slow movers (i.e. products with low inventory turnover ratios) should be produced only after orders are received, rather than held in stock.

3.3.2 Agile Manufacturing (AM)

Agile Manufacturing is a business concept. Its aim is to put the enterprises way out in front of its major competitors. In detail, it needs to combine the organisation, people, and technologies into an integrated and co-ordinated whole. The agility that arises from this can be used for competitive advantage, by being able to respond rapidly for the changing market environment, and can use the ability to use and exploit fundamental resource-knowledge.

Fundamental to the exploitation of this resource is the idea of using technologies to lever people's skills and knowledge. People must be brought together, in dynamic teams formed around clearly identified market opportunities. By doing so, it becomes possible to lever one another's knowledge. These processes also can contribute to the transformation of knowledge and ideas into new products and services, as well as improvements to existing products and services.

The key to agility however, lies in several areas (organisation, people, and technology). An agile enterprise needs highly skilled and knowledgeable people who are flexible, motivated and responsive to change. It also needs new forms of organisational structures that empowers non-hierarchical management styles and stimulate and support individuals, as well as co-operation and teamwork. AM enterprises also need advanced computer based technologies to gain competitive advantages. (Interested reader is refer to **Agile Manufacturing**)

3.4 Modelling

A model may represent a system (e.g. a manufacturing system), an object (e.g., designed artefact), or a problem (e.g., designing a shaft) and is typically constructed for the purpose of analysis. Models are needed to describe existing systems, as well as to evaluate the feasibility and anticipated performance of proposed systems. Although models must capture enough detail to facilitate reliable experimentation, the purpose of modelling must not be violated by including unnecessary information that results in investment exceeding the cost of building and/ or experimenting on the actual system. A thorough understanding of functions, data, resources, and the organisational structure is essential in modelling processes. A model of the system can provide this understanding without disturbing the actual environment. An executable version of the model can simulate and even control the actual process [1, 6].

For manufacturing systems, mathematical models, conceptual models and processing models are widely used as the modeling tools.

3.4.1 Mathematical Models

Mathematical models are patiently constructed using a well-tried process and can be based either on data (on crop yields and fertilizer in this case) or on assumptions (in this case about how crop yield responds to fertilizer treatment) or usually a combination of both. Mathematical modeling can be defined as the activity of translating a real problem into mathematics for subsequent analysis. A mathematical model will be created and its solution will usually provide information, which is useful in dealing with the original real problem. On the other hand, manufacturing systems are typically discrete systems. There are different approaches to modeling such system for different purposes, an indicated below.

3.4.1.1 Discrete Models

In most of examples it can assumed that the variable was modelled by a particular function of time, $x = f(t)$, and then we can find the rate of change, $x' = f'(t)$. Very often modelling assumptions are expressed in terms of rate of change. For discrete models the essential ingredient is an equation of the form.

Next value = Function of {present value and previous values and possibly time}

Or in terms of a difference equation,

$$X_{n+1} = f(X_n, X_{n-1}, \ldots, t) \qquad (2\text{-}1)$$

Note that before writing it down, a considerable amount of thinking and choice of modelling assumptions are normally necessary. Solving a difference equation means finding an explicit expression for X_n in terms of n and initial values such as X_0. Note, however, that this can be a rather difficult task and not strictly essential, because a model in the form of a difference equation can be used without knowing the mathematical solution of the equation. This is because if the present and previous value of X are know, it can always use the difference equation to generate the next value, followed by as many values as is needed. The advantage of having a formula for X_n in terms of n is that it can substitute any value of n into the formula to get an immediate answer.

3.4.1.2 Finite-State Machines with Output Model

Many kinds of machines can be modelled using the method of finite-state machines. All these versions of finite-state machines include a finite set of states, with a designated starting state, an input alphabet, and a transition function that assigns the next state to every state and input pair.

It can describe how the machine works by specifying its state, how it changes state when input is received, and the output that is produced for every combination of input and current state. A finite-state machine $M = (S, I, O, f, g, s_0)$ consists of a finite set S of states; a finite input alphabet I; a finite output alphabet O; a transition function f that assigns to each state and input pair a new state; an output function g that assigns to each state and input pair an output; and an initial state s_0. We can use a state table to represent the values of the transition function f and the output function g for all pairs of states and inputs. Another way to show the actions of a machine is to use a directed graph with labelled edges, where a circle represents each state with the edges representing the transitions. Edges are labelled with the input and the output for that transition. (For more details, refer to **Mathematical Modelling Skills**)

3.4.1.3 Queuing Theory

Queuing theory is a powerful tool for manufacturing process systems modeling [8, 14]. For this project, it can treat LCIM as a cell, where jobs arrive to be processed or machined. The jobs are kept in the buffer zone (work-in-process inventory) until their process begin and are transferred to other areas in the facility after their process is completed. The machine may experience random failures. The flow of incoming jobs, either one by one or in batches forms the arrival stream, which can be identified by the statistical characteristics of the time between consecutive arrivals. Service time is usually the processing times of jobs in the LCIM cell. Random failures and repairs may augment processing times. Thus, if a LCIM includes one or more machine tools, then it can be looked at as a queuing system with single or multiple servers, with finite or infinite waiting room (input buffer) and with a service policy depending on the operating characteristics of the work place.

The LCIM cell is commonly used to manufacture medium size batches of precision parts. If there are multiple channels and different kinds of machines, then it can use Multiple-Server Model (M/M/c) to analysis the process characters, such as queue length or waiting time for performance analysis. (Interested reader can look for **Operational Research** and **Performance Analysis of Manufacturing System** for details)

3.4.2 Conceptual Modelling

For CIM, CIM Open System Architecture (CIM-OSA) is a well-known model that developed by the AMICE Consortium, and the most important CIM initiative within the ESPRIT program [45]. The term AMICE is a reversed acronym for 'European CIM Architecture'. This project is to elaborate open system architecture for CIM and to define a set of concepts and rules to facilitate the building of future CIM systems. The two main results of the project are the Modelling Framework, which is well known, and the Integrating Infrastructure (see Fig. 3-7). The Modelling Framework supports all phases of the CIM system life cycle from requirements definition, through to design specification, implementation, description and execution of the daily enterprise operation [6].

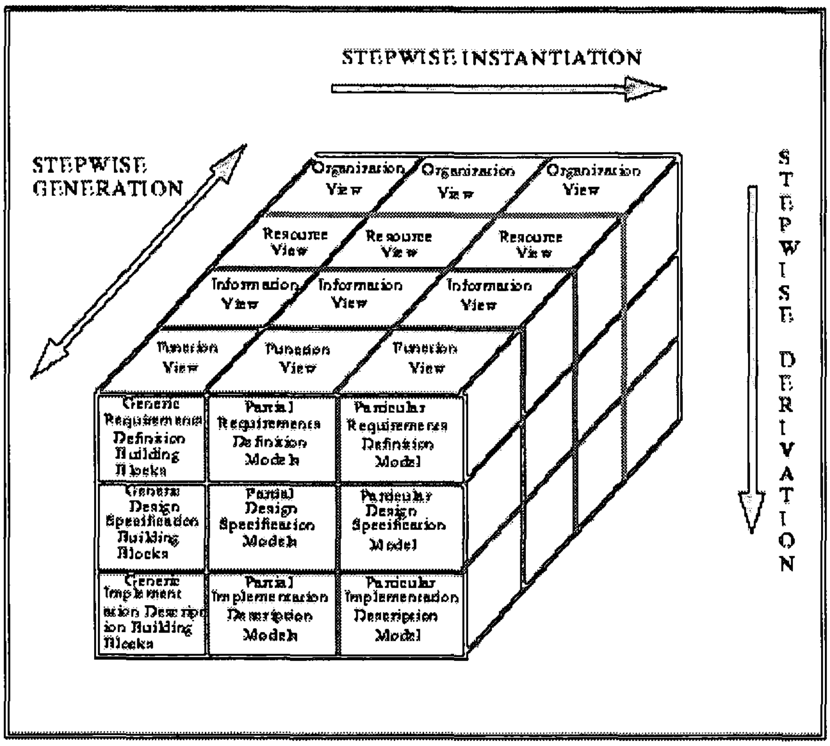


FIGURE 3-7: CIM-OSA STRUCTURE

CIM-OSA incorporates an event-driven, process-based modelling approach with the goal to cover essential enterprise aspects in one integrated model. The main aspects are the functional, behavioural, resource, information and organisational aspects [31]. For each of these aspects, modelling construction is available. This enables one to model the aspects of business processes independently from each other. Furthermore, CIM-OSA is also aiming at the execution of business processes, and not only the modelling of those. The goal is to drive an information infrastructure with the processes modelled.

CIM-OSA provides Reference Architecture for the specific description of a particular enterprise. The architectural constructs of CIM-OSA guide the CIM user in the design process to obtain a consistent system description to fit the defined user requirements through the Requirements Definition Modeling Level, Design Specification Modeling Level and Implementation Description Modeling Level. These are achieved via the Function View, Information View, Resource View and Organization View [31]. For example, the Function View at the Requirements Definition Model level specifies *what* is required, namely the required structure, content, behavior, control and required capabilities. The Function View at the Design Specification Model level specifies *how* these requirements will be implemented, namely the detailed design of the functional system in terms of functional operations. The Function View at the Implementation Description Model level describes the actual implementation i.e. the decision system and the physical system.

3.5 Simulation

Modeling and simulation of system design trade off is good preparation for design and engineering decisions in real world jobs. It needs a proper knowledge of both the techniques of simulation modeling and the simulated systems themselves.

A simulation is the execution of a model, represented by a computer program that gives information about the system being investigated. The simulation approach of analyzing a model is opposed to the analytical approach, where the method of analyzing the system is purely theoretical. As this approach is more reliable, the simulation approach gives more flexibility and convenience. The activities of the model consist of events, which are activated at certain points in time and in this way affect the overall state of the system [2].

In addition to its use as a tool to better understand and optimize performance and/or reliability of systems, simulation is also extensively used to verify the correctness of designs. Simulation early in the design cycle is important because the cost to repair mistakes increases dramatically the later in the product life cycle that the error is detected. Another important application of simulation is in developing "virtual environments". Simulation can generate dynamic environments with which users can interact "as if they were really there."

System Simulation is the mimicking of the operation of a real system, such as the running of an assembly line in a factory, in a computer. Instead of building extensive mathematical models by experts, the readily available simulation software has made it possible to model and analyze the operation of a real system by non-experts, who are managers but not programmers. By combining the emerging science of complexity with newly popularized simulation technology, the simulation provides an analysis tool for management at the beginning of the project.

3.5.1 Discrete-event Simulation

Physical experimentation within the system itself frequently is too disruptive and costly. For this reason, discrete event computer simulation is an attractive alternative. Discrete-event simulation is a powerful tool for solving many problems, especially in manufacturing. Formulating the problem, building the simulation model, running the model and analysing the output are the basic steps in a simulation study. Since building a simulation model can be a difficult and a time-consuming task, it will be useful if a decision-maker could reuse a simulation model if possible and change it to solve a different problem or evaluate another option. Thus, it is desirable to have adaptable simulation models that are easy to change with little or no programming effort. Using simulation models in real-time scheduling and operational settings also requires adaptable simulation models that can represent the changing shop floor. Also, as a manufacturing system progresses from a concept to a detailed design to an installed and operating facility, the simulation model of the system must change. Adaptable simulation models will reduce the time, effort and cost of using simulation in these types of scenarios.

As to the modeling logic of discrete simulation, there are at least four programming approaches in which a discrete-event simulation model can be written. They are the event approach, the activity approach, the process interaction approach and the three-phase approach. (Interested reader refer to **Manufacturing Systems Design and Analysis**)

3.5.2 Simulation Tools

There are a lot of tools for simulation purpose. It includes the special computer simulation language, such as Simulation languages: GPSS, SIMAN, ECSL, Slam and Programming languages: FORTRAN, BASIC, C, C++ etc.

It is also possible to use the commercial package for practice and research as well. Software package such as ARENA is used for scheduling and performance research [7]. It includes four components to build up the simulation model, namely Entities, Resources, Control Logic and Statistics. The simulation process is made up of the Define, Formulate, Verify/validate, Analysis and Recommend. The shortcoming for commercial package is that some of the settings are too arbitrary. One has to tailor one's purpose to meet the requirement.

Along with the simulation development, there are tools available that take advantage of both the Language type and Commercial type. MATLAB simulation toolbox – Simulink is one of these types. Simulink provides a Graphical User Interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. It also affords one to customize and create one's own blocks.

3.6 Artificial Intelligent

The control technology has been moving from conventional control to today's math-model based modern control and knowledge based intelligent control. The applications of intelligent systems and combination of modern control with intelligent systems have been recognised as one of the most important directions in developing a new generation of industrial control. The representations of AI are the Expert System, Fuzzy Logic Control (FLC) and Neural Network.

3.6.1 Fuzzy Logic Control

The basic FLC consists of four principle components, namely

- The condition interface;
- A knowledge base that consists of the fuzzy control rules and fuzzy set definitions database;
- The fuzzy controller itself; and
- An action interface.

The condition interface (also denoted as "justification interface") is responsible for three main activities:

- Measurement of input variables;
- Scaling the input values into a corresponding universe of discourse;
- Mapping the input variables into suitable linguistic value. This activity is refereed to "fuzzfication"

The Knowledge base contains the knowledge of the application domain and the control goals. It consists of a set of control rules and a set of definitions. These definitions represent the descriptions of the fuzzy sets that correspond to the various input parameters.

The Fuzzy Logic Controller is the unit that combines all the elements of the system into a decision-making unit than operates in real time, at timely intervals, or based on interrupt signals.

The action interface is the unit responsible for the communication of the decision made by the FLC back into the controlled system. This unit translates the decision variable into a corresponding universe of discourse and converts the fuzzy values into a crisp control action or decision (this activity is termed "defuzzification") (For more details, refer to **Intelligent Design and Manufacturing**)

The operation of the FLC consists of four steps. In the first one, the input variables are measured. Next, these variables are converted into fuzzy linguistic terms such a "Long", "Very Short", etc. These linguistic terms are then used to evaluate the fuzzy control rules. The result of this evaluation is itself a fuzzy control decision. This fuzzy control decision is then converted into a crisp control action, which is executed.

3.6.2 Neural Network

The main component of a neural network is the processing element. A neuron is an information-processing unit that is fundamental to the operation of a neural network. There are three basic elements of the neuronal model:

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own;
2. An adder for summing the input signals, weighted by the respective synapses of the neuron; the operations described here constitute a linear combiner;
3. An activation function for limiting the amplitude of the output of a neuron.

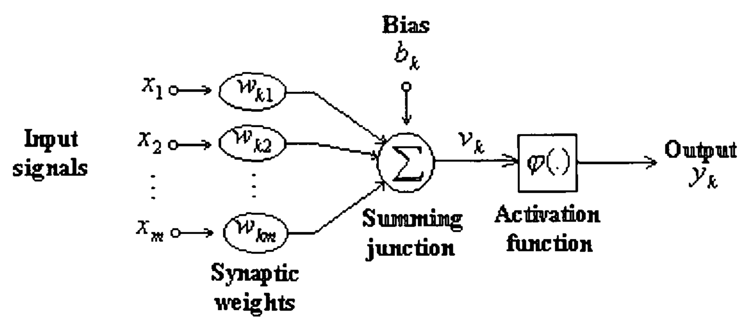


FIGURE 3-8: NEURON STRUCTURE

In mathematical terms, it can describe a neuron k by writing the following of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad (2\text{-}7)$$

and

$$y_k = \varphi(u_k + b_k) \quad (2\text{-}8)$$

where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights for neuron k; u_k is the linear combined output due to the input signals; b_k is the bias; $\varphi(.)$ is the activation function; and y_k is the output signal of the neuron. (For information regarding with this section, looking for **Neural Networks: A Comprehensive Foundation** for details)

3.6.3 Intelligent Control

Generally, industrial control covers a wide variety of functions, such as dedicated dynamic control, supervisory control, fault diagnosis, production scheduling and planning, etc. They are organised in a hierarchical and/or distributed structure and intend to perform a plant or corporation – wide optimisation under desired global (production and business) performance criteria with corresponding constraints of production facilities, resources and marketing, etc.

For a long time, industrial control has faced three major bottlenecks: process modelling, advanced control strategies and realistic approaches to system analysis and synthesis for complex systems. The control technology has been moving from conventional control to today's math- model based modern control and knowledge based intelligent control [27].

- Firstly, industrial processes and facilities have been changing and advancing considerably during the past few decades. The integration of strategic business plan, information, management, and production has been considered as a key in the successfully managing and running of an enterprise.
- Secondly, modern control theory and techniques have rapidly developed and successfully applied to system identification, estimation, optimization, robust and adaptive controls, particularly for linear systems.

- Thirdly, there has been astonishing progress in computer science and hard ware technology.

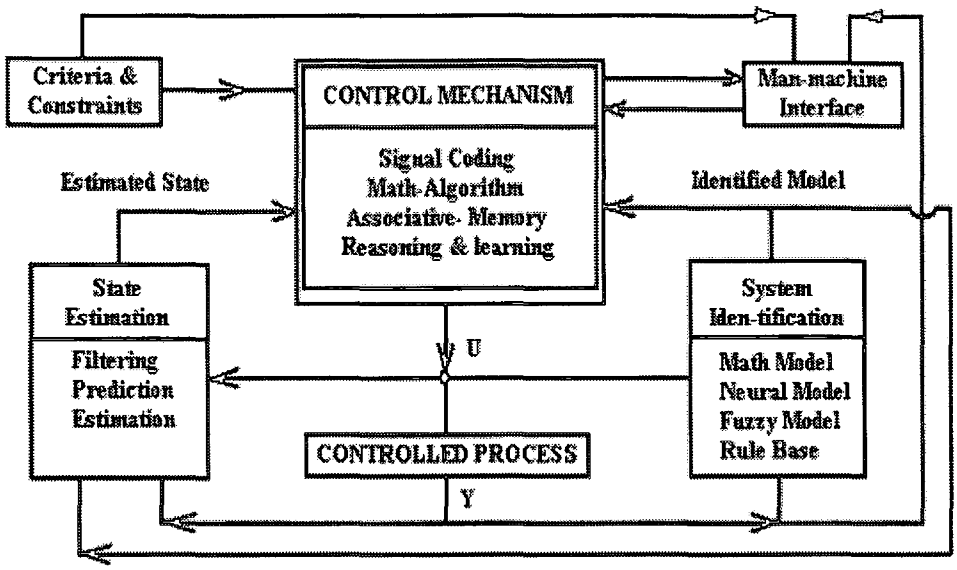


FIGURE 3-9: TYPICAL INTELLIGENT CONTROL

The new application in developing a new generation of industrial control is the combination of modern control with intelligent systems. The revolution in computer technology, modern control and machine intelligence has opened a platform for the new generation of industrial control that may provide significant economic information and system management capabilities. (For more details, refer to **Industrial Intelligent Control**)

3.7 Summary

This study is more complex than a simple pure mechanical design. It requires other fields of knowledge such as logistics management, business management, modern control theory, operational research, etc.

Modeling and algorithm development is needed to understand the behavior of these systems. To achieve this, it needs investigation of the nature of the system, with a clear picture in mind. It can use mathematical models for the analysis of the routing, queuing problems in a manufacturing environment, and, for the system structuring; it requires the use of conceptual modeling techniques.

This research cannot purely count on discrete simulation techniques, and therefore needs a higher-level of simulation tools to integrate the advanced functions. Here the discrete model can be used to generate lower-level signals that are used as the input data for the process and management analysis. AI could then also be used to fulfill the management function.

The modelling and simulation method needs a mix of various techniques. Input-output model can give a picture of the whole system. For conceptual modelling, CIM-OSA gives a very good illustration of how to systematically skim all the functions. The detail information can be generated from models such as performance models. Considering its open structuring, MATLAB seems a powerful tool for this project.

The manufacturing trends influence the validity of the research result, therefore the up-to-date research information needs to be considered. In this study, the ML and AM gives a broad picture of the manufacturing environment. This recent development requires further investigation, and will from hereon form the basis for discussion of this dissertation.

4. LCIM Modelling

As far as technical portion is considered. From the literature review it can be summarised that the manufacturing research be directed in four fields [19]. See Figure 4-1,

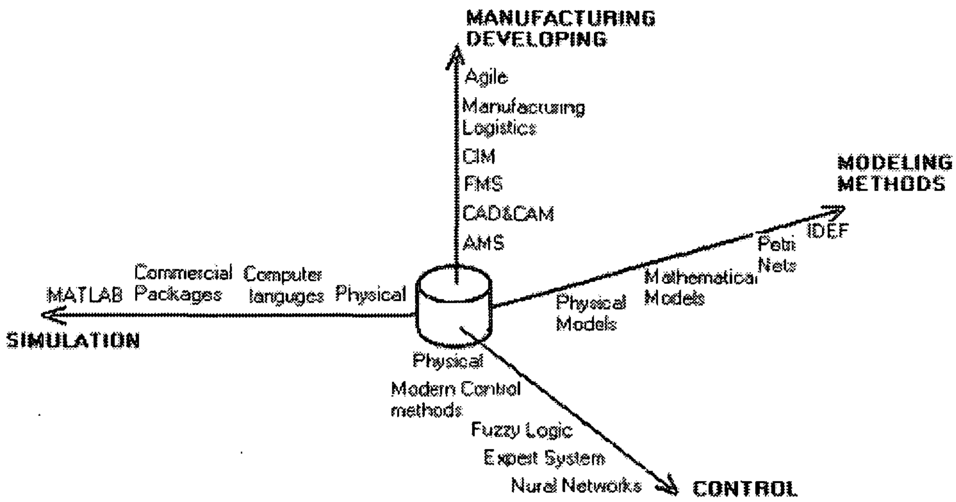


FIGURE 4-1: DIMENSION OF MANUFACTURING RESEARCH

Manufacturing developing normally is based on control methods developing, combined with different modelling methods. It is classified as different developing stages. It is the technical sides that act as the basement for manufacturing developing innovation, i.e. include Control, Modelling & Simulation integrating.

4.1 Integrating Approach for CIM Modelling

CIM can be defined as the use of computer systems and automation systems to operate and control production. To achieve CIM, all aspects of the manufacturing establishment must be integrated so that they can share information, communicate with one another and provide a picture as to the state of the entire manufacturing facility at any time.

CIM research can be divided into two trends. One is at the micro-level; it focuses on the decision-making problem of introducing advanced technologies to the company or factory and makes use of *ad-hoc* cost-benefit analysis and case studies. The other approach is at the macro level; it characterizes the technology diffusion process by means of statistical data and econometric methods. This project is at the micro level to investigate the new trends for CIM system.

4.1.1 Methods for Modelling

Many CIM models are been used for research, which are as IBM concept of CIM and the SIEMENS concept of CIM (For details, **Computer Integrated Manufacturing and Engineering**).

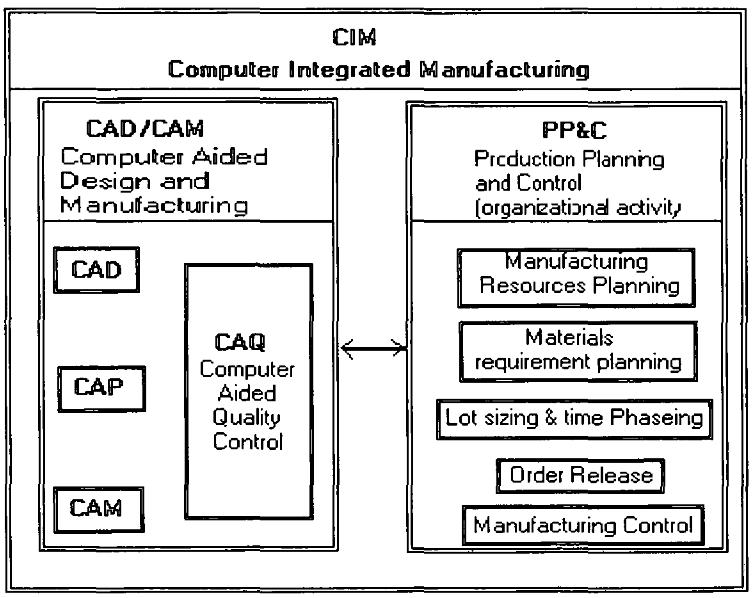


FIGURE 4-2: NORMAL CIM CONCEPTUAL MODEL

One of the shortcomings for these models is the development of a case analysis chain from the CIM model, and further more it is not integrated into development such as M L, AM.

A CIM cell should be in a balanced situation and can cover the various system functions. There is a CIM FIA framework based on IT perspective on Flexibility, Integration and Automation [30]. The CIM-OSA model provides a good idea that is based on certain variables and is divided by different levels to develop different models, which can cover the related functions.

Using the conceptual modeling method, it can propose a new CIM conceptual model. The required outputs from this conceptual modeling phase are identified here. As can be seen Figure 4-3, this phase must identify the building blocks of the system concerned. These blocks are a combination of manufacturing functions, together with the necessary controlling functions. The inner relationship will be analyzed via a case study.

4.1.2 New CIM Conceptual Model

In order to make it adaptable, the model should be designed as modular as possible, otherwise it will still remain in one aspect of manufacturing. This new CIM model includes three modules, namely physics, process, and management module. Through integrating, all of these and outside environment bond together. Each module also can be divided into sub-function modules.

Basically, three essential sub-systems is needed within a CIM cell- an auditoria sub-system; an operational sub-system, and a managerial sub-system. The operational sub-system consists of the physical resources, including humans and machines, which is used to transform raw materials into end products. This sub-system is controlled by the managerial sub-system, which makes the periodical decisions to reach the objectives set by the wider system. The auditoria sub-system provides accurate information for decision-making.

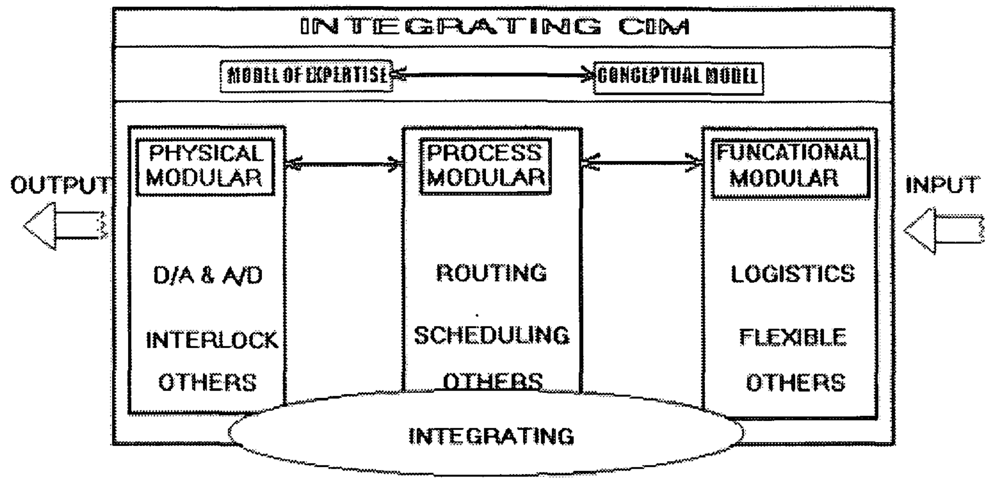


FIGURE 4-3: NEW CIM CONCEPTUAL MODEL

As discussed in literature review, CIM's activities can be classified into three basic function blocks. The first is for physical activities, the second is to fulfill information processes, and the last is to meet the tasks such as Logistics & Management requirements. This has an advantage against with former models and has a clear classification for current CIM activities. This model has adopted the modular base and it also takes the advantage of the Input-Output model that packs those modular as a whole to interface with outside environment. In this way, it gets a clear task for inner CIM integrating.

4.1.3 Implementation for Lab's CIM Problem

As stated for CIM Lab situation, it uses one robot (Gryphon) to server two CNC machines (one is DENFORD CNC Milling, and the other is Emco PC TURN 120). This small change will bring lots of integrating problems for the CIM cell. It involves three function modules.

4.1.3.1 Interlock Problem

Some interlock problems do exist, such as door and chuck interlock problems, which are already settled down by programmable logic controller (PLC) as CNC built in function. The problem here, which is present, is the Robot integrating interlock problem.

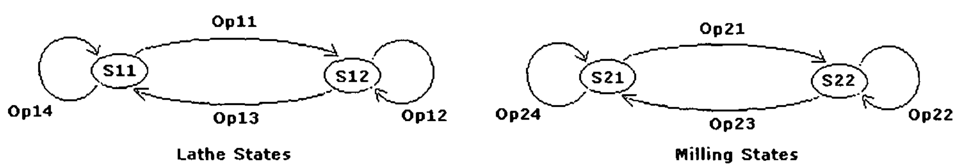


FIGURE 4-4: ROBOT INTEGRATING INTERLOCK STATES

S11: CNC Lathe chuck open, S12: CNC Lathe chuck close,

S21: CNC Milling Verse open, S22: CNC Milling Verse close.

Op11: Robot loads raw materials into the Lathe,
Op12: Robot finishes load raw materials back from Lathe,
Op13: Robot operates unload finished part to Lathe,
Op14: Robot operates unload finished part back from Lathe,
Op21: Robot operates load raw materials to Milling,
Op22: Robot finishes load raw materials back from Milling,
Op23: Robot operates unload finished part to Milling,
Op24: Robot operates unload finished part back from Milling,

There have conflicts with such situations:

$S11 \bigcup S21, Op11 \bigcup Op21$

And further more it should wait to fulfill JIT demand. So intelligence is needed to arrange the parts routing.

4.1.3.2 Routing problem

Modern computerized manufacturing systems increasingly accentuate small batch production with high variability of products. This tendency makes routing and scheduling combinatorial intractable problems. In addition the complexity of the problem is increased and the time frame in which the solution is needed keeps decreasing. To efficiently use the facilities, CIM cell needs routing arrangements to work. The fuzzy logic approach is a tool for this.

4.1.3.3 The Routing Problem Description

The Fuzzy Logic Control (FLC) is required to put to fight defeat utilizes parts in this CIM system, which consists of two similar machines, one robot to serve one milling and one lathe. It is the task of the FLC to rout the parts to the machines in a way that maximizes the production throughput of all part types combined. This means that when an operation is completed, the FLC chooses the machine for the next operation. The processing times of the various operations of each part are stored in a database. For simulation purposes, the processing time assumed to be normally distributed with a standard deviation that is also a part of the process description.

The parts arrive continuously to the manufacturing environment with a given inter-arrival time interval. The machines are subject to breakdowns, where each machine has a different probability for such an event (time between machine failure is exponentially distributed, while repair time is normally distributed). The FLC operation is based on three input variables from the system. The input variables are Job Processing Time, Queue Length in each machine's buffer, and the machine's Breakdown Rate.

The three input variables are associated with linguistic values, which are defined as follows:

Processing time: very short (VS), short (S), medium (M), Long (L), and very long (VL).

Queue length: very short (VS), short (S), medium (M), long (L), and very long (VL)

Machine breakdown rate: very small (VS), small (S), medium (M), large (L), and very large (VL)

The linguistic values are represented using fuzzy triangular numbers of the form [a, b, c]. Here the values are defined based on a discrete event simulation of the system, or chosen heuristically.

Table 4-1: Fuzzy Values of Input Variables

Input Variable	Linguistic Value	Fuzzy Value
Processing Time	VS	(1,1,12)
	S	(1,12,24)
	M	(12,24,34)
	L	(24,35,47)
	VL	(35,47,47)
Queue Length	VS	(0,0,4)
	S	(0,4,10)
	M	(4,10,23)
	L	(10,23,44)
	VL	(23,40,44)
Machine Breakdown	VS	(2%, 2%, 5%)
	S	(2%, 5%, 25%)
	M	(5%, 25%, 40%)
	L	(25%, 40%, 50%)
	VL	(40%, 50%, 50%)

To make the selectibility membership, the management demand should be considered. In real-life discrete-part manufacturing situations, the total number of jobs normally exceeds the number of work centres, resulting in a queue of jobs to be processed. Shortest

Processing Time (SPT) rule gives the highest priority to the job with the shortest processing time. This rule results in the lowest mean completion time (That is, average manufacturing lead-time) and consequently the lowest work-in-process inventory. It can base on this to set up the selectibility factors.

In order to calculate the selectibility factor, a weight *Wij* is associated with each input parameter *i*. In this case the parameter *j* represents the linguistic value of the input variable. **The intention of the weights is to create preference of the machine with the shortest queue, fastest processing time, least break down rate and highest slack time.** In all these cases the weight is the highest. Furthermore, **the weights are decreasing exponentially rather than linearly when the parameters become less accommodating for a particular operation.**

Table 4-2: The Value of the Parameters Weight

Parameter	Values				
	VS	S	M	L	VL
Processing Time	1	exp-1/3	exp-1	exp-2	exp-7/3
Queue Length	1	exp-1/2	exp-1	exp-3/2	exp-2
Breakdown rate	1	exp-1/3	exp-1	exp-2	exp-7/3

Initially, the Selectibility Factor (SF) is calculated as:

$$SF = \frac{\sum_{i}^{n} W_{ij}}{n}$$

The SF is assumed to have one of the following possible values: {Very Large, Large, Medium, Small, and Very Small}. The fuzzy description of the SF is presented in Figure 4-5.

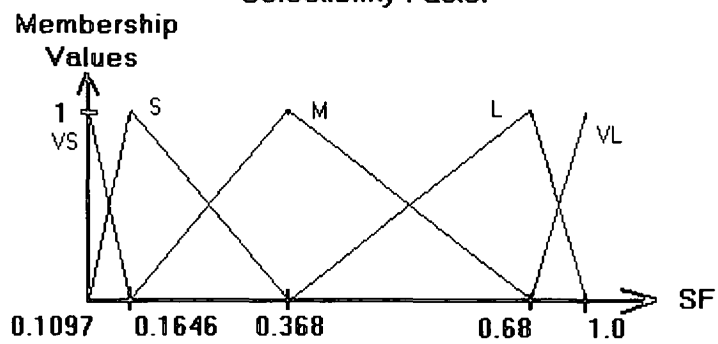


FIGURE 4-5: FUZZY REPRESENTATION OF THE SELECTIBILITY FACTOR

Set down Fuzzy members, Fuzzy rules, a Fuzzy Interface System (FIS) based on MATLAB can be developed (as illustrated in Fig 4-7). It still needs to be integrated into the system through performance analysis.

4.1.3.4 Performance Evaluation

Typical performance measurements in a manufacturing environment are output rates or throughputs (number of units produced per unite time) average inventory levels (inventory levels averaged over a planning horizon) utilization (percentage of time a machine is actually busy). This uses queuing theory to model CIM cell to measure the performance of it. The process is illustrated in figure 4-6.

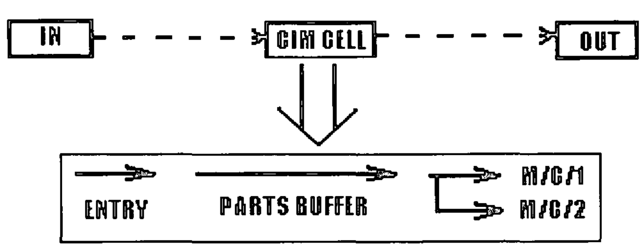


FIGURE 4-6: QUEUING THEORY APPROACH

For this case the CIM cell is treated as A Single –Stage System with Two Customers (Use one robot server two machines). The underlying stochastic process is stopped due to complicated interactions among the products, and it processes an infinite number of states. The approximation method is used to obtain the performance measures of interest. The accuracy of the approximation can be checked via simulation.

4.1.3.5 Integrating Model Control Loop

Through performance measurements, it can provide feedback adjustment for routing arrangement. For example, it can adjust the related weight to make CIM cell more adaptable for the requirements.

In this case, it gives a whole control loop for the problem. It can be illustrated as follows:

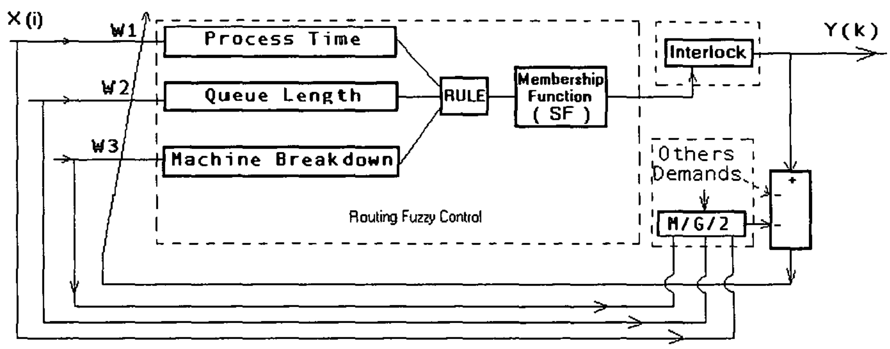


FIGURE 4-7: FUZZY-QUEUE-CONTROL LOOP

It is clear that CIM must use an integrating method to solve the problems. It is better to develop methods with the integrating CIM system concepts.

4.1.4 Summary

In this stage, the methods for CIM modeling are analyzed. Based on the inner and outside environment requirements, a new integrating CIM model was presented. Through a case study, it is proved that it is the system's requirement to deal with a single problem that must involve the whole system to handle. But the disadvantage for this model is that the logistics function, which has to be included in the management function. There is no significant improvement for CIM to suit for logistics requirement. This needs further investigation.

4.2 How to Implement Logistics Function

The three function modules can cover basic CIM functions. The Logistics function has been classified to management function module. Considering the core for logistics function in this level is to streamline the whole system, not only for the internal co-operation, but also interface with outside. In this point of view, it should separate the logistics out of the management function, because it gets bigger scope and a deep influence for the inner cooperation.

Here the logistics function is not limited for inventory control and short lead-time management. It covers the whole CIM system activities that will lead to CIM flexible and meet the all "right" requirements. But this concept is ambiguous. This project takes a popular manufacturing flexibility problem for analysis so that it can get the core of CIM logistics.

4.2.1 Logistics Supporting Concept: Flexibility Analysis for CIM Cell

Along with marketing philosophy central changed from production, product & sell concept to market, societal and customer-oriented concept, and it is urging the manufacturing more adaptable and flexible. During the past few years, flexibility in manufacturing has aroused interest among researchers and professionals [25, 32].

Technologies and management innovations continued to contribute manufacturing flexibility. Technology will promote the optimisation of outputs, qualities and so on. But in practice, many enterprises has adopted the new technologies but failed to improve the system flexibility [33].

How to implement the new technology and in the meanwhile to achieve a satisfy integration? It needs a deep investigation, especially from logistics co-operation view [20].

4.2.1.1 Manufacturing Flexibility

Manufacturing flexibility, generally speaking, is the ability of the system to adapt effectively to changes, and based on the scope of manufacturing system, comprehensive flexibility defines flexibility with respect to the manufacturing system as a whole, and quantifies it as a performance measure [11].

Former researchers identify flexibility on two distinct dimensions, which range (what changes are possible) and response (how easy changes can be made), and give the framework for analysing manufacturing flexibility. It is divided into three parts: dimension (the things that require changes must be determined in order to understand exactly why flexibility is needed); time scale (the period of time over which these changes will occur must be determined); elements (the three elements of flexibility that are most important to the manufacturing system must be determined --range, uniformity across range or mobility) [3].

Current manufacturing trends require the combining of the organization, people and technology into an integrated and coordinated system that incorporates good agility. This makes the flexibility of manufacturing system more important. Manufacturing flexibility requires many fields integration. It includes manufacturing systems, operation and management, logistics, control theory, artificial intelligence and knowledge of different support systems.

4.2.1.2 The Nature of Flexibility

The lack of understanding the complex nature of flexibility has caused managers to doubt about the advantages of CIM as tools for improving the flexibility of their manufacturing systems. It is important to view flexibility from a practical approach.

1) Flexibility changes in certain systems, when it is compared with others.

2) Flexibility is the description for system's performance. It must pay attention to the environment effects. Under certain situations, even the same input can get absolutely different results, and different inputs can give the same outputs.

3) Flexibility is more meaningful for the real case analysis.

4) Flexibility of manufacturing is a dynamic concept rather than static one. Flexibility normally will decrease along with time, but it also can increase in certain situations. It has a FLC like goods.

Some characteristics can be illustrated as in figure 1. Different approaches can arrive to the same relative flexibility. Same approach under different situation can drive to different relative flexibility.

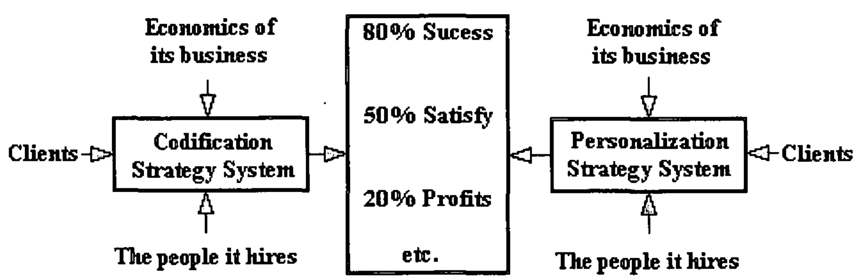


FIGURE 4-8: DIFFERENT APPROACHES TO FLEXIBILITY

Thus a performance criterion seems to be suitable technique to measure manufacturing flexibility types. The other methods are extremely difficult to obtain the required data or some of the methods get very complex as the system size increases [5]. So here it will use performance concept to analyse CIM system flexibility.

4.2.1.3 CIM Flexibility Analysis

It should be aware that many computerized systems have been installed for reasons other than increasing plant flexibility. Those reasons include facilitating the tracking and delivery of information and improving the quality of the process. But, in the meanwhile, it gives the potential to gain more flexibility for the system.

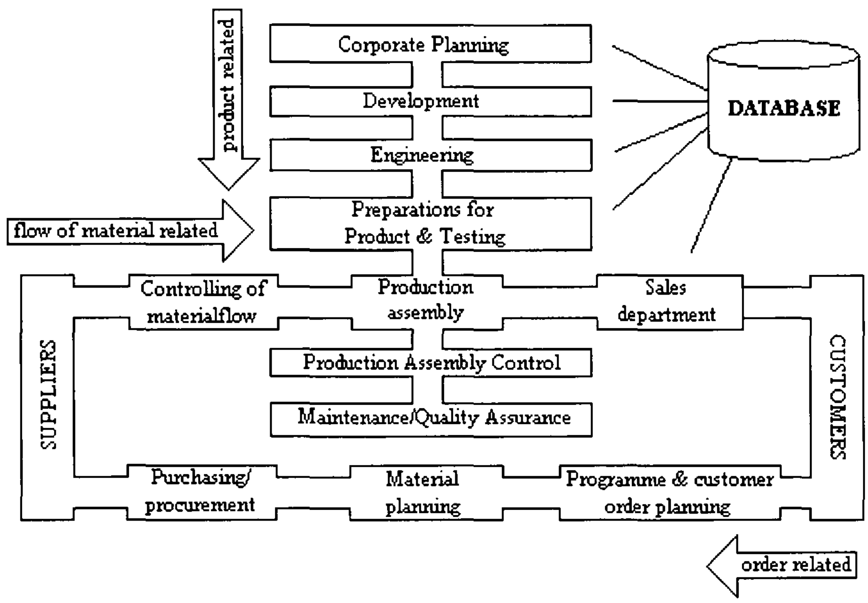


FIGURE 4-9: BMW CIM LOGISTICS STRUCTURE

Generally the complaints for CIM flexibility come from management requirements. It is true that there are a number of so-called intangible benefits that can be derived from CIM. It can increase certain flexibility, faster response to market shifts, and reduced throughput and lead times. But judged by current manufacturing requirements, CIM still lacks flexibility [33].

Before analysing manufacturing flexibility, it needs to precisely identify what kinds of flexible capabilities are required to support the competitive needs. The *CIM logistics* diagram illustrates a production phase CIM logistic layout. From this structure it can be derived that there exist two main flexibility frames, namely Technology flexibility cycle and Business (Management) flexibility.

Basically, Technology Flexibility is the product-related line. It starts from corporate planning to maintenance/quality assurance. Business Flexibility normally can refer to order related & flow of material related cycle. In fact, even in this two main flexibility cycle, they still merge with each other. Here they are divided into these two groups just for easy location of support target purpose due to having different characters.

4.2.1.3.1 Technology Flexibility & Business Flexibility

Flexibility means different things to different people. At the CIM plant level, it could be understood that technical flexibility such as machine flexibility, material-handling flexibility, process flexibility, routing flexibility, programme flexibility etc. and business flexibility such as operation flexibility, product flexibility, volume flexibility, expansion flexibility, market flexibility, etc, will normally maintain a relatively stable level when CIM system is set down. It will decrease as time increases. The various types of flexibility can be classified. See Table 4-3 (next page).

4.2.1.4 Logistics View Approach for the Integrating

There are some conflicts between technical flexibility and business flexibility. Plant with more CIM hardware has less range and longer changeover times. The software has misled many companies that have installed software–based CIM to improve flexibility. In many cases, the additional complexity resulting from computer integration has demanded new and expensive skills that have not translated into clear advantages in the market. More data is not the same as more information. Systems often sidetrack manufacturing organization from the tasks they should be doing well [33]. So it is necessary to develop a support method to balance this situation.

Table 4-3: Items measuring Manufacturing Flexibility (Gupta and Somers, 1992)

Flexibility Types	Measuring Items
Programming Flexibility	-The manufacturing system is capable of running virtually unattached during the second and the third shift.
Process Flexibility	-Changeover cost between known production tasks within the current production programme is extremely low.
	-The ratio of the total output and the waiting cost of parts processed are extremely low.
Machine Flexibility	-The number of different operations that a typical machine can perform without requiring a prohibitive time/cost in switching from one operation to another is very high.
Routing Flexibility	-Cost of production lost as a result of expediting a pre-emptive order is extremely low.
	-Decrease of throughput because of a machine breakdown is extremely low.
Material Handing Flexibility	-The ability of the material handing system to move different part types for popper positioning and processing through the manufacturing facility is extremely high.
	-The ratio of the number of paths the material handing system can support to the total number of paths is very high.
	-The material handing system can link every machine to every other machine
Volume flexibility	-The range of volumes in which the firm can run profitably is extremely high.
Marketing Flexibility	-Shortage cost of finished products is extremely low.
	-Cost in delay in meeting customer orders is extremely low.
Product & Production Flexibility	-Number of new parts introduced per year is very high.
	-Size of the universe of parts the manufacturing system is capable of producing without adding major capital equipment is very large
Expansion Flexibility	-The time required to introduce new products is extremely low.
	-Time required adding a unit of production capacity is extremely low.
	-The cost of doubling the output of the system is likely to be extremely low.
	-The capacity (e.g. Output per unit time) of the system can be increased when needed with ease.
	-The capacity (e.g. Quality) of the system can be increased when needed with ease.

However, there is a method to integrate technical flexibility and business flexibility. It analyses manufacturing flexibility via two models: one is to start from business flexibility, which links the system's characteristics (capacity, batch size, lead-time) with four desirable flexibilities (products, mix, volume, delivery). The technical flexibility links the former with the resource characteristics of machines, labour and infrastructure. But the integration is a complex process, and one possible result is as illustrated in Figure 4-10.

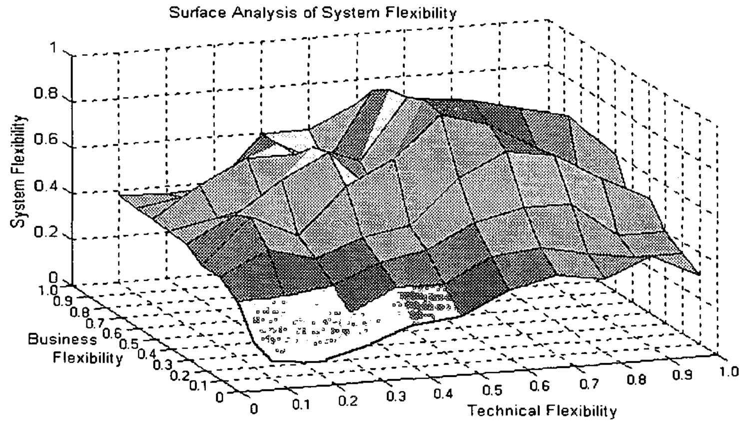


FIGURE 4-10: SURFACE VIEW OF SYSTEM FLEXIBILITY

Use logistics support concept to spiral developing manufacturing flexibility is a flexible tool for this situation, and Logistics support can be implemented through the whole FLC. There are two steps to go.

1) Using Fuzzy Logic Control to model and simulate the CIM flexibility. Fuzzy logic control is a good tool for vague condition control. It starts from Top-Down method to locate the Business Flexibility demand, then simulate from Bottom-Up method to test the Technical Flexibility. Once started, it can enter the dynamic evaluation support cycle. It is possible to use its self-adapt function to automatically adjust to response for different life cycle phase.

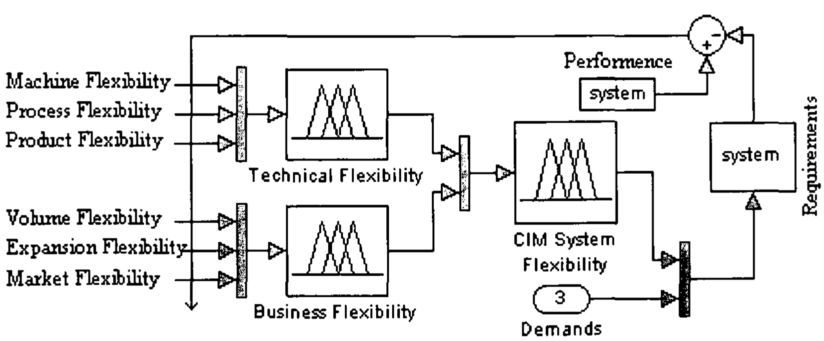


FIGURE 4-11: DYNAMIC EVALUATION SUPPORT CYCLE

2) Revise the dynamic logistic support processing. It is try to use logistic support plan to streamline the whole performance. Use logistics concept, it can combining Computer and People to build CIM flexibility, designing the right mix of machines, computer systems, and people, and figuring out the most effective way to orchestrate them.

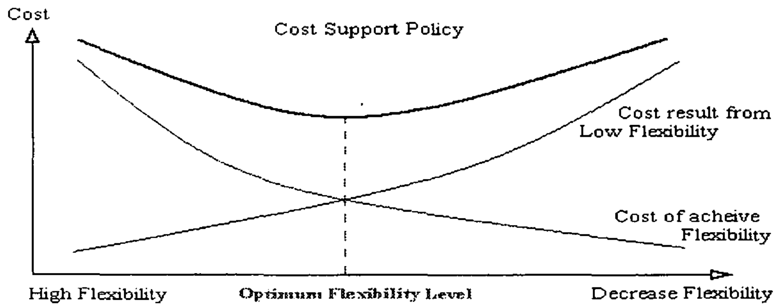


FIGURE 4-12: ONE FLEXIBILITY SUPPORT POLICY

One of the weak points for CIM system is the Human Machine Interface. Turning only to machines, hardware and software as the solution will not be sufficed. People and management count more than machines. It is important during the logistic system maintenance phase to develop related human integrating support system to maintain it at a certain level. Figure 4-13 gives a kind of logistics support plan for the system.

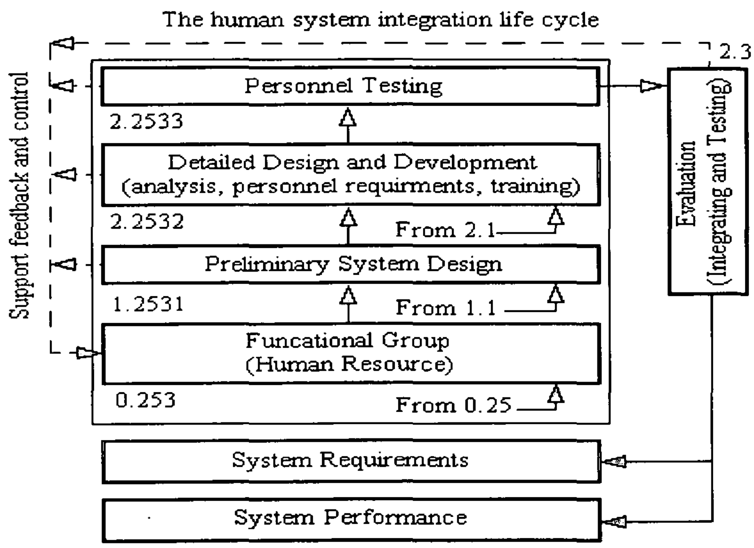


FIGURE 4-13: FLEXIBILITY SUPPORT

Through proper supporting, people can be brought together, in dynamic teams formed around clear identified market opportunities, so that it becomes possible to lever one another's knowledge. Through these processes is sought the transformation of knowledge and ideas into new products and services, as well as improvements to existing products and services.

Normally for systems, in the initial stage, the technical flexibility is more 'flexible' than business flexibility. But along with technical flexibility's development, it demands more management's support. Thus different support evaluation methods are needed.

4.2 .2 Integrating LCIM Model

From CIM flexibility analysis, it perceives that logistics concept could be interpreted from logistics supporting in the macro point of view. By doing so, the former logistics activities such as inventory control, MRP, lead-time etc. can be viewed as support implementation tools.

Generally, the requirement of LCIM comes from management organization. Technology will promote the optimisation of outputs, qualities and so on. But the management methods, such as JIT, TQM, which emphasize on only the right thing to be done on the right time. The cooperation between the technology and management is highly required. But how much cooperation is required? It depends on the industry environment.

Manufacturing planning and control is concerned with manufacturing the right product types, in the right qualities, at the right time, at minimum cost and meeting quality standards. Without logistic cooperation, it is hard to achieve such requirements.

4.2.2.1 The Core for CIM Logistics

In agile manufacturing, the aim is to combine the organization, people and technology into an integrated and coordinated whole. The agility that arises from this can be used for competitive advantage, by being able to respond rapidly to changes occurring in the market environment and through the ability to use and exploit fundamental resource-knowledge. The manufacturing system requires many fields integrating, which includes manufacturing system, operation and management, logistics, control theory, artificial intelligent and other knowledge support. All of those raised co-operate and integrating problems.

Logistics supporting is the way to solve these problems. It should focus on integration and co-operation that requires spiral development and dynamical integration as the key points.

4.2.2.2 The LCIM Model

Based on former integrating CIM model and take considering for logistic function, LCIM can be interpreted as the following model [21].

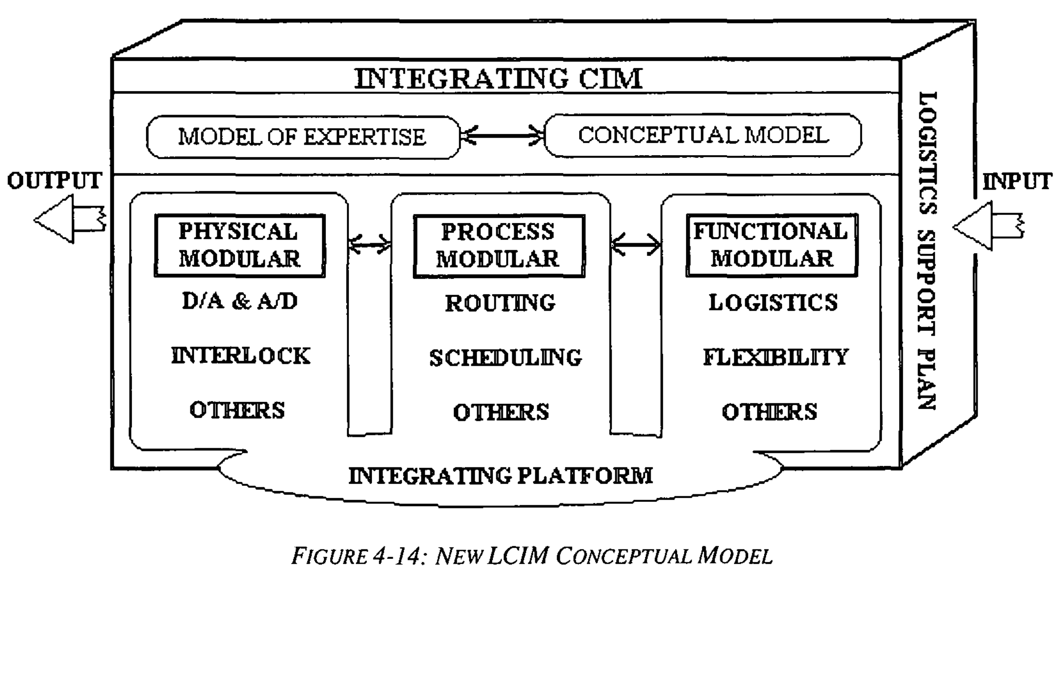


FIGURE 4-14: NEW LCIM CONCEPTUAL MODEL

The inner function modules are still classified as three modules, namely physic, process, and management functions. The logistics supporting acts as a fundamental interface with outside environment, but still needs to cover the inner logistics problems. Through integrating, the inner logistics and outside environment bond together. Each module also can be divided into sub-function modules.

Based on this integrating CIM model, logistics support can be applied, and this is the beginning of LCIM. The logistics support plan can cover all associated activities during the production process.

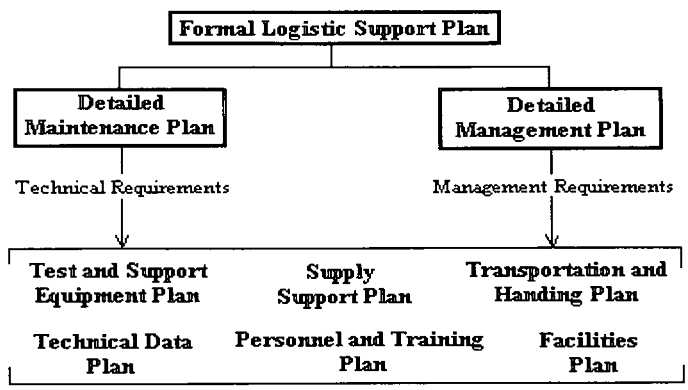


FIGURE 4-15: LOGISTIC SUPPORT PLAN DEVELOPMENT

Due to the modules having different properties; it needs different support plans for different modules. For example, during the operation, the physical module can concentrate on maintenance support. An example of one possible support cycle is shown in Figure 4-16.

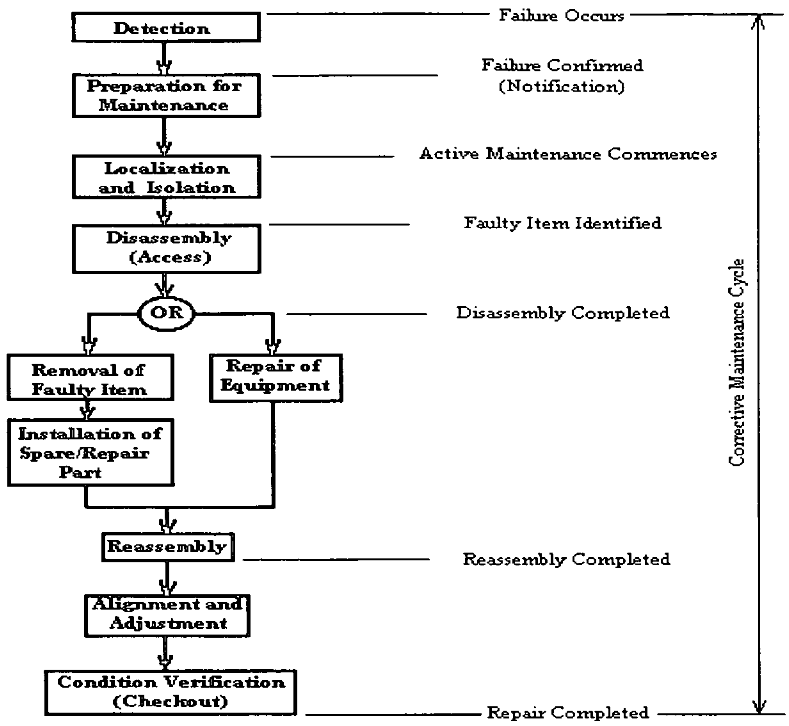


FIGURE 4-16: CORRECTIVE MAINTENANCE CYCLE

The LCIM cell requires planning, co-ordination and support functions to carry out manufacturing activities. It requires integration across multiple manufacturing system facilities and builds up a communication channel with other co-operate functional areas. This needs basing on the local CIM cell's processing and inventory control. Together, it is mixed for the broad and narrow view of manufacturing logistics.

4.2.3 Summary

From the above analysis, it is clear that the logistics supporting is the medium that can link the outside function with the internal process functions for CIM. Logistics supporting is the start point application for LCIM. It needs to base on the information which generated from the inner functions to design the supportive scheme.

The logistics concept can be interpreted by a supportive scheme. In practice, it requires the understanding of the supporting target. For example, CIM flexibility is achieved by the following process, which starts from the problem understanding, variables classification (i.e. here based on the CIM logistics structure, divided flexibility into two kinds of sub-flexibility), to supportive method identification (for this case, flexibility support is use FLC to balance those two groups).

The development of systematic methods for implement logistics into CIM is challenging the researchers. Much work still needs to be done. Works such as developing a clear understanding of what is meant by the requirement of logistics– in a form suitable for manufacturing – and identifying the characteristics of operations that support that capability, the ways that managers can use to identify and exploit the requirements, to meet the requirements, thus to provide a lasting competitive advantage, etc.

5. LCIM Simulation

Computer simulation has found a wide range of applications in the design of Advanced Manufacturing System. A computer simulation model can cope with the complexity of the system and in addition, provides an effective communication tool to explain operation clearly to the prospective investors. Compared with direct real experimentation, the computer simulation approach has the advantages of lower cost, shorter time, greater flexibility and much smaller risks. This methodology has been extensively used in the area of manufacturing systems studied by both academic researchers and practical engineers. Academic researchers can use simulation techniques to seek a better understanding of the characteristics of various manufacturing systems in an attempt to develop more rational organisation structures, more effective manufacturing strategies and more production control polices [2].

One of the significant advantages for computer simulation is flexibility. In this project, LCIM model is developed with a flexible structure, which can modify quickly and cheaply to integrate with new functions. By using this model, different results can be derived from different input data.

In reality, there are uncertainties for LCIM to work in a changing environment, i.e., different orders will come in the different time, thus it will require different input data and speed to work. In order for LCIM to work more effectively, the computer simulation must be built up. (LCIM should be able to analysis the transient features)

In this project, the task for LCIM simulation is to interpret the LCIM conceptual model into "real" construction. In Modelling part, the LCIM concept has already been explained. But, does it work in the real world practice? And what attention should be paid in the implementation? Here CIM Lab's case study still be used, which was mentioned in LCIM modelling, as an example. There are two CNC machines and one Robot in the laboratory at M L SULTAN Technikon. Figure 5-1 below shows how the two machines and the robot operate when LCIM is implemented.

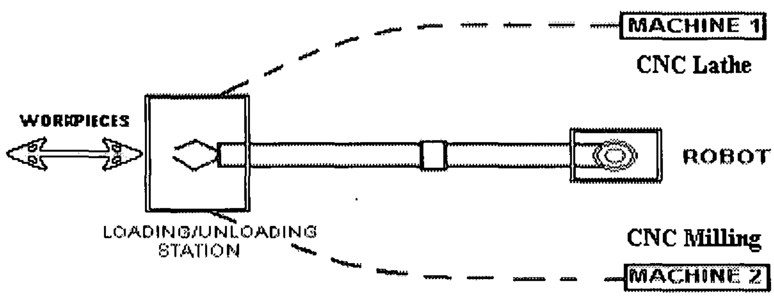


FIGURE 5-1: CELL ILLUSTRATION

5.1 Simulation Tools

This project considered using AREANA for simulation purposes, and it has developed a model that tries to simulate the production process [18]. AREANA can be used for simulation, and it can generate the data for management analysis, but there are limitations for using AREANA. Compared to MATLAB, it lacks open structure. MATLAB Simulink is more suitable for this project.

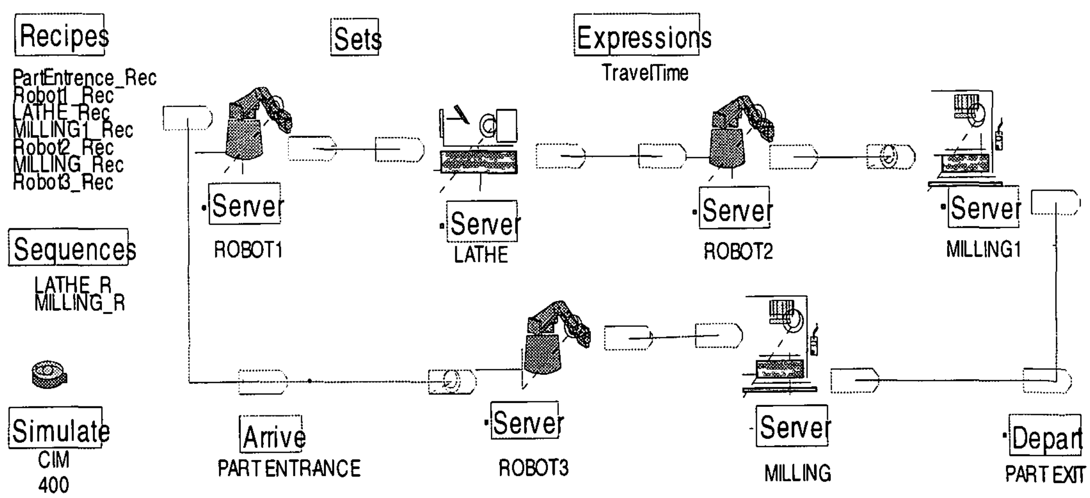


FIGURE 5-2: ARENA MODEL

5.1.1 Simulink

Simulink is a software package for modelling, simulating, and analysing dynamical systems. It supports linear and non-linear systems, modelled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates.

For modelling, Simulink provides a Graphical User Interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. Simulink includes a comprehensive block library of sinks, sources, linear and non-linear components, and connectors. It also can customise and create own blocks. The simulation results can be put in the MATLAB workspace for post processing and visualization.

One of the key features of Simulink is that it is built on top of MATLAB. As a result, Simulink users have direct access to the wide range of MATLAB- based tools for generating, analyzing, and optimizing systems implemented in Simulink. These tools include MATLAB application toolboxes, specialized collections of M-files for working on particular classes of problems.

Toolboxes are more than just collections of useful functions; they represent the efforts of some of the world's top researchers in fields such as controls, signal processing, and system identification. MATLAB application toolboxes therefore let you 'stand on the shoulders' of the world-class scientists.

5.3 LCIM Simulation Based on MATLAB

The LCIM is a big system for developing. Here the simulation is only to present the discussed process cycle in model. In real practice, the process can be divided into the internal routing and external routing. The whole cycle is central with intelligent control. It acts as the conjoint to link other activities together. FLC will be used to solve the external routing arrangement. The internal routing can be solved using the similar way.

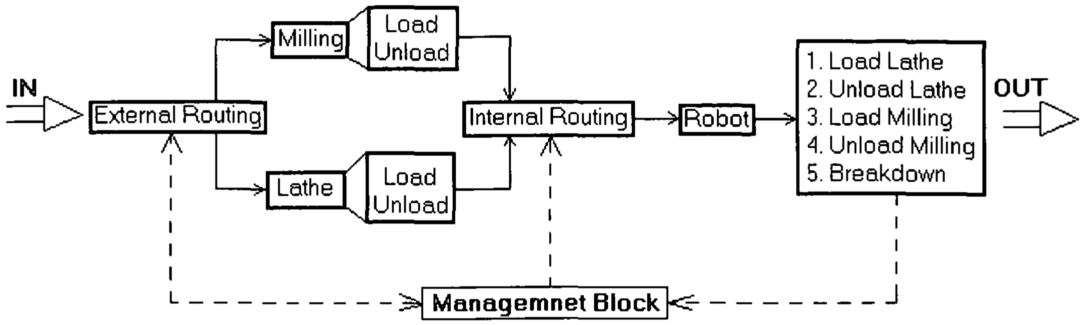


FIGURE 5-3: LCIM PROCESS CYCLE

5.3.1 Physic Function Modular Design

The physic process is the robot loading and unloading movement. The basic robot active cycle is shown in Figure5-4 below.

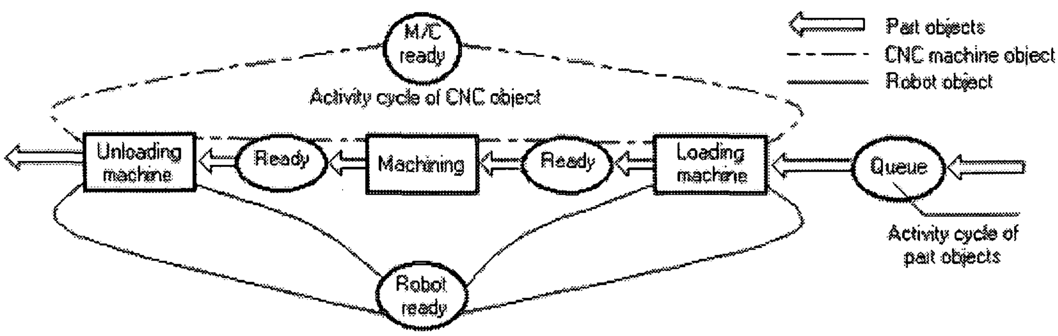


FIGURE 5-4: ACTIVITY CYCLE OF ROBOT

From input-output view, the physic process can be represented by SWITCH signals. It is the positive signal that triggers the robot active the loading and unloading program. The negative signal means that the robot is in idle state.

The basic decision making flow is as shown as the flow chart as follows. It shows how the 'SWITCH' can be triggered.

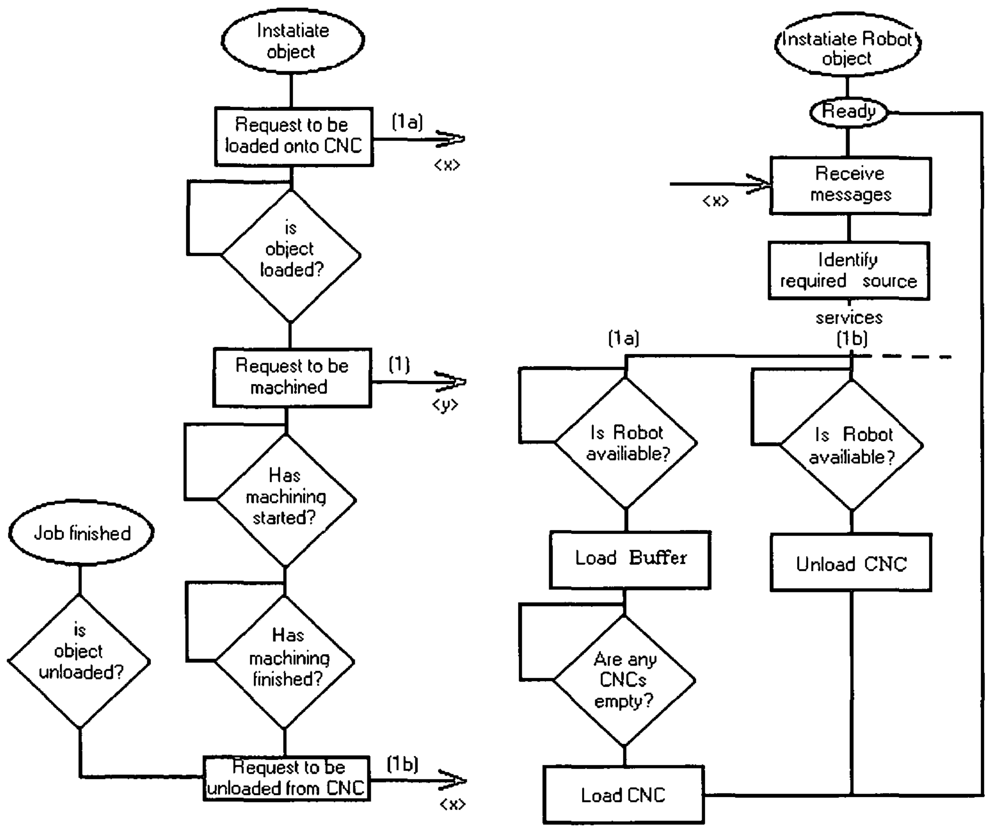


FIGURE 5-5: CONTROL LOOP

In real implementation, the robot activity is controlled by a control signal. The movement can be illustrated on computer by building up a simple animation model. The animation model is shown below in Figure 5-6.

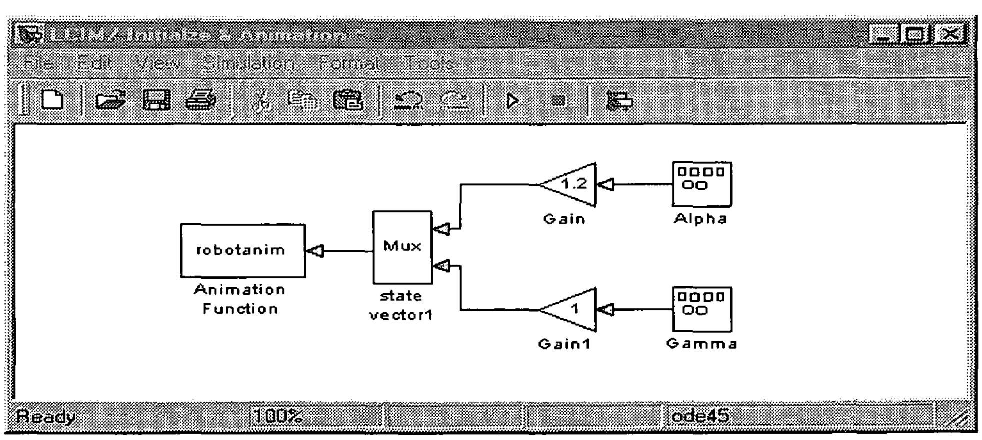


FIGURE 5-6: ANIMATION MODEL

This model can be controlled by the animation-function call. The two states of animation figures are shown below in Figure 5-7.

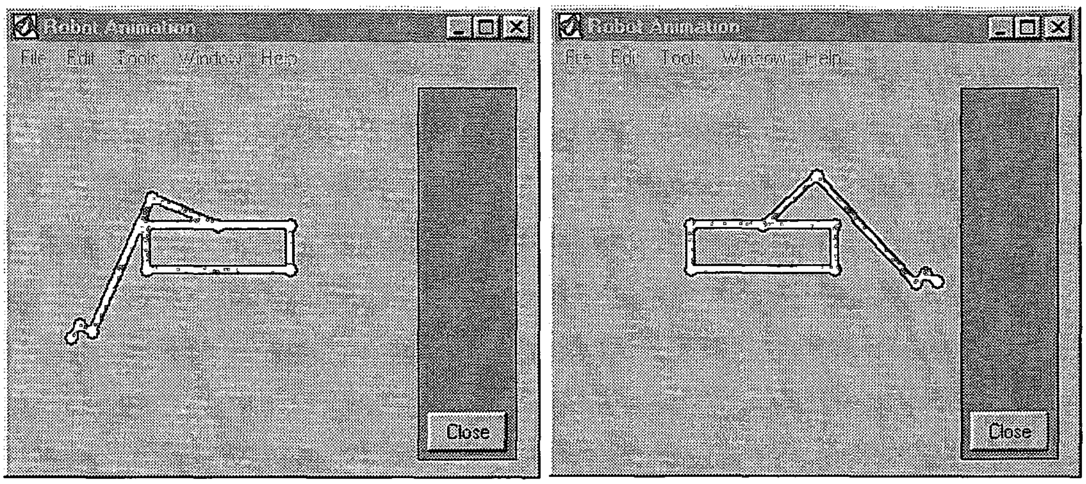


FIGURE 5-7: ANIMATION DISPLAY

In the real world, it uses A/D & D/A to translate the trigger signal into real time control signal. Normally, the control can be done using a PC Card. The final possible working connection is shown in Figure5-8.

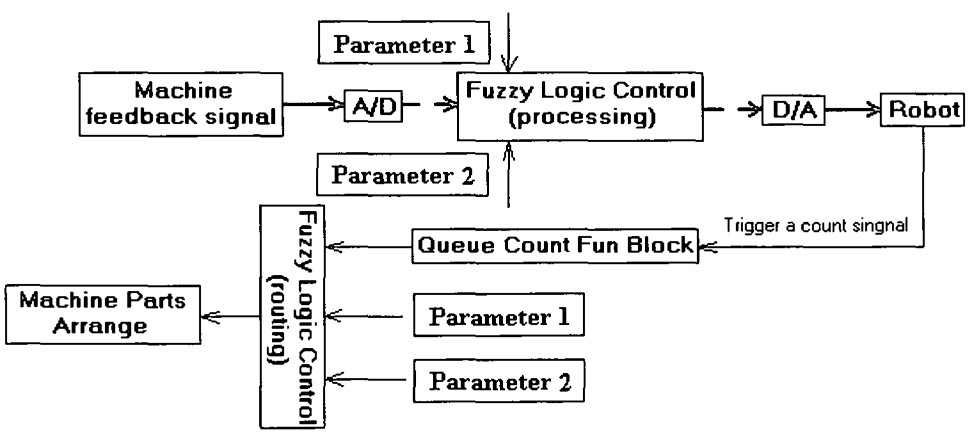


FIGURE 5-8: CONTROL CONNECTION

5.3.2 Process Function Modular Design

As discussed in modelling part, the routing process is based on the intelligent control. This project directly benefits for using MATLAB fuzzy logic control box. The MATLAB Fuzzy Interface System (FIS) is a powerful tool to edit the fuzzy rule. It is much easier than before to use this tool to design fuzzy logic control. Figure 5-9 below illustrates the MATLAB FIS editor.

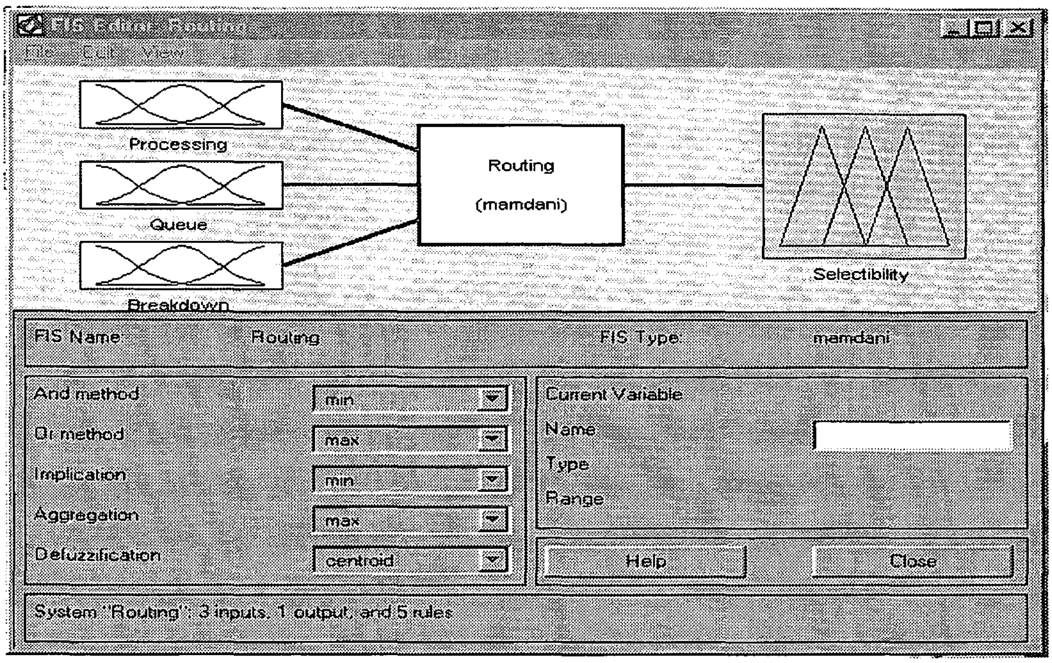


FIGURE 5-9: FIS EDITOR

It considers three inputs memberships for routing. (They are Queuing Length, Processing Time, and Breakdown Rate). It is important to realise that due to the partial matching between the input parameters and their fuzzy descriptors, more than one rule can fire at the same time. It is important to explain what deffuzification method is used by FIS.

Mamdani's fuzzy inference method is used for this FLC. Mamdani-type inference expects the output membership functions to the fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. There are five built-in defuzzification methods: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum and smallest of maximum. This FIS has adopted centroid calculation method, which returns the centre of area under the curve.

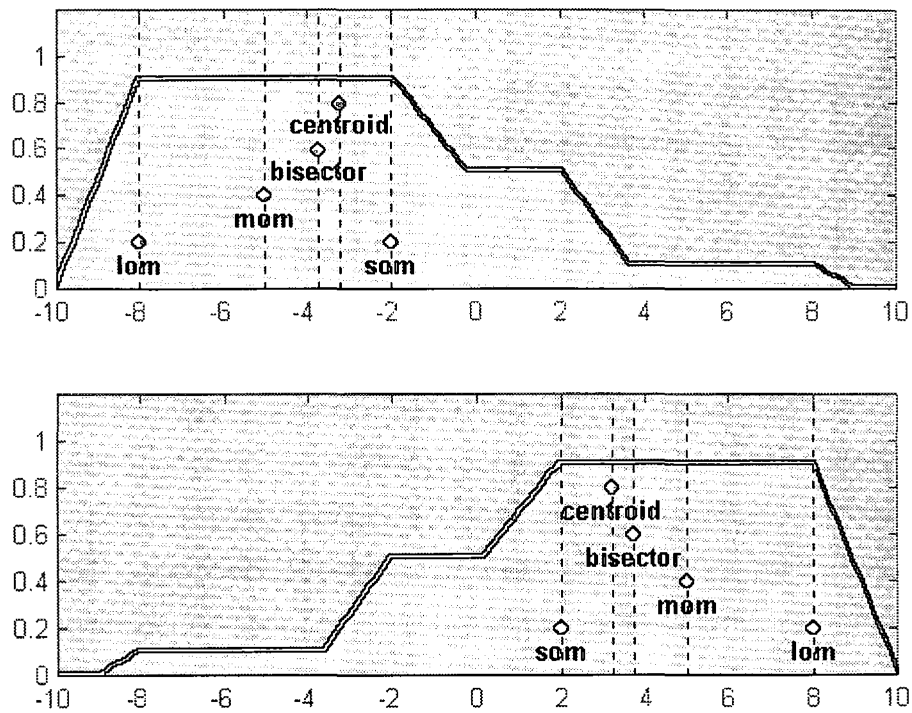


FIGURE 5-10: MAMDANI METHOD

The output membership that has defined in modelling part is important for FLC, because it represents the real management requirement. The better the decision method chose, the better the output result come out.

For this system, it focuses on the external cycle. There are two groups of data that needs to be dealt with for FIS. It needs one set FIS for Milling (Processing Time, Queuing Length, Breakdown Rate) and one set FIS for CNC lathe (Processing Time, Queuing Length, Breakdown Rate). The output is the signal for choosing Milling or Lathe. The simulation model for this project using external fuzzy logic control is designed as follows:

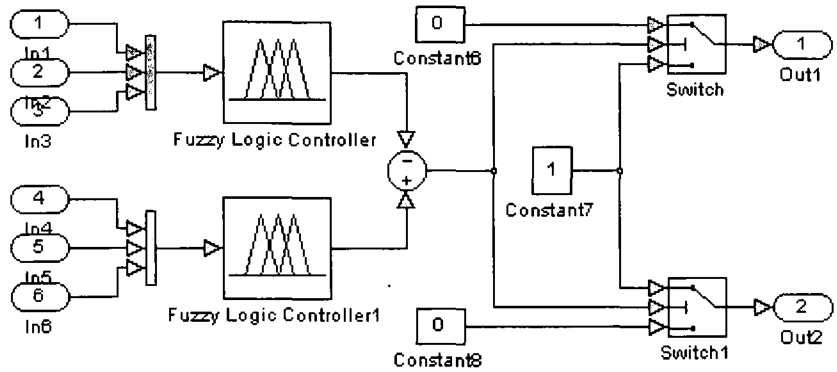


FIGURE 5-11: ROUTING MODEL

Figure 5-11 shows the comparison of the Lathe & Milling's selectibility (the picture above: Fuzzy Logic Controller is used for Lathe FIS and Fuzzy Logic Controller 1 is used for Mill FIS functions) and chooses which machine should be arranged for serving. For the Milling & Lathe's switch threshold value, it considers that milling need more attention in practice that it sets the Lathe's Switch threshold as 0.00000001. This is used to sort out the conflict if the two machines' selectibility is the same.

Actually, it is also possible to use only one Fuzzy Logic Controller to solve this problem, but when considered this will involve numerous rules (it will get ($5^6 = 15625$) rules). It uses two sets of FIS to make it simple (Each machine controller gets only 125 rules). It will make the controller easier to change and maintain. Further more, it gives a good interface between human and computer.

5.3.2.1 Fuzzy Control Rules

It can be grouped into four approaches:

- i Based on a human operator's experience.
- ii Based on a fuzzy model of a process.
- iii Based on observed operator's control actions.
- iv Based on learning. In this approach the controller improves its control capabilities by gradually developing better control rules.

The approach adopted in this implementation is a combination of method (i) and (iii), where the knowledgeable user sets the rules. The rules can later be tuned up in order to improve the system's performance (using simulation), thus reflecting somewhat a learning based approach.

The fuzzy control rules examine all the possible combinations of the fuzzy linguistic input variables. Hence the knowledge rule base contains 5 X 5 X 5 =125 rules.

An example of such a rule is given below:

IF Processing Time is Very Small

AND Queue Length is Very Long

AND Machine Breakdown Rate is Very Small

THEN Selectibility Factor is Very Large. (As shown in figure 5-12)

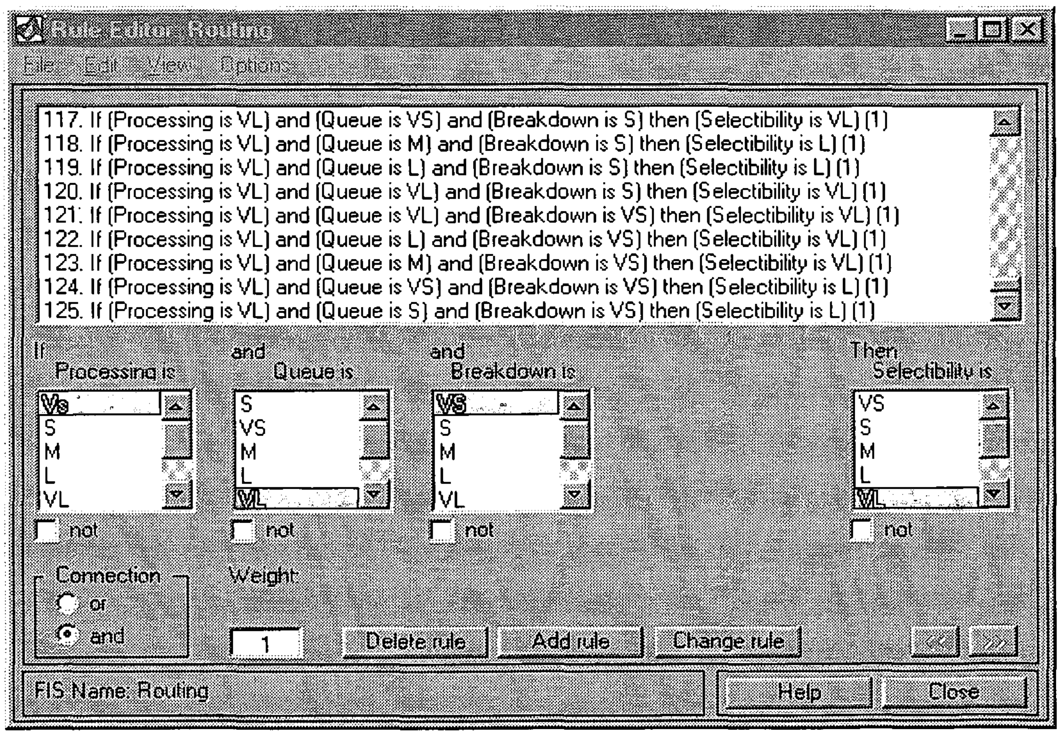


FIGURE 5-12: FUZZY RULES

The fuzzy rules are designed to represent the decision-making process. It can set up the rules for "First In, First out" control policy or "Shortest Processing Time" policy.

Combine the physical queue and the fuzzy control unit for milling and Lathe, can develop the further simulation layout shown as follows in Figure 5-13. Here the subsystem is used to simple the system layout. This model can dynamically monitor how many parts in the queuing and then send this data to management model for analysis.

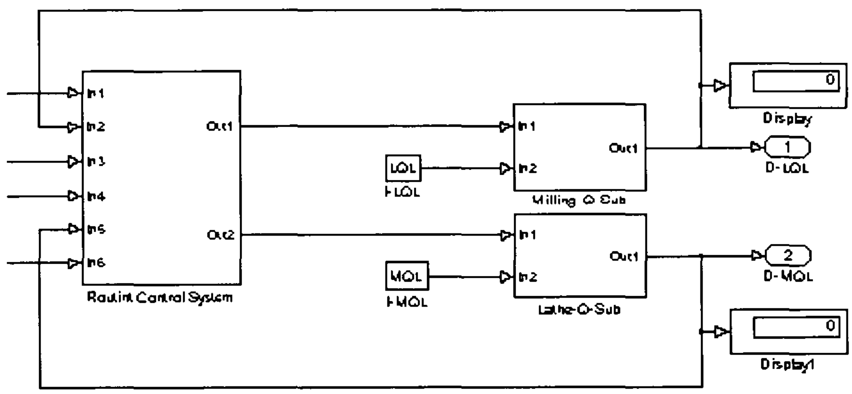


FIGURE 5-13: THE WHOLE ROUTING MODEL

5.3.3 Management Function Modular Design

This is a problem when using one robot to serve two machines, therefore a limited Source queuing model is adopted for this case. (The queue is limited)

M/M/1/m/m

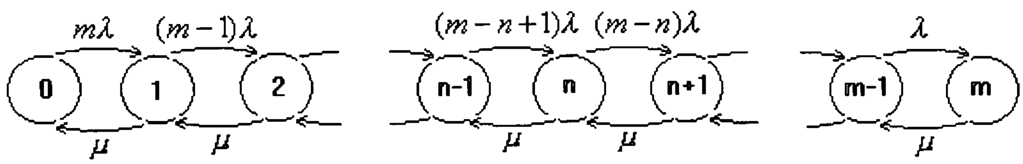


FIGURE 5-14: STATES FOR M/M/1/M/M

λ : Mean arrival rate μ : Mean service rate P_w : Probability that an arriving unit must wait for service

It can drive three equations from this state

$$\mu P_1 = m\lambda P_0$$

$$\mu P_{n+1} + (m-n+1)\lambda\ P_{n-1} = [(m-n)\lambda + \mu]P_n, 1 \le n \le m-1 \qquad (5\text{-}2)$$

$$\mu P_m = \lambda P_{m-1}$$

From this states transfer equation, it implies that,

$$P_0 = \frac{1}{\sum_{i=0}^{m} \frac{m!}{(m-i)!}\left(\frac{\lambda}{\mu}\right)^i}$$

$$P_n = \frac{m!}{(m-n)!}\left(\frac{\lambda}{\mu}\right)^n P_0 \qquad (1 \le n \le m) \qquad (5\text{-}3)$$

$$L_s = m - \frac{\mu}{\lambda}(1-P_0)$$

$$L_q = L_s - (1-P_0)$$

$$W_s = \frac{m}{\mu(1-P_0)} - \frac{1}{\lambda} \qquad (5\text{-}4)$$

$$W_q = W_s - 1/\mu$$

L_s : Mean length of the system L_q : Mean length of the queue

w_q : Mean time spent waiting in the queue w_s : Mean time spent waiting in the system

In this case, m=2 (two CNC machines). In general, CNC Milling processing time is not equal to CNC Lathe. For the analysis purpose, the assumption that those two CNC machines have the same processing time is used to check the queue preference. It can use the average processing time to represent the Queuing processing time. Based on this, it can get the simulation model layout as follows. See Figure 5-15.

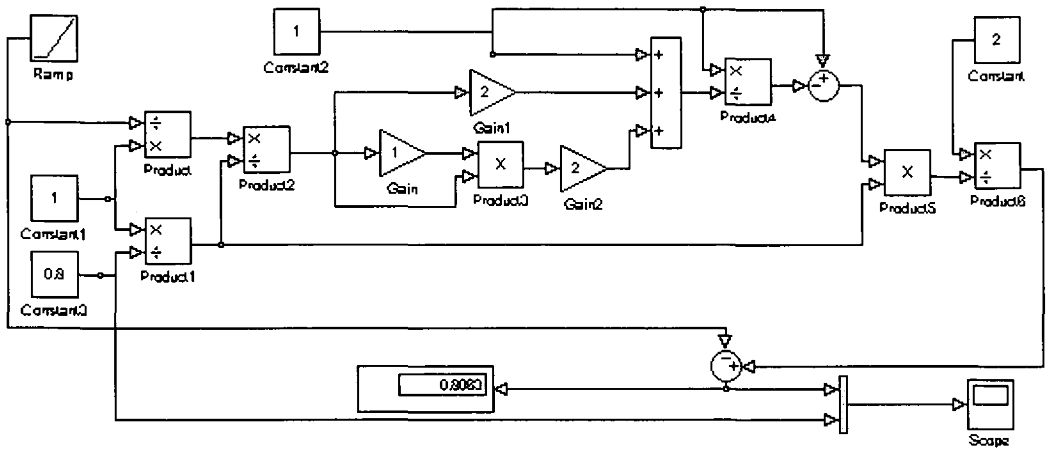


FIGURE 5-15: QUEUING MODEL

Based on this model, it can be used for management analysis. If we only consider one variable input, for example the Processing Time. Then the result is shown in figure 5-16.

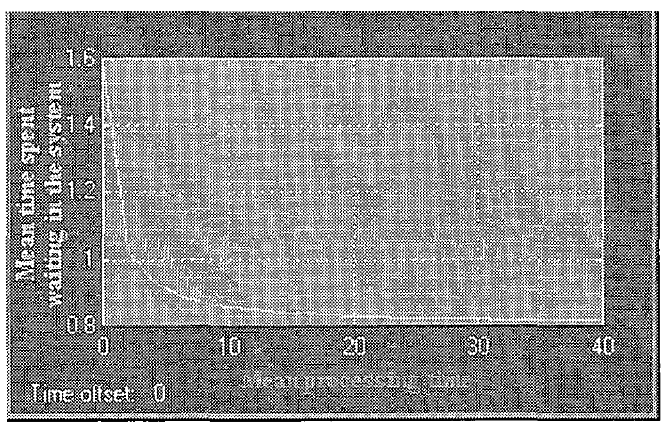


FIGURE 5-16: THE PLOT FOR QUEUING ANALYSIS

From this plot it is very clear that when the Robot process time is certain (μ =0.8), the part waiting time in a certain area use one robot to server two machines is practicable (Waiting Time w_s get a very deep down slop). It is clear that when the machine processing time is small, the use of one robot is not feasible.

5.3.4 Integrating Model

This model proposes the methods of integrating intelligent control with management system for CIM cells, and introduces the design idea, main system scheme, key technology and development process in the integrate system.

Already it has developed three basic function blocks. As stated before, it needs to use the FLC as the conjoint point for integration. The main control flow is shown in the following flow chart (Figure 5-17).

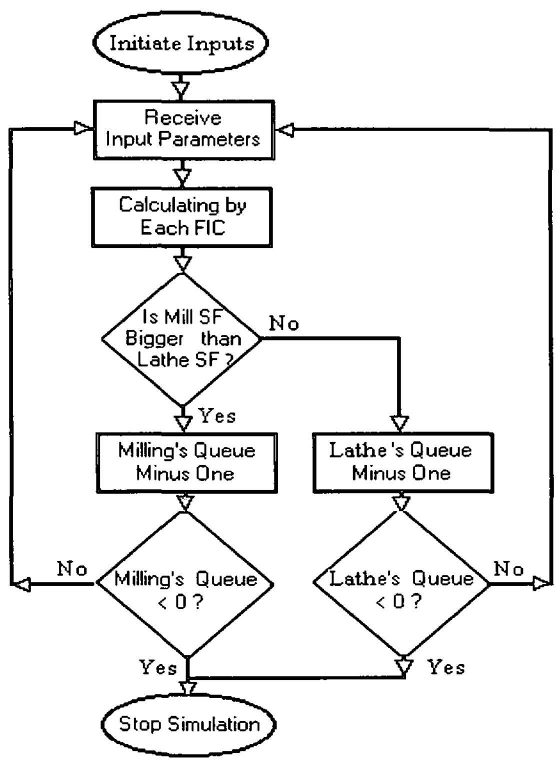


FIGURE 5-17: SIMULATION CONTROL LOOP

The initial input can import data from database. For easy evaluation, random-seeding method is adopted. The output has been translated into numbers remained in the queue, and sends this out to MATLAB workspace that can be retrieved by a management function block.

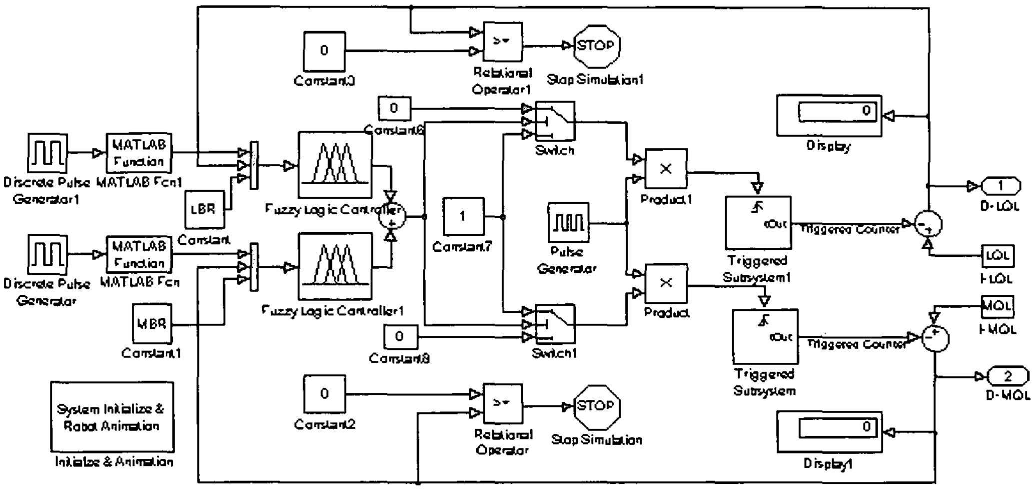


FIGURE 5-18: FINAL PROCESSING MODEL

Using this simulation model, it can simulate thousands of parts in a few seconds. The changing of FIS rules makes it very easy to test different management policy [21]. Depending on the requirement, the rules can be changed quickly to give the "What if" result.

5.4 MATLAB Model Listing

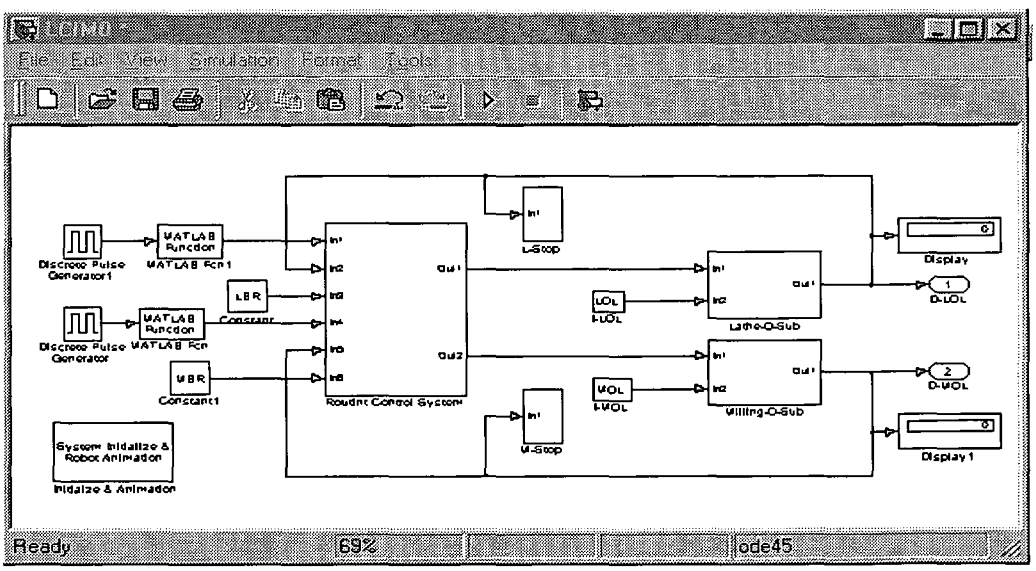


FIGURE 5-19: LCIM MODEL WITH SUB-SYSTEM

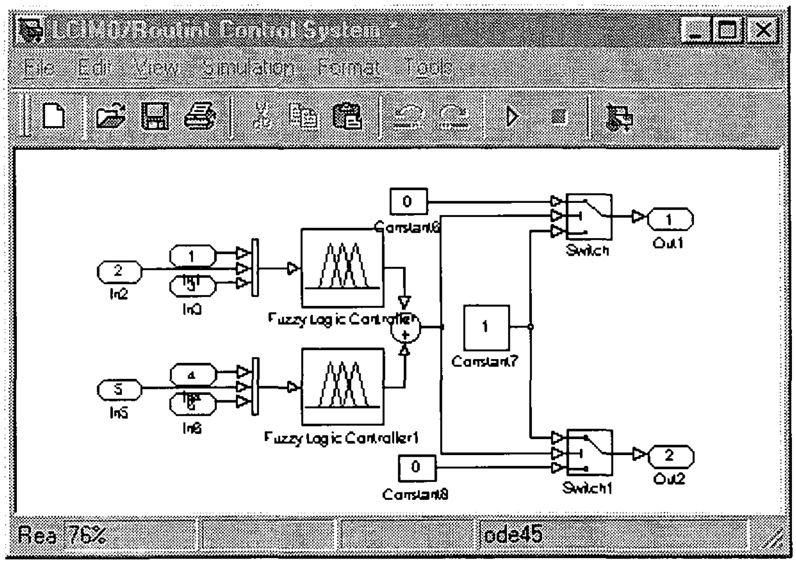


FIGURE 5-20: ROUTING CONTROLLER SUB-SYSTEM

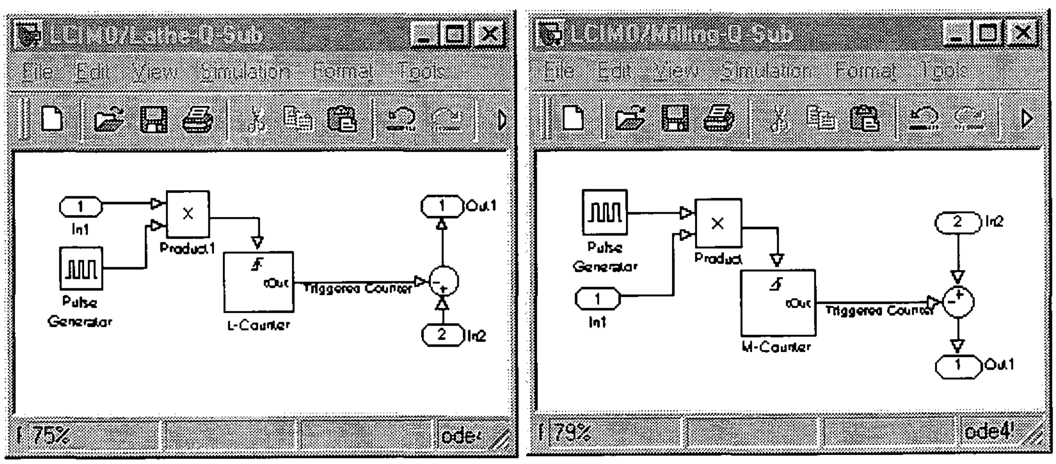


FIGURE 5-21: QUEUING SUB-SYSTEM

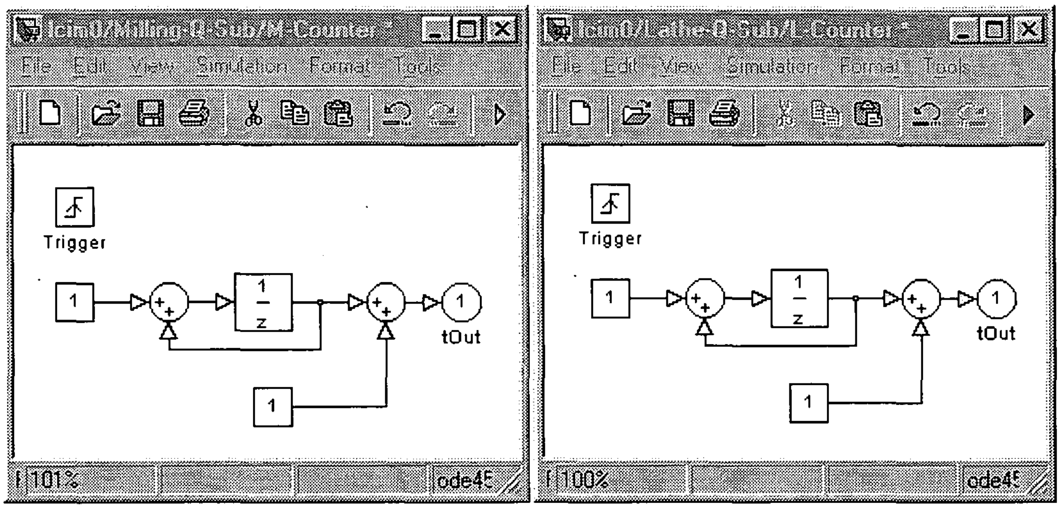


FIGURE 5-22: QUEUING COUNTER

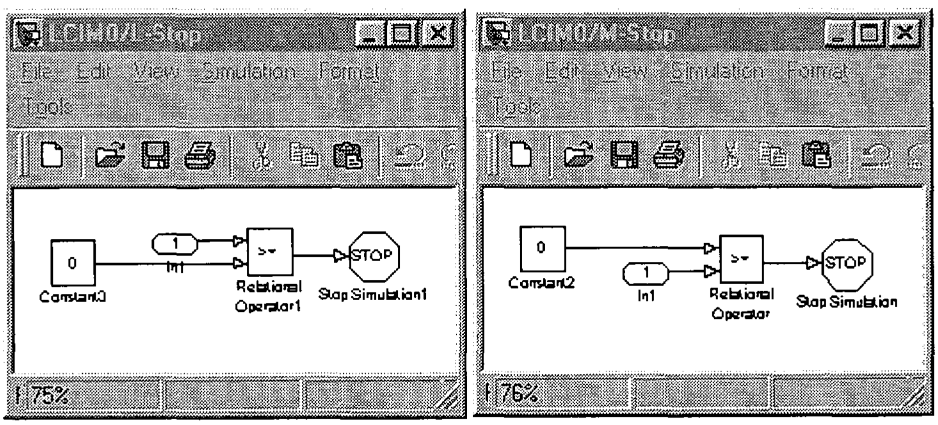


FIGURE 5-23: STOP SUB-SYSTEM

6. Graphical User Interface

GUI is very important for LCIM simulation. It gives a friendly user-system interface. In other words, it will make the system easier to understand and can improve the efficiency. This section concentrates on the GUI design for MATLAB simulation model. For real-time implementation, Database design and a GUI for Robot Controlling is needed. A brief description will be give later on the following chapter.

6.1 GUI Design

The goal for using this GUI is to provide the user editable inputs for Routing model, and give a switch to run and stop the simulation. Further more, it will provide a plot function to display the simulation result and can make the analysis become easier to comprehend. During the process of GUI design, the results, which are based on LCIM simulation parts are used.

To design GUI, the basic idea is to make it simple. Our concept for GUI design is in line of the following:

1) First of all, the GUI provides a **START SIM** button, so that when it is clicked the LCIM simulation will start; It also provide a **STOP SIM** push button to stop the simulation; the third one is the **CLOSE** button to close the GUI.

2) It needs an edit block to change the input data. It provides six editable text boxes to change the input data for **LPT, LQL, LPT, MPT, MQL, and MBR**. When it is changed, it will automatically transfer to Simulation model when the **START SIM** button is pressed. Trough this way it makes it flexible for testing. There are six static texts to indicate the usage of those editable text boxes, respectively;

3) It needs to display the dynamic changes for processing, so that needs add axes for plotting the figure to show this, such as to show when the lathe process is activated

and when the milling process is invoked. This gives us the idea of how these models operate. It uses one Push Button 'PLOT' to make this function.

4) A pop up box needs to be added to give the choice for processing principles, such as 'First **Come, First Severed**' and so on. This determines which fuzzy rule should be used.

5) Finally, it uses one Text to ensure the parameters are correct. The original display is "Parameters are greater than Zero". If any is below zero, it will display "All parameters must be greater than zero".

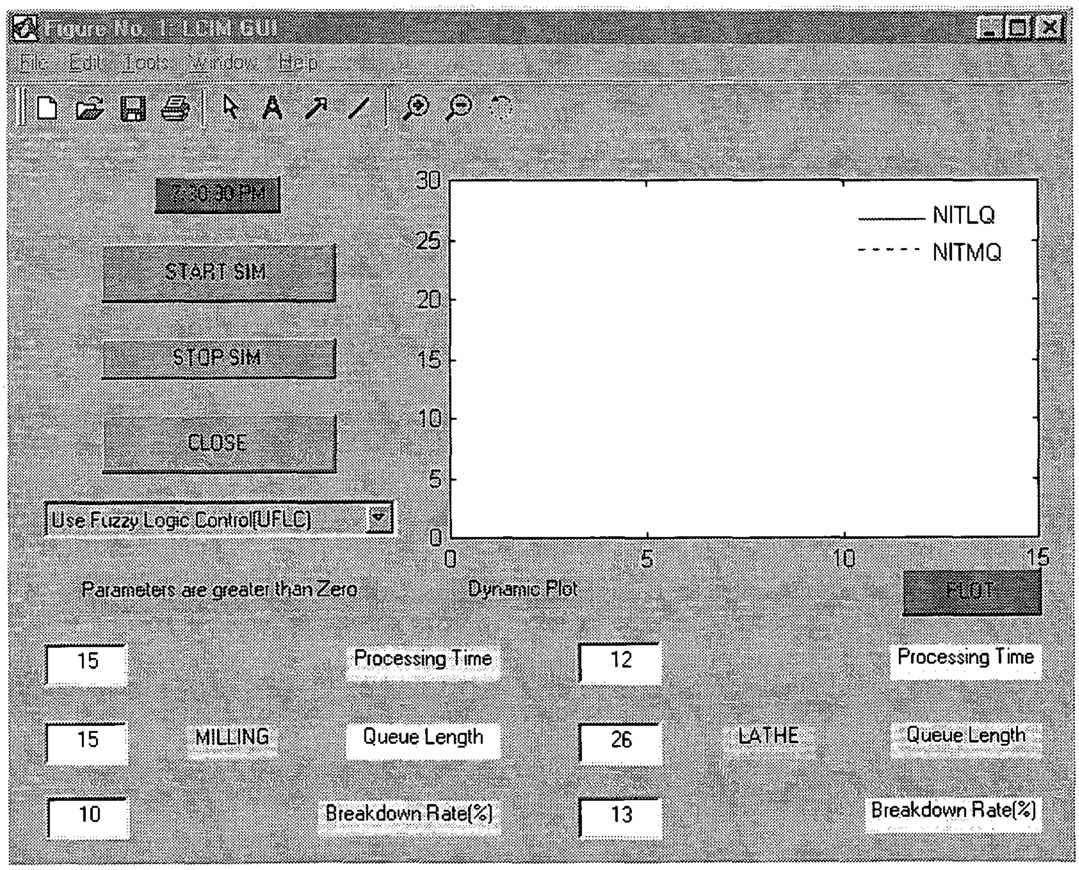


FIGURE 6-1: THE APPEARANCE OF GUI

Up to now, simulation GUI has been designed as shown in figure 6-1

The **STOP SIM** and **PLOT** buttons can be set to light color when it is disabled (set its Enable property to 'off') to avoid an error result which is caused by clicking the **STOP SIM** before **START SIM** button is clicked. Only after the **START SIM** button gets pushed, the **STOP SIM & PLOT** button gets activated. Thus when we push the **PLOT**, it displays the right plot for the simulation.

The following figure is one that is in use.

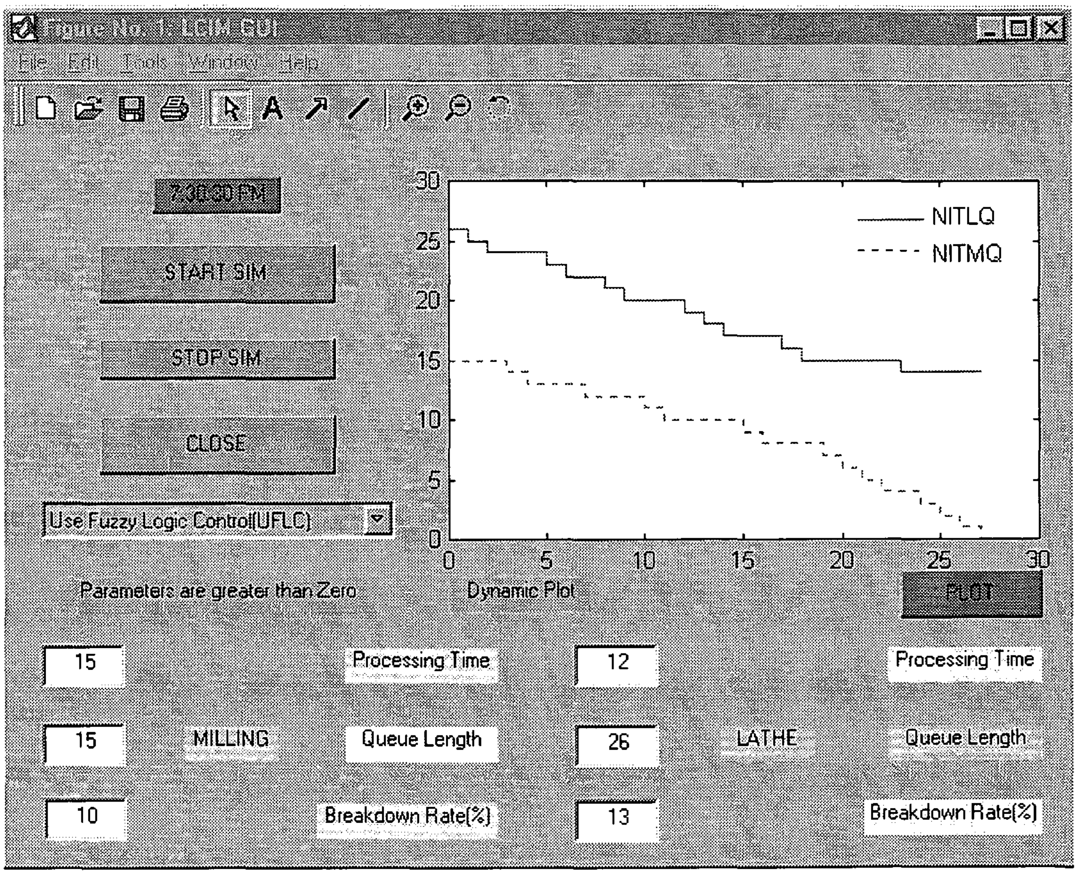


FIGURE 6-2: REAL-TIME GUI PLOT

6.2 Callbacks

Now, we need to manage what happens when a button is depressed, or when we type in editable text boxes. For this we need to make callbacks for each graphical handle.

i. When one clicks on the **START SIM** button, it starts running the LCIM simulation model. It uses the initial data for the beginning if it has not changed the editor box data. If the input data is changed before Push **START SIM** button, it will use the new data for simulation.

ii. The six input editable text boxes are used to change the simulation parameters. It works when the **START SIM** button is pressed. It will automatically put the Variable value into the workspace thus change the simulation parameters.

iii. Anytime when the **STOP SIM** button is pushed, it will halt the simulation process. If the **PLOT** button is pushed, it will plot the dynamic line for NITLQ & NITMQ changes. It will display the process until the simulation is stopped (when the **STOP SIM** is pushed).

iv. The **PLOT** button will plot the NITLQ & NITMQ dynamic changes for the simulation.

v. When clicking the **CLOSE** button, the window closes.

To speed up the GUI, it uses Function Callbacks. Instead of giving long strings of commands to callbacks, it gives short function names to each callback. The long strings of commands are in the function file. For example, the function START is the callback of **START SIM** button. It also makes the GUI more robustness, and the ease of coding causes it to execute in the function's workspace, as opposed to the callback string, which gets executed directly in the base workspace where it can collide with other data.

```
Function START

if strcmp(get_param('LCIM','simulationstatus'),'stopped')
  set_param('LCIM','simulationcommand','start')
 return
end

% Active STOP SIM & PLOT button
StHndl=findobj(gcbf,'String','STOP SIM');
Set (StHndl,'Enable','on');
PtHndl=findobj(gcbf,'String','PLOT');
Set (PtHndl,'Enable','on')
```

Then send all of the callbacks to one single function named LCIMfun.m. We use switchyard programming to prevent function proliferation.

```
Function LCIMfun (action)

Switch (action)
Case 'start'
.
.
.
Case 'plot'
.
.
.
```

The whole function LCIMfun.m is available in Appendix.

7. Simulation Testing

In order to assess the decision quality of the FLC, a simulation study was conducted. In the simulation experiments, three routing heuristics were illustrated:

1) User Designed Rule (UDR)-here refer to Use Fuzzy Logic Control (UFLC).
2) Short Processing Time (SPT).
3) Shortest Queuing Length (SQL).

As stated before, it can use the FIS rule to realize the management policies. It all depends on how one builds the rule sets.

7.1 Rule Set Design & Simulation

When using the UDR for control, the surface view of UDR rule sets is as per the following figure.

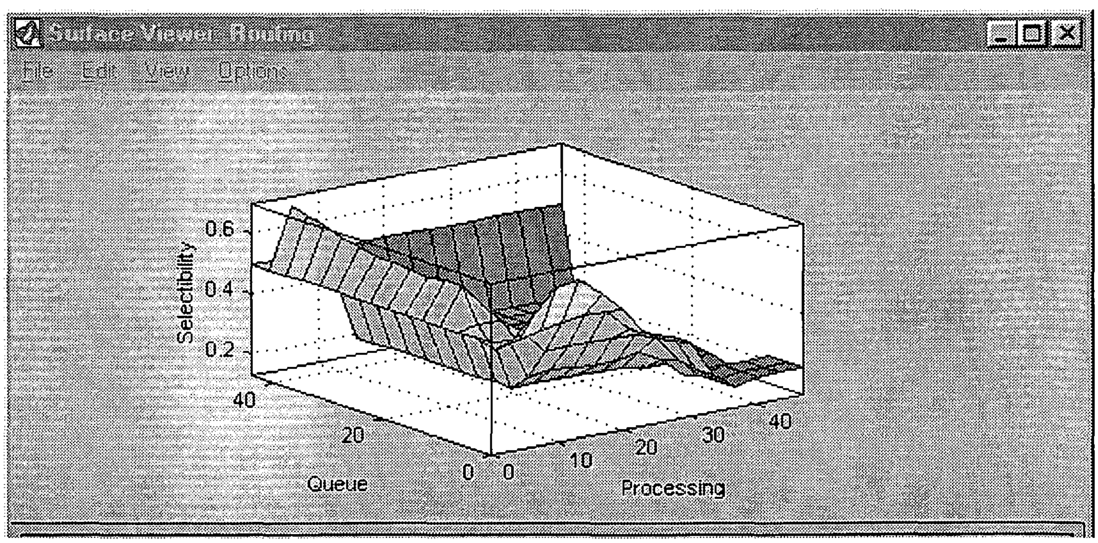


FIGURE 7-1: UDR SURFACE VIEW

Running the simulation, the following plot is generated,

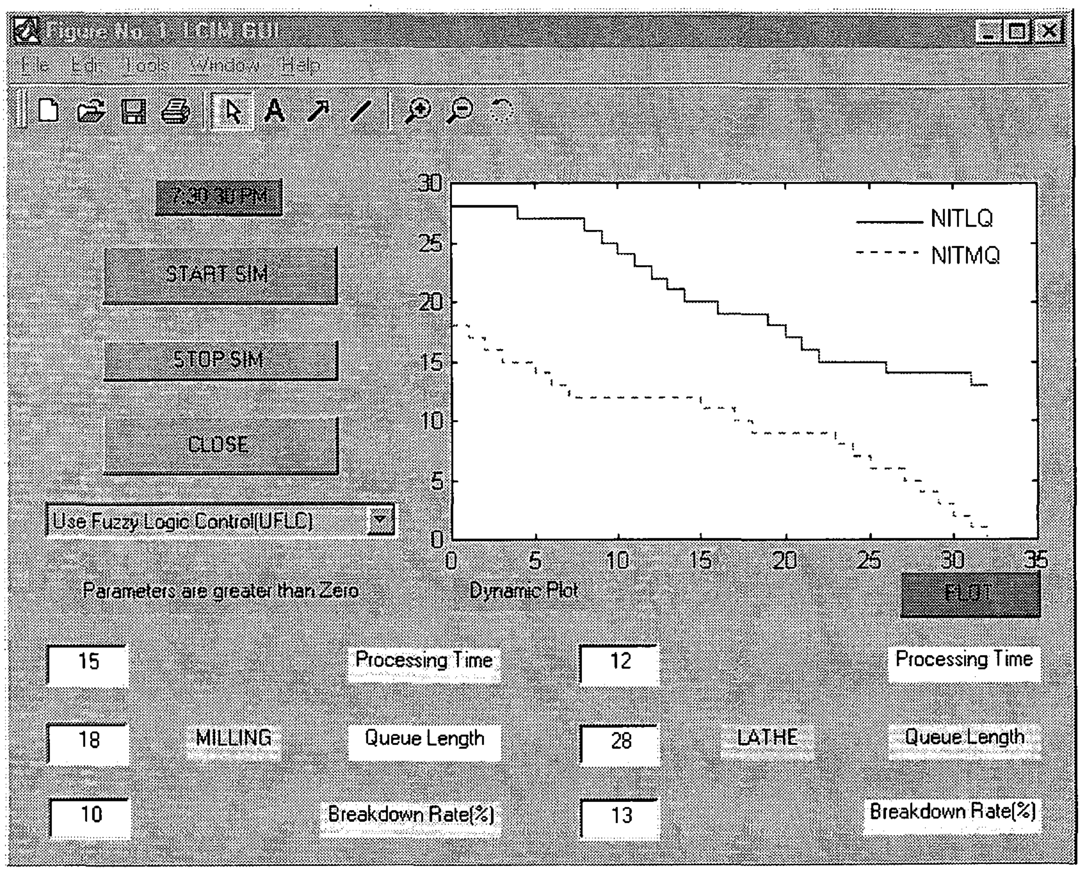


FIGURE 7-2: UDR SIMULATION RESULT

As shown in this plot, the lathe numbers in the queue remains no changes at first. After three milling parts have been served (number in the queue reach 15), the robot serve one part for lathe. Using the input parameter condition (as shown above) to control, we can get a balance distribution for two machines.

Using the MATLAB FIS rule editor, The FLC FIS block's parameter only needs to be changed to derive the SPT rule sets,

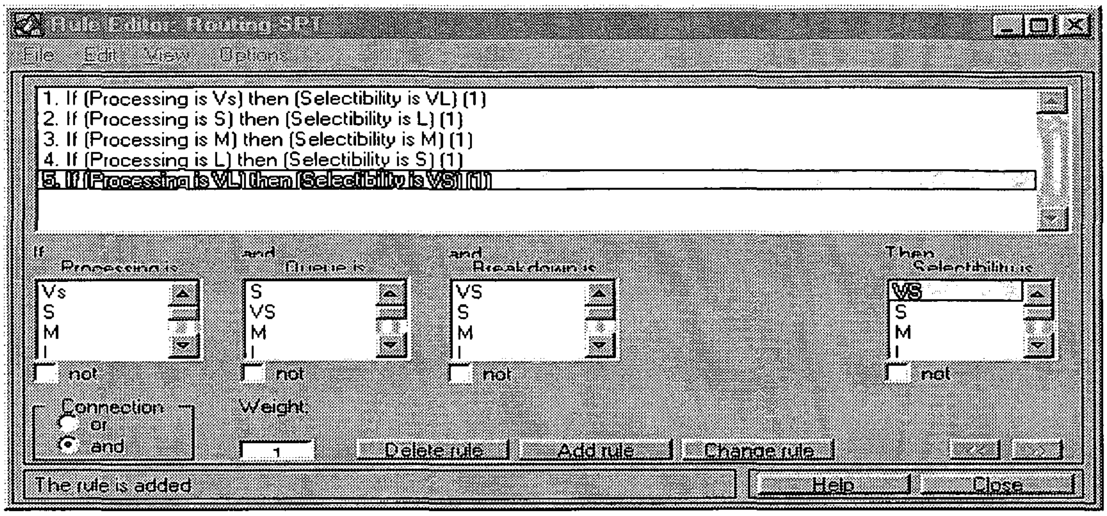


FIGURE 7-3: SPT RULE SETS

The surface plot for SPT is as follows (*N.B. the Selectibility is still the combine result for Processing Time, Queuing Length and Breakdown Rate*). It can be seen from the surface view that along with Processing Time increasing, the Selectibility is decreased.

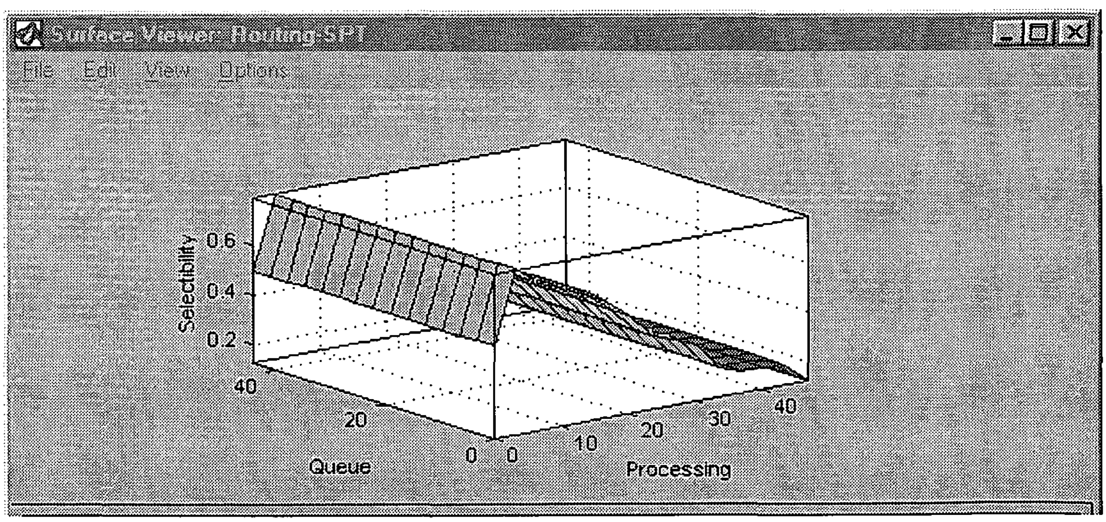


FIGURE 7-4: SPT SURFACE VIEW

The processing time is assumed as Poisson distribution. One of the simulation results is depicted as follows,

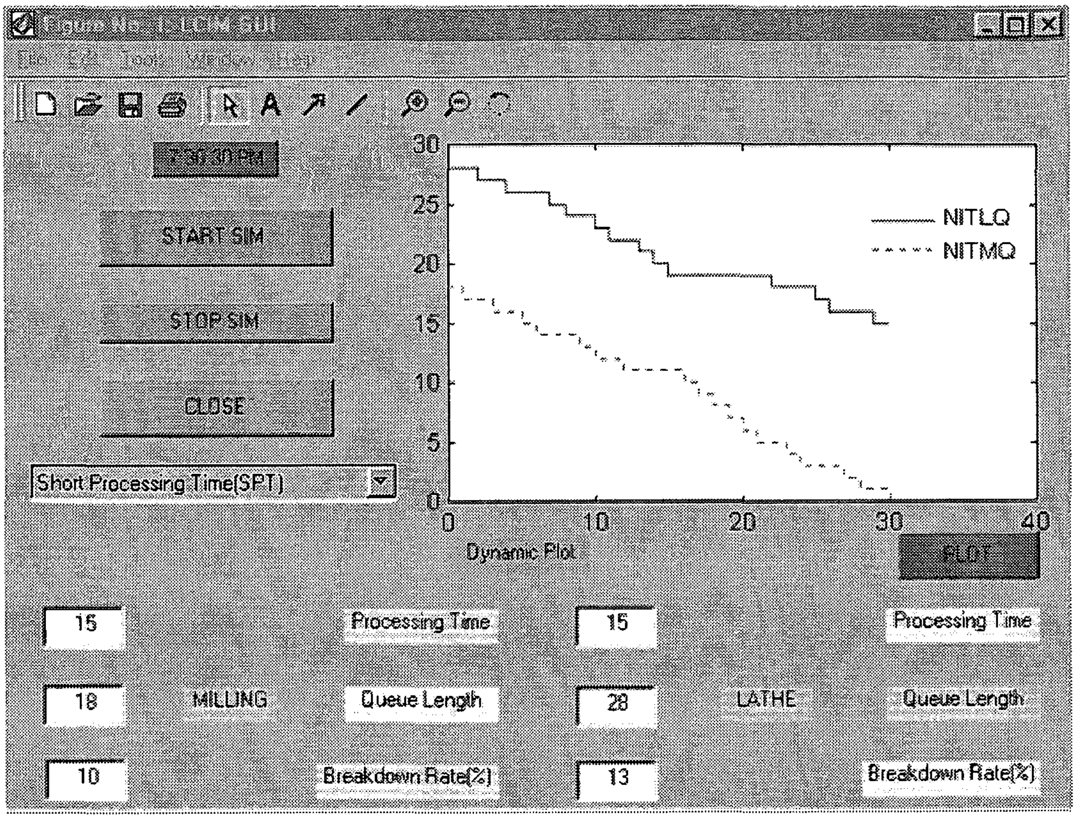


FIGURE 7-5: SPT SIMULATION RESULT

When using SPT rule, the FLC becomes sensitive for Processing Time varying therefore a special FLC rule needs to be designed for further use. As compared to SQL rule, it needs to make a change for the input. The order for CNC queue is in Poisson distribution. Given a small modification for LCIM model (i.e. change the input, and applying the SQL rule for FLC), can be used for processing again. It means that the FLC is especially advantageous when the inter arrival rate of parts is high and the system is congested.

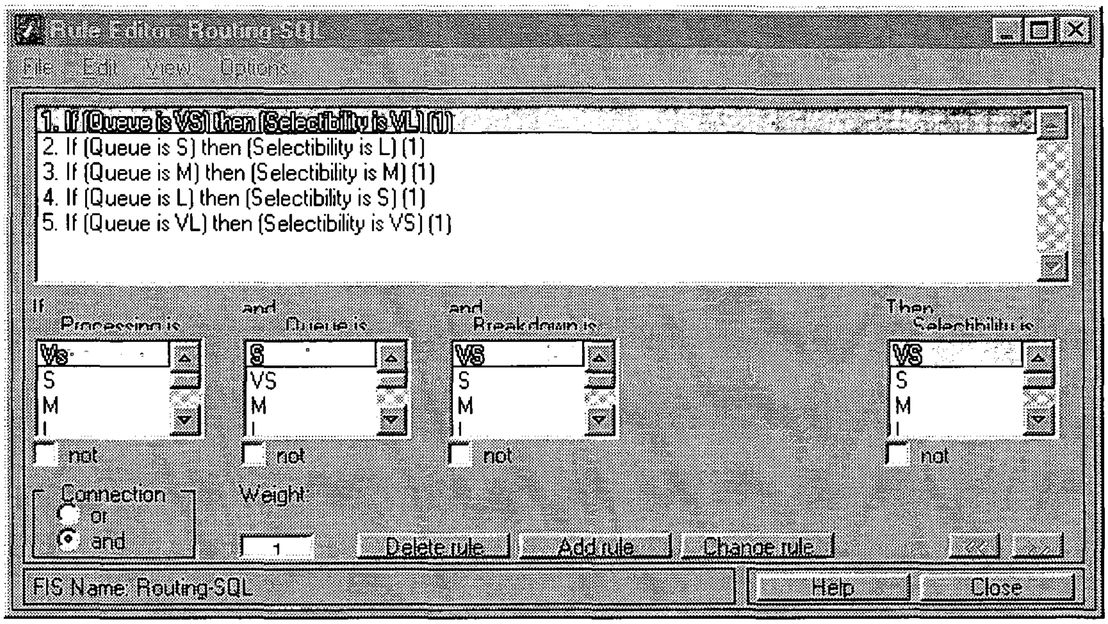


FIGURE 7-6: SQL RULE SETS

If the former Selectibility formula is used, then the result of the SQL surface view as shown in Figure 7-7 below,

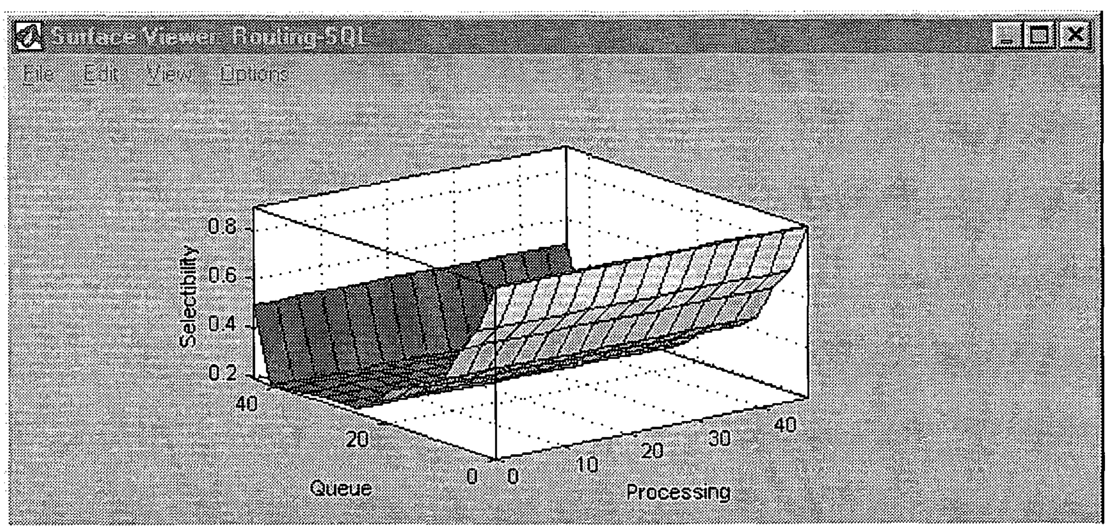


FIGURE 7-7: SQL SURFACE VIEW

It is clear that this model can easily recognize the real manufacturing process. It also has a good communication interface with management and production, and the process is depicted. Based on this platform, the whole picture of LCIM environment is realized.

7.2 The Real Time Implementation

Real Time Workshop is a function of Simulink. This function can produce codes directly from Simulink models, and build programs automatically. The generated program can be run in a variety of environments, including real-time systems and stand-alone simulations. With the Real-Time-Workshop, LCIM Simulink model can run in real-time on a remote processor.

To use the Real-Time workshop, the environment to place the generated code needs to known. This is known as targeting, the environment itself is called the target. The host is the place where you run MATLAB, Simulink, and the Real-Time Workshop. Using the build tools on the host, it can create codes and an executable that runs on the target system.

All these features make it much easier than before to implement the real time control. But considering the software costing, alternative ways to test the result's applicability should be used. The implementation is divided into two parts. One is using the simulation output signal to control the Robot-Lathe process. This will illustrate the physical control connection with simulation model. The other part is designed to enhance this simulation model, which is building the input and output channel with outside database.

7.2.1 Robot Loading Control Illustration

The simulation model that has been developed uses one set of data and is verified by random seeding. The output signal has been translated to counter number and plotted in the GUI. In practice, it needs to setup the communication between MATLAB and outside database application. Microsoft Access is used as the external database container.

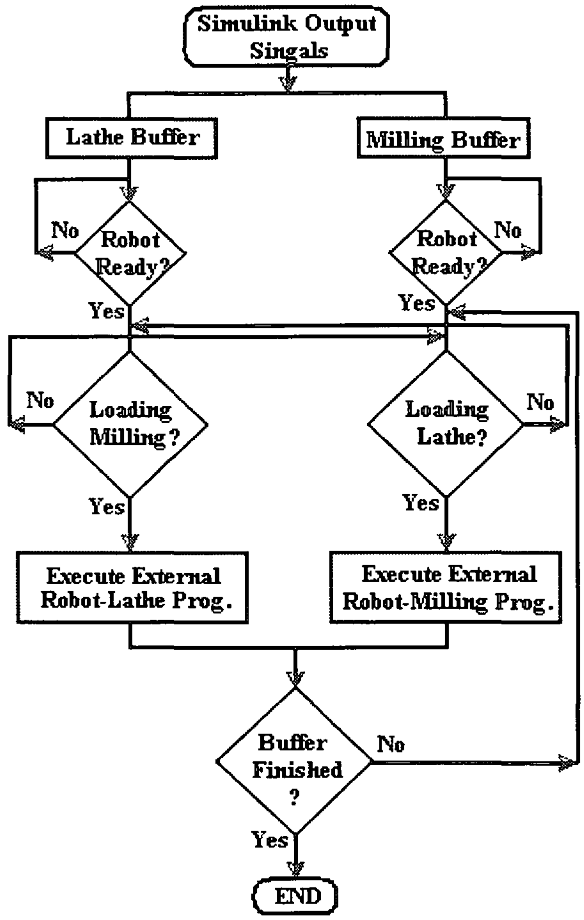


FIGURE 7-8: ROBOT CONTROL LOOP

The Robot Ready signal can be recognized by either captures the robot sensor signals or retrieving data from Robot Control Shell.

The method to retrieve data from Robot Shell is adopted here (Walli3---Workcell Amalgamated Logical Linguistic Instructions, Version 3). Trough setting up the linkage between Robot-Shell and Robot-Monitor database (robControl.xls), the robot position becomes available for control programming.

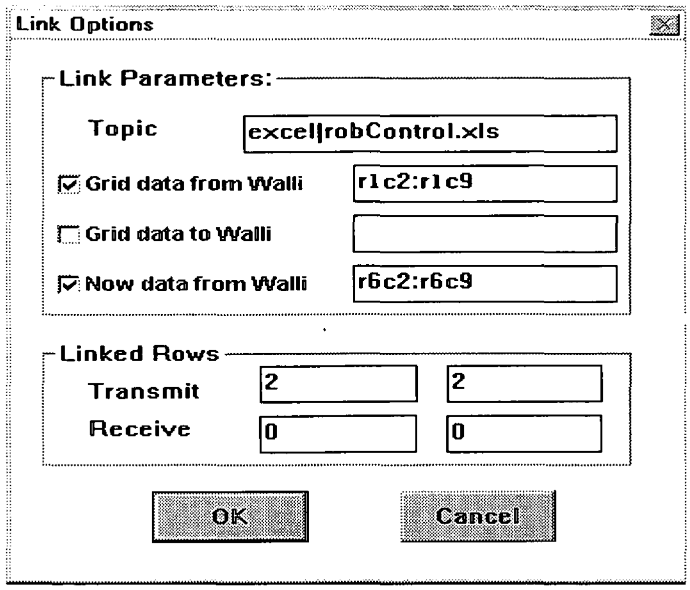


FIGURE 7-9: ROBOT-EXCEL LINKAGE

The detail data communicated relations are illustrated as follows:

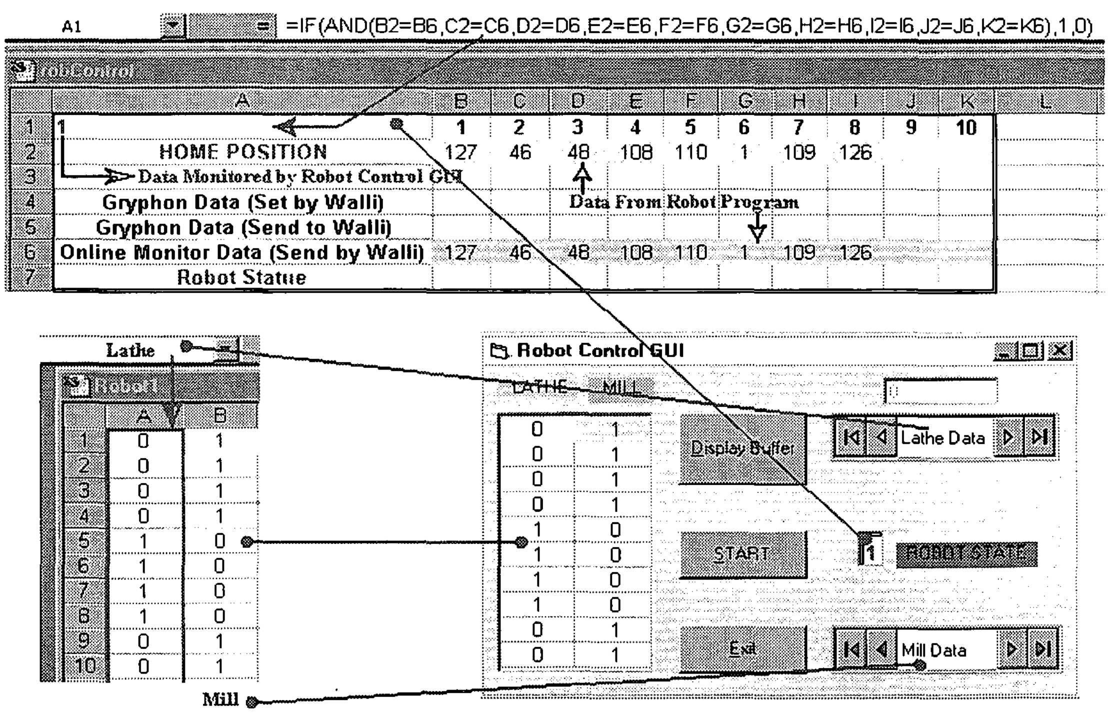


FIGURE 7-10: ROBOT CONTROL DATA LINKAGE

7.2.1.1 Programming Gryphon

It is via Robot Shell (Walli3) to communicate with robot control box. The on-line state is true if the software detects that the robot is connected correctly to host computer through the RJ45 serial form of interface.

This project is automatically controlled for CIM cell, which means it needs to set up the communication channel between Host computer, Robot and CNC machines. The robot's control box only has 16-pin port to access outside. This can be used to set up communication with one CNC machine. The detail connection for RI45 Board and Robot Control Box's is:

Table 7-1: Cell Definition

GRYPHON	RJ45 Broad-Device 2
Digital Sensor 1	Output to GRYPHON Digital Input 6
Digital Sensor 2	Output to GRYPHON Digital Input 6
Digital Sensor 3	Output to GRYPHON Digital Input 3
Digital Sensor 4	Output to GRYPHON Digital Input 4
Digital Sensor 5	Output to GRYPHON Digital Input 5
Digital Sensor 6	Output to GRYPHON Digital Input 6
Digital Sensor 7	Output to GRYPHON Digital Input 7
Digital Sensor 8	Output to GRYPHON Digital Input 8
Analog Sensor 1	Output to GRYPHON Analog Input 1
Analog Sensor 2	Output to GRYPHON Analog Input 2
Analog Sensor 3	Output to GRYPHON Analog Input 1

The detail definition for Gryphon Robot Control Box 16 pin port is as shown as follow:

Table 7-2: GRYPHON Robot Connected to PC TURN 120—Interface Definition

Input Pin	Representation
1	Machine Ready (Not used)
2	Door Open
3	Chuck (collect) Open
4	Chunk Close
5	Door Close
6	Program Stopped (M30)
7	Not Connected
8	Not Connected

Output Pin	Representation
1	Open Door (Active Low)
2	Open Chunk
3	Close Chunk
4	Close Door (Active Low)
5	Cycle Start
6	Agreement Switch (Active Low)
7	Not Connected
8	Not Connected

7.2.1.2 Robot Control GUI Functions

This Control GUI is based on Virtual Basic. Before assigning the function to the control command button, it needs to run all the background applications. It uses **Robot.bat** script to execute the Robot-Lathe program and dynamically monitoring the robot's position via **robControl.xls**, and the process buffer data is retrieved from **robot1.xls** (Detail data linkage can check figure 7-10).

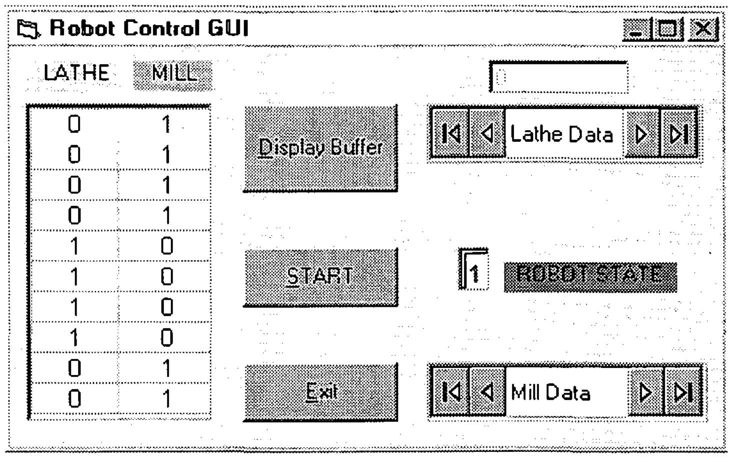


FIGURE 7-11: ROBOT CONTROL GUI

If the **Start** Button is clicked, it first checks whether the robot is ready (It is designed when it is back to **Home** position, then it is ready). If it is ready, then it checks the process is for the lathe or not (If it is the lathe process, the Buffer value is 1. Otherwise it is 0). If it is the lathe process, it executes the Robot-Lathe process. Due to it only setting up the control between robot and lathe, the GUI checks the entire index in the lathe Buffer that is located in robot1.xls, and the Buffer data is retrieved from MATLAB workspace.

7.2.2 Input & output Database Structuring

Before setting up the communication between simulation model and outside database, it is important to understand MATLAB and external data structure. MATLAB communicates data with outside via its workspace. It can set up the channel through dynamic database exchange programming.

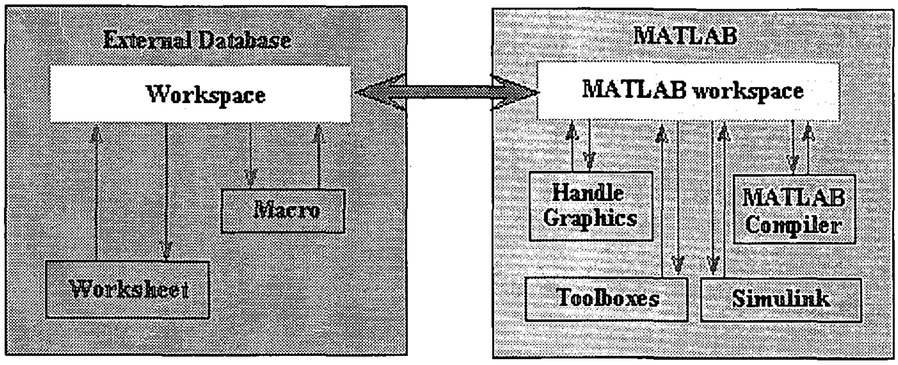


FIGURE 7-12: DATA EXCHANGING

This simulation model, can retrieve input data from outside database, and send the routing arrangement result out. It uses one database, named LCIMdb, and gets two basic table-sheets inside, namely **LCIMinput, LCIMout**.

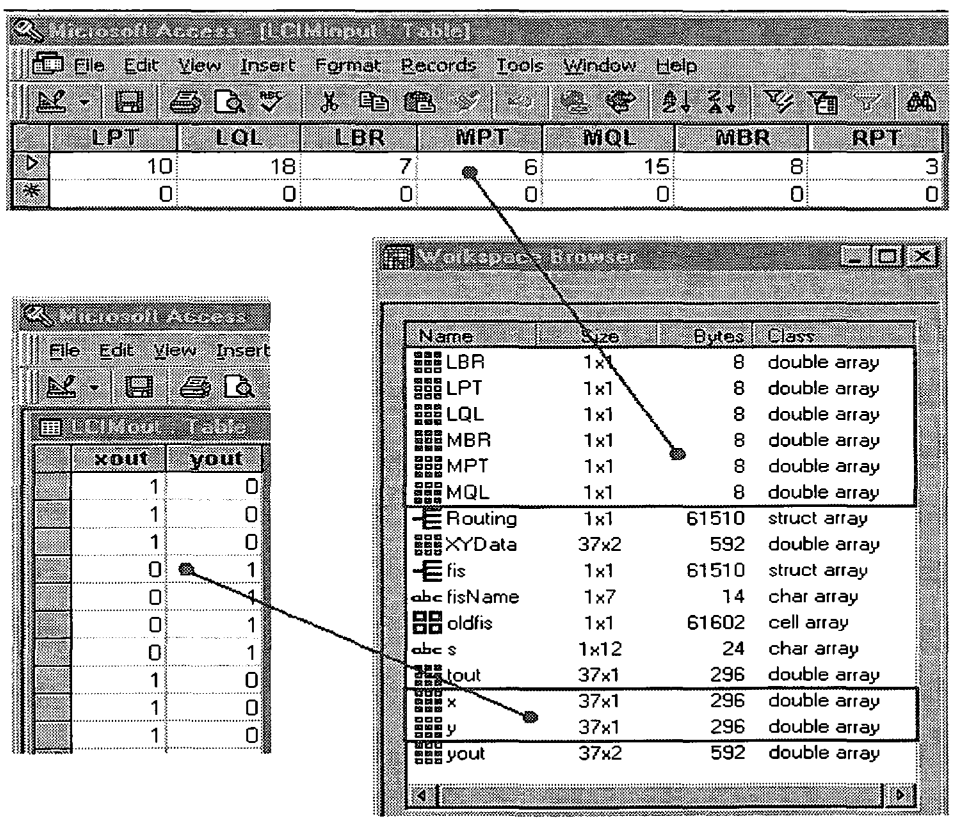


FIGURE 7-13: DATA COMMUNICATION

8. Discussion

There are numerous logistics arrangement software available in the market and the system analysis package as well. Normally they provide consulting, software solutions and support services. One such type consulting software is PRAGMA (www.pragma.co.za). The others provide sophistical software packages, such as BHR software (www.bhrsoftware.com). Compare with the special problem investigation approach and commercial software approach, this research provides a cost-saving approach. The special problem investigation normally is an ad-hoc process. The commercial packages normally only can provide limited information and the design is not precisely to the problem which is need investigation. This research has shown that can based on some popular software, and following correct analysis methodologies, it can explore some manufacturing problem and get the analysis results that are comparable with commercial software.

This research only provides a way to model and simulate LCIM cell. The details of the techniques used still need further exploration. For example, based on the testing results from Figure 7-2, when a robot is loading the 30^{th} part, the plot shows this simulation model triggered lathe and milling loading in the same time. It indicates that it needs more suitable techniques to distinguish the FIS output to avoid such conflicts (Although the chance that it gets the same selectibility is small). However, when the simpler rule is adopted, the chance for such conflict is much lower (shown from Figure 7-5).

The conceptual modeling method is used in this project. It classified the manufacturing functions into three basic groups, but in the real word it is difficult get a clear classification. This kind of design is not as flexible as other methods, such as vertically company design method [16]. The feasible methods for expand such module design concept for enterprise complexity problem analysis needs further investigation also.

This is only one step towards LCIM. The structure and methodology is still open for discussion.

9. Conclusion

In this study, a structure for LCIM was developed. Based on this structure, the LCIM modeling and simulation was conducted. It was proved that the LCIM structure is flexible and practicable. This was due to it being an open system which provided easy integration. The know-how for system design and intelligent control contribute a lot for this project.

The basic LCIM concept comes from LM and AM. According to the core of AM (i.e. integrating and corporation), it demands that LCIM must be open and should be viewed as a system. Considering LM, the dynamic support scheme is important for current CIM systems. Combining these two trends, it sets up the LCIM structure.

LCIM operations exhibit many system characteristics, and should be considered as such. This is the major reason why it is now felt that a systems approach should be viewed as a more adequate framework for the analysis of problems that are generated by current LCIM operations. The most effective manufacturing operations always have clearly defined objectives. While the setting of these objectives can be a difficult and unstructured problem, the actual process of systems design which must create a system capable of fulfilling these objectives are often a structured problem to which a relatively hard systems approach is suited. LCIM is started from a system view in order to locate the CIM cell. It therefore could meet some of the requirements for current advanced manufacturing.

LCIM also approaches problems that are currently experienced in CIM. Using the new LCIM frame, several simulation models have been developed. This demonstrates that it is possible to integrate management, process control and physical functions into one cell, provided that the correct tools and technologies are used, which will result in this process becoming much more easier than before.

1. Advantages of the new LCIM concept

LCIM is a new concept in manufacturing and is a system response for ML & AM. This brings new ideas for CIM design, both in practice and research. The simulation model based on this structure seems flexible and practicable; hence it has included the advanced technologies, such as Fuzzy Logic, Neural Network and Intelligent Control. What has been tested here will provide a guideline for further practical applications.

LCIM offers the designer a complete framework for the development of a modern CIM system. It is an open system, using spiral evaluation and developing methods, which can develop continuously.

2. Limitations

Firstly, the deffuzification method has not been explored in detail. The deffuzification method is the core for FLC. Development thereof must be based on former experience and data, which will influence the initial estimations. An efficient deffuzification method therefore involves a great deal of mathematical calculations that needs a developing method to guide this process and would make LCIM development easier.

Secondly, this model still lacks some logistic concepts. Now, the spiral support development system is still in conceptual design and modeling of this is still a problem. Also, the management function needs more enhancement and refinement.

Finally, LCIM should be approached from the viewpoint of a responsible manager who is actually involved in the operation of the system concerned; in the meanwhile it also needs feedback from the technical specialist. LCIM is needed for certain enhanced methodologies which will draw together more of the important considerations to the designer, which will be useful in real-life business situations, rather than for the purposes of theoretical research units alone.

10. Recommendations for Further Work

Further work based on this LCIM structure can be divided into two main directions. One is in line with the lab's LCIM facilities implementations. For example, it can physically connect in a One-Robot-Two-Machines cell. However, the current RJ45 interface board limits this implementation, since this board can only supply 8 inputs and 8 outputs at present. Therefore, only basic activities can be accomplished. The recommended control switches can be as follow:

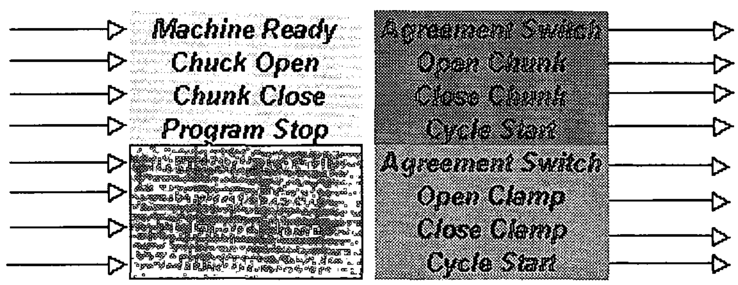
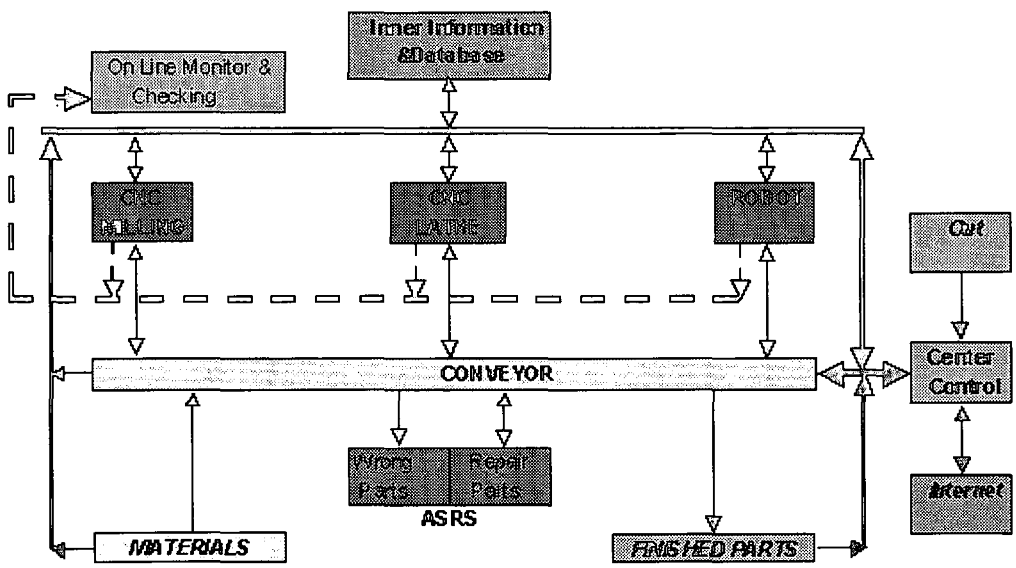


FIGURE 9-1: STAGE ONE

This can also be expanded horizontally to add an ASRS and an online Monitoring system. (These projects are currently under development in the AMT laboratory)



CIMCELL

FIGURE 9-2: STAGE TWO

The other direction is focused on manufacturing management. The LCIM model can be used to analyse the management requirement in different manufacturing situations, e.g. (based on certain assumptions) the different rule selection can be evaluated. The model could then be used for the analysis of the above in order to check the inner logistics, for example inventory control. However, the macro logistics concept still needs further investigation.

11. References

[1] Altintas, Y., Newell, N., and Ito, M., Modular CNC Design for Intelligent Machining, Part 1: Design of a Hierarchical Motion Control Module for CNC Machine Tools, Part 2: Modular Integration of Sensor Based Milling Process Monitoring and Control Tasks, Journal of Manufacturing Science and Engineering, Vol.118. Pp.506-521, 1996

[2] Arsham, H., System Simulation: The Shortest Path from Learning to Applications, http://ubmail.uboilt.edu/~harsham/simulation/sim.htm

[3] Benjaafar, S. and D. Tirupati, "Recent Advances in Manufacturing Flexibility", *International Journal of Flexible Manufacturing Systems*, (invited editorial for special issue), **Vol. 10, No. 4**, pp. 323-325, 1998.

[4] Charles Anderson & Peter Bunce, Next Generation Manufacturing System, http://www.cam-i.org)

[5] Choi, S. and Kim, J. ,A Study on the Measurement of Comprehensive Flexibility in Manufacturing Systems, *Computers and Industrial Engineering*, **Vol.34, No.1**. Pp.103-118, 1998

[6] Drake, Glenn R., Smith , Jeffrey S., and Peters, Brett A., An Overview of CIM Enterprise Modeling Methodologies, *http://www.okstate.edu/cocim*

[7] Drake, Glenn R., Smith, Jeffrey S., and Peters, Brett A., Simulation as a Planning and Scheduling Tool for Flexible Manufacturing Systems, 2000, *http://www.okstate.edu/cocim*

[8] Dudin A. N. and Klimenok, V. I., A BMAP/G/1 Queueing System with Alternating Operation Model, *Journal of Automation and Remote Control*, **Vol. 60**, pp.1445-1452, 1999

[9] Ehmann, K. F., Kapoor, S. G., Devor, R. E. and Lazoglu, I., Machining Process Modelling: A Review, *Journal of Manufacturing Science and Engineering*, **Vol.119**. Pp. 655-661, 1997

[10] Feng Chang-xue (Jack), Kusiak A. and Huang Chun-Che, Scheduling Models for Set-up Reduction, *Journal of Manufacturing Science and Engineering*, **Vol.119**. Pp571-579, 1997

[11] Fourie, C.J., The Impact and Measurement of Flexibility to Improve the Quality manufacturing, *Proceedings, International Conference on Quality Manufacturing*, pp184-194, 1999

[12] Golenko-Ginzburg, D., Kats V., Sitnyakovskii, S. and Itskovich, E. L. Control of the Three-Level 'Man-Computer' Production System, *Journal of Automation and Remote Control*, **Vol. 61**, Pp.866-882, 2000

[13] Hsu, Pau-lo, Fann, Wei-Ru, Fuzzy Adaptive Control of Machining Processes with a Self- Learning Algorithm, *Journal of Manufacturing Science and Engineering*, **Vol.118**. Pp.522-530, 1996

[14] Ibragimov, A.A., Controlled Markov Absorption Processes, *Journal of Automation and Remote Control*, **Vol. 60**, Pp.1739-1746, 1999

[15] Kuo Chung-Hsien, Huang Han-Pang, Wei Kuang C. and Tang Steve S. H., System Modelling and Real-Time Simulator for Highly Model-Mixed Assembly Systems. *Journal of Manufacturing Science and Engineering*, **Vol.121**. Pp. 282-289, 1999

[16] Krivonozhko, V.E., Mangazeev, V.P., and Propoi, A. I., Modelling the Development of Vertically Integrated Companies, *Journal of Automation and Remote Control*, **Vol.60**, Pp.1622-1632, 1999

[17] Lee Jay, Overview and Perspectives on Japanese Manufacturing Strategies and Production Practices in Machinery Industry, *Journal of Manufacturing Science and Engineering*, **Vol.119**. Pp.726-731, 1997

[18] Li, Q., Modelling & Analysis of a Production Queuing System for Small CIM Cell (Research Report, November 06, 2000)

[19] Li, Q., Achar, V., Pillay, S. & Bright, G., System Integrating Approach Towards CIM Modelling, *CARS & FOF*, **Vol. 1**, pp42-51, 2001, South Africa

[20] Li, Q., Jay. P, Sha, D., Logistics Support View to Develop Flexibility of CIM System, *CARS & FOF*, 2002, Portugal

[21] Li, Q., Walker, M., Sha, D., LCIM Cell Based on MATLAB/SIMULINK, International Journal for Computer, Systems and Signals, (Accepted for Review on June, 2002)

[22] Liao, T. Warren, Chen, L.J. Manufacturing Process Modelling and Optimization Based on Multi-Layer Perceptron Network, *Journal of Manufacturing Science and Engineering*, **Vol.120**. Pp109-119, 1998

[23] Mitrofanov, Yu. I. and Yudaeva, N.V., Control of Routing in Queuing Systems, *Journal of Automation and Remote Control*, **Vol.60**, Pp.1558-1567, 1999

[24] Mounayri, H. EI, Spence, A. D., and Eibestawi, M.A., Milling Process Simulation-A Generic Solid Modeller Based Paradigm, *Journal of Manufacturing Science and Engineering*, **Vol.120**. Pp. 213-221, 1998

[25] Nilsson, C.-H. and Nordahl, H. , Making manufacturing Flexibility Operational, *Integrated Manufacturing Systems (UK)*, **Vol 6 No 1**, p.5 (7 pages), 1995 , http:/www.mcb.co.uk/imc/lund/online.htm

[26] Noble, J. S., C.M. Klein and Midha, A. An Integrated Model of the Material Handling System and Unit Load Design Problem, *Journal of Manufacturing Science and Engineering*, **Vol.120**. Pp802-806, 1998

[27] Pashchenko F.F and Chernyshov, K. R., Knowledge-Based Methods and Systems for Control and Identification, *Journal of Automation and Remote Control*, **Vol. 61**, Pp.177-199, 2000

[28] Sha Daohang, A neural network robust controller for real-time control of induction motor, *The International Journal of Computers, Systems and Signals*, **Vol.1**, Pp.181-194, 2000

[29] Stori, J.A., Wright, P.K. and King, C. Integration of Process Simulation in Machining Parameter Optimization, *Journal of Manufacturing Science and Engineering*, **Vol.121**. Pp. 134-142, 1999

[30] Subhash Wadhaw Abhinav Aggarwal, Synergism of Flexibility, Integration and Automation in CIM Systems, 1999, *http:// www.iitd.ernet.in*

[31] Tham, K.D., CIM-OSA: Enterprise Modelling, Enterprise Integration Laboratory, University of Toronto.

[32] Upton, David M., Flexibility as Process Mobility: The Management of Plant Capabilities for Quick Response Manufacturing. *http://www.people.hbs.edu/dupton/papers/mobility/WorkingPaper.html#REF42782*

[33] Upton, David M., What Really Makes Factories Flexible? *Harvard Business Review*, Pp74-84, July-August 1995

[34] Wu, S. David, Roundy Robin O., Storer Robert H. and Martin- Vega. Louis A., Manufacturing Logistics Research: Taxonomy and Directions, 1999, *http://www.Lehigh.edu*

[35] Wu, S. David, Manufacturing Logistics Workshop: A Summary of Research Directions: Toward a Distributed Paradigm for Manufacturing Logistics, *http://www.Lehigh.edu*

[36] Wu, S. David and Golbasi, H., Manufacturing Planning over Alternative Facilities: Modeling, Analysis and Algorithms, 1999, *http://www.Lehigh.edu*

Bibliography:

1) Ayres, R.U., Haywood, W. and Tchijov, I., (1992) Computer Integrated Manufacturing Volume III, CHAPMAN & HALL,

2) Blanchard, Benjamin S. (1998) Logistics Engineering and management, Pentice Hall

3) Denford reference Manual (1995), Denford Machine Tools LTD.

4) Edwards, D. & Hamson, M. (1986) Mathematical Modelling Skills, Machmillm Press LTD

5) Friedland, B, (1996) Advanced Control System Design, Pentice Hall

6) Fuzzy logic Toolbox, MATLAB Reference.

7) Gang, Y.A., (1990) Operational Research, Tsinghua University Press.

8) Glynm, Peter. (1997) Performance Analysis of Manufacturing Systems, Springer

9) Haykin, S., (1999) Neural Networks: A Comprehensive Foundation, Prentice Hall

10) John F. Barlow, Excel Models for Business and operations management, John wiley & Sons, 1999

11) Karl, J., (1990) Computer Controlled System, Prentice-Hall

12) Kidd, Paul T., (1994) Agile Manufacturing, Addison-Wesley Publishing Company

13) Kusiak, A., (1992) Intelligent Design and Manufacturing, John Wiley & Sons.

14) Lu Yong-Zai, (1996) Industrial Intelligent Control, John Wiley & Sons

15) Mather, H., (1999) Comptitive Manufacturing, Woodhead Publishing Limited.

16) Parsael, Hamide R., Jamshid, M., (1993) Design and Implementation of Intelligent Manufacturing, PRENTICE HALL P T R,

17) Programmer's Guide: Microsoft Visual Basic, 1995

18) Rembold, U., Nnaji, B.O.and Storr, A., Computer Integrated Manufacturing and Engineering, Addison-Wesley Publishing Company.

19) Simulink-Dynamic System Simulation for MATLAB, MATLAB Reference.

20) Wu, B., (1996) Manufacturing Systems Design and Analysis, Chapman & Hall,

Appendix:

1. MATLAB Programs

Program 1: LCIM Model Initialize

```
%INITTLCIM  Initialize variables in the demo LCIM.m
%    M L SULTAN

global NITLQ NITMQ

LPT=8;
LQL=30;
LBR=6;
MPT=18;
MQL=5;
MBR=10;
RPT=2;

winTitle = 'LCIM';
[exist_flag, fig] = figflag(winTitle);
if exist_flag,
    animtbu([], [], init_cond, [], 'clear');
end
fismat = readfis('routing');
disp('Done reading FIS matrix and initial condition.');
```

Program 2: Robot Animation

```
function [sys,x0]=cimr2(t,x,u,flag,ts);
%Robotanim S-function for animating the motion of a robot.

%   Qingxue Li, 6-21-2001
%   M L SULTAN TECHNIKON, Durban

global cimr2

if flag==2,
  if any(get(0,'Children')==cimr2),
    if strcmp(get(cimr2,'Name'),'Robot Animation'),
      set(0,'currentfigure',cimr2);
      hndl=get(gca,'UserData');
      b=1; c=2; d=0.2; e=0; f=0.2;
      xSFP=0; ySFP=0;
      xB1=-1; yB1=0;
      xB2=-1; yB2=-0.6;
      xB3=1; yB3=-0.6;
      xB4=1; yB4=0;
      xBFP=xSFP+b*sin(u(1)); yBFP=ySFP+b*cos(u(1));
      xMC=xBFP+c*sin(u(2)); yMC=yBFP-c*cos(u(2));
      xC1=xMC+d*sin(u(1)+e); yC1=yMC+d*cos(u(1)+e);
      xC2=xC1+d*sin(u(2)+f*u(2)); yC2=yC1-d*cos(u(2)+f*u(2));

      x=[xSFP xB1 xB2 xB3 xB4 xSFP xBFP xBFP xMC xC1 xC2];
      y=[ySFP yB1 yB2 yB3 yB4 ySFP yBFP yBFP yMC yC1 yC2];

      set(hndl,'XData',x,'YData',y);
      drawnow;
    end
  end
  sys=[];
```

```
elseif flag==0,
  % Initialize the figure for use with this simulation
  animinit('Robot Animation');
  [flag,cimr2] = figflag('Robot Animation');
  axis([-2.5 2.5 -3 2]);
  hold on;

  % Set up the geometry for the problem
  % SFP=Space Fixed Pivot
  % BFP=Body Fixed Pivot
  b=1; c=2; d=0.02; e=0.5; f=0.5;
  xSFP=0; ySFP=0;
      xB1=-1; yB1=0;
      xB2=-1; yB2=-1;
      xB3=1; yB3=-1;
      xB4=1; yB4=0;
  xBFP=xSFP; yBFP=ySFP+b;
  xMC=xBFP; yMC=yBFP-c;
  xC1=xMC+d; yC1=yMC-d;
  xC2=xC1+0.08; yC2=yC1-0.08;

  x=[xSFP xB1 xB2 xB3 xB4 xSFP xBFP xBFP xMC xC1 xC2];
  y=[ySFP yB1 yB2 yB3 yB4 ySFP yBFP yBFP yMC yC1 yC2];


  hndl=plot(x,y, ...
           'EraseMode','background', ...
           'LineWidth',5, ...
           'Marker','.', ...
           'MarkerSize',20);
  set(gca,'DataAspectRatio',[1 1 1]);
  set(gca,'UserData',hndl);

  sys=[0 0 0 2 0 0];
  x0=[];

end
```

Program 3: Fuzzy Interface

```
[System]
Name='Routing'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=125
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Processing'
Range=[0 47]
NumMFs=5
MF1='Vs':'trimf',[1 1 12]
MF2='S':'trimf',[1 12 24]
MF3='M':'trimf',[12 24 34]
MF4='L':'trimf',[24 35 47]
MF5='VL':'trimf',[35 47 47]

[Input2]
Name='Queue'
Range=[0 44]
NumMFs=5
MF1='S':'trimf',[0 4 10]
MF2='VS':'trimf',[0 0 4]
MF3='M':'trimf',[4 10 23]
MF4='L':'trimf',[10 23 44]
MF5='VL':'trimf',[23 40 44]


[Input3]
Name='Breakdown'
```

```
Range=[0 50]
NumMFs=5
MF1='VS':'trimf',[2 2 5]
MF2='S':'trimf',[2 5 25]
MF3='M':'trimf',[5 25 40]
MF4='L':'trimf',[25 40 50]
MF5='VL':'trimf',[40 50 50]

[Output1]
Name='Selectibility'
Range=[0 1]
NumMFs=5
MF1='VS':'trimf',[0.1097 0.1097 0.1646]
MF2='S':'trimf',[0.1097 0.1646 0.368]
MF3='M':'trimf',[0.1646 0.368 0.68]
MF4='L':'trimf',[0.368 0.68 1]
MF5='VL':'trimf',[0.68 1 1]

[Rules]
1 5 1, 5 (1) : 1
1 4 1, 5 (1) : 1
1 3 1, 4 (1) : 1
1 2 1, 3 (1) : 1
1 1 1, 3 (1) : 1
1 5 2, 5 (1) : 1
1 4 2, 4 (1) : 1
1 3 2, 3 (1) : 1
1 2 2, 3 (1) : 1
1 1 2, 3 (1) : 1
1 5 3, 5 (1) : 1
1 4 3, 4 (1) : 1
1 3 3, 4 (1) : 1
1 2 3, 3 (1) : 1
1 1 3, 2 (1) : 1
1 5 4, 4 (1) : 1
1 4 4, 3 (1) : 1
1 3 4, 2 (1) : 1
1 2 4, 2 (1) : 1
```

1 1 4, 2 (1) : 1
1 5 5, 4 (1) : 1
1 4 5, 4 (1) : 1
1 3 5, 3 (1) : 1
1 2 5, 2 (1) : 1
1 1 5, 1 (1) : 1
2 2 1, 5 (1) : 1
2 1 1, 5 (1) : 1
2 3 1, 4 (1) : 1
2 4 1, 3 (1) : 1
2 5 1, 3 (1) : 1
2 2 2, 3 (1) : 1
2 1 2, 3 (1) : 1
2 3 2, 3 (1) : 1
2 4 2, 4 (1) : 1
2 5 2, 5 (1) : 1
2 2 3, 4 (1) : 1
2 1 3, 3 (1) : 1
2 3 3, 2 (1) : 1
2 4 3, 2 (1) : 1
2 5 3, 2 (1) : 1
2 2 4, 2 (1) : 1
2 1 4, 2 (1) : 1
2 3 4, 2 (1) : 1
2 4 4, 2 (1) : 1
2 5 4, 2 (1) : 1
2 2 5, 3 (1) : 1
2 1 5, 3 (1) : 1
2 3 5, 2 (1) : 1
2 4 5, 1 (1) : 1
2 5 5, 1 (1) : 1
3 2 1, 4 (1) : 1
3 1 1, 4 (1) : 1
3 3 1, 5 (1) : 1
3 4 1, 5 (1) : 1
3 5 1, 5 (1) : 1
3 2 2, 4 (1) : 1
3 1 2, 4 (1) : 1

3 3 2, 4 (1) : 1
3 4 2, 5 (1) : 1
3 5 2, 5 (1) : 1
3 2 3, 3 (1) : 1
3 1 3, 3 (1) : 1
3 3 3, 2 (1) : 1
3 4 3, 2 (1) : 1
3 5 3, 1 (1) : 1
3 2 4, 2 (1) : 1
3 1 4, 2 (1) : 1
3 3 4, 1 (1) : 1
3 4 4, 1 (1) : 1
3 5 4, 1 (1) : 1
3 2 5, 2 (1) : 1
3 1 5, 2 (1) : 1
3 3 5, 1 (1) : 1
3 4 5, 1 (1) : 1
3 5 5, 1 (1) : 1
4 2 1, 1 (1) : 1
4 1 1, 1 (1) : 1
4 3 1, 2 (1) : 1
4 4 1, 3 (1) : 1
4 5 1, 5 (1) : 1
4 5 2, 5 (1) : 1
4 4 2, 4 (1) : 1
4 3 2, 4 (1) : 1
4 2 2, 2 (1) : 1
4 1 2, 2 (1) : 1
4 1 3, 1 (1) : 1
4 2 3, 1 (1) : 1
4 3 3, 1 (1) : 1
4 4 3, 2 (1) : 1
4 5 3, 2 (1) : 1
4 5 4, 1 (1) : 1
4 4 4, 1 (1) : 1
4 3 4, 1 (1) : 1
4 2 4, 2 (1) : 1
4 1 4, 2 (1) : 1

4 1 5, 2 (1) : 1
4 2 5, 2 (1) : 1
4 3 5, 1 (1) : 1
4 4 5, 1 (1) : 1
4 5 5, 1 (1) : 1
5 5 5, 1 (1) : 1
5 4 5, 1 (1) : 1
5 3 5, 1 (1) : 1
5 2 5, 1 (1) : 1
5 1 5, 1 (1) : 1
5 1 4, 1 (1) : 1
5 2 4, 1 (1) : 1
5 3 4, 2 (1) : 1
5 4 4, 2 (1) : 1
5 5 4, 2 (1) : 1
5 5 3, 1 (1) : 1
5 4 3, 1 (1) : 1
5 3 3, 1 (1) : 1
5 2 3, 2 (1) : 1
5 1 3, 2 (1) : 1
5 1 2, 4 (1) : 1
5 2 2, 5 (1) : 1
5 3 2, 4 (1) : 1
5 4 2, 4 (1) : 1
5 5 2, 5 (1) : 1
5 5 1, 5 (1) : 1
5 4 1, 5 (1) : 1
5 3 1, 5 (1) : 1
5 2 1, 4 (1) : 1
5 1 1, 4 (1) : 1

Program 4: GUI Design

```
function fig = CIM_fig()
% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
% This problem is solved by saving the output as a FIG-file.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.
%
% NOTE: certain newer features in MATLAB may not have been saved in
this
% M-file due to limitations of this format, which has been superseded
by
% FIG-files.  Figures which have been annotated using the plot editor
tools
% are incompatible with the M-file/MAT-file format, and should be saved
as
% FIG-files.

load CIM_fig

h0 = figure('BackingStore','off', ...
    'CloseRequestFcn','CIMgui done', ...
    'Color',[0.8 0.8 0.8], ...
    'Colormap',mat0, ...
    'FileName','C:\MATLABR11\work\Project\test\CIM.fig', ...
    'MenuBar','none', ...
    'Name','CIM Model GUI', ...
    'NumberTitle','off', ...
    'PaperPosition',[18 180 576 432], ...
    'PaperUnits','points', ...
    'Position',[19 130 645 420], ...
    'Tag','fig 1', ...
```

```
    'ToolBar','none', ...
    'UserData',mat1);
h1 = uimenu('Parent',h0, ...
    'Callback','filemenufcn(gcbf,''FilePost'')', ...
    'Label','&File', ...
    'Tag','                    ');
h2 = uimenu('Parent',h1, ...
    'Callback','filemenufcn(gcbf,''FileSaveAs'')', ...
    'Label','Save &As...', ...
    'Tag','figMenuFileSaveAs');
h2 = uimenu('Parent',h1, ...
    'Callback','filemenufcn(gcbf,''FileSave'')', ...
    'Label','&Save', ...
    'Tag','figMenuFileSave');
h2 = uimenu('Parent',h1, ...
    'Callback',' close(gcbf)', ...
    'Label','&Close', ...
    'Tag','                        ');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'Callback','CIMgui start', ...
    'ListboxTop',0, ...
    'Position',[0.02325581395348837 0.7333333333333334
0.2697674418604651 0.06904761904761905], ...
    'String','START SIM', ...
    'Tag','Pushbutton3');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'Callback','CIMgui close', ...
    'Enable','off', ...
    'ListboxTop',0, ...
    'Position',[0.02325581395348837 0.6404761904761905
0.2697674418604651 0.06904761904761905], ...
    'String','CLOSE FIG', ...
    'Tag','Pushbutton1');
h1 = uicontrol('Parent',h0, ...
```

```
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','CIMgui update', ...
    'ListboxTop',0, ...
    'Position',[0.02 0.4 0.07000000000000001 0.05], ...
    'String','100', ...
    'Style','edit', ...
    'Tag','area');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.1 0.4 0.2 0.05], ...
    'String',mat2, ...
    'Style','text', ...
    'Tag','StaticText6');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','CIMgui update', ...
    'ListboxTop',0, ...
    'Position',[0.02 0.35 0.07000000000000001 0.05], ...
    'String','10', ...
    'Style','edit', ...
    'Tag','hilim');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.1 0.35 0.2 0.05], ...
    'String',mat3, ...
    'Style','text', ...
    'Tag','StaticText5');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
```

```
    'Callback','CIMgui update', ...
    'ListboxTop',0, ...
    'Position',[0.02 0.3 0.07000000000000001 0.05], ...
    'String','2', ...
    'Style','edit', ...
    'Tag','lolim');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.1 0.3 0.2 0.05], ...
    'String',mat4, ...
    'Style','text', ...
    'Tag','StaticText4');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','CIMgui update', ...
    'ListboxTop',0, ...
    'Position',[0.02 0.25 0.07000000000000001 0.05], ...
    'String','50', ...
    'Style','edit', ...
    'Tag','outflow');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.1 0.25 0.2 0.05], ...
    'String',mat5, ...
    'Style','text', ...
    'Tag','StaticText3');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','CIMgui update', ...
    'ListboxTop',0, ...
```

```
    'Position',[0.02 0.2 0.07000000000000001 0.05], ...
    'String','10', ...
    'Style','edit', ...
    'Tag','inflow');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.1 0.2 0.2 0.05], ...
    'String',mat6, ...
    'Style','text', ...
    'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'FontSize',12, ...
    'FontWeight','bold', ...
    'HorizontalAlignment','left', ...
    'ListboxTop',0, ...
    'Position',[0.01 0.01 0.95 0.08], ...
    'String','Push Start to begin', ...
    'Style','text', ...
    'Tag','StaticText1');
h1 = axes('Parent',h0, ...
    'CameraUpVector',[0 1 0], ...
    'Color',[1 1 1], ...
    'ColorOrder',mat7, ...
    'DataAspectRatioMode','manual', ...
    'DrawMode','fast', ...
    'Position',[0.2 0.1 0.9 0.9], ...
    'Tag','picture', ...
    'UserData',mat8, ...
    'Visible','off', ...
    'WarpToFill','off', ...
    'XColor',[0 0 0], ...
    'XLim',[-4 16.25079079039277], ...
    'XLimMode','manual', ...
```

```
    'YColor',[0 0 0], ...
    'YLim',[-6 19.50658424208974], ...
    'YLimMode','manual', ...
    'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
    'Position',[6.057738673068826 -7.623761331061413 163.1472009670314],
...
    'Tag','pictureText4', ...
    'VerticalAlignment','cap', ...
    'Visible','off');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
    'Position',[-5.646874254998881 6.651807037853535 163.1472009670314],
...
    'Rotation',90, ...
    'Tag','pictureText3', ...
    'VerticalAlignment','baseline', ...
    'Visible','off');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','right', ...
    'Position',[-22.2904278983784 19.43892751996219 163.1472009670314],
...
    'Tag','pictureText2', ...
    'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
```

```
   'Position',[6.057738673068826 19.98018129698266 163.1472009670314],
...
   'Tag','pictureText1', ...
   'VerticalAlignment','bottom', ...
   'Visible','off');
set(get(h2,'Parent'),'Title',h2);
h2 = line('Parent',h1, ...
   'Color',[0 0 0], ...
   'EraseMode','xor', ...
   'LineWidth',3, ...
   'Marker','.', ...
   'MarkerEdgeColor',[0 0 0], ...
   'MarkerFaceColor',[0 0 0], ...
   'MarkerSize',2, ...
   'Tag','pictureLine5', ...
   'XData',[0 10 10 0 0], ...
   'YData',[0 0 12.5 12.5 0]);
h2 = line('Parent',h1, ...
   'Color',[0 0 0], ...
   'EraseMode','xor', ...
   'LineWidth',3, ...
   'Marker','.', ...
   'MarkerEdgeColor',[0 0 0], ...
   'MarkerFaceColor',[0 0 0], ...
   'MarkerSize',2, ...
   'Tag','pictureLine4', ...
   'XData',[4.49670787895513 5.50329212104487 5.50329212104487
4.49670787895513 4.49670787895513], ...
   'YData',[14.5 14.5 17.50658424208974 17.50658424208974 14.5]);
h2 = line('Parent',h1, ...
   'Color',[0 0 0], ...
   'EraseMode','xor', ...
   'LineWidth',3, ...
   'Marker','.', ...
   'MarkerEdgeColor',[0 0 0], ...
   'MarkerFaceColor',[0 0 0], ...
   'MarkerSize',2, ...
   'Tag','pictureLine3', ...
```

```
    'XData',[-2 4.49670787895513 4.49670787895513 -2 -2], ...
    'YData',[16.5 16.5 17.50658424208974 17.50658424208974 16.5]);
h2 = line('Parent',h1, ...
    'Color',[0 0 0], ...
    'EraseMode','xor', ...
    'LineWidth',3, ...
    'Marker','.', ...
    'MarkerEdgeColor',[0 0 0], ...
    'MarkerFaceColor',[0 0 0], ...
    'MarkerSize',2, ...
    'Tag','pictureLine2', ...
    'XData',mat9, ...
    'YData',[-2 -2 4.250790790392765 4.250790790392765 -2]);
h2 = line('Parent',h1, ...
    'Color',[0 0 0], ...
    'EraseMode','xor', ...
    'LineWidth',3, ...
    'Marker','.', ...
    'MarkerEdgeColor',[0 0 0], ...
    'MarkerFaceColor',[0 0 0], ...
    'MarkerSize',2, ...
    'Tag','pictureLine1', ...
    'XData',[10 12 12 10 10], ...
    'YData',[2 2 4.250790790392765 4.250790790392765 2]);
h2 = patch('Parent',h1, ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-0.9, ...
    'Tag','In Valve - Closed', ...
    'VertexNormals',mat10, ...
    'Vertices',mat11);
h2 = patch('Parent',h1, ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-0.5, ...
    'Tag','In Valve - Open', ...
```

```
    'VertexNormals',mat12, ...
    'Vertices',mat13);
h2 = patch('Parent',h1, ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-0.9, ...
    'Tag','Out Valve - Closed', ...
    'VertexNormals',mat14, ...
    'Vertices',mat15);
h2 = patch('Parent',h1, ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-0.5, ...
    'Tag','Out Valve - Open', ...
    'VertexNormals',mat16, ...
    'Vertices',mat17);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch19', ...
    'VertexNormals',mat18, ...
    'Vertices',mat19);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch18', ...
    'VertexNormals',mat20, ...
    'Vertices',mat21);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
```

```
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch17', ...
    'VertexNormals',mat22, ...
    'Vertices',mat23);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch16', ...
    'VertexNormals',mat24, ...
    'Vertices',mat25);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch15', ...
    'VertexNormals',mat26, ...
    'Vertices',mat27);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch14', ...
    'VertexNormals',mat28, ...
    'Vertices',mat29);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
```

```
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch13', ...
    'VertexNormals',mat30, ...
    'Vertices',mat31);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch12', ...
    'VertexNormals',mat32, ...
    'Vertices',mat33);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch11', ...
    'VertexNormals',mat34, ...
    'Vertices',mat35);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','xor', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch10', ...
    'VertexNormals',mat36, ...
    'Vertices',mat37);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor',[0.8 0.8 0.8], ...
    'Faces',[1 2 3 4 5], ...
    'Tag','picturePatch9', ...
```

```
    'VertexNormals',mat38, ...
    'Vertices',mat39);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch8', ...
    'VertexNormals',mat40, ...
    'Vertices',mat41);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor',[0.8 0.8 0.8], ...
    'Faces',[1 2 3 4 5], ...
    'Tag','picturePatch7', ...
    'VertexNormals',mat42, ...
    'Vertices',mat43);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch6', ...
    'VertexNormals',mat44, ...
    'Vertices',mat45);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor',[0.8 0.8 0.8], ...
    'Faces',[1 2 3 4 5], ...
    'Tag','picturePatch5', ...
    'VertexNormals',mat46, ...
    'Vertices',mat47);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
```

```
    'EraseMode','none', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch4', ...
    'VertexNormals',mat48, ...
    'Vertices',mat49);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor',[0.8 0.8 0.8], ...
    'Faces',[1 2 3 4 5], ...
    'Tag','picturePatch3', ...
    'VertexNormals',mat50, ...
    'Vertices',mat51);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch2', ...
    'VertexNormals',mat52, ...
    'Vertices',mat53);
h2 = patch('Parent',h1, ...
    'EdgeColor','none', ...
    'EraseMode','none', ...
    'FaceColor','flat', ...
    'Faces',[1 2 3 4 5], ...
    'FaceVertexCData',-1.5, ...
    'Tag','picturePatch1', ...
    'VertexNormals',mat54, ...
    'Vertices',mat55);
if nargout > 0, fig = h0; end
```

Program 5: LCIM GUI Function

```
function LCIMgui(action)

%   LCIM GUI for LCIMModel
%   ML SULTAN TECHNIKAN

    if nargin~=1
    action='none';
    end

    Switch action
    case 'start'
    %it is the call back for "START SIM" button

    if stremp (get_param('LCIM','simulationstatus'), 'stopped')
     set_param('LCIM','simulation command','start')
      return
    end
    %turn on the "STOP SIM" button
    SpHndl=findobj(gcbf,'string','stop')
    set(SPHndl,'Enable','on')
    %turn on the "plot"button
    PtHndl=findobj(gcbf,'string",'plot');
    set(Pt Hndl,'Enable','on')
   %turn off parameter changes
    set(h.edit,'Enable','off');

   LCIMedit

   case 'stop'
   %halt Sim if it is running
   if stremp(get_param('LCIM','simulationstatus'),'running')
      set_param('LCIM','simulationcommand','stop')
      return
   end
```

```
      % Enable START SIM button
        stHndl=findobj(gcbf,'string','start');
        set(stitndl,'Enable','on')
      % turn on parameter changes
        set(h.edit,'Enable','on');

  %%%%%%%%%%%%%%
set(gcbf,'UserData',yout);
XYData=get(gcbf,'UserData');
x=XYData(:,1);
y=XYData(:,2);
plot(tout,x,'r-') ;
hold on;
plot(tout,y,'m:')
hold off
%%%%%%%%%%%%%%%%%%%%%%
     case 'plot'

        x=xout;
        y=yout;
        plot(x,tout,x'r_')
        hold on
        plot(y,tout,y'b__')
        hold off
     end

     case 'close'
        close(gcbf)
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function LCIMedit
 % subfunction for paramters initial value

    MPT=str2num(get(h.edit(1),'string');
    MQL=str2num(get(h.edit(2),'string');
    MBR=str2num(get(h.edit(3),'string');
    LPT=str2num(get(h.edit(4),'string');
    LQL=str2num(get(h.edit(5),'string');
    LBR=str2num(get(h.edit(6),'string');

if MPT<0 | MQL<0 | MBR<0 | LPT<0 | LBR<0
set(h.text(7),'String','All parameters must be greater than zero')
    set(h.edit(1),'String',num2str(evalin('base','LBR')))
    set(h.edit(2),'String',num2str(evalin('base','LQL')))
    set(h.edit(3),'String',num2str(evalin('base','LPT')))
    set(h.edit(4),'String',num2str(evalin('base','MBR')))
    set(h.edit(5),'String',num2str(evalin('base','MQL')))
```

Program 6: Database Function:

```
function dbimportLCIM()
%DBIMPORTLCIM Imports data into MATLAM from LCIMinput database.

%M L SULTAN

% Set maximum time allowed for establishing a connection.

timeoutB=logintimeout(5)

% Connect to a database.

connB=database('LCIMinput','','')

% Check the database status.

ping(connB)

% Open cursor and execute SQL statement.

curs=exec(connB,'select * from LCIMinput');

% Fetch the first 10 rows of data.

curs=fetch(curs)

% Display the data.

columnnames(curs)
curs.data(1,:)
%AA=cursorB.Data

input1=curs.data(1,:)

LPT=input1(1,1)
   LQL=input1(2,1)
   LBR=input1(3,1)
   MPT=input1(4,1)
   MQL=input1(5,1)
   MBR=input1(6,1)
   RPT=input1(7,1)


% Close the cursor and the connection.

close(cursorB)
close(connB)
```

```
function ddinsertLCIM()
%DBINSERTLCIM Inserts rows into LCIMdb database table.

% Connect to a database.

connA=database('LCIMdb','','')

% Open cursor and execute a SQL statement.

% Get the Data information from Matlab.

[m,n]=size(XYData)

xtest=num2cell(x)
ytest=num2cell(y)

% Create a cell array for the data to be exported.

C=cell(m,n)

C(:,1)=xtest
C(:,2)=ytest
% Define a string array of column names in table where data will be
exported.

colnames={'xout', 'yout'}

% Determine autocommit status.

get(connA, 'autocommit')

insert(connA, 'LCIMtest', colnames, C)

%%% update(conna, 'LCIMtest',colnames,C, whereclause)

% Close the cursor and the connection.

close(cursorA)
close(connA)
```

2. DOS Script

Robot.bat Script

```
cd\

cd c:\Walli3

Robot-Lathe.wal

Cd\

REM Cd c:\lcim

REM Robot1.xls

REM RobControl.xls
```

Close

3. GRYPHON Program

Program 1:Robot Movement Program

	Axis 0 Waist	Axis 1 Shldr	Axis 2 Elbow	Axis 3 Wrist	Axis 4 Wrist	Axis 5 Grip	(Elev)	(Rotn)	Spd
Start	0	5500	0	0	1600	0	800	0	3
2	1924	300	2820	1456	1051	0	1253	1002	3
3	1924	300	2820	1456	1051	1	1254	1002	3
4	1925	3683	2814	1454	1053	1	1253	1000	3
5	4766	3683	2814	1454	1053	1	1253	1000	3
6	4766	3683	5083	1508	81	1	794	1513	3
7	4436	608	5083	1507	82	1	794	1512	3
8	4420	580	5095	1508	84	1	796	1512	3
9	4379	580	5094	1505	84	1	794	1510	3
10	4350	580	5094	1506	84	1	795	1511	3
11	4349	580	5094	1506	84	0	795	1511	3
12	4486	580	5094	1506	84	0	795	1511	3
13	4486	728	5094	1506	84	0	795	1511	3
14	4488	3887	5095	1506	85	0	795	1510	3
15	4487	580	5094	1506	84	0	795	1511	3
16	4335	580	5094	1506	84	0	795	1511	3
17	4335	580	5094	1506	84	1	793	1510	3
18	4455	580	5094	1505	84	1	794	1510	3
19	4455	867	5094	1505	84	1	794	1510	3
20	4455	4478	5095	1506	85	1	795	1510	3
21	8028	3687	618	1506	84	1	795	1511	3
22	8028	1501	1706	1455	1056	1	1255	999	3
23	8028	1100	2000	1455	1056	1	1255	999	3
24	8028	1100	1999	1456	1055	0	1255	1000	3
25	8028	2000	1999	1456	1055	0	1255	1000	3
26	8028	5500	0	1030	632	0	831	999	3
27	62	5500	0	1030	632	0	831	999	3
28	62	5500	0	1030	632	0	831	999	3

Limitations:

	Waist	Axis 2 (Elbow)	Wrist (Vertical, Horizontal)
Range Limitation	$-135^0, 135^0$; 0, 9000	0,9950	1600,0

Program 2: The Main Program for Robot-Lathe (Based on Walli3)

SET GRYPHON OUTPUT 6

1 SET GRYPHON OUTPUT 1

2 SET GRYPHON OUTPUT 4

3 WAIT 5 SECONDS

4 GOTO SUBPOUTINE **Open**

5 GOTO SUBPOUTINE **Release**

6 GOTO SUBPOUTINE **Load**

7 GOTO SUBPOUTINE **Close**

8 WAIT 3 SECONDS

9 SET GRYPHON OUTPUT 5

10 WAIT 3 SECONDS

11 RESET GRYPHON OUTPUT 5

12 WAIT UNTIL DIGTAL SENSOR 6 IS ON

WAIT 3 SECONDS

13 GOTO SUBPOUTINE **Open**

14 GOTO SUBPOUTINE **Unload**

15 GOTO 1

16 **Load** REMARK Load a part into the chuck

17 MOVE GRYTHON THROUGH POSITION Start

18 FOR A=2 TO 10

19 MOVE GRYTHON THROUGH POSITION A

20 NEXT A

21 GOTO SUBPOUTINE **Grip**

22 FOR A=11 TO 14

23 MOVE GRYTHON THROUGH POSITION A

24 NEXT A

25 RETURN FROM SUBROUTINE

26 **Release** REMARK Open the chuck

SET GRYPHON OUTPUT 2

27 WAIT UNTIL DIGITAL SENSOR 3 IS ON

28 RESET GRYPHON OUTPUT 2

29 RETURN FROM SUBROUTINE

30 **Grip** REMARK Close the chuck

31 SET GRYPHON OUTPUT 3

32 WAIT UNTIL DIGITAL SENSOR 4 IS ON

33 RESET GRYPHON OUTPUT 3

34 RETURN FROM SUBROUTINE

35 **Unload** REMARK Take a part out of the chuck

36 FOR A =15 TO 17

37 MOVE GRYTHON THROUGH POSITION A

38 NEXT A

39 GOTO SUBROUTINE **Release**

40 FOR A =18 TO 27

41 MOVE GRYTHON THROUGH POSITION A

42 NEXT A

43 RETURN FROM SUBROUTINE

44 **Open** REMARK Open the door

45 RESET GRYPHON OUTPUT 6

RESET GRYPHON OUTPUT 1

46 WAIT 3 SECONDS

47 WAIT UNTIL DIGITAL SENSOR 2 IS ON

48 SET GRYPHON OUTPUT 6

49 SET GRYPHON OUTPUT 1

RETURN FROM SUBROUTINE

50 **Close** REMARK Close the door

RESET GRYPHON OUTPUT 6

51 WAIT 1 SECONDS

52 RESET GRYPHON OUTPUT 4

53 WAIT 2 SECONDS

54 SET GRYPHON OUTPUT 6

55 SET GRYPHON OUTPUT 4

56 WAIT UNTIL DIGITAL SENSOR 5 IS ON

57 RETURN FROM SUBROUTINE

4. Visual Basic Program

Program 1:Robot Control GUI

RobotCUI.vbp

```
' This program is designed for Robot_Lathe controlling

Function RobotStart()

' Make the destination program the active program.
AppActivate "Walli3"

'Send characters to the destination program
SendKeys "{F5}", True

End Function
```

```
Function RobotEnd()

' Make the destination program the active program.
AppActivate "Walli3"

'Send characters to the destination program
SendKeys "{enter}", True

End Function
```

```
Private Sub Form_Load()
Link = 1

Dim Msg
Msg = "make sure Robot-Lathe programe is Runing"
MsgBox Msg

'Dim ID
'Execute the destination program
'ID = Shell("Demo_M1.Wal", 1)
'ID = Shell("Walli3.exe", 1)

End Sub
```

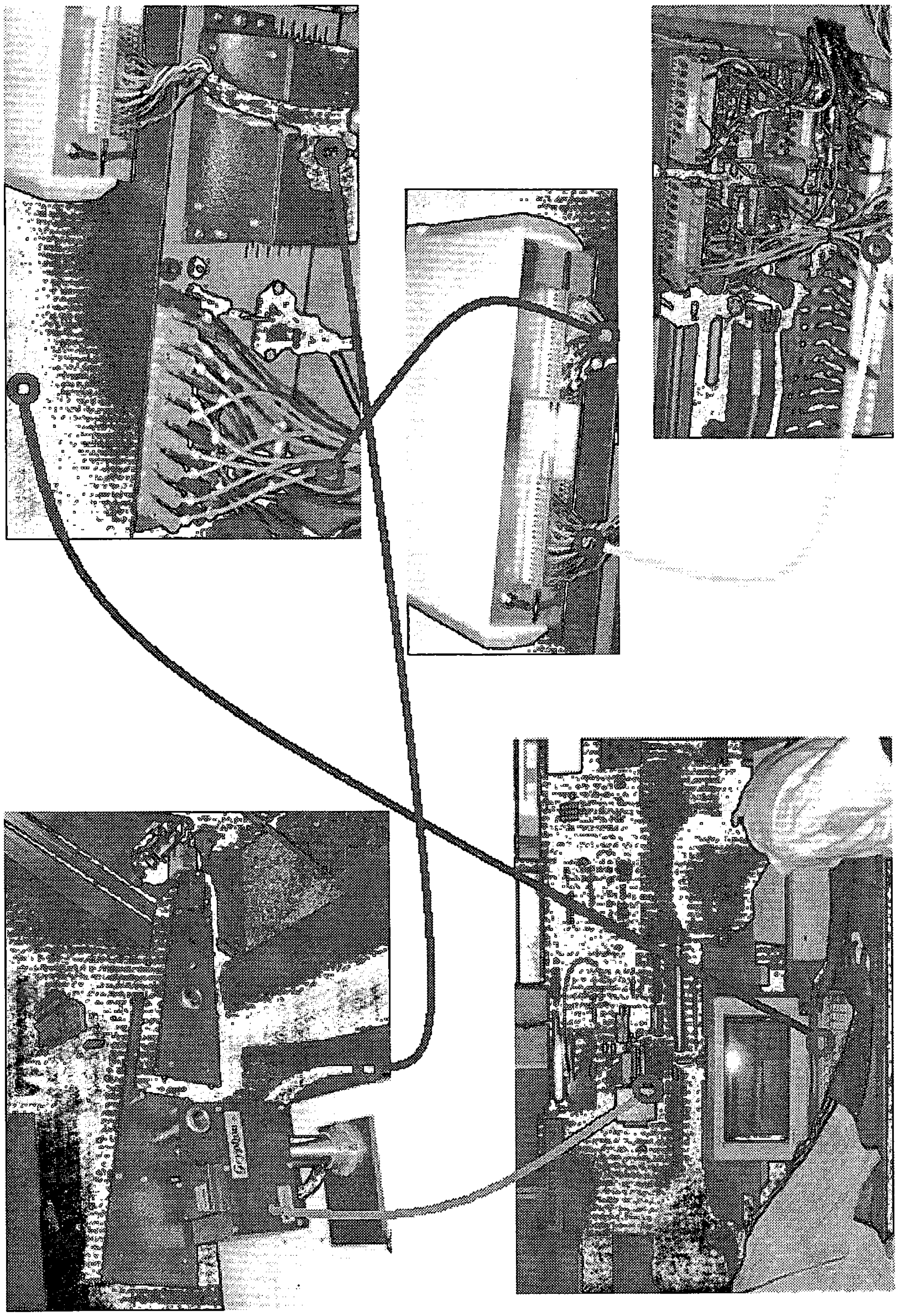
```
Private Sub cmdExit_Click()
Link = 0

RobotEnd
'Updating Database
'Data1.Refresh

End
End Sub
```

```
Private Sub cmdRobot_Click()

'Detect Robot State
Dim CurData As String

Dim ObjVar As Object
Set ObjVar = GetObject("C:\Temp\Li\robControl.xls")

'Open "Robot.bat" For Random As Link = 1

Data1.DatabaseName = "C:\TEMP\Li\Robot1.xls"
Data1.RecordSource = "Lathe"
Data1.Refresh

i% = 0

Do While Data1.Recordset.EOF = False

''For i% = 0 To Data1.Database.TableDefs("Lathe").Indexes.Count - 1

RobReady = 1

If RobReady = 1 Then

CurData = Data1.Recordset.Fields(i%).Value
MsgBox CurData
  Text1.Text = CurData
  Current = Text1.Text
  If Current = "1" Then

    'Time1.Interval = 1000
    RobotStart

    'RobotReady = 0

  End If
  Data1.Recordset.MoveNext
End If

'update the sequence

Loop

Dim Msg
Msg = "Waiting for Robot Ready"
MsgBox Msg

End Sub
```

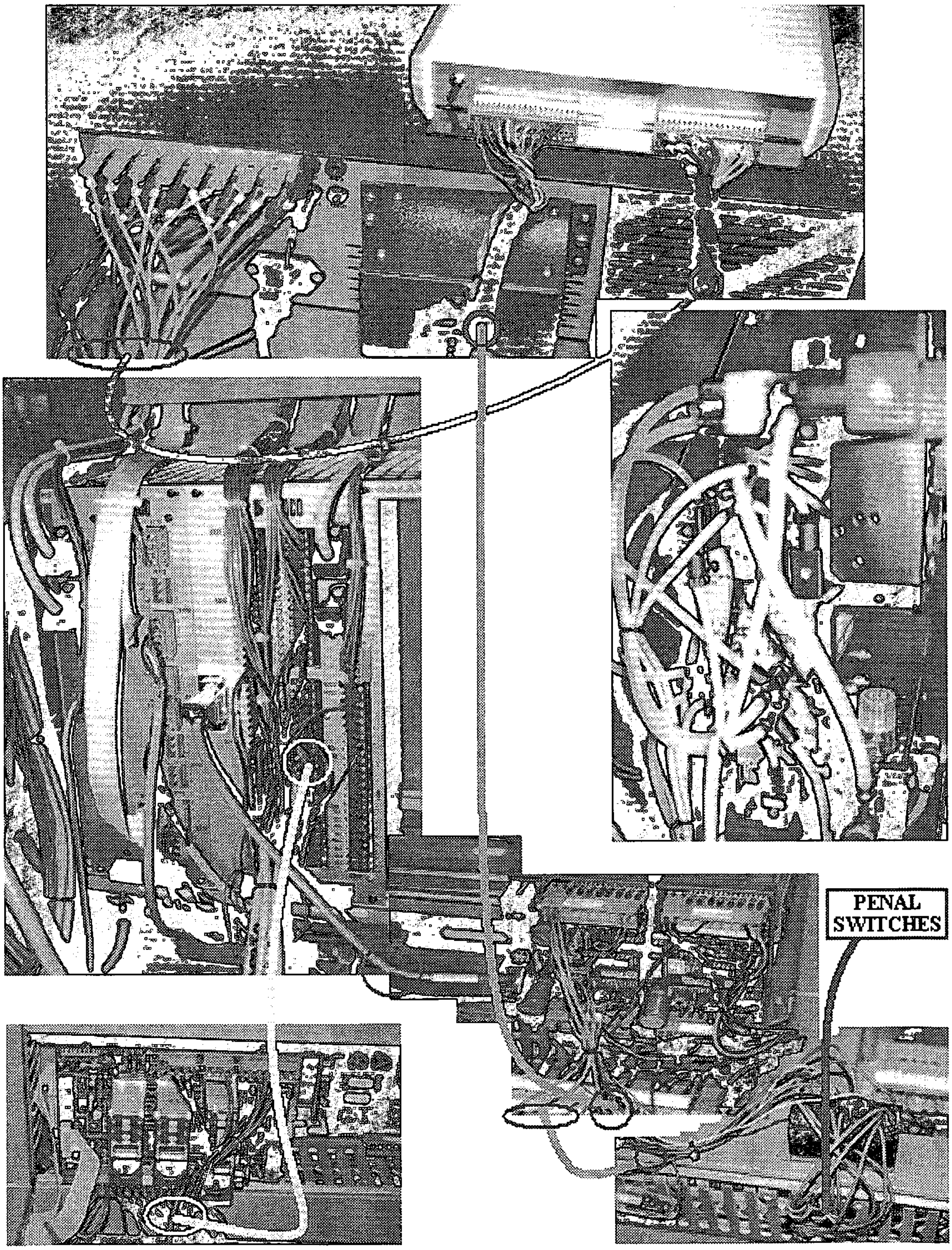
```
Private Sub cmdData_Click()
RobotOLE.UpdateOptions = 2
End Sub
```

```
Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
RobotOLE.UpdateOptions = 1
End Sub
```

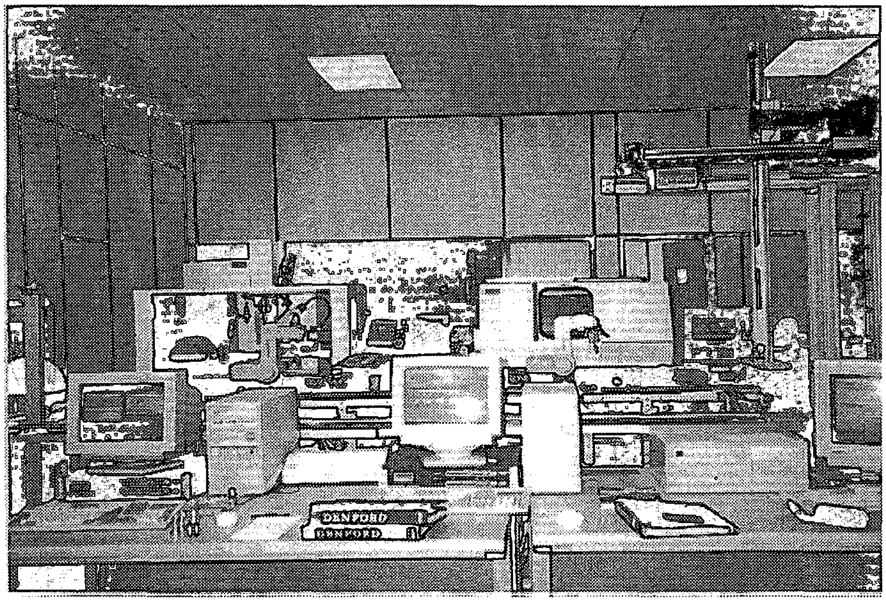
5. Pictures



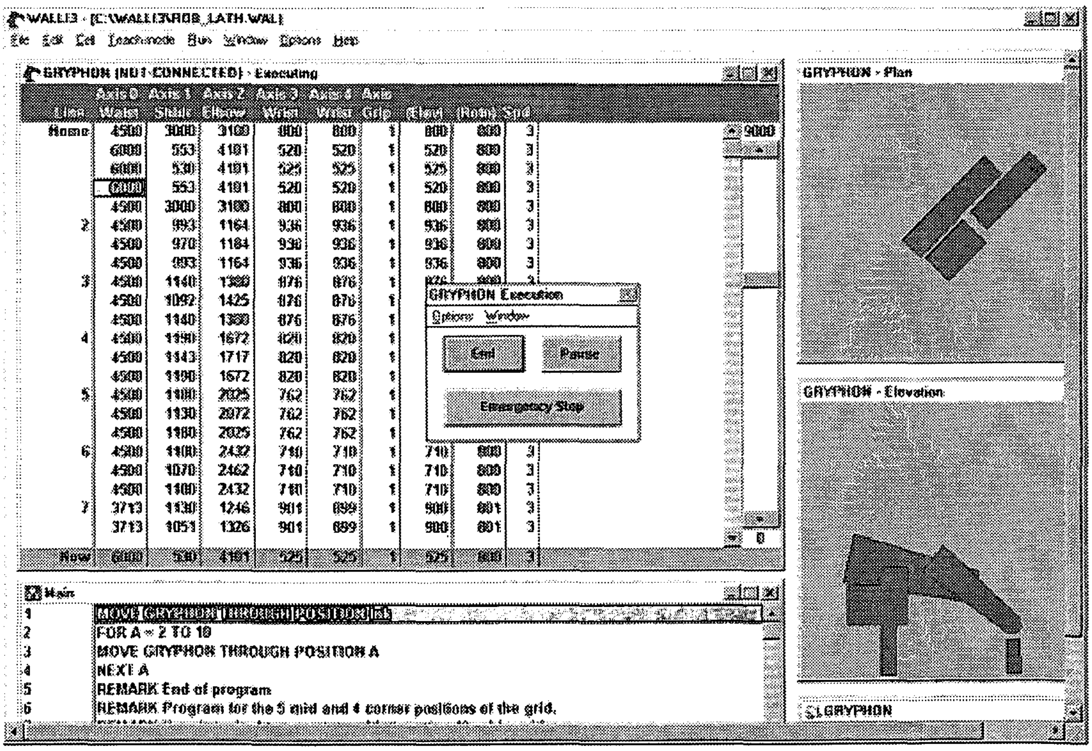
The Whole Robot-Lathe Connection



Details for Robot-Lathe Connection
(This reflects the Gryphon controller interface definition- refer to Table 7-2 for details)



DENFORD CIM Laboratory



The Robot Controller Shell (Walli3)
(Robot control GUI (figure7-11, page 92) and RobotGUI.vbp (page 135) is used with this application)