DECISION MAKING IN MANUFACTURING SYSTEMS: AN INTEGRATED THROUGHPUT, QUALITY AND MAINTENANCE MODEL USING HMM

By

BASEL SHADID, B.Sc., M. Sc.

A Thesis
Submitted to the School of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree
Doctor of Philosophy

McMaster University

DECISION MAKING IN MANUFACTURING SYSTEMS: AN INTEGRATED THROUGHPUT, QUALITY AND MAINTENANCE MODEL USING HMM

Doctor of Philosophy (2008) McMaster University
(Mechanical Engineering) Hamilton, Ontario

TITLE: Decision Making in Manufacturing Systems: An Integrated Throughput, Quality and Maintenance Model Using HMM

AUTHOR: Basel Shadid, B.Sc. (University of Jordan, Amman Jordan), M.Sc., (Cleveland State University, Cleveland Ohio)

SUPERVISORS: Professor Mohamed A. Elbestawi

Professor Stephen C. Veldhuis

NUMBER OF PAGES: 204

Abstract

The decision making processes in today's manufacturing systems represent very complex and challenging tasks. The desired flexibility in terms of the functionality of a machine adds more components to the machine. The real time monitoring and reporting generates large streams of data. However the intelligent and real time processing of this large collection of system data is at the core of the manufacturing decision support tools.

This thesis outlines the use of Frequent Episodes in Event Sequences and Hidden Markov Modeling of throughput, quality and maintenance data to model the deterioration of performance in the components that make up the manufacturing system. The thesis also introduces the concept of decision points and outlines how to integrate the total cost function in a business model.

This thesis deals with the following three topics:

First, the component-based data structure of the manufacturing system is outlined especially throughput, quality and maintenance data. In this approach, the manufacturing system is considered as a group of components that interact with each other and with raw materials to produce the manufactured product. This interaction creates a considerable amount of data which can be associated with the

relevant components of the system. The relations between the manufacturing components are established on a physical and logical basis. The components properties are clearly defined in database tables specifically created for this application. The thesis also discusses the web services in manufacturing systems and the portable technologies used in plant decision support tools.

Second, the thesis presents a novel application of Frequent Episodes in Event Sequences to identify patterns in the deterioration of performance in a component using frequent episodes of operational failures, quality failures and maintenance activities. A Hidden Markov Model (HMM) is used to model each deterioration episode to estimate the states of performance and the transition rates between the states. The thesis compares the results generated by this model to other existing models of component performance deterioration while emphasizing the benefits of the proposed model through the use of the plant data.

Finally the thesis presents a methodology using HMM probability distributions and Bayesian Decision theory framework to provide a set of decisions and recommendations under the condition of data uncertainty. The results of this analysis are then integrated in the plant maintenance business model.

It is worthwhile mentioning that to develop the techniques and validate the results in this research; a Manufacturing Execution System (MES) was developed to operate in an automotive engine plant. All the data and results in this research are based on the plant data. The MES which was developed in this research provided significant benefits in the plant and was adapted by many other GM plants around the world.

ACKNOWLEDGEMENT

First of all I would like to express my sincere gratitude to General Motors St. Catharines Powertrain organization for their continuous and generous support. The years I spent in the engine plant were great learning experience and productive sharing of views. It certainly refined my knowledge of manufacturing system management. It also gave me an edge to explore research and development opportunities while keeping my eyes on the reality of the plant floor. The research in this thesis was inspired by the organization needs and enhanced by their constructive feedback.

I am also very grateful for the collaboration with GMR - Manufacturing Systems Research Lab. I had the opportunity to work with highly professional and extremely intelligent researchers and to learn from their wealth of talent and experience. I would like thank Dr. K P Unnikrishnan, Dr. Guoxian Xiao, Dr. Jorge Arinez and Dr. Leandro Barajas. I also appreciate the confidence the lab managers had in my organization which was very clear in the deployment of enormous resources to create the research platform. This fertile research environment hosted the research in the thesis and many other GMR research projects.

I also would like to express my genuine appreciation to my supervisors; Dr. Mohamed Elbestawi and Dr. Stephen Veldhuis and to my supervisory committee members, Dr. Barna Szabados and Dr. Saeid Habibi. I truly believe that both of my supervisors had provided me with best support, advice and patience I need within my special research environment. Dr. Elbestawi's vision, guidance and directions were essential to set the road map of this research. Special thanks to Dr. Veldhuis whose constructive criticism ensured that I did not get lost in some of the details of the work presented here and that I stayed on a straight path leading towards the completion of this thesis.

Finally, I wish to thank my wife for patiently sacrificing time and attention that she truly deserved and to my two little daughters who have waited too long to get their fair share of their father's time.

This thesis is dedicated to my wife and my parents, who committed their lives to raising their children with unconditional love.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

AI	Artificial Intelligence
CBM	Condition Based Maintenance
DSS	Decision Support System
EM	Expectation Maximization
ERP	Enterprise Resource Planning
FTQ	First Time Quality
GM-GMS	General Motors Global Manufacturing System
HMM	Hidden Markov Model
HPU	Hour per Unit
IOHMM	Input-output HMM
JDBC	Java Database Connectivity
KDD	Knowledge Discovery in Databases
MCBF	Main time between failures
MDP	Markov Decision Process
MES	Manufacturing Execution System
MTTR	Main time to repair
OLE	Object Linking and Embedding
OO	Object Oriented
OPC	OLE for Process Control
PFMEA	Process Failure Modes and Effects Analysis
PM	Preventive Maintenance
POMDP	Partially Observable Markov Decision Process
RCN	Reliability-centered Maintenance
RPN	Risk Priority Number
SPQRC	Safety – People – Quality – Responsiveness – Cost
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol/ Internet Protocol
TPS	Toyota Production System

Chapter 1

Introduction

1.1 Motivation

The motivation behind this research comes from my personal experience working on a complex manufacturing management system. Working as a manufacturing engineer on automotive production lines for many years, I have always been challenged to make real-time decisions that deliver optimal results with considerable uncertainty in the given data and information. This fact applies to many decision makers in the plants whether the decisions are related to throughput management, quality control, maintenance activities or supply chain management.

The first difficulty that faces decision makers in manufacturing systems is the information overload. In an operating plant a considerable amount of data is generated constantly and at a very fast rate. Not only is the size of the datasets overwhelming but the fact that the data is in various formats, sampled at different rates, includes inaccurate records and sometimes missing decisive segments of key events. All of this adds to the complexity of decision making. To meet the challenge of this complexity a new data structure is needed to overcome these difficulties while still being manageable in a modern manufacturing environment.

The second difficulty is the constrained nature of the available data analysis tools. The existing tools are designed to work with one type of data such as production, quality or maintenance, however decisions are made, or in general are supposed to be made, based on a multidisciplinary approach where data from different fields are taken into consideration and carefully weighed before making the final decision.

The third difficulty is the urgent need for real time analysis to meet real time plant requirements. The rapid change that a manufacturing system can undergo requires rapid analysis and decision making. Some decisions have to be made with no delays especially in the case where quality is concerned or reactive maintenance needs to be initiated.

Finally, the forms of outputs for many of the existing analysis tools are incomprehensible to many decision makers in manufacturing systems. While the output of these tools is scientifically sound, it is still very difficult to get decision makers to utilize them. The integration of the output from the analysis tool into the business model has been a significant challenge even for the advanced users of these tools.

1.2 Problem Statement

As stated in the motivations section above, the research problem is to improve the decision making processes related to the management of maintenance in production lines. It includes the following items:

- Universal data structure that encompasses all relevant manufacturing system data and information needed in the decision making process.

- A set of Frequent Episodes and HMM analysis tools that model the deterioration process in the performance of the components of a manufacturing system and the effect of the deterioration on throughput, quality and maintenance.

- A business cost model that links the conditional risks of possible decisions to the deterioration states of a component performance to identify the optimal course of decisions based on Bayes decision theory.

1.3 Summary of Contributions

- The combination of frequent episodes analysis and HMM to identify episodes of performance deterioration in a component of the manufacturing system and to model the deterioration in a sequence of states.

- The use of HMM and Bayes decision theory to identify optimal decisions for the component replacement at the decision points of the system.

1.4 Thesis Outline

This thesis consists of eight chapters including this introductory chapter that covers the motivations and the research problem.

Chapter 2 provides a brief review of the literature covering the relevant research performed in the performance evaluation of manufacturing systems. It also investigates the existing decision support tools associated with lean manufacturing. This chapter explores the use of temporal data mining to identify deterioration patterns in the component performance and the use of HMM to determine the dynamics of the change processes. E-manufacturing concepts and the work conducted using web services and mobile technologies to support plant decision makers are also reviewed. Finally this chapter presents some principles of the Toyota Production System (TPS) and the GM Global Manufacturing System (GMS) as well as a discussion on how these standards are used for many decisions especially in the automotive industry.

In chapter 3, the concept of objects in manufacturing system is introduced with a description of the object paradigm. The details of the physical and logical relationships of manufacturing system components are discussed as well as their properties and data sources. This chapter presents the two sides of the decision making process; data collection and structuring from one end and the knowledge delivery on the other end.

Chapter 4 studies the fundamentals of decision support tools in manufacturing systems; types of analysis, advantages and limitations. This chapter investigates the inherent problems with manufacturing data such as inaccurate and missing records, noise and irregular sampling rates.

In chapter 5 the concept of frequent episodes is presented focusing on the duration and sequence of events. This chapter also presents the HMM, its topology and algorithms. Chapter 5 explains the method used to identify episodes of deterioration and HMM to identify the states of deterioration.

In chapter 6 the cost function is constructed and utilized to provide the optimal sequence of decisions based on Bayes decision theory. Chapter 6 also introduces the concept of decision points and the link between the decision making process and the system decision points.

Chapter 7 presents a plant case study where the data associated with different components is analyzed. Episodes of performance deterioration of the components are identified and deterioration states are estimated using HMM. The cost functions of the components on the case study are constructed and the optimal sequences of decisions are identified.

Finally chapter 8 summarizes the conclusions and makes recommendations for future research in this field.

Chapter 2

Literature Review

2.1 Decision Support Systems

Decision Support Systems (DSS) are generally computerized systems that assist management in making decisions by combining data, sophisticated analytical models and tools, as well as user-friendly software into a single system that can support semi-structured or unstructured decision making. [Kiang, Chi and Tam, 1993 [1]] described DSS as a system that provides users with an interactive, flexible, and adaptable set of tools and capabilities that can be used for analyzing important blocks of data. Systems with this kind of capability are also called Knowledge-based Systems, Expert Systems or On-Line Analytical Processing (OLAP) systems.

DSS can be used in many different knowledge domains. It has been successfully applied in medical diagnosis, business forecasting, market trending, transportation route optimization and management of manufacturing systems. DSS have provided enormous value in today's manufacturing systems because of its ability to provide highly competitive decisions in a fast paced business environment. These decisions are generally determined by lean manufacturing principles that cover almost all disciplines of the manufacturing systems such as production planning, inventory control, product quality control, maintenance management, scheduling and management of the supply chain.

Management of Maintenance in particular is depending on new tools offered by decision support systems to identify, plan and optimize maintenance activities. [Angeli, Atherton, 2001 [2]] included health monitoring tools, condition based maintenance, age based maintenance, root cause analysis and e-diagnostics as examples of DSS tools used by maintenance teams. Research in this area is fertile; case studies were conducted in almost all fields of manufacturing. The section below reviews some of these applications.

2.1.1 Decision Support Systems in Maintenance

[Behzad Foroughi, 1999 [3]] discussed the decision making process in manufacturing systems and proposed an approach based on Infinitesimal Perturbation Analysis (IPA), for discrete-event systems in combination with fuzzy set theory to determine relative service priorities for automatic stations in an assembly line. Foroughi's work was based on the accurate gradient estimation model for each machine which was practically difficult to obtain given the complex dynamics of the manufacturing system. The fuzzy control logic also had limitations per station configuration that made the approach impractical to be applied in large manufacturing systems.

[Dey, 2004 [4]] presented a risk-based DSS that reduces the amount of time spent on maintenance inspections. The risk-based DSS involved an Analytic Hierarchy Process (AHP), which was a multiple attribute decision-making technique, to identify the factors that influenced failure in specific segments. In this work Dey analyzed the failures effects by determining the probability of occurrence of the risk factors. The effect of a failure caused by each risk factor was established in terms of cost and then the cumulative effect of the failure was determined through probability analysis. Even though Dey's work was focused on the oil industry; it had strong relevance to manufacturing systems where maintenance inspection tasks have been instrumental in preventive maintenance planning. The limitations of Dey's approach were the complexity in defining all failures factors and their impact on all of the components of a machine. While this approach can still be used during certain failures it is limited in application when considering large scale manufacturing systems.

[Jiang, 2001 [5]] introduced a model to arrange preventive replacement optimally based on the available information about the health condition of the system. It emphasized fundamental methodologies to grasp the essence of PM.

This comprehensive study by Jiang focused on age-based and condition-based maintenance models and tried to provide a unifying approach. Production line throughput and inventory levels were not taken into account for scheduling maintenance disregard to the fact that such decisions were made usually by joint cross functional teams. Also in his work Jiang did not address the statistical analysis for maintenance optimization.

[Saranga, 2002 [6]] presented an approach to prevent the failures due to the gradual deterioration of a mechanical item over time and was used to improve system reliability and availability. The approach was based on monitoring relevant condition predictors of significant constituent maintenance items of the system, taking into account the availability and cost effectiveness of the monitoring techniques. Review of all constituent items was carried out and a systematic approach was used to decide on an optimal maintenance policy for each corresponding group of items. Saranga's approach depended on the existence of a cost effective condition-monitoring technique for significant items of the system which was not the case for many components. It also missed the deterioration that could be inferred from other information such as throughput and quality performance.

[Fonseca, 2000 [7]] presented a knowledge base system for the evaluation of industrial equipment in terms of preventive maintenance criticality. A deterministic rating methodology based on the principles of Reliability Centred Maintenance (RCM) constituted the logic of the analytical model. The final output which was generated by the expert system consisted of a listing of the pieces of equipment that should receive special consideration for maintenance purposes, along with a description of the possible failure modes. The study attempted to analyze the failure at the component level which was generally practical in most plant maintenance systems. The limitation of the approach was that it greatly depended on engineering knowledge to define all of the failure modes of a component and the rating based on the principles of RCM which was an estimation of the deterioration phase and was not based on an analytical model.

[Al-Najjar, Wang, 2001 [8]] proposed a conceptual model that integrated the available condition information, the deterministic models used in condition monitoring and the probabilistic models used in the area of operational research. The model covered fault detection of a mechanical component such as a rolling element bearing, prediction of its vibration level in the near future, assessment of the probability of failure of the component over a finite period of time of interest. The conceptual model included several sub-models. However the successful application of the sub-models did not always guarantee a successful application of the whole concept. Thus the concept was limited to certain applications and depended on the selection of the relevant data.

2.1.2 Decisions Based on Lean Concepts

Lean Manufacturing is a unified, comprehensive set of philosophies, rules, guidelines, tools, and techniques for improving and optimizing discrete processes. While lean concepts were created for mass production with highly repetitive processes typical of the automotive industry, when applied lean principles can benefit all manufacturing processes. It is not one of the objectives of this thesis to review the details of lean manufacturing, however it is essential to relate the decision making process to lean standards. Almost all of the decisions made in manufacturing systems, especially in the automotive industry, are initiated and measured by lean principles.

Among the key principles that directly affect the decisions related to the performance of a machine and the deterioration of its components are Jidoka (intelligent automation), Kaizen (continuous improvement), waste minimization and Poka-yoke (error proofing).

[Kim, 2005 [9]] studied the principle of Jidoka in the Toyota Production System (TPS). Jidoka deals with stopping an operation when a problem occurs and preventing the production of additional defective items. Kim's study tried to identify the optimal stopping policy for multiple-yield quality failures. The study concluded that Jidoka was likely to be optimal when factories were operating under desirable conditions (e.g., high process capability ($C_p > 1$), infrequent occurrences of assignable causes, and short repair time). Kim's analysis was based on reliable inspections for quality failures. Most of the inspection on the plant floor had a small percentage of false positives. The dynamic of a false positive was not uniform which made the stopping policy after finding the first defect not practical. Also Kim did not consider all the costs associated with Jidoka which may vary on a real time basis.

2.1.3 Temporal Data Mining and Event Sequencing in Manufacturing Data

[Fayyad, Madigan and Shapiro, 1996 [10]] defined data mining as the search through databases for relationships, trends, and patterns, which prior to the search, are not known to exist nor are visible. These relationships or trends are usually assumed to be there by the domain experts, but need to be proven by the data itself. A factor that is making data mining a necessity is the fact that the rate of growth of databases completely exceeds the rates that traditional 'manual' analysis techniques can cope with.

Temporal data mining is the search for hidden relations between sequences and subsequence of events in databases. It can be a very powerful tool for analyzing time-ordered sequences in manufacturing plants, especially, in situations where it is difficult to model the underlying physics of the manufacturing processes. The use of a frequent episode discovery framework is a model-free method that can be used to deduce (temporal) correlations among events.

[Kamrani, Rong and Gonzalez, 2001 [11]] developed a Generic Algorithm (GA) based system for intelligent knowledge discovery for machine diagnosis. The rules extracted from the heuristic knowledge were used in the design of a knowledgebase system for fault diagnosis. This approach was both time and resource consuming and did not provide much linkage with the problem domain.

[Fan, Guo and Chen, 2001 [12]] proposed techniques for automatically extracting the process knowledge from production databases in terms of fault diagnosis and for optimizing performance with fixed targets. An integrated parametric analysis scheme was developed using graphical methods to facilitate the interpretation of the results. It consisted of five phases: Device Variation Partition, Key Node Screening, Linear Equipment Modeling, Graphically Aided Interpretation, and Control Policy Re-evaluation. The concepts of quality control, data mining, and process knowledge were integrated in this scheme. This field data case study showed that the integrated parametric analysis scheme was able to diagnose the parametric yield problem, help engineers construct the knowledge base, predict the yield and provide insights for yield enhancement. The work provided a good application for mining quality data to infer device performance. This concept will be expanded in this thesis topic where throughput and maintenance data will be used as well in an integrated approach.

[Dabbas, Chen, 2001 [13]] presented an integrated relational database approach for modeling and collecting semiconductor manufacturing data from multiple database systems and transforming the data into useful reports. Reports were generated to monitor factory performance by tracking different key metrics. These reports were implemented in wafer fabrication processes and have contributed significantly in improving factory performance. The work was a good example of temporal data mining of multiple databases to generate relevant production reports to support the business model. It was not based on any analytical model but still it could be used for incremental throughput improvements in the production lines.

The pioneering work of [Mannila, Toivonen and Verkamo, 1997 [14]] presented the concepts for discovering frequent episodes in event sequences. This work had a significant impact on many frequent episode algorithms including the algorithm used in thesis. The algorithms developed by Mannila, Toivonen and Verkamo were deployed to discover episodes and rules in a telecommunication network fault management database. While the work of Mannila was momentous, it did not link the rules produced by the analysis tool to the decision making process. The data driven analysis tool was not feasible unless it was deployed in a more comprehensive knowledgebase system with decision support capabilities that could eliminate the statistically significant but practically trivial rules and enabled decision makers to use the rules in business decisions.

[Laxman, Shadid, Unnikrishnan and Sastry, 2007 [15]] used temporal data mining in analyzing fault logs in an engine assembly plant. The tool was based on a frequent episode mining framework for the analysis of data containing faults. It showed how the output from such a framework, can be used to help plant engineers to interpret the large volumes of faults logged in an efficient and convenient manner. This tool was used to identify correlations between machine failures. This research work linked rules from the temporal data mining analysis process to the business model. The tool provided better information for the decision maker but not the required knowledge for decision making. It did not integrate data mining information with other information, specially the maintenance records.

2.1.4 Markovian Approach for Modeling Process Change and Performance Deterioration

Hidden Markov Models (HMMs) are particularly useful for the monitoring of machine health because computationally efficient methods exist for computing likelihood. This feature is important since it holds the promise that signal processing tools based on HMMs can be implemented cost effectively. Furthermore, there exist efficient techniques which can be used for system identification using HMMs. This means that HMMs can be used to build data-driven models of a machine thereby somewhat relieving the need to identify specific features in the data which are commonly used as health indicators.

There are, at least, two reasons why a HMM provides a realistic representation of the diagnostic process in a system. First, an HMM is capable of characterizing an embedded stochastic process with an underlying stochastic process that, although unobservable (hidden), can be observed through another set of stochastic processes. In the performance deterioration problem, the deterioration states of the system are not observable directly, i.e., they correspond to the hidden part of the embedded stochastic process. The hidden states of the system can be observed through another set of stochastic processes that produce the sequence of uncertain measurements. The key problem related to performance deterioration is to choose the most likely (hidden) state sequence, given the sequence of uncertain measurements. Secondly, a HMM is a parametric model characterized by the state transition probabilities, the emissions probabilities of measurements given the system state and the initial state distribution. These parameters can be adaptively estimated by the well-known Baum–Welch algorithm. Therefore, HMM algorithms are capable of considering the two problems of finding the most likely state sequence and of estimating model parameters within the same theoretical framework.

[Bunks, McCarthy and Al-Ani, 2000 [16]] proposed performing CBM using vibration measurements and HHM. Prior knowledge of how a machine operated or information about the relative frequency of the occurrence of different types of defects was used to improve the performance of the classification algorithm. A doubly stochastic process was constructed which described probabilistically how the machine was likely to transition from state to state and what the statistics were associated with each state. However, the work did not explain how to train the HMMs for CBM. This was a very critical part of the application. The work was also only based on vibration analysis though it mentioned the possibility of including other observations like load and temperature.

[Tai, Ching and Chan [17]] applied HMM to the detection of machine failure in a process control problem with two models one for each case of indistinguishable production units and distinguishable production units. This work was a good example of inferring the machine failure from the quality of the products. However the work did not discuss the deterioration states before the failure state. It is useful in case of multi-cell operations where it may be difficult to identify the defective operation causing the low quality products.

[Ying, Pattipati, and Hine, 2000 [18]] presented a HMM algorithm for fault diagnosis in systems with partial and imperfect tests. The HMM based algorithm found the most likely state evolution, given a sequence of uncertain test outcomes over time. On-line estimation of the HMM parameters, namely, the state transition probabilities as well as the emission probabilities of test measurements given the system state and the initial state distribution was presented. The efficacy of this parameter estimation method was demonstrated by comparing the diagnostic accuracies of an algorithm with complete knowledge of the HMM parameters with those established using an adaptive approach. This approach was limited to one fault in a time, which was a practical assumption in real systems. This work used a systematic approach and was only limited because of its dependence on one type of failure data.

[Schick, Gershwin, 2005 [19]] pioneered the area of modeling the deterioration of a machine based on a Markovian model of manufacturing quality and quantity. In the research Schick and Gershwin analyzed how the design of a production system, quality, and productivity are inter-related in production networks. The work developed analytical and numerical methods that evaluate, compare, and optimize the performance of competing designs. More specifically, it summarized simulation as well as analytical and numerical results concerning the effect of separating inspection from an operation. The research in the thesis topic was based at the concepts and results of the research conducted by Schick and Gershwin. One limitation of Schick and Gershwin approach was the entity level of monitoring. While the approach used the machine as a base entity, most of the maintenance decisions and activities were based on the component level.

2.1.5 Baysian Decision Theory and Dynamic Programming in Manufacturing DSS

Bayesian decision theory is a method for computing optimal decision rules given a prior distribution, a likelihood function, and a cost function. It is a tool broadly used in pattern recognition and it provides a solid theoretical foundation for thinking about problems of action and inference under uncertainty. Among the implementations of Bayesian decision theory, there are many systems that implement decisions automatically (e.g., in navigation, chemical engineering, manufacturing, and pattern recognition), as well as systems that support in the decision making processes such as DSS.

Dynamic Programming is a recursive method for solving sequential based decision problems. The idea is to break a large problem down into incremental steps so that at any given stage optimal solutions are known for specific subproblems or decision stages that can be addressed sequentially, normally by working backward from the last stage. When the technique is applicable, this condition can be extended incrementally without having to alter previously computed optimal solutions into subproblems. Eventually the condition applies to all of the data and if the formulation is correct, this together with the hypothesis that nothing remains untreated gives the desired answer to the complete problem.

2.1.6 Partially Observed Hidden Decision Process

In the Markovian framework when the state transitions can justifiably be represented as random, a Markov Decision Process (MDP) framework is appropriate for formulating maintenance policies. If the state of the system is not directly observable by the decision-maker but must be inferred on the basis of possibly inaccurate observations, the MDP framework is generalized to the Partially Observable Markov Decision Process (POMDP). This realistic extension of MDPs dramatically increases the complexity of POMDPs, making exact solutions virtually intractable. In order to act optimally, the analysis might need to take into account all the previous history of observations and actions, rather than just the current state it is in. A POMDP is comprised of an underlying MDP, extended with an observation space and observation function.

[Ivy, Pollock, 2005 [20]] characterized the structure of optimal policies for maintenance & replacement actions over a finite horizon. The context was machine monitoring when the machine experienced increasing levels of deterioration, the observations were related probabilistically to the state of the process and the machine's state was known with certainty only immediately after a replacement. In this research the theory & results of partially observable Markov decision processes (POMDP) were used to prove the policy that minimized the total expected cost of system maintenance had a "marginally monotonic" structure. POMDP represented a solid framework for the decision making processes when the states of the deterioration process were not completely observed specially in the case of HMM. The practical implementation of the POMDP had many difficulties in real time analysis. That reason made Bayes decision theory more applicable in this research.

2.2 Performance Evaluation in Manufacturing Systems

The performance of a manufacturing system is largely dependent upon the conditions of its system components. By closely monitoring the conditions of critical system components and carrying out timely system diagnosis as soon as a fault symptom is detected will help to reduce system downtime as well as improve overall productivity and quality.

[Chan, Yung, 2002 [21]] presented a performance measurement system to determine the priority given for the execution of quality improvements according to the performance of the process. A fuzzy set of quality measures were introduced to calculate numerical quality ratings for each process. While this technique was based on Total Quality Management (TQM) and manufacturing resource planning (MRP II), the model did not link the quality rating of the processes to its productivity and maintenance cost leaving the final performance more biased to quality concerns.

[Yurdakul, 2002 [22]] provided a multi-criteria performance measurement model to measure the profitability performance of a manufacturing system based on an Analytic Hierarchy Process (AHP) and its extension to a System-With-Feedback (SWF). In this approach information was decomposed into a hierarchy of criteria and sub-criteria. After forming the hierarchical structure, pair wise comparisons between criteria were made to establish their weights, which reflected the relative importance of each criterion, and the performance of a manufacturing system was rated. This work highlighted the inter-relations between the system criteria and how change in one would affect the other. The model criteria of interest were system dependability, system flexibility, quality, cycle time and finally cost. While this approach covered a wide rage of manufacturing criteria, it was based on mainly financial terms making it difficult to integrate the model outputs into the manufacturing business model.

2.3 E-Manufacturing

Information technology serves not only as an enabler or facilitator of organizational processes and functions, but also as a transformer or stimulus for new organizational paradigms. By reducing barriers to collaboration, compressing lead time, eliminating physical movement, and enriching decision-making, IT helps manufacturers achieve their goal of meeting customer needs better, quicker, and cheaper. By providing global reach and easy connectivity, information technology has fostered cooperation while increasing market competition, and heightening customer expectations.

[Koç, Ni, Lee, 2003 [23]] defined E-Manufacturing as a transformation system that enables the manufacturing operations to achieve predictive near-zero-downtime performance as well as be synchronized with business systems through the use of web-enabled and tether-free infotronics technologies. These technologies integrate information and decision making among data flow, information flow, and cash flow frameworks.

[Lee, 2003 [24]] proposed a framework for e-manufacturing and e-maintenance systems based on real-time access to information at various and remote locations through web services. E-Maintenance provided predictive intelligence tools to monitor manufacturing assets (equipment, products, process, etc.) through internet wireless communication systems to prevent them from unexpected breakdown. In addition, e-manufacturing tools compared the performance of a product through globally networked monitoring systems to allow companies to focus on degradation monitoring and prognostics rather than fault detection and diagnostics. This framework conducted its analysis on the machine component level in line with the plant maintenance tracking systems. While this approach considered the whole manufacturing process in its vision, it was not giving a specific link to the product quality analysis or to constraint analysis in its maintenance intelligence model.

For decades, the dominant manufacturing model was based on principles of mass production. Standardized parts and processes made economies of scale achievable, but provided limited design flexibility and customization. The outsourcing and lean manufacturing movements of the 1980s and 1990s drove the emergence of a new paradigm, termed the Quality Management era. Outsourcing shifts critical elements of the design and production process onto a manufacturers' supply chain. In the E-manufacturing era, design cycle times and inter-company costs of manufacturing complex products were to implode (Figure 2.1, Source: NACFAM)

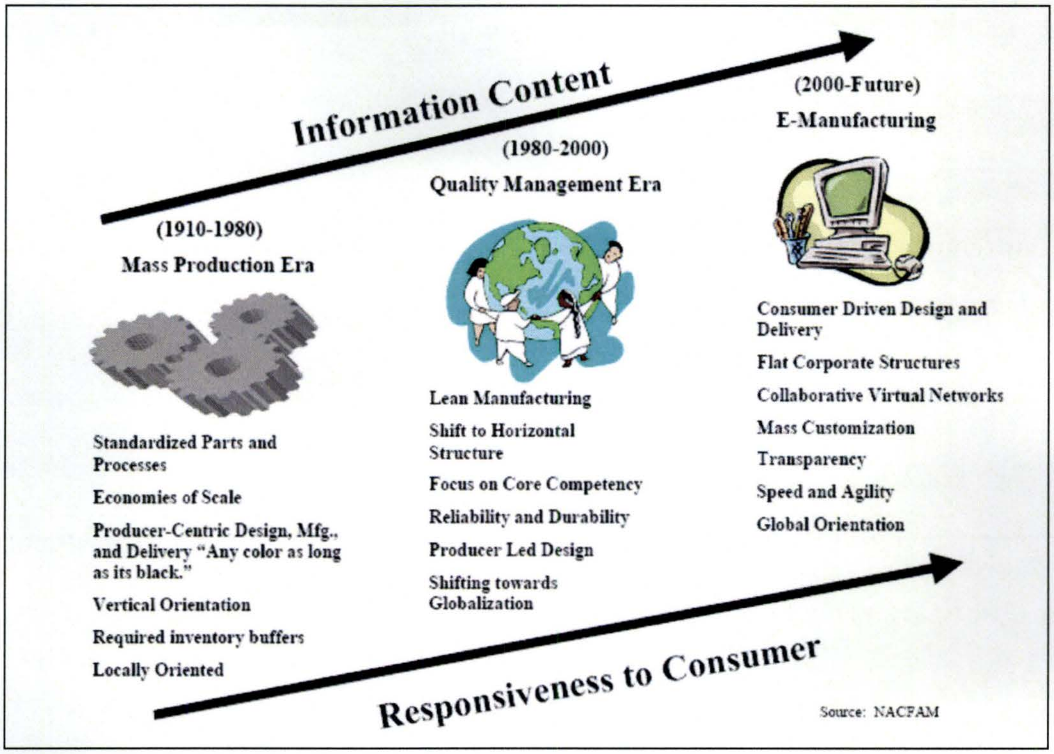


Figure 2.1 - Evolving of Manufacturing System Principles and Core Values, Source: NACFAM

Most of IT systems which are used in manufacturing were either upgraded to web-based systems or in the process of becoming web-enabled. The advantages of web-enabled systems are significant and in most cases justify the upgrading costs. Accessibility, agent mobility and standardization are among the most valued features for manufacturing systems. The market growth for web-based manufacturing systems for design and operation is expected to exceed $ 3 billion per year according to a study by the ARC Advisory Group conducted in 2001 (Figure. 2.2)

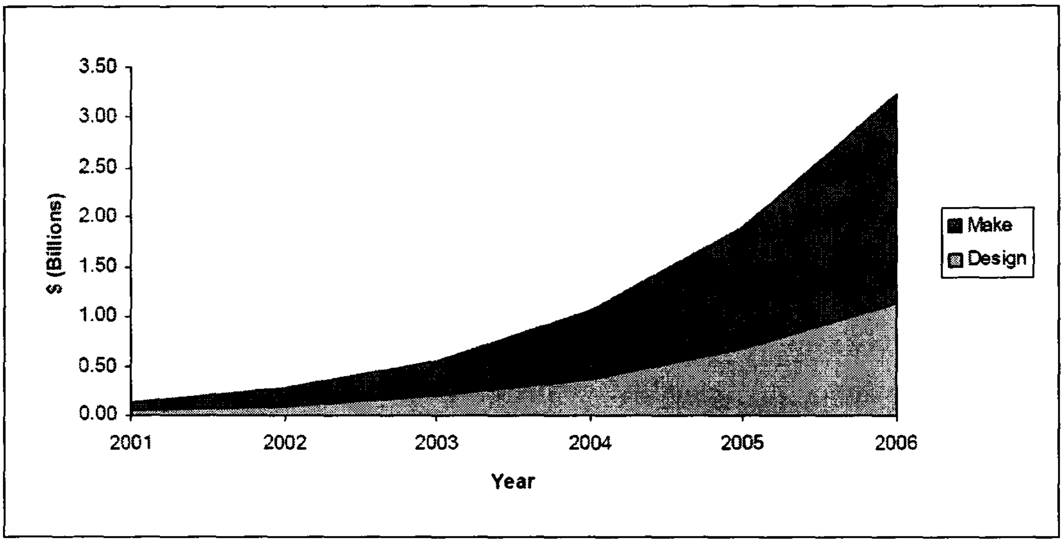


Figure 2.2 - Total Market Size of Web-based Solution for Manufacturing Systems, Source: ARC Advisory Group (2001)

2.3.1 Web-based Services in Manufacturing Systems

Web applications can be implemented in many programming languages and environments. Two of the most popular and competing choices are the ActiveX technology and Java technology. Following the client – server architecture, a web application usually includes two parts: the application server and the application client. They can be deployed in various ways either dependent or independent of web browsers and/ or web servers.

[Jiang, Zhou, Liu, 2002 [25]] developed a prototype of an ASP-driven e-service platform for online manufacturing based on the concept of e-service with a Java web solution. It included the mobile agent using an application service provider (ASP) principle. One of the most important contributions of this research was a conceptual extension of e-service, from after manufacturing support to in manufacturing analysis. This approach was under the process of improvement to handle specific limitations such as the evaluation for dynamic manufacturing capabilities. The platform which was developed in this research had similar IT architecture with more focus on the component-based data structure for real time decision support tools.

2.3.2 E-Manufacturing as a Form of Decision Support System

The expanded vision of E-manufacturing made it similar to a Manufacturing Execution System (MES) in supporting multi-discipline cross functional plant activities with more focus on the web services and wireless communications. Both, E-manufacturing and MES formed the link between the control systems in the plant and the ERP system. E-Manufacturing can be considered as a web-based DSS with OLAP capabilities based on the following tools: Data and information transformation tools, Prediction tools, Optimization tools and Synchronization tools.

2.4 Manufacturing Systems Management in the Automotive Industry

Many improvements in the management of manufacturing systems were attributed to the automotive industry. The highly competitive nature and the complexity of this industry motivated several enhancements which materialized into frameworks of processes. These processes were adopted globally and through different industry initiatives like lean manufacturing, Statistical Quality Control (SQC), Total Quality Management (TQM) and Reliability Centred Maintenance (RCM). Toyota Production System (TPS) and General Motors Global Manufacturing Systems (GM-GMS).

2.4.1 Toyota Production System

TPS is a set of principles that organizes Toyota's manufacturing operation. TPS stands as a major part of the lean manufacturing framework. The success of the Toyota Production System has spurred much research into manufacturing system design and management. The objective of TPS is to design out overburdened processes, smooth production and eliminate waste. Figure 2.3 shows Just-In-Time (JIT) and machine automation (JIDOKA) as the pillars of a successful TPS system. JIT and JIDOKA have a significant impact on the design and operation of many production lines. Also many of the decision processes which are used to manage automotive production lines are in-line with these two principles.

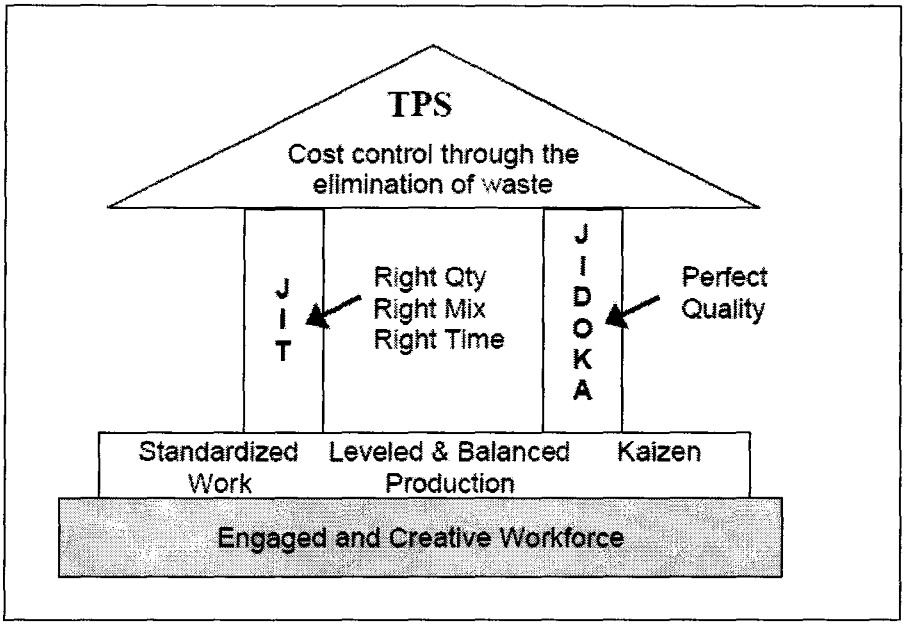


Figure 2.3 - Toyota Production System Principles

2.4.2 GM Global Manufacturing System

GM-GMS is a set of principles and core values that drives the decision making process in GM plants in almost all areas. The system is very similar to the Toyota Production System in many aspects. GM-GMS consists of 5 principles and 33 elements related to manufacturing and non-manufacturing activities. GM-GMS specify business performance in five key measures: safety, people, quality, responsiveness and cost (SPQRC). The order of these items set the priority in decision making. While these measures have different meanings for different people, in GM-GMS "Safety" measures the implementation of actions that protect the health and safety of each employee. "People" measures the involvement of the entire workforce in the attainment of the company's goals. "Quality" measures the improvements in attaining customer satisfaction in terms of product/service quality. "Responsiveness" measures the response to internal and external customer issues. Finally "Cost" measures effectiveness of actions to eliminate waste.

As the case study of this research was implemented in a GM plant, it is necessary to discuss some GM-GMS items that are directly relevant to the integration of the analytical tool in the business model. The case study shows that while the GM-GMS items provide prior knowledge and act as guidelines for planning and decision making, it is the proper estimation of the SPQRC measures that is essential in forming the cost function.

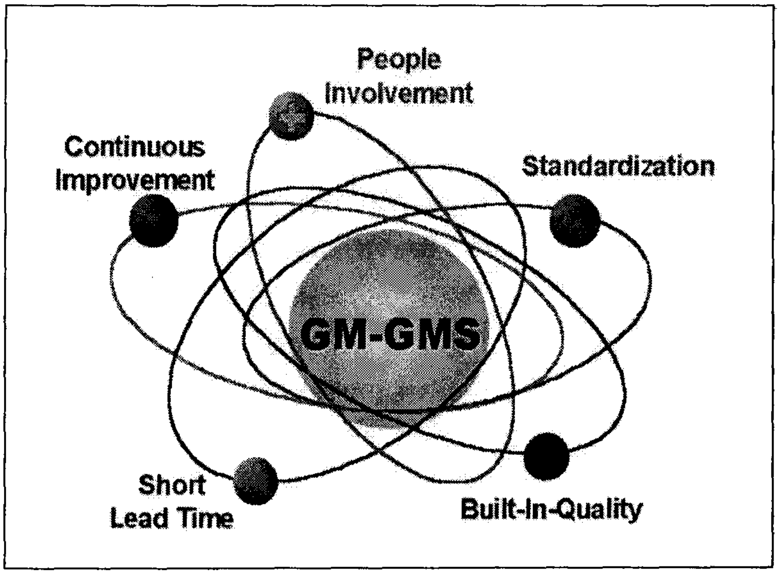


Figure 2.4 - GM-GMS Principles

In the current practice, decisions made within the GM-GMS framework are initiated by the review of Business Plan Deployment (BPD). The BPD works as a closed loop process of four items: plan, do, check and action.

Summary of Literature Review

- Decision support systems and systems with similar functionalities are used in almost all areas of management to help in automating the decision making processes. Existing DSS in today's manufacturing systems are specialized and customized tools that cover a range of decisions and most of them are not scalable and nor mobile.

- The performance measures in manufacturing systems are in general belonging in the spaces of production throughput, product quality and manufacturing cost. Different business models established specific criteria within these spaces to better drive the decision making process. There is a need for an integrated approach which combines throughput, quality and cost measures.

- Improvements in the information technology field in the areas of data storage, data analysis and knowledge representation are driving much of the development in the management of manufacturing systems. The research in this area is growing rapidly. However, there is a need for standard data and information structure that links the components that make the manufacturing system to the data sources.

- Frequent episode analysis was used in manufacturing systems but the applications were limited to report patterns that could be used in specific manufacturing decisions. Frequent episodes analysis and HMM were not combined to identify the patterns and the evolution of performance deterioration in a machine component.

- Performance deterioration in the components has been modelled with Markov processes. HMM was used in many studies to model uncertainties but it was limited to one or two groups of manufacturing data sources mainly quality and vibrations.

- Finally, Baysian decision theory and POMDP are commonly used for sequential decision optimization with uncertainties. Baysian decision theory was deployed in many DSS, however the research is limited for manufacturing and maintenance DSS.

Chapter 3

Component-based Data Structure and Decision Making in Manufacturing Systems

3.1 Manufacturing Systems Components

The object methodology was created in the programming field to emphasize modularity in software as a way to reduce software complexity. A module generally must be a component of a larger system, and operate within that system independently from the operations of the other components. Components of manufacturing system are considered independent in many decision making processes, for example repair and replacement decisions for a component from a machine are made independently. Decisions to validate and calibrate inspection apparatus are also made independently. The fact that decisions are usually made for individual component make the component-based data structure fits appropriately for building decision support tools in manufacturing systems.

The objectifying of the components that make up a manufacturing system enabled the proper linking of components with their data sources. It also established the relations among the components in both physical and logical spaces.

3.1.1 The Need for a Component-based Data Structure

The information overload is a dilemma in today's manufacturing systems. Decision makers are drowning with too much data and information whether it is historical records or real time data streams. Uncertainties in the available data and information with low signal-to-noise ratios make it even more difficult to identify what information is relevant for use in decision making. Different data formats and sample rates add more complexity to the inference processes. All this necessitates the development of data structures that link the sources of existing data and information across the manufacturing IT systems to the components of the manufacturing system. In this research, component-based representation of the data was used.

3.1.2 The Object Paradigm

Object-oriented concepts have evolved from three distinct disciplines: artificial intelligence, conventional programming languages, and database technology. In brief discussion, the object-oriented paradigm is based upon the following primary concepts:

1. An object is an entity that has state, attributes and services.

2. Attributes represent the state of the object.
3. A class is a way of grouping objects with similar attributes or services.
4. Services are the operations that all objects in a class can do.
5. Relationships represent the links between objects.

- Not every component of the manufacturing system was considered as an object in the data structure. Only components which were associated with the data collection processes, retained decision related information or had an observed change that modified its attributes were considered as objects. The components of a manufacturing system were grouped in classes which represent the physical structure of the system such as a line, operation, station or machine component.

As an example to illustrate the component-based data structure which was used in this research, the servo torque unit is discussed in detail. The servo torque unit shown in Figure 3.1 is used extensively in fastener monitoring systems in assembly lines. While each individual servo toque unit in a bolt torque station is considered as an independent component, each station itself is also considered as a component of the manufacturing system.

The servo torque unit as a component has many attributes such as the operational failures rate, the quality failures rate, effective production rate, reactive maintenance records, preventive maintenance records, etc. These attributes are linked to data sources that estimate the states of a component. For example the operational failure state is estimated by the probability distribution of the machine faults when the torque gun malfunctions. The torque gun performs a set of specific services and operations. It receives a torque command, runs down the bolt and reports the values of torque and angles for each torque phase. The torque gun also has a set of relations with other components like a parent/child relation to a parent station.

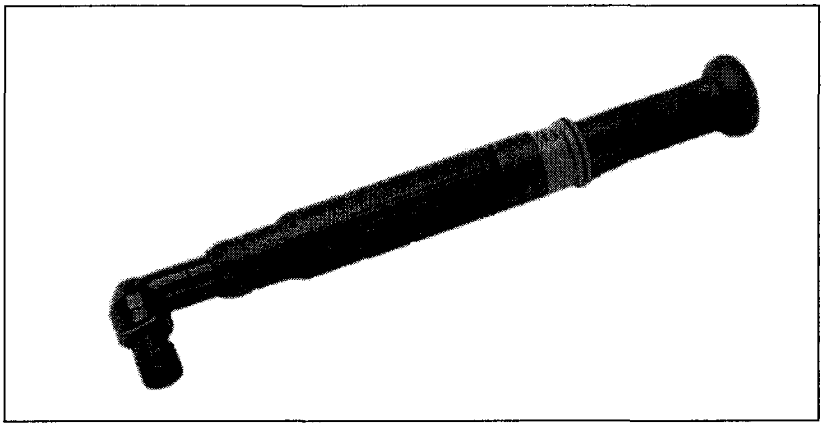


Figure 3.1 - Servo Torque Gun Used in Assembly Processes

3.1.3 Components Physical and Logical Relations

A relation is defined among two or more components. It is important to distinguish between components physical and logical relationships. This distinction is essential in extracting the structure of the system for different analyses.

Physical Relations describe the operational location of the component in the manufacturing system in relation to other components. It also applies to machine layout in production line topology. Figure 3.2 shows the physical relation between components in hierarchical order displayed on the user interface of MES which was developed in this research.

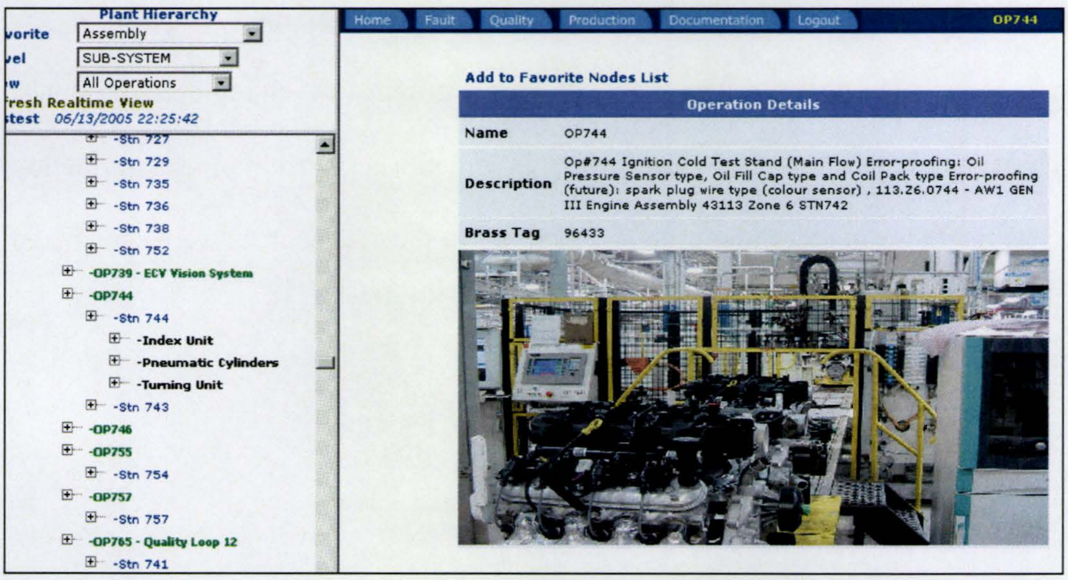


Figure 3.2 - Physical Relations between Components in Hierarchical Order

Logical Relations describe the relations between system components from the functional and behavioural side. This includes the relations between the components that affected specific quality failures or have the same operational failures. Logical relations also identified on the operation class for a group of machines that perform the same function, such as drilling or leak testing, etc.

While the physical relations between the components of the system are very limited due to the static structure of the production lines, the number of logical relations increases in proportion with the complexity of the system.

3.1.4 Components Attributes and Data Sources

An attribute is an abstraction of a single characteristic that describes the component. The component usually has many attributes which represent its states. These states are estimated by data which is sometimes stored in other information systems. In this research the data and the links to the data sources were considered as components attributes.

3.2 Information Systems in Production Lines

In today's production lines, it is common to find several information systems to support the manufacturing activities. Such systems include maintenance management systems, product quality control and accountability systems, a plant production monitoring systems, machine component health monitoring systems, employee training records, an inventory of spare parts, a statistical quality control systems and a plant CAD drawing systems.

A major phase in this research was the development of a Manufacturing Execution System (MES) to host the decision support tools. The developed MES has three tiers: data and an information access tier, the application server tier and a knowledge presentation tier. The interactions between the system tiers and other manufacturing information systems are based on standard and industrial communication protocols as shown in Figure 3.3.

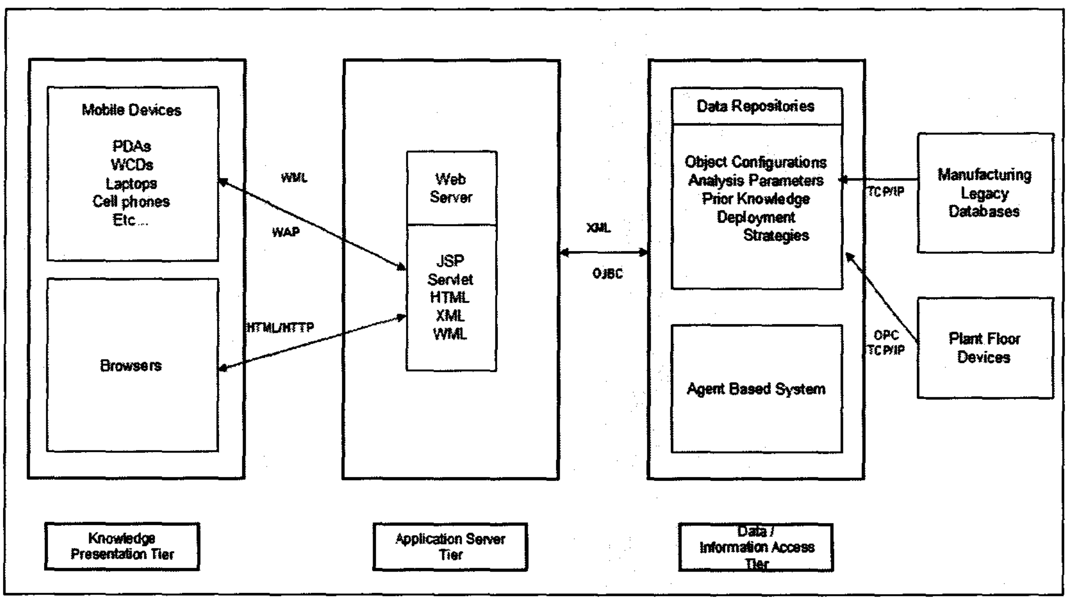


Figure 3.3 - MES Three-Tier System

The data/information access tier hosted the configuration of the components. All components were defined along with their classes, attributes, services and relations. The configurations resided in a relational database. Figure 3.4 shows an example of the parent/child relationship as a physical relation between the components measured variables and its parent machine.

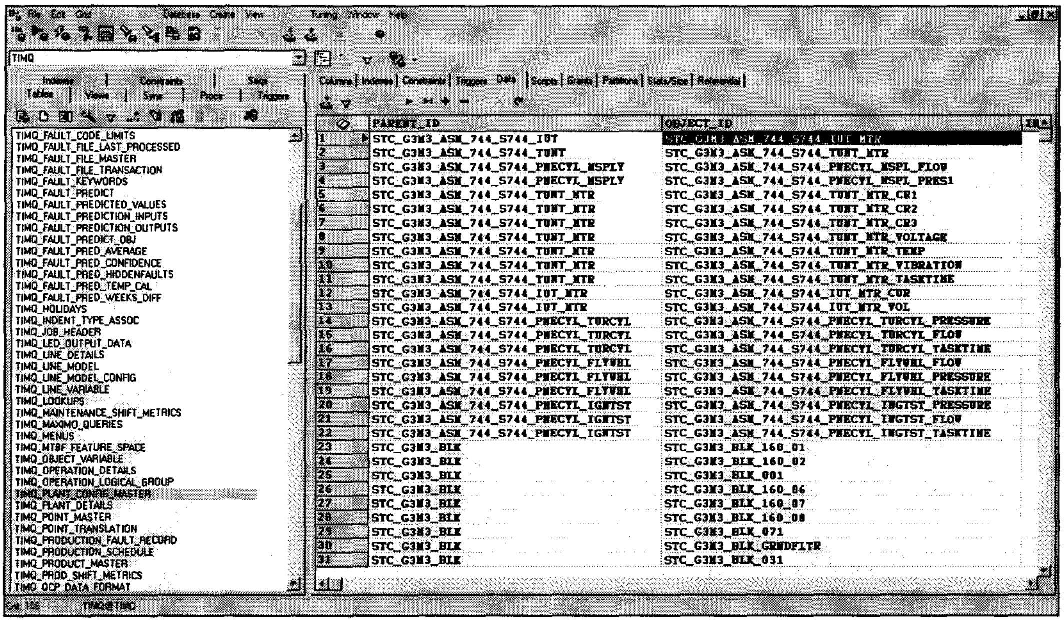


Figure 3.4 - Physical Relation between Machine Components

3.2.1 System Components and Database Models

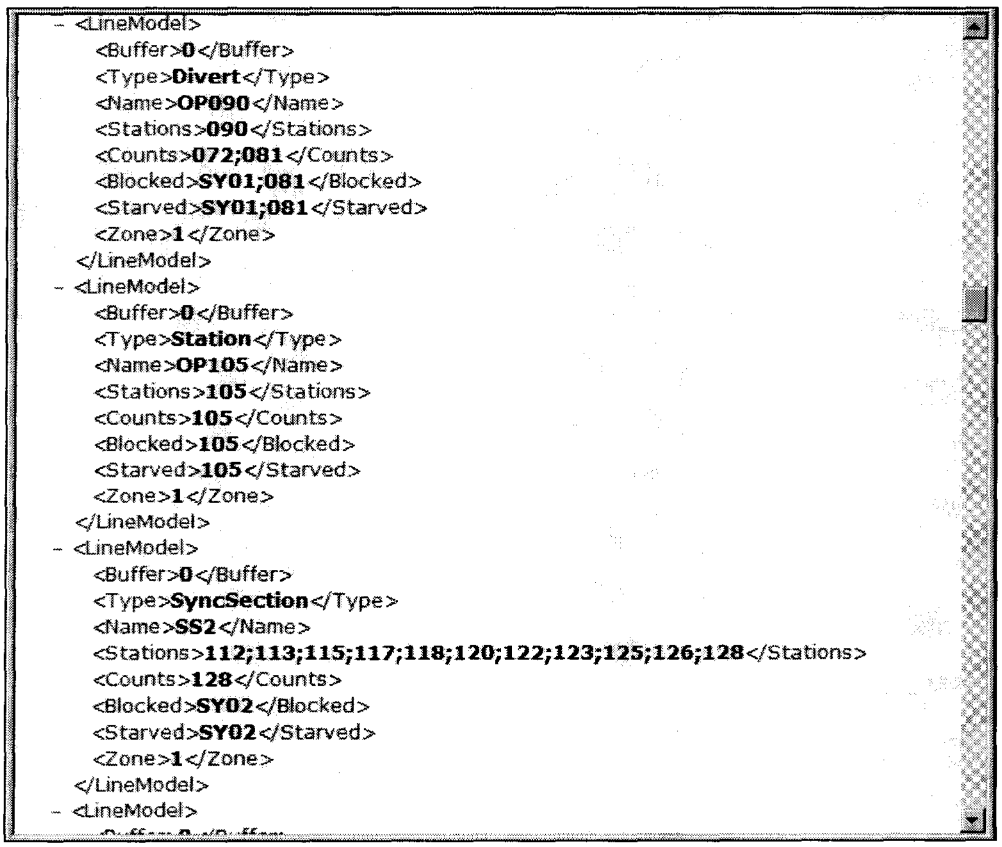
Many manufacturing applications extensively involve database management. Although object-oriented databases are perceived to be naturally more suitable for manufacture applications, relational databases are still widely used for their well-established and standardized SQL. There is a wide selection of commercial relational database management systems. Microsoft SQL server, ORACLE and SYBASE are typical RDBMS with a client – server architecture. The application in this research is based on ORACLE as its data main repository. The Database is logically divided into different modules and each of the modules & their indexes are stored in separate sections of the database memory. The large size and the required flexibility of the MES in this research involved the creation of many tables. The main classes of components are kept in separate tables. Common component attributes are stored in separate tables with mapping for data sources.

3.2.2 System Components and XML Schema

The world wide popularity of XML and its ability to integrate easily with web technologies made XML-based technologies the ideal candidate for the MES communication in this research. Using XML architecture reduced the need to develop custom solutions to address intra-applications and inter-application communication and integration.

The use of mobile AI tools enables the utilization of distributed computing power. In this research, multiple processes were executed in distributed environments. The performance bottlenecks were monitored and the required resources were devoted to the functional contexts.

Most of the AI tools which were used in this research were based on XML as the main form of input files. The XML file format proved to be very efficient as it contained the configuration of a component, the data, and the analysis parameters in a one structured file. XML file was easily extracted from the relational database and its format provided a faster way to convert its contents into a dataset rather than being parsed and then inserted in the dataset. Figure 3-5 shows a XML file that contained the topology of an engine assembly production line. This file also contained the row data which was needed for a constraint analysis to identify the location and the severity of the constraint in real time.



```
- <LineModel>
    <Buffer>0</Buffer>
    <Type>Divert</Type>
    <Name>OP090</Name>
    <Stations>090</Stations>
    <Counts>072;081</Counts>
    <Blocked>SY01;081</Blocked>
    <Starved>SY01;081</Starved>
    <Zone>1</Zone>
  </LineModel>
- <LineModel>
    <Buffer>0</Buffer>
    <Type>Station</Type>
    <Name>OP105</Name>
    <Stations>105</Stations>
    <Counts>105</Counts>
    <Blocked>105</Blocked>
    <Starved>105</Starved>
    <Zone>1</Zone>
  </LineModel>
- <LineModel>
    <Buffer>0</Buffer>
    <Type>SyncSection</Type>
    <Name>SS2</Name>
    <Stations>112;113;115;117;118;120;122;123;125;126;128</Stations>
    <Counts>128</Counts>
    <Blocked>SY02</Blocked>
    <Starved>SY02</Starved>
    <Zone>1</Zone>
  </LineModel>
- <LineModel>
```


Figure 3.5 - File with XML Format of the Line Configuration and Constraint Data

3.3 Types of Data in Manufacturing Systems

The data records in manufacturing systems are mainly time-stamped records that take one of two forms:

Event-based records:

These are data records of events that would change the state (attribute) of a component at a specific time. These records signify the behaviour observations of a component as it unfolds, but lack the resolution that is needed to represent the dynamics of the change.

The event-based records can be found in almost all manufacturing IT systems especially systems that report operational or quality failures. The records in these cases report that a certain failure occurred at a specific time. Table 3.1 shows the data of quality failures in an engine assembly line.

ENGINE NUMBER	PRODUCT SUFFIX	TESTED DATE (YYYY/MM/DD)	TESTED TIME (HH:MM:SS)	REJECTED STATION	ERROR CODE	ERROR DESCRIPTION	OPERATOR COMMENTS
	DCF	2007/08/24	05:07:15	81	8119	Missing/Damage Bolt	-- No Data --
	DCF	2007/08/24	05:03:01	84	8420	Error Proofing Verification	-- No Data --
	DCF	2007/08/24	04:48:46	87	8706	Missing Side Bolt	-- No Data --
	DCF	2007/08/24	04:22:39	87	8706	Missing Side Bolt	-- No Data --
	DCF	2007/08/24	03:20:57	87	8706	Missing Side Bolt	-- No Data --
	DCF	2007/08/24	03:14:53	87	8702	Tight Crank - Debris on Upper Bearing	-- No Data --
	DCF	2007/08/24	01:38:51	87	8706	Missing Side Bolt	-- No Data --
	ZAF	2007/08/24	01:15:25	84	8403	Studs Not Torqued #3	-- No Data --
	ZAF	2007/08/24	01:13:33	97	9720	-- No Data --	mixed block from agile pulled at op14 no
	ZAF	2007/08/24	00:47:00	87	8706	Missing Side Bolt	-- No Data --

Table 3.1 - Quality Failures Event-based Records

Time-based records:

These are records that capture the attribute value of a component at a specific sample rate. These records reduce the bias from random observing and capture more of the gradual change. It gathers piecemeal data that may miss some important events for the case of relatively slow sampling. Table 3.2 shows the production data of multiple operations sampled at the end of each production shift.

From Date	To Date	Machine Name	Downtime (Minutes)	RunningTime (Minutes)	Blocked (Minutes)	Starved (Minutes)
2007/08/22	**2007/08/22**	-OP014 - RF Tag Init.	1.0	220.0	176.0	13.0
2007/08/22	2007/08/22	-OP023 - Cam Loader	4.0	337.0	68.0	1.0
2007/08/22	2007/08/22	-OP043 - Loosen Outer Cap Bolts	3.0	206.0	110.0	91.0
2007/08/22	2007/08/22	-OP046 - Loosen Inner Cap Bolts	0.0	285.0	68.0	57.0
2007/08/22	2007/08/22	-OP050 - Remove Mains	10.0	302.0	68.0	30.0
2007/08/22	2007/08/22	-OP054 - Install Upper Bearings	9.0	202.0	120.0	79.0
2007/08/22	2007/08/22	-OP058 - Crank Loader	12.0	284.0	86.0	28.0
2007/08/22	2007/08/22	-OP063 - Install Main Bearings	10.0	310.0	46.0	44.0
2007/08/22	2007/08/22	-OP066 - Press Mains/Crank Key	1.0	216.0	76.0	117.0

Table 3.2 - Production Data as Time-based Records Sampled at the End of Each Production Shift

The focus of this research is to capture the gradual change in the performance of a component. It illustrated how production, quality and maintenance data were used to model the change more accurately when considered simultaneously rather than individually. The research also studied the cost associated with the gradual deterioration in performance and the cost of stopping the deterioration process by forcing an alternate sequence in case of component replacement. All this was done by analyzing different types of data associated with manufacturing system components. Figure 3.6 shows an example of data associated with a machine component.

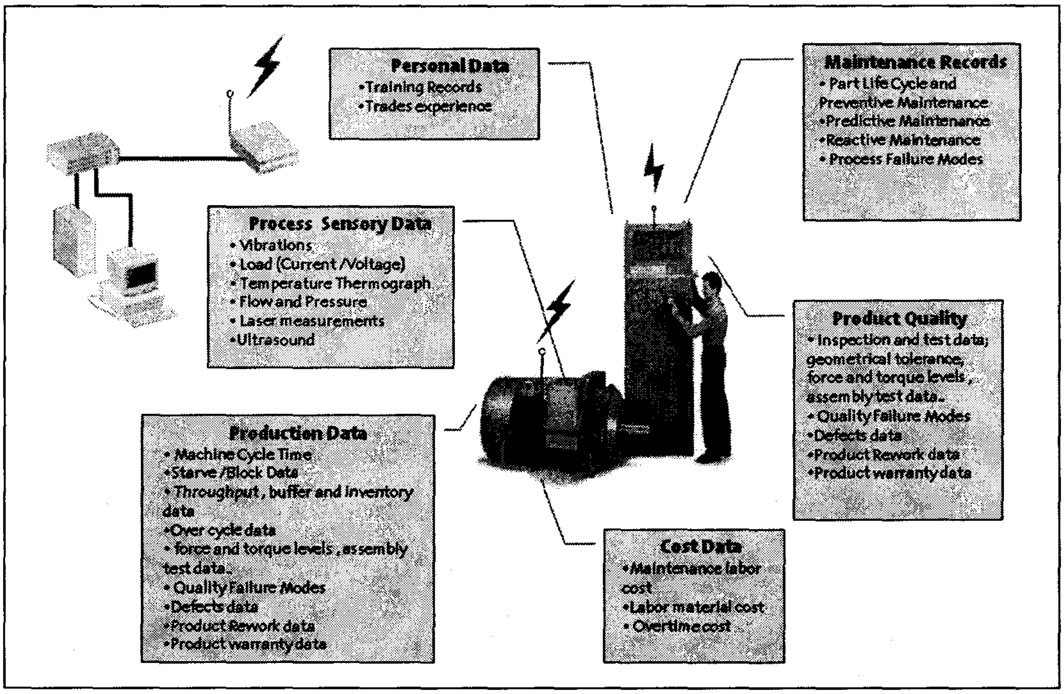


Figure 3.6 Data Associated with Machine and Its Components

3.3.1 Production Monitoring Data

Production monitoring data that is collected in a manufacturing system can be classified into two groups of variables; one-dimensional and multi-dimensional variables. The one-dimensional variables are observed or measured independently such as machine production status (running, blocked, starved or down) which is measured in time units and in number of occurrences, throughput of good parts, throughput of defective parts and machine cycle time. On the other hand, multi-dimensional variables which are calculated as a percentage of two or more variables such as effective production rate or job per hour (JPH), mean time to repair (MTTR), mean time between failures (MTBF), machine up time (efficiency). In this research only machine operational failures, machine down status, were considered to represent the component performance deterioration effect on throughput.

- Machine operational failures (machine faults)

  A manufacturing process can be considered as a sequence of controlled steps with conditions prescribed for each step which must be satisfied; otherwise an operational failure (machine fault) is reported. For example, associated with every step is a time limit. If the time taken by the step exceeds this limit, a fault is reported. Another reason why a machine fault may be reported is that if a precondition of the step changes during the process. All these faults are diagnosed automatically by the machine controller and logged into the appropriate database.

3.3.2 Quality Control Data

Quality records include product test and measurement data as well as the quality failure records. While product test data is more relevant to the manufactured product, a change in the distribution of these measurements can be a strong indicator for deterioration of performance for the components. An example of product measurement data that indicates deterioration of performance is the shift in the distribution of engine leak test measurements which may indicate deterioration in the test sealing components.

In discrete manufacturing processes, quality failures can be divided into two groups; external quality failures and internal quality failures. External quality failures represent nonconformity of a product to its specifications, observed out of the manufacturing process and reflected in customer dissatisfaction. External quality failures are usually calculated as Part per Million (PPM) also called Defects per Million Opportunities (DPMO). Other measures of quality failures include C_{pk} and P_{pk}. On the other hand, internal quality failures are product nonconformities which are observed through the manufacturing and inspection processes before the product is sent to the next customer. These records include product re-work records and scrap records. These data are captured per event or aggregated in variables such as First Time Quality (FTQ) and quality buy Rate (QBR). In this research only internal quality failure records were considered to represent effects of the performance deterioration of a component on the product quality.

3.3.3 Maintenance Records

Maintenance records include all maintenance activities to support the manufacturing processes including the type and the length of maintenance jobs as well as the details of the components that are repaired or replaced. Maintenance activities are classified in the following groups:

Reactive maintenance refers to all replace or repair actions performed on system components as a result of operational or quality failures. Reactive maintenance actions are pursued in an effort to change the state of the component back to its initial running state. Figure 3.7 shows a sample report of reactive maintenance activities where the corrective action was a component repair.

	DAYS	Brass Tag # 96387	W/O# A000441275	W/O Status COMP	Work Type EMR
OP 54 Install Upper Main Bearings			AW1 Engine Assembly 43113 Zone 1 STN54		
Original Problem	ANDON EL for AW1 Engine Assembly 43113 Zone 1 STN54				
Corrective Action	EL called to check a prox switch for bearing gripped. Adjusting at break.				
Failure Class		Target Date	Status Date 24-Aug-07		Completion Date 24-Aug-07

Figure 3.7 - Reactive Maintenance Record Associated with Machine Components

Preventive maintenance refers to a series of repair, replace or inspect actions that are performed on time-based schedule on running machines in or out of production times. Preventive maintenance actions are designed to detect, preclude, or mitigate degradation of the components in a system.

Studies by the US Department of Energy conducted in (2000) [27] concluded that reactive maintenance was the predominant mode of maintenance. The Study categorized the average maintenance program as (>55% Reactive, 31% Preventative, 12% Predictive, 2% other)

3.3.4 Cost Data

Cost data are the monetary values associated with events occurring to a component in a manufacturing system. It includes loss of production due to quality or operational failures, cost associated with high levels of inventory, cost associated with customer dissatisfaction due to external quality failures, cost of product re-work or scrap. Costs associated with maintenance activities include: reactive, predictive and preventive maintenance with both; cost of labour and cost of materials.

3.4 Types of Plant Decisions and its Relations to the Components of Manufacturing Systems

The decisions made in manufacturing systems are reflected in the system control policies. Those policies are sets of procedures, usually documented, which represent sequential reactions to specific events in order to optimize a manufacturing metric. The most common form of these procedures is a flowchart. The decision flowchart is also known as a decision tree.

The control policies and procedures cover all areas and activities in the manufacturing system. The component-based data structure is an important step in the decision making process by properly linking the decision branches to the components states. This approach not only enabled the creation of a reaction plan that responded accurately to the state of a component, but also was essential to help building the decision support tools.

The decisions are usually materialized in the form of actions which are classified into two categories: long term scheduled actions and real time actions. While decisions can be made in real time base, the value of real time decisions is only realized in the case of real time actions.

3.4.1 Production Management and Inventory Control

In the case of production management the main operational decisions are made to reduce the constraints and to eliminate bottlenecks. While production flow lines are designed to be balanced and constraint free, the dynamics and variations in the real world create production constraints that cause throughput losses. Most of the constraint analysis models in the manufacturing literature are based on the entire machine as a modeling entity which is suitable to identify the location of a constraint. However, the machine based model sometimes fails to link the constraint to the proper management decision. The component-based approach in this research adds more flexibility to the constraint analysis algorithm and provides a link to the required decisions.

A production line constraint in a discrete manufacturing process is usually generated because of one of three conditions: machine stoppage because of operational or quality failures, overcycle condition where a machine is taking more than the design rated cycle time causing a slow flow of production, or line stoppage because of a shortage in the supply chain. Figure 3.8 represents constraint analysis results with a 3D graph for an engine assembly line with locations and severity of constraints on the machine level. In this graph SS3, synchronous sections 3, is section of manually performed processes that experienced a continuous overcycle loss over the period of a shift. On the other hand, OP12 is a robot loading engine blocks that created a temporary line constraint at the third hour of the shift due to an operational failure with its end effecter. The red colour represents the throughput loss of the machine and the height represents the severity of the constraint that created block/starve effect to upstream and downstream machines. Severity scale is 0 to 1 scale; constraint with severity equals to 1 means full stoppage.

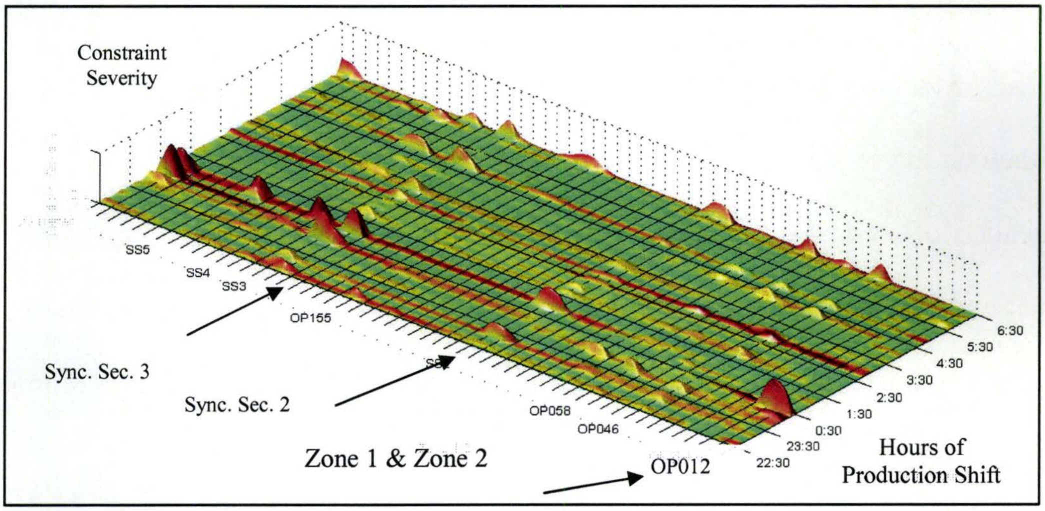


Figure 3.8 - Example of Constraint Analysis in Production Line with Machine Level

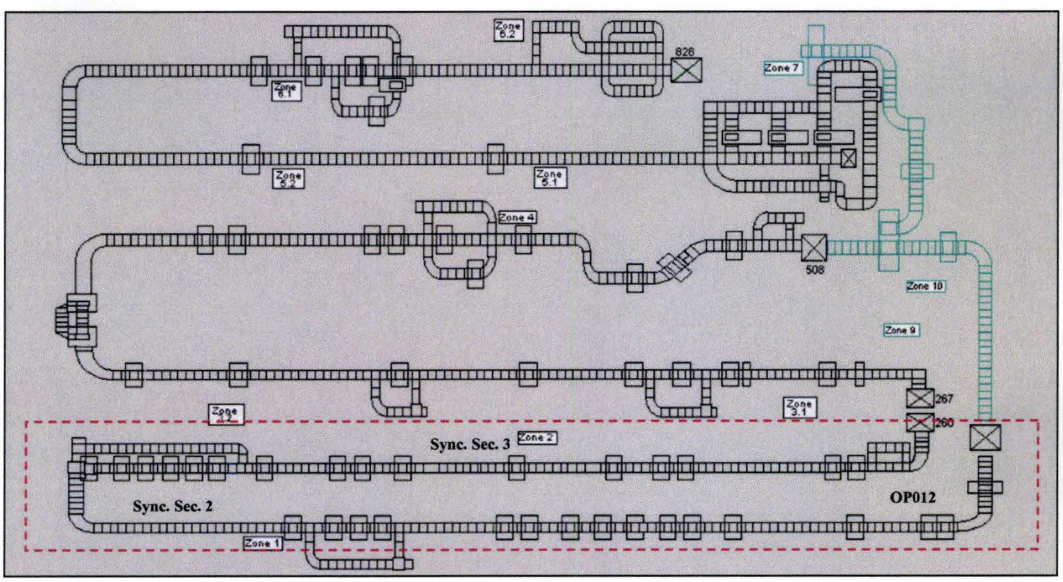


Figure 3.9 – Engine Assembly Line, Zone 1 & 2 Constraint Analysis

Real time decisions reacting to a production constraint are influential and can significantly improve throughput. These decisions are limited and usually came with an extra cost, which could only be justified if making no change is causing more losses to line throughput.

Proper maintenance activities to stop a running machine from creating high level of quality failures are among the possible decisions to reduce or eliminate a throughput constraint. Other options include change in production schedule to build product which is less sensitive to the current deteriorating state of the machine. Adding temporarily labour power is also used to overcome a slow machine. In this research only the option of initiating maintenance activities was considered.

3.4.2 Quality Control and Re-work Processes

External quality failures causing customer dissatisfaction are considered high risk. External failures could cause devastating results on the business. The method which is usually used to identify quality risk in the manufacturing processes is called Process Failure Mode and Effect Analysis (PFMEA). A Risk Priority Number (RPN) is used to identify the risk in the process that required faster reaction. While the RPN is based on variables such as the probability of quality failures and the reliability of the detection method, these variables are associated with the machine process not with the actual component. The component-based data structure used in this research properly associated the RPN to the components of interest and provided real time PFMEA.

In the case of internal quality failures, the control policy reactions are similar to the production constraint reactions which include change in schedule, more labour power or proper maintenance activity. In this research only maintenance activities were considered as possible decisions to reduce internal quality failures.

3.4.3 Diagnostic and Reactive Maintenance

Troubleshooting and diagnostic activities are the most important part of reactive maintenance. Studies in the automotive industry showed that diagnostic time in highly automated production lines in average exceeded 80% of the total maintenance time [Bandyopadhyay, Biller, Xiao, 2003 [28]]. Troubleshooting is a systematic approach to solving problems quickly and efficiently by identifying root causes which usually result in a defective component. The component-based data structure used in this research associated the process data and information to the components which was essential to providing accurate prior knowledge about the potentially defective components.

3.4.4 Prognostics and Preventive Maintenance

Prognostics are the backbone for Condition Based Maintenance (CBM) or predictive maintenance. The prediction of system behaviour is extremely valuable not only in manufacturing but in almost every business field. In the case of maintenance, a reliable prediction is the base for a better and improved preventive maintenance program. The proper association of data and information to the components of a machine is essential for building the models used for performance prediction.

3.5 Advantages of Component-based Data Structure to Represent Manufacturing System Data and Information

This section summarizes the advantages of using the component-based data structure to represent the manufacturing system data and how this representation is essential in all types of decisions made in the production lines.

- Using a single concept in modeling all conceptual entities of the system standardizes the component configuration and allows for a faster deployment process.

- The approach facilitates the construction of analysis tools that are loosely parallel to the application domain.

- Complex systems are divided into manageable data structures where data sources are associated easily and decisions are made on a modular base.

- The database structure and linkages to other manufacturing IT systems are created on the component level.

Chapter 4

Decision Support Tools in Manufacturing Systems

4.1 Decision Support Tools in Manufacturing Systems

Decision making is the process of sufficiently reducing uncertainty about alternatives to allow a reasonable choice to be made based on the values and preferences of the decision maker. It is a cognitive process that leads to an action or an opinion. The decision making process includes: assessing the problem, collecting and verifying data and information, identifying alternatives, anticipating consequences of decisions, making a choice using sound and logical judgement based on available information, informing others of the decision together with its rationale and finally evaluating the decision after implementation.

Managing complex systems are generally performed by humans who greatly depend on software applications to support their decision making processes. These software applications are typically called decision support tools functioning as modules in information systems. The performance of these tools is measured by its effectiveness in selecting the optimal decision and by its efficiency to perform on-time.

The DSS appeared as a term in the late 70's, together with managerial decision support systems. While the term covers a substantial area of managerial activities, it was excessively used by some vendors and tied to product with limited capabilities. Applications like OLAP data mining can be considered as a part of a DSS.

Decision making processes in manufacturing systems are similar to other industries in terms of its elements and its dynamics but are different in its analysis algorithms and domain knowledge. Manufacturing DSS take different forms such as MES and E-Manufacturing. It is mostly based on lean manufacturing principles. This draws heavily on the progress in the fields of artificial intelligence, information technologies and communication.

There are many decision makers in manufacturing systems with different functions and responsibilities including all layers of management and plant personnel, from the line operator all the way to the plant manager. DSS in manufacturing systems includes the ability to monitor the plant floor assets, predict the variation in performance as well as support dynamic rescheduling of production and maintenance operations. The interaction between the decision maker, the decision support tools and the system parameters is illustrated in Figure 4.1

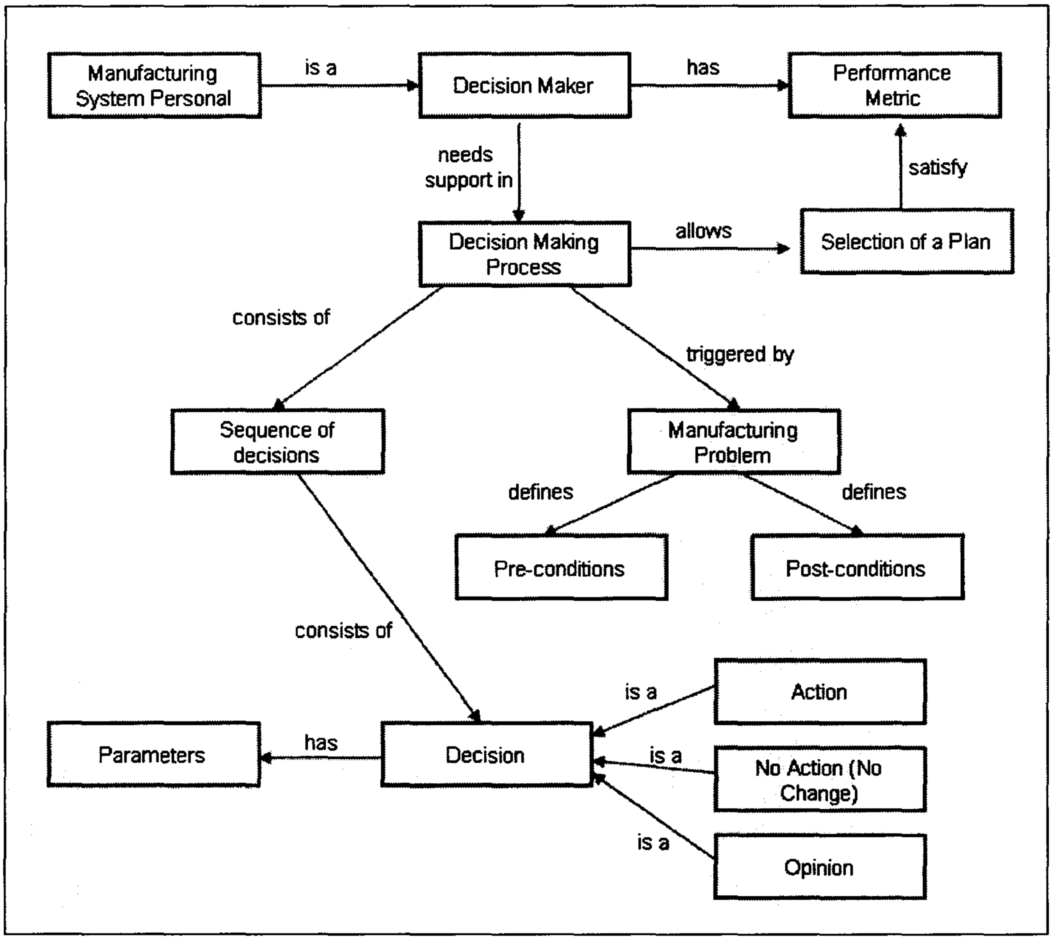


Figure 4.1 - Decision Maker Interaction with the Decision Support Tool

Figure 4-1 illustrates that the decision maker has a performance metric that should be satisfied. The decision maker needs support in the decision making process to come up with a plan which is a sequence of decisions. The decisions can lead to action or no action; adjustment or affirmation of an opinion.

4.1.1 Performance Metrics as Decision Requirements

The role of performance metrics in a manufacturing system is first to provide the business managers with a method to assess the firm's current competitive position with respect to its competitors, and to identify avenues for improvement. Second, it monitors the business's progress in moving towards its strategic objectives. The most common problem associated with traditional performance metrics is the failure to provide a strategy to achieve simultaneous objectives. The decision makers of manufacturing systems at all levels must make decisions to achieve the goals of safety, people involvement, quality, responsiveness to customer demands and last but not least manufacturing costs all simultaneously. The concurrent achievement of these goals effectively and efficiently is the goal of a manufacturing decision support system.

People are the most important asset in any business including manufacturing industries. A person's safety and involvement are fundamental for the success of a business. Any decision that jeopardises a person's safety or undermines the involvement in the business process would have long term devastating effects on the performance of the system. Metrics that measure safety include rate of recorded injuries and rate of lost business days because of personal injuries. The involvement of people can be measured by the rate of absenteeism and the rate of improvement suggestions provided by the employees. It is not within the scope of this research to elaborate more on safety and people involvement metrics but it is important to highlight their significant weight in any decision making process in a manufacturing system.

Responsiveness to a customer's demands, the product quality and the cost of the manufacturing processes are the main metrics which are greatly affected by tactical and operational decisions. Responsiveness includes measures such as production rates which are commonly measured in a discrete manufacturing system by job per hour (JPH), level of system automation measured by human hour per unit (HPU), and the uptime of a manufacturing system or its efficiency which is also related to other maintenance metrics such as MTTR and MCBF. Product quality are measured by first time quality (FTQ) and quality buy rate (QBR) for internal quality and rate of defective parts delivered to the customer part per million (PPM) as an external quality metric. The manufacturing cost includes financial measures such as cost per unit which includes all manufacturing expenses. It can also be in specialized areas of the business such as inventory and maintenance cost.

4.1.2 Knowledge Extraction in Manufacturing Systems

Extracting knowledge in a manufacturing system is the most critical and domain relevant. In this research it consists of advanced data mining and analysis tools that integrate all related information and data into domain knowledgebase. The cognitive hierarchy consists of three levels.

Level 1: Data – Raw data is observed and collected from many data sources of the manufacturing components. This data is structured and includes both historical and real time data.

Level 2: Information – Data which is analyzed and organized into forms that report on certain aspects of the manufacturing system or its components. The analysis that transforms data into information mostly involves data grouping and sorting over individual datasets.

Level 3: Knowledge – Information which is integrated from multiple sources. It includes the information from the data analysis and the prior information associated with the manufacturing components. Figure 4.2 shows the data, information and knowledge transformation model and its interface with other components of business model.

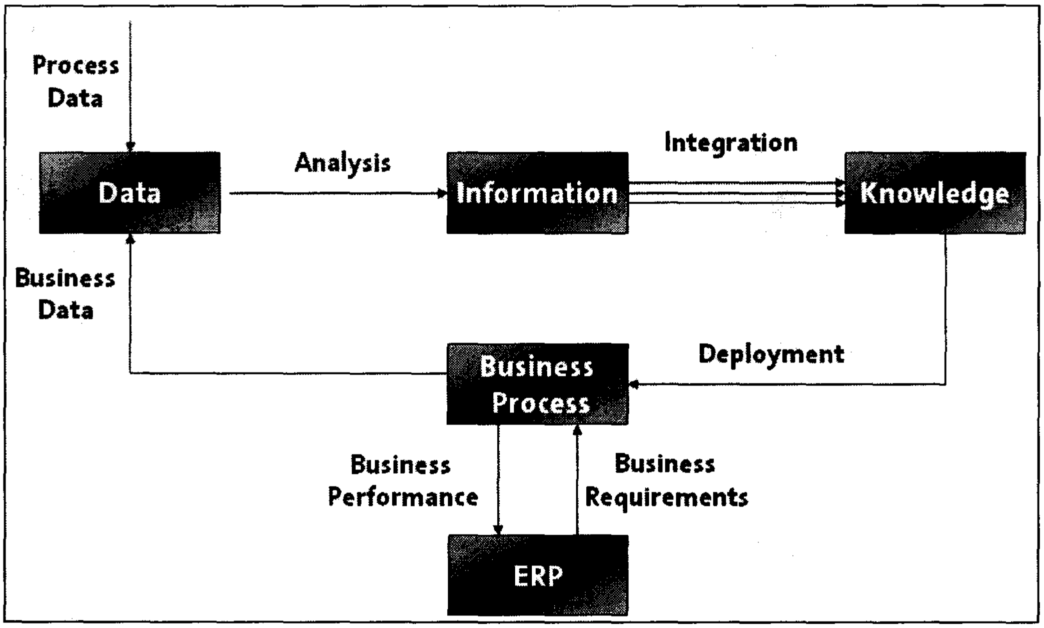


Figure 4.2 - Data, Information and Knowledge Transformation Model in Manufacturing Systems

Figure 4-2 shows that ,in general, there are two groups of data; manufacturing processes data which is generated from the interaction between the manufacturing system components with raw materials and the business data which is the business requirements.

4.1.3 Software Agents

Software agents are the most appropriate software applications for DSS. In this research, AI software applications were used to transform information into knowledge and agents were used for deploying the knowledge in the business model. A software agent is a persistent software entity that waited in the background and performed an action when a specified event occurred, using artificial intelligence in the pursuit of the goals of its clients. Its actions simulate a human relationship by doing something that another person can do. The agents have some key cognitive characteristic.

Autonomy: agents are independent and able to perform the required tasks without the direct intervention of humans or other agents; the agents have control over their actions. The research platform included some agents that functioned as independent modules whether in monitoring the components of a system, or reporting and alerting decision makers.

Social ability: agents are able to interact with other software agents and humans in order to complete their tasks and to help other agents with their activities where it is appropriate. The agents in the research platform interacted with an alert agent who was responsible to alert the human decision maker of the situation and recommended sequence of actions based on the decision maker's preference.

Responsiveness: agents respond in a timely fashion to changes which occur in their space and are proactive in its predictive capabilities.

4.2 Comparison of Decision Support Tools

Decision support tools are used in many applications in manufacturing systems to interpret data to infer situation descriptions, use predictions to deduce likely consequences, conduct diagnosis to conclude system malfunctions, continue to monitoring to compare observations, and debug to prescribe remedies for malfunctions. DSS tools are different in many aspects.

4.2.1 Types of Data used in Analysis Tools

Some of the DSS tools use one type of data while others use multiple sources of data. The data which was used in the analysis tools in this research was different in its sampling rate and whether it was generated from a stationary process or from a process with time-varying statistical parameters.

4.2.2 Decision Time Frame

Decisions in manufacturing systems are classified temporally into three categories: strategic, tactical, and operational according to the time horizon of the decisions.

Strategic Decisions target the long-term objectives of the manufacturing system. It guides policies from a design and planning perspective. Strategic decisions influence the long term competitive status of the business based on its performance metrics. They also affect technical design aspects such as reliability requirements; its automation and its labour levels together with its product quality assurance strategies. The time horizon of these decisions is several years and depends on many factors.

Tactical Decisions are the decisions that are required to effectively manage the manufacturing system according to strategic level decisions. These decisions establish the immediate performance goals of the manufacturing system and make the plan to achieve these goals. These decisions are usually made by business managers. It covers all activities of a manufacturing system such as modifying the production rate to meet the takt time, changing quality control procedures to meet target FTQ and reducing inventory size to improve cost per unit metric. The time intervals of tactical decisions range from weeks to months.

Operational Decisions are short-term decisions and are generally focused on real-time activities. The effectiveness and efficiency of making operational decisions has significant impact on the performance of a manufacturing system. These decisions include variation reduction of operations parameters such operational failures, quality failures and reactive maintenance management.

4.2.3 Real Time Data and Real Time Decisions

If the information is delivered immediately after data collection, the data is called real time data; otherwise the data is called historical data. Operational decisions are usually made on sets of both real time and historical data. On the other hand, tactical and strategic decisions are generally made on historical data.

Operational decisions are considered real time decisions if they initiated real time actions in response to some variation in the performance metrics. The benefits of real-time decisions are clearly demonstrated in two areas of plant-floor systems; production scheduling control and maintenance resource allocation.

In this research, operational failures were the most common events that affected the manufacturing system where a defective component of a machine stopped a production line. There are several plant floor systems that are simultaneously used in the decision making process to generate a control plan. The plan consist of sequence of actions including alerting the production and maintenance groups, collecting the details of the operational failures, looking for patterns of fault sequences and providing the maintenance group with real time on-line recommendation. Figure 4.3 presents the utilization of a software agent based on a temporal data mining technique to find patterns of fault sequences in a real time manufacturing system. Figure 4.4 presents the parameters of a fault pattern including the number of correlated operational failures and the confidence in the order of event sequence.

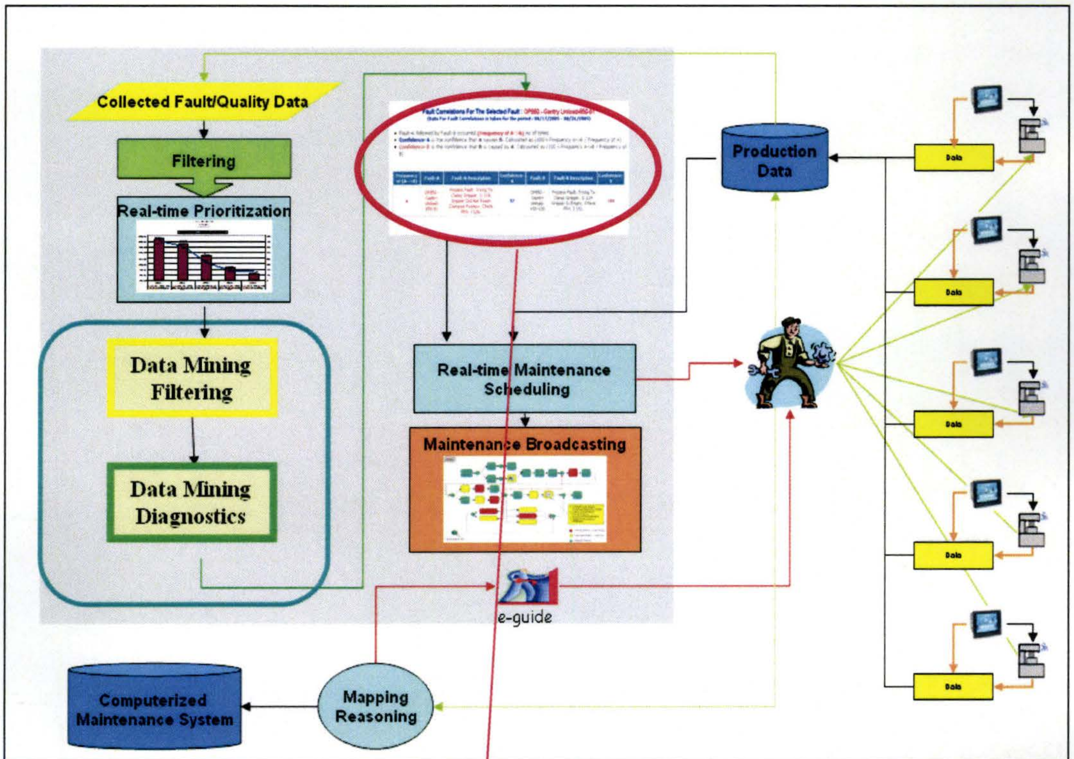
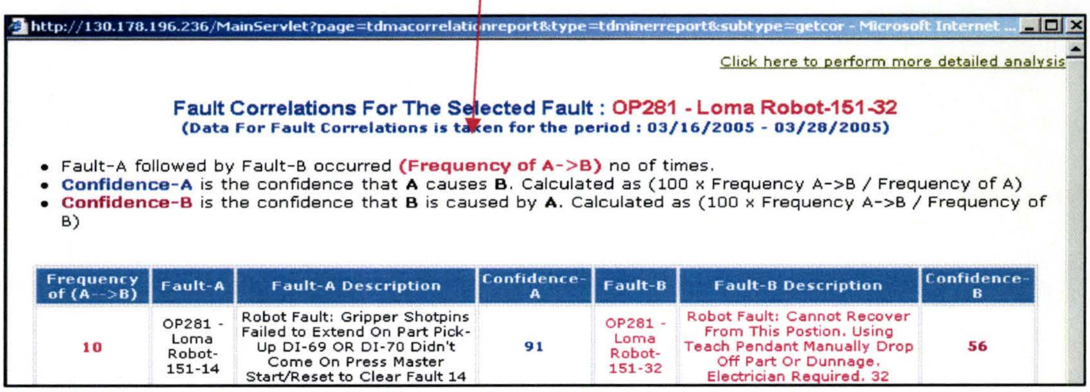


Figure 4.3 - Real Time Data Mining for Patterns of Fault Sequences



http://130.178.196.236/MainServlet?page=tdmacorrelationreport&type=tdminerreport&subtype=getcor - Microsoft Internet ...

Click here to perform more detailed analysis

Fault Correlations For The Selected Fault : OP281 - Loma Robot-151-32
(Data For Fault Correlations is taken for the period : 03/16/2005 - 03/28/2005)

- Fault-A followed by Fault-B occurred **(Frequency of A->B)** no of times.
- **Confidence-A** is the confidence that **A** causes **B**. Calculated as (100 x Frequency A->B / Frequency of A)
- **Confidence-B** is the confidence that **B** is caused by **A**. Calculated as (100 x Frequency A->B / Frequency of B)

Frequency of (A-->B)	Fault-A	Fault-A Description	Confidence-A	Fault-B	Fault-B Description	Confidence-B
10	OP281 - Loma Robot-151-14	Robot Fault: Gripper Shotpins Failed to Extend On Part Pick-Up DI-69 OR DI-70 Didn't Come On Press Master Start/Reset to Clear Fault 14	91	OP281 - Loma Robot-151-32	Robot Fault: Cannot Recover From This Postion. Using Teach Pendant Manually Drop Off Part Or Dunnage. Electrician Required. 32	56

Figure 4.4 - Parameters of a Fault Episode

4.3 Limitations in Decision Support Tools

The raw data and information about a process usually contain uncertainty in the form of inaccurate and/or incomplete records. Inaccurate data refers to data records that do not conform to the true value while inaccurate information refers mainly to the raw information that is fuzzy, conflicting or contradicting. On the other hand, incomplete data refers to missing records in the dataset which is caused by the unavailability of equipment or the oversight of operators during data collection. Missing information refers to missing heuristic prior knowledge about the system. These inaccurate and incomplete data and information are the greatest obstacles to automated knowledge extraction processes.

Other limitations of the DSS include the growing complexity of the analysis algorithms. This complexity significantly weakens the confidence interval and eventually makes the DSS unable to reduce the uncertainty to practical decision making levels. Maintaining a small search space with reliable knowledge and data is also a limitation for the DSS.

4.3.1 Inaccuracy in the Data

Most of the data in manufacturing systems, including the data which was investigated in this research, is temporal data records representing events with specific durations. Accuracy in that data is the closeness of the results of the observations to the true values or values which are accepted as being true. The inaccuracy in the data records is either in the time frame of the event or the details (contents) of the data record. In the research case study, the time associated with the data collection processes was very accurate, in millisecond in its worst case. However, there was a relatively high level of inaccurate and wrong contents in some data records. The main reason for incorrect content was programming errors in the controls system or programming errors in the data collection system.

4.3.2 Missing Data

Missing data records in this research were usually caused by communication outages between the control system and the data collection system. Very few devices had an internal buffer or local database to overcome the data loss due to communication problems. Data records which were based on human data entry were generally partially complete in most cases.

There are different techniques to deal with missing values in empirical learning. These techniques can be generally grouped into three main categories as follows:

- Eliminating training examples with unknown attribute values.

- Viewing the unknown value of an attribute as an additional special value, denoted as "not known".
- Using probability theory which replaces the unknown attribute by the most probable value. During the estimation process, the missing values are assumed to be missing at random.

In this research, incomplete data records were eliminated from the training dataset.

4.3.3 Noise

Noise in a data collection process is the addition of irrelevant factors to the stream of data being collected in the database. Usually these irrelevant data records are generated by a malfunctioning data collection system which generates cases of duplicate records, partial records, mixed records or records with garbage data. Many software filters were set in this research to eliminate noise in the database and in the analysis tools.

4.3.4 Limitations of Data Collection and Complexity of the Analysis

While data collection processes have improved significantly in today's manufacturing systems, there were some important events that were not fully captured in the research case study. The reasons for this were limitations in data storage, limitations in communication bandwidth and the high cost of data collection systems.

The complexity of analysis algorithms intensified significantly because of the increasing number of variables in a manufacturing system which included: product mix, demand variability, flexibility and the reconfigurability of the machines. On-line and real-time decision making becomes an enormous knowledge processing task which can overburden conventional computing devices. These limitations create some tradeoffs in DSS performance such as decision quality versus response time. The more comprehensive as well as sophisticated the DSS analysis tools and architecture are, the more likely the decision quality incrementally improve. Another trade-off is the optimal solution versus an acceptable solution.

4.3.5 Impractical Business Deployment

The most important part of the decision support system in this research was the proper and practical deployment in the business model. Identifying domain knowledge and business alternatives consistent with a business model is in many times a complicated task. It depends on the business procedures and control policies. It is common to see some contradictions between the business procedures as they are prepared by different business teams. Another challenge is the knowledge presentation. While most of the advanced decision support systems are web based and accessible through mobile computing devices, there are still great technical difficulties in setting an interactive system with social ability to support decision makers in a timely manner. Some DSS tools require many inputs from the decision makers to reduce the uncertainty of the situation and to structure the problem. This lengthy process moves many decision makers away from using this tool and thus makes the tool impractical to implement.

4.4 The Research Platform

The structure which was outlined in chapter 3 and 4 in this thesis was used to produce a web-based Java solution. A Manufacturing Execution System that was successfully deployed in an automotive production line and served as a DSS research platform. The internet computing model, web portals, server-side configuration, database, security strategies were key functional points. The development of web portals was done using HTML/JSP techniques. The Web database appeared in a distributive mode. It provided a huge pool to support all platform tools to generate, exchange, and store manufacturing data. It was also located on the server-side. An SQL-driven server and JDBC package were used to implement it.

Chapter 5

Frequent Episodes in Event Sequences and Hidden Markov Modeling

5.1 Modeling the Deterioration Process in the Performance of Manufacturing System Components

The main hypothesis in this research is that the deterioration process in the performance of a manufacturing system component can be modeled as a Markov process. The sequence of deterioration states is considered as a latent variable and inferred by the sequence of observations of measurable variables, which are the component attributes. The second part of the hypothesis is that the decisions related to manufacturing system components are based on the component performance state. Accordingly, only attributes that initiate the decision process are included in the estimation of performance state.

In the case of a component of a machine, the optimal decision of replacing the component depends on the performance state. In many cases, the components have several failure modes with different sequences of deterioration. Frequent episode analysis was used in this research to identify sequences of performance deterioration for each failure mode. The work in chapter 3 provided the data structure to identify the relevant events associated with the deterioration patterns of a component. The work in chapter 4 discussed the types of decisions and the control policies that controlled the decision making process.

In Section 5.2 the theory of frequent episode in event sequences is discussed. The concept and the algorithms used to identify composite episodes are also explained. Section 5.3 introduces the theory of HMM and the algorithms used to find HMM parameters, specially the expectation maximization method which was used in this research.

5.2 Frequent Episodes in Event Sequences

Data mining is the analysis of datasets to find unexpected relationships and to summarize the data in different ways that are both relevant and interesting to the decision makers [Laxman, 2006 [30]]. Temporal Data Mining is the mining of sequentially (or temporally) ordered data using the frequent episodes discovery framework to discover hidden relations between sequences and sub-sequences of events. A sequence is composed of a series of nominal symbols from a particular alphabet which is usually called a temporal sequence. A sequence of continuous, real-valued elements is known as a time series. Data mining is part of the artificial intelligence field that is referred to as Knowledge Discovery in Databases (KDD).

Data mining consists of interdisciplinary techniques from statistics, database technology, machine learning and pattern recognition.

Frequent episodes in event sequences is an important data mining application area where the data to be analyzed consists of a sequence of events. A change occurring in the life cycle of a manufacturing system component is usually observed by a sequence of events with different durations. An example of an event sequence is represented in Figure 5.1. A, B, C, D and F are event types which can represent different types of operational failure associated with a component in a manufacturing system marked on a time line.

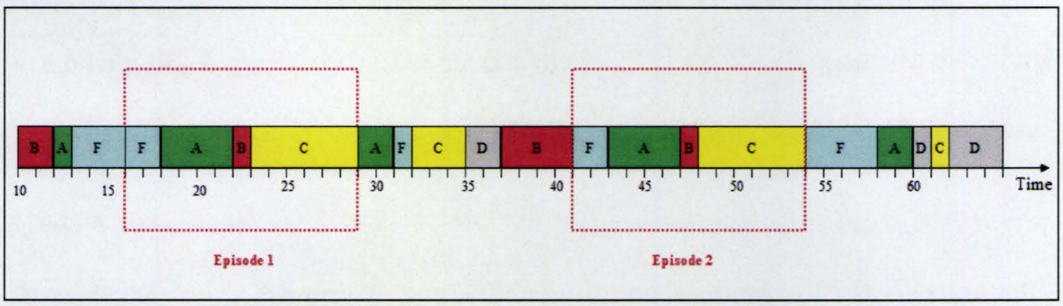


Figure 5.1 - Frequent Episodes in Event Sequences. The Alphabets Represent Events Types. The Time line Shows Event Durations

An episode is an ordered sequence of event types. For example, (A → B → C) is a 3-event episode. An episode is said to occur in an event sequence if there are events in the sequence which have the same time ordering as that specified by the episode.

As it was discussed in chapter 3 of this thesis most of the data records associated with the components in manufacturing systems are time stamped and represent a change in the state of a component or its attributes at a certain period of time. The events dataset used in this research is considered as a long sequence of ordered pairs, (A_i, t_i). In each event, (A_i, t_i), A_i is referred to as an event type and t_i is the time of occurrence of the event using the same notation and terminology of [Mannila, Toivonen, Verkamo, 1997 [14]]. The event type can actually contain several attributes; only a single valued event type was considered in this research.

Let E denotes a finite set of event types, $E = \{E_1, E_2, \ldots\ldots E_m\}$, and n be the number of events in the dataset. An event sequence $\boldsymbol{s}$ is a triple (s, T_s, T_e), where $s = \langle(A_1, t_1), (A_2, t_2), \ldots\ldots(A_n, t_n)\rangle$ is an ordered sequence of events such that $A_i \in E$ for all $i = 1, \ldots\ldots n$, and $t_i \leq t_{i+1}$ for all $i = 1, \ldots\ldots n\text{-}1$. T_s and T_e are integers (time unit): T_s is called the starting time and T_e the ending time, and $T_s \leq t_i < T_e$ for all $i = 1, \ldots\ldots n$

For example, the two event sequences in Figure 5.1 consisted of four events: $\langle(F, 16), (A, 18), (B, 22), (C, 23)\rangle$ *and* $\langle(F, 41), (A, 43), (B, 47), (C, 48)\rangle$

An episode α can be defined as a triple $(V_\alpha, \leq_\alpha, g_\alpha)$ where V_α is a set of events, $\leq_\alpha$ is a partial order on V_α, and $g_\alpha : V_\alpha \rightarrow E$ is a mapping associated with each event of an event type. The interpretation of an episode is that the events in $g_\alpha(V_\alpha)$ have to occur in the order described by $\leq_\alpha$. The size of α, denoted $|\alpha|$, is $|V_\alpha|$.

- A parallel episode is a set of event types for example {A, B, C}. In diagrams, these episodes are represented by a vertical box with A, B, and C within. The intent of a parallel episode is that each of the events in the episode occurs (within a window of time), but the order is not important. Formally, episode α is parallel if the partial order $\leq_\alpha$ is trivial (x *is not* $\leq y$ for all $x; y \in V_\alpha$ such that $x \neq y$)
- A serial episode is a list of event types for example (A, B, C). In diagrams, these events are shown in a horizontal box, in order. The intent is that within a window of time, these events occur in order. Formally, episode α is serial if the relation $\leq_\alpha$ is a total order ($x \leq y$ or $y \leq x$ for all $x; y \in V_\alpha$).
- A composite episode is built recursively from events by serial and parallel composition. A composite episode is either: an event, the serial composition of two or more events, or the parallel composition of two or more events.

Figure 5.2 shows an example of a composite episode. It is the serial composition of three episodes. The first is the single event A. The second is the parallel episode {B, C, D}. The third is a composite episode consisting of the parallel composition of the serial episodes (E, F) and (G, H). Two examples of orders of these 8 events that are consistent with this episode are ABCDEGFH and ACDBGHEF

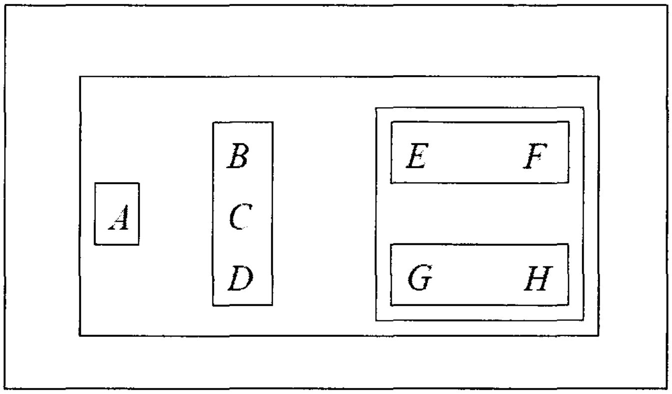


Figure 5.2 – Example of Composite Episode which Consists of Multiple Serial and Parallel Episodes

The sequence mining algorithms are categorized into one of three broad classes that perform the task regardless of the format of the dataset, Apriori-based horizontal database format, Apriori-based vertical database format or projection-based pattern growth.

Constraints to guide the mining process are employed by many algorithms, not only in sequence mining but also in association mining [Mooney, 2006 [31]]. A constraint C on a sequential pattern is a Boolean function of the form C(α). The constraints include: Item Constraint, Length Constraint, Model-based Constraint, Aggregate Constraint, Regular Expression Constraint, Duration Constraint and Gap Constraint.

There is a considerable amount of research and algorithms to discover frequent episodes which operate on time stamped data. The major difference between the traditional algorithms and the approach which was taken in this research is the candidates generation and counting phases. The main algorithms are discussed in the following section.

Generalized Sequential Patterns (GSP) algorithm is designed for transactional data where each sequence is a list of transactions ordered by transaction time and each transaction is a set of items. Timing constraints such as maximum span, event-set window size, maximum gap, and minimum gap are applied in this approach. The algorithm finds all sequences that satisfy these constraints and whose support is greater than the user-specified minimum. The support counting method used is COBJ (One occurrence per system component). This algorithm consists of two phases: i) the first phase scans the database to identify all the frequent items of size one, and ii) the second phase is an iterative phase that scans the database to discover frequent sequences of the possible sizes. The second phase consists of the candidate generations and pruning steps wherein sequences of greater length are identified; sequences that are not frequent are pruned out from further iterations.

Episode Discovery (ED) is a data mining algorithm that discovers behavioural patterns in a time-ordered input sequence. This approach is based on the

Minimum Description Length (MDL) principle and discovers multiple characteristics of the patterns such as its frequency, periodicity, order and the length of a pattern. It uses a compression ratio as the evaluation measure since a greater compression ratio results in a shorter description length.

WINEPI algorithms are designed for discovering serial, parallel or composite sequences that represent a frequent episode. In addition to the above, events of the sequences must be close to each other, which is determined by the window parameter. A time window is slid over the input data and only the sequences within the window are considered. The support for the sequence is determined by counting the number of windows in which it occurred. This approach along with the **MINEPI** approach is used in this thesis and is discussed in more detail below. The algorithm MINEPI follows the same basic principles as WINEPI with the exception that minimal occurrences of candidate episodes are located during the candidate generation phase of the algorithm.

The framework of frequent episode discovery using the WINEPI approach is a temporal pattern mining technique that is applied on event sequences. To be considered interesting, the events of an episode must occur close enough in time which is relative in the time scale. There are two variables that should be known to find the frequent episodes; the width of the time window within which the episode must occur and the number of windows an episode has to occur to be considered frequent. A window is defined as a slice of an event sequence. In other words, a window on an event sequence **s** = (s, T_s, T_e) is an event sequence **w** = (w, t_s, t_e), where $t_s < T_e$ and $t_e > T_s$, and w consists of those pairs (A, t_i) from s where $t_e < t_i < t_e$. The time span $t_e - t_s$ is called the width of the window **w**, and it is denoted width (**w**). The number of windows is determined by the width of the window. Given an event sequence **s** and an integer *win*, the set of all windows **w** on s such that width (w) = *win* is denoted by $\mathcal{W}$(s, *win*). The example at Figure 5.1 shows two windows of width 13 on the sequence **s**. A window starting at time 16 is Episode 1.

$\langle(F, 16), (A, 18), (B, 22), (C, 23), 16, 28\rangle$

5.2.1 Episode Occurrence

An episode occurs in an event sequence (s, T_s, T_e) if there are events in the sequence whose time of occurrence conform to the order prescribed in the episode. Formally, given a data sequence $\langle(A_1, t_1), \ldots, (A_n, t_n)\rangle$, an occurrence of episode $\alpha = (V_\alpha, \leq_\alpha, g_\alpha)$ is a map $h: V_\alpha \rightarrow \{1, \ldots, n\}$ (where n is the number of events in the sequence **s**) such that for all $v, w \in V_\alpha$:

$$g_\alpha(v) = A_{h(v)} \qquad (5\text{-}1)$$

$$v \leq_\alpha w \text{ in } V_\alpha \Rightarrow t_{h(v)} \leq_\alpha t_{h(w)} \qquad (5\text{-}2)$$

An episode can also be a sub-episode of another; this relation is used extensively in the algorithms for discovering all frequent episodes. Let $\alpha = (V_\alpha, \leq_\alpha, g_\alpha)$ and $\beta = (V_\beta, \leq_\beta, g_\beta)$ be two episodes. β is said to be a sub-episode of α (denoted $\beta \preceq \alpha$) if there exists a $1-1$ map $f_{\alpha\beta} : V_\beta \rightarrow V_\alpha$ such that, $g_\beta(v) = g_\alpha(f_{\alpha\beta}(v))$ for all $v \in V_\beta$, and for all $v, w \in V_\beta$ with $v \leq w$, and $f_{\alpha\beta}(v) \leq f_{\alpha\beta}(w)$ in V_α.

In other words, $\beta \preceq \alpha$ if the entire event types in β appear in α as well, and if the partial order among the event types of β is the same as that for the corresponding event types in α.

5.2.3 Frequent Episode Algorithms

There is a considerable body of work on sequential mining of data, most of them deal with time point data and make several iterations over the entire dataset for discovering frequently occurring patterns.

The statistical significance of frequent episodes is measured by episode frequency or as a number of occurrences. Frequency of an episode can be defined as the fraction of windows in which the episode occurs. That is, given an event sequence s and a window width *win*, the frequency of an episode α in s is the number of windows in which α occurs divided by the number of total windows of width *win* from **s** which can be represented as follows:

$$fr(\alpha, \mathbf{s}, win) = \frac{|\{\mathbf{w} \in \mathcal{W}(\mathbf{s}, win) \mid \alpha \text{ occures in } \mathbf{w}\}|}{|\mathcal{W}(\mathbf{s}, win)|} \tag{5-3}$$

The approach used in this research consisted of applying the Apriori-style level-wise algorithm in sequences. The goal was to find out which were the interesting frequent episodes; so, in every processed window, the frequency of those episodes that fit in the window was incremented whereas the frequency of those episodes that did not occur in the window was decremented. This process started with discovering frequent 1-node episodes. These episodes were then combined to form candidate 2-node episodes and then by counting their frequencies, 2-node frequent episodes were obtained.

This process was continued till frequent episodes of all lengths were found. Like in the Apriori algorithm, the candidate generation step declared an episode as a candidate only if all its subepisodes had already been found to be frequent.

5.2.3.1 Candidate Generation

The most important part in this step is to designate an episode as a candidate only if all the subepisodes obtained by dropping a node from it have already been found to be frequent.

In order to build candidate (k + 1)-node episodes, the candidate generation step requires, as input, the set, $\mathcal{F}_k$, of frequent k-node episodes. An episode is represented simply as an array of event types. In case of parallel episodes, entries in the array are maintained in sorted order, while for serial episodes, entries in the array are stored according to the total order prescribed by the episode. Episodes in $\mathcal{F}_k$ are organized into blocks such a way that any two episodes in a block had the same first (k − 1) entries.

Potential candidates are then identified by combining pairs of episodes within each block. For the case of parallel episodes, let α and β be two k-node frequent parallel episodes in the same block of $\mathcal{F}_k$ with α appearing ahead of β in lexicographic order. These are then combined to generate a potential (k + 1)-node candidate, say γ, as follows. The k event types of α constitute the first k entries of γ. The $(k + 1)^{\text{th}}$ entry in γ is set equal to the last (i.e. k^{th}) event type of β. Then γ is declared a candidate if all of its k-node subepisodes are already known to be frequent (i.e. if they can be found in $\mathcal{F}_k$).

In case of serial episodes, the same process is used to generate candidates, except that, now α and β are combined to generate not one but two potential candidates – the first is obtained as shown earlier, while the second is obtained by copying β into the first k entries of γ and setting the last event type of α as the $(k + 1)^{th}$ entry of γ. The time complexity of such a candidate generation procedure is shown to be a polynomial of the size of the set of frequent episodes [Laxman, 2006 [30]].

5.2.3.2 Frequency Counting

Parallel episodes are treated like item sets and so counting the number of sliding windows in which they occurred is much like computing the support of an item set over a list of transactions. An $O((n + l^2)k)$ algorithm is used for computing the frequencies of a k-element set of l-node parallel episodes in an event sequence of length n. Counting serial episodes, on the other hand, is more involved. This was because, unlike for parallel episodes, a mechanism to find ordered sequence is needed to recognize serial episodes. That mechanism corresponding to an episode accepted l-node episode and rejected all other input. For example, for the episode (A $\rightarrow$ B $\rightarrow$ C), there would be a 3-state mechanism that transits to its first state on seeing an event of type A and then waits for an event of type B to transmit to its next state and so on. When this mechanism is transited to its final state, the episode is then recognized (to have occur once) in the event sequence. Such mechanisms are needed for each episode whose frequency is being counted.

For a k-element set of l-node serial episodes the algorithm has $O(lk)$ space complexity. The corresponding time complexity is given by $O(\Delta T lk)$, where ΔT denotes the number of window-shifts the algorithm carries out over the event sequence [Mannila, Toivonen, Verkamo, 1997[14]].

5.2.4 Applications of Frequent Episode Algorithms in a Manufacturing Database

The framework of frequent episodes in event streams provided a good abstraction for mining useful temporal patterns from time-ordered data. In case of the manufacturing plant data, there are many applications where frequent episodes analysis is used to identify meaningful sequences. Operational failures, product quality failures, machine status and maintenance records are among the fields where frequent episodes analysis is used. However, the existing applications in manufacturing systems are restricted to episodes of one type of data representing the event type.

5.2.4.1 Composite Episodes of Manufacturing Events

The research in this thesis explored the frequent episodes that involved multiple record types sharing the same timeline and creating overlapping events. Records from different event types were converted to one dataset with new event type based on the composite episodes. Figure 5.3 represents an abstraction of the dataset that was considered in this research. The dataset consisted of three record types of data which were operational failure, quality failures and maintenance activities. The data was temporal in its form and represented events with types and its time (A_i, t_i). The task of the data mining step was to identify the composite episodes in the dataset.

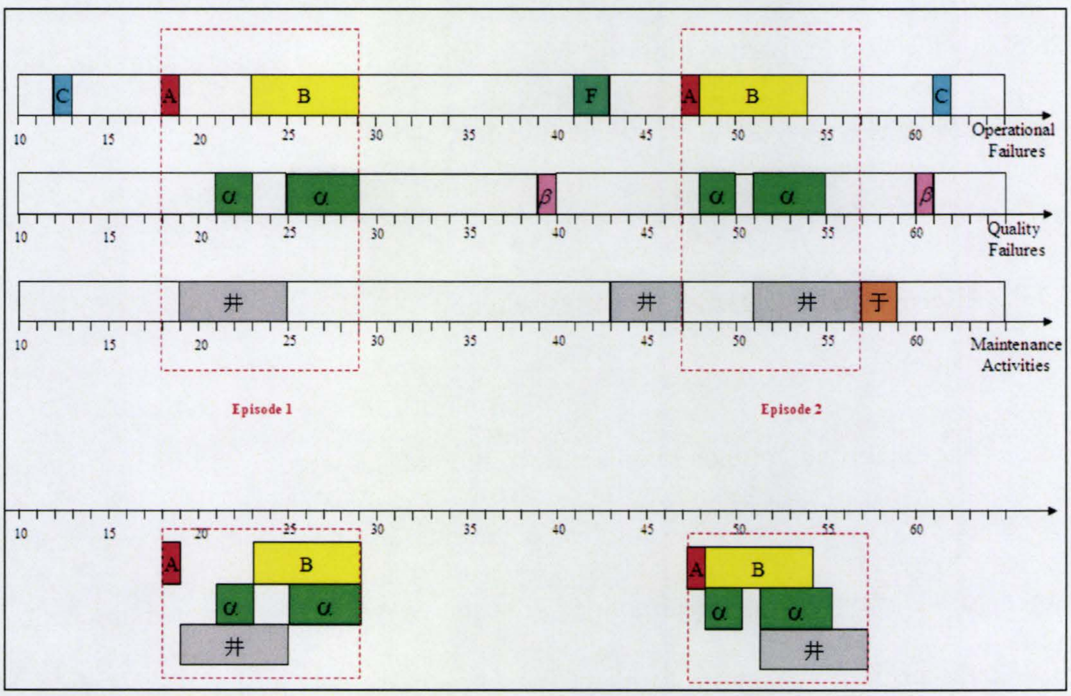


Figure 5.3 - Composite Episodes in Manufacturing Dataset. The Composite Episodes Combines Operational Failures, Quality Failures and Maintenance Activities

The no-event periods in the dataset which are represented with the white unlabeled box in Figure5.3 are of very long duration in comparison with the failure events.

There is little attention to the discovery of composite episodes in the literature. The main reason for that is the number of frequent patterns that quickly explodes for composite episodes. The algorithm used in this research to identify the composite episodes was based on a combination of parallel and serial algorithms. It was very practical as it handled all episodes like parallel episodes and checked the correct partial ordering only when all events are in the window. Parallel episodes were located efficiently; after they have been found; thus checking the correct partial ordering was relatively fast. The algorithms of the frequent episodes are explained in detail at Appendix 1.

5.2.4.2 Hierarchal Frequent Episode Discovery

The identified composite episodes in the dataset were used to construct a new dataset that reduced the number of events in the original dataset and projected all events over one timeline. A second layer of a restricted frequent episode algorithm was used to identify serial episodes in the new dataset. Maintenance events such as component replacement or major repairs were used as a restriction on the potential window size as the objective of the analysis was to identify the possible sequence of composite episodes over the life cycle of a manufacturing system component as represented in Figure 5.4.

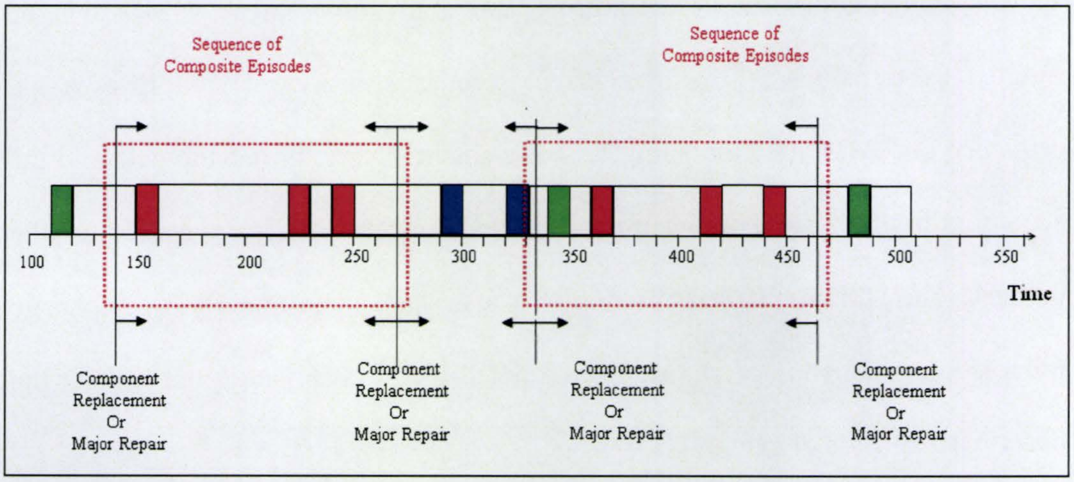


Figure 5.4 - Sequence of Composite Episodes over the Life Cycle of the Component

The case study shows that some components of a machine had a finite number of change patterns through their life cycles and sequences of composite episodes were identified within these life cycles. The dataset was divided then into segments that have the same change (deterioration) pattern based on the sequences identified. Each segment was used to train HMM. The number of composite episodes in a life cycle of a component was used as an initial value for the number of states "N" in the deterioration model for the HMM.

5.3 The Hidden Markov Model

Hidden Markov Models (HMMs) are widely used in bioinformatics, speech recognition, process monitoring and many other areas. HMM can be defined as a model of two concurrent stochastic processes: the first is the sequence of HMM state modeling for the sequential structure of the data and the second is a set of state output processes modeling the stationary aspects of the process. The HMM is called "hidden" because there is an underlying stochastic process (the sequence of states) that is not directly observable, but affects the observed sequence of events.

HMMs are divided into continuous and discrete models according to their probability density description. Discrete HMMs can represent any distribution as no assumptions are made about the underlying distribution of the observed symbols.

In HMM, there is no one-to-one correspondence between the hidden states and the observed measurements. It is not possible to tell what hidden state the model is in when the observation is generated just by looking at the observation. For the special case of a discrete, first order, Markov chain, the probabilistic description is represented by the current and the predecessor state.

Using the notation and terminology of [Rabiner, 1989 [32]] a discrete HMM is usually characterized by the following elements:

- N, the number of hidden distinct states in the model. The system may be described at any time as being in one of the hidden states. In some real-world applications, the state may contain certain physical significance; however, for most practical applications there is often no clear meaning attached to the states or to the sets of states of the model. Individual states are denoted as $S = \{s_1, ..., s_N\}$ and the time instants associated with state changes are denoted as t = 1, 2.....T. The state at time t is q_t and the state sequence $Q = \{q_1, q_2, ...q_T\}$
- M, the number of distinct observation symbols per hidden state. The observation symbols correspond to the physical output of the system being modeled. Individual symbols are denoted $V = \{v_1, ..., v_M\}$ .The observation sequence O is $o_1, o_2, ..., o_T$ where ($o_i \in V = \{v_1, v_2, ..., v_M\}$)
- The state transition probability distribution matrix $[A]_{ij} = \{a_{ij}\}$ is N × N

  where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$ and where $1 \le i, j \le N$ (5-4)

  In other words it is the probability of [entering state S_j at time t+ 1 giving that it was in state S_i at time t]. This is expressed as

  $$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i \qquad (5\text{-}5)$$

- The observation symbol probability distribution in hidden state *j*, $[B]_{jk} = \{b_j(v_k)\}$, is N x M probability distribution at time *t* of each state where:

  $$b_j(k) = P[o_t = v_k | q_t = S_j] \ , \quad \text{and where} \, 1 \le j \le N \,, 1 \le k \le M \qquad (5\text{-}6)$$

  In other words it is the probability of [producing v_k at time *t* giving that it was in state S_j at time *t*] expressed as

  $$\sum_{k=1}^{M} b_j(k) = 1 \quad \forall j \qquad (5\text{-}7)$$

- The elements in initial state distribution $\Pi = \{\pi_i\}$ where

  $$\pi_i = P(q_1 = S_i) \qquad (5\text{-}8)$$

  Where $1 \le i \le N$ and Π is N x1 initial distribution matrix

  $$\sum_{i=1}^{N} \pi_i = 1 \qquad \forall i \qquad (5\text{-}9)$$

Given appropriate values of *N*, *M*, *A*, *B* and Π, the HMM can be used as a generator to give an observation sequence

$$O = \{o_1 o_2 o_3 \cdots o_T\}$$

Where *T* is the number of observations in the sequence and each observation o_t is one of the symbols from *V*. A complete specification of an HMM requires two model parameters (N and M), observation symbols, and three probability measures A, B, and Π. For convenience, the compact notation is used.

$$\lambda = (A, B, \pi) \qquad (5\text{-}10)$$

The probability that observation sequence $O = o_1 o_2 \cdots o_T$ is generated by the model λ is denoted as $P(O|\lambda)$, assuming independence of observations and Markov transitions.

$$P(O|\lambda)= \sum_{Q_k \in S_N} \left(\prod_{i=1}^{N} \underbrace{P(o_i|q_1)}_{\substack{\textit{Emmission.} \\ \textit{distributions}}} \right) \left(\prod_{i=2}^{N} \underbrace{P({}^k q_n | {}^k q_{n-1})}_{\substack{\textit{Transition} \\ \textit{distributions}}} \right) P \underbrace{({}^k q_1)}_{\substack{\textit{Initial} \\ \textit{state} \\ \textit{probability}}} \tag{5-11}$$

HMM has many features including nonlinear time warping, probabilistic interpretation of recognition results, simultaneous recognition and possible segmentation. The widespread popularity of the HMM framework can mainly be attributed to the existence of the efficient training procedures for HMM.

Among these algorithms, the Baum-Welch and segmental k-means are two of the most commonly used procedures for the estimation of HMM parameters. By assuming the HMM parameters to be fixed but unknown, these parameter estimators have been derived purely from the training observation sequences, sample information, plus some constraints that these parameters must obey without any prior information included.

There may be many cases in which the prior information about the HMM parameters is available. Such information may come from subject matter considerations and/or previous experience.

As an example of HMM, Figure 5.5 shows HMM with 4 states and a set of transition and symbol distributions from each state. This comes together to produce discrete observations with a state sequence $Q = \{1, 1, 2, 3, 3, 4\}$.

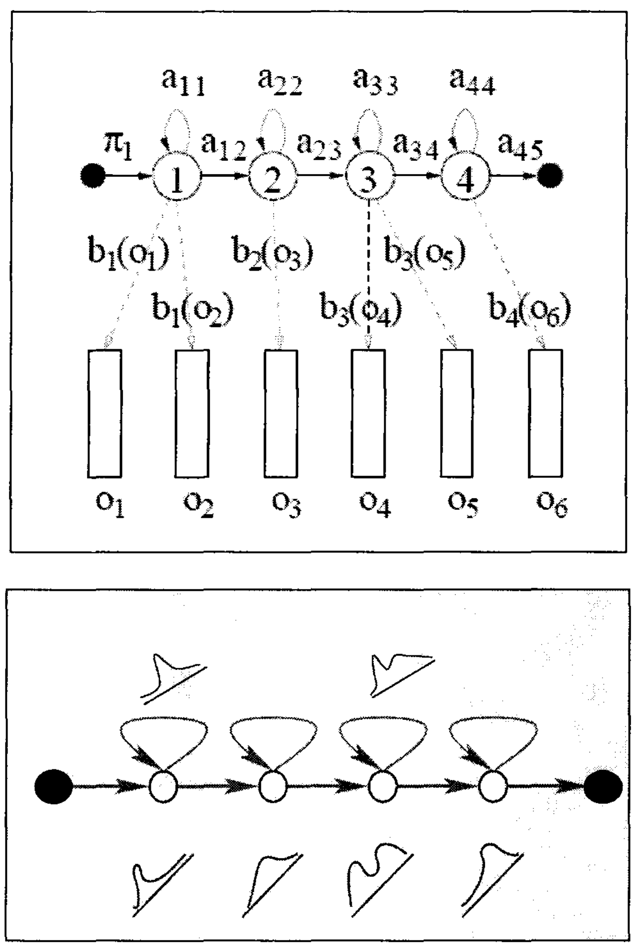


Figure 5.5 - Example of 4-state HMM

5.3.1 Stochastic Finite State for Monitoring Performance Deterioration in Manufacturing Systems Components

A change in the components of a manufacturing system was considered in this research as a Markov process. A Markov process, sometimes called an observed Markov model, is defined as a discrete-time stochastic process where conditional probability distribution of future states of the process, given the present state and all past states, depends only on the present state and not on any past states. The Markov process is represented by a set of states; some states emit symbols and some are silent and some include a set of transitions with associated probabilities.

The Markov assumption on the probability of a random variable at a given time depends only on the value at the preceding time.

$$
\begin{aligned}
&P(q_i \mid q_{i-1}) \\
&= P(q_1)P(q_2 \mid q_1)P(q_3 \mid q_2, q_1)\ldots P(q_N \mid q_1, \ldots, q_{N-1}) \\
&= P(q_1)P(q_2 \mid q_1)P(q_3 \mid q_2)\ldots P(q_N \mid q_{N-1}) \\
&= P(q_1)\prod_{i=2}^{N} P(q_i \mid q_{i-1})
\end{aligned}
$$

The application of HMMs for monitoring the performance deterioration in a component is of great interest because of their strong capability in representing failures and maintenance data, which are non-stationary, by statistical parameters. Furthermore, in pattern recognition, there always exists uncertainty, randomness and incompleteness from various sources. Stochastic models are known to deal with these problems efficiently by using probabilistic models. Among various stochastic approaches, HMMs have proven very effective in modeling both dynamic and static signals. HMMs have proven to be tremendously useful in speech recognition where signals are inherently time varying; same case in this research for the deterioration process in a component of a machine.

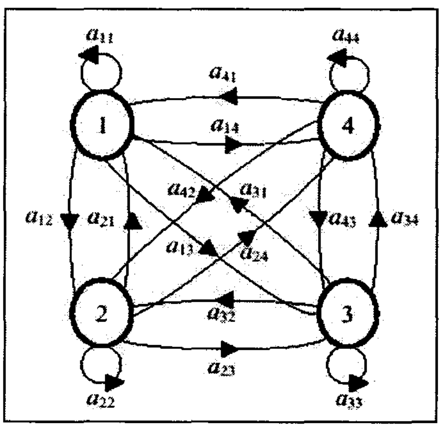
5.3.2 HMM Topology

The HMM represents the overall process behaviour in terms of movement between states and describes the inherent variations of the observations within a state. With discrete HMMs, the continuous multivariate observations can be mapped to a discrete set of observation symbols by vector quantization.

There are different HMM architectures, depending on the limitations imposed on the state transition probability matrix A. Some of the HMM topologies are more suitable to represent the temporal evolution of the process like the Bakis model which fits the performance deterioration process in the components of a machine as in this research. The following section illustrates the difference between two HMM architectures; the full connected Ergodic model and the monotonic Bakis model.

Ergodic Model

This is a topology of HMM where every state can reach any other state of the model in a single step. The state transition matrix (A) is a full matrix. It is also called the fully connected HMM as represented in Figure 5.6.

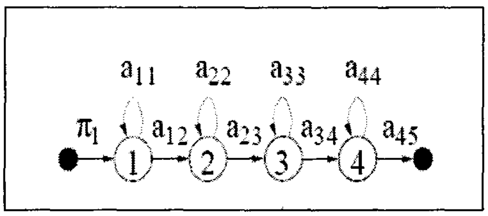


$$a_{ij} = \begin{Bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{Bmatrix}$$

Figure 5.6 - Ergodic HMM; Full State Transition Matrix

Bakis Model:

This HMM is also called a left-to-right model as shown in Figure 5.7. The state transition matrix is not a full matrix with constraints $a_{ij} = 0$ for $j < i$ and $\pi_1 = 1$. Its properties depend on the order of the model. The left-to-right model always starts at the first ("leftmost") state, and transitions are allowed only toward later ("right") states or to the same state. Bakis model is better than the Ergodic model at capturing the dynamic characteristics of data by imposing a temporal order to the model. Bakis models are used in a number of temporal pattern recognition applications such as speech recognition, human gesture recognition, and process monitoring with great success.



$$a_{ij} = \left\{ \begin{matrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{matrix} \right\}$$

Figure 5.7 - Bakis HMM; Restricted State Transition Matrix

There are three main classes of topology estimators: grammatical inference techniques, decomposition of large Ergodic structures, and information-theoretic approaches. The grammatical inference techniques are limited to the estimation of temporal topologies. Topology estimation by decomposition of a large Ergodic HMM is not restricted to a temporal structure, but the topology estimate tends to be complex and usually not an efficient model of the process. The information theoretic approaches are specific to the problem of estimating the order (number of states) for the HMM topology but do not address which transitions should be allowed between the states.

5.3.3 HMM Algorithms

An influential tutorial by Rabiner, introduced the idea that Hidden Markov Models should be characterized by three fundamental problems [Rabiner, 1989 [32]]

Computing Likelihood

Given an HMM $\lambda = (A, B, \pi)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

For a particular hidden state sequence $Q = q_0, q_1, q_2, \ldots, q_T$ and an observation sequence

$O = o_1, o_2, \ldots, o_T$, the likelihood of the observation sequence is:

$$P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i) \tag{5-12}$$

Forward Algorithm is an efficient ($O(N^2T)$) algorithm. It is in the form of a dynamic programming algorithm using a lattice structure based computation, which uses a table to store intermediate values as it builds up the probability of the observation sequence. The forward algorithm computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence. It does this efficiently by implicitly folding each of these paths into a single forward trellis.

Most Probable Path Decoding

Given an observation sequence O and an HMM $\lambda = (A, B, \pi)$, discover the hidden state sequence Q that best explain the observations.

For any model, such as an HMM, that contains hidden variables, the task of determining which sequence of variables is the underlying source of some sequence of observations is called the decoding task. The most common decoding algorithm for HMMs is the Viterbi algorithm. Like the forward algorithm, Viterbi involves dynamic programming, and makes uses of a dynamic programming trellis for the alignment of observation and state transitions.

Parameter Learning

Given an observation sequence O and the set of states in the HMM, the algorithm must learn the HMM parameters A and B that maximizes $P(O|\lambda)$. This is regarded as the most difficult problem associated with using HMMs. This is because there is no analytical solution for the model which maximizes the probability of the observation sequence. Given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. However, using an iterative procedure such as the Baum-Welch method to locally maximize $P(O|\lambda)$ for a selected $\lambda = (A, B, \pi)$, a solution can be obtained.

The EM method used in estimating the HMM parameters is discussed with more detail in the following section as it was used in conjunction with frequent episodes to estimate the stages of performance deterioration of the manufacturing system components

5.3.3.1 Expectation-Maximization (EM)

The problem of parameter estimation consists of finding the model that best explains the training data. This is typically done by maximizing a certain optimality criterion such as the likelihood of the training data under the current model. The maximum likelihood (ML) criterion is one of the most common in statistical algorithms to estimate the parameters in probabilistic models. The model depends on unobserved latent variables; in HMM the state sequence is the latent variable.

The standard algorithm for HMM training is the forward-backward or Baum-Welch algorithm (Baum, 1972) which is a special case of the Expectation-Maximization method. This algorithm takes into consideration that the state sequence for the HMM is not observed (latent) in the training phase and maximises an auxiliary function which is based on the expected values of the hidden variable. [Rabiner, 1989 [32]]

The Baum-Welch algorithm is based on two main concepts for HMM parameter estimation. The first idea is to iteratively estimate the transition and observation probabilities, and then iterate to improve the estimates. The second idea is that the estimated probabilities are obtained by computing the forward probability for an observation and then dividing that probability mass among all the different paths that contributed to this forward probability. The steps of EM method are presented in Appendix 3

In summary, the forward-backward algorithm starts with some initial estimate of the HMM parameters $\lambda = (A,B)$. Then iteratively run two steps; the **expectation** step, or **E-step** which calculates the expectation with respect to the latent data given the current estimate of the parameters and the observations. The **maximization** step, or **M-step** which estimates a new set of parameters according to a Maximum Likelihood (ML) criterion.

In the E-step, the expected state occupancy count γ and the expected state transition count ξ are computed from the earlier A and B probabilities. In the M-step, γ and ξ are used to re-compute new A and B probabilities.

Function FORWARD-BACKWARD (*observations* of length *T*, *output symbols V*, *hidden state set Q*) **returns** *HMM=(A,B)*

initialize *A* and *B*

iterate until convergence

E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} \qquad \forall t \text{ and } j \tag{5-13}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_t(j)}{\alpha_T(N)} \qquad \forall t, i \text{ and } j \tag{5-14}$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i.j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N}\xi_t(i,j)} \tag{5-15}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. O_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)} \tag{5-16}$$

return *A*, *B*

Although in principle the forward-backward algorithm can do completely unsupervised learning of the *A* and *B* parameters, in practice the initial conditions are very important. For this reason the algorithm is often given extra information.

5.4 Relation between Temporal Data Mining and HMM

Datasets with temporal dependencies are common in manufacturing systems. The general techniques for temporal data analysis are broadly classified into two approaches: pattern discovery and learning generative models. The central idea in frequent pattern discovery is to seek temporally ordered sequences of attribute values. On the other hand, learning generative models is another important perspective in time series analysis. An additional level of stochasticity may be introduced to better handle the temporal uncertainties by using HMMs.

The techniques for pattern discovery are often more useful for data summarization and rule generation applications. Most such techniques use counting-type arguments and have what may be called a computer science viewpoint. Model-based techniques, on the other hand, use stochastic methods and have a statistical framework. These techniques provide a principle approach to describing/modeling the statistics that govern data generation.

The research in this thesis integrated the two approaches of temporal data analysis. The frequent pattern discovery analysis was used to pre-process the manufacturing data associated with a manufacturing system component and to identify data segments that resembled patterns of performance deterioration. The HMM model was used to describe the temporal evolution of the process in the form of states and transition probabilities. For that reason the Markovian framework was essential for optimizing the decision making process which was considered as sequence of actions in response of changes in the performance of a component.

Chapter 6 introduces the decision making process based on Bayesian decision theory using the states of the deterioration process based on HMM. The plant case study in chapter 7 provides examples of using the developed approach for the component replacement decisions for two components when the operational failures, quality failures and maintenance activities are observed. Bayes rule provides an update for the probabilities of the states of nature by inverting the causal relationship among deterioration states and the observations. Figure 5.8 shows the main steps in the temporal data analysis approach which was used in this research.

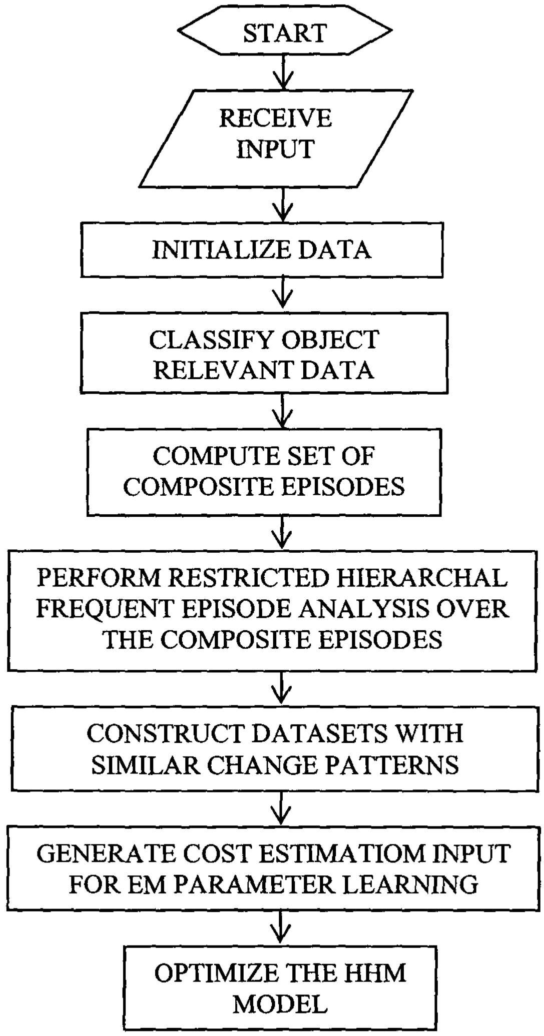


Figure 5.8 - Temporal Data Analysis Approach in Manufacturing System

Chapter 6

Decision Optimization and Control Policies

Chapter 5 provided a description of the process which was used in this research to model the deterioration process in the performance of component as a HMM. The decision options in the deterioration model were investigated to identify the optimal control policy; these decision options were either keeping the component running (in its sequence of performance deterioration states) or replacing the component.

It is clear that taking the machine down for major repair or component replacement is the optimal control policy when operational failures are incurable (i.e. once an operational failure occurs, the machine would not return to running condition until a component replacement) or when quality failures are persistent (i.e. once a quality failure occurs, every subsequent part would be defective until the component is replaced). On the other hand, this is certainly not a good policy when operational failures are recoverable or when the quality failures are not persistent. An example is when a machine is returned to the running condition by resetting the machine faults.

Taking a machine down for extensive servicing entails a cost; likewise, allowing the machine to keep running can be costly if incidents of operational failures increase in occurrence and duration and if defective parts continue to be produced. It is necessary to weigh these respective costs and find a way to make decisions in a rational manner.

As the costs associated with throughput losses, quality failures and maintenance activities are well defined in this research, the appropriate decision making process can be based on a Bayesian decision theoretic approach. Decision making under uncertainty involves choosing among courses of action whose exact consequences are not determined solely by the decision maker's choice.

Decision theory provides a solid theoretical foundation for problems of action and inference under uncertainty. With few exceptions, decision theories are formulated using the analytical framework that consists of a set of component states, Ω; a set of consequences (costs) C; and the set A, of all the mappings from the set of states to the set of consequences. Elements of A, referred to as acts, correspond to conceivable courses of action; a decision maker is characterized by a preference relation on A. States resolve the uncertainty in the sense that once the (unique) true state becomes known, the unique consequence implied by each and every act becomes known [Karni, 2006 [33]]. In the case the states of the component are not known with certainty, the expectation of the consequences would be estimated as in the analysis of this research.

6.1 Bayesian Decision Theory

The framework used in the decision making process in this research is a probabilistic framework based on Bayesian decision theory for choosing the optimal action based on the performance of the components in manufacturing system. Bayesian Decision Theory is a fundamental statistical approach that quantifies the tradeoffs between various decisions using probabilities and costs that accompany such decisions.

For the manufacturing systems under consideration, the control policy consisted of a rule that analytically determines what to do with the observed sequences of operational failures, quality failures, and maintenance activities for system components. The framework is consisted of writing out the risk which is the expected cost, as a function of the control policies, and minimizing it off-line. Figure 6.1 represents a conceptual framework of the research as this section represents minimizing the cost function based on the states probabilities to provide rules for optimal decisions.

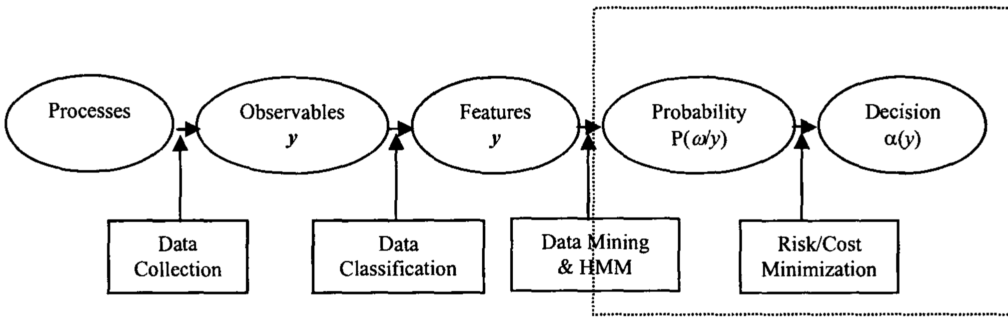


Figure 6.1 - Research Conceptual Framework Showing the Steps for Optimal Decision Making

The answer to the decision problem is a plan of reactions to something that happened to the component of a machine. The component can be in any of a finite set of performance states of nature $\Omega = \{\omega_1, \ldots\ldots \omega_c\}$ and the decision maker can react by choosing an action among a finite set of possible actions A $= \{ \alpha_1, \ldots, \alpha_a\}$ which are $\{\mathcal{T}=$ take component down for maintenance , $\mathcal{L}=$ Leave the component running$\}$. The performance of decision making is evaluated by the loss function λ: A $\times\ \Omega \rightarrow \mathbb{R}$ which establishes the cost $\lambda\ (\alpha_i|\omega_j)$ of choosing action α_i when the component state is ω_j. As mentioned before, in this research the states of the components were not observed directly. They were inferred indirectly by observing events of failures, which were represented by the variable $y \in \mathbb{R}^d$. The function $\alpha : \mathbb{R}^d \rightarrow$A is called decision function.

For a stochastic description of the variables which are involved in the decision making analysis:

- y denotes the d-component vector-valued random variable also called the feature vector (the total cost of losses).
- ω_j denotes a random variable which represents the component state.
- $P(\omega_j)$ denotes the probability that the component was in state ω_j; also called a prior probability and it indicates how likely the component assume a certain state of nature.
- $\{p(y|\omega_j)\}$ denotea the class conditional density function for y representing the probability that the process caused by ω_j would result in effect y and is called the likelihood of ω_j with respect to y.

The prior distribution $P(\omega)$ and the family of conditional distributions $\{p(y|\omega_j) \mid \omega_j \in \Omega\}$ were estimated in HMM and consequently the full joint distribution was constructed as:

$$p(y, \omega_j) = p(y|\omega_j)P(\omega_j) \qquad , \qquad y \in \mathbb{R}^d,\ \omega_j \in \Omega \tag{6-1}$$

The joint distribution is a complete description of the stochastic model. Therefore, the prior and the family of conditional distributions are adequate to specify the model. By decomposing the joint probability in the two factors $P(\omega_j)$ and $\{p(y|\omega_j)\}$, the causal relationship is exploited.

In practice, $P(\omega_j)$ and $p(y|\omega_j)$ are significantly simpler to model than the joint probability directly and it is used to evaluate the average decision performance in stochastic model. When action α_i is selected at the time the measurement is y, the average loss, also known as conditional risk, could be evaluated by:

$$R(\alpha_i \mid y) = E[\lambda(\alpha_i \mid \omega) \mid y] = \sum_{\omega_j \in \Omega} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid y) \tag{6-2}$$

The term $P(\omega_j \mid y)$ accounted for the fact that, once the measurement is observed, the probability of having a certain state of nature changed. The term $P(\omega_j \mid y)$ is computed from the known quantities by using Bayes rule:

$$P(\omega_j \mid y) = \frac{p(y \mid \omega_j) P(\omega_j)}{p(y)} = \frac{p(y \mid \omega_j) P(\omega_j)}{\sum_{\omega_i \in \Omega} p(y \mid \omega_j) P(\omega_j)} \tag{6-3}$$

The terms in the Bayesian rule clearly represent the relationship between posterior knowledge, prior knowledge and their likelihood.

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

$$\text{P(parameters} \mid \text{data)} = \frac{\text{P(data} \mid \text{parameters)} \times \text{P(parameters)}}{\text{P(data)}}$$

$p(y)$ in (6-3) is called the evidence representing how frequently a pattern with feature value y is measured. It also works as a scale factor that guarantee that the posterior probabilities sum to 1. Given the measurement y, Bayes rule provides an update for the probabilities of the states by inverting the causal relationship among states and measurements.

For decision rule α , α (*y*) is substituted for α_i in (6-2) to obtain the average loss $R(\alpha\ (y)|y)$ given that the measurement is *y*. By averaging over all possible measurements, the overall average loss, "Total Risk" is obtained as:

$$R_{\alpha} = E[R(\alpha(y) \mid y] = \int_{\Re^d} R(\alpha(y) \mid y) p(y) dy \qquad (6\text{-}4)$$

The expected risk is a single scalar that measures the overall performance of the decision. Then, the best decision rule that could be picked is the one that minimized the risk (Bayes risk), i.e.

$$\hat{\alpha} = \arg\min_{\alpha} R_{\alpha} = \arg\min_{\alpha} \int_{\Re^d} R(\alpha(y) \mid y) p(y) dy \qquad , \forall y \in \mathbb{R}^n \qquad (6\text{-}5)$$

By minimizing point-wise the function under the integral sign, the optimal function is:

$$\hat{\alpha}(y) = \arg\min_{\alpha_i \in A} R(\alpha_i \mid y) \quad , \qquad \forall y \in \mathbb{R}^n \qquad (6\text{-}6)$$

The best choice when given a measurement *y* is to choose the action that minimizes the conditional risk $R(\alpha_i|y)$. Accordingly the conditional risk $R(\alpha_i|y)$ is estimated at each decision point and the replacement decision is made at the point when the replacement condition risk is less than running conditional risk.

The model which is used in this research to represent the performance deterioration process in manufacturing system component was based on the assumption that a deterioration process could be modeled by Bakis Markov topology which was described in section 5.2.2. For the objective of this research, state N and state M were added to the sequence.

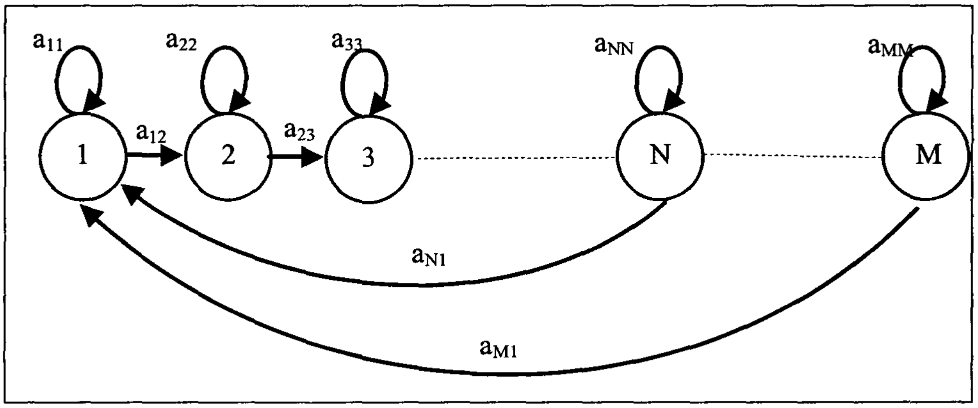


Figure 6.2 – Performance Deteriorating Model for Manufacturing System Component with N State for Optimal Maintenance State and M State as Terminal State with no Preventive Maintenance

The operational states 1, 2,…,N,…M represent a discretization of the deterioration process. State 1 represents the component immediately following replacement or major repairs. The component state is known with certainty only immediately after a replacement in state 1. The change process is assumed to be monotonic. State N is assumed to be the optimal state of the minimum conditional risk for the action of replacing the component with deteriorating performance. State M is assumed to be terminal state at which the component could not deteriorate for further states, such as in run-to-fail control policy.

As the decision makers in this research did not know with certainty the current component state, the posterior probability distribution of the performance state was used to determine the conditional risk associated with each possible decision.

6.2 Cost Function Based on Total Cost

The loss function stated exactly how costly each action was. It was used to convert a probability determination into a decision. Loss function treated situations in which some kinds of decision mistakes were more costly than others. The work in this research was focused on the case in which the decision maker had to decide whether to take the machine down for servicing, or to keep it running after experiencing a reoccurrence of operational failures, quality failures or high maintenance activities.

The following notation was used to define the total cost:

ω_j Component state of nature at step j

$\mathcal{R}$ Scheduled running time (manufacturing system in production time)

$\mathcal{D}$ Scheduled downtime (manufacturing system in scheduled downtime)

O Operational failure occurred

$\mathcal{Q}$ Quality failure occurred

$\mathcal{A}$ Maintenance activities for the component

$\mathcal{T}$ Action to take the component down for maintenance

$\mathcal{L}$ Action to leave the component (machine) running

In addition to the terms and events which were defined above, the following conditional costs were known in the case study:

$C_{O}^{\mathcal{T}}$ Throughput losses (cost) of taking the machine down at state ω_j

$C_{\mathcal{Q}}^{\mathcal{T}}$ Quality losses (cost) of taking the machine down at state ω_j

$C_{\mathcal{A}}^{\mathcal{T}}$ Maintenance losses (cost) of taking the machine down at state ω_j

$C_{O}^{\mathcal{L}}$ Throughput losses (cost) of keeping the machine running at state ω_j

$C_{\mathcal{Q}}^{\mathcal{L}}$ Quality losses (cost) of keeping the machine running at state ω_j

$C_{\mathcal{M}}^{\mathcal{L}}$ Maintenance losses (cost) of keeping the machine running at state ω_j

Φ Estimated cost for stopping the production time per time unit

The expected cost (the conditional risk) of taking the line down for replacing a component or performing a major repair of the machine at any deterioration state j is separated into two regions; scheduled running time and scheduled downtime.

From equation in (6-2)

$$C_O^{T}P(\omega_j \mid O)+C_Q^{T}P(\omega_j \mid Q)+C_{\mathcal{A}}^{T}P(\omega_j \mid \mathcal{A}) \tag{6-7}$$

On the other hand the expected cost (the conditional risk) of keeping the machine running at any performance deterioration state j is given by:

$$C_O^{L}P(\omega_j \mid O)+C_Q^{L}P(\omega_j \mid Q)+C_{\mathcal{A}}^{L}P(\omega_j \mid \mathcal{A}) \tag{6-8}$$

Using the Baysian rule, the cost formulas in (6-7) and (6-8) can be estimated as

$$C_O^{T}\left[\frac{P(O \mid \omega_j)P(\omega_j)}{P(O)}\right]+C_Q^{T}\left[\frac{P(Q \mid \omega_j)P(\omega_j)}{P(Q)}\right]+C_{\mathcal{A}}^{T}\left[\frac{P(\mathcal{A} \mid \omega_j)P(\omega_j)}{P(\mathcal{A})}\right] \tag{6-9}$$

$$C_O^{L}\left[\frac{P(O \mid \omega_j)P(\omega_j)}{P(O)}\right]+C_Q^{L}\left[\frac{P(Q \mid \omega_j)P(\omega_j)}{P(Q)}\right]+C_{\mathcal{A}}^{L}\left[\frac{P(\mathcal{A} \mid \omega_j)P(\omega_j)}{P(\mathcal{A})}\right] \tag{6-10}$$

From equation (6-6) the best optimal state to replace the component or to perform major repair is the state when

$$C_O^{T}\left[\frac{P(O \mid \omega_j)P(\omega_j)}{P(O)}\right]+C_Q^{T}\left[\frac{P(Q \mid \omega_j)P(\omega_j)}{P(Q)}\right]+C_{\mathcal{A}}^{T}\left[\frac{P(\mathcal{A} \mid \omega_j)P(\omega_j)}{P(\mathcal{A})}\right]$$

$$<$$

$$C_O^{L}\left[\frac{P(O \mid \omega_j)P(\omega_j)}{P(O)}\right]+C_Q^{L}\left[\frac{P(Q \mid \omega_j)P(\omega_j)}{P(Q)}\right]+C_{\mathcal{A}}^{L}\left[\frac{P(\mathcal{A} \mid \omega_j)P(\omega_j)}{P(\mathcal{A})}\right]$$

6.2.1 Assessing the Costs

The costs associated with the two decision options, take the machine down for maintenance or keep the machine running, were investigated in two time frames; scheduled running time and scheduled downtime. Most decision makers in production lines have common understanding to avoid taking the machine down and interrupt production. It is preferred to delay maintenance activities until the next scheduled downtime such as shift breaks or weekends. Obviously this is a justified decision unless the cost associated with keeping the machine running is greater than the cost associated with taking the machine down for maintenance. In the case of a preventive maintenance decision for component replacement or major repair, the second decision which has to be made is which scheduled downtime should be selected to optimize the life cycle of the component and avoid premature preventive maintenance.

The cost function in this research depends on the reliability and maintainability characteristics of the components in this study. The Parameters which were considered were the cost per time unit of operational failure, cost per time unit of quality failure, the cost per time unit of corrective maintenance, the cost per time unit of preventive maintenance and the cost of replacements or major repairs.

6.2.1.1 The Elements of the Costs

Each control policy can incur two kinds of costs:

- Fixed costs
- Recurring costs

Combining these two quantities into a single expression was not straight forward. In the case where the decision was selected to take a machine down for maintenance, the fixed cost was per outage, the recurring cost per cycle. To combine the two, it was necessary to multiply the recurring cost by the number of cycles in an outage; this meant the MTTR had to be used to estimate the number of lost cycles in an outage that in the case that the replacement decision was made not in a scheduled downtime.

It was clear that the temporal dimension of the costs associated with various decisions could best be addressed by using dynamic programming. There were no recurring costs when decisions were made at scheduled downtime. The fixed costs were assumed to be the same in the case of scheduled running time or scheduled downtime. The fixed costs and the recurring costs were further decomposed into costs related to throughput losses, costs related to quality problems and costs related to maintenance activities per system component.

6.2.1.2 Cost of Taking a Component (Machine) Down for Maintenance

In case of taking the machine down at scheduled downtime, the fixed costs included the following components:

- Shut-down cost
- Repair/Replacement cost
- Start-up cost

The shutdown and start up costs are independent of the state of the machine. They involve raw or semi-processed material wasted while the machine is going down and coming up, this is considered quality related cost as the parts have to either be repaired or scrapped for quality failures. On the other hand, repair/replacement cost is a function of the machine state. The costs include charges for repair, parts, and installation. In either case, fixed costs are relatively straight-forward, as they do not change depending on whether the machine is in isolation or embedded in a production system.

In light of the above discussion, the cost parameters were calculated as follows.

<u>In case of taking the machine down at scheduled downtime</u>

$C_O^{\tau} = 0$

There were no throughput losses as there was no throughput expected.

C_Q^{τ} = Shutdown cost + Startup cost

Re-work or scrapped parts were considered quality failures

$C_{\mathcal{A}}^{T}$ = Maintenance labor and parts

Labor calculated at straight time and overtime.

<u>In case of taking the machine down at scheduled running time</u>

C_{O}^{T} =($MTTR_{replacement}$) Φ

Which included throughput looses during repair

C_{Q}^{T} = Shutdown cost + Startup cost

Re-work or scrapped parts

$C_{\mathcal{A}}^{T}$ = Maintenance labor and parts

Labor calculated at straight time and overtime.

6.2.1.3 Cost of Allowing a Component (machine) to Remain Operational

It is reasonable to assume that allowing a machine to keep running do not entail any fixed costs. However, since the machine remained operational and continued to produce parts, there are recurring losses. The expected recurring cost of allowing the machine to continue producing parts has several elements:

Expected benefits of good productivity, i.e. good parts known to be good are considered as throughput losses. Expected cost of bad productivity, i.e. bad parts known to be bad, which must be scrapped or re-worked are considered as quality losses. Expected cost maintenance activities that are needed to keep the machine running without replacing the component. The costs are summarized as

$C_O^{\mathcal{L}}$ = – Benefit of good production = ($\mathrm{MTTR}_{reset\ and\ recovery}$) Φ

This includes the recurring cost of losing throughput due to recoverable operational failures.

$C_Q^{\mathcal{L}}$ = Cost of bad production + scrapped parts = Φ (Re-work constraint factor) + scrapped parts cost

$C_{\mathcal{A}}^{\mathcal{L}}$ = Maintenance labor

Maintenance labor is calculated at straight time. This includes the cost of maintenance labor to recover the machine from quality or operational failures without major repair.

6.2.2 The Control-Point Policy in Maintenance

Optimal control point in maintenance management is concerned with decisions to maintain the performance of the system components at a level that would maximizes profit or minimizes cost over the life cycle of the components. Optimal Maintenance strategies are often constructed using stochastic models and focus on finding an optimal inspection time or the optimal acceptable degree of system deterioration before maintenance and/or replacement.

In this section, the maintenance policies for multi-state deteriorating components were presented for the case of imperfect observations such as the case in the plant case study. The approach of control-point policy in this research addressed the uncertainty about the state of the component before and after the replacement. In general, maintenance control policy is considered a set of rules, or a procedure, that prescribes the action to take after observing new information. There are three main methods to decide on the maintenance control policy.

Run-to-fail approach

Run-to-fail is the most common maintenance strategy in manufacturing and can only be improved by efficient reactive maintenance. In this approach, failures are usually persistent either operational or quality. Constant servicing from the maintenance group can also be considered failure. In general, failure condition is decided by the decision maker and it varies based on the decision maker preferences.

Age-based approach

Age-based maintenance models are the most classical strategies. The basic idea is to describe the system deterioration by a single index, the age. This quantity has some desirable analytical properties; deterministic, one-dimensional, and monotone. All this make the analysis of this kind of models elementary. This approach is very intuitive. Many fundamental concepts, such as failure rate, minimal repair and replacement, etc, are defined in this framework. In the plant case study, the control policy of some components is based on age-based approach. The approach is usually very conservative in the age of the component and based on worst case incident in the maintenance records and depends on the cost of the replaced component. CBM or run-to fail approaches are usually used for costly components or components with high risk repairs.

CBM approach

CBM approach is based on observing and monitoring the state of the component and updates some analysis tools with real time data to provide best course of actions. CBM includes many prediction tools that estimate the expected performance of the component as the case in this research.

6.3 Decision Points

In this section the concept of the decision points is introduced to fit within the business model. Decision points can be defined as points of time during the component life cycle where the decision maker should take an action or make an opinion that can modify or keep the performance deterioration path. Theoretically, the decision point occurs at any time as the decision maker can take actions at any time, but practically the decision points occur after observing changes in the component performance. Also, the decision points occur at every scheduled downtime for either regular shift breaks or weekend. The production schedule in most manufacturing systems consists of fixed production periods and fixed break periods such as shift breaks and weekend. It is very common in production lines to evaluate the maintenance activities at every scheduled downtime (break) to prioritize tasks in order to optimize the system performance. The plant case study identified the types of decisions associated with the components in the study and their specific decision points according to the production schedule.

The decision points concept is very important in the training of HMM. The practical selection of the decision points is decisive for the sequence of observation time step selection. The fact that decisions must be made at the decision points make it necessary for the observations to describe the state of the system covering the periods between decision points. The decision points concept along with the total cost of the losses reduce the conditional risk comparison to

$$C_{total}^{T}\left[\frac{P(C_{total}\mid\omega_j)P(\omega_j)}{P(C_{total})}\right] < C_{total}^{\ell}\left[\frac{P(C_{total}\mid\omega_j)P(\omega_j)}{P(C_{total})}\right] \tag{6-11}$$

Where,

C_{total}^{T} = The total cost of taking the machine down over throughout, quality and maintenance at the decision points

C^{L}_{total} = Total cost of throughput, quality and maintenance losses for leaving the machine running between decision points.

6.4 Optimal Decision State

The main objective of this research is the development of methodology that would be transformed into a software agent to support the maintenance decision maker when replacing components of a machine in CBM. As it was discussed in section 6.3, the point of time for replacing a deteriorating component is at the decision point when the conditional risk of keeping the machine running is larger than the conditional risk of replacing the component as in the relation (6-9).

In practical implementation of the method the terms in relation (6-9) were calculated to perform the comparison of the conditional risks. As it was discussed in section 5.3, the states of component performance deterioration could not be observed directly, but must be inferred from imperfect observations.

In this research, the assumption was made that Viterbi algorithm for most probable path provided a reasonable estimation of the model state given the sequence of observations. This assumption was used in the calculations to identify the optimal replacement state and kept the decision making process within the Baysian decision theory framework rather than POMDP.

The calculation of the conditional risk was performed as described below:

- The depreciated value of the component was calculated at each decision point using the prior distribution from the HMM. Using the prior distribution linked the depreciated value to the component performance state.
- The conditional risk of replacing the component at a scheduled downtime consisted of the following costs:
    1. The depreciated value of the component.
    2. The quality failures from machine shutdown and start up.
    3. The cost of maintenance activities to schedule the replacement as preventive maintenance work.

It was very clear that the more the component value was depreciated, the lower the conditional risk would be.

- The calculations of conditional risk for keeping the component running consisted of three steps:
    1. The state of performance deterioration at the decision points was estimated by Viterbi algorithm as the last state of the most probable path given the sequence of observations.

2. The losses associated of keeping the component running at the state of the decision points and all consequent states were calculated from the HMM observations matrix. The last state of the model had an extra cost of downtime associated with the component replacement time.
3. The posterior at the state of the decision point and each consequent state was calculated from the HMM and the measured sequence of observations.

Chapter 7

Plant Case Study

The plant case study investigated the main hypothesis of the thesis which was discussed in chapter 5. The results from the case study showed that for some components in a manufacturing system, the deterioration of performance contained a sequence of episodes of operational failures, quality failures and maintenance activities. The results also showed that the dataset that contained the deterioration in performance could be modeled as a first order left-to-right Markov model. The research used HMM analysis to estimate the states transition probabilities and the states observation probabilities which were used to improve the maintenance control policies in the plant. Decisions were made to optimize the expected performance as a function of cost. The case study also compared the results obtained by the new approach to the control plans and made the appropriate recommendations for updating the plant control policies.

7.1 Process Description

The segment of the engine assembly line which was determined to be the most appropriate for the case study appears in Figure 7.1A. This segment consists of the following machines:

OP136: Error-proofing: Laser check—piston ID/orientation;

OP140: Torque two connecting rods bolts —Number 1 and 2 pistons;

OP143: Torque two connecting rods bolts —Number 5 and 6 pistons;

OP146: Torque two connecting rods bolts —Number 7 and 8 pistons;

OP149: Torque two connecting rods bolts —Number 3 and 4 pistons;

OP152: Crankshaft with eight pistons: Torque to turn/camshaft torque to turn;

OP170: Re-work Loop – Re-work defective engines from OP136 to OP152.

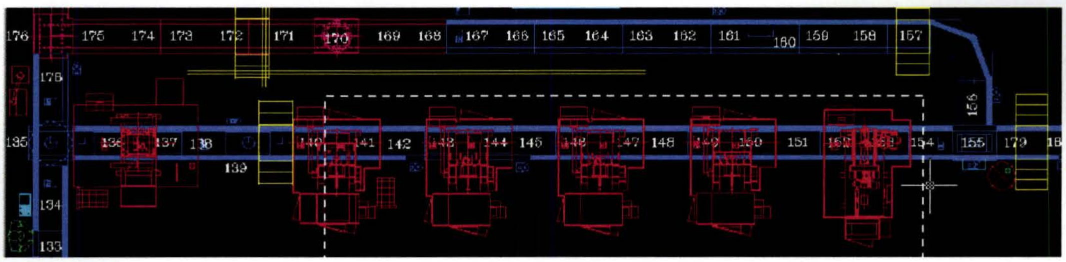


Figure 7.1A - Section of the Engine Assembly Line with the Four Torque Machines for the Connecting Rods. Station 155 Divert Quality Failures to Re-work Station 170

Figure 7.1B – Connecting Rods Torque Operations (OP146)

The data from the torque stations in Figure 7.1B was used in this study. The reason for selecting this section of the line is the size of the sample data. OP140, OP143, OP146 and OP149 are all identical in their function. Each operation has 4 servo torque units to torque four bolts, two for each connecting rod of a piston. Also each operation has a pneumatically controlled turning unit that turns the crankshaft to the appropriate orientation to allow proper alignment between the servo torque unit and the bolts at each piston as in Figure 7.1C and Figure 7.1D.

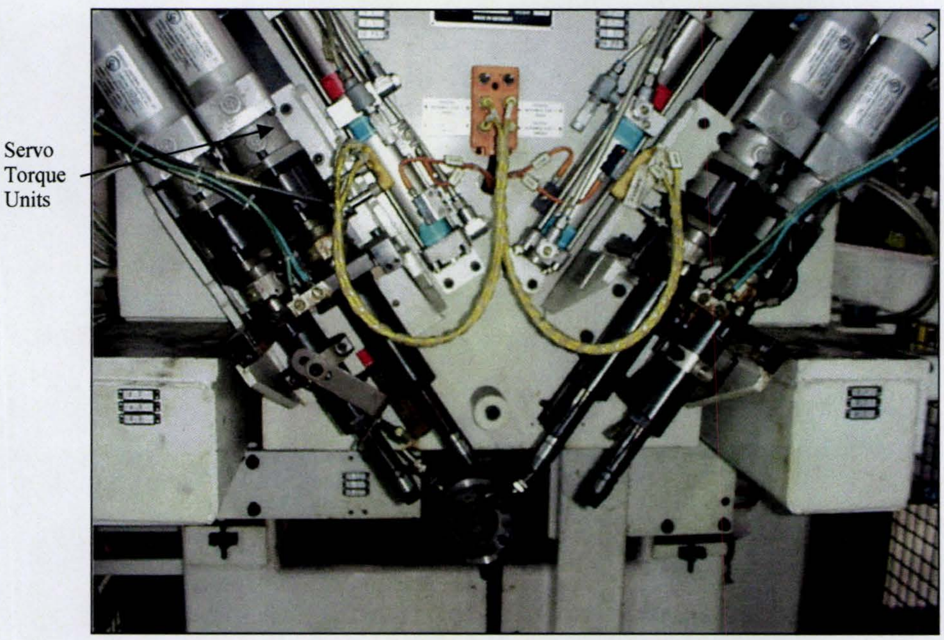


Figure 7.1C –Servo Torque Units

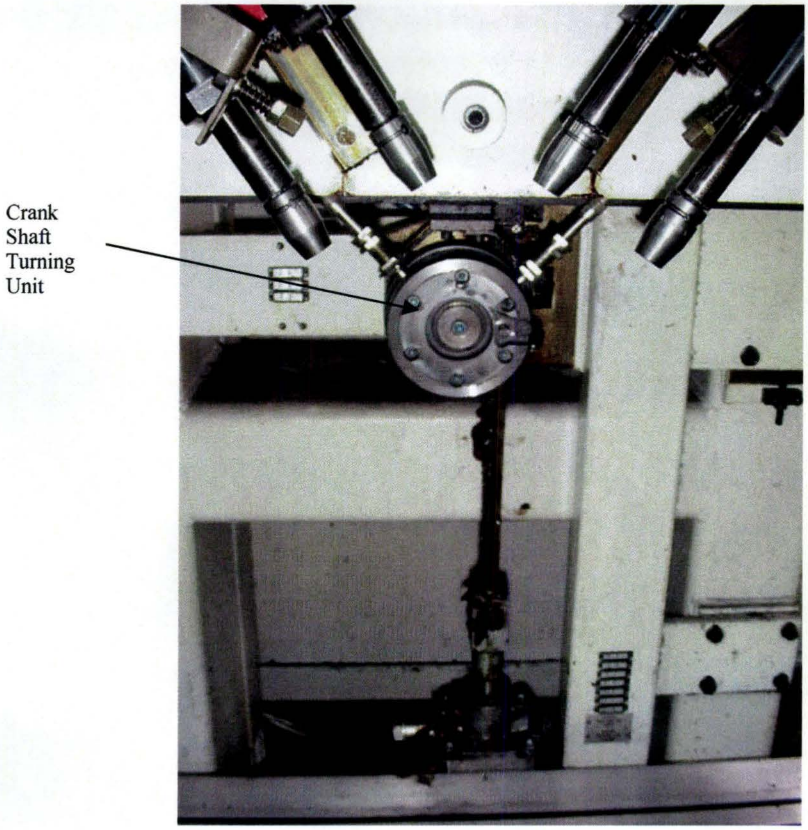


Figure 7.1D –Crank Shaft Turning Unit

In this research it was assumed that all 16 servo torque units experienced the same failure modes. This also applied to the four crankshaft turning units, and that they have the same performance deterioration patterns. This assumption increased the size of the data per component which allowed better patterns identification using the frequent episode analysis and provided a larger training dataset for the HMM.

7.2 Components Used in the Study and its Data Sources

The case study is focused on two components for identifying the performance deterioration patterns; the servo torque units and the crank shaft turning unit. The selected components could not be partially repaired. Repairs of these components could only be done either by replacement or major repair. Simulation data was not used in this research. The variations in the plant data were sufficient enough to cover wide range of conditions.

7.2.1 Components Investigated in the Case Study

Servo Torque Unit

The servo torque unit as a component was described in details in section 3.1.1. It is one of the most common components in the engine assembly line. The maintenance control policies did not have any reliable failure model of the unit. The component life cycle as provided by the manufacturer did not match the failures patterns in the maintenance records because of the different process parameters for each application. The repairs which were performed on the unit were replacement of the main unit, replacement of the internal retention spring and replacement of the bolt socket. While the spring and the socket could be considered as different components, the work in this research considered them part of the parent component and that their failure modes were part of the main component failure modes. This assumption was made to maintain the data integrity of the operational and quality failures.

Crank Shaft Turning Unit

The crank shaft turning unit rotates the crank shaft so that the piston bolt set aligned with the servo torque unit. The deterioration of the turning unit performance would result in a misalignment between the torque units and the bolts. This would do one of two things. If the misalignment was large enough to be detected by the machine position sensors, it would stop the machine in an operational failure mode. If the misalignment was relatively small, it would create quality failures in terms of defective parts. To repair the misaligned unit, major repair would have to be done, which could result in long downtime. These major repairs were usually done on a weekend in a scheduled downtime and involved many skilled trades.

7.2.2 Components Data Sources

As discussed in section 3.3 of this thesis the research platform provided the links between the system components and its data sources. The data considered in this research was the operational failures, quality failures, and all maintenance activities to keep the component functioning without replacement. The data used in the case study covered the time frame from 15 July 2005 to 15

July 2007. The raw data which was collected from the manufacturing database is described below.

• **Operational Failures**. For each machine, the dataset contained the following fields: Machine number, Fault Code 1, Fault Code 2, Fault Description, Start date and time, End date and time, and Total Downtime. Each record corresponded to a particular operational failure. When Fault Codes 1 and 2 are concatenated, the resulting code is uniquely defined for each machine. A sample of the data is shown in Table 7.1

Machine Number	Fault screen	Fault Subscreen	Fault Text	Start Time	End Time	Down Time (s)
STC_G3M3_ASM_140	650	71	Process Fault.Trying To Lower Nutrunner Slide 1. O 103.Pitch Adjustm. 1 Must Be In 83mm Position Check I 124.	5/25/2007 16:07	5/25/2007 16:09	142
STC_G3M3_ASM_140	650	23	Process Fault. Trying To Open Prestop, O 115. Prestop Did Not Move. Check Prestop PRX, Not I 117.	5/23/2007 22:11	5/23/2007 22:13	89
STC_G3M3_ASM_140	645	1	Fault: Too Many Consecutive Rejects. Machine Stops At End Of Cycle. Machine Must Be In Initial Pos.To Reset Fault.	5/17/2007 10:08	5/17/2007 10:08	50
STC_G3M3_ASM_140	650	40	Process Fault. Nutrunner Cycle Not Completed. Check I 208 Or I 209.	5/16/2007 12:45	5/16/2007 12:47	93

Table 7.1 - Sample of the Operational Failures

• **Quality Failures**. For each machine, this dataset contained the following fields: Engine type, Operation Number, Defect Code, Defect Date, and Defect Type. Each record corresponded to a rejected part. Only internal quality failures were considered in this research. Costs because of external quality failures which were detected downstream in the supply chain were not considered. External quality failures had significant effects on the decision making process in the manufacturing system management, however such analysis is outside the scope of this thesis. A sample of the data is shown in Table 7.2

Engine Type	Operation	Defect Code	Defect Date	Defect Type
CBC	140	14008	5/24/2007 19:19	Alignment Issue 5.3L HO
CHA	140	14008	5/24/2007 19:44	Alignment Issue 5.3L HO
CHA	140	14008	5/24/2007 20:00	Alignment Issue 5.3L HO
CHA	140	14008	5/24/2007 20:06	Alignment Issue 5.3L HO
ZLF	140	14011	5/25/2007 9:20	Reversed Cap
CHA	140	14002	5/26/2007 10:27	Bolts Not Torqued
CHA	140	14002	5/26/2007 18:12	Bolts Not Torqued
CHA	140	14002	5/26/2007 18:15	Bolts Not Torqued

Table 7.2 - Sample of the Quality Failures

• **Maintenance Activities**. For each machine in the segment, this file contained the following fields: Work order number, Machine number, Skilled trade type, Material cost, Labor cost, Service cost, Start date, End date, and Downtime. A sample of the data is shown in Table 7.3. Many of maintenance records were partially complete and some were entirely missing. Further investigation was done with the plant maintenance group to get some details especially in the case of component replacement. Many times it was difficult to get an agreement on the particulars of the maintenance events from the maintenance group. The data used in this research represented a reliable dataset giving all the uncertainties of the data collection.

STATUSDATE	WORKTYPE	LEADCRAFT	DESCRIPTION
05/12/2007 12:10:00.000	EMR	TR	ANDON TR for AW1 Engine Assembly 43113 Zone 2 STN140
05/18/2007 08:36:00.000	RC	MW	43113 Krause Guarding Check Loop 1
05/18/2007 08:37:00.000	EMR		Spindle alignment off slightly. Adjusted to stop crank position sooner
05/18/2007 10:28:00.000	RC	EL	43113 E/L running check
05/25/2007 07:47:00.000	EMR		spindle not aligned
05/25/2007 07:49:00.000	RC	MW	43113 M/W running check

Table 7.3 - Sample of the Maintenance Records

7.2.3 Maintenance Decisions Related to the Components of the Case Study

As described in section 3.4 of the thesis, it is the responsibility of the maintenance group to take all necessary actions to maintain the desired performance of the system. The decisions related to the component of a machine in the case study included the component inspection schedule and the component replacements or major repairs. The case study showed the difference in results between the currently adopted maintenance policies for both tactical and operational decisions and the method which was developed in this research. The decision point concept which was introduced in section 6.3 was also utilized for determining the aggregation levels of total cost. The fact that the optimal decision was based on replacement or major repair restricted the decision points to weekends for the 24 hour production day schedule. This limitation allowed the aggregation of the cost estimation on a weekly basis.

7.3 Identifying Episodes of Performance Deterioration

7.3.1 Classifying Components Events

The first step of the analysis is the identification of frequent episodes of operational and quality failures associated with maintenance activities. Classifying the failure records per component is a necessary step to filter out other components records. Operational failures and quality failures were classified using fuzzy logic and decision tree tools. The details of the classification algorithm are provided in Appendix 2. Maintenance records were classified by some filters which were created for the test dataset. Figures 7.2 A & B show the difference in a dataset for the operational failures, quality failures and maintenance activities before and after classification. Only the servo torque unit and the crank shaft turning unit records were presented in Figure 7.2 B.

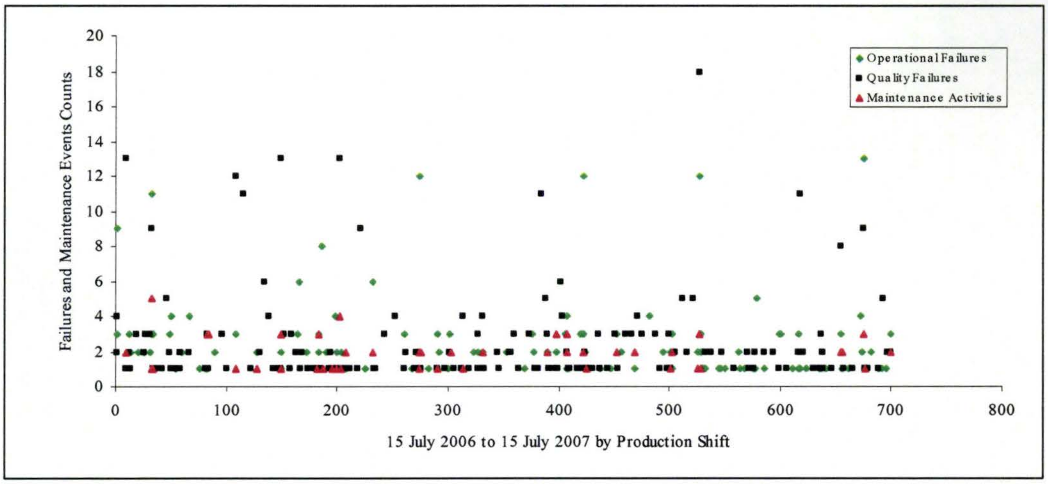


Figure 7.2A - Operation 140 Failures and Maintenance Activities before Component Classification

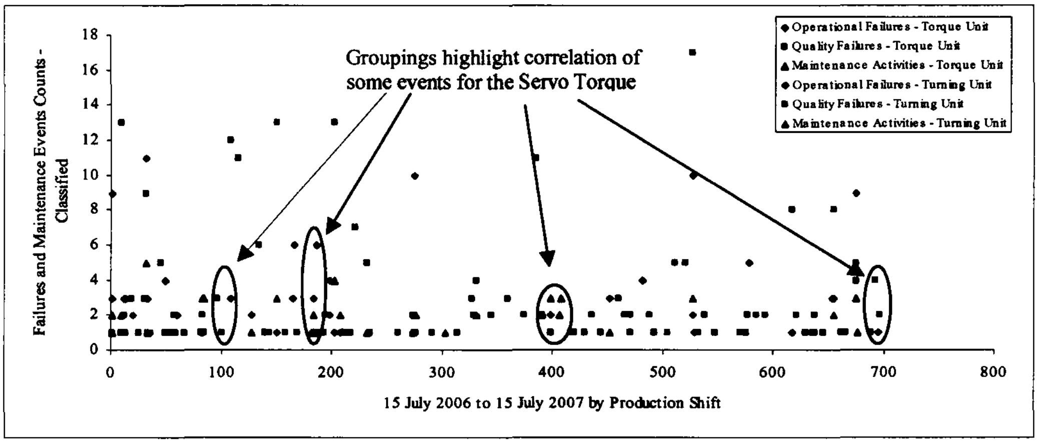


Figure 7.2B - Operation 140 Failures and Maintenance Activities after Servo Torque Unit and Crank Shaft Turning Unit Classification

7.3.2 Frequent Episodes Identification

As described in section 5.2.4.1 the composite episode is the universal case of the frequent episodes. Individual serial and parallel episodes were considered as a special case of a composite episode. The composite episodes were identified by using an algorithm based on parallel and serial episode algorithms which were described in Appendix 1. The Algorithm identified associations between events, then identified the sequence of events starting with parallel sequences and then tested for serial sequences. The frequent episode algorithm was modified to handle special cases in the dataset. These cases were:

- **Time delay in quality failure records**

  There was an inherent time delay in the time stamp of the quality records in the dataset. The delay was the result of the data collection process where the quality failure records were created when the defective engines were rejected to the re-work loops. In the normal line flow, the time delay should not exceed 10 minutes. This was one of the reasons why the size of the search window was increased to 30 minutes. The delay time affected the order of the quality failure events in the composite episode analysis, for this reason the algorithm was changed to accept a change in the order of the quality failure records only.

- **Events of the same type**

  There were cases where the search window had repeated events. These multiple events affected the order of the events within the composite episode. In that case the algorithm was modified to deal with this as one extended event. This case was clear for the example of multiple operational failures when the machine fault was reset several times within a short period of time.

Table 7.4 below describes the output of the frequent episode algorithm. The analysis was performed based on the component records and system shared events. In the case of common operational failures like guard door faults and system master control faults, they were left in the dataset as part of the actions associated with maintenance activities.

Failure Mode	**Number of Composite Episodes Types**	**Number of Episodes in the Dataset**	**Composite Episode Type**	**Operational Failures in the Episode**	**Quality Failures in the Episode**	**Maintenance Records in the Episode**
Servo Torque Unit – Main Unit Failure	4	26	STU-M1	645-1		EMR- EL
		29	STU-M2	645-1, 650-40	140-02	EMR- EL
		17	STU-M3	650-42	140-15	
		12	STU-M4	650-40, 600-1	140-02 140-15	EMR- EL EMR- MR
Servo Torque Unit – Spring Failure	3	19	STU-S1	650-88, 600-16	140-02	EMR- EL EMR- MR
		17	STU-S2		14015, 140-02	EMR- MR
		6	STU-S3	645-1	140-15	EMR- EL
Servo Torque Unit – Socket Failure	2	47	STU-T1	645-1	140-02	
		41	STU-T2	600-16	140-02, 140-15	
Crank Shift Turning Unit-Alignment Failure	3	9	CRT-A1	645-6 , 600-1	140-08	EMR- MR EMR- PF
		13	CRT-A2	650-70, 650-54		EMR- MR
		9	CRT-A3	645-98	140-08, 140-02	MR

Table 7.4 - Composite Episodes Identified in the Dataset

Explanation of Some Composite Frequent Episodes:

- **STU-M2**

  This episode started with quality failure event 140-02 "Bolts Not Torqued" for one or multiple times. Then it was followed by operational failure 645-1 "Fault: Too Many Consecutive Rejects. Machine Stops At End Of Cycle. Machine Must Be In Initial Pos. To Reset Fault." Then the fault was followed by an operational failure 650-40 "Process Fault. Nut runner Cycle Not Completed. Check I 208 Or I 209." This was then followed by a maintenance call "EMR- EL" for an electrician. This episode was similar to multiple occurrences of other episodes like quality failure event 140-02 and operational failure 650-40. This episode was an indication of deterioration of the servo unit and it was repeated multiple times in the Servo Torque Unit – Main Unit Failure.

- **CRT-A1**

  This episode started with quality failure event 140-08 "Alignment Issue 5.3L HO" multiple times followed by an operational 645-6 "Fault: Too Many Consecutive Rejects Crank Orientation. Machine Stops At End Of Cycle. Machine Must Be In Initial Pos. To Reset Fault". This was followed by two maintenance service calls "EMR- MR" and "EMR- PF" for machine repair and pipe fitter skilled trades, which was then followed by 600-1 operational fault "Station System Fault. Master Control Relay Is Not On, I 104. Press Master Start". This episode reflected a change in the crank shaft turning unit. The work was done by the maintenance trades to compensate for the misalignment but the events indicated deterioration.

- **STU-T2**

  This composite episode consisted of multiple occurrences of quality failure event 140-02 "Bolts Not Torqued" and quality failure event 140-15 "Machine Fault" followed by operational failure 600-16 "Station System Fault. Guard Door Is Not Closed. Check 362LS, I 136". This was identified as the last deterioration state of the socket. The "Machine Fault" code was usually selected by the team leader when the servo torque unit failed to complete the task without an operational fault. The 600-16 operational failure was the indication that socket replacement took place.

The search window size had significant effect on the number of the discovered episodes along with the frequency threshold. In this research the maximum size of the search window for the events start time was limited to 30 minutes. As it was discussed in section 5.2.1, events had to occur within a fixed proximity of time for the episode to be considered interesting. The intention of using the frequent episodes analysis was to find correlation between events. In reality those events were the observations of the consequences for a change in the component state. For this reason it was practical to assume that the start point of all these events should fit within the 30 minute search window size. However there was no limit imposed on the duration of each event within a composite frequent episode. Figure 7.3 shows the relation between the size of the search window; number of episodes and the frequency threshold for one of the composite episodes in the servo torque unit; composite episode STU-M2. Cases of search window of more than 40 min. were investigated. While there were more events in STU-M2 composite episodes, these extra events were not related and did not reflect the change in the component as in the case of 30 min. search window.

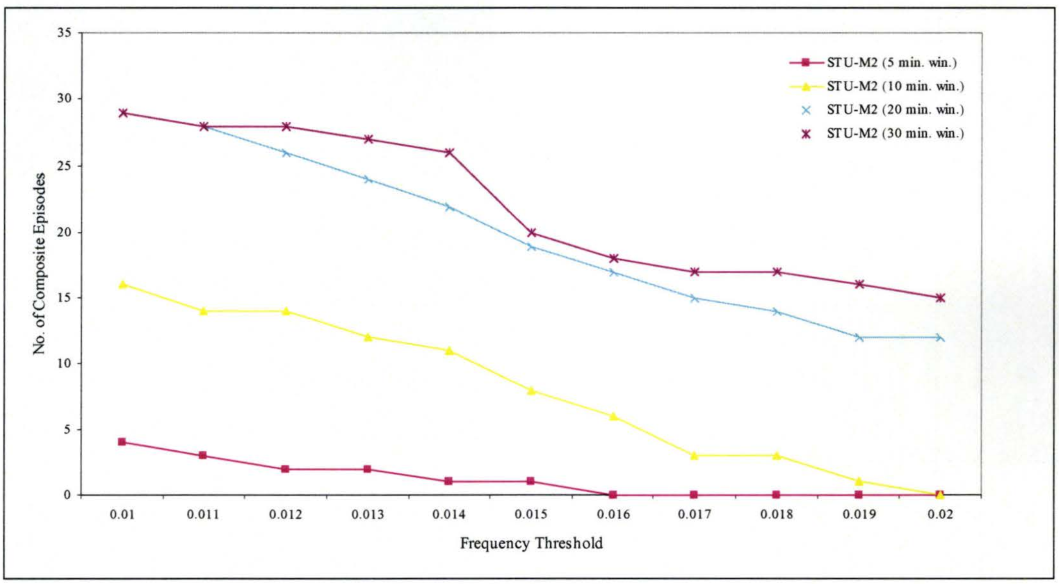


Figure 7.3 - Results of Modifying the Search Window Size on the Number of Composite Episodes by the Threshold Value

The results of the frequent episode analysis were only possible with the help of plant expert. The expert role included the proper classification of some operational failures and quality failures where the automated method failed. The plant expert also verified some of the missing maintenance records especially for the components life cycle. It was very important to filter out some of the noise in the failures records manly because of the limitation in the data collection process. The expert also helped in eliminating some of the discovered associations between events as some composite episodes were not relevant to the performance deterioration process.

7.3.3 Frequent Sequences of Composite Episodes

After identifying all of the composite episodes in the dataset, a new dataset was constructed for each the servo torque unit and the crank shaft turning unit using only the discovered episodes. The dataset also contained the maintenance records that were identified for component replacement or major repair for the components under investigation. The search window was limited in the second level frequent episode analysis that covered the period between the component replacements.

The results of the frequent episode discovery were significant as described in section 5.2.4.2. Many of the findings had to be investigated further in detail with the maintenance group to get reasonable explanations. The results are summarized in Table.7.5.

Component	Failure Mode	Number of Replacements/Repairs from Maintenance Records	Matching Sequences	Replacements With no Match , with Explanation	Replacements With no Match , without Explanation
Servo Torque Unit	Main Unit Fail – Complete Replacement	13	5	2	6
	Retention Spring – Complete Replacement	40	7	24	9
	Socket Fail – Complete Replacement	Unknown	23	Unknown	Unknown
Crank Shaft Turning Unit	Turning Unit – Major Repair	7	3	1	3

Table 7.5 - Results of Frequent Episodes Analysis for the Composite Episodes Dataset

Servo Torque Unit

- In the case of the servo torque unit – main unit failure mode, the control policy in the plant is run-to-fail. The reason for adopting the run-to-fail policy is the price of the unit, the time and risk involved in unnecessary replacements and the lack of any reliable predictive tool. The failure data for the servo torque unit was a sufficient dataset for identifying performance deterioration over the life cycle of the component. The results showed 5 out of 13 units had similar sequence of episodes. Given the variations in the process and the uncertainty in the data, the number of matching sequences was significant. The results also showed 2 cases which were marked in the table as having no matching sequence but with an explanation. After further investigation of these cases, it was found that in these incidents the servo torque units were replaced prematurely due to wrong troubleshooting process. In many cases of emergency breakdown, due to the run-to-fail policy, the wrong process of troubleshooting might lead to the unnecessary replacement of many components. This would entail extra cost that could be avoided if a reliable component life cycle estimation model is available. The table also shows 6 out the 13 component replacements did not have any match in the sequence of the composite episodes. There was no clear explanation for the patterns of deterioration in these six cases. One possible cause could be that some servo torque units were previously used unit when they were installed. Another reason could be the existence of more than one failure mode because each operation had four servo torque units and failure records were not specific for each servo torque unit.

- For the case described as the servo torque unit – spring failure mode, there was an active preventive maintenance program for replacing the springs. The plant control policy in place is an age-based model. The number of cycles estimated by the maintenance group covered 16 weeks of production based on the throughput level of the line at that time. The 33 replacements that were not identified the sequence of composite episodes can be explained by the fact that only the reliable stage of the life cycle of the spring was observed in the dataset. In this reliable stage there were few failures and consequently very few or no composite episodes. On the other hand, the 7 matching sequences were explained by one incident in which the preventive maintenance replacement was not conducted. The delay in the preventive maintenance replacement reflected in more failures to point where the composite episodes were identified.

- For the case of the servo torque unit – socket failure mode, the plant control policy was not clear. The plant had a preventive maintenance program for the socket change using an age-based model, but it was also left to the production team leader's judgment for cases when a high level of torque failures occurred. The results of the search identified some sporadic frequent episodes, which were identified as socket replacements after further investigation.

Crank Shaft Turning Unit

- For the case of the servo torque unit – alignment failure mode, the control policy was run-to-fail. However, failure was not the persistent non functioning state; it was an advanced state of deterioration where high number of quality failures was observed. There were many reasons behind this policy. The repair was major in its nature since a major part of the turning unit had to be disassembled, properly aligned and then reassembled. Thus this repair time was long and involved a large maintenance team. The second reason was the lack of a reliability model that could identify and track the performance deterioration states. The results of applying frequent episodes to identify the sequence of composite episodes were that 3 out of 7 major repairs. Thus there were four repairs that did not match the sequence of composite episodes. Further investigation of the maintenance records showed that there was one incident where the repair was initiated as part of PM for all four machines.

7.4 Identifying HMM for Different Deterioration Patterns

The results of the frequent episode analysis provided a classification for the dataset where the dataset was reduced to similar patterns of correlated operational failures, quality failures and maintenance activities. That process condensed the data into a harmonized dataset with similar patterns of failures per component. The new dataset was used as the training dataset for the HMM.

7.4.1 Converting the Failures and Maintenance Data to Cost Data

The filtered dataset contained the original type of events mainly operational failures, quality failures and maintenance activities. These three events types had to be converted into a single index to represent the deterioration in performance of a component. The deterioration state that initiated the plant floor decision was the level of losses compared to the total running cost. In section 6.3 the cost of deterioration in performance was estimated by projecting the three event types in the cost space using the following assumptions:

Cost of Throughput Losses

The plant control policy for throughput losses was based on the assumption that losses had to be recovered on overtime with a full staff of production and all support groups. In that case, the cost associated with downtime during normal production was estimated mainly with the labour cost to run the production line on an overtime bases, which was denoted in section 6.2 as Φ. This value of Φ was different in the case of constraint sections and non-constraint sections. There were also indirect costs associated with downtime which affected the competitive position of the plant. Indirect costs were not included in the calculations in the case study.

Assumptions

- The throughput losses were based on a single-machine model and not on a production line model. This means the buffer sizes and status as well as the constraint analysis of the section were not considered in the estimation of the throughput losses.
- The durations of the component operational failures were considered as machine stoppage time. The machine was not considered operational during the operational failures.
- Downtime due to operational failures occurred during scheduled downtime (shift breaks) were not considered as throughput losses.

Cost of Quality Failures

The cost associated with quality failures were the results of internal quality cost. The internal quality cost was calculated as a First Time Quality (FTQ) cost. The cost included the delay time associated with rejecting and re-introducing the engine to the production line along with the cost associated with the re-work processes. The time for rejecting and re-introducing engines to the production line was estimated to be on average 16.3 second per reject at that section of the line. The cost associated with this event was similar to periods of downtime. The re-work process associated with the quality failure of the components included replacing the used bolt and the torque of the new bolt.

Assumptions

- The cost of the re-worked bolts was removed from the calculations because of the very small value of the scraped bolts.
- The effect of quality failures to delay production throughput was assumed to be linear. Conditions related to a large number of quality failures filling the re-work area and causing a full stoppage of the line were not considered in this study.

Cost of Maintenance Activities

The cost of maintenance was directly calculated from the maintenance records for the event associated directly with the components of interest.

Assumptions

- Scheduled events of "machine running checks" (PM program), which included inspection for multiple components of the machine, were considered as a cost associated with the component.
- Multiple skilled trades who worked on the same maintenance activity associated with the component were considered to work for the whole duration of the activity.

According to the assumptions above, the weekly costs of these failure events and maintenance activities were calculated for the sequence of observations used in

HMM. The weekly total cost was based on the concept of decision points. The total losses in cost terms for the components in the study were estimated by:

$$C_{Total} = O_{downtime}(\Phi) + Q_{Re\text{-}work}(\Phi) + \mathcal{M}_{time}(C_{Maint}) \tag{7-1}$$

Where:

$O_{downtime}$: Total accumulated downtime for component operational failures (hr)

$Q_{Re\text{-}work}$: Total accumulated delay time because of component quality failures (hr)

$\mathcal{M}_{time}$: Total accumulated maintenance time associated with the component (hr)

Φ : Cost constant representing total line losses created by downtime ($/hr)

$C_{Maint.}$: Cost of maintenance skilled trade ($/hr)

All values were aggregated to the decision point time interval which was a week for the components in this study. Based on the weekly total cost, the observation sequence O was determined.

7.4.2 Modeling the Cost Data using HMM

Based on the total cost, the cost dataset was constructed and the deterioration in performance was estimated from the sequence of observations over the component life cycle. Figure 7.4 shows the losses for the servo torque unit-main unit failure mode for two deterioration incidents; 27 weeks and 36 weeks of life cycle. The graph shows a bias in the component total cost. The bias was due to the fact that each machine consisted of four servo torque units. In the stable phase of the component life cycle there were random quality failures from the four servo torque units. These failures created some losses common to all identified component life cycles.

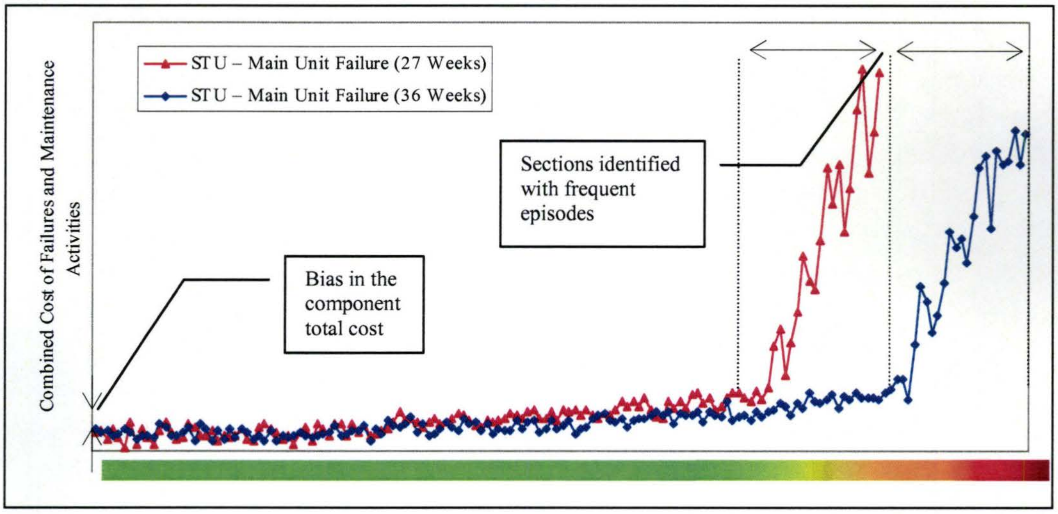


Figure 7.4 – Total Losses for the Servo Torque Unit Performance Deterioration over Two Life Cycles

The training of the HMM was performed using Jahmm, a Java implementation of HMM related algorithms and the HMM Matlab Toolbox. In the training process the initial probability of the initial state distribution $\Pi = \{\pi_i\}$ was set to be 1 for state zero and zero for the rest of the states. This assumption was based on the nature of deterioration process where the sequence of observations started at the first state after component replacement or major repair. $\Pi = \{1,0,0\ldots0|_{S=N},\ldots0|_{S=M}\}$ for the performance deterioration model in section 6.1. The test variable of the learning process was the number of states in the HMM.

The criteria for most appropriate HMM was the model with maximum likelihood for the observations given the HMM, $P(O|\lambda)$ for different HMM state number as in equation (5-11); as well the conformity with the deterioration model which was discussed in section 6.1. An initial start point for the HMM state number was the number of composite episodes identified in the dataset plus the starting state and the terminal state. This assumption was made because the composite episodes consisted of multiple events. These multiple events entailed multiple costs which were reflected in a change in the losses. The assumption in this approach considered these losses as a change in performance.

7.4.2.1 The Servo Torque unit case

Servo Torque Unit – Main Unit Failure Mode

The number of observations was different for the deterioration segments of the dataset because of the length of the component life cycle. Servo torque units in the dataset which had similar deterioration patterns were replaced in periods between 27 and 36 weeks. The reason behind the difference in the component life cycles could be explained by the variations in the process parameters during the study or because of the difference in the internal components of the servo unit itself.

The number of composite episodes for the servo torque unit – main unit was 4 in the identified sequences. The HMM was tested for a 6-state model. This included the beginning stable phase state and the terminating state. The transition probabilities from the model are displayed in the matrix below.

$$a_{ij} = \begin{bmatrix} 0.862 & 0.086 & 0.053 & 0 & 0 & 0 \\ 0.158 & 0.316 & 0.526 & 0 & 0 & 0 \\ 0 & 0.182 & 0.659 & 0.159 & 0 & 0 \\ 0 & 0 & .165 & 0.632 & 0.263 & 0 \\ 0 & 0 & 0 & 0 & 0.286 & 0.741 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The structure of the transition probability matrix was very close to the left-to-right Markov model which reflected the deterioration pattern. The initial state of the model, state 0, had a high stationary probability. This matched the expectation of the deterioration pattern as the component stayed at a high performance state at the beginning stage of its life cycle. The model also showed a relatively high transition probability from state 4 to state 5. This can be explained by the fast rate of performance deterioration experienced at the end of the life cycle. This deterioration of performance was observed as a large number of failures and maintenance activities.

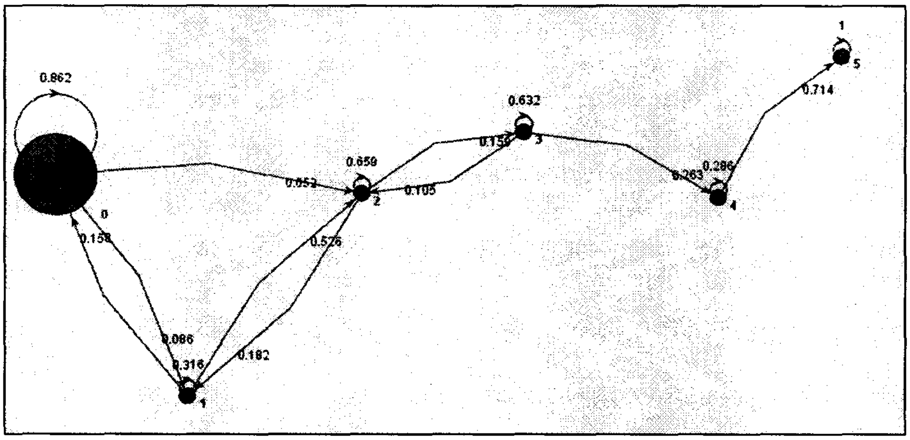


Figure 7.5 - HMM for the Servo Torque Unit – Main Unit Failure Mode (6-State Model)

Based on a 6-state HMM used to capture the deterioration in the servo torque performance the most likely state sequence was created. Using Viterbi analysis as described in section 5.3.3, the sequence of states was the following:

STU-MUS-1:

Q = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 1, 2, 1, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 4, 5, 5]

STU-MUS-2:

Q = [0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 2,1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 4, 4, 5, 5]

STU-MUS-3:

Q = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 5, 5]

STU-MUS-4:

Q = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 2, 1, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 4, 5, 5]

STU-MUS-5:

Q = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3,4, 4, 5, 5]

A test for a different number of states for HMM was made to compare the best sequence of states that described the process. A 5-state model and 7-state model were learned using the same sequence of observations of the cost. In the case of the 5-state model, the transition probability between the last two states was relatively low. Thus the 5-state model missed one of the important ending states.

In the case of the 7-state model, there was no change at the final states of the deterioration from the 6-state model. The new state was thus an intermediate state which can be explained by the stable performance of the component at the middle of its life cycle with only small alterations in performance.

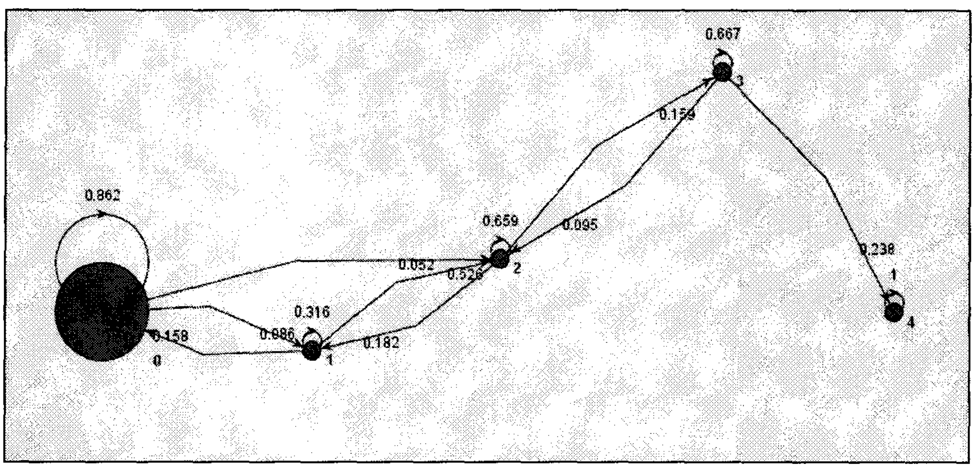


Figure 7.6 - HMM for the Servo Torque Unit – Main Unit Failure Mode (5-State Model)

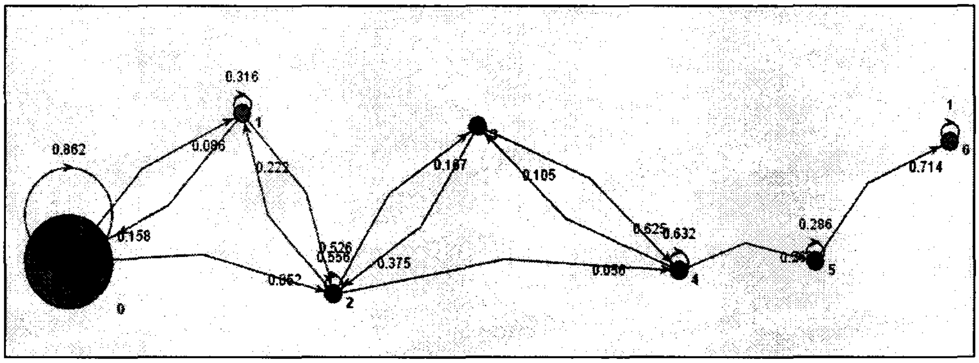


Figure 7.7 - HMM for the Servo Torque Unit – Main Unit Failure Mode (7-State Model)

The HMM training was also performed for the servo torque unit – spring failure mode. The training started with the 5-state model. In this case the results which were produced were a left-to-right model as expected. This is displayed in Figure 7.8. However the transition probability between the last 2 states was equal to 1 which means that the rate at which the component would progress to state 4 was too fast. That condition made the selection of the 5-state model impractical for the decision making process.

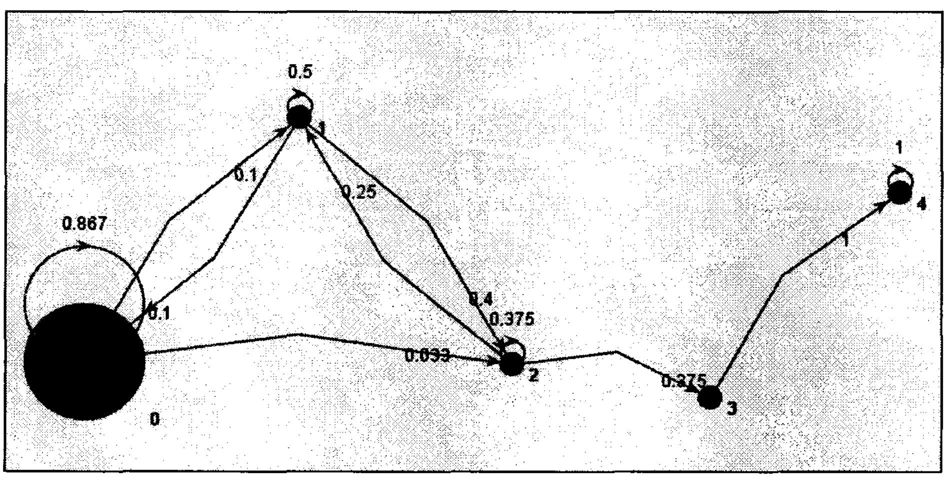


Figure 7.8 - HMM for the Servo Torque Unit – Spring Failure Mode (5-State Model)

When the spring failure mode observations were trained with the 6-state model, state 4 was created as displayed in Figure 7.9. While the transition from state 3 to state 4 remained high, the fact that the model created an intermediate state allowed the option of considering the decision making process at that state.

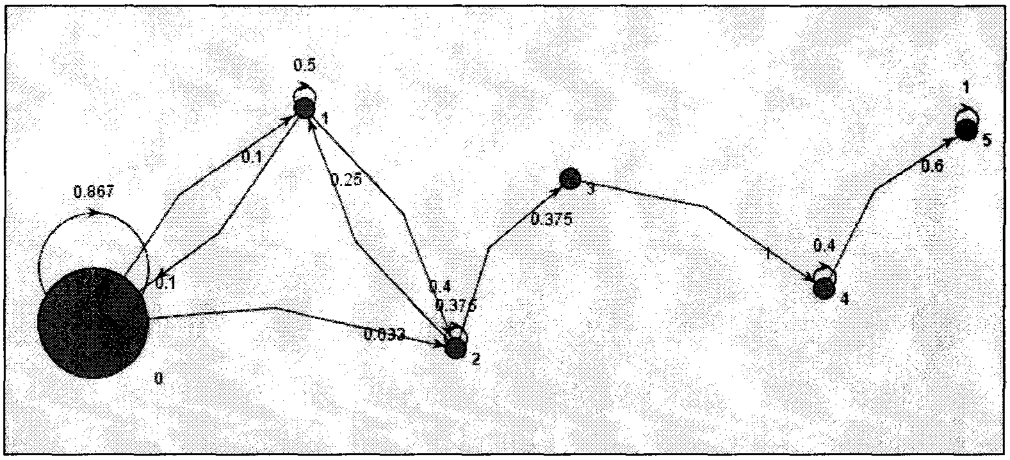


Figure 7.9 - HMM for the Servo Torque Unit – Spring Failure Mode (6-State Model)

When the 7-state model was trained using the observation data, a new state was introduced to the model as showed in Figure 7.10. The new state changed the structure of the HMM and worked as a second terminating state. The fact that the component would return from state 6 to state 5 with a high rate was also not reasonable for the decision making processes. For that reason, the 6-state HMM was used to describe the deterioration of the spring in the servo torque unit.

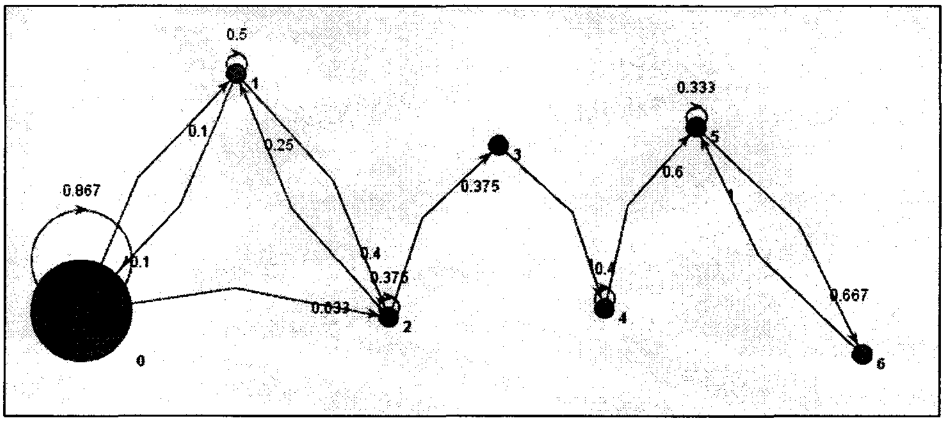


Figure 7.10 - HMM for the Servo Torque Unit – Spring Failure Mode (7-State Model)

7.4.2.2 The Crank Shaft Turning Unit

In the case of the crank shaft turning unit, there were some difficulties in training the HMM. This is explained by the fact that the unit did not reach its terminating state as in the case of the servo torque unit-main unit failure mode. Also from the fact that the unit was serviced many times during the deterioration time, which in some incidents changed the rate of deterioration. The set of observations were trained for 4-state, 5-state and 6-state models. The results of the 4-state model as displayed in Figure 7.11 were the most reasonable for the gradual deterioration pattern, but it had a low number of states. The reason behind that is the unit did not reach the terminating state.

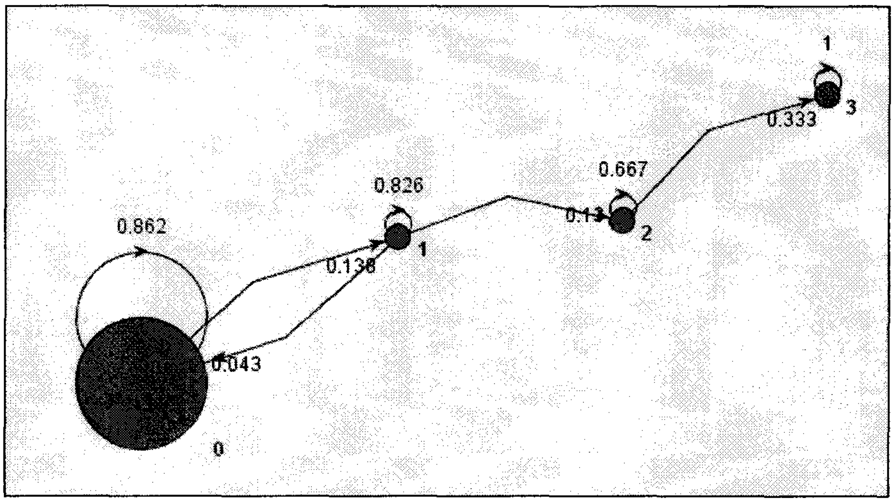


Figure 7.11 - HMM for the Crank Shaft Turning Unit – Alignment Failure Mode (4-State Model)

Based on the 4-state HMM, the most likely state sequences of the observations using the Viterbi analysis were the following:

CST-AS1:

$Q = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 3, 3]$

CST-AS2:

$Q = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 2, 3, 3]$

CST-AS3:

$Q = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3]$

The results of the 5-state model as displayed in Figure 7.12 had some problems with making the transition between state 2 and state 3. The fact that the transition probability was high indicated that the deterioration sequence moved from state 1 to 3 or from 1 to 2 with a very fast transition to state 3. This is similar to the 4-state mode.

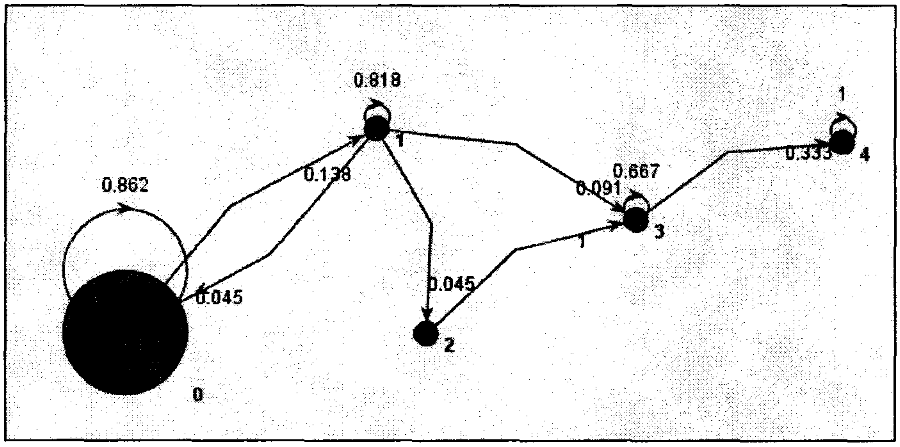


Figure 7.12 - HMM for the Crank Shaft Turning Unit – Alignment Failure Mode (5-State Model)

The results of the 6-state model did not add much to the resolution of the ending states. It added only an extra early state as displayed in Figure 7.13. This did not add much to the decision making process as it is usually done by comparing the conditional risk mainly when the component at the final states. For that reason the 4-state model was selected to model the performance deterioration of the crank shaft turning unit.

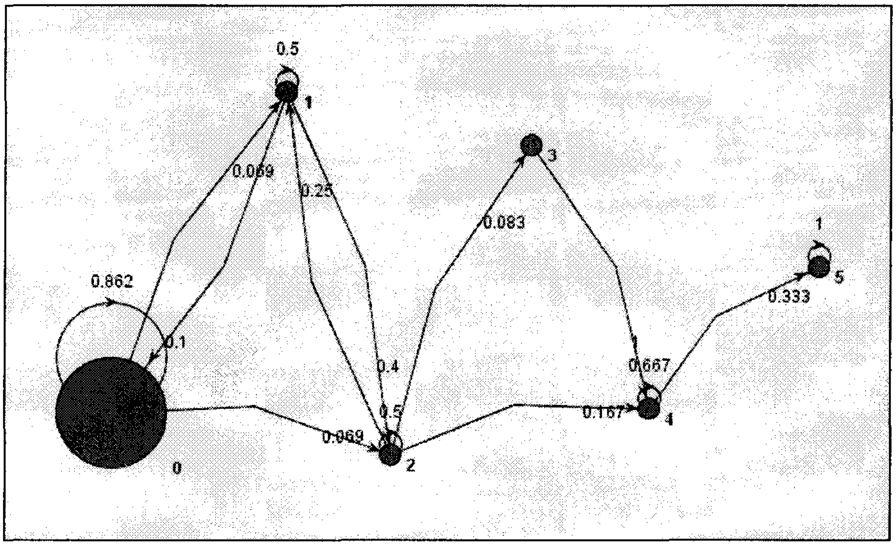


Figure 7.13 - HMM for the Crank Shaft Turning Unit – Alignment Failure Mode (6-State Model)

7.4.3 Optimal Maintenance State for Component Replacement

The optimal maintenance state for the component is the HMM state at which the conditional risk for replacing the component or performing major repair is less than the conditional risk for keeping the component running. This state is state N in the deteriorating Model which was discussed in section 6.1 and section 6.4.

Section 6.2.1 of this thesis illustrated the detail of the cost associated with the two decisions related to machine components in this research. It also showed that the costs were grouped as fixed costs and reoccurring costs as a function of the component state. In this section, the optimal states for component replacement were calculated for the servo torque unit. The depreciation of the component value over its life cycle was included in the calculations based on the prior distribution of the HMM states.

There are two options for decisions as discussed in chapter 6.4; component replacement; or leave the machine running. When discrete observations and decision points were considered, the conditional risk of replacing the component at a scheduled downtime consisted of only the maintenance labour cost, shutdown and start up cost and the value of the component after depreciation. On the other hand, for the case of leaving the component running the conditional risk consisted of the total cost of the component operational failures, quality failures and maintenance activities. In the plant case study, the conditional risk for replacing the servo torque unit and conditional risk of keeping the unit running over its life cycle were calculated weekly for one observations set, 36 week life cycle. The comparison between the conditional risks showed that in the 6-state HMM, state 4 was the state at which the conditional risk for keeping the component running was larger than the conditional risk of replacing the servo torque unit. Based on the most probable sequence of states for that observations sequence, the first decision point at which the component was at state 4 was week no. 33 of its life cycle. This means that the developed method in this research predicted component was due to be replaced 3 weeks before its terminating failure state. Figure 7.14 presents the comparison between the conditional risks for both actions

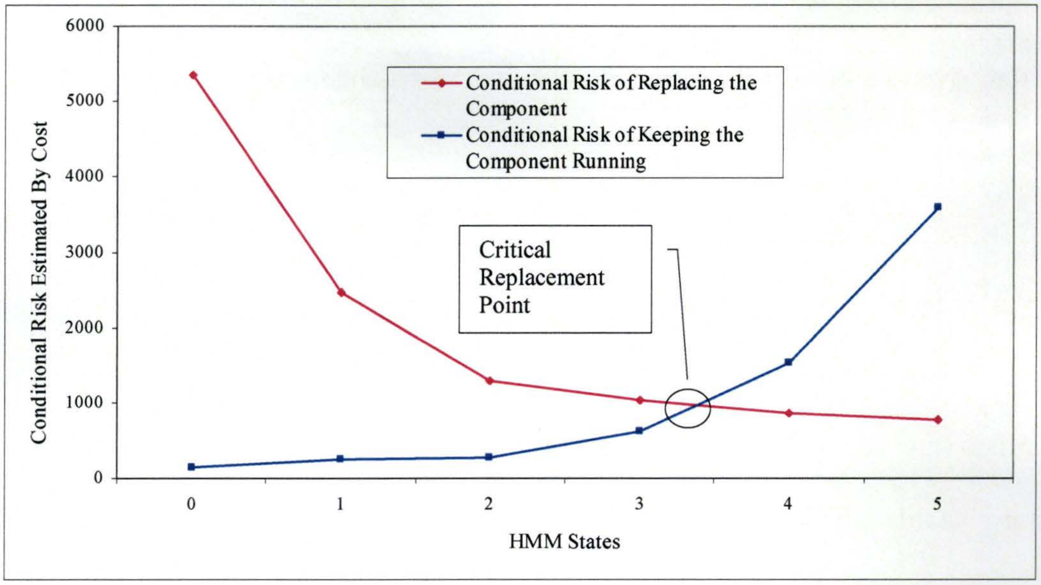


Figure 7.14 - Conditional Risk Comparisons between Keeping the Servo Torque Unit Running and Replacing the Unit at each HMM State

The conditional risk of replacing the component is high at the beginning states and decreases with the increase of the HMM state number. The high risk is explained by the losses expected when replacing component with good performance.

7.4.4 HMM States Relation to the Component Life Cycle

The relation between the HMM states and the component life cycle was interesting and highlighted some practical consideration in the decision making process. At the decision point level the deterioration of performance appears to be a semi-monotonic increasing trend which matched the expectations of gradual performance deterioration. On the other hand, the states of the HMM could be perceived as distinct points of time in the life cycle of the component. In this way they overlapped where the component shared a common distribution of performance for each phase. The estimation of the transition probabilities between the states represented the strength of using the HMM in these applications, as it provided a method to estimate the expected cost using the state probabilities.

7.5 Comparing the HMM results to Other Maintenance Planning Tools in the Plant

A comparison of results was performed for the servo torque unit – main unit failure mode, considering the new approach to the results of the existing methods which are currently used for the control policies. As discussed in section 6.2.2; there are three main maintenance strategies which are used for setting the maintenance control policy in a manufacturing system. The plant set the component replacement policy by run-to-fail, age-based, or condition based maintenance. The method which was developed in this research is considered as CBM approach. It was compared with other predictive analysis tools which were already available in the plant.

Run-To-Fail

Run-to-fail is the most common maintenance strategy in manufacturing systems and can only be improved by efficient reactive maintenance. In the case study, run-to-fail was the strategy which was used for the servo-torque gun – main unit replacement. There were no gains in that policy except maximizing the life cycle of a component. The losses had three factors. The first was the material and labour cost which were similar to the other maintenance strategies. The second factor was downtime which was needed to replace the component. The persistent failure condition occurred always during a scheduled production because the complete failure condition could only be detected at running time. The third part of the cost was the losses from the operational failures, quality failures and maintenance activities accumulated from the time of the optimal decision state to the time of the terminating state.

Age-Based Model

This model was only used in the case of the springs in the servo torque unit. Applying the age-based model to the servo torque unit would not be a practical option for many reasons. However, for the sake of comparison the age-based model could be applied with the same assumptions as used in the plant control policy. The age-based model was based on the shortest complete life cycle of the servo torque unit which was identified by the frequent episode analysis to be 27 production weeks at regular throughput levels.

CBM Model

The method developed in this research was considered as a condition based maintenance approach where conditions of the component were used to specify the optimal control plan. There were other predictive analysis tools which were developed in the research platform to predict the performance of a component on a machine. These tools included the Finite State Markov Machine for predicting trends in time series data, curve fitting and evolutionary hybrid data mining. The predictive tools were only based on operational failures. This would make the comparison with the method developed in this research only valid if the cost impact due to operational failures is considered.

Based on the results in section 7.4.3 and the assumptions mentioned above, a comparison was made on the servo torque unit replacement using the existing tools and the new method developed in this research. The comparison was made on one dataset of servo torque unit life cycle, the 36 week dataset. The comparison calculated the point of time for component replacement. The results are presented in Figure 7.15

- Run-to-fail approach

  The real data from the plant represented run-to-fail approach. The component was replaced after 36 week. The cost of replacement in scheduled runtime was estimated by 30 min downtime.

- Age-based approach

  As it was discussed before the age-based approach was based on the shortest life cycle which was identified in dataset as 27 week. There was no estimation of number of cycles in this case.

- The operational failure prediction tools

  When the three prediction tools available were applied to the operational failure records, only one failure classified for the servo torque component showed an increasing trend which was failure 645-1. The three tools showed different prediction values with different confidence levels. The cost was estimated on the evolutionary hybrid data mining. The component replacement decision was assumed to be made when the total downtime caused by the estimated operational failure exceeds the cost of replacing the component in preventive maintenance activity.

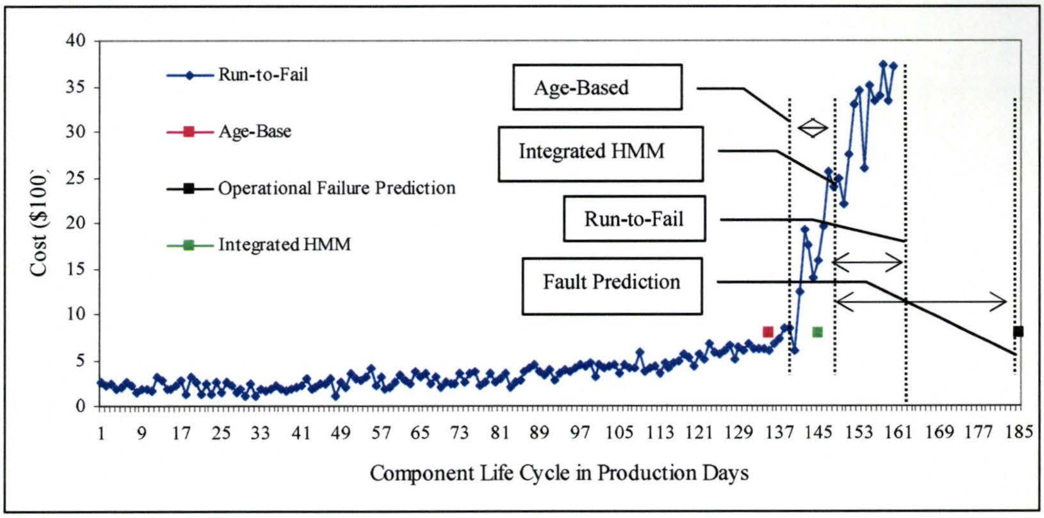


Figure 7.15 - Comparison between Integrated HMM Method, Run-to-Fail, Age-based and Prediction Tools

Figure 7.15 shows that the operational failures (machine faults) which were used in the prediction tools. The tool predicted that the component replacement would take place almost 5 weeks after the persistent failure, the terminating state. This could be explained because of the prediction tool analysis was based only on operational failures and did not include other failures/. The total cost of one operational failure was considerably lower than the combined costs of quality and operational failures. For that reason, the operational failure needed more time to initiate replacement decision.

The age based approach looked very appropriate for this dataset as the point of time for replacement was selected just before the increase in performance deterioration. However, same age would not be useful for a shorter life cycle because of the cost associated with failures and maintenance activities at the deterioration stage.

The comparison between the existing tools and the developed method showed that the integrated HMM would provide more accurate estimation of the optimal replacement state. It combined all costs and provided a robust decision support tool which was not sensitive for the variations in the length of the component life cycle.

Table 7.6 summarizes the comparison between the integrated HMM approach and the other tools.

Method	Component Type	Computations Complexity	Analysis Timeframe	Decision Process
Run-to-Fail	Only justified for expensive components which have no deterioration pattern	No computations evolved	No time tracking	Replacement is made after failure. Failure is defined by decision maker
Age-based	Very conservative estimation of component life cycle – good for inexpensive components	Based on stationary and deterministic deterioration. Age in life cycles should be converted to production time	Beginning of life cycle should be identified. Age in number of cycles converted to production time	Replacement is made at pre-defined time according to PM schedule
Prediction - One Data Source	Good for components with deterioration patterns which can be modeled by one type of failure	Extensive computations based on artificial intelligent analysis	Large historical data is needed for training. More weights is given to short historical data	Failures are predicted by the tool. Decision maker should evaluate the risk.
HMM - Integrated Analysis	Good for components which deteriorate as a Markov process	Extensive computations based on artificial intelligent analysis	Beginning of life cycle should be identified. Decision points should be identified	The method evaluate the risk and suggest optimal decision

Table 7.6 - Comparison between the HMM Integrated Approach and Other Methods Used in the Plant for Component Replacement Control Policy

7.6 Integration of the Decision Support Tool in the Plant Business Model

As the case study of this research was done in an automotive plant, the analytical tool which was developed in this research had to fit within GM-GMS relevant principles to be integrated in the business model. It was very clear that GM-GMS items provided prior knowledge and guidelines for planning and decision making processes; however it was the accurate estimation of the SPQRC measures (observations) that was essential in making the decisions to achieve the optimal performance.

In the current practice, decisions which were made within the GM-GMS framework were initiated by the review of the Business Plan Deployment (BPD). The BPD worked as a closed loop process of four items: plan, do, check and action. The MES which was developed in this research provided real time monitoring capabilities which were able to monitor the QRC metrics for the components. The integrated HMM approach which was developed in this research identified the deterioration state at which replacing the component would be an optimal decision. The model used to identify the deterioration state sequence was based on the decision point concept.

For full integration within the plant control policies, the method which was developed in the research should be used to create software agent within the existing MES. The agent should continuously track the observations from operational failures, quality failures and maintenance activities for each component of the system that requires a replacement control policy. Accordingly it should create a tentative maintenance work order within MAXIMO® (Maintenance Management System). This would leave the final decision to the plant maintenance group. Thus, it would fulfill the main objective of this research which was providing the decision makers in the manufacturing system with the tools needed for the decision making processes.

Chapter 8

Conclusions and Recommendations

The research in this thesis addressed some common problems in today's manufacturing systems by focusing on the maintenance decision making processes in mass-production automotive lines. It outlined the development of a new tool for maintenance management based on the integration of throughput, quality and maintenance data using frequent episodes analysis and HMM.

8.1 Conclusions

- The component-based data structure which was introduced in this research provided a solution for one of the main challenges facing the decision makers, which is data and information overload. The proper linking between the system components and their data sources significantly helped in transforming the data into actionable information. The results of the research findings were reflected in an accurate real time analysis and decision support for many decision makers in the plant.

- The research introduced a new approach for the information to knowledge transformation in manufacturing systems. This was achieved by integrating the throughput losses, quality failures and maintenance activities into a single cost performance function per component per failure mode. The approach was based on integrating the results of the frequent episode analysis with the HMM in the performance deterioration model.

- The frequent episodes analysis was improved to handle the uncertainty in the event stamp time of quality failures as a result of quality information lags. Composite episodes with slight differences in the order of the quality failure events were considered as one type of composite episodes. Operational failure and maintenance activities maintained the proper order for the identified episodes.

- The size of the search window for some composite episodes varied between failure modes. The analysis also showed differences in the length of the sequences of composite episodes for the failure mode in the same component. This was explained by the changes of the process parameters such as torque parameters and throughput levels over the period of data collection.

- The number of frequent episodes which were identified over the life cycle of the component through the performance deterioration process was used as initial values for the number of the states in the HMM learning.

- The results of the frequent episode analysis depended on a set of pre-filters and classification tools for the failures and maintenance records. It also relied on a set of post-filters that removed trivial correlations and combined what could be similar episodes.

- The HMM state transition probabilities were in line with the plant maintenance records especially for the components with run to fail control policies. Components which were replaced prematurely under preventive maintenance program showed good results only in the last section of the operation period. The relatively good estimation by the HMM for the deterioration in performance was explained mainly by preprocessing of the dataset to provide harmonized training dataset.

- The HMM provided a suitable framework to deal with the uncertainty of the data in the decision making process. The concept of modeling deterioration by sequence of states provided a better analogy for the decision makers in the plant where decisions were made based on alarm limits.

- The approach used in this research was suitable for monotonic and gradual deterioration in the performance. Cases like "silent failures" in which the gradual deterioration is not observed cannot be identified using frequent episodes analysis and cannot be modeled in HMM framework.

8.2 Suggestions for Future Work

- The level of objectifying system components needs more analysis to set a standardized process. The very broad approach used in this research considered every component with data source as an independent object. While this is theoretically sound, it may be practically unnecessary and also it may affect data integrity. Thus it is more reasonable to consider only entities with independent decision processes and independent data sources as objects.

- The complexity in identifying composite episodes created multiple episodes for the same set of events. These cases were handled manually in this research. An improved algorithm could possibly identify variations in the same composite episode and automate the judgment on whether they can be considered as one type of composite episode.

- The frequent sequences which were discovered in the constructed dataset of the composite episodes were identified using a second level of frequent episode analysis. This step had some difficulties because of the differences of composite episodes within the dataset. Other pattern recognition techniques may be explored to improve this process.

- The estimation of total cost was based on the assumption that machines containing the components in the study were not embedded in a production line. This was a reasonable approximation for the decision processes. However, under certain conditions the costs associated with throughput losses, quality failures and maintenance activities may vary with the production line conditions. This includes line constraints, buffers sizes and re-works loop sizes. A more accurate decision can be made if the cost function includes the other line properties.

- In this research the cost of external quality failures were removed from the cost calculations. The reason behind that was the timeframe of external failure detection which was long after the decision timeframe for component replacement. However, external quality failures can be accounted for if the RPN from the real time PFMEA is considered. In this case a weight function based on the RPN can be used to increase the estimated cost of internal quality failures in proportion with the probability of creating external quality failures.

- The left-to-right HMM used in this research is a very logical model given the nature of deterioration of components performance. However, changes in the manufacturing process parameters may improve the performance of a component without replacements or repairs such as changes in the product type. In the case of engine assembly, the production line was designed to build different types of the same engine family. It is very common to notice bad performance of the machines when building some engine types over others. The current approach can be expanded to include the product type in the model.

- The selection of the number of states for the most appropriate HMM to conform to the deterioration model was done manually. It is based on optimizing the probability of observations given the models options. For practical implementation of the method, this task should be automated to update the existing HMM of component and to build new deterioration HMM for other components with no user involvements.

- The integrated HMM approach was applied to components of machines. This approach can be expanded to other manufacturing system components that need other types of decisions. Constraint management is one field as the dynamics that create a temporary throughput constraint can be considered as a Markov process.

- The data used in the development of this analysis tools is real plant data. Simulation data can be used to test the validity of the model over wider range of variations. However the plant dataset used in this research was enough to represent the variations of the performance deterioration process. Also, the plant data allows the proper comparison between the existing control policies and the approach which was developed in this research.

References

[1] Melody Yihwa Kiang; Robert Chi; "A Distributed Knowledge Acquisition System in a DSS"; Journal of Management Information System, Spring 1993 Vol.9 No 4 pp 59-82

[2] Chrissanthi Angeli; Derek Atherton; "A Model-based Method for an Online Diagnostic Knowledge-based System"; Expert Systems, July 2001, Vol. 18, No. 3

[3] Behzad Foroughi; "Decision –Making in Manufacturing Systems: An IPA/ Fuzzy Approach"; Master Thesis Mechanical and Manufacturing Engineering Calgary Alberta April 1999

[4] Prasanta Kumar Dey; "Decision Support System for Inspection and Maintenance: A Case Study of Oil Pipelines"; IEEE Transactions on Engineering Management, Vol. 51, No. 1, February 2004

[5] Xiaoyue Jaing; "Modeling and Optimization of Maintenance Systems"; Ph.D. Thesis Department of Mechanical Engineering, University of Toronto 2001

[6] Haritha Saranga; "Relevant Condition-Parameter Strategy for an Effective Condition-based Maintenance"; Journal of Quality in Maintenance Engineering, Vol. 8 No. 1 2002 pp.92-106

[7] Daniel J. Fonseca; "A Knowledge-based System for Preventive Maintenance"; Expert Systems, November 2000, Vol. 17, No. 5

[8] Basim Al-Najjar; Wenbin Wang; "A Conceptual Model for Fault Detection and Decision Making for Rolling Element Bearings in Paper Mills"; Journal of Quality in Maintenance Engineering, Vol. 7 No. 3 2001 pp.192-206

[9] Jongyoon Kim; "Integrated Quality and Quantity Modeling of a Production Line"; PhD Mechanical Engineering, MIT 2005

[10] Fayyad; U., Madigan; Piatetsky-Shapiro; "Data Mining to Knowledge Discovery In Databases"; P. (1996) AI Magazine, 17, 37-54

[11] Ali Kamrani; Wang Rong; Ricardo Gonzalez; "A Generic Algorithm Methodology for Data Mining and Intelligent Knowledge Acquisition"; Computer & Industrial Engineering 40 (2001) 361-377

[12] Chih-Min Fan; Ruey-Shan Guo; Kuo-Ching Hsu; "Data Mining and Fault Diagnosis Based on Wafer Acceptance Test Data and In-Line Manufacturing Data"; 0-7803-6731-6/01, 2001 IEEE

[13] Russ M. Dabbas; Hung-Nan Chen; "Mining Semiconductor Manufacturing Data for Productivity Improvement - An Integrated Relational Database Approach"; Computers in Industry 45 (2001) 29-44

[14] Heikki Mannila; Hannu Toivonen; A. Inkeri Verkamo; "Discovery of Frequent Episodes in Event Sequences"; Data Mining and Knowledge Discovery 1, 259–289 (1997)

[15] Srivatsan Laxman; Basel Shadid; K. P. Unnikrishnan and P. S. Sastry; "Application of temporal data mining for analyzing fault-report logs in an assembly line"; submitted for publishing

[16] Carey Bunks; Dan McCarthy; Tarik Al-Ani; "Condition-based Maintenance of Machines Using Hidden Markov Models"; Mechanical Systems and Signal Processing (2000) 14(4), 597-612

[17] Allen H. Tai; Wai-Ki Ching; L.Y. Chan; "Hidden Markov Model For The Detection Of Machine Failure" ; Department Of Industrial And Manufacturing Systems Engineering ;The University of Hong Kong

[18] Jie Ying; T. Kirubarajan; Krishna R. Pattipati; and Ann Patterson-Hine; "A Hidden Markov Model-based Algorithm for Fault Diagnosis with Partial and Imperfect Tests"; IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, Vol. 30, No. 4, November 2000

[19] Irvin Schick; Stanley Gershwin; Jongyoon Kim; "Quality / Quantity Modeling and analysis of Production Lines Subject to Uncertainty"; GM R&D Cr – 2005-03

[20] Julie Simmons Ivy; Stephen Pollock; "Marginally Monotonic Maintenance Policies for a Multi-State Deteriorating Machine with Probabilistic Monitoring, and Silent Failures"; IEEE Transactions on Reliability, Vol. 54, No. 3, September 2005

[21] Donald C.K. Chan; K.L. Yung; Andrew W.H.; "An Application of Fuzzy Sets to Process Performance Evaluation"; Integrated Manufacturing Systems 13/4 [2002] 237-246

[22] Mustafa Yurdakul; "Measuring a Manufacturing System's Performance Using Saaty's System with Feedback Approach" Integrated Manufacturing Systems 13/1 [2002] 25-34

[23] Muammer Koç; Jun Ni; Jay Lee; "Introduction of E-Manufacturing"; NAMRC 2003 E-Manufacturing Panel, McMaster Univ. May 2003

[24] Jay Lee; "E-manufacturing—fundamental, tools, and transformation"; Robotics and Computer Integrated Manufacturing, Sep, 2003

[25] Pingyu Jiang; Zhou; Yong Liu; "ASP-Driven E-Service Platform for Web-based Online Manufacturing"; Integrated Manufacturing Systems 13/5 [2002] 306-318

[26] Srikanth Krishnamurthy; Ibrahim Zeid; "Distributed and Intelligent Information Access in Manufacturing Enterprises through Mobile Devices"; Journal of Intelligent Manufacturing 15, 175-186, 2004

[27] US The Department of Energy (2000), http://www.energy.gov/ (Aug 26, 2007)

[28] Pulak Bandyopadhyay; Stephan Biller; Guoxian Xiao; "E-Manufacturing White Paper"; ESL – 129; GM R&D Enterprise Systems Lab

[29] S.A. Ehikioya; E.G. Truelove; T.T. Tran; "Intelligent Database Support for Manufacturing and Processing of Industrial Materials"; Proceedings of the Second International Conference on Volume 1, Issue , Jul 1999 Page(s):139 - 144 vol.1

[30] Srivatsan Laxman; "Discovering frequent episodes: Fast algorithms, connections with HMMs and generalizations"; PhD Thesis, Department of Electrical Engineering, Indian Institute of Science, March 2006

[31] Carl Howard Mooney; "The Discovery of Interacting Episodes and Temporal Rule Determination in Sequential Pattern Mining"; PhD thesis - Flinders University of South Australia, December 1, 2006

[32] Lawrence R. Rabiner; "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"; IEEE Vol.77 No.2 February 1989

[33] Edi Karni; "Bayesian Decision Theory and the Representation of Beliefs"; Johns Hopkins University, Department of Economics, September 25, 2006

[34] Shih-Yang Chiao and Costas S. Xydeas; "Behavior Modeling Using a Hierarchical HMM Approach" ; Fourth International Conference on Hybrid Intelligent Systems (HIS'04)

[35] Michael T. Johnson; "Capacity and Complexity of HMM Duration Modeling Techniques"; IEEE Signal Processing Letters, Vol. 12, No. 5, May 2005

[36] Srivatsan Laxman; P.S. Sastry; K.P. Unnikrishnan "Discovering Frequent Episodes and Learning Hidden Markov Models: A Formal Connection"; IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 11, November 2005

[37] Kyoung-Jae Won; Adam Prügel-Bennett; Anders Krogh; "Evolving the Structure of Hidden Markov Models"; IEEE Transactions on Evolutionary Computation, Vol. 10, No. 1, February 2006

[38] Yangsheng Xu; Ming Ge; "Hidden Markov Model – based Process Monitoring System"; Journal of Intelligent Manufacturing 15, 337-350, 2004

[39] Litao Wang; Mostafa G. Mehrabi ; Elijah Kannatey-Asibu Jr. ; "Hidden Markov Model-based Tool Wear Monitoring in Turning"; Engineering Research Center for Reconfigurable Machining Systems University of Michigan, Ann Arbor

[40] Les Atlas; Mari Ostendorf; Gary D. Bernard; "Hidden Markov Models for Monitoring Machining Tool-Wear"; Department of Electrical Engineering; University Of Washington

[41] Shi Zhong; Joydeep Ghosh; "HMMs and Coupled HMMs for Multi-channel EEG Classification"; Department of Electrical and Computer Engineering. The University of Texas at Austin

[42] Julie Simmons Ivy; Stephen M. Pollock; "Marginally Monotonic Maintenance Policies for a Multi-State Deteriorating Machine with Probabilistic Monitoring, and Silent Failures"; IEEE Transactions on Reliability, Vol. 54, No. 3, September 2005

[43] Dongxin Xu; Craig Fancourt; Chuan Wang; "Multi-Channel HMM"; 0-7803-3192-3/96 1996 IEEE

[44] L. P. Khoo; L. Y. Zhai; "Multi-Concept Classification of Diagnostic Knowledge to Manufacturing Systems: Analysis of Incomplete Data with Continuous-Valued Attributes"; Int. J. Prod. Res., 2001, Vol. 39, No. 17, 3941-3957

[45] Mark Hasegawa-Johnson; "Multivariate-State Hidden Markov Models for Simulation Transcription of Phones Formants"; Doctoral Thesis

[46] Wei Wu; Ari Arapostathis; "Optimal Control of Stochastic Systems with Costly Observations - The General Markovian Model and the LQG Problem"; 2005 American Control Conference June 8-10, 2005. Portland, OR, USA

[47] David Heckerman; "A Tutorial on Learning with Bayesian Networks"; Technical Report MSR-TR-95-06, Advanced Technology Division, Microsoft Corporation

[48] Qiang Huoyz ; Chorkin Chany; " Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition" ; Department of Computer Science; The University of Hong Kong

[49] John Breese; David Heckerman; "Decision – Theoretic Troubleshooting: A Framework for Repair and Experiment". The Twelfth Conference on Uncertainty Intelligence, August 1996

[50] Ross Shachter;"Decision-Theoretic Foundations for Causal Reasoning"; Heckerman and Shachter (1995). Journal of Artificial Intelligence Research, 3:405-430.

[51] Eyal Amir; "Learning Partially Observable Deterministic Action Models"; Computer Science Department -University of Illinois, Urbana-Champaign

[52] Xi-Ren Cao; Xianping Guo; "Partially Observable Markov Decision Processes With Reward Information"; 43rd IEEE Conference on Decision and Control, December 14-17, 2004; Atlantis, Paradise Island, Bahamas

[53] Michael Jonas; "Modeling Speech using a Partially Observable Markov Decision Process"; PhD dissertation, Computer Science - TUFTS UNIVERSITY

[54] Ron Sun; Chad Sessions; "Self-Segmentation of Sequences: Automatic Formation of Hierarchies of Sequential Behaviors"; IEEE Transactions

on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 30, No. 3, June 2000

[55] Peter L. Bartlett; Jonathan Baxter; "Stochastic Optimization of Controlled Partially Observable Markov Decision Processes"; Proceedings of the 39 IEEE Conference on Decision and Control Sydney, Australia December, 2000

[56] Charles H. Rosa; Datta Kulkarni; Jeffrey Tew "Web-Services Technology"; CL2005 12 MSR; GM R&D Manufacturing Systems Research Lab

[57] Jingshan Li; Dennis Blumenfeld; Samuel P. Marin; "Manufacturing System Design to Improve Quality Buy Rate"; GM R&D-10031

[58] Massoud Aghamirian; Jorge Arinez;Pulak Bandyopadhyay; "Real-Time Decision Making For Flexible Manufacturing Systems"; Cl – 2004 – 30; GM R&D Manufacturing Systems Research Lab

[59] Narayan Srinivasa; Leandro G. Barajas Guoxian Xiao Pulak Bandyopadhyay; "Prognostic Tools for E-Manufacturing"; MSR– 210; GM R&D Manufacturing Systems Research Laboratory

[60] Frank Shin; Bala Ram; Roland J. Menassa "A Decision Tool for Assembly Line Breakdown Action", CL –2004-15-MSR; GM R&D Manufacturing Systems Research Lab

[61] T. L. Murlidharan; K. Balaji Rao; J. Durgaprasad; T. V. S. R. Appa Rao; "Handling Uncertainties in Knowledge Modeling and Acquisition for the Development of a Knowledge-based System"; Computer-Aided Civil and Infrastructure Engineering 14 (1999) 141–154

[62] Melody Yihwa Kiang; Robert Chi; "A Distributed Knowledge Acquisition System in a DSS"; Journal of Management Information System, Spring 1993 Vol.9 No 4 pp 59-82

[63] Paul G. Ranky; Mick F. Ranky; "A Dynamic Operation Control Algorithm with Multimedia Objects for Flexible Manufacturing Cells and Systems"; Int. J. Computer Integrated Manufacturing, 2000, Vol. 13, No. 3, 245 - 262

[64] George Rzevski; "A Framework for Designing Intelligent Manufacturing System"; Computers in Industry 34(1997) 211-219

[65] Bjorn Regnell; Michael Andersson; Johan Bergstrand; "A Hierarchical Use Case Model with Graphical Representation"; ECBS'96, IEEE International Symposium and Workshop on Engineering of Computer-based Systems, March 1996.

[66] Ya Wu; Philippe Escande; R. Du ; "A New Method for Real-Time Tool Condition Monitoring in Transfer Machining Stations" ; Journal of Manufacturing Science and Engineering May 2001, Vol. 123

[67] Sandeepa Goel; "A Simulation Approach to Track Quality, Cost and Time during the Manufacturing of Metal Bonded"; PhD Dissertation Department of Mechanical and Industrial Engineering, Winnipeg Manitoba, Canada

[68] Muammer Koç; Jay Lee; "A System Framework for Next-Generation E-Maintenance Systems"; Center for Intelligence Maintenance Systems (IMS), University of Wisconsin-Milwaukee, USA

[69] Yulan Shen; "Combining Fuzzy Logic and Neural Networks for Decision Support in an Industrial Environment"; Master Thesis Department of Mechanical Engineering University of Montréal December 1996

[70] Basim Al-Najjar; "A Concept for Detecting Quality Deviation Earlier than When Traditional Diagram in Automotive: A Case Study"; Journal of Quality in Maintenance Engineering, Vol. 18 No. 9 2001 pp.917-940

[71] Kanthi Mathi Nathan Muthiah; "Diagnostic Factory Productivity Metrics"; Master of Science, Industrial Engineering, University of Cincinnati, 2001

[72] Chin-Yin Huang; "Distributed Manufacturing Execution Systems: A Workflow Perspective"; Journal of Intelligent Manufacturing 13, 485-497, 2002

[73] Jay Lee; "E-Intelligence Strategies for Product and Manufacturing Innovation"; Intelligent Maintenance Systems (IMS) Center; Univ. of Wisconsin-Milwaukee

[74] Sudhindra Kulkarni; Vinayak Oak; "E-manufacturing & SAP-Creating Responsive Shop Floor in the Supply Chain"; WIPRO white paper, Dec. 2003

[75] "Exploiting E-Manufacturing: Interoperability of Software Systems Used by U.S. Manufacturers"; February 2001 National Coalition for Advanced Manufacturing (NACFAM) Washington, D.C.

[76] Yu Ding ; Dariusz Ceglarek; Jianjun Shi ; "Fault Diagnosis of Multistage Manufacturing Processes by Using State Space Approach"; Journal of Manufacturing Science and Engineering ;MAY 2002, Vol. 124 /313

[77] Duncan Stevenson; Matt Adcock; Gary White; Leandro Barajas; "Immediate Impact Opportunities for Wearable Computing Application in Plant Floor Systems"; CL-2006/26/MSR

[78] P. Wang, N. Propes ; N. Khiripet ; Y. Li ; G. Vachtsevanos ; "An Integrated Approach to Machine Fault Diagnosis"; School of Electrical and Computer Engineering; Georgia Institute of Technology; Atlanta, Georgia 30332-0250

[79] Ming Rao; Haibin Yang; Heming Yang; "Integrated Distributed Intelligent System Architecture for Incidents Monitoring and Diagnosis"; Computers in Industry 37 1998.143–151

[80] Marek B. Zaremba; Geranrd Morel; "Integration and Control of intelligent in Distributed Manufacturing"; Journal of Intelligent Manufacturing 14, 24-42 ; 2003

[81] Sanjiv Patel; Ali Kamrani; "Intelligent Decision Support System for Diagnosis and Maintenance of Automated Systems"; Computers and Industrial Engineering 30 No. 2, pp 297-319, 1996

[82] Tung-HSU Hou; Wang –Lin; Li Lin; "Intelligent Remote Monitoring and Diagnosis of Manufacturing Process Using an Integrated Approach of Neural Network and Rough Sets"; Journal of Intelligent Manufacturing, 14 239-253 , 2003

[83] Jack Hollingum; "Internet Strategies for Manufacturing"; Assembly Automation; Volume 20. Number 2. 2000. pp. 140-143

[84] Soo-Yeo Ryu and Jong Hee Park; "An Integrated Fault Diagnostic Mechanism Based on Statistical characteristics of Products for Heterogeneous Manufacturing Environments"; Computer Integrated Manufacturing System Vol. 11, No1-2 " 35-42, 1998

[85] Subash Babu; S.A. George; R.P. Mohanty; "Manufacturing Performance Measurement Systems: A Review"; Int. J. Manufacturing Technology and Management, Vol. 5, Nos. 5/6, 2003

[86] Terrence R.M. Lalla; Winston G. Lewis; Kit Fai Pun; Kwai Sang Chin; "Manufacturing Strategy, Total Quality Management and Performance Measurement: An Integrated Model"; Int. J. Manufacturing Technology and Management, Vol. 5, Nos. 5/6, 2003

[87] J. Li;D.E. Blumenfeld; N. Huang;R.R. Inman ; S.P. Marin; "Manufacturing System Design For High Product Quality"; GM R&D Manufacturing Systems Research Lab

[88] Jay Lee; "Measurement of Machine Performance Degradation Using a Neural Network Model"; Computers in Industry 30(1996) 193-209

[89] Sev. V. Naalingam; Grier C. I. Lin; "A Methodology to Select Optimal System Component for Computer Integrated Manufacturing by Evaluating Synergy"; Computer Integrated Manufacturing Systems Vol. 11, No. 3. P.217-228, 1998

[90] Jeffrey L. Stein; Kunsoo Huh; "Monitoring Cutting Forces In Turning: A Model-based Approach"; Journal of Manufacturing Science and Engineering; Vol. 124, February 2002

[91] Kossi Adzakpa; Farouk Aylaoui; "On-Line Maintenance Job Scheduling and Assignment to Resources in Distributed systems by Heuristic-based Optimization"; Journal of Intelligent Manufacturing 15, 131-140, 2004

[92] M A Sharif; R I Grosvenor; "Process Plant Condition Monitoring and Fault Diagnosis" Proc Instn Mech Engrs Vol 212 Part E

[93] Muammer Koç; "Remote Monitoring and Predictive Maintenance of Machine Tool Systems"; NSF Workshop April 29, 2003 Ann Arbor, Michigan

[94] Richard Cassady; Edward Phhl; W. Paul Murdock; "Selective Maintenance Modeling for industrial Systems"; Journal of Quality in Maintenance Engineering Vol.7 No.2, 2001

[95] P. Olley; A. K. Kochhar; "Self-Supervised Learning for an operational Knowledge-based system"; Computer Integrated Manufacturing Systems Vol.11 no. 4 P.297-308, 1998

[96] W. Hu; A.G. Starr; Z. Zhou b; A.Y.T. Leung; "A Systematic Approach to Integrated Fault Diagnosis of Flexible Manufacturing Systems"; International Journal of Machine Tools & Manufacture 40 (2000) 1587–1602

[97] Basim Al-Najjar; "Total Quality Maintenance: An Approach for Continuous Reduction in Costs of Quality Products"; Journal of Quality in Maintenance Engineering, Vol. 2 No. 3, 1996 pp. 4-20.

[98] D. Tony Liu; X. Williams Xu; "A Review of Web-based Product Data Management Systems"; Computer in Industry 44 (2001) 251-262

[99] Lihui Wang; Weiming Shen; Sherman Lang; "Wise-Shop Floor: A Web-based and Sensor-Driven e-Shop Floor"; Journal of Computing and Information Science in Engineering"; Vol. 4 March 2004

[100] Robert St. Amant; Yoshitaka Kuwata; Paul R. Cohen; "Monitoring Progress with Dynamic Programming Envelopes"; Proceedings of the Seventh International IEEE Conference on Tools with Artificial Intelligence

[101] Ilya Shpitser; Judea Pearl; "Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models"; Twenty First National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA 1219-1226, July 2006

[102] Blai Tirant Bonet Bretto; "Modeling and Solving Sequential Decision Problems with Uncertainty and Partial Information"; PhD Thesis Department of Computer Science University of California Los Angeles

[103] Judea Pearl; "Bayesian Networks"; M. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, 2001

[104] Anantaram Balakrishnan; Soundar R. T. Kumara; Shankar Sundaresan; "Manufacturing in the Digital Age; Exploiting Information Technologies for Product Realization" Information Systems Frontiers, Kluwer Publishers, Inaugural Issue, Feb. 1999

[105] Andrew Kusiak; "Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing"; IEEE transactions on Electronics Packaging Manufacturing, Vol. 24, No. 1, January 2001

[106] Christine Gertosio; Alan Dussauchoy; "Knowledge Discovery from Industrial Databases"; Journal of Intelligent Manufacturing, 15, 29-37, 2004

[107] F. G. Filip; D. A. Donciulescu; Cr. I. Filip; "Towards Intelligent Real-time Decision Support Systems for Industrial Milieu"; Studies in Informatics and Control, Vol.11, No.4, December 2002

[108] David A. Koonce; Cheng-Hung Fang; Shi-Chi Tsai "A Data Mining Tool for Learning from Manufacturing Systems"; Computers Ind. Eng. Vol. 33,1-2, pp. 27-30, 1997

[109] A. P. Tchangani; "Decision-making with uncertain data: Bayesian linear programming approach"; Journal of Intelligent Manufacturing, 15, 17-27, 2004

[110] Raymond J. Curts; Douglas E. Campbell; "Avoiding Information Overload through the Understanding of OODA Loops, A Cognitive Hierarchy and Object-Oriented Analysis and Design"; Command & Control Research & Technology Symposium (CCRTS), 2001

[111] David S. Cochran; Yong-Suk Kim; Jongyoon Kim; "The Impact of Performance Measurement on Manufacturing System Design"; First International Conference on Axiomatic Design, Cambridge, MA – June 2000

[112] Nick Jennings; Michael Wooldridge; "Software Agents"; IEE Review, January 1996, pp 17-20

[113] Zhengxin Chen; "Computational Intelligence for Decision Support"; CRC Press International Series on Computational Intelligence.

[114] Sagar Hasmukh Savla; "Efficient Main Memory Algorithms for Significant Interval and Frequent Episode Discovery"; Master of Science in Computer Science. University of Texas at Arlington, December 2006

[115] W. Ching; M. Ng; "Markov Chains: Models, Algorithms and Applications"; 2006, ISBN: 0-387-29335-3

[116] Dejan Mitrovic; "Reliable Method for Driving Events Recognition"; IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No. 2, June 2005

[117] Ronnie Bathoorn; Arno Siebes; "Finding Composite Episodes"; The 18th European Conference on Machine Learning, Warsaw, Poland 2007

APPENDIX 1

Frequent Episodes Algorithms

[Mannila, Toivonen and Verkamo, 1997 [14]]

Algorithm 1
Input: *A* set *E* of event types, an event sequence *s* over *E*, a set ε of episodes, a window width *win*, a frequency threshold *min_fr*, and a confidence threshold *min_conf*.
Output: The episode rules that hold in s with respect to *win*, *min_fr*, and *min_conf*.
Method:
1. /* Find frequent episodes (Algorithm 2): */
2. compute $\mathcal{F}$(s, *win*, *min_fr*);
3. /* Generate rules: */
4. *for* all $\alpha \in \mathcal{F}$(s, *win*, *min_fr*) *do*
5. *for* all $\beta \prec \alpha$ *do*
6. *if* $fr(\alpha)/fr(\beta) \geq = min\_conf$ *then*
7. output the rule $\beta \rightarrow \alpha$ and the confidence $fr(\alpha)/fr(\beta)$

Algorithm 2
Input: *A* set *E* of event types, an event sequence s over *E*, a set ε of episodes, a window width *win*, and a frequency threshold *min_fr*
Output: The collection $\mathcal{F}$(s, *win*, *min_fr*) of frequent episodes.
Method:
1. $C_1 := \{\alpha \in \varepsilon \mid |\alpha|=1\}$;
2. $l := 1$;
3. *while* $C_l \neq 0$; *do*
4. /* Database pass (Algorithms 4 and 5): */
5. compute $:= \{\alpha \in C_l \mid fr(\alpha, s, win) \geq min\_fr\}$;
6. $l := l + 1$;
7. /* Candidate generation (Algorithm 3): */
8. compute $C_l = \{\alpha \in \varepsilon \mid |\alpha| = l$ and for all $\beta \in \varepsilon$ such that $\beta \prec \alpha$ and $|\beta| < l$ we have $\beta \in \mathcal{F}_{|\beta|}\}$;
9. *for* all *l* *do* output $\mathcal{F}_l$;

Algorithm 3
Input: *A* sorted array $\mathcal{F}_l$ of frequent parallel episodes of size *l*.
Output: *A* sorted array of candidate parallel episodes of size $l+1$.
Method:
1. $C_{l+1} := 0$;
2. $k := 0$;
3. *if* $l = 1$ *then for* $h := 1$ *to* $|\mathcal{F}_l|$ *do* $\mathcal{F}_l.block\_start[h] := 1$;
4. *for* $i := 1$ *to* $|\mathcal{F}_l|$ *do*
5. *current_block start* $:= k + 1$;

6. *for* (j :=I; $\mathcal{F}_l$.*block_start*[j] = $\mathcal{F}_l$.*block_start*[i]; j := j + 1) *do*
7. /* $\mathcal{F}_l[i]$ and $\mathcal{F}_l[j]$ have l -1 first event types in common,
8. build a potential candidate α as their combination: * /
9. *for* x := 1 *to* l *do* $\alpha[x] := \mathcal{F}_l[i][x]$;
10. $\alpha[l+1] := \mathcal{F}_l[j][l]$;
11. /* Build and test subepisodes β that do not contain $\alpha[y]$: */
12. *for* y := 1 *to* l -1 *do*
13. *for* x := 1 *to* y -1 *do* $\beta[x] := \alpha[x]$;
14. *for* x := y *to* l *do* $\beta[x] := \alpha[x+1]$;
15. *if* β is not in $\mathcal{F}_l$ *then* continue with the next j at line 6;
16. /* All subepisodes are in $\mathcal{F}_l$, store α as candidate: */
17. $k := k + 1$;
18. $C_{l+1}[k] := \alpha$;
19. C_{l+1}.*block_start*[k] := *current_block_start*;
20. output C_{l+1};

Algorithm 4

Input: *A* collection C of parallel episodes, an event sequence **s** = (s, T_s, T_e), a window width *win*, and a frequency threshold *min_fr*.

Output: The episodes of C that are frequent in **s** with respect to *win* and *min_fr*.

Method:

1. =* Initialization: *=
2. *for* each α in C *do*
3. *for* each A in α *do*
4. A.*count* := 0;
5. *for* i := 1 *to* $|\alpha|$ *do* *contains*(A, i) := 0;
6. *for* each α in C *do*
7. *for* each A in α *do*
8. a := number of events of type A in α;
9. *contains*(A, a) := *contains*(A, a) $\cup$ $\{\alpha\}$;
10. α.*event_count* := 0;
11. α.*freq_count* := 0;
12. /* Recognition: */
13. *for start* := T_s - *win* +1 *to* T_e *do*
14. /* Bring in new events to the window: */
15. *for* all events (A, t) in s such that t = *start* +*win* - 1 *do*
16. A.*count* := A.*count* + 1;
17. *for* each $\alpha \in$ *contains.A*; *A:count/ do*
18. α.*event_count* := α.*event count* + A.*count*;
19. *if* α.*event count* = $|\alpha|$ *then* α.*inwindow* := *start*;
20. /* Drop out old events from the window: */
21. *for* all events (A, t) in s such that t = *start* - 1 *do*
22. *for* each $\alpha \in$ *contains*(A, A.*count*) *do*
23. *if* α.*event count* = $|\alpha|$ *then*
24. α.*freq_count* := α. *freq_count* - α.*inwindow* +*start*;
25. α.*event_count* := α.*event_count* – A.*count*;

```
        A.count := A.count - 1;
/* Output: */
for all episodes α in C do
        if α.freq_count=(Te - Ts + win – 1) ≥ min_fr then output α;
```

Algorithm 5

Input: *A* collection C of serial episodes, an event sequence **s** = (s, T_s , T_e), a window width *win*, and a frequency threshold *min_fr*

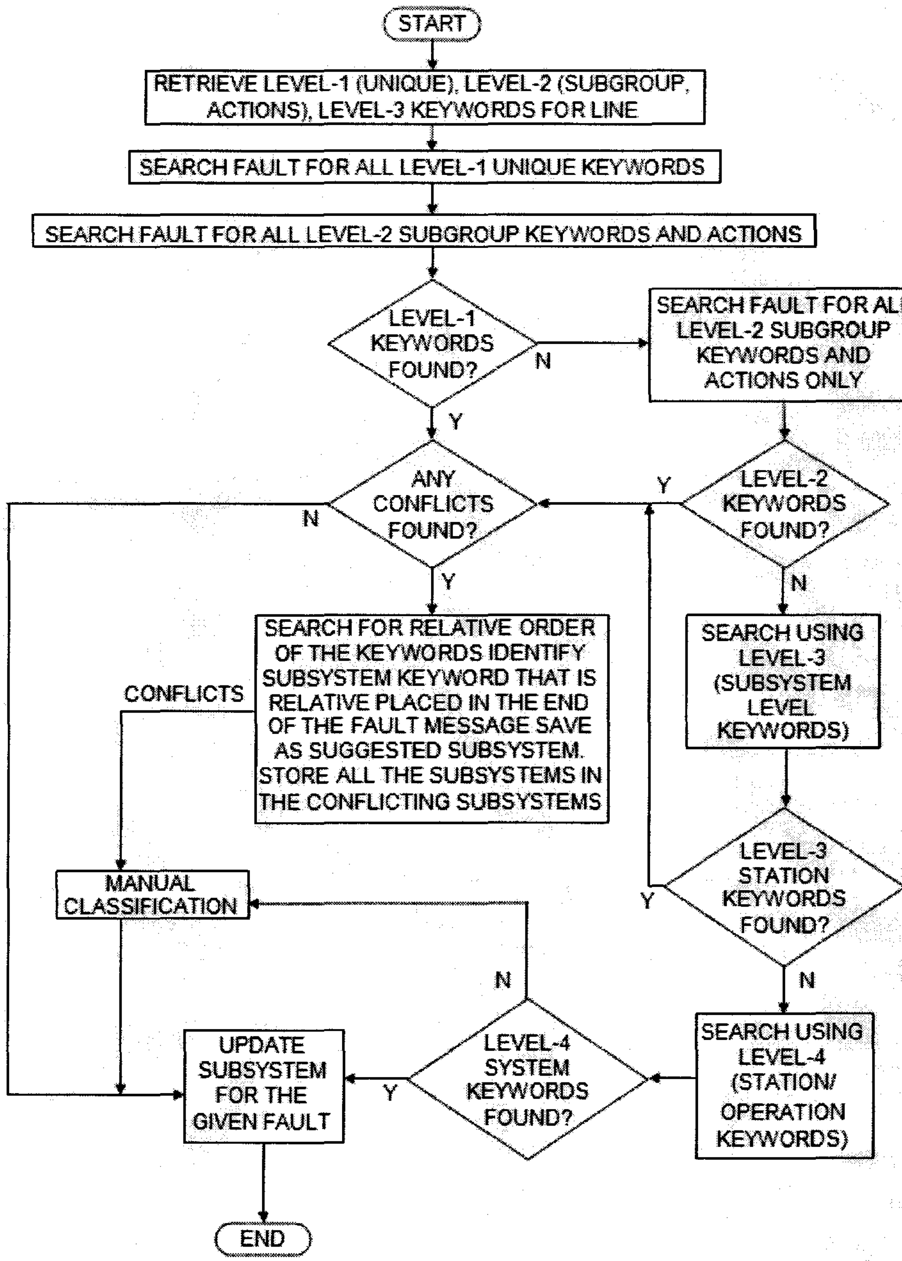
Output: The episodes of C that are frequent in **s** with respect to *win* and *min_fr*

Method:

```
=* Initialization: *=
for each α in C do
        for i := 1 to |α| do
        α.initialized[i] := 0;
        waits( α [i ]) := 0;
for each α ∈ C do
        waits( α [1]) := waits( α [1]) ∪ {( α, 1)};
        α(freq_count := 0;
for t := Ts-win to Ts -1 do beginsat(t) := 0;
/* Recognition: */
for start := Ts-win +1 to Te do
        /* Bring in new events to the window: */
        beginsat(start + win- 1) := 0;
        transitions := 0;
        for all events (A, t) in s such that t = start + win- 1 do
                for all (α,j ) ∈ waits(A) do
                        if j = |α| and α.initialized[ j ] = 0 then α.inwindow := start;
                        if j = 1 then
                                transitions := transitions ∪ {( α, 1, start + win- 1)};
                                else
                                transitions := transitions ∪ {( α, j, α.initialized[ j- 1])};
                                beginsat( α.initialized[ j- 1]) :=
                                        ( beginsat( α.initialized[ j- 1]) \ {( α, j- 1)};
                                α.initialized[ j- 1] := 0;
                                waits(A) := waits(A) \ {( α, j )};
                for all ( α, j, t) ∈ transitions do
                        α.initialized[ j] := t;
                        beginsat(t) := beginsat(t) ∪ {( α, j )};
                        if j < |α| then waits( α [ j + 1]) := waits( α [ j + 1]) ∪ {( α, j + 1)};
                /* Drop out old events from the window: */
                for all ( α, l) ∈ beginsat(start- 1) do
                        if l = |α| then α. freq_count := α. Freq_count- α.inwindow + start;
                        else waits( α [l + 1]) := waits( α [l + 1]) \ {( α, l + 1)};
                        α.initialized[l] := 0;
/* Output: */
for all episodes α in C do
if α. Freq_count=(Te -Ts + win- 1) ≥ min_fr then output α;
```

APPENDIX 2

The decision tree is used for classification of operational failures to the component level. This algorithm is part of US Patent 20060288260



APPENDIX 3

The maximum likelihood estimate of the probability a_{ij} of a particular transition between states *i* and *j* can be estimated by counting the number of times the transition was taken, which could be called $C(i \rightarrow j)$, and then normalizing by the total count of all of the times it took any transition from the state *i*:

$$a_{ij} = \frac{C(i \rightarrow j)}{\sum_{q \in Q} C(i \rightarrow j)} \qquad \text{(A3-1)}$$

The backward probability β is very important to illustrate the algorithm and can be defined as the probability of seeing the observations from time t+1 to the end, given that process is in state j at time t and given λ:

$$\beta_t(i) = P(o_{t+1}, o_{t+2} \ldots o_T \,|\, q_t = i, \lambda) \qquad \text{(A3-2)}$$

It is computed in a similar manner to the forward algorithm with the following steps:

1. **Initialization:**

$$\beta_T(i) = a_{i,F},\ 1 \le i \le N \qquad \text{(A3-3)}$$

2. **Recursion** (since states 0 and q_F are non-emitting):

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad, 1 \le i \le N, 1 \le t < T \qquad \text{(A3-4)}$$

3. **Termination:**

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1)\beta_1(j) \quad \text{(A3-4)}$$

The following section shows how the forward and backward probabilities are used to compute the transition probability a_{ij} and observation probability $b_i(o_t)$ from an observation sequence, even though the actual path taken through the machine is hidden.

From equation (5-15)

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \quad \text{(A3-5)}$$

At this point the probability ξ_t can be defined as the probability of being in state i at time t and state j at time $t+1$, given the observation sequence and the model:

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda) \quad \text{(A3-6)}$$

In order to compute ξ_t, the "not-quite-ξ_t" probability should be computed. This value is similar to ξ_t, but differs by including the probability of the observation in the computation.

$$\text{not-quite-}\xi_t(i,j) = P(q_t = i, q_{t+1} = j, O | \lambda) \quad \text{(A3-7)}$$

Based on the β function from the recursion step

$$\text{not-quite-}\xi_t(i,j) = \alpha t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad \text{(A3-8)}$$

In order to compute ξ_t from *not-quite-*ξ_t, based on the laws of probability the formula should be divided by $P(O|\lambda)$, since:

$$P(X|Y,Z) == \frac{P(X,Y|Z)}{P(Y|Z)}$$

The probability of the observation given the model is the forward probability of the whole term, which can be computed as:

$$P(O|\lambda) = \alpha_T(N) = \beta_T(1) = \sum_{j=1}^{N} \alpha_t(j)\beta_t(j) \tag{A3-9}$$

The final equation for ξ_t is:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_t(j)}{\alpha_T(N)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_t(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_t(j)} \tag{A3-10}$$

The expected number of transitions from state i to state j is then the sum over all t of ξ. To estimate $\hat{a}_{ij}$ in equation (A3-5), the total expected number of transitions from state i is needed, which can be obtained by summing over all transitions out of state i with the following formula

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i.j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N}\xi_t(i,j)} \qquad \text{A3-11)}$$

A formula for re-computing the observation probability is also needed. This is the probability of a given symbol v_k from the observation symbol V, given a state j:

$$\hat{b}_j(v_k) = \frac{\text{expected number of transitions from state } i \text{ and observation symbol } v_k}{\text{expected number of times in state } j}$$

(A3-12)

The probability of being in state j at time t, is called $\gamma_t\,(j)$ and can be defined as:

$$\gamma_t\,(j) = P(q_t = j|O,\lambda) \tag{A3-13}$$

Equation (A3-13) will be computed by including the observation sequence in the probability:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \tag{A3-14}$$

The numerator is just the product of the forward probability and the backward probability

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} = \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)} \tag{A3-15}$$

At this stage, b could be computed. For the numerator, the summation γ_t (j) for all time steps t is performed in which the observation o_t is the symbol v_k of interest. For the denominator, the summation of γ_t (j) over all time steps t is done. The result will be the percentage of time that the process was at state j and symbol v_k was produced.

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{A3-11}$$

Where the notation $\sum_{t=1 s.t. O_t = v_k}^{T}$ means "sum over all t for which the observation at time t was v_k"

The formulas in equations (A3-11) & (A3-11) can be used to re-estimate the transition A and observation B probabilities from an observation sequence O assuming having a previous estimate of A and B. These re-estimations form the core of the iterative forward-backward algorithm.