CHAPTER 3

LITERATURE REVIEW

3.1 INTRODUCTION

The literature review assesses the past and current status of research work in the area of software based mechatronics system. There is relatively little information published on the software based mechatronics systems and its analysis. The work done by the earlier researchers in this area are classified in the following aspects.

i. Design of software based mechatronics system

ii. Analysis of software based systems

iii. FMEA and Bayesian model

iv. Software quality and reliability

v. Component based software development

vi. Quality tools and techniques Vs Software quality

vii. Distributed software system and its reliability

viii. Optimal design and Taguchi methods

ix. Virtual instrumentation
3.2 DESIGN OF SOFTWARE BASED MECHATRONICS SYSTEM

The term “Mechatronics” was first introduced by Yasakawa electric company in 1969. (Kyura and Oho 1996) Yasakawa defined mechatronics in this way: The word, Mechatronics, is composed of “Mecha” from mechanism and the “tronics” from electronics. In the past two decades the rapid development of the computer, and then the microcomputer, embedded computers, and associated information technologies and software advances, made mechatronics an imperative in the later part of the twentieth century (Onwuvulu 2006). The International Federation for the Theory of Machines and Mechanisms (IFTMM) provides a good definition of mechatronics: Mechatronics is the synergistic combination of precision mechanical engineering, electronic control, and systems thinking in design of products and manufacturing processes. The first journal on mechatronics was released by IEEE in the year 1996. Then many leading publishers started to publish the findings in mechatronics and its related technologies for the past one decade.

Auslander (1996) defined clearly about mechatronics and its system elements in his paper of “What is Mechatronics?” in the first issue of IEEE transactions on mechatronics. He focused mainly about the real time software and its criticalities in mechatronics systems. He discussed in detail about software properties, computational performance and the software portability in the software based real time systems.

In the same issue of IEEE, Kyura and Oho (1996) also discussed the fundamentals of mechatronics equipment and the issues in designing of mechanisms, mechanism controls, advanced machine control considerations and also prospects of mechatronics products for different applications.
Rzevski (2003) has proposed the conceptual design of mechatronics systems based on multi-agent technology. He used the digital information and communication technologies to design an intelligent mechatronics network. He considered distributed systems are capable of achieving considerably better results in terms of performance/cost ratio and reliability than conventional centralized large systems and structures in dynamic environment. He also suggested the different types of intelligent mechatronics networks in the system design with the effective utilization of latest communication technologies.

Barron and Powers (1996) described the role of electronics controls for future automotive mechatronics systems in their paper. They considered the integrated circuit technology, power electronics technology and software for automotives in the dynamic conditions. These parameters are considered as response affecting parameters in the Taguchi’s parameter optimization in the chapter 7.

3.3 ANALYSIS OF SOFTWARE BASED SYSTEMS

The integration of heterogeneous components in mechatronics systems requires broadening the concepts and cooperation between the technical disciplines involved to develop a common conception of the product and should come up with optimized solution. If the system quality is poor, the total system performance will result in serious injury to the system or significant property damage. So, associated component quality measurement, analysis and improvements are an essential issue in the construction of new generation mechatronics systems.

Ramamoorthy et al (1979) described a systematic approach to the development and validation of critical software for nuclear power plant. This
methodology was explained based on two software program developed from the same specification by two independent teams. This was a new initiative in the time period of second generation computers.

Gomaa (1984) suggested an effective software design method for real time systems. He developed a design method based on data flow, task communication, data abstraction, time and functional requirements.

Leveson (1986) explained the importance of software safety and its need in controlling real time, safety critical processes. This survey attempts to explain why there is a problem, what the problem is and what is known about and how to solve it.

Feret et al (1990) developed architecture for real time diagnostic systems. The paper focuses the measurement and inspection of real time systems. But, the paper is limited to measurement of real time systems only.

Zanoni and Pavan (1993) assessed the reliability and improvement of automotive electronic components by considering various parameters. They addressed the reliability problems of automotive electronic components due to thermo mechanical stress and thermal fatigue, electrical over stress, electrostatic discharge, electromagnetic interference, and latch-up in CMOS (complex metal oxide semiconductors) integrated circuits and electro migration. They described the failsafe operation of automotive electronic components in this paper. This approach considers only the problems in the electronics side.

Son and Seong (1995) explained the qualitative evaluation of safety critical software at the early stage of development by taking an example of interposing logic system software. They analyzed the modules involved in the
safety critical systems based on entropy concept. They also suggested some design changes and re-evaluated the enhanced design.

Littlewood and Strigini (2000) pointed out the research challenges of reliability, complexity and quality assurance in software based systems.

Gargantini and Morzenti (2001) advocated the real time systems analysis based on its involved components. They insisted the total reliability of the system can be identified based on its components and sub components study. This paper concept is very useful in developing our case study.

Barone et al (2006) paper aims at setting up of a statistical tool for improving the reliability of the on board diagnostic system for automotives. This paper concept is used in many ways in this thesis and also in the case studies.

Peters and Parnas (2002) developed the monitoring system for real time systems. This is done using a monitor, a system that observes the behavior of a target system and reports if that behavior is consistent for different requirements

Wattanapongshakorn and Levitan (2004) presented four models for optimizing the reliability of embedded systems considering both software and hardware reliability under cost constraints. The models can be used to maximize the reliability without increasing the budget.

3.4 FMEA AND BAYESIAN MODEL

Failure Mode and Effects Analysis (FMEA) is one of the well-known analysis methods having an established position in the traditional
reliability analysis. Here, the purpose of FMEA is to identify possible failure modes of the system components, evaluate their influences on system behavior and propose proper counter measures to suppress these effects. Bayesian models provide therefore a useful and attractive method of constructing reliability models, which are based on the Bayesian inference. The theory of Bayesian statistics is well established and the method has been applied in various areas including software, automation systems, medical diagnosis and geological explorations.

Kmenta and Ishii (2004) criticized the reliability analysis using failure mode and effects analysis and proposed an advanced method related to FMEA named as Scenario based FMEA. They introduced failure scenarios in conjunction with expected cost with some assumptions. However, coupling failure scenarios improves the clarity of failure representation and risk evaluation. This paper introduced a new scenario based FMEA to find reliability. The paper is limited to mechanical component failure alone.

Vollmer (2003), a BMW system safety engineer discussed the FMEA analysis in active chassis control system which is mainly focused the sensor, controller area networking, microcontroller and actuator. He summarized that FMEA is a useful tool to ensure the safety and availability of the system and to reduce the risk hazards.

Campodonica and Sinngpurwalla (1994) proposed a Bayesian approach for predicting the number of failures in a piece of software, along with logarithmic poisson research model. In this paper, the experts’ opinion and software testing failures are combined by using a Non Homogeneous Poisson Process model (NHPP) based on the method of maximum likelihood estimation.
Percy (2002) presented a general review and new ideas for improving decisions by adopting Bayesian methodology to allow for the uncertainty of model parameters. The paper contributes to the theory of decision analysis in maintenance and replacement where stochastic model can be developed to describe the processes involved.

Pentti et al (2004) explained the essential issues in the construction of automation system. This Finland’s nuclear power plant research report considered the Bayesian inference to assess automation reliability applications. The report’s aim is to provide a support for the authorities and to give adequate confidence in the licensing of computer based automation systems based on quantitative reliability assessment.

Paasch and Durgi (2003) described the use of behavioral model and multistage decision making model in Bayesian network for representing the troubleshooting process for electromechanical system. This paper says that a Bayesian network method can be applied to diagnose a complex electromechanical system. The discussed approximate method is not as accurate as decision analysis due to the assumption of conditional independence.

3.5 SOFTWARE QUALITY AND RELIABILITY

Generally, software quality is conformance to requirements because if the software contains too many functional defects, the basic requirement of providing the desired function is not met. To increase the software quality and preventing software errors, the focus must be on comprehensive requirements and software testing. Software testing is the basic requirement for software reliability engineering (Whittaker 2000). Software reliability engineering is integrally connected and is in fact a keystone to total quality management. It provides a user oriented metric that is correlated with customer satisfaction. It
cannot determine reliability of software based systems without its associated software reliability measure (Musa 2005). Software reliability engineering is based on a solid body of theory (Musa et al 1987) that includes operational profiles, random process software reliability models, statistical estimation, and sequential sampling theory. Software personnel have practiced software reliability engineering extensively over a period dating back to 1973 (Musa and Iannio 1991). But still it is under research because of new technologies emerging in software based systems.

Haynes and Thompson (1980) discussed the statistical methods used in the specification, test and analysis of the reliability of complex systems. This method is based on statistical inference, using a Bayes interpretation. This approach is restricted to the reliability of software only.

Ravishankar and Velardi (1985) demonstrated a methodology for simulating the interaction between hardware and software as it relates to system reliability. The impact of hardware and software errors on the system is evaluated by measuring the effectiveness of system recovery in containing the propagation of hardware related software errors. This approach is used effectively by the authors to identify the hardware related software errors.

Musa et al (1987) too presented exponential distribution for reliability estimation model for assessing the reliability of individual components based on its field failure intensity.

Osterweil et al (1996) concluded in his paper in the summary topic in this way: “The quality of software products is clearly unacceptably poor. A large and challenging program of long-term research is needed to support developing the technologies needed”.
Rooijmans and Aerts from Philips Electronics and Genuchten from S&P Consulting and Blenks Groupware (1996) discussed about software quality in consumer electronics products. He analyzed the software quality on the basis of software development process, software architecture, requirement engineering, design engineering, project evaluations, inspection and testing and cost benefit analysis. They proposed a proper approach to improve the quality of software in electronic products. This approach is partly considered in this research and proposed to extend for mechatronics product.

Xie et al (2006) analyzed the fault detection and fault correction processes. In their paper, some useful approaches to the modeling of both software fault-detection and fault-correction processes are discussed.

Wong (2006) identified the explicit inputs and examined the effect of explicit inputs on software review performance in practice.

Actually, the literature is jammed to overflowing software reliability models, so the practitioner should be aware of selecting models for a particular application, which may be quite suited to the situation. The two most important criteria for selecting a software reliability model and estimation method for use in tracking reliability growth are model simplicity and maturity. The model which is used to predict the reliability should be in simple format which should be able to understand by the software engineers without extensive mathematical background.

3.6 COMPONENT BASED SOFTWARE DEVELOPMENT

Today the component -based software products are becoming popular in real time systems. It is very important to reduce the development time of software based real time systems to update the implementation of
latest technologies (Liu 2001). It is most important to develop the real time software with high quality as early as possible to construct such type of real time systems. So, it is motivated to utilize the Commercial Off-The-Shelf (COTS) software components for rapid development in the field of software development. Component based software engineering is a process that emphasizes the design and construction of computer based systems using software components. Component based software applications are expected to have high reliability as a result of deploying trusted components.

Voas (1998) presented the summary of the advantages that can be gained by developing a system using COTS components.

i. Functionality is instantly accessible to the developer

   ii. Components may be less costly than those developed in-house.

   iii. The component vendor may be an expert in the particular area of the component functionalities.

Yacoub et al (2004) introduced the reliability model and a reliability analysis and technique for component based software. The technique is proposed for a scenario based reliability analysis. In this method, they considered the various parameters to estimate total software component reliability based on software component, interfacing reliability and link reliability.

After the selection of right components, the assembly of the component is to be processed. Correct automatic assembly of software components is considered an important issue of Component Based Software Engineering (CBSE). Inverardi et al (2003) has proposed an approach to the
integration problem by assuming a well defined architecture style to detect the integration anomalies.

Nunn et al (2002) described a component architecture which is defined by an extensible mark-up language that specifies the composition of components into a persistent, suitable, composite application.

Bucchiarone et al (2006) proposed a software architecture based approach in which architectural analysis and code synthesis are combined together in order to efficiently and correctly assemble a system. However, an effective methodology is not yet proposed and it is needed in software assembly.

3.7 QUALITY TOOLS AND TECHNIQUES VS SOFTWARE QUALITY

To control and improve the quality, the first important step is to measure the quality based on inspection or testing. The tools and techniques used for quality improvement in manufacturing industries can be effectively used in software applications also (Parnas and Lawford 2003). Statistical quality control is an effective system for controlling the process parameters by comparing it with standards and take corrective action if there is any deviation. These tools and techniques are applied to improve the product quality by improving processes.

Nakajo and kume (1991) used cause and effect relationship between human error, program fault, and process flaw and design method. This is presented based on case-based analysis. This is an excellent paper to analyze the software errors using cause and effect technique. But it is restricted to cause and effect relationships between different factors.
Weller (2000) discussed the practical application of statistical process control techniques in software. He analyzed that the possibilities of statistical process control in software.

Card (2000) claimed that six sigma provides a generic quantitative approach for improvement that applies to any software process. He stated that six sigma offers another path toward measurable improvement for Capability Maturity Model (CMM) level organizations. He also suggested that six sigma tool kit provides the basic technology necessary for continuous improvement beyond the CMM levels.

Henry et al (2003) discussed the simulation testing of real-time software provides great advantages in cost savings, early error detecting and requirements clarification prior to development. This paper describes a method for developing simulation tests, evaluating simulation test results and utilizing the test evaluating to implement improvements in simulation testing.

Parnas and Lawford (2003) emphasized the importance of the role of inspection in software quality assurance. This paper concept is used in the software testing and inspection part in this research.

Pan and Xia (2006) discussed several issues in quality management and control from the views of the new sciences and proposed the concept of computer aided statistical automatic process control.

Oakland and Tanner (2006) stressed the change management importance in the quality management for any organizations. He concluded that managing change and reliability of software system according to its functional requirement is more important in quality management.
Liu et al (2006) described the importance of knowledge management for software product. They also examined the variation in software development by taking 67 software products.

Jalote and Saxena (2002) discussed the use of statistical process control and control charts to software quality control processes. They analyzed the setting of control limits for software processes.

This literature review shows the importance of software quality control in real time products and processes in the development stage and also the need of a good quality management technique in an easy understandable format.

3.8 DISTRIBUTED SOFTWARE SYSTEM

The introduction of software, digital technology and communication engineering in real time systems has many advantages both concerning flexibility and reliability. Recent technology of networking allows designing the distributed software system consisting of a number of software sub systems at various hierarchy levels which are physically interlinked to provide a solution for any software application. In such type of systems, software and hardware reliability has a continuously increasing impact on system reliability (Kant 1990). So, reliability is an essential issue in the construction of new generation distributed software systems. But most of the literatures are concentrating only on the individual component reliability.

Mahalik (2003) presented the importance of distributed control system and networking in mechatronics systems. In his book of mechatronics, he discussed and compared the distributed system with centralized control system and hybrid control system.
Zhao et al (2001) presented an efficient on line mode estimation algorithm for a class of sensor–rich, distributed embedded systems. In their paper, they have given an algorithm for diagnosis of distributed embedded systems.

3.9 OPTIMAL DESIGN AND TAGUCHI METHODS

Parameter design is an off-line quality engineering technique, where the parameters for a process or product are determined, so that the effect of noise on the response of the process or the function of the product is minimized. So, the method, called Taguchi method, helps to improve quality without increasing the cost, through experimentation.

Kanchana and Sarma (1999) applied the Taguchi method to optimize the software design process to enhance the software quality. They considered number of requirements per module, cyclomatic complexity and coupling as the model factors and errors in software as response. This paper is used in the parameter design of software based mechatronics system in the proposed case study in chapter 7.

Xydas et al (2005) used the Taguchi method for the optimal design of barcode scanner, and conduct the experimental design using the L_{18} orthogonal array. Though the barcode scanner consists of software and hardware components, they considered the control factors related to only hardware.

Lian Ha et al used Taguchi method for optimal design of micro-positioning of Scott-Russel (SR) mechanism. (SR mechanism is a simple structure with main feature of displacement amplification and straight line motion which is used in positioning stage and precision cutting). In this case
study, they have taken four control factors and three levels to find the micro positioning.


3.10 VIRTUAL INSTRUMENTATION

A virtual instrumentation system is a computer based instrumentation system that a user would employ to develop a computerized test and measurement system, for controlling from a computer desktop an external measurement hardware device, and for displaying test or measurement data collected by the external device on instrument-like panels on a computer screen. Virtual instrumentation extends also to computerized systems for controlling processes based on data collected and processed by a computerized instrumentation system. A large variety of data collection instruments designed specifically for computerized control and operation were developed and made available on the commercial market, creating the field now called "virtual instrumentation."

Goldberg (2000) discussed the virtual instrumentation in his paper of “What is Virtual Instrumentation?”. In this paper, he clearly pointed out the difference between the instrumentation and virtual instrumentation.

Moschioni (2003) applied the virtual Instrumentation based measurement technique to measure the tallest Medieval Bell tower in Europe. In this paper, a virtual instrumentation based data acquisition system is presented.
Robinson and Dhandapani (2005) applied the virtual instrumentation to develop an electronic fuel injection system in a two-stroke SI engine. They found some experimental investigations on fuel injection using the technology of virtual instrumentation.

So, the virtual instrumentation offers a best solution at a flexible cost, compact setup, and control over the software based system design. M/s. National Instruments of Texas, USA is providing software and hardware to implement virtual instrumentation effectively and as a result of this thesis, many software based systems developed using virtual instrumentation with M/s. National Instruments hardware and software.

3.11 CONCLUSION

Based on the detailed literature review carried out to identify the performance analysis and improvement of software based mechatronics systems, it can be inferred that although some papers touched the analysis of software based real time systems, no one integrates in the following areas.

i. Study of behavior of software based mechatronics system.

ii. Analysis of the reliability of software based mechatronics system.

iii. Identification of failure modes in software based mechatronics system and also the causes.

iv. Analysis of the impact of software in mechatronics system.

v. Study of interaction between software and hardware and also the critical factors in interfacing.

vi. Analysis of distributed software systems and its detailed reliability analysis.
vii. Optimal design factors to achieve the optimal response of software based programmable system using Taguchi method considering both software and hardware control factors.

viii. Development of sophisticated software based mechatronics system using virtual instrumentation.

Hence, it has been decided to carry out the research focusing all of these aspects in software based mechatronics system applying different tools and techniques based on the literature review.