Nelson Mandela
Metropolitan
University

for tomorrow

WIRELESS ROTATIONAL PROCESS MONITORING SYSTEM

By

Morné Odendaal

Submitted in complete fulfillment of the degree:

MAGISTER TECHNOLOGIAE:
ENGINEERING: ELECTRICAL

In the

Department Electrical Engineering
School of Engineering
At the
Nelson Mandela Metropolitan University

April 2011

Promoters:
Prof. Dr T.I. van Niekerk
Dr G. H. Kruger

DECLARATION

I, Morne Odendaal, hereby declare that:

- the work done in the thesis is my own;
- all sources used or referred to have been documented and recognized; and
- this thesis has not previously been submitted in full or partial fulfillment of the requirements for an equivalent or higher qualification at any other educational institution.

Signature: **Date:**

ACKNOWLEDGEMENTS

I would like to acknowledge and sincerely thank individuals and institutions for their contributions:

- Prof. Theo Van Niekerk and Dr. Grant Kruger for their tireless support, guidance and encouragement;
- The National Research Foundation and the Research Committee of Nelson Mandela Metropolitan University for its financial assistance;
- Sincere thanks to my mentor, Frank Adlam, for all his support, motivation and patience during the development of this project;
- My appreciation and thanks to Prof. Shih and the University of Michigan, for giving me the opportunity to visit their laboratories, and to work on some of their research projects;
- My appreciation also to the University of Michigan, for allowing me to use their manufacturing equipment and machines;
- Thanks to all my South African friends and also my newly-formed friends from America, for their support, encouragement and love;
- My deepest thanks to my Family, for everything they have done, and the love they have shown;
- Lastly, I would like to thank the Lord, Jesus Christ and the Holy Spirit; for giving me strength, knowledge, and will power to complete this research to the best of my ability.

ABSTRACT

The manufacturing industry is constantly looking for ways to reduce production costs and at the same time to increase productivity. Automation of common manufacturing operations is one of these methods. By automating common manufacturing operations; various machines, robots, control systems and information technologies are used to reduce the overall human input requirement (mental and physical). Recent advances in technology have made it possible to now also automate (or facilitate) the maintenance requirement of these machines and tools.

Modern tools and machines, which can estimate when it will fail or when failure is imminent have obvious advantages for predictive maintenance purposes. Another function of this technology is to determine how efficiently a tool or machine operates, or what the quality of the produced goods is. Predictive maintenance can decrease manufacturing plant or machine down times – which have a positive effect on cost-savings – has gained considerable importance over the last two decades.

TABLE OF CONTENTS

DISCUSSION, CURRENT LIMITATIONS, FUTURE DEVELOPMENTS AND CONCLUSION

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The demand in the manufacturing industry to reduce production costs as well as to increase productivity has led manufacturers to automate many operations. Recent technological advances have led to the solution of yet another important manufacturing requirement: tools and machines that “know” what their working condition is and therefore when maintenance would be required. The application process of devices which monitor whether a tool or machine is developing a fault (or when a fault is imminent) is called Condition Monitoring. This technology forms a major tool for predictive maintenance and has gained considerable importance over the last two decades as, and according to Brophy, B., Kelly, K., & Byrne, G. (2002), it significantly influences the process economy and the machined part quality.

Owing to today’s rapidly changing manufacturing environment and in order to meet the needs of an ever more demanding consumer market, manufacturers worldwide have begun an international collaborative research programme in manufacturing – known as the Intelligent Manufacturing Systems (IMS) program. One of the major projects of IMS is called the Holonic Manufacturing Systems (HMSs). A holon may be described as an identifiable part of a system that has a unique identity (a sub-system of a system) but is made up of smaller, inferior parts (which are part of the system) that continuously and independently perform their task. One of the primary advantages of HMSs is that it enables manufacturers to create highly complex systems which are robust (even to disturbances) but which have the ability to easily adapt to the manufacturing environment.

The technologies mentioned above namely Condition Monitoring and HMSs, could be used to develop a self-functioning monitoring device thus improving predictive maintenance. The School of Engineering at Nelson Mandela Metropolitan University (NMMU), has implemented this technology in the form of a Wireless Rotational Process Monitoring System (WRPMS), on a Friction Stir Welding (FSW) spindle.

Bluetooth was selected as the wireless technology because of its low power consumption and low cost. Additionally, it provides a secure wireless path between a huge variety of devices such as mobile phones, personal computers, laptops etc. The use of Bluetooth for wireless transmission of data means that an operator will be able to access tool- or process-data using a range of Bluetooth-enabled devices. For example an operator is able to use a Bluetooth-enabled cellular telephone or laptop to gain access to data. Bluetooth is a wire-replacement protocol, that provides built-in security at short ranges (1m, 10m or 100m depending on the device class), at various data rates (1 Mbit/s, 3 Mbit/s or 24 Mbit/s depending on the Bluetooth protocol version). The power consumption of a Bluetooth device is directly related to its communication range (this means that the further the Bluetooth device can transmit, the higher its power consumption will be) and is also defined by the class of the Bluetooth device. A Class 1 Bluetooth device consumes 100mW and can attain a transmission range of approximately 100m. A Class

3Bluetooth device consumes 1mW and can attain a transmission range of approximately 1m.

1.1 PROBLEM STATEMENT

Rotational processes form a major part of today's manufacturing industry. Processes like milling, drilling, routering and even the recent uptake of FSW; are all examples of such processes. Failure to successfully monitor these processes could result in catastrophic breakages and material waste.

1.2 HYPOTHESIS

To design and implement a Wireless Rotational Process Monitoring System (WRPMS) which will be able to communicate with a wide range of Bluetooth-enabled devices (like cell phones and notebooks). The WRPMS device should be able to monitor process variables (like force, torque and temperature) for a wide range of machines like FSW machines or even milling machines.

The WRPMS will be able to connect to another Bluetooth-enabled device, in order to send monitoring data or receive control data using the Bluetooth protocol.

1.3 MAIN AIM

- To design and implement a Wireless Rotational Process Monitoring System combined with signal processing, and to analyse the information obtained using existing software packages.
- To develop and complete a functional experimental setup, which must be able to function in the industrial environment
- To look at the different sub-components within the system in detail, and by looking at different design approaches, select the most feasible options for the project
- To determine and understand the different physical and electrical design issues, in order to create and manufacture the best possible solution.

1.4 OBJECTIVES

- Implement a simple Bluetooth communications model that sends monitoring data and receives control data wirelessly to and from a remote PC.
- The designed system will perform its task as part of a bigger project, consisting of the following different sub-components:
    i. Communications – different approaches to technology and software design will be considered, different features of the Bluetooth standard will be evaluated.

ii. Signal Conditioning – different types of signal conditioning will be considered: amplification/attenuation, isolation, multiplexing, filtering, excitation and cold-junction compensation

iii. Power Supply – different types of supplies will be considered like capacitive, inductive as well as battery-operated.

iv. Processor – will be required for house-keeping functions (different processors will be evaluated).

v. PC – will be used to analyze the information obtained (different software packages will be considered).

1.5 DELIMITATION OF PROJECT

It is hoped, within a limited period, to implement, develop and test a wireless monitoring device not only on an FSW machine - at the NMMU north campus – but also on similar machines, like SFW and milling machines. Processing data, received from the monitoring device on to a graph or spreadsheet, will not form part of the project. A small personal computer (p.c.) application could be created to process the data instead.

1.6 OUTLINE OF THE SYSTEM DEVELOPED

The system allows sensor data from the rotating experimental setup to be collected and stored on a PC. A wireless Bluetooth communication link is used as the communication path between the Microcontroller and the PC. See the figure below.

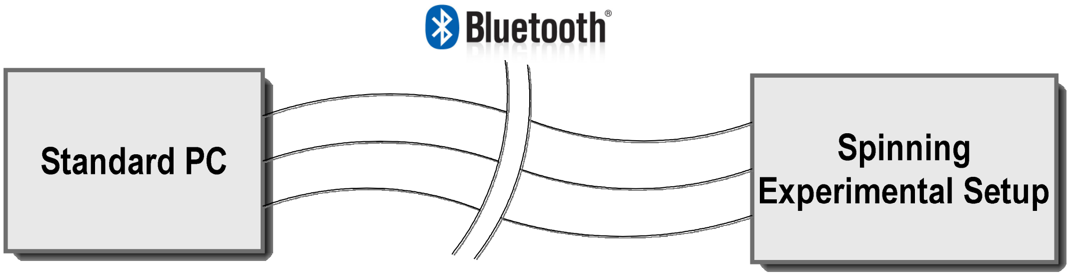


Figure 1: The proposed system – the WRPMS

The WRPMS is divided into two parts:

1. A standard PC:
    a. A standard PC uses Microsoft Windows XP (32-bit) operating system;
    b. Includes a USB Bluetooth v2.0 + EDR dongle which is connected to the PC;
    c. Includes a GUI, which was specifically designed using LabVIEW, controls the Bluetooth dongle, handles the wireless transmissions and creates a data log file.
2. An embedded Microcontroller system (Spinning Experimental Setup):
    a. Contains a type-K temperature sensor provides the actual analogue temperature measured at the tip of the spindle;
    b. Contains a reconfigurable Field Programmable Analogue Array (FPAA) sensor-board which provides filtering and/or amplification of the sensor's analogue value;

c. Includes the Microcontroller that reads and processes the filtered and/or amplified sensor value and transmits the value (via the Bluetooth Radio Module);
d. And a Bluetooth Radio Module which provides communication for the Spinning Experimental Setup and allows data to be sent to the PC.

The system is designed using standard, easily obtainable hardware and software packages.

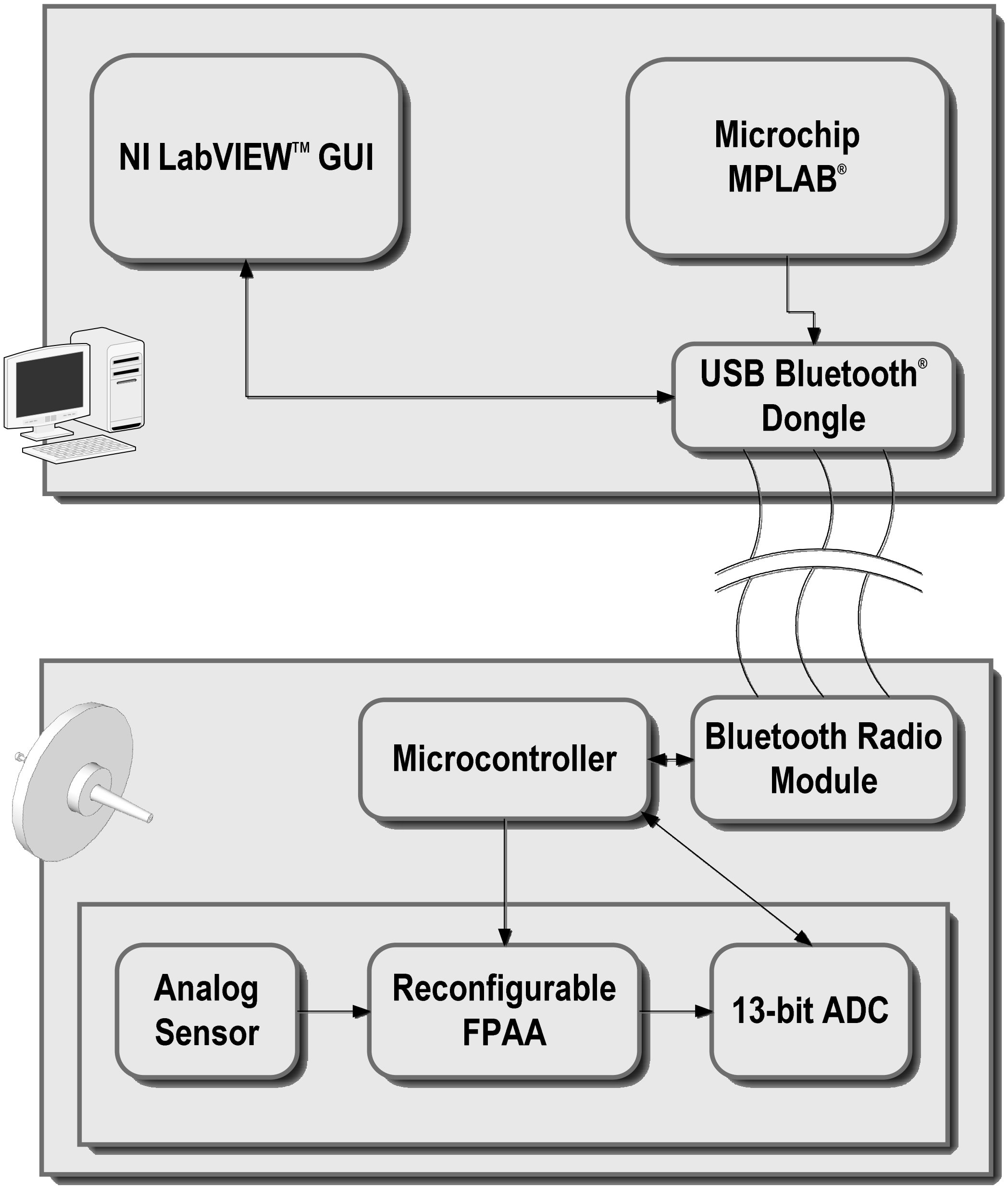


Figure 2: Block Diagram of the WRPMS system

1.7 SIGNIFICANCE OF THE PROJECT

1.7.1 Within the Nelson Mandela Metropolitan University (NMMU)

The research carried out in this investigation will be shared with the University of Michigan. Any additional knowledge obtained with respect to wireless data transfer should facilitate the joint research and collaboration which has taken place between the universities with respect to the field of intelligent machining and machine monitoring. At the NMMU, the efficiency of research activities in this field would also improve significantly

1.7.2 For International Research

This research is intended to benefit the international manufacturing industry by developing an industrial wireless monitoring system which can be used to gather data from a rotating spindle (or chuck). The data collected can then be used to develop systems which perform maintenance (predictive maintenance) wirelessly.

1.8 CONCLUSION

A WRPMS was designed in order to create a wireless link between a sensor and another device (standard PC) via Bluetooth. Being completely wireless and mobile, allows the system to be used for various monitoring tasks for example:

- spot FSW processes (see Chapter 2)
- machine-tool spindle monitoring (see Chapter 3)
- closed-loop control applications (see Chapter 4)

CHAPTER 2

LITERATURE OVERVIEW – SPOT FSW PROCESS

2.1 OVERVIEW

Friction Stir Spot Welding (FSSW) is a new solid-state joining process. This process has the advantage that the material doesn't melt during the welding process. Rather, friction is used to generate enough heat for the material to enter a plastic state. According to Hovanski, Santellab and Grant (2007, p. 1) FSSW is "similar in concept and appearance to its predecessor, resistance spot welding (RSW) ... ".Currently FSSW remains the focus of extensive international research, especially from the automotive industry. FSSW has already proved to be a cost-effective and productive means for joining lightweight structural alloys. It is also presently the preferred method for spot joining of aluminium alloys.

"FSSW avoids the severe heating and cooling cycles induced during the resistance method" (Hovanski et al., 2007, p. 1).

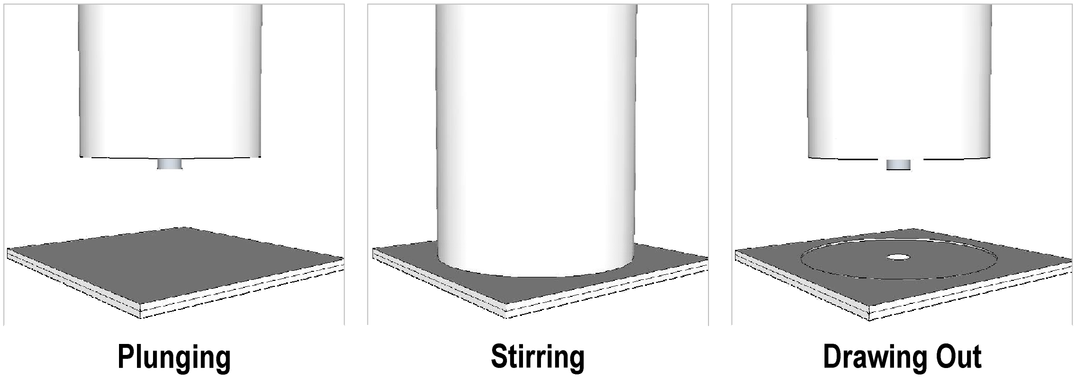


Figure 3: Visual schematic of the three-step friction stir spot welding (Adapted from Hovanski, Santellab & Grant, 2007, p2)

FSSW makes use of a rotating tool which consists of a shoulder and a pin (sometimes threaded), to produce spot welds. The pin has a much smaller diameter than the shoulder, and is the first to make contact with the material to be welded.

FSSW is an extension of Friction Stir Welding (FSW). FSW is performed when the non-consumable rotating tool plunges into the metal and after a short dwell period “starts moving along the butting surfaces of two rigidly clamped plates placed on a backing plate.”(Nandan, DebRoy & Bhadeshia, 2008, p. 981)The two plates to be welded are “clamped in a manner that prevents the faying surfaces from separating.”(Blignault, Hattingh, Kruger, van Niekerk & James, 2006, p. 2)Owing to vertical force applied to the tool, heat is generated by friction, mostly caused between the tool shoulder and the metal. This friction causes “localised heating” (Su, Nelson & Sterling, 2005, p. 1), which softens (plasticizes) the material around the pin. When the metal is in this form – after the dwell period – the tool is moved (whilst still rotating) along the joint line, mixing and forging the softened material of the two clamped metal plates behind the trailing face of the pin. Thus the weld is produced whilst the two metals are in a ‘solid state’. (Mishra & Ma, 2005, p. 2)

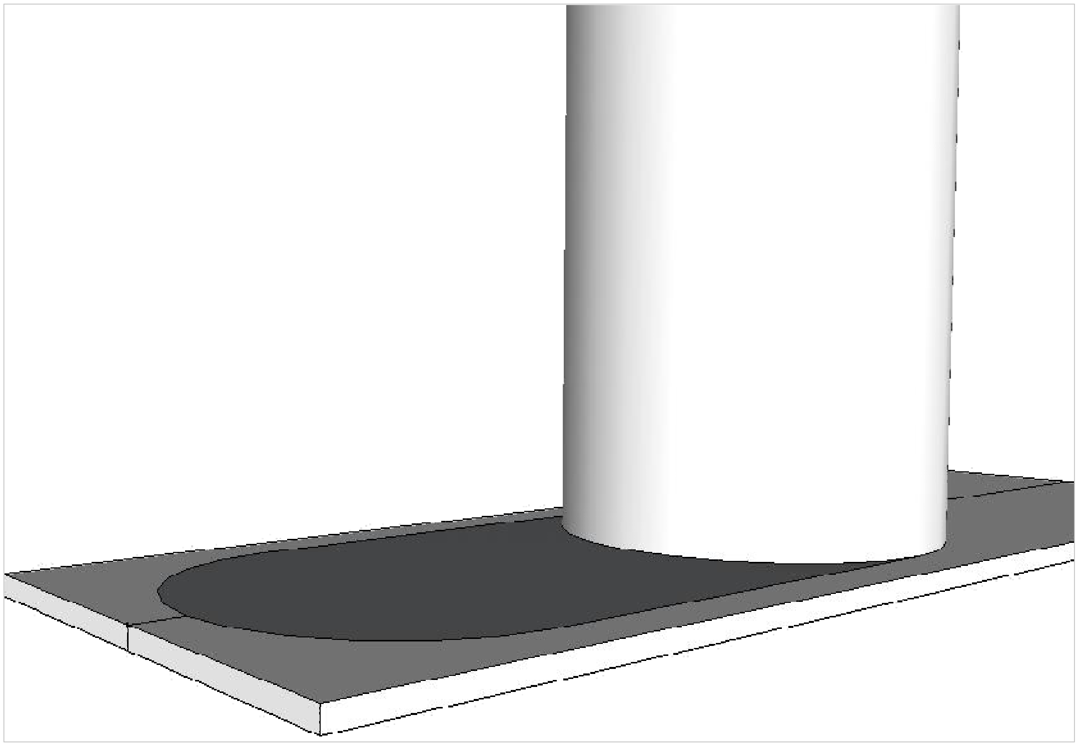


Figure 4: Schematic diagram of friction stir welding technique (Adapted from Xu, Liu, Luan & Dong, 2008, p2)

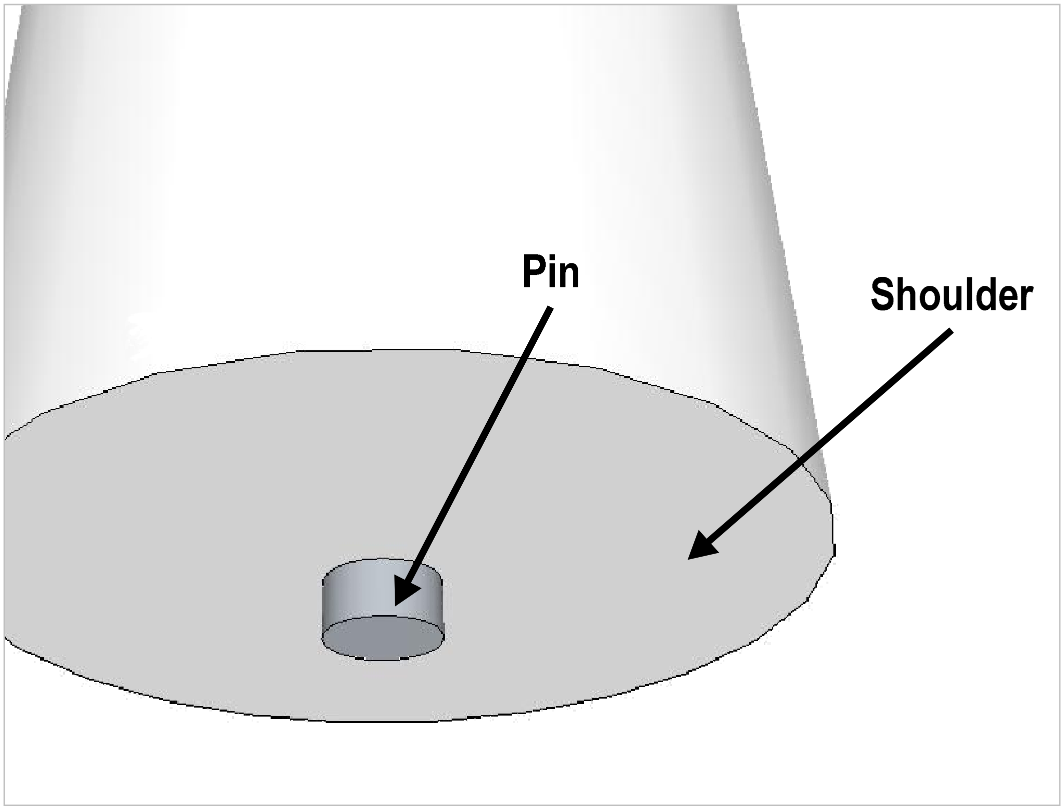


Figure 5: Representation of an actual tool. The pin penetrates the material whilst the shoulder creates friction and therefore generates heat.

FSW is used when the metals being joined together should maintain their original metal characteristics as far as possible. Primarily, FSW is used on aluminium when the welded pieces are too large to be effectively heat-treated after the weld in order to recover temper characteristics. FSW was invented by Wayne Thomas and his colleagues at The Welding Institute (United Kingdom) in December 1991.

Since its conception, friction stir welding has rapidly “grown from a laboratory curiosity to an established manufacturing process for aluminium alloys within a decade of its invention” (Threadgill & Nunn, 2003, p. 4).This considerable growth can be attributed to the advantages of the process as well as to the quality of the joints which it can produce. Threadgill and Nunn (2003) also mention that one of the important abilities of friction stir welding during its inception “was the ability to weld the high strength 2xxx and 7xxx alloys”. This ability sparked a great deal of interest in the research communities as these alloys are generally difficult to weld using fusion processes.

Mitlin, Radmilovic, Pan, Chen, Feng and Santella (2005) give the following differences when comparing FSW with FSSW. They indicate that FSW is usually used for joining “thick-butted (edge-to-edge)” specimens; whilst FSSW is “performed on lap (back-to-back) specimens” which can be “between 1-2 mm thick”. Additionally they highlight the difference in the flow of the material during FSW and FSSW. During FSW the pin penetrates 80% of the entire thickness of the welded specimens and “moves transversely”, causing the flow of the material to move from the front of the tool to the back. During FSSW, however, the pin penetrates “50-70% of the total thickness”, and because the tool remains stationary, “the process requires three-dimensional axisymmetric material flow” to ensure joint integrity.

2.2 ADVANTAGES

Lombard, Hattingh, Steuwer and James (2007, p. 1)mention that FSWoffers such benefits such as relatively low defect populations (when compared to fusion welding) as well as “the ability to join dissimilar metals”, creating joints with “high fatigue strength”, and “low preparation and little post-weld dressing”.

Mathon, Klosek, de Carlan and Forest (2009) explain that using FSW to weld Oxide Dispersion Strengthened (ODS) alloys, provides the advantage of avoiding “the oxides dissolution but induces severe microstructure evolution.”

Metals are joined without fusion or filler materials.

Kim, Fujii, Tsumura, Komazaki and Nakata (2006, p. 1) pointed out that “aluminum die casting alloys are widely used in the automotive, electronics, machine and building industries” due to their lightness, recyclability, excellent wear and fatigue properties. It is difficult to use fusion welding for aluminum die casting alloys due to the forming of welding defects (blowholes and deformation), which in turn results in decreased mechanical properties. Owing to the popularity of aluminum alloys, and the inability to create sound joints using fusion welding, FSW has become the preferred method to perform these welds.

FSW creates joints with better mechanical properties than joints created by conventional fusion methods.

Fratini, Buffaa and Shivpuri (2007, p. 1) points out that, when FSW is used to weld normally difficult-to-weld materials, it can provide advantages because the welded material never reaches melting temperatures. The reason provided by Xu et al., (2008, p1) for these advantages is that – in contrast to fusion welding – FSW results in “a much lower distortion and residual stresses owing to the low heat input characteristic of the process”. They also mention the advantage of not requiring filler material.

Mishra and Ma (2005, p. 2) refers to FSW as “the most significant development in metal joining in a decade” as well as it being a “green” technology. This is attributed to friction stir welds being considerably energy-efficient, versatile and environmentally friendly. No cover gas, flux, filler metal is required. It is also mentioned by Mishra and Ma (2005, p. 2) that any aluminum alloy (similar, dissimilar, and even composites) “can be joined without [any] concern for the compatibility of composition, which is [normally] an issue in fusion welding”.

2.3 USES OF FSW

Owing to its advantages and the extensive amount of research already performed to enhance its weld quality, FSW has gradually emerged as a new solid-state joining technique and is currently being applied in an increasing number of joining applications worldwide.

According to Arul, Kruger, Miller, Pan and Shih (2008, p. 1) FSW has been used for over a decade in the aerospace industry to join aluminium alloys. Since its conception it has “attracted significant interest from the aerospace and transportation industries and an extensive literature exists on FSW" (Lombard, Hattingh, Steuwer & James, 2007, p. 341).

FSW is especially used to weld high-strength Al-alloys which are difficult to weld using conventional fusion techniques (Su, Nelson & Sterling, 2005, p. 1).

Mathon, Klosek, de Carlan and Forest (2009) performed research on using FSW to weld materials reinforced by oxide dispersion (referred to as ODS – Oxide Dispersion Strengthened). These materials can be used in a vast number of applications because of their excellent mechanical resistance at medium to high temperatures. Using FSW on these materials provides a possibility of preserving the oxides dispersion in the metal matrix.

Moreira, Figueiredo and Castro (2007, p. 169) compares fusion welding with FSW. Conventionally, fusion welding was used to weld aluminium alloys. An example of this fusion welding process is Metal Inherent Gas (MIG) welding. This type of welding process uses a gas shield to protect the arc and weld from atmospheric contamination. Some of these hydrogen gasses are entrapped in the weld, causing a weld prone to defects such as porosity. Furthermore, "an electric potential is established between the electrode and the work piece causing a current flow, which generates thermal energy in the partially ionized inert gas". FSW is much more versatile, environmental-friendly and conserves energy. Additionally, defect-free welds are obtained.

Nandan, DebRoy and Bhadeshia (2008, p. 980) report that FSW has become the primary aluminium welding technique. The use of FSW to join difficult-to-weld metals (as well as metals other than aluminium) is slowly growing. Widespread benefits have been obtained from using FSW in different industries (such as aerospace, shipbuilding, automotive and railway industries).

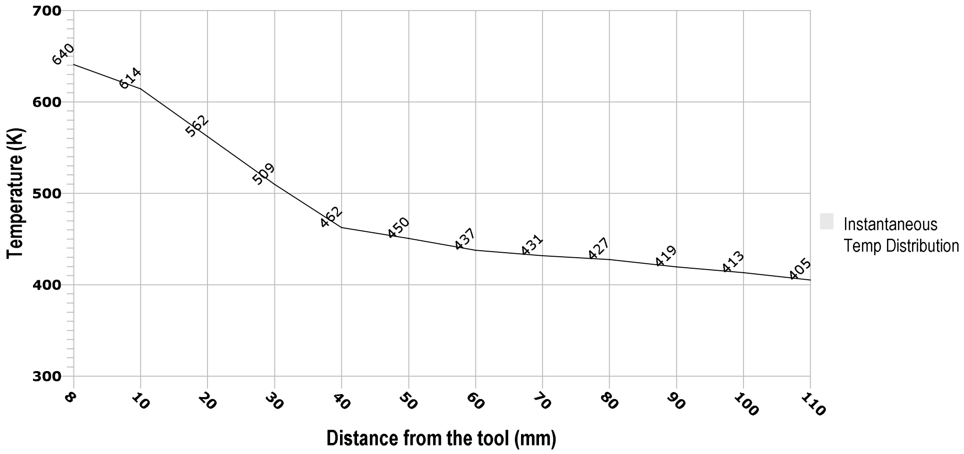


Figure 6: Instantaneous temperature distribution, measured using neutron-diffraction, on the top surface along the weld centerline behind the tool pin, for FSW of AA 6061-T6. (Adapted from Nandan, DebRoy & Bhadeshia (2008, p. 993))

FSW is considered the most significant development in metal joining in a decade (Mishra & Ma, 2005, p. 1). A new development deriving from FSW is friction stir processing (FSP). This process is used “for localized microsctructural modification for specific property enhancement.”

Mishra and Ma, (2005) claim that FSW provides an opportunity for enabling the wide use of the highly alloyed 2XXX and 7XXX series of aluminium alloys.

These alloys are generally classified as being non-weldable. FSW can alter the traditional approach for producing these lightweight assemblies.

2.4 PREVIOUS RESEARCH

Lombard, Hattingh, Steuwer and James (2007, p. 352) research shows that rotational speed is the key parameter governing tool torque, temperature, frictional power and hence tensile strength and fatigue performance. In order to explain and predict weld performance using basic weld process parameters, correlations between the two energy parameters (heat input and frictional power) and the tensile strength and fatigue life of the welded specimens need to be found. They proved that lower values of frictional power input provide higher tensile strengths. In order to ensure adequate plasticization of the alloy during the welding process, a sufficiently high temperature is required. If this temperature can be controlled it could lead to a lower power requirement during the process.

Research carried out by Lombard et al. (2007, p. 354), has shown the importance "… to understand and model the heat and power input during friction stir welding …" and to be able to "… predict the likely tensile and fatigue performance …" based on "…particular combinations of tool feed rate and rotational speed." They have also shown that there "… is a strong correlation between frictional power input, tensile strength and low cycle fatigue life." Rotational speed controls the generation of defects within the weld.

"Future investigations of FSW using instrumented welding tools will allow a much less empirical approach to be developed to making welds with optimised sets of process parameters." (Lombard et al., 2007, p. 354)

Sua, Nelson and Sterling (2005, p. 277) describes the deformation of the material during FSW/FSP as "intense plastic deformation at elevated temperatures". This causes a very fine (1–10um) grain structure "in the center of the weld/processed region."They claim that "Understanding the evolution of this microstructure is of significant technological and scientific interest." Furthermore, Sua et al., (2005, p. 285) suggest that the "final microstructures of processed material" are strongly dependent on "tool design, processing parameters and cooling rate."

Nandan, DebRoy and Bhadeshia (2008) performed research on heat generation, heat transfer, plastic flow during welding, the elements of tool design, understanding defect formation and the structure and properties of the welded materials. They claim that "a problem in the calculations of heat generation is that the friction coefficient cannot be determined from fundamental principles." (Nandan et al., 2008, p. 983)

They also observed that "the heat generation rate is influenced by the rotational speed but not by welding speed". Furthermore, they maintain that the welding parameters and material properties “affect the temperature profiles, cooling rates, microstructure and the resulting properties of the welded joint.” (Nandan et al., 2008, p. 993)

When compared with conventional fusion welding processes the peak temperatures attained in FSW are significantly lower. This is due to the material entering a plastic form instead of melting, when being welded with FSW. In order to obtain temperature information thermocouples and “*in situ neutron-diffraction*” (lattice distortion is related to change in temperature) can be used. If experimental data are not available, Nandan et al., (2008, p. 993) suggests that computational models can be used after “the model has been properly validated with experiments.”

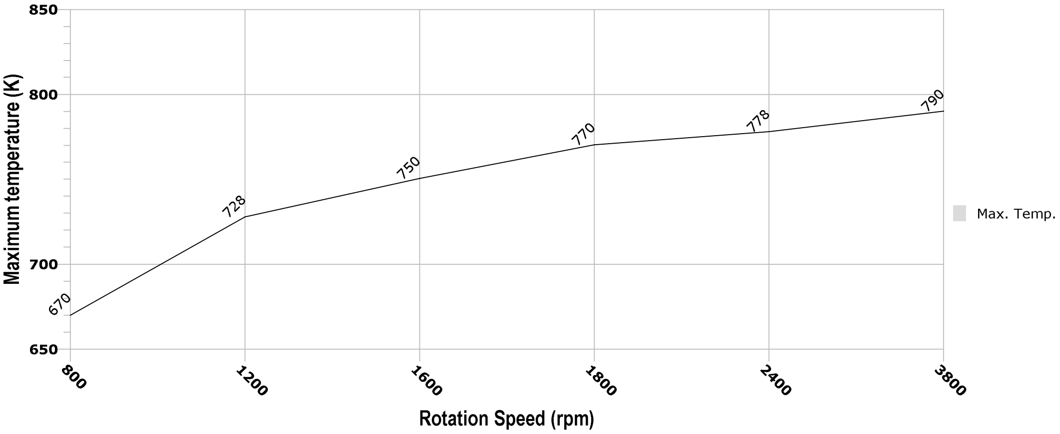


Figure 7: Relationship between rotational speed and peak temperature in FS-welds of AA 6063 [110]. (Adapted from Nandan et al. (2008, p997))

It has been highlighted by previous research (Fratini, Buffaa & Shivpuri, 2007, p. 210) that the tensile properties of joints created by FSW are strongly affected by the operating parameters. In order to produce effective joints, both the feed rate and the rotating speed should be chosen carefully. Tools to help with this choice are needed.

Xu et al. (2008, p. 2) used thermocouples (K-type with an outer diameter of 1mm) to record the “temperature evolutions within the weld area”. This was done by embedding the thermocouples “in the regions adjacent to the rotating pin”. Temperature transducers were used to magnify the small signal produced by the thermocouples (into an acceptable voltage range of 1~5Vdc). They used an industrial control computer to record and process the data. The problem of measuring the temperature in the immediately region of the rotating pin was identified. This is due to the severe plastic deformation which occurs here. Small blind holes (diameter 1.2mm) were drilled into the cross section of the workpiece to avoid the destruction of the thermocouples. Xu et al.’s research (2008) has shown that the peak temperature in the FSW zone increased by 40–70ºC when increasing the tool’s rotary speed from 300 rpm to 400 rpm, and increased by 80–100 ºC when the tool’s rotary speed was increased from 300 rpm to 600 rpm keeping the tool traverse speed constant. It was also shown by Xu et al. (2008, p. 3) that the peak temperature on the advancing side is 15–25ºC higher when compared to the temperature of the retreating side. In addition Xu et al. (2008, p. 4) found that when the traverse speed was increased (whilst keeping the rotary speed constant), the peak temperatures dropped. The temperature rise rate seems unaffected.

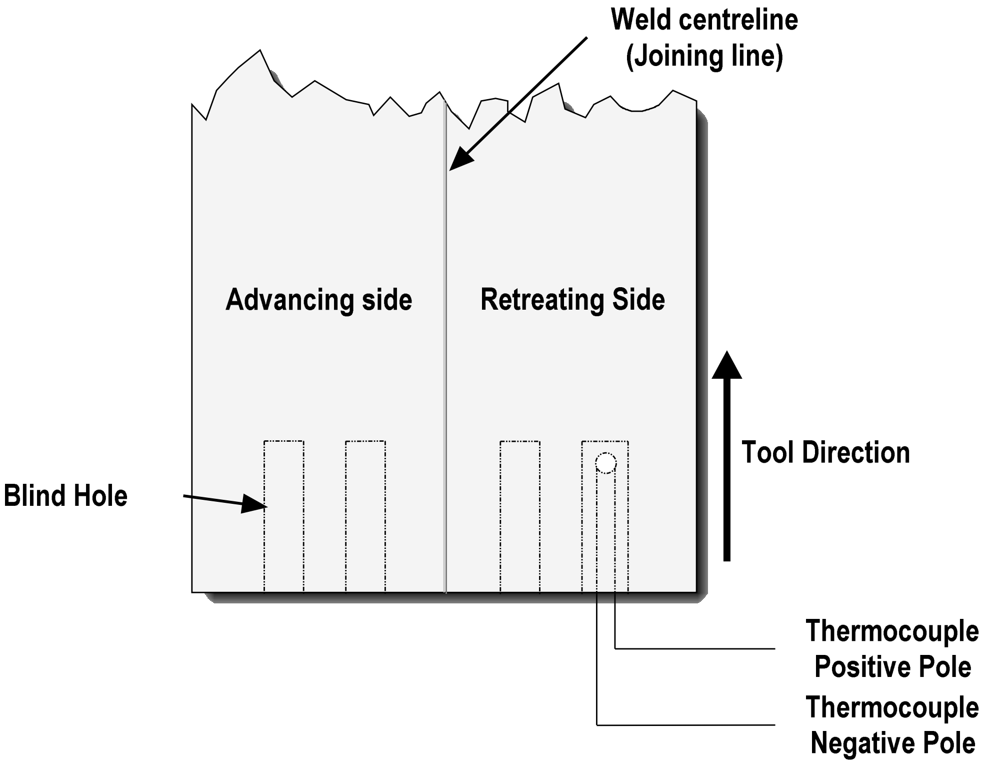


Figure 8: Arrangement of blind holes for measuring temperatures during tool inserting. (Adapted from Xu et al., 2008, p2)

Thus, Xu et al. (2008) has shown that FSW parameters (rotary speed and traverse speed) considerably affect the temperature. The joint's temperature increases with the increase in rotary speed which can be attributed to the fact that the growing friction input of the tool. With an increase in the traverse speed however; a lower heat input results. Thus, by increasing the rotary speed, more frictional heat and greater mechanical stir is produced by the tool. Higher temperature causes the material to become more plastic. Xu et al. (2008, p. 7) maintains that this increase in rotary speed, which results in a higher temperature and an increase in the plasticity of the weld material, can improve the weld quality. Increasing the rotary speed causes larger plastic deformations, enhancing the strain-strengthening effect and also improving the joint's strength.

Mishra and Ma (2005, p. 17) reports that previous research has shown that "the maximum temperature was observed to be a strong function of rotation rate(ω, rpm) while the rate of heating was a strong function of traverse speed. It was also noted that there was a slightly higher temperature on the advancing side of the joint..."

Mishra and Ma (2005, p. 14) have also affirmed that "temperature measurements within the stirred zone are very difficult [to obtain] due to the intense plastic deformation produced by the rotation and translation of tool." They also remark that maximum temperatures within the FSW stirred zone, have previously either been estimated or recorded using embedded thermocouples "in the regions adjacent to the rotating pin."

Mishra and Ma (2005, p. 18) concluded that there are many factors which influence the thermal profiles that occur during FSW. They have concluded that:

1. The maximum temperature rise within the weld zone is below the melting point of aluminium (the aluminium is plasticized);
2. The tool shoulder is the greatest contributor to heat generation during FSW;
3. The maximum temperature:

    a. Increases when the tool rotation rate increases (at a constant tool traverse speed);
    b. Decreases when the traverse speed is increased at a constant tool rotation rate;
    c. Increases when increasing the ratio of tool rotation rate/traverse speed.
4. The maximum temperature rise occurs at the top surface of weld zone.

Experimental verification of various theoretical or empirical models (for pseudo-heat index of the joint) proved difficult (Mishra & Ma, 2005, p. 18).

Mishra and Ma (2005, p. 20) reported that different FSP parameters have different effects on the microstructure and properties of cast A356. Results indicated that, using the same tool geometry, a lower tool rotation rate (300-500 rpm) resulted in the generation of basin-shaped nugget zones, whilst higher rotation rates (>700 rpm) produced elliptical nugget zones.

K-type thermocouples (glass insulated) were also used by Mandal, Rice and Elmustafa (2008, p. 415) for temperature measurements. In their experiments, the thermocouples were positioned 2mm from the pin and 6.5mm from the top of the welding surface. The thermocouples were placed inside 2mm diameter holes and in similar positions at the leading and trailing side. A modified milling machine was used to enable load measurement capabilities. Their setup included a miniature button-type load cell (positioned to always be in contact with the tool which was placed inside the machine arbor) and connected to a rotating electrical connector (REC) at the top of the machine head (operated similar to a slip ring). The forces and temperature data were measured using a Data Acquisition system (DAQ), equipped with LabView.

In their research Mandal et al., (2008, p. 5) investigated the plunge stage using experimental and numerical modelling. Their machine could only be operated manually. This made it difficult to plunge with a high plunge speed and therefore the plunge speed in their simulation was slowed down to match the plunge speed of the experiment.

It has been reported by Zhang and Zhang (2008, p. 241) that the microstructure and the mechanical properties of the nugget zone can be affected by FSP. They also referred to other FSW research which showed that the quality of the friction stir weld is determined by the different welding parameters (the welding speed, the rotating speed, the ratio of the welding speed and the rotating speed and the axial pressure on the shoulder). Zhang and Zhang suggested that in order to improve FSW, it is necessary to accurately control these process parameters. Zhang and Zhang (2008, p. 241) investigated the "mechanism of controlling of welding parameters" using numerical methods.

Blignault et al., (2006, p. 1) makes the point that a scientific understanding of the FSW process is necessary in order to successfully automate the process. Owing to the complexity of the interaction between the rotating tool and the alloy, it is difficult to determine which parameters will provide the desired weld properties. In their research they have constructed a rotating multi-axial transducer. Online and offline measurement is possible using this system and these measurements can be incorporated into the feedback control system.

2.5 NEED FOR DEVELOPMENT OF TEMPERATURE MONITORING

Lombard, Hattingh, Steuwer and James(2007, p. 341) report that, to date, there are few reported systematic studies of process optimisation in terms of the relationships among the various process parameters such as: tool rotational speed and feed rate, forces on the tool, defect population, residual stresses, mechanical properties and fatigue performance. Furthermore, Lombard et al. (2007) shows that the application of instrumented tools, that provide data on the heat (and the amount of heat energy) and power input into the weld, is still a new idea. Lombard et al. (2007, p. 342) alleges that the previous norm to investigate and to perform FSW was to establish the welding parameters empirically and for individual cases. An obvious advantage therefore is “to be able to reduce this empiricism” and to clearly “identify optimum process parameter combinations” in order to obtain the required mechanical properties and “dynamic” performance of FSW (Lombard et al., 2007, p. 342).

Reynolds (2008, p. 338) holds that the “flow of material around the welding tool during friction stir welding (FSW) is closely linked to many of the key issues related to the process.” He adds that “strain and strain rate experienced by a material element are directly related to (in fact are determined by) the material's flow history" (Reynolds, 2008, p. 338). Reynolds highlights an important point that "FSW is a fully coupled thermomechanical process". The temperature history is closely related to the flow history.

Additionally Reynolds, (2008) makes the point that further investigations into material flow and its associated effects are needed. Earlier observations and studies made use of snapshots of the flow field at various times during the process. Perhaps real time monitoring of the temperature can provide further insight?

The understanding of both the welding process, and the structure and properties of the welded joints of FSW has proved useful in reducing defects and improving uniformity of FSW welds. This has also facilitated the use of FSW for other more diverse applications. Nandan, DebRoy and Bhadeshia (2008, p. 1016) claim that better and more "efficient tools have been devised" with our improved "quantitative understanding of the underlying principles of heat transfer, material flow, tool-work-piece contact conditions and effects of various process parameters [of FSW]". This is important for the expansion of the applicability of FSW for other purposes.

Reynolds (2008, p. 338) draws attention to the value of using simulations that could predict the effects of tool geometry and weld parameters on material flow and weld quality.

Buffa, Donati, Fratini and Tomesani (2006, p. 344)show that, in order to get an effective joint, the proper conditions of pressure, temperature, strain and strain rate must be obtained. These conditions depend on several parameters:

- Tool geometry;
- The force superimposed on the tool;
- The tool sinking;
- The rotating speed and the feed rate.

Reynolds (2008, p.342) stresses the importance of fully coupled, "transient simulation[s]" which can capture and potentially" help to explain features which appear to be fundamental to FSW". Reynolds concludes that "accurate strain and temperature histories for the material being friction stir welded" should be available and comprehended. Only then, Reynolds insists, can "microstructure prediction[s]", or "intelligent decisions" be made with regards to "how to modify processing to achieve [the required] microstructure".

Blignault et al. (2006) discussed how to develop a rotating multi-axial transducer to measure various load inputs, torque and temperatures acting on the rotating tool during FSW. Blignault et al. (2006, p. 32) suggested that the variation of the process parameters (forces, torque and temperature used to perform the weld) could provide "valuable insights" into the mechanisms of FSW. These "insights" could then be used for process optimization should they be "correlated with weld quality (microstructure, hardness, etc.)".

Blignault et al. (2006, p. 32) hopes that these acquired “insight” would be used to interpret various aspects of FSW, and to “reduce the empiricism” with which FSW parameters are normally selected.

Nandan, DebRoy and Bhadeshia, (2008, p.982) points out that “a unique feature of the friction stir welding process is that the transport of heat is aided by the plastic flow of the substrate close to the rotating tool. The heat and mass transfer depend on material properties as well as the welding variables [process parameters].” It was suggested that the “effects of heating and cooling must also be considered [together with the plastic flow, its dynamic recrystallisation and the behaviour of metal at high strain rates].”

Nandan et al. (2008, p. 983) writes that “errors in the calculation of heat generation rates” could arise from various sources. One such a source is the value of the friction coefficient. Nandan et al. (2008, p. 983) remarks that there is “no straightforward method to estimate the coefficient of friction or to predict how it changes with temperature or relative velocity.” Another factor to consider is that when heating results from plastic deformation, there will be “a reduction in the heat generation rate” due to “the decrease in yield strength” when the material starts to plasticize (the yield strength reduces to zero in the event of the material melting). Nandan et al. (2008) also reports that “most of the heat generated due to plasticity occurs close to the tool-workpiece interface.” Having a temperature sensor in the stir zone would have obvious advantages.

Nandan et al.(2008, p. 995) observed that the “peak temperatures in the work-piece are attained close to the edge of the tool shoulder.” They claim that obtaining temperature measurements close to a rotating tool is difficult. The following reasons are given by Nandan et al. (2008, p. 995):

1. “The material transport caused by the motion [stirring] of the tool makes it difficult to focus on a single location.” Embedded thermocouples “may become displaced due to the plastic flow.”
2. “The strong temperature gradient near the tool means that a small error in thermocouple location can lead to a large error in temperature.”

In their research Nandan et al., (2008, p. 995) show that “thermal cycles affect the structure and properties of welded materials.” As an example precipitation-strengthened aluminium alloys was considered. With these alloys “the hardness profile depends greatly on the precipitate distribution and only slightly on the grain-size. If a peak temperature greater than 675K is attained at any location, complete dissolution and re-precipitation” will occur in that location. Nandan et al. (2008, p. 996) report that this “may lead to softening”, and that this occurrence starts to happen “even when the peak temperature is lower than 675 K”, because “the density of the strengthening precipitate reduces” as the temperature draws nearer to the 675 K value. Nandan et al. (2008, p. 1001) also state that “most of the heat generation occurs at the interface between the tool shoulder and the work-piece.”

Nandan et al. (2008, p. 1016) have determined several “uncertain input parameters” for FSW:

- The Friction coefficient;
- The extent of slip between the tool and the work-piece;
- The heat transfer coefficient at several work-piece surfaces;
- Delivery of the heat between the work-piece and the tool at the tool-workpiece interface,
- The computed values of non-Newtonian viscosity based on the available constitutive models.

Nandan et al. (2008, p. 1017) highlight an important difficulty of existing process models. They claim that these models "are unidirectional", taking as input various "welding parameters, thermophysical properties, tool and work-piece geometry", and for the output giving "the temperature and velocity fields and cooling rates at various locations." It is suggested, by them, that engineers should rather "be able to specify the cooling rates, the geometry of the stir zone and/or other attributes of FSW as input", and then obtain "several alternative sets of welding parameters [combinations of welding speeds, rotational speeds, tool dimensions and other variables]."Nandan et al. (2008, p. 1017) concludes that FSW process models should have a "reliability component built into them" and they need to be "bi-directional in nature" in order to become more widely used and/or accepted.

Fratini, Buffa and Shivpuri, (2007, p. 213) observed that, in order to get good results when performing FSW, it is important that each piece's thermal profile "should be as similar as possible", thereby ensuring a similar recrystallisation and aging of the HAZ.

Xu, et al. (2008, p. 1) argue that temperature is an “important influencing factor to the quality of the joints”, as it affects “the formation and flow” of the plasticized metal.

Mishra and Ma (2005, p. 6) claim that the two most important FSW parameters are: “tool rotation rate (ω, rpm) … and tool traverse rate (v, mm/min) along the line of joint”. They also assert that “higher tool rotation rates generate higher temperature because of higher friction heating” and this results in “more intense stirring and mixing of material”. Mishra and Ma (2005, p. 7) also insist that the effects of “preheating and cooling are also important” but only for “specific FSW processes”. Preheating is normally used when materials have a high melting point (such as steel of titanium) and thus facilitates the plasticization. Cooling is sometimes used when the material has a lower melting point (such as aluminium and magnesium) “to reduce extensive growth of recrystallized grains and dissolution of strengthening precipitates in and around the stirred zone”. In order to optimize the microstructure of the weld, Mishra and Ma (2005, p. 51) suggest that it the “process temperatures and cooling rates” should be controlled.

Mishra and Ma (2005, p. 14) mention that “temperature measurements within the stirred zone are very difficult [to obtain] due to the intense plastic deformation produced by the rotation and translation of [the] tool”. They maintain that “the maximum temperatures within the stirred zone during FSW have either been estimated from the microstructure of the weld" or by using "embedded thermocouples in the regions adjacent to the rotating pin".

Mishra and Ma (2005, p. 15) reports that previous research show that the temperature increase rate is lower at higher tool rotation rates. They add that heat generation is dominated by the shoulder rather than the pin of the FSW tool. The following reasons for this are provided by Mishra and Ma (2005, p. 16):

- The contact area of the shoulder is bigger than that of the pin;
- The vertical pressure between the shoulder and the workpiece is much larger than those between the pin and the workpiece;
- The linear velocity of the shoulder is much higher than that of the pin, because the shoulder has a larger radius.

Zhang and Zhang (2008, p. 269) showed that the maximum temperature increases with an increase of tool rotation speed. Zhang and Zhang (2008, p. 262) also determined that the temperature distribution becomes more uniform with the increase of rotation speed. They also found that increasing the rotation speed can "improve the friction stir weld quality" but it also increases the occurrence of weld flash, "which can have a negative effect on FSW weld quality."Zhang and Zhang (2008, p. 269) suggest that "the simultaneous increase" of the tool's rotation and translating speed (to prevent the occurrence of weld flash)could lead to an increase of the weld's residual stress.

Arul, Kruger, Miller, Pan and Shih (2008, p. 2) highlights the advantages of refill friction stir spot welding. This variation of the SFW process creates joints which "have higher shear strength compared to fixed probe-pin SFW [Spot Friction Welding] joints."

Lathabai, Painter, Cantin and Tyagi (2006, p. 899) mention that "the four main FSJ [Friction Spot Joining] parameters are":

- Tool rotational speed;
- Plunge rate;
- Plunge depth, and
- The dwell period (the time after the tool has plunged into the material, during which the tool just rotates in that position).

Furthermore, Lathabai et al. (2006, p. 1) write that even though "FSJ is already used in industrial production, information on how these parameters affect weld quality is limited."

Buffa, Fratini and Piacentini (2008, p. 310) performed research on Friction Stir Spot Welding (FSSW). After the initial plunging stage, the tool was fed "along a prescribed path close to its initial print." By using this method a "wider spot weld" and a larger stir zone is obtained. Buffa et al. (2008, p. 4) performed tests using different spindle speeds. Ultimately they decided that a relatively low speed of 300rpm provided optimum results. These results differed from FSW of butt joints where the optimum spindle speed was higher than 300 rpm. Buffa et al. (2008, p. 316) explain that "the proper range of rotational velocities for the modified FSSW process is lower than the optimal one for FSW of butt joints” and they attribute this to “the absence of the tilt angle” within their experimental setup as the tool was kept vertical during its traversing along the predefined patterns.

Buffa et al. (2008, p. 310) writes that “the applied load [after the tool pin has penetrated the material]” together with “the chosen tool rotating speed” determine the “friction forces” and consequently “the generated heat flux.” It stands to argue that controlling the rotational speed would therefore allow the control of the frictional heat input.

CHAPTER 3

LITERATURE OVERVIEW – MACHINE, TOOL AND SPINDLEMONITORING

3.1 OVERVIEW AND BACKGROUND

Rauber, Barata and Steiger-Garcao (1994, p. 1) declare that the main objective of monitoring and prognostics "is to ensure the [continuous delivery of] quality of manufactured goods" by detecting any malfunctions, and to eliminate them as early as possible.

Monitoring can also enhance the efficiency of manufacturing by improving throughput time as well as reducing production costs. A major problem with today's manufacturing systems is poor availability of data relating to manufacturing processes (like temperatures, forces, and torques etcetera).

Jeong, Kim and Jang (2005, p. 1) remarks that "when a new industrial product appears on the market, it is expected to be more functional, lighter, more accurate, and cheaper than previous products." This can be ensured by using "an advanced technological automated manufacturing system with intelligence to machine and assemble products is appropriate." (Jeong et al., 2005, p.1) They also add that in order "to maintain high quality in machining, an in-process product checkup is more recommended than an after-process checkup" as it "guarantees lower costs and less lap time" and enhances "productivity and quality".

Desforges and Archimede (2006, p. 1) write that “the changes in the customers’ demands” is the reason why manufacturing equipment evolved (and still evolves). Desforges and Archimede (2006, p. 1) suggest that the markets of today “increasingly demand more customized high quality products with shorter life cycles.” For this reason, Desforges and Archimede claim that “new manufacturing equipment” should be able to “deal with the improvement of”:

- Flexibility – by implementing programmable numerically-controlled machine tools;
- Re-configurable – by using modular designs which enhance extendibility, scalability, interoperability and portability.
- Productivity – through development of high-speed machining processes;
- Reactivity – by implementing diagnostic, monitoring or adaptive control functions. (The term ‘reactivity’ refers to the need to maintain productivity and to minimise downtime.)

3.2 USES

Tansel, Bao, Reen and Kropas-Hughes (2005, p. 1) state that “many monitoring techniques have been developed to detect tool breakage and to estimate tool-wear” by evaluating signals from sensors like accelerometers, thermocouples and acoustic emission sensors. Tansel et al. (2005, p. 1) explains that one of the first approaches to monitoring was “to evaluate the characteristics of the signal at different tool conditions and to set the limits or to relate them to tool wear.” They add that “intelligent computational algorithms” were later introduced and used to automate this task.

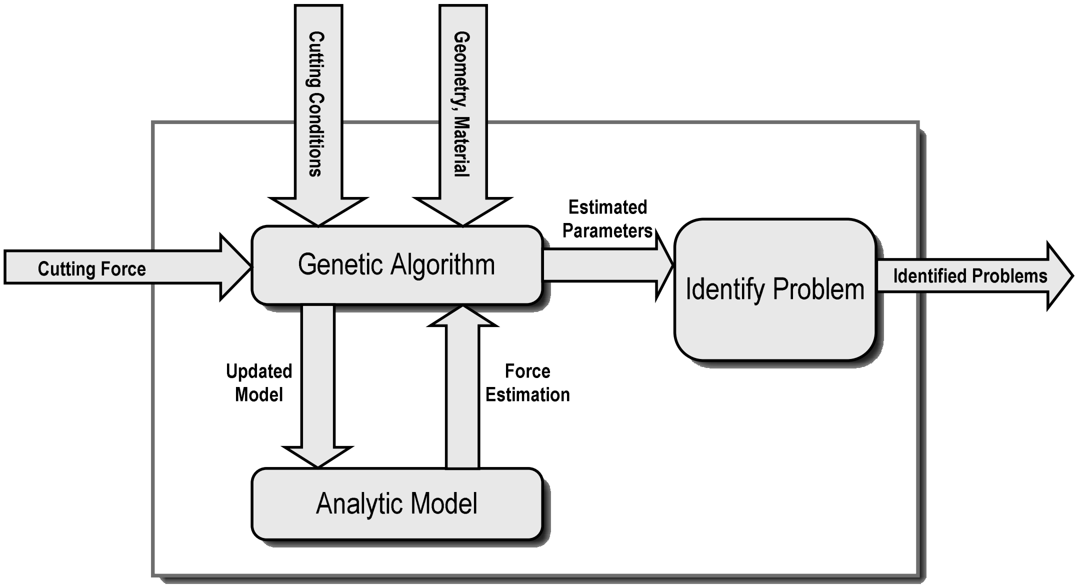


Figure 9: Block diagram of the GTM (Genetic Tool Monitor) (Adapted from Tansel et al., 2005, p.3)

“A major topic relevant to metal-cutting operations is monitoring tool decay, which affects process efficiency and product quality, and implementing automatic tool replacements.” (Cuppini, D’errico & Rutelli, 1990, p. 1) They maintain that “an autonomous tool replacement strategy” should be able “exploit tool life and to prevent failures”. The success of such a tool, according to Cuppini et al. (1990, p. 1), is “the availability of in-process measurements of tool wear” by using a “suitable sensor for real-time application”.

Rivero, López De Lacalle and Luz Penalva (2008, p. 627) explain that “unattended machining” processes require “intelligent monitoring systems”. These systems should be able to “detect the different events that can happen during the machining process” as well as real-time detection of “tool wear”. Rivero et al. (2008, p. 627) suggest that this is especially true “when dry machining is applied”.

Rivero et al. (2008, p. 627) write that “a worn tool causes vibrations during the machining process”. This affects “the quality and integrity of the machined surface”. They explain that it is therefore important to determine tool wear and/or when a tool is worn. The level of the tool’s wear provides an estimation of the tool’s remaining life.

Two methods to monitor tool wear are described by Rivero et al. (2008, p. 627): ‘online’ and ‘offline’ monitoring. ‘Offline’ methods require human or artificial investigation and involve halting the machining process. ‘Online’ methods are more suitable to detect tool wear as they produce data about tool wear during production which can reduce down-time. ‘Online’ methods can also be used “for tool condition monitoring instead of the direct wear measurement” Rivero et al. (2008, p. 627).

Hådeby and Sohlenius (1989, p. 1) assert that when a production line “… is completely unmanned, monitoring is a must [essential]”. Additionally they explain that, irrespective of how errors occur, modern machines should be able to detect and automatically correct any errors or worn-out tools – resulting in no loss of productivity as interruptions should not occur.

Hådeby and Sohlenius (1989, p. 1) express their opinion, that; should limited, skilled or unskilled, manpower exist within a manufacturing plant, it would become necessary that machine tools are equipped with a system of sensors to monitoring the condition of the tool. In such a scenario it would be advantageous to have such a system to predict and detect worn-out tools.

Byrne, Dornfeld, Inasaki, Ketteler, Konig and Teti (1995, p.3) believes that monitoring modern manufacturing systems is required to insure optimum performance:

- On machines for diagnostics and performance;
- On tools or tooling to check state of wear, lubrication and alignment;
- On the workpiece for geometry and dimensions, surface features and roughness, tolerances, and possible metallurgical damage; and
- The monitoring the process itself for chip formation, temperature and energy consumption

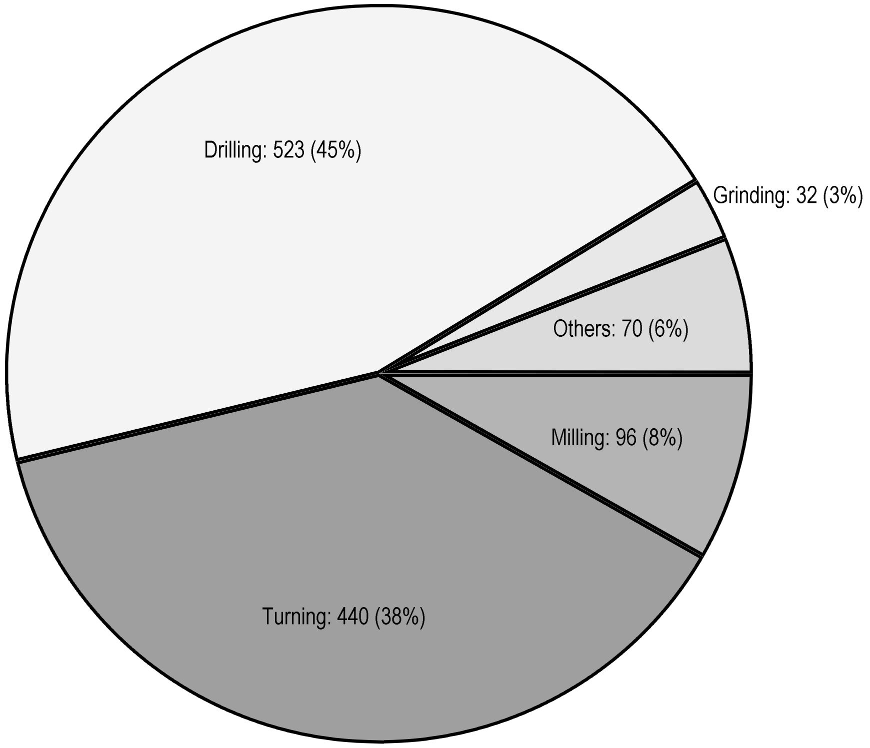


Figure 10: Distribution of monitoring systems by area of application. (Adapted from Byrne et al., 1995, p.3)

Jantunen (2002, p. 1) lists reasons why tool wear and failure monitoring has become a subject of interest among researchers and users. The main reasons are:

- Unmanned production is only a viable option if a method exists to monitor tool wear and to detect tool-breakage;

- Tool wear degrades the quality of the surface finish as well as the dimensions of the parts being manufactured;
- In order to gain the maximum economical tool life, the actual condition of the tool needs to be known and/or determined. Each tool should be seen as an individual tool, having its own tool lifetime.

Dimla (2000, p.2) states that "cutting tool wear can be classified into several types":

- Adhesive wear (associated with shear plane deformation).
- Abrasive wear (resulting from hard particles cutting action).
- Diffusion wear (which occurs at high temperatures).
- Fracture wear (such as chipping due to fatigue).

Monitoring variables which can determine the various tool wear conditions could thus be used to limit or control tool wear, thus insuring optimum performance of the systems.

Dimla (2000, p. 4) indicates that tool condition monitoring systems can:

- Provide "advanced fault detection" systems for "cutting and machine tool[s]";
- "Check and safeguard machine process stability";
- Provide a "means by which machining tolerance is maintained on the workpiece to acceptable limits [i.e. quality is improved]";
- Provide "a compensatory mechanism for tool wear offsets";

- Augment “machine tool damage avoidance system[s]”.

3.3 METHODS

Rauber et al. (1994, p. 1) declares that “sensorial data” forms “the basis for monitoring and supervision” of machinery and tools. They suggest that “an appropriate” sensor should be selected which would best be suited to monitor the selected process variable. Sensor placement and/or positioning should also be considered, as this could adversely affect the performance. The sensor obtains raw data from the running process. The next step, Rauber et al. (1994, p. 1) elaborate, is to transform the raw sensor data into “machine readable and discriminative data”.

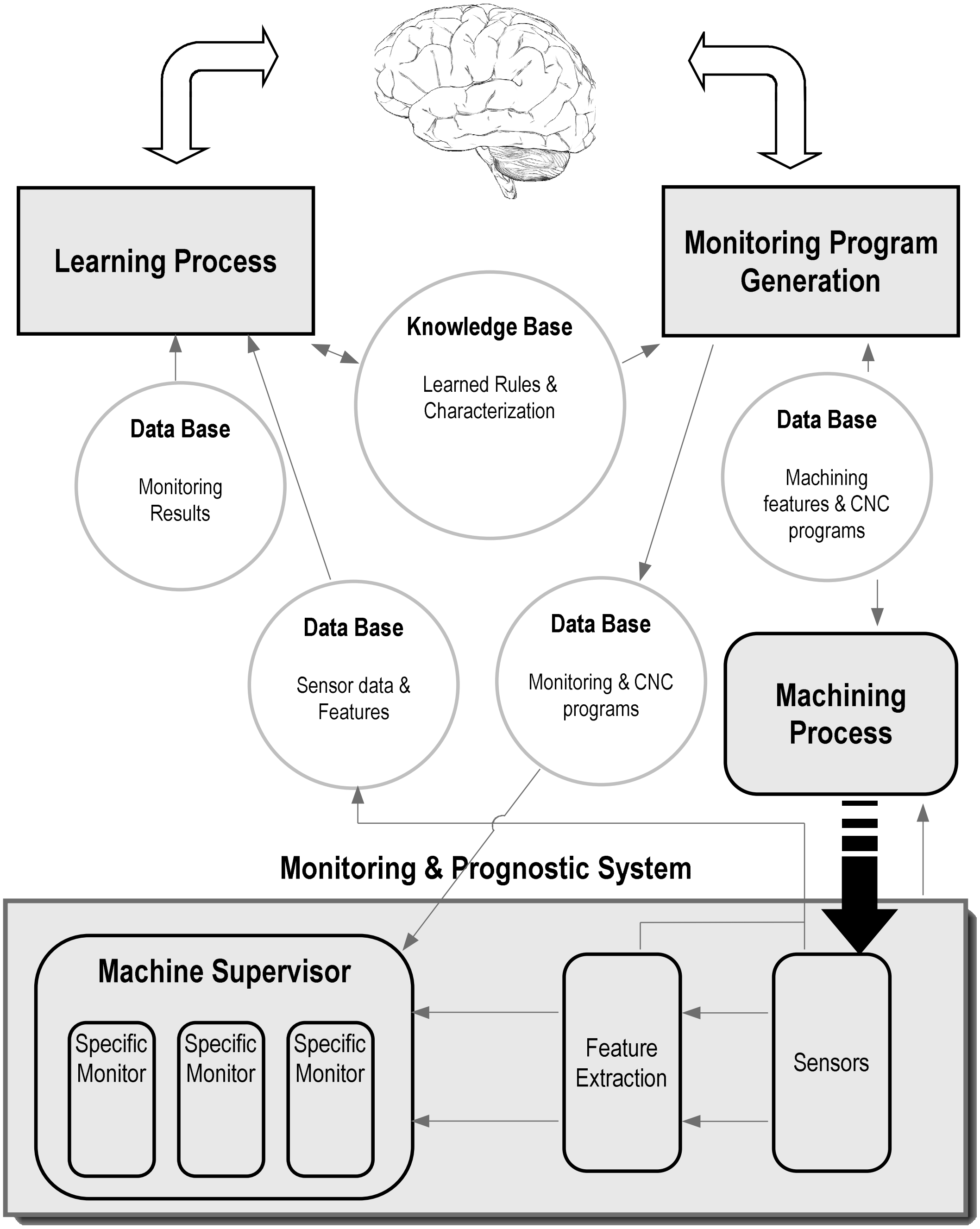
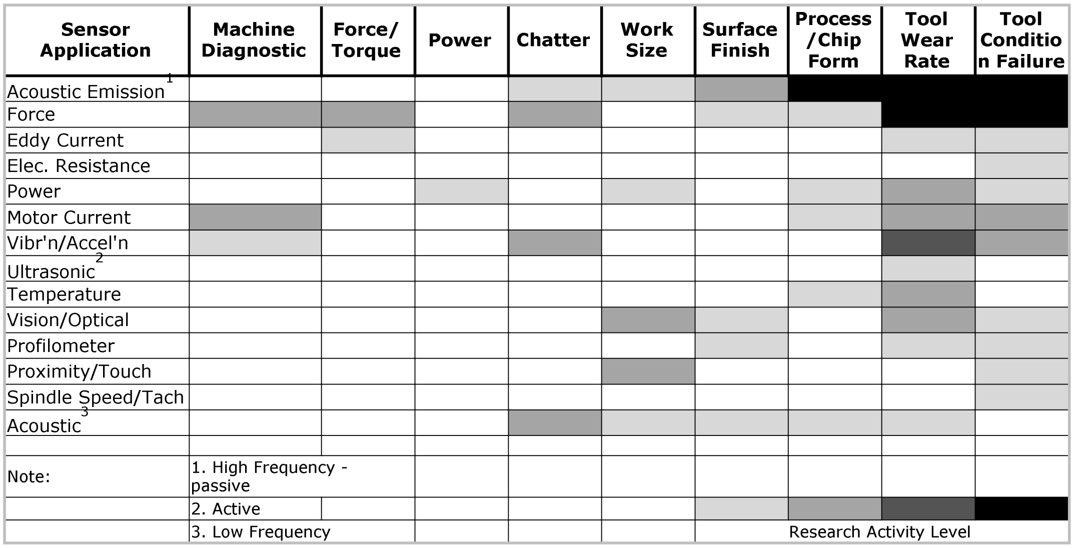


Figure 11: Generic reference model for the learning of a machine tool supervision task (Adapted from Rauberet al., 1994, p.2)

In order to select which sensor to use, Rauber et al. (1994, p. 2) mention the following factors which need to be considered:

- The price of the sensor;
- The real time requirement of the process;
- The ease with which the sensor can be integrated into the rest of the monitoring system.

Sensors are required to ensure the reliability of a system and to provide information. Byrne et al. (1995, p. 2) report that most of the industrial applications of sensor systems are “related to tool breakage monitoring, and tool wear was considered less important”. Byrne et al. (1995, p. 2) highlights that studies have shown that future requirements for sensor systems will be “to insure reliable operation of the process, with quality monitoring [considered] less important” and that systems should be self diagnostic.



Sensor Application	**Machine Diagnostic**	**Force/ Torque**	**Power**	**Chatter**	**Work Size**	**Surface Finish**	**Process /Chip Form**	**Tool Wear Rate**	**Tool Condition Failure**
Acoustic Emission[1]				░	░	▒	█	█	█
Force	▒	▒		▒		░	░	█	█
Eddy Current		░						░	░
Elec. Resistance									░
Power			░		░		░	▒	░
Motor Current	▒						░	▒	▒
Vibr'n/Accel'n	░			▒				▓	▒
Ultrasonic[2]								░	
Temperature							░	▒	
Vision/Optical					▒	░		▒	░
Profilometer						░		░	░
Proximity/Touch					▒				░
Spindle Speed/Tach									░
Acoustic[3]				▒	░	░	░	░	
Note:	1. High Frequency - passive								
	2. Active					░	▒	▓	█
	3. Low Frequency					Research Activity Level			

Figure 12: Table I Sensor Research for Tool and Process Monitoring in Machining (Adapted from Byrne et al., 1995, p.12)

Normally, data received from a sensor is in a raw form, which cannot be processed directly by a digital computer. Also, noise is a natural occurrence of sensor output. Filters are used to counter noise.

Byrne et al. (1995, p.3) holds that sensors used for process monitoring should:

- Measure as close to the machining point as possible;
- Not reduce the static and dynamic stiffness of the machine tool;
- Not reduce the working space, or affect the process parameters;
- Be wear- and maintenance-free, easily changeable and affordable;
- Be resistant to dirt, and mechanical, electromagnetic and thermal influences;
- Function independent of a tool or workpiece;
- Have adequate metrological characteristics;
- Provide reliable signal transmission (for example from rotating to fixed machine components).

Byrne et al. (1995, p.4) draw attention to different types of sensor systems used for process monitoring. Continuous as well as intermittent sensor systems exist. Additionally, sensor systems are classified into direct and indirect measuring systems. Continuous systems produce the "measured variable" continuously; intermittent systems measure the variable "during intervals in the machining process". Direct measuring systems measure the actual value of the variable, indirect systems measure "suitable auxiliary quantities" and "deduce the actual quantity via empirically determined correlations”. Each of the different types (classes) of monitoring systems has advantages and disadvantages. Byrne et al. (1995, p. 4) suggest that “the direct sensor” which measures continuously “is the optimal combination with respect to accuracy and response time”.

Byrne et al. (1995, p. 6) suggest that the continuous measurement of the cutting temperature is a “valuable variable which can be measured for the purpose of indirect wear determination”. As a cutting tool wears down the temperature rises due to “increased friction and energy conversion”. Byrne et al. (1995, p. 6) mention that there are other indirect methods for tool-wear monitoring. These include the cutting force, vibration and current measurements.

The smart or intelligent sensor and actuator concept was conceived to address “the lack of reliability in complex systems inherent to numerous sensors and actuators” (Desforges and Archimede, 2006, p. 2).The concept requires combining “information processing and bi-directional communication abilities to the main service functions of sensors and actuators.”

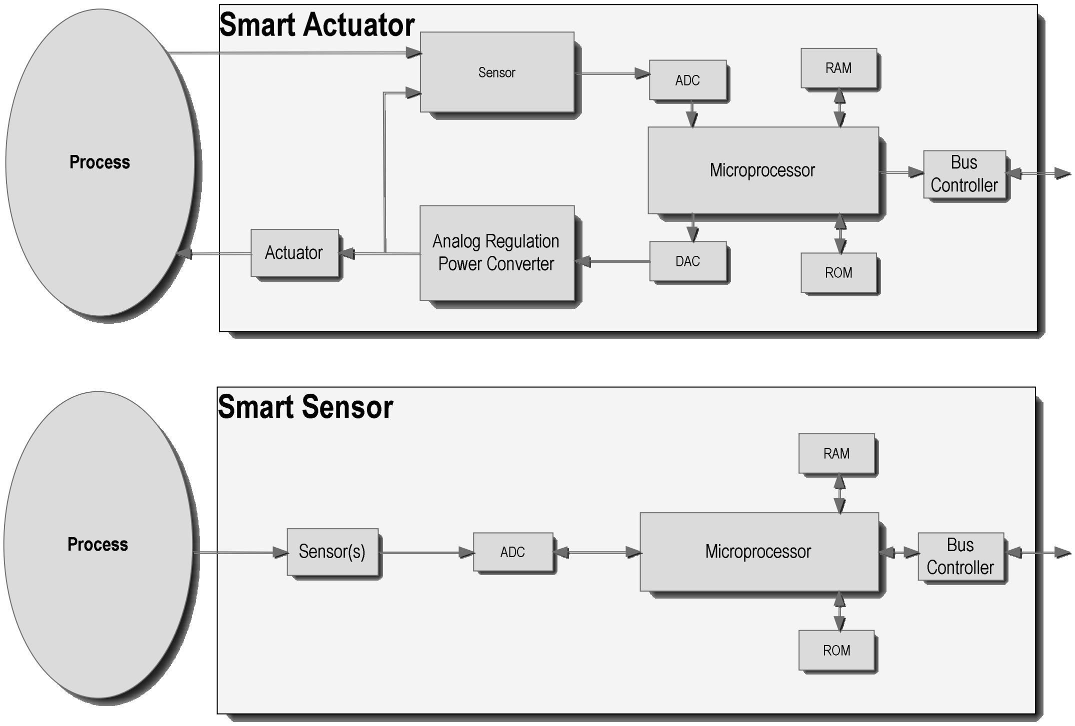


Figure 13: Physical structure of smart sensors and actuators. (Adapted from Desforges and Archimede, 2006, p.2)

As can be seen from Figure 13, smart sensors and actuators have an "information processing ability" (note the embedded microprocessors within the smart sensor and actuator). The primary functions of this "information processing ability", as described by Desforges and Archimede (2006, p. 2), is:

- Measuring;
- Monitoring and diagnosing;
- Acting safely (adaptive control);
- Communicating;
- Managing the activity of the instrument;
- Managing the internal database.

Desforges and Archimede (2006, p.3) report that these outlined functions are designed to improve or ensure:

- The metrological quality and/or certainty of measurements using validation treatments;
- The reliability of the system by providing reliable information;
- The reliability of the sensor or actuator due to self-monitoring and self-diagnosis functions.

The effectiveness of a condition-monitoring system relies on two elements, outlined by Milfelner, Cus and Balic (2005, p.2), are:

1. The number and type of sensors (this affects the hardware and cost of the system), and
2. The signal processing and simplification methods required to extract usable information (this would affect the speed and efficiency of the system).

Systems with high efficiency and speed, short development time, and with the smallest (feasible) amount of sensors are favoured.

Rauber et al. (1994, p.2) followed the strategy of obtaining and providing as much sensor data for a “specific situation”. They suggest that “analyzing tools” could then be utilized to extract useful information from the “total amount of [cached] data”.

Rauber et al. (1994, p. 2) expresses the opinion that certain parts of a process could “be sensed and checked autonomously”. Rauberet al. (1994, p. 1) used methods which differed from “model-based supervision and fault detection systems” as “no analytical model of the process” was established. Supervision of the system was carried out using “an inductive supervised learning … [method based on]… pattern recognition” (Rauber et al., 1994, p. 1).

Rauber et al. (1994, p. 1) insists that in order to obtain an automatic machine supervision system, an “autonomous inductive learning methods have to be employed”. According to Rauber et al. (1994, p.1) the components for learning involve:

- A human expert;
- Semi-automatic tools for sensor data analysis;
- Autonomous building blocks for the identification of process states.

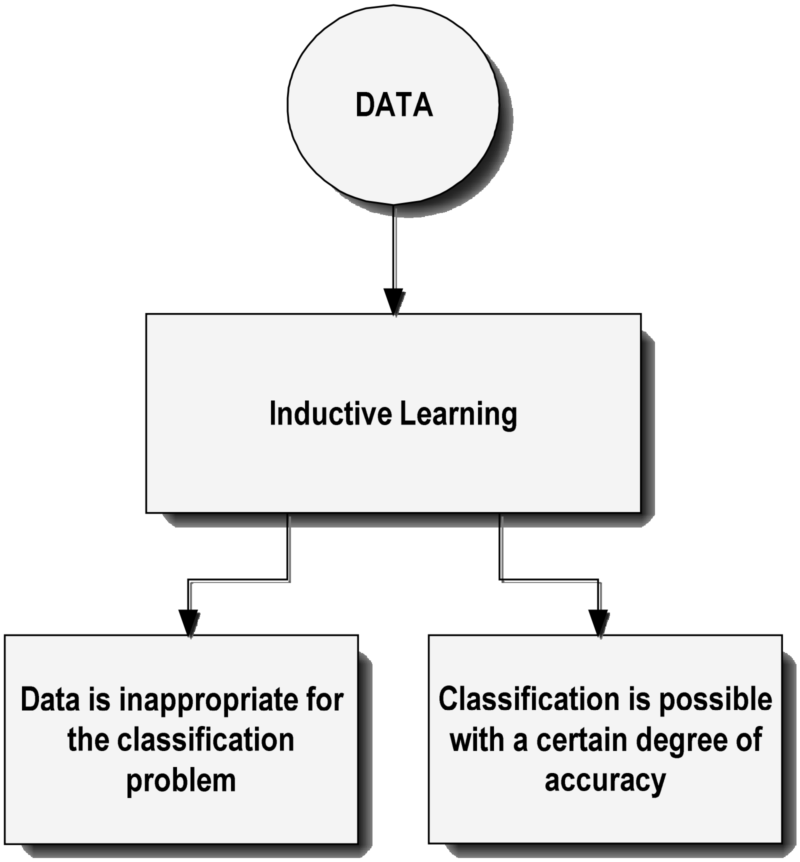


Figure 14: Conclusions from the inductive learning step (Adapted from Rauber et al., 1994, p.3)

Rauber et al. (1994, p. 1) maintains that it is possible to symbolize data from autonomous inductive learning methods, by “starting from a set of continuous sensor [data] features that describes the process pattern”. Symbolizing the data into recognizable patterns allows “learning in the traditional sense of pattern recognition” to occur. Rauber et al. (1994, p. 1) also used a “software toolbox” to perform “statistical feature selection” and a “supervised clustering procedure” was used to generate the specific process monitor algorithm.

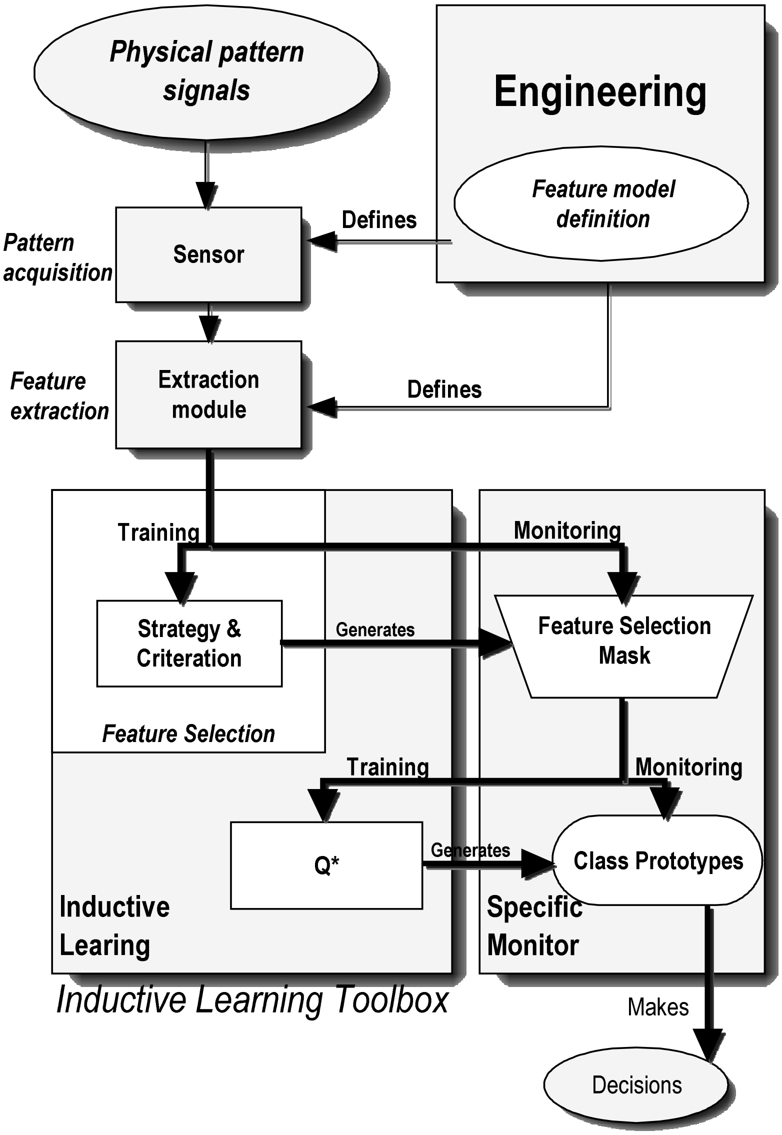


Figure 15: Inductive learning of a Specific Monitor (Adapted from Rauber et al., 1994, p. 3)

3.3 PREVIOUS RESEARCH

Blignault et al. (2006, p. 34) performed research on FSW, using 0.5 mm type-K thermocouples to measure tool temperature. These thermocouples were calibrated to measure up to 800ºC, and their small diameter ensured a rapid response time. The thermocouples were placed within the tool's tip. A sampling speed of 1 kHz was used which provided high accuracy. It was also suggested that by acquiring multiple force and temperature responses during FSW, the effects of process parameter changes and current and optimal process states could be modelled more effectively. Additionally, the possibility for an automated intelligent monitoring and control of the welding process exists.

Mishra and Ma (2005, p. 50)reports that FSW has the potential for creating effective welding joints of titanium alloys, but also suggested that "more research is needed in order to understand the microsctructural evolution of titanium alloys during FSW and other operative mechanisms". It was stressed that one of the most critical issues of FSW which need to be investigated is the "control of process temperatures and cooling rates" for "FSW of α/β titanium alloys to optimise the microstructure".

Rivero et al. (2008, p. 630) claim that heat produced during the "dry milling processes is critical to tool life because high temperatures cause the aluminium to adhere on tool surfaces". Tool damage, according to Rivero et al., is mainly caused by formation of built-up layer (adhesion layer) or built-up edge. Monitoring systems that measure a tool's critical temperature points, would therefore be advantageous. Rivero et al. (2008, p. 631) used “a non-contact method to measure the tool temperature”. This method was based on using infrared to create a thermal image, as “the intensity of infrared emissions is related to the target temperature”. Using this method it was concluded that at best the non-contact method’s measurements, should be seen as “an indicator of the evolution of the tool temperature” and not as absolute values. “The accuracy of the measurement is low, emissivity changes between 0.7 and 0.82 depending on the tool surface.”

Hsueh and Yang (2009, p. 145) explain that in order to successfully automate machine processes, tool breakage should be reliably and effectively detected and responded to in real-time. Hsueh and Yang also write that “without the use of compliance in the tool-monitoring system, automation of many machining processes would be simply impossible”. They also suggest that using smart sensors instead of conventional sensors, the overall robustness and reliability of the monitoring system is increased.

Choi and Yang (1999, p. 706) suggest that tool monitoring systems require sensors which can distinguish between normal operational state and abnormal states “such as excessive tool war and tool breakage”. They suggest using threshold values which the sensor can use to detect abnormal states. When different process parameters are used, the threshold values should be changed.

Baek, Ko and Kim (2000, p. 266) highlight the importance of having a highly reliable and accurate monitoring system for monitoring tool breakage in real time. They argue that because hard tools are frequently used for cutting purposes, tool failure has become more important. Baek et al. (2000, p. 266) explain that tool failure during face milling is especially troublesome as the cutting process is interrupted. They recommend for tool breakage, the face milling process should be monitored so that tool failure can be detected "within one revolution of the cutter". In their research they presented a "new monitoring system using digital signal processor (DSP) technology embedding neural networks to monitor tool failure in real time".

3.4 NEED FOR FURTHER DEVELOPMENT

Ritou, Garnier, Furet and Hascoet (2006, p. 2033) suggest that "tool condition monitoring (TCM) systems can improve productivity and ensure workpiece quality". Ritou et al. report that there is a "lack of reliable TCM solutions for small-batch manufacturing of industrial parts in milling". TCM systems, Ritou et al. (2006, p. 2026) maintain, must "be completely reliable as soon as the first part is machined" and no false alarms can be allowed. Ritou et al. (2006, p. 2026) report that a "teaching method is used for mass production and most commercial TCM systems". This requires "trial cuts" in order to measure a reference signal, after which thresholds are heuristically set on either side of the signal. This "teaching method" is not possible for small batch manufacturing as "monitoring trial cuts is impossible".

It is however, according to Byrne et al. (1995, p. 541), not adequate to have information relating solely to the tool condition. Byrne et al. suggests that additional capabilities such as “in-process quality control and machine tool diagnostics are a requirement of the future”. They claim that there is a “shift from monitoring the tool condition to monitoring the process condition and the resulting [manufactured] part quality”.

Byrne et al. (1995, p. 541) claim that “new demands are being placed on monitoring systems” within the manufacturing environment, due to “recent developments and trends [such as high speed machining and hard turning]”. There is also a remarkable requirement for precision manufactured parts. This, Byrne et al. writes, leads to the “requirement for cleaner machining processes with minimized emissions and waste generation”.

Jantunen (2002, p. 997) mentions that currently “tool changes are made based on conservative estimates of tool life”. This excludes “sudden failures and at the same time [could] lead to an unnecessarily high number of changes” causing wastage of production time, as well as not exploiting the tool’s “full lifetime”. Jantunen suggest that “automated production control is not really possible without a means for tool wear monitoring”.

When metal is cut, a significant amount of heat is generated. Dimla (2000, p. 1082) highlights two factors that dominate tool temperature, from previous research which was performed on “tool temperature effects on interrupted metal cutting”. These factors are:

- The length of cutting cycles;
- The length of cooling interval between cycles.

"The temperature measurements were found to be lower when the cutting was occasionally interrupted", reports Dimla (2000, p. 1082). Continuous cutting generates more heat under the same cutting conditions. They also found that it "was difficult to instrument a thermocouple for tool-chip or tool-workpiece interface temperature sensing when the workpiece was not hollow". Dimla concluded that the best option was to use "a non-contact measurement technique" (such as infrared thermal imaging) but also saw that the measurements received from this technique was, at best, to be considered as averages rather than the true temperatures. To be able to use thermocouples would prove invaluable.

Dimla (2000) reports that "remote thermocouple sensing appear to be the only worthy way to measure the workpiece-tool temperature [for practical applications such as online TCM]", because a direct measurement of the tool tip or rake face temperature cannot be easily obtained (lack of direct access to the cutting zone).

CHAPTER 4

LITERATURE OVERVIEW – CLOSED-LOOP CONTROL THEORY

4.1 OVERVIEW

Schoitsch (1986, p. 327) declares that "'process-computing' and 'real-time programming' are very closely related terms". These terms, Schoitsch declare, "cover all aspects and fields of application of process computers and microprocessors in industry, research and development and for administrative purposes – i.e. not only (real-time) data acquisition and process control but also communication systems, network control of multiprocessor systems, office automation, interactive database systems, interactive graphics and operating systems development."

Closed-loop control can be applied to many processes, to improve overall machining quality, productivity and cost. Haber-Haber, Haber, Schmittdiel and del Toro (2007, p. 2290) report that during the drilling operation, "as the drilling depth increases, chips cease to be discharged smoothly from the hole" causing "a continuous increase in friction between the drill and the work piece". Haber-Haber et al. explain that this creates a relative increase in "the cutting force and cutting temperature". Real-time control and monitoring of these variables can therefore prevent tool vibration, excessive tool wear (and breakage) as well as protect the spindle mechanism from being damaged. If these variables are not controlled or monitored, Haber-Haber et al. explain, that these damages can cause interruptions in production, which could lead to costly and unnecessary expenses. Furthermore, Haber-Haber et al. express their opinion that closed-loop control can also “increase machining productivity”, by allowing the machine to function at optimized process parameters (overly safe process parameters is normally utilised), therefore “improving process efficiency”, reducing cycle time and avoiding tool breakage.

Hjalmarsson (2005, p. 1) writes that the “ever increasing productivity demands and environmental standards”, calls for the implementation of more “advanced control methods” within industrial processes. Hjalmarsson adds that these methods normally “require a model of the process” which can be expensive to obtain. Hjalmarsson quotes another reference who claims that “obtaining the process model is the single most time-consuming task in the application of a model-based control”.

Hjalmarsson (2005, p. 1) cite the same reference highlighting that when modelling a process with the purpose of controlling it, it is crucial to base the model “on criteria related to the actual end use”, otherwise the results will prove disappointing. Hjalmarsson write that efficient, industrial-based modelling and system identification techniques “have become important enablers for industrial advances”.

Hjalmarsson (2005, p. 2) write that simple (first-order) models which have been “identified from step response tests” have proved to be successful in numerous applications “of PID-control to non-linear processes”. Hjalmarsson writes that these simple models often provide acceptable closed-loop performance. Hjalmarsson (2005, p. 2)explain that "high loop gain in a frequency band makes the closed-loop system insensitive to the quality of the model and the properties of the open-loop system in this frequency band, provided stability can be maintained; cf. having an integrator in the controller which gives unity steady-state gain regardless of the open loop steady-state gain."

Hjalmarsson (2005, p. 2) suggests that the model which are acquired "by system identification" should "be shaped such [so that] high performance can be obtained" and also, if required; that extra robustness can be added without decreasing the performance too much.

Hjalmarsson (2005, p.2) suggests, as outlined below, that different design variables like "experimental conditions, model structure and performance specifications" be selected systematically to ensure stability and performance:

- Ensure that the set of delivered models relate to and describe the "true" system;
- Understand which properties of the modelled system should be accurately modelled and which can be treated as being insignificant. How they affect the performance specifications should also be considered;
- Then, experiments need to be designed which can reveal the validity of the previous two steps;

- Representing the data gained from the experiments mathematically "in a way that is not overly complex";
- The performance-demands should then be adjusted so that the design becomes robust while keeping within the limitations of the modelling accuracy.

Jelali, (2006, p. 441) shows the importance of implementation of control performance monitoring/control performance assessment (CPM/CPA). CPM/CPA, Jelali writes, "is an important asset-management technology" and can be used for maintaining a highly efficient "operation performance of automation systems in production plants". Jelali uses the term "monitoring" to refer to "the action of watching out for changes in a statistic that reflects the control performance over time" and the term "assessment" to refer to "the action of evaluating a statistic that reflects control performance at a certain point in time".

Jelali (2006, p. 441) also insists that "the main objective of CPA is to provide an online automated procedure" used to determine whether "specified performance targets and response characteristics are being met by the controlled process variables" and for evaluating "the performance of a control system".

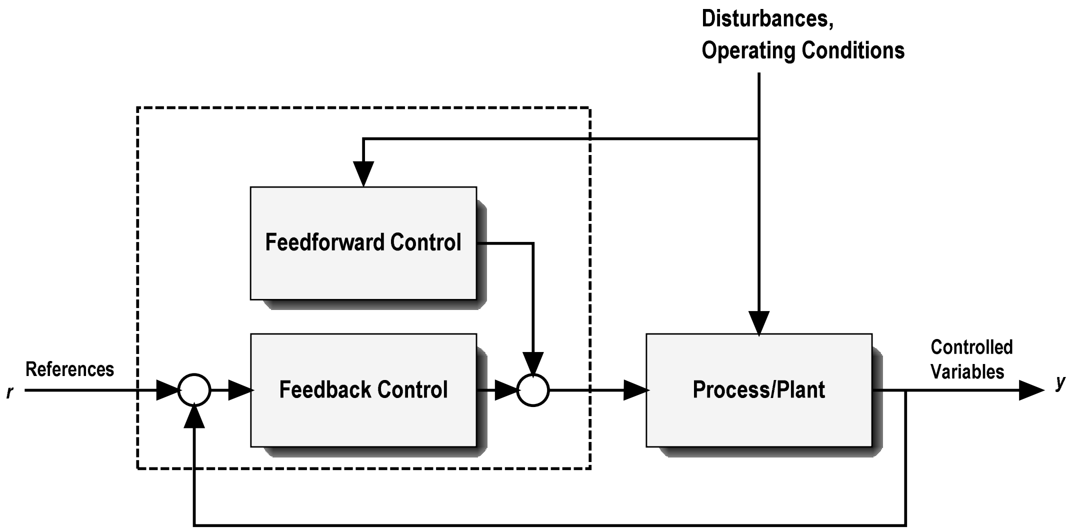


Figure 16: Simplistic statement of control performance assessment problems. (Adapted from Jelali, 2006, p2)

Zhao, Xu & Xie (2008, p. 200) state that "manufacturing is a collection of inter-related activities" which may include "product design and documentation, material selection, planning, production, quality assurance, management and marketing of goods". Zhao et al. remark that the highest level of computer numerical controlled (CNC) automation is obtained with "closed-loop machining (CLM)". This method, they claim, maximizes efficiency by "maintaining a tight control in the manufacturing system". Some of the required elements of a CLM system, given by Zhao et al., are:

- Reliable machines;
- Robust processes;
- Automatic data collection; and
- Continuous improvement of machining process.

Furthermore, Zhao et al. (2008, p. 200) suggest that monitoring is an essential element within a closed-loop manufacturing system. It gathers data in order “to achieve precise measurement and monitor the machine tool’s performance during the operation”.

Zhao et al. (2008, p. 201) mentions that “on-machine” (also referred to as on-line) monitoring offer the following fully automated and integrated features:

- Measurement-taking (whilst being productive)
- Data collection
- Data feedback
- Process adjustment

Zhao et al. (2008, p. 201) adds that “a [manufactured] part can be measured at the machine and corrected there” when by means of on-machine monitoring. “Small samples can be made and then checked immediately.” Over and under-compensation “can be identified at an early stage”. Zhao et al. list the benefits of CLM, which are obtained by “incorporating” on-machine monitoring:

- A reduction in the reject rate “by proofing fixture and part setups, automating offset adjustments, and monitoring the machine’s ‘health’ through self-checks”;
- A reduction in the cost of parts because tools can be standardized more easily and operator intervention is reduced;
- A reduction in costs of inspection by “eliminating hard gauging, increasing the flexibility of measuring methods, avoiding cost of

acquiring a coordinate measuring machine (CMM), and overcoming the CMM bottlenecking problems"; and

- Data can be collected from "parts, processes and equipment for real-time, adaptive control".

To summarise, Zhao et al. (2008, p. 201) pronounce that "utilizing CLM [together] with on-line monitoring], have the ability to:

- Increase in the reliability of a machine;
- Provide a more controlled environment; and
- Reduce human operating errors.

4.2 METHODS

Lakshiminarayanan, Emoto, Ebara, Tomida and Shah (2001, p. 587) declare that for a model-based control strategy to be successful, high-quality process models should be available. "Most controller-design techniques assume the availability of a suitable parsimonious representation of the process". Lakshiminarayanan et al. writes that models based on first principles are often considered inflexible for controller-design at a regulatory level while linear empirical models, "developed from plant input-output data", are considered as "prime candidates for the design of low level control systems". In order to identify these linear empirical models, Lakshiminarayanan et al. report that "open-loop identification experiment[s]" are traditionally used, "where the manipulated variables of the process are perturbed using well-designed signals". Lakshiminarayanan et al. explain that by using this "experimental" plant's input-output data, "an identification algorithm" can be established which can then be used to determine the "parameters of an appropriate model structure by optimizing a criterion of fit".

Using these open-loop experiments to identify the process model, Lakshiminarayanan et al. (2001, p. 587-8) give the following disadvantages:

- The experiments require a considerable amount of time;
- "May let the plant drift away from the intended point of operation", causing the "process nonlinearities [to be brought] into play" as well as a "considerable amounts of off-specification products being manufactured";
- The experimenter needs to take the safety limits of the variables and the physical limits of the tools and machine into consideration;
- It is impossible to obtain useful "open-loop plant data from open loop unstable of inherently closed loop systems".

The disadvantages as outlined, Lakshiminarayanan et al. (2001, p. 588) add, have led to extensive research in closed-loop identification where "one or all of the loops are closed during the identification experiment". Lakshiminarayanan et al. claim that closed-loop identification "is also an important component for controller performance assessment", another topic which has drawn considerable attention.

Landau (2001, p. 51) writes that "design methodology which is the core of an engineering discipline has an integrating character". Landau adds that in order to design a controller “in the most efficient way”, “various areas of control” are required to interact. “This interaction is also extremely beneficial”, Landau explains, in order to gain “a better understanding of control problems” as well as to develop “new algorithms” with a “more coherent approach to control design”. Landau mentions one such example: “the interaction between system identification and control design”.

Landau believes that model identification in closed-loop can provide good design models, and is considered “in practice to be one of the most convenient method[s]” to use.

Landau (2001, p. 52) mentions one of the key problems in many applications is the requirement of a lower order controller. One can reduce the order of the controller by reducing the order of the model and then redesigning the controller or “to directly attempt to reduce the order of the designed controller”. Landau affirms that these two methods should “be carried on with the objective of preserving the closed-loop properties” and both “can be formulated as closed-loop identification problems”.

Landau (2001, p. 52) remarks that the “’sensitivity’ functions that play a key role in robust control, also play a fundamental role for identification in closed-loop”.

According to Landau (2001, p. 52) the two main motivations for “plant model identification in closed-loop” are:

- “Identification of plants with integrator behaviour” or those which “are unstable in open-loop”; and
- “Identification of a plant model when a controller already exists”.

Some of the advantages of “an iterative combination of identification in closed-loop, and control redesign” are summarized by Landau (2001, p. 52) as follows:

- The designed controller can be validated and be re-calibrated (re-tuned) on-site;
- Better models are obtained for the design of controllers;
- Maintenance of controllers is possible;
- Iterative identification in closed-loop and controller redesign; and
- The order of the controller is reduced.

Landau (2001, p. 52) explain “validation of the designed controller” as the “comparison of the performance achieved of the real system with the desired performance achieved of the ‘design’ system”. According to Landau this may involve:

- Time-domain comparisons;
- Frequency-domain comparisons;
- Closeness of the poles; and
- Statistical tests.

De la Sen (2009, p. 960) states that adaptive control is gaining “interest in real-life processes”. It is “based on parametrical estimation of totally or partly unknown physical processes”.

Collis, Joslin, Seifert and Theofilis (2004, p. 248) state that it is important for closed-loop control that “both actuators and sensors be designed and utilized in an effective way”. Controllability and observability are considered as key concepts in “linear control theory” as “they are related to the design and use of actuators and sensors”. They are explained as follows:

- “Controllability is a property of both the actuator system and the state that determines whether all the state modes can be arbitrarily influenced by the control actuator(s)”;
- “Observability is related to the ability of a particular sensor system to reliably measure changes in the state”.

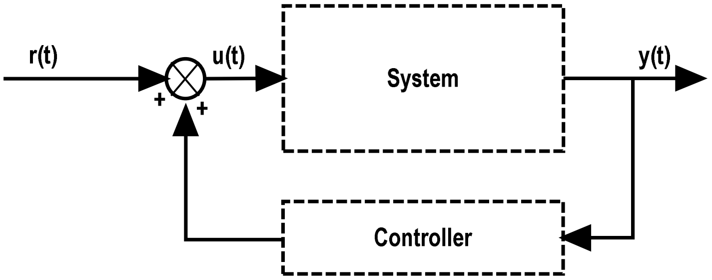


Figure 17: Typical closed-loop block diagram. (Adapted from Collis et al., 2004, p. 249)

4.3 PREVIOUS RESEARCH

Schoitsch (1986) studied the software aspects of real-time system design. Sequential programming was not applicable for their design due to the real-time constraints. Three methods of running processes were explained by Schoitsch (1986, p. 330) (also see Figure 18):

- Sequentially, that is when Process 1 needs to finish before Process 2 can start.
- Parallel, used in a multi-processor environment, that is Process 2 can start running whilst Process 1 is running.
- Quasi-parallel, used in single-processor environment where Process 1 and Process 2 run concurrently, but are “sliced” so that they run at different time slices.

Parallel processes are also referred to as concurrent processes by Schoitsch (1986, p. 329). Schoitsch (1986, p. 330) also explain the term “process” as “a (small) part of the computer program, a logical unit of sequential statements” and “it is the smallest program unit (often called a ‘task’) to be handled by the real-time kernel.

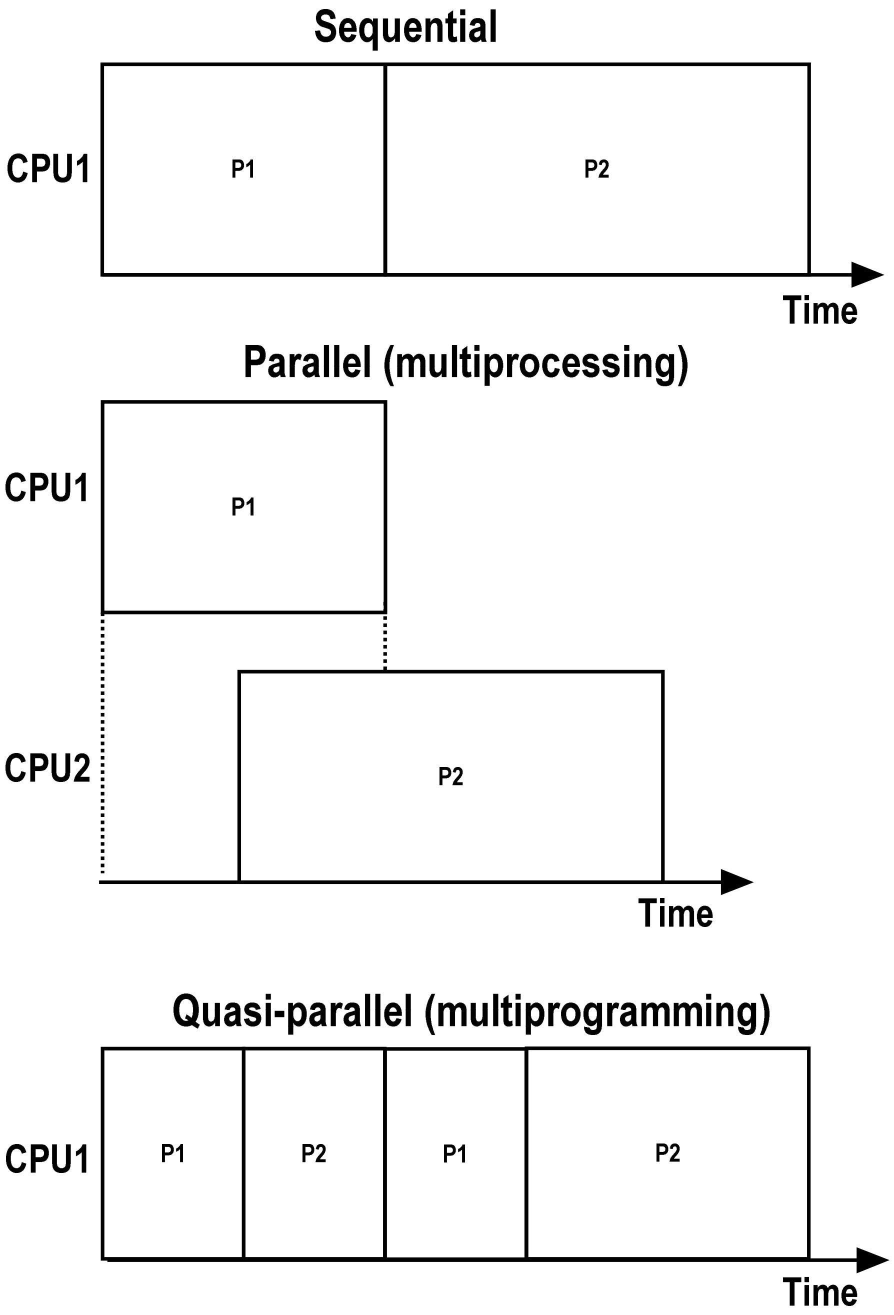


Figure 18: Types of multi-process systems (Adapted from Schoitsch, 1986, p. 330 – p. 331)

Schoitsch (1986, p. 330 - 331) describes that processes on their own, can be written and executed in three different ways (also see Figure 19):

- As a single process – which is executed from a beginning-statement to an end-statement;
- As a cyclic process – after being initialized, the process will wait for a certain trigger-event (wait-condition) and, when triggered, executes sequentially afterwards returning to the wait-condition;
- As a generalized-cyclic process – this process has more than one pre-programmed wait-condition.

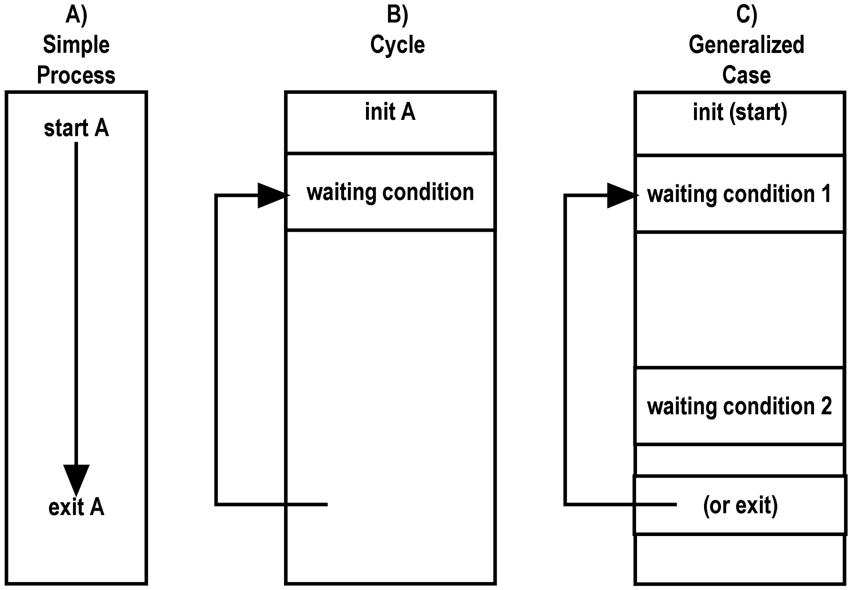


Figure 19: Types of processes. (Adapted from Schoitsch 1986, p. 331)

Schoitsch (1986, p. 330 - 331) reports that processes can also be called using the following methods:

- Sequential call: “this is the standard procedure [or function] call.” After being called the procedure or function is active and will execute until finished, returning to the calling process/program.

- Concurrent call: “this call starts a new program”, thus both programs exist concurrently. The called program does not return to the calling program. The way (the actual sequence) in which the concurrent programs are executed depends on priority, timing conditions or events or other process environment conditions (like external events, available resources, parameters).

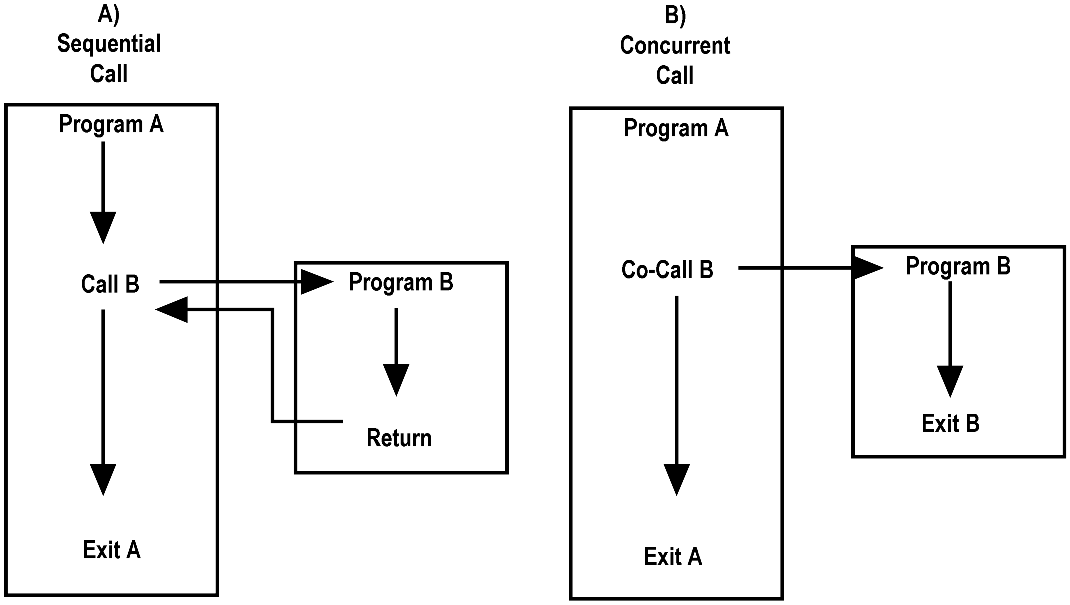


Figure 20: Program flow control in case of (a) sequential call; (b) concurrent call. (Adapted from Schoitsch, 1986, p. 331)

Schoitsch (1986, p. 331) explain that using a sequential approach to writing software (this includes single processes or whole procedures and functions) can produce a process or whole procedures and functions which are:

- “Independent of the speed of execution of statements”;

- Results are reproducible, that is the same data gives the same result;
- "Stepwise testable" and traceable, meaning that it has testing advantages.

Schoitsch (1986, p. 332) writes that these properties allow testing of the process by "tracing, detecting and elimination of errors [in the process' code] in a systematic manner".

Schoitsch (1986, p. 332) explains that "a truly concurrent solution" has to deal with:

- A data-acquisition process request "may interrupt a data-processing cycle and has to be served immediately [due to the process-environment's existing timing conditions]".
- A request may arrive unexpectedly. If a process interrupts the execution of another process, with the same priority, both processes may need to be executed asynchronously.

Schoitsch (1986, p. 332) suggests that "asynchronous processes need to be controlled carefully" in order to cope with "undefined interrupts, unexpected events or other disturbances". Schoitsch adds that by "increasing the degree of concurrency", the "real-time-response" may be improved on condition that process-switching and data buffering are effectively implemented.

Thus, it is much more difficult to find errors and therefore eliminate them from the process' code. Independent code sections can still be evaluated in a sequential manner, according to Schoitsch (1986, p. 332), but this is not the case for the critical code sections "where common data are modified". If concurrent processes are allowed to exist and are in no way restricted; the system will run in an uncontrollable fashion, being controlled by events triggered in the process environment. Schoitsch claims that this causes "time-dependent errors" due to the results of the uncontrolled processes not providing reliable data or operations on data. These errors are difficult to trace because their effect often only becomes apparent later, when the cause of the error has disappeared. Schoitsch insists that this is "one of the main drawbacks in real-time programming".

Schoitsch define critical regions and critical sections:

- Critical regions – refers to shared data and resources which control the interaction of parallel processes;
- Critical sections – refers to software code sections (program sections) which have access to these critical regions.

"Implicit interactions between processes occur if critical sections are overlapped in time."(Schoitsch, 1986, p. 333)

In order to avoid this, Schoitsch(1986, p. 334) suggests using "some mechanism" in order to "guarantee the so-called 'mutual exclusion' of critical regions". Various approaches to this problem exist:

- By creating semaphores – These are variables which allow “only two basic and uninterruptable operations”;
- “the critical regions are high-level language constructs, which associate in a block-structured manner [both] critical sections and critical regions and avoid separate semaphore variables”

Schoitsch (1986, p. 334) suggests that in order “to provide a feasible solution to the ‘mutual exclusion’ problem the following conditions must hold”:

- “Only one process gets access to shared data during its critical section”;
- If another process needs access to “the same set of shared data”, this should be possible “within a finite amount of time”;
- Each process needs to “leave its critical section within a finite period of time”.

Schoitsch (1986, p. 334) also suggest the following guidance with regards to critical sections and critical regions:

- Critical sections need to be “as short as possible”.
- Critical regions should be as small as possible. Having a large number of small critical regions is preferred.
- Critical timing conditions should be handled “in a well structured manner”. (Critical sections should be executed within a finite amount of time).

Haber-Haber et al. (2007) used a Kondia HS1000 HSMC to control an open CNC controller to perform experimental tests on high-performance drilling. Using experimental identification they created their model. A PC was used to implement the control algorithm and also to record the input and output data. This data was used for post-processing and analysis. Haber-Haber et al. recorded data with nominal cutting conditions for their drilling process. This data was then processed in order to obtain the best open-loop transfer function to model the system. A PID-controller for high-performance drilling was designed and implemented. A sampling frequency of 1 kHz was used to measure the orthogonal force components. It was found that using a closed-loop control system gave an "unquestionable superiority" over the system without control (Haber-Haber et al., 2007, p. 2296). A lower cycle time was also obtained which proved an obvious advantage in productivity. A further advantage was having a continuous constant cutting force when using the control system. This would increase tool life.

Endelt, Nielsen and Danckert (2006) performed research which focused on designing of a control system for deep-drawing operations. Owing to the non-linearity of the system, a simplified linear model would "most likely not apply" (Endelt et al., 2006, p. 427). They applied a non-linear finite element model. They also applied a non-linear least square solver to identify the gain factors. The drawing error decreased from 9mm (open-loop) to 0.6 mm (closed loop). Their control system also proved that it could handle large noise impacts.

Poignet, Gautier, Khalil and Pham (2002) developed a model for a high speed machine tool axis. They used this model to tune the control parameters in order to predict the closed-loop performances for their machine. “A frequency approach using optimization routines from Matlab” was used to calculate control parameters (Poignet et al., 2002, p. 485). The mechatronic system was simulated using Simulink. They reported a notable difference between the real and the calculated values. This was attributed to a mismatch “between the actual and calculated values of the model parameters, especially the friction parameters” (Poignet et al., 2002, p. 485). They proposed that “in the case of an existing machine tool” identification techniques could be used to estimate these parameters.

Zhao et al. (2006) concluded that the advantages of on-machine monitoring speak for themselves. Rejected parts are reduced, “parts costs are reduced as well as inspection costs” (Zhao et al., 2006, p. 212). On-machine monitoring also enables real-time, closed-loop control.

Lakshiminarayanan et al. (2001, p. 598) used “a systematic identification approach to the identification of an industrial chemical process”. An analysis technique called the canonical variate analysis was used to identify an empirical plant model. They achieved “satisfactory closed-loop control” even “during major process upsets”.

D’Errico (1998, p. 43) modelled and designed a computer-based adaptive system for closed-loop control of the cutting temperature of a lathe-turning process. A standard thermocouple was used and “embedded in the tool holder”. The cutting temperature was estimated through electromotive force (EMF) measurements obtained from the thermocouple. The control system’s objective was “to maximise metal removal rate” whilst keeping tool damage (due to temperature-dependent mechanisms) to a minimum. D’Errico also writes that “the cutting temperature is therefore a subject of current interest which relates to the development of adaptive control (AC) systems for machining purposes”. It is also reported that the current usage of AC systems is under-exploited within the industrial sector which can be attributed to the lack of reliable, non-intrusive sensors. Another reason, D’Errico adds, is the requirement for suitable dynamic models “to describe a priori information (mainly the range of the steady-state response and time scale of the transient response)” as well as “to track process dynamics variation (mainly related to the cutting parameters and to the tool and workpiece materials)”.

4.4 METHODS

In order to apply a real-time closed-loop control system for a drilling process, Haber-Haber et al. (2007, p. 2292) followed the following steps:

1. The first task is the creation of a mathematical model of the process. “The model serves as a basis” for designing of the entire system. Experimental identification is carried out to create this model;

2. To perform these experimental identification tasks, “a specific input function excites the system, and [the] output variables are [then] recorded”. “Strict synchronicity” is required “to preserve any process lag or dead time”;
3. After all identification trials are processed, “the open-loop transfer function that best approximates” the actual process is determined;
4. The next task is to design the PID-controller based on the mathematical model. This will determine the system’s performance.

Endelt, Nielsen and Danckert (2006, p. 426) state that one of the first challenges with designing a closed-loop control system is “the choice of measurable state variables”. When the measurable state variables are determined, the second challenge is designing the control system. In order to achieve this, a suitable strategy needs to be decided upon. Additionally a decision has to be made on which process variables to use as actuators.

CHAPTER 5

LITERATURE OVERVIEW – WIRELESS REAL-TIME COMMUNICATIONS SYSTEMS AND SENSORS

5.1 REAL-TIME SYSTEMS

Campa, Kelly and Santibanez (2004, p.1022) writes that a real-time system is a system where it is “absolutely imperative” that the response time of the system falls within “the required deadline”. They remark that “real-time systems arise from the intersection of control and computing engineering, and they are the base for the operation of PC-based control systems”. Campa et al. also reports that some research has already been done on real-time control systems running on Linux, QNX and Windows NT.

Younis, Aboutabl and Kim (2004, p. 649) writes that the “use of computers for embedded real-time control systems” has significantly increased during recent years. Real-time applications are “distinguished by the fact that their functional semantic is coupled with temporal correctness”. Younis et al. (2004, p. 650) explains that an embedded computer needs to:

- Perform the correct control algorithm;
- Meet all the timing constraints associated with the algorithm;
- Ensure robustness and uninterrupted service.

“Some real-time applications are mission-critical and require a high level of dependability”, Younis et al. (2004, p. 650) writes, and mentions “avionics and nuclear reactors systems” as good examples. For these systems “it is necessary to be able to show, with a high level of assurance, that a problem or failure in one application cannot disrupt or corrupt other [running] applications".

Lam, Kuo, Kao, Lee and Cheng (2002, p. 123) alleges that "previous research in real-time concurrency control" mainly focused on the "schedulability guarantee of hard real-time transactions" as well as "the reduction of miss rate of soft real-time transactions".

Younis et al. (2004, p. 650) insist that "development and maintenance costs of control units" can be reduced by using "an integrated design approach". This method is alleged to encourage "the use of general-purpose components and sharing common resources".

Lam et al. (2002, p. 123) write that "transactions in a real-time database system (RTDBS) are usually associated with deadlines". They explain that deadline violations of any "hard real-time transaction" can cause a catastrophic failure, whilst deadline violations of a "soft real-time transaction" could only result in a "degraded level of system performance".

Lam et al. (2002, p. 124) report that "researchers have proposed various efficient real-time concurrency control techniques to either reduce the number of deadline violations for soft real-time transactions or guarantee the deadlines of hard real-time transactions". Within these systems it is critical "to handle events and requests in a ‘real-time’ manner”. Lam et al. mentions some of the potential applications of RTDBS:

- programmed stock trading systems
- air-traffic management systems
- air-navigation systems, and
- critical patient monitoring systems

Lam et al. (2002, p. 124) explains that different “performance goals” exist for processing “soft real-time transactions” and “non-real-time transactions”. For “soft real-time transactions” the goal is “to minimise the number of deadline violation[s] (or to meet the system’s statistical timing constraint)”. This differs from when “processing non-real-time transactions”, where it is more important to “maximise system throughput (or to minimise the mean response time)”.

5.2 THE WIRELESS SENSOR

5.2.1 Overview

This chapter will describe the basics of a wireless sensor. The different components of the sensor will be investigated. Advances in miniaturisation and low-cost, low-power electronics have led to vigorous research being made in wireless networks created by small, wireless, low-power sensors and actuators.

5.2.2 Background

The converging of the Internet, communications, and information technologies; together with advances in technology, has paved the way for a new generation of inexpensive sensors and actuators. Sensor design has advanced and will continue to advance in the future. Sensors will become smaller, weigh less and cost less. Furthermore, they will become more accurate and consume less energy.

> "Automated sensing, embedded computing, and wireless networking are not new ideas. However, it has only been in the past few years that computation, communication, and sensing have matured sufficiently to enable their integration, inexpensively, at low power, and at large scale." (Raghavendra, Sivalingam and Znati, 2006, p. 8)

Out of this integration (of computation, communication and sensing) the wireless sensor was born.

Wireless technologies and low-power VLSI (Very Large Scale Integration) designs became feasible during the 1990s. It was then, according to Krishnamachari et al. (2005, p.1), that researchers began "envisioning and investigating large-scale embedded wireless sensor networks for sensing applications."

Krishnamachari et al. (2005, p.1) write that these technological advances "allows us [engineers] to envision a future where large numbers of low-power, inexpensive sensor devices are densely embedded in the physical environment, operating together in a wireless network." Typical applications of these wireless sensor networks are:

- ecological habitat monitoring
- structural and seismic monitoring
- environmental-contaminant detection
- industrial process control
- military surveillance and target tracking

Scheideleret al. (2004) maintain that networks of wireless sensors will be widely used in the future, because they greatly extend our ability to monitor and control the physical environment from remote locations.

Some features and advantages of sensor networks include:

- large numbers of sensors
- large area that can be covered (if large numbers of sensors are used)
- low energy use
- Collaborative signal processing
- failure of individual sensors has no major impact on the rest of the network
- human intervention can be minimized
- hostile and unattended environments can be reached

Wireless sensor networks provide bridges between the virtual world of information technology and the real physical world. They are embedded in the real world. They sense and detect the world's physical nature (like light intensity, temperature, sound, proximity of objects etcetera) and they affect the world in some way (by a fan being switched on, by making a noise, or exerting a force). Krishnamachari et al. (2005, p. 1) write that wireless sensors provide unsurpassed abilities to "observe and understand large-scale, real-world phenomena at a fine spatio-temporal resolution."

Wireless sensor networks will open up and provide access to further technological advances.

5.2.3 The Components of a Wireless Sensor

Despite their variety, all wireless sensors have the same fundamental components. There are several key components that make up a wireless sensor namely:

- Microcontroller (or controlling element)
- Sensor
- Radio transceiver
- Power source

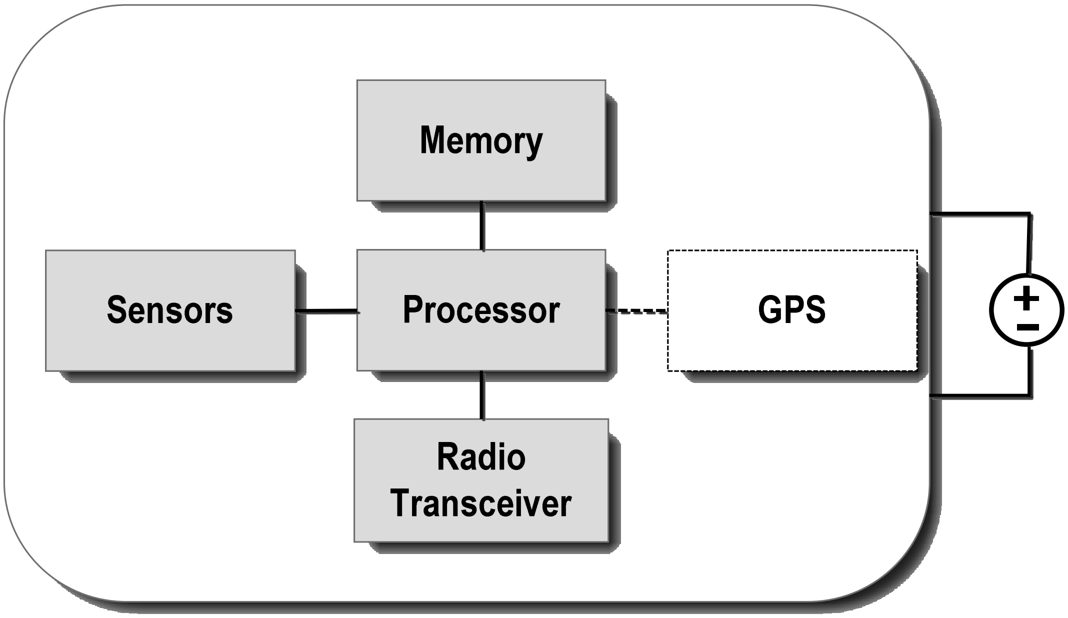


Figure 21: Block Diagram of a Wireless Sensor (Adapted from Krishnamachari et al., 2005, p.3)

Raghavendra et al. (2006, p. 208) also explains that "a typical [wireless] sensor node" will consist of "one or more types of sensors to collect data, an embedded processor with limited memory", functioning on battery power, and it will have an "ability to communicate wirelessly with other sensor nodes and devices."

5.2.3.1 The Microcontroller

Ellis (1999, p. 1) writes that the first microprocessor (µP) was developed by Intel during the 1970s. It was designed specifically for an electronic calculator. Soon afterwards, the microprocessor industry exploded into the fastest growing industry in the world.

Many of the first microprocessors were used in microprocessor-based control units. Intel then developed their 8031/51 family of embedded microcontrollers (also referred to as μC's and micro-controller-units (MCUs)).

A microcontroller can be described as a whole computer system on a single integrated circuit (IC) chip. They revolutionised the industry by introducing a device that featured a microprocessor core, program memory (to store the program, also referred to as read only memory (ROM)), data memory (to store data, also referred to as random access memory (RAM)), special peripherals, and programmable input/output (I/O) ports and timers. Some MCUs even have an on-board oscillator. MCUs form the heart of embedded system design.

Embedded means that the user is not always aware that a computer or processor is actually controlling the whole system, like digital radios, engine valve timing, and modern microwave ovens etcetera.

In 1965, Gordon Moore famously predicted that the number of transistors per square inch on an IC would double every year. This means that each year, the computational power of MCUs increases dramatically. The first MCUs were simple 4-bit controllers. At present MCUs have the same computational power (or more) than the high-end computer systems from two or three decades ago, but at a fraction of the cost and size, as well as consuming far less power.

Recently the pace might have slowed down, but in general, the pace has remained true to Moore's (1965) prediction.

With regard to the Wireless Sensor, the MCU needs to process the locally-sensed information as well as process information received (from a PC or other sensors). Krishnamachari (2005, p.3) writes that MCUs are often significantly constrained in terms of computational power. According to Moore's (1965) law, future MCUs will become more powerful, and more energy-efficient.

5.2.3.2 RADIO TRANSCEIVER

Wireless sensor nodes communicate via radio. Thus, the wireless sensor includes a short-range (<100m) low-power radio transceiver, and usually has limited bandwidth for communication. Typical data rates range between a comparatively few tens of Kbps to a few hundred Kbps. Raghavendra et al. (2006, p. 208) highlights that it is "critical to ensure that a sensor network meets its application data communication requirements." Data can be compressed if necessary, reducing the required data rate as well as reducing the overall energy consumption.

Using this transceiver, the MCU can communicate wirelessly with other nodes, and vice versa.

Raghavendra et al. (2006,p.22) notes that "recent advances in wireless communications, digital electronics, and analog devices have enabled sensor nodes that are low-cost and low-power to communicate untethered in short distances and collaborate as a group."

Krishnamachari et al. (2005, p. 3) explains that the radio communication is the most power-intensive component of wireless sensors; therefore the radio must include energy-efficient sleep and wake-up modes. Future radio transceivers will become more sophisticated, cost less, be immune to noise and less fading and interference will occur.

5.2.3.3 SENSORS

A sensor can be defined as a device or component which can be used to measure or detect (sense) a certain physical quantity, property or condition (stimulus), and when this physical quantity is detected it is then converted into another signal or form (like a reading on a calibrated instrument) which can be read and understood by an observer or another device, such as a microcontroller. For example, a speedometer, which is a simple sensor, is used to measure the speed of the car after which it converts the measured speed into a rotating arm which moves on a calibrated scale. (For accuracy, all sensors need to be compared and calibrated against certain known standards.)

De Silva (2007, p. 1)writes that sensors "are necessary to measure output signals (process responses) and to measure input signals for feed-forward control; to measure process variables for system monitoring, diagnosis, and supervisory control; and for a variety of other purposes."

According to Fraden (2004, p. 7) there are many different types of sensors. A sensor may be classified as being:

- Passive, that is when a sensor does not need any additional energy source. The measured input stimulus energy is directly converted by the sensor into an output signal. The sensor therefore generates an electric signal in response to what is being measured; or
- Active, that is when a sensor requires an external power source for its operation. This external power is called an excitation signal. This signal is modified by the sensor in order to produce an output signal.

In addition, a sensor can be classified as being:

- Absolute, that is when a sensor detects a stimulus in reference to an absolute physical scale, which is independent of the measurement conditions; or
- Relative, that is when a sensor detects a stimulus and relates it to a special case.

Fraden (2004, p. 7) suggests that another way to view a sensor is to look at all of its properties. These properties include what is being measured (stimulus), what its specifications are, what physical phenomenon it is sensitive to, how it employs its conversion mechanism, from what materials the sensor is manufactured, and for what application it was designed.

Sensors are available with different response times and differing sensitivities, and can be outlined as follows:

- Response time, as explained by Answers.com (2008, Response Time: Definition from Answers.com, ¶ 2) as “the time required for the output of a control system or element to reach a specified fraction of its new value after application of a step input or disturbance”.

- Sensitivity, as explained by Answers.com (2008, Sensitivity: Definition from Answers.com, ¶ 3), is “the degree of response” of a sensor (measured at the output of a sensor) to “a change in the incoming signal”. Simplified it is, how much the output of the sensor changes when the measured quantity (that is the input to the sensor) changes.

Krishnamachari et al. (2005, p. 4) states that there are numerous types of sensors, each measuring a certain quantity as well as converting the measured quantity into a certain form of output. Some of the most popular quantities which can be measured by sensors, according to SensorWiki.org (2008, Sensors: introduction) include: acceleration (accelerometers), acoustic, biosignals, chemical, flow, force/pressure/strain, humidity, linear position, orientation/inclination, rotary position, rotary velocity, switches (like reed and limit switches), temperature, vibration and light sensors. According to

Wikipedia (2008, Sensor), sensors are used: in “cars, machines, aerospace, medicine, manufacturing and robotics”.

Some of the most useful sensors for mechatronic applications are sensors which measure forces and pressure, heat transfer-rate and temperature. Several common types of sensors in this category are presented, for temperature and force measurements, in the following paragraphs.

Temperature

Huddleston (2007, p. 13) writes that temperature is a widely-used parameter in control application for industrial processes. In order to measure temperature, the temperature measurement device normally senses the required temperature through heat transfer occurring from the source (what needs to be measured) to the measuring device. De Silva (2007, p. 332) explains that a temperature sensor almost always undergoes a physical or a chemical change which is somehow related to the temperature being measured.

Huddleston (2007, p. 13) claim that one of the most common temperature sensors is the thermocouple. De Silva (2007, p. 332) explains how a thermocouple works. When the temperature changes – at the junction formed by joining two dissimilar conductors – the resulting heat transfer will cause the junction’s electron configuration to change. The reconfiguration of electrons produces a voltage (also called an electromotive force (EMF)). This effect is known as the Seeback effect.

Dunn (2006, p. 319) writes that a thermocouple therefore generates a voltage that is proportional to the difference in temperature between the metal junctions.

Huddleston (2007, p. 13) describes a basic thermocouple as a two-wire sensing element, consisting of two dissimilar metals joined together at a single point. The Seeback effect produces a small, but measurable, voltage that varies with the temperature of the junction. Very low temperatures, as well as very high temperatures can be measured.

Thermocouples, according to Huddleston (2007, p. 163), are known to be highly non-linear over their range of measurement.

The graph below shows the response of various thermocouples to temperature.

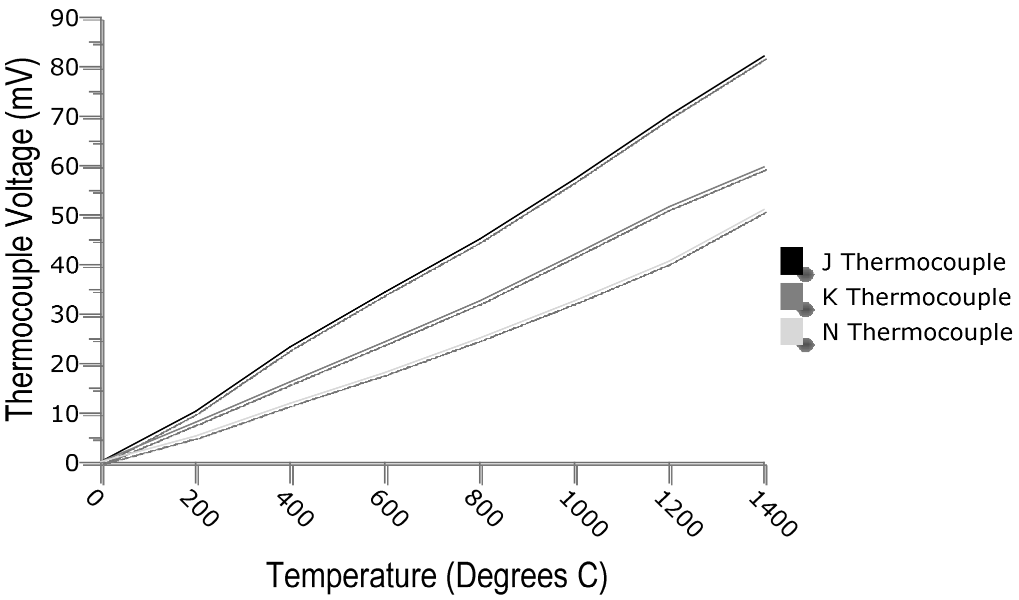


Figure 22: Response curve of various thermocouples (Adapted from Huddleston, 2007, p. 163)

Thermocouples are a popular temperature sensor because:

- They are relatively inexpensive;
- They work over a wide range of temperatures; and
- Their limitations (like its non-linear response) can be compensated for.

Force

Fraden (2004, p. 324) categorizes the different methods of sensing force:

1. Balancing the unknown force against the gravitational force of standard mass.

2. Measuring the acceleration of a known mass to which the force is applied.

3. Balancing the force against an electromagnetically developed force.

4. Converting the force to a fluid pressure and measuring that pressure.

5. Measuring the strain produced in an elastic member by the unknown force.

According to Fraden (2004, p. 325) the most common method is method 5, whilst 3 and 4 are also used occasionally.

Fraden (2004, p. 324) also classifies force sensors into:

- Quantitative – a sensor measuring the force's (stimulus), and converts it into an electrical signal. For a whole range of forces an output is produced.
- Qualitative – a sensor measuring the threshold of a force. These devices are not concerned with good fidelity of the representation of the force value. They are used as threshold devices, merely to indicate whether a sufficiently strong force is detected or applied.

A strain gauge, as described by Fraden (2004, p. 325), is a resistive elastic sensor whose resistance is a function of applied strain. Dunn (2006, p. 319) explains that a strain gauge therefore "converts information about the deformation of solid objects when the objects are acted upon by a force into a change of resistance." All materials resist deformation and require some force in order to deform. Thus, the resistance can be related to an applied force, their relationship is called the piezoresistive effect.

Fraden (2004, p. 326) writes that a wire strain gauge consists of a resistor bonded with an elastic carrier, a backing. The backing is applied to the object for which the force or stress should be measured. Placement of the sensor is critical because the strain must be reliably coupled with the gauge wire. The gauge wire should be electrically isolated from the object under strain.

Fraden (2004, p. 326) explain that strain gauges are designed to have long longitudinal and short transverse segments in order to improve sensitivity.

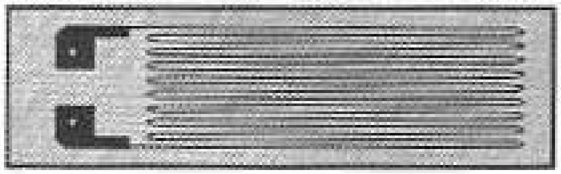


Figure 23: A wire strain gauge bonded on elastic backing (Fraden, 2004, p. 326)

Fraden (2004, p. 326) observes that Wheatstone bridge circuits are typically used to obtain the force reading from the strain gauge. The interface circuit also needs to include temperature–compensating networks because “semi-conductive strain gauges are quite sensitive to temperature variations”.

5.2.3.4 POWER SOURCE

Krishnamachari (2005, p. 4) claim that wireless sensors are almost always battery operated, because this allows the whole system to be much more flexible with regards to where it can be installed or positioned. The limited battery energy is the most critical resource bottleneck for most wireless sensors. Energy harvesting techniques are used in some cases to provide a degree of energy renewal.

CHAPTER 6

THE EXPERIMENTAL SETUP: HARDWARE

6.1 INTRODUCTION

The purpose of this project is to be able to control and monitor process variables wirelessly using Bluetooth technology. An embedded microcontroller capable of sensing and performing control operations is interfaced with a Bluetooth module. This module, in turn, provides a wireless communication link between the microcontroller and a PC. Data and/or commands can be sent or received via this link. The microcontroller's firmware (software which controls the whole system and are stored in ROM) can be reconfigured and reprogrammed wirelessly and without the need of any physical connection between the PC and the microcontroller.

The main focus of this chapter is to explain the different components, what they do, and how they interact or interface with each other. Together they all form part of the experimental setup which was used to perform the tests described in the next chapter.

6.2 OVERVIEW OF SYSTEM

The system consists of a sensor board, microcontroller, Bluetooth module, Bluetooth dongle and a PC.

To monitor, a microcontroller reads the data from the sensor. This data is then communicated to a PC via a Bluetooth wireless link.

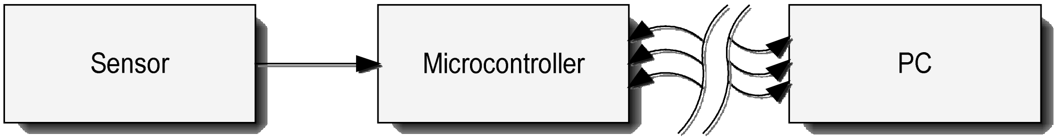


Figure 24: Sending the sensor data wirelessly to the PC

To control, the user (or a software program) sends data (like commands, or configuration/calibration data) to the microcontroller. The microcontroller will receive the data from the module, and can therefore perform the required control function. See diagram below.

Figure 25: Sending the commands or configuration/calibration data to the Microcontroller

The experimental system consists of a:

- Microcontroller (or controlling element)
- Sensor
- Radio transceiver
- Power source

Below is the block diagram of the complete system.

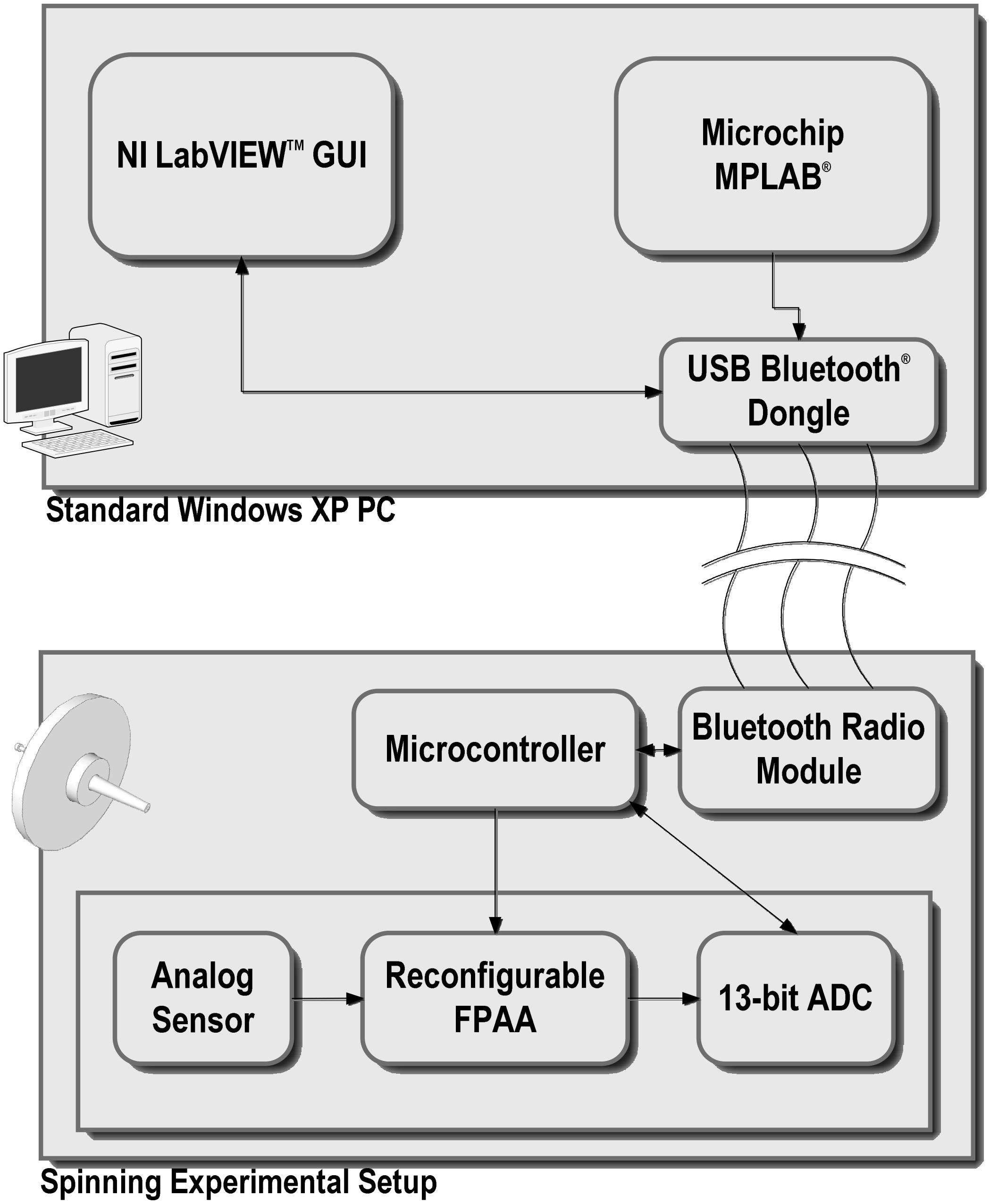


Figure 26: The complete system

The following sections will describe and explain the different sections of the experimental system and which devices were used.

6.3THE EXPERIMENTAL SETUP

The spinning experimental setup contains a sensorboard, a microcontroller, Bluetooth Radio module and a power supply. This section will explain the various components in detail.

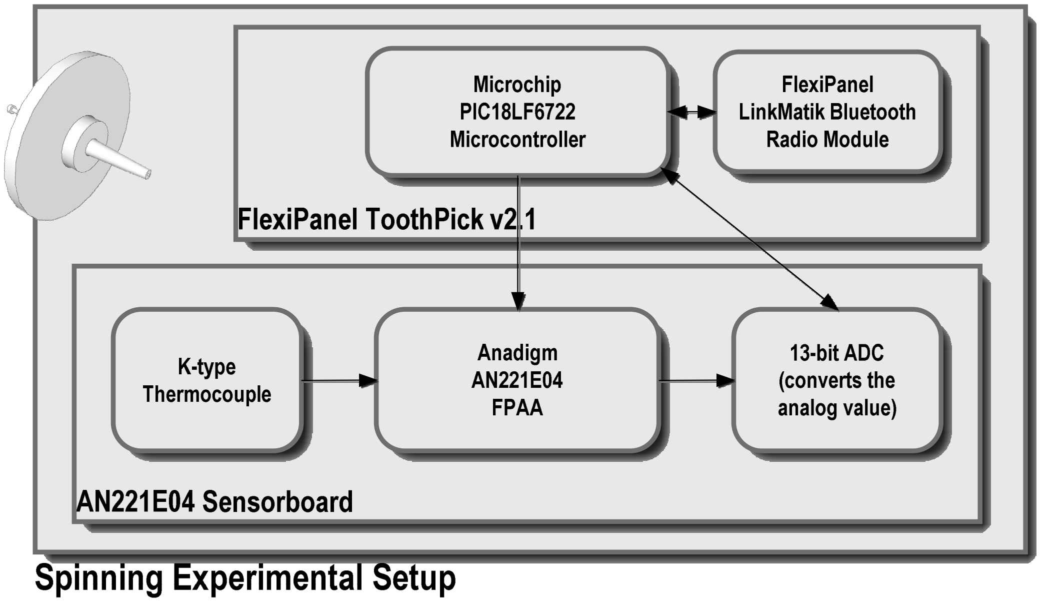


Figure 27: The various components of the spinning experimental setup

6.3.1 The Microcontroller

The Toothpick 2.1 was chosen for this project. The Toothpick combines a PIC18LF6722 and a LinkMatik Bluetooth radio into a small, single printed circuit board (PCB) design. The reasons for this selection are as follows:

- Compactness

A PIC16F877 was originally used to communicate with a separate Bluetooth module. The NMMU's multiPIC board was used to provide a building platform and all the interfaces were connected to this board. This system proved to be overly large and not as robust as anticipated for the considered application (spinning the experimental setup at high speeds). It was decided that a smaller and more robust system should be designed and utilized. The Toothpick 2.1 offers a small and robust solution. The TP21-DIL package was used and provides the Toothpick 2.1 in a 28-pin Dual In-Line (DIL) package, capable of running on a supply voltage between 3.3V and 5V. The DIL package provided the possibility to easily connect the Toothpick to other interfaces without needing to design a PCB. The package dimensions are 51mm x 22mm. The height is less than 10mm.

- Wireless reprogrammability

  The Toothpick 2.1 offers wireless reprogrammability via any Bluetooth-equipped PC. Thus the project's firmware can be updated or changed using Wireless Field Programming or conventional in-circuit programming.

- Programming Interface
  - Wireless field programming (WFP) is a facility which allows Toothpick 2.1 to be programmed via Bluetooth. `ToothpickWFP.exe`– a separate Windows software

application – is used for wireless field programming. This program can also create Service Packs (specialized .exe files for either Windows and/or Pocket PCs) which provides an easy method for distributing firmware upgrades.

- Using this method to program the Toothpick is almost as fast as regular in-circuit programming. This is because the Bluetooth communication time is negligible when compared with the time to write to flash memory.

- The Toothpick enters the WFP mode when the pushbutton is held down for several seconds after power-up. When initialization is complete and the Toothpick enters the WFP mode, the LEDs will flash rapidly and simultaneously. The Toothpick will then wait for the Wireless Field Programmer to connect and begin programming. The required PIN code is four zeroes “0000”. Once the WFP has begun, the LEDs will start flashing rapidly but alternately until programming is complete or fails. After completing the programming, the Toothpick will reset and start executing the code. If the developer’s code is corrupted (for example during programming, the programmed device loses power or goes out of range) recovery is possible by re-entering WFP mode and reprogramming.

- Development software

  Applications for the Toothpick 2.1 can be developed using one of four ways:

  - By using a pre-compiled Standalone Firmware Solution. This method provides a straightforward solution. The source-code, which is provided by FlexiPanel Limited, can be used as a starting-point and for customization.
  - By using the Toothpick Slave Firmware Solution which allows the Toothpick to be controlled by a host processor.
  - MPLAB C18 can be used to develop applications. This allows the developer to take advantage of the Toothpick Services provided by FlexiPanel Limited.
  - Hi-Tech, CCS or any other suitable microcontroller development system can develop applications although the Toothpick Services would no longer be available.

- Power Consumption

The Toothpick 2.1's power consumption is largely dominated by the LinkMatik Bluetooth radio. This device's current consumption is 55mA during connection and device discovery. The Toothpick may be powered with a 3V3–5V regulated input on the Vdd pin. Alternatively an unregulated input between 3V3 and 10V may be applied to the Vin pin where it will be regulated by a 400mA regulator. In this case the Vdd pin will act as a 5V regulated power source for other external circuitry.

- Features and peripherals
    - The toothpick's features and peripherals are:
    - FCC / CE certified Class 1 Bluetooth V2.0 radio, 100m range, integral antenna
    - 128Kbyte Flash, 3.5K RAM, 1K EPROM up to 512Kbyte I2C external memory
    - 12× 10-bit A to D converter
    - 5 × 10-bit PWM outputs
    - Serial UART, I2C and SPI communications
    - 2 interrupts
    - PCM audio features anticipated
    - 24MHz and 32KHz oscillators
    - Low dropout 400mA power regulator
    - 51 x 22 mm through-hole and surface mounts
    - Low cost "Lite" version available

Microchip's MPLAB C18 was used to write the Toothpick's firmware. Their ICD2 was used to develop and test the firmware. Wireless Field Programming was used to program the Toothpick wirelessly, for minor firmware updates.

6.3.2 The Bluetooth Module

The Toothpick includes a LinkMatik Bluetooth Radio. The LinkMatik Bluetooth Radio is also available as a separate product from FlexiPanel Limited. The Bluetooth Radio has regulatory approval for use in the United States of America (USA) and certain European countries. This is described in the LinkMatik documentation. These regulatory points are:

- Approval applies only if the existing, unmodified integral antenna is used.
- The exterior of the product should be marked as follows: Contains Transmitter Module FCC ID: QOQWT11
- The CE (European Community/Conformity) mark on the module indicates that it does not require further Radio and Telecommunications Terminal Equipment (R&TTE) certification.

The LinkMatik Bluetooth Radio is a 2.4GHz Class I Bluetooth device with an integral antenna. In order to achieve 100m range the corresponding Bluetooth device must also be Class I.

6.3.3 The Sensor Board

The sensor board connects the sensors to the Microcontroller. A maximum of two sensors can be connected to a single sensor board. The sensor board contains the following devices and their necessary circuitry:

- An AN221E04 FPAA (Field Programmable Analog Array)
- Sensor Driver
- MCP3301 SPI device 13-Bit Differential Input, Low Power A/D Converters with SPI™ Serial Interface

The AN221E04 device features an advanced input/output structure that allows the FPAA to be programmed with up to six outputs. A single AN221E04 can now be used to process multiple channels of analog signals. With dynamic reconfigurability, the functionality of the AN221E04 can be reconfigured in-system or on-the-fly in order to adapt or maintain precision operation despite system degradation and aging. With analog functions defined by software, the analog subsystem can be controlled using C-code automatically generated by AnadigmDesigner®2. Configuration data is stored in an on-chip SRAM configuration memory.

The Sensor Driver circuit interfaces the actual sensor to the AN221E04. An E-type thermocouple was used as the actual sensor.

The MCP3301 A/D converter features full differential inputs and low power consumption in a small package which is ideal for battery-powered systems and remote data acquisition applications. The AN221E04 outputs the measured signal to the MCP3301 which in-turn converts this analog voltage to a digital value. The microcontroller reads this value via the SPI communication protocol.

6.3.4 The Power Supply

To power the experimental setup, four rechargeable 1.4V NiMH batteries were chosen. These batteries are suitable for mobile digital electronic devices like mp3 players and digital cameras. They were connected in series, giving a total voltage of 5.6V. The batteries were then connected to the Toothpick's unregulated voltage input. This input allows voltages in the range 3.2V - 10V to be connected to it. LinkMatik is separately regulated to operate at 3V3.

All the various components were mounted on a machined piece of Teflon plastic base. A stainless steel plate was used to cover the components and to fasten them to the spinning chuck. The chuck contained the tool (an FSW tool in this case) with a thermocouple at the tip. The batteries were bolted on top of the Teflon plastic disc. See Figure 28 below.

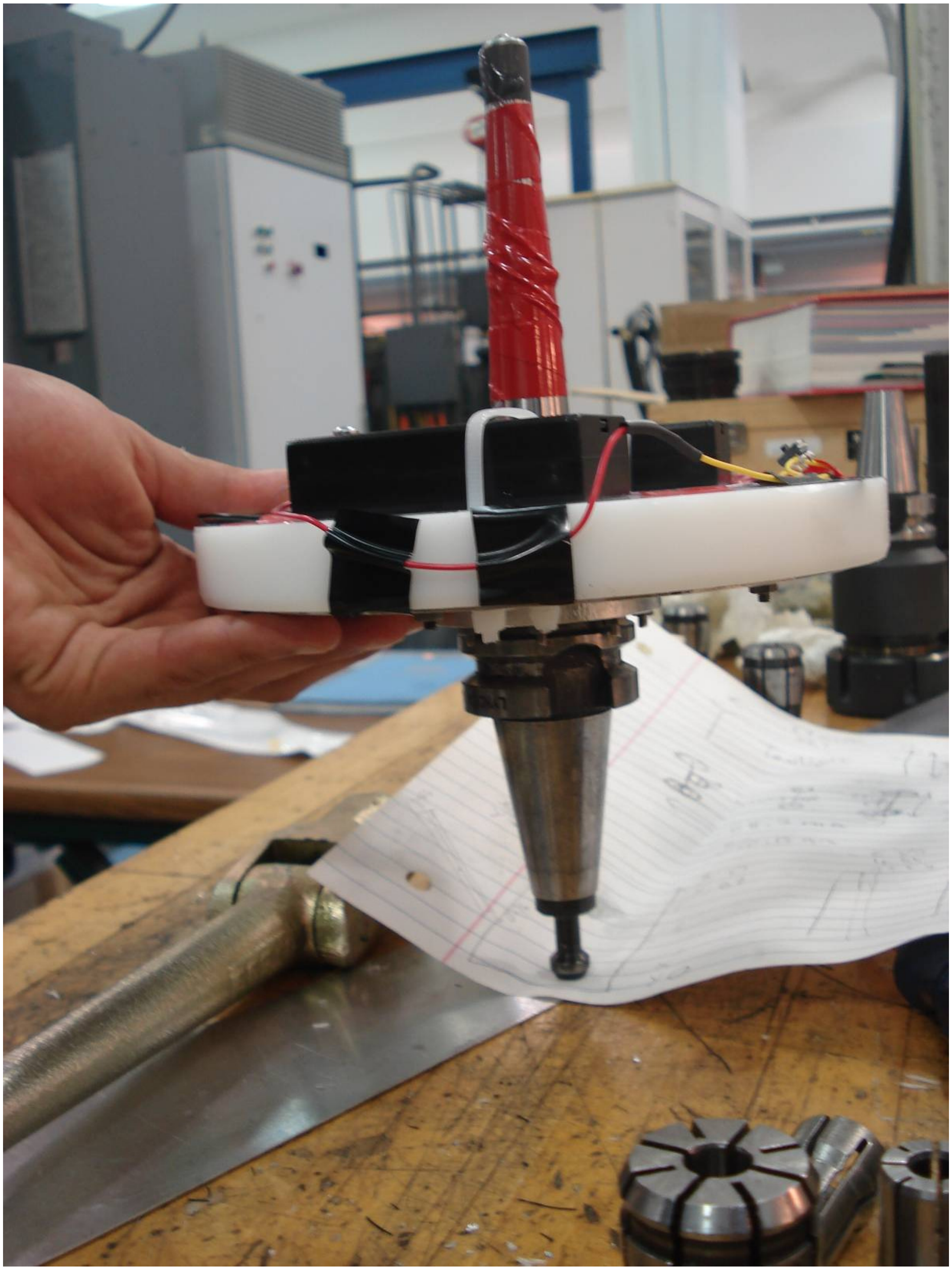


Figure 28: Side view of the experimental setup

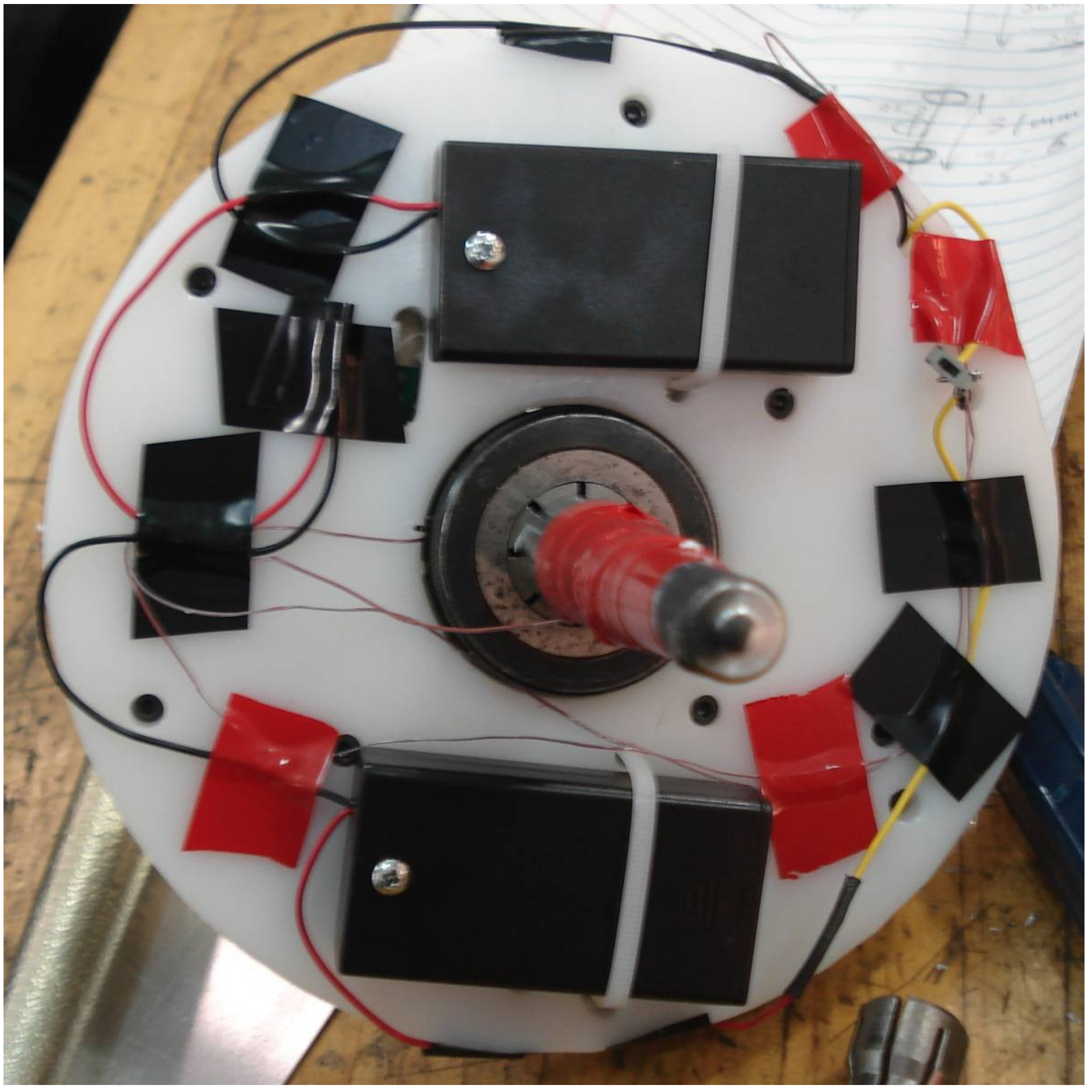


Figure 29: Top view of experimental setup

6.4 THE INDUSTRIAL MACHINE USED

A Mori Seiki TV-30 was used to spin the experimental setup together with the tool. The tool is clamped with a chuck which is compatible with the machine. The experimental setup was fastened to the chuck.

Figure 30: The Mori Seiki TV-30 Milling machine

Figure 31: The experimental setup inside the Mori Seiki TV-30

6.5 THE PC HARDWARE USED

The PC is used to connect with the spinning experimental setup. Commands, controlling the experimental setup are sent using a GUI. All data streamed from the experimental setup is logged into a text file. This section will describe the standard PC components which were used to communicate with the experimental setup.

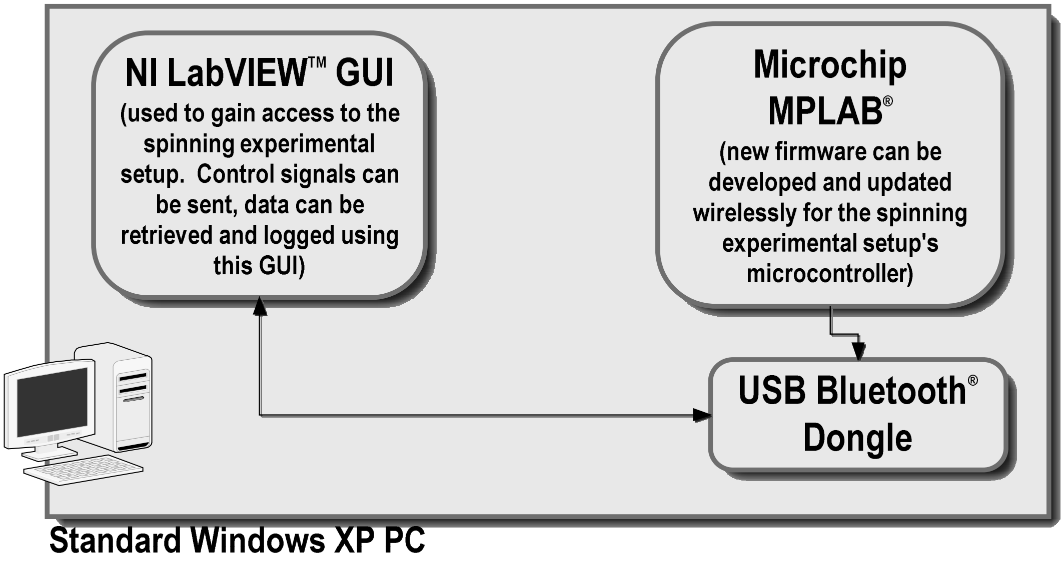


Figure 32: The PC connects and communicates with the experimental setup

6.5.1 The USB Bluetooth Dongles

Various USB Bluetooth dongles were purchased. These dongles allow the PC to connect wirelessly via Bluetooth to the Toothpick 2.1. Data can then be streamed and logged on the PC.

LabVIEW was chosen to log all data received by the PC. In order for USB Bluetooth devices to be used in LabVIEW, the device must use Window's Bluetooth stack. Windows XP would not install the Windows Bluetooth stack for most of the purchased Bluetooth USB devices. These devices required their own device drivers (which came on a CD) to be installed. A work-around was found on the National Instruments' web-site in order to manually install the Windows Bluetooth stack for these USB Bluetooth devices. The site address is:

http://digital.ni.com/public.nsf/allkb/8295C04C0A038E8686257500005CCA99

6.5.2 The PC

The PC which was used for communicating with the experimental setup and to log the data had a Pentium 4 processor with 1024 MB RAM. Windows XP was chosen to be the operating system as most modern PCs is capable of running Win XP. Thus, Win XP was a generic choice. LabVIEW 8.6 was used to develop the software for the PC. A GUI was developed, which enabled the user to:

- Send commands to the Toothpick, commanding the Toothpick which transmission method to use and which data to transmit; and
- Receive the data from the Toothpick, which was logged and shown on the GUI's graph.

Excel 2003 was used to analyze the data and to create visual representations of the data in tables and graphs.

Figure 33: The PC with the experimental setup

CHAPTER 7

THE EXPERIMENTAL SETUP: SOFTWARE

7.1 INTRODUCTION

Software was developed for the PC and for the microcontroller (Toothpick). The task of the PC is to:

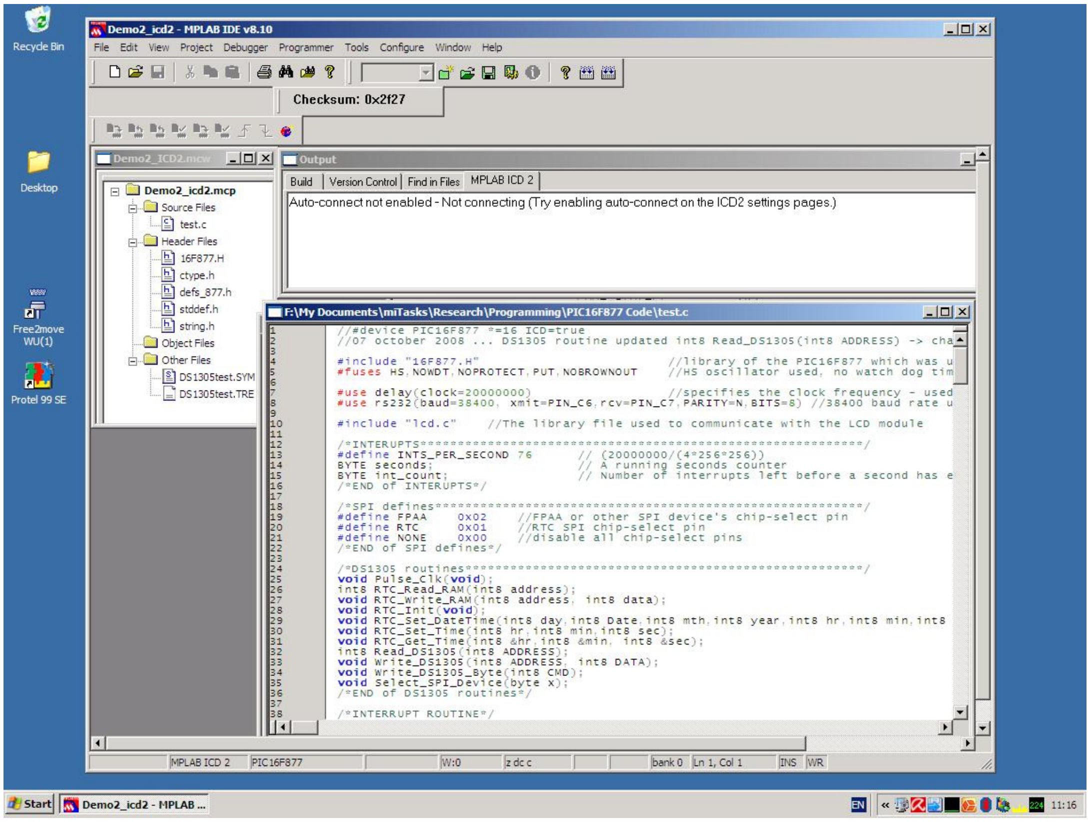
1. Send a command to the microcontroller which informs the microcontroller which transfer method should be used.
2. To receive the streamed data.

The microcontroller needs to:

1. Receive the command sent from the PC and to interpret it.
2. Stream the data to the PC.

7.2 PROGRAMMING THE TOOTHPICK

Microchip's software suite, MPLAB IDE V8.30, was used to develop the software for the Toothpick. MPLAB is an Integrated Development Environment (IDE) software package, created by Microchip. It combines all of the necessary tools required by embedded software developers and can be downloaded freely from www.microchip.com. Microchip's MPLAB C18 was used to compile the code.



Auto-connect not enabled - Not connecting (Try enabling auto-connect on the ICD2 settings pages.)

```
//#device PIC16F877 *=16 ICD=true
//07 october 2008 ... DS1305 routine updated int8 Read_DS1305(int8 ADDRESS) -> cha

#include "16F877.H"                         //library of the PIC16F877 which was u
#fuses HS,NOWDT,NOPROTECT,PUT,NOBROWNOUT    //HS oscillator used, no watch dog tim

#use delay(clock=20000000)                  //specifies the clock frequency - used
#use rs232(baud=38400, xmit=PIN_C6,rcv=PIN_C7,PARITY=N,BITS=8) //38400 baud rate u

#include "lcd.c"    //The library file used to communicate with the LCD module

/*INTERUPTS***********************************************************/
#define INTS_PER_SECOND 76      // (20000000/(4*256*256))
BYTE seconds;                   // A running seconds counter
BYTE int_count;                 // Number of interrupts left before a second has e
/*END of INTERUPTS*/

/*SPI defines*********************************************************/
#define FPAA    0x02    //FPAA or other SPI device's chip-select pin
#define RTC     0x01    //RTC SPI chip-select pin
#define NONE    0x00    //disable all chip-select pins
/*END of SPI defines*/

/*DS1305 routines*****************************************************/
void Pulse_Clk(void);
int8 RTC_Read_RAM(int8 address);
void RTC_Write_RAM(int8 address, int8 data);
void RTC_Init(void);
void RTC_Set_DateTime(int8 day,int8 Date,int8 mth,int8 year,int8 hr,int8 min,int8
void RTC_Set_Time(int8 hr,int8 min,int8 sec);
void RTC_Get_Time(int8 &hr,int8 &min, int8 &sec);
int8 Read_DS1305(int8 ADDRESS);
void Write_DS1305(int8 ADDRESS, int8 DATA);
void Write_DS1305_Byte(int8 CMD);
void Select_SPI_Device(byte x);
/*END of DS1305 routines*/

/*INTERRUPT ROUTINE*/
```

Figure 34: The MPLAB Integrated Development Environment

According to the datasheet of the Toothpick 2.1, four ways exist to develop an application for the Toothpick:

1. Using a pre-compiled Standalone Firmware Solution.
2. Using the Toothpick Slave Firmware Solution to allow Toothpick to be controlled by a host processor.
3. Developing applications in C using MPLAB C18 from Microchip Inc.
4. Developing applications using any suitable microcontroller development system such as Hi-Tech or CCS.

Firmware Solutions (including the Toothpick Slave) were inspected to distinguish the programming techniques employed to program the Toothpick. They provide suitable starting points for a variety of applications.

Microchip's MPLAB C18 was chosen to develop the application for the Toothpick. The software written for the Toothpick will be explained in the following section.

7.3 THE TOOTHPICK'S FIRMWARE

As soon as the Toothpick is applied with power, the firmware is executed. Firstly, the firmware initialises variables, clears the semaphores and sets the direction of the microcontroller's pins (input/output) used in the code. The Toothpick makes use of the following I/O definitions throughout the firmware to access its own port pins. Some pins are used to communicate with the sensor board via SPI.

7.3.1 The Compiler Directives Used

```
#define ADC              0x02
#define FPAA             0x01
#define NONE             0x00
#define FPAA_INIT_SIZE   579
```

Figure 35: Compiler Directives

ADC, FPAA, NONE: These defined directives are used in conjunction with the Select_SPI_Device function and lets the

Toothpick’s microcontroller select which SPI device it wants to communicate with.

FPAA_INIT_SIZE: This is used to tell the microcontroller how big the FPAA’s hex file is (how many bytes). The hex file is stored in the microcontroller’s ROM as a group of 8-bit hex values. This hex file is used to program the sensorboard’s FPAA with the required analog circuit. This analog circuit is designed and compiled using AnadigmDesigner®2. This software compiles the analog circuit and exports the circuit to a C file containing the hexadecimal representation of the analog circuit. See Figure 36 below.

```
//FPAA circuit****************************
rom const unsigned char analog_lvdt [FPAA_INIT_SIZE] =
{
        0x00, 0x00, 0x00, 0x00, 0x00, 0xD5, 0xB7, 0x22, 0x00, 0x80, 0x01, 0x05,
        0xCC, 0x00, 0x14, 0x20, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x78, 0x00,
        0x08, 0xFF, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x2A,
        0xCE, 0x01, 0x12, 0x01, 0x48, 0x00, 0xE0, 0x80, 0x02, 0x20, 0x00, 0x00,
        0x0E, 0x40, 0xC3, 0x00, 0x00, 0x48, 0x00, 0x48, 0x00, 0x2A, 0xC0, 0x02,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0xFA, 0x00, 0x10, 0xFA, 0x01, 0x81, 0x19,
        0x00, 0x10, 0x19, 0x01, 0x81, 0x00, 0x04, 0x01, 0x01, 0x82, 0x04, 0x00,
        0x00, 0x14, 0x01, 0x82, 0x86, 0x00, 0x20, 0x00, 0x14, 0x01, 0x82, 0x00,
        0x04, 0x00, 0x00, 0x90, 0x81, 0xAC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x98, 0x00, 0xC1, 0xAC, 0x00, 0x00, 0x00, 0x41, 0xA5, 0x41, 0xDD, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x90, 0x81, 0xAC, 0x08, 0x10, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x2A, 0x80, 0x10, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x2A, 0x00
};
```

Figure 36: A typical FPAA configuration file

7.3.2 The Variables Used and Explained

Various flags are used to indicate whether certain events occurred. Variables are used to store information which can be values or strings (groups of characters).

szHello: is a string variable constant containing the message "Hello\r\n\0". It is mainly used as a simple test to see if the Bluetooth communication link has been established.

szWorld: is a string variable constant containing the message "World\r\n\0". It is mainly used as a simple test to see if the Bluetooth communication link has been established.

*AnalogVal*_13bit : This variable is used to store the 13-bit analog value read from the 13-bit ADC.

i_loop : is used as a looping variable (for-loops). This is used to transmit the BER and RSSI of the Toothpick module continuously and for a set number of repetitions (i_loop is the variable which is used to count and monitor the number of repetitions).

time: is used to count the amount of seconds during various operations. The Toothpick's microcontroller interrupts its own operation and increments this variable after every second.

T3Time : is used to store the value of the Toothpick's microcontroller's Timer3. This value is sent during some commands to give an indication of the amount of time between consecutive transmissions. It is cleared after each transmission command and also before the start of transmission.

overflow: is used to count the amount of Time3's overflows which might have occurred between consecutive transmissions. Together with T3Time it gives an indication of the amount of time between consecutive transmissions. It is cleared after each transmission command and also before the start of transmission.

blue_char: is used to store the received character from the PC. The Toothpick's microcontroller will determine which character was received and perform the required actions.

char_rcvd: this flag is set by an interrupt and indicates that a character was received. The Toothpick's microcontroller will clear this flag and then process the character (contained in char_rcvd). Depending on which character was received, the microcontroller will enter into a loop and transmit the required Bluetooth message. The different characters basically choose the format of the Bluetooth message and also what data is contained in the message.

buf: is a string variable consisting of 20 characters. It is used to format the message which will be transmitted via Bluetooth.

blinkBER,blinkRSSI: these variables are both string variable consisting of 20 characters. They contain the value of BER and RSS respectively.

getBER, : is a flag and it indicates that blinkBER was updated and contains the latest value of the Toothpick's BER value. The Toothpick's microcontroller requests the BER value from the Toothpick's Bluetooth module. Once the value of BER is received an interrupt occurs. The latest value of BER is then written into the blinkBER string variable and getBER is set. The microcontroller actually waits (pauses the program) until getBER is set before continuing to run the program. Once set, the program will be executed as normal (from where it was paused). getBER is cleared each time before requesting BER from the Bluetooth module.

getRSSI : is a flag and it indicates that blinkRSSI was updated and contains the latest value of the Toothpick's RSSI value. The Toothpick's microcontroller requests the RSSI value from the Toothpick's Bluetooth module. Once the value of RSSI is received an interrupt occurs. The latest value of RSSI is then written into the blinkRSSI string variable and getRSSI is set. The microcontroller actually waits (pauses the program) until getRSSI is set before continuing to run the program. Once set, the program will be executed as normal (from where it was paused). getRSSI is cleared each time before requesting RSSI from the Bluetooth module.

BlueConnected: is set once the Toothpick's Bluetooth module is connected with another Bluetooth device.

7.3.3 The Firmware Explained

When data is received, from the remote device, it is put in the receive buffer and then the LowInterrupt callback is called with the flag SWI_LMTData set. Once this interrupt is executed the character will be placed in the blue_char variable and the char_rcvd flag will be set. The character will then be removed from the receive buffer using the command LMTRxAdvanceCh.

The main program monitors two flags:

- *Blueconnected flag*: This flag uses the Toothpick's green LED flash to indicate the Bluetooth module's connection status. Longish green LED flashes indicate that the Bluetooth module is connected (an active link exists between the Toothpick and another Bluetooth device). Brief green-LED flashes indicate that no active Bluetooth link exists.

- *char_rcvd flag*: this flag is monitored to determine whether a character was received. Once it detects that a character was received it will clear the flag and then process the character. Each valid character received has a corresponding action associated with it. These actions basically determine how the Toothpick will transmit data and what the transmitted data is. See the explanation of the LMTTransmit Service in the following paragraph. This service is used to transmit data. It has several parameters which determine how data is sent.

The LMTTransmit Service is used to transmit serial data. The microcontroller waits until all the data has been conveyed to the LinkMatik for transmission and then continues executing the rest of the code. The LMTTransmit service has several parameters. See figure below.

```
{
     unsigned char *pTxDataR,
     unsigned char *pTxData,
     unsigned char nBytes,
     unsigned char msTimeOut,
     unsigned char LogicalChannel
}
```

Figure 37: The various LMTTransmit Service parameters

These parameters determine how the data will be transmitted and also what data will be transmitted. If pTxDataR is non-zero, the data is sourced from that ROM address; otherwise it is sourced from the RAM address pTxData. nBytes contains the number of bytes to be transmitted, or zero if the data is a zero-terminated string. The number of bytes to be transmitted should be 100 bytes or less. LogicalChannel shall contain the logical channel (0-3) on which to transmit the data. If the LinkMatik flow control blocks transmission, LMTTransmit will wait for up to msTimeOut for it to clear. If at the end of that period there is still insufficient space, it will return with the value False, without adding anydata to the transmit buffer. In addition, if msTimeOut was an odd number, an ERR_TXTIMEOUT error statusevent will be generated before returning. If msTimeOut is zero, LMTTransmit will not return until transmission is complete. LinkMatik has a tendency to translate the data from each LMTTransmit command into a single Bluetooth frame. In order to achieve high data rates, is it recommended that as much data should be contained in each LMTTransmit command. Another method is to ‘batch transmit’. It is then necessary to advise the Toothpick in advance that the LMTTransmit service will be called several times in succession for a single channel. The data will be concatenated and sent in as few frames as possible. In order to do this, prior to transmitting the data, call LMTTransmit must be called once with bothpTxDataR and pTxData equal to zero and nBytes equal to the total number of bytes (max 65535) which will be transmitted sequentially. See Figure 38 below.

```
LMTTransmit( 0, 0, 3000, 0, 0 ); // expect 3000 bytes
For (i=0; i<1500; i++)
{
        LMTTransmit( 0, pBuff1, 2, 0, 0 );
}
```

Figure 38: How to use batch transmission

When using this technique the logical channel should remain the same throughout the sequential execution of the LMTTransmit service. The number of bytes to transmit must also exactly match the number of bytes expected. If not, the ERR_BATCH_MISMATCH error will be generated.

Testing was done to determine how the different parameters affected the performance of the LMTTransmit service.

7.4 PROGRAMMING THE PC

National Instruments (NI is well known for their automated test equipment and virtual instrumentation software. Some of their well known products include LabVIEW, Multisim, PXI, VXI, VMEbus. NI uses commercially available technologies, such as industry-standard computers and the Internet, in order to deliver customer-defined measurement and automation solutions.

LabVIEW is a graphical programming environment and is used by engineers and scientists in order to develop sophisticated measurement, test, and control systems by using easy-to-learn graphical icons and wires. Whole systems can be designed by connecting these icons using these “virtual” wires. The end product resembles a flowchart. LabVIEW can be integrated with basically any hardware devices and provides useful built-in libraries for advanced analysis and data visualization.

7.5 THE PC’S SOFTWARE

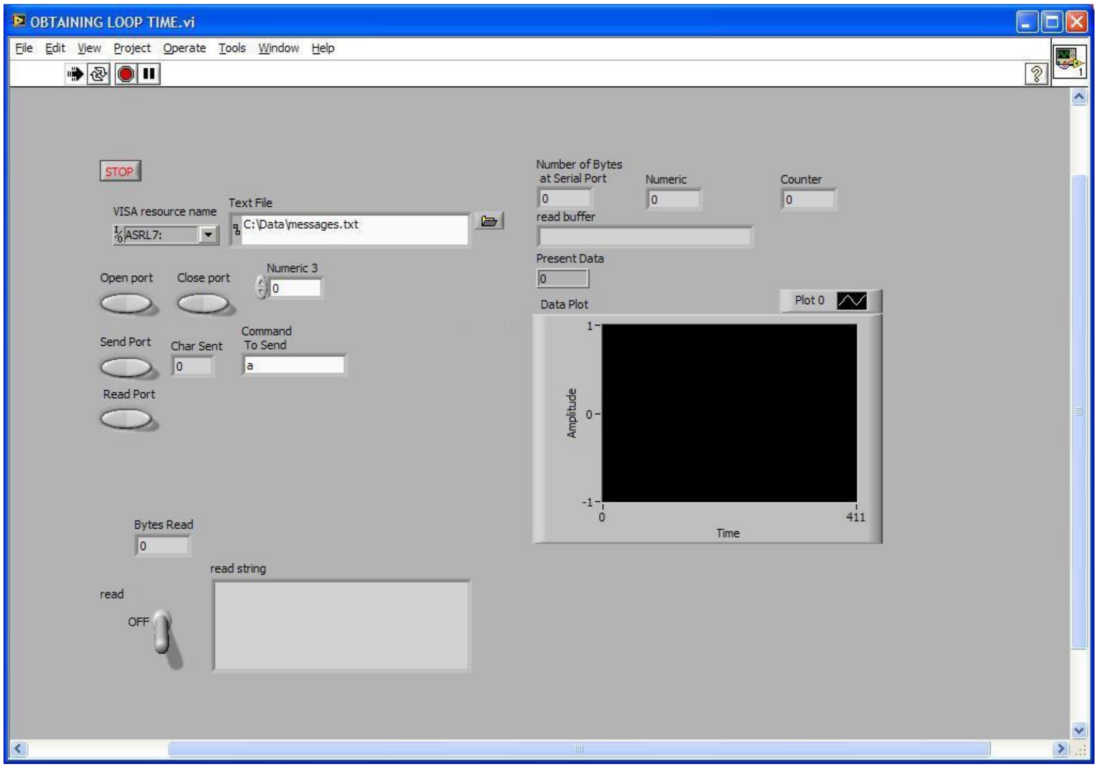
Using NI’s LabVIEW, a virtual instrument (VI) was created in order to communicate with the experimental setup. VISA, a standard I/O application programming interface (API) for instrumentation programming – which basically is a single library of functions used to communicate with serial (and other) devices – was used to control the PC’s Bluetooth hardware. Message-based communication was implemented via the Bluetooth’s serial COM port. Standard, high-level ASCII character strings were used as the communication format between the PC and the Toothpick experimental setup. This was done because the Standard Commands for Programmable Instruments (SCPI) standardizes the ASCII command strings used to program compliant instruments. Similar instruments often use similar commands. This was a generic choice.

7.5.1 The LabVIEW Virtual Instrument

NI LabVIEW was used to create a virtual instrument which could run on a standard Windows XP PC. It consists of two parts, the front panel (the GUI) and the block diagram (the programming). The front panel of the GUI allows the user easy access to:

- select the correct COM port of the Bluetooth USB dongle;
- select the baud rate of the COM port;
- select where to save the log file;
- choose a filename for the log file;
- connect to the experimental setup via Bluetooth (Open Port);
- disconnect from the experimental setup (Close Port);
- send characters and commands to the experimental setup;
- read the data manually and automatically from the Bluetooth's Rx buffer.

The GUI indicates the number of characters which was read from the Rx buffer. When the Rx buffer is read, the actual characters read, are displayed in a box. Once streaming of data commences, a graph on the GUI creates a visual representation of the temperature sensor's actual value. Below, as seen in Figure 39, is the virtual instrument's front panel.



VISA resource name: ASRL7:

Text File: C:\Data\messages.txt

Numeric 3: 0

Char Sent: 0

Command To Send: a

Bytes Read: 0

read string:

Number of Bytes at Serial Port: 0

Numeric: 0

Counter: 0

read buffer:

Present Data: 0

Figure 39: The Virtual Instrument's Front Panel

7.5.2 The Virtual Instrument Explained

Before the GUI is executed, the PC's Bluetooth dongle should be paired with the Toothpick experimental setup. The security code for the experimental setup is "0000". Once this is done, a virtual COM port will be created by the PC's Bluetooth software.

Once the GUI is "RUN", the user should select the correct VISA resource name from the drop box. This will be the COM port which was created when the PC's Bluetooth dongle was initially paired with the Toothpick experimental setup. The following steps can then be followed:

1. Enter the correct path and filename for the message log file.

2. Click “Open port”. This will cause the PC to connect to the experimental setup.
3. Enter a command to send in “Command To Send”. The commands are in single characters. This will instruct the experimental setup what data to stream, what transmission method to use, and how long the message should be.
4. Click “Send port”. This will immediately trigger the “Read port” button to be enabled, and also disable the “Send port” button. Once the experimental setup receives the command, it will start streaming the required data and the PC will log the data in the specified filename in “Text file”. Data will be stored in plain text. Data is streamed and stored in the Bluetooth’s Rx buffer, once there are 23 characters stored in it, the application will read 23 characters from the buffer and then write those characters to the log file. The sensor’s data will be displayed on the Graph. The number of bytes read successfully will be displayed above the graph.

The “Read” toggle switch is used for troubleshooting and to manually clear the Bluetooth’s Rx buffer.

Below as seen in Figure 40, is the Block Diagram of the GUI. It shows the different icons representing libraries, commands and functions, which were used to create the inner-workings of the GUI. It looks similar to ordinary flow charts and is easily understandable.

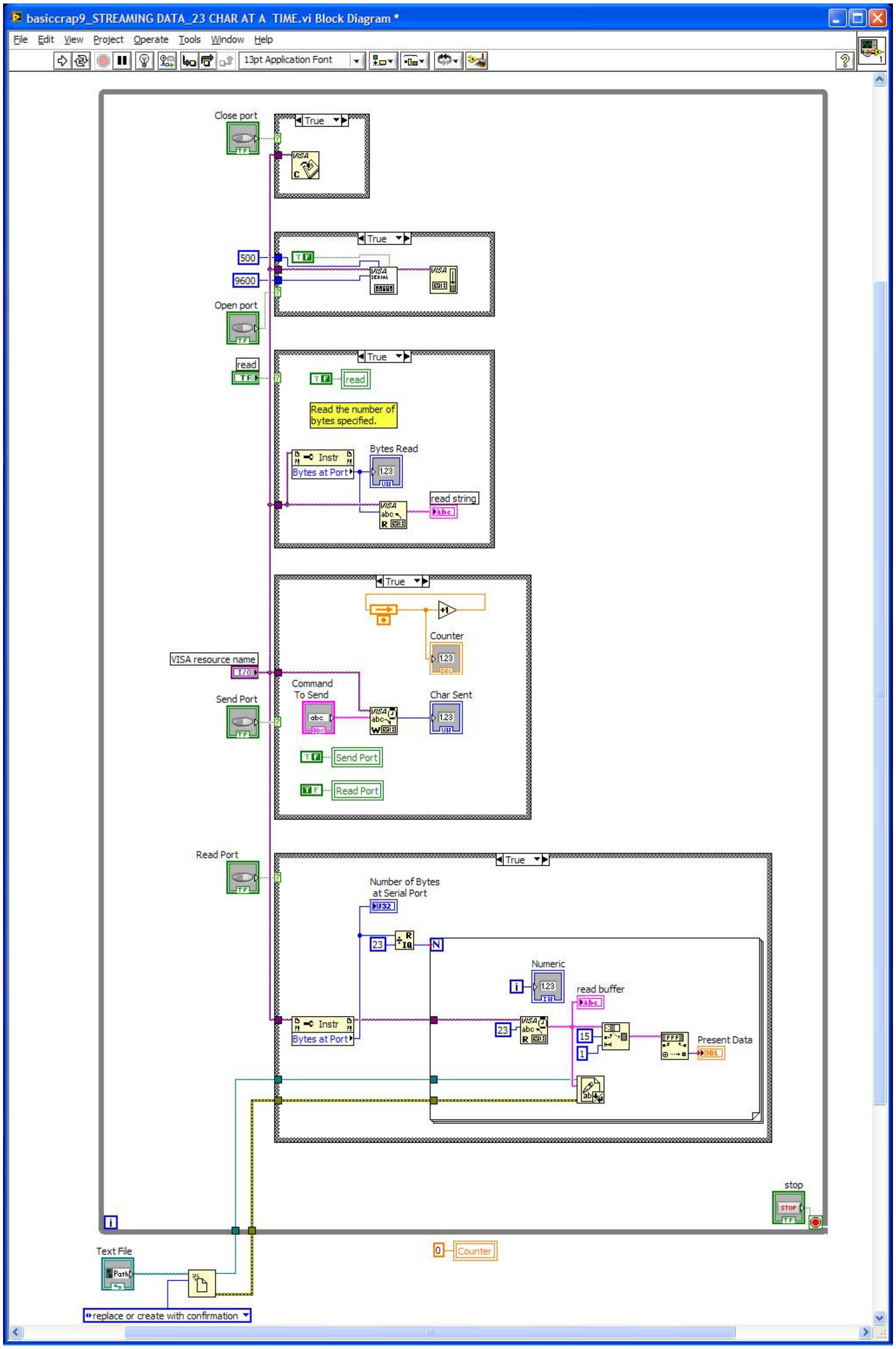


Figure 40: The Virtual Instrument's Block Diagram

Various data was streamed from the experimental setup to the PC via Bluetooth, using various fixed message lengths and transmission methods. The streamed data was stored in ordinary text files. The equals sign or ‘=’ characters were used to create a separation between the different variables sent. Every new line contains a new message. Using Microsoft Excel, the data was analysed and placed into visual graphs as seen below in Figure 41.

```
0=8df0=  24=0.0075
1=85c2=  25=0.0050
2=9b2a=  25=0.0050
3=f1fd=  24=0.0025
4=df65=  25=0.0025
```

Figure 41: How the data was stored in the text file

The most important data which was sent was the hardware timers of the Toothpick. Using these timers, it was possible to determine the time it took to transmit the streaming data. Different transmission methods generally gave different transmission times.

7.6 CONCLUSION

Chapter 7explained the firmware which was written for the Toothpick experimental setup, as well as the software designed for the PC. All the capabilities of the firmware and software were listed. It was also demonstrated how an End-User will use the PC’s GUI to gain information from the spindle.

In chapter 8 an attempt is made to explain how the tests were conducted, using code extracts from the embedded code and the communication steps between the PC and the experimental setup.

CHAPTER 8

THE EXPERIMENTS

This chapter describes the different experiments which were performed. Code extracts and the communication steps between the PC and the experimental setup are explained. The results are also shown.

Two types of experiments were carried out:

1. Performance Tests
2. Physical Tests

8.1 THE PERFORMANCE TESTS

The performance tests were simple, but essential, preliminary tests. They were used to find the transmission method which gave the best performance.

8.1.1 The Attributes of the Performance Tests

The performance tests experimented with different methods of transmitting data (from the Toothpick to the PC).Each test used the method of the PC initiating the communication loop by transmitting an ‘A’ character. This would initiate the timers on the Toothpick experimental setup and it would then respond with either a single or multiple characters. This is where each of the tests is different. All tests end with the Toothpick experimental setup eventually transmitting the value of its hardware interrupt timers giving an indication of the response of the communication loop. The Toothpick experimental setup was left motionless (static and not spinning in any way) and within 20cm from the PC's Bluetooth antennae. The Toothpick's performance was tested by transmitting one character from the PC, and after receiving the character, the Toothpick would transmit a certain fixed amount of characters back to the PC.

8.1.2 The Aims of the Performance Tests

To determine which transfer method and which parameters of the LMTTransmit service gave the highest performance i.e. highest character throughput, fastest executing and transmitting speed. Thus the optimum software solution which gave the highest hardware performance level had to be determined.

8.1.3 Details of the Performance Tests and Results

These tests (1, 2, 3, 4 and 5 seen below)measure the time it takes for the Toothpick to respond with either a single character or a certain number of characters, after receiving a single character from the PC. These tests form the baseline tests. The experimental setup is not gyrated (spun) at all during the course of these tests.

Each test has four consecutive steps. These steps are repeated a certain number of times in order to see how the Toothpick performs over time (the consistency of the Toothpick's performance is tested). The steps outlined occur as follows:

1. The test starts with the PC sending a single character 'A'. This character is used to initiate the transmission test and is called the *initiate* character.
2. After receiving the character, the Toothpick
   a. Clears its WriteTimer3 and overflow variables. These variables are the microcontroller's hardware timers and are used basically as a stopwatch. It measures the time required to transmit a fixed amount of data from the Toothpick to the PC. Thus the timing will start immediately after clearing these variables.
   b. Responds with either a single character ('B') or a certain predefined number of characters (for example'12345Y'). In all cases a certain *termination* character is sent, signalling the end of the string of characters. This can be a 'B' (when transmitting a single character) or a 'Y' (when transmitting multiple characters).
3. After receiving the termination character, the PC will respond with a single character ('C') as fast as possible.
4. Once the Toothpick receives this character, it will respond with a 'D', and then with the selected variables for example WriteTimer3, overflow, i_loop, or AnalogVal_13bit etc. The PC will receive the 'D' as well as the variables, and it will log the data in a data file.

The transmission time of the system is measured between steps 2 and 4. Performing these tests over a period of time gives an indication of the consistency of the system's performance and transmission time. The minor differences between these tests and their results will be explained below as well as the results obtained.

8.1.3.1 Test 1

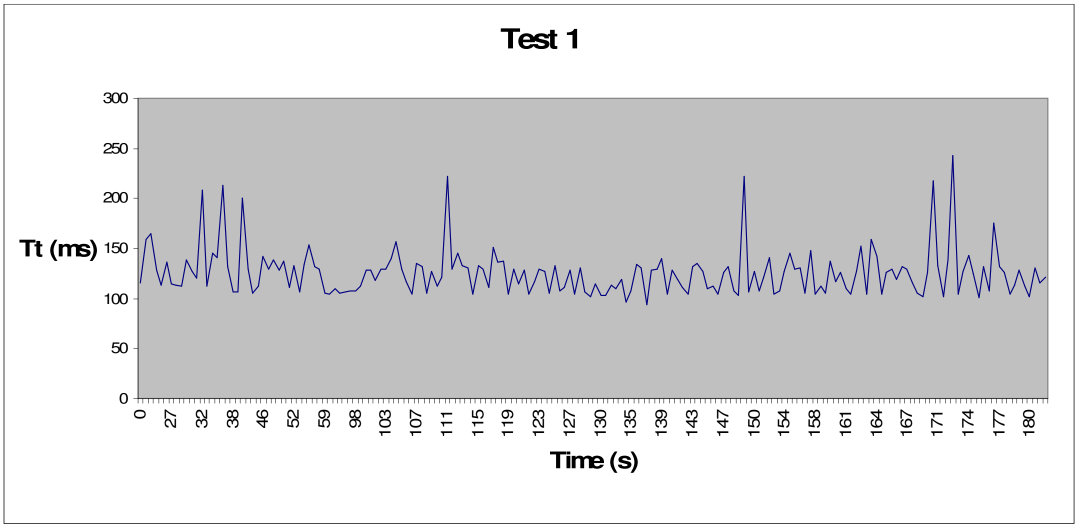
This test measures the transmission time of the system when a single character is sent. The Toothpick responds with a single 'B' character after receiving the initiate character 'A' from the PC. The communications steps are summed up in the following points:

1. PC sends 'A' (initiate the test)
2. Toothpick receives the 'A' character, clears the hardware timers and transmits 'B' (the termination character – signalling the end of transmission)
3. PC receives the 'B' character, and transmits 'C' as fast as possible.
4. Toothpick receives the 'C' character, captures the values of the hardware timers, and transmits 'D' and the hardware timer values to the PC in the following format: Time=T3Time=overflow=test_no. The '=' characters is used to separate the different variables.

See the figure below for the coding of Test 1.

```
case 'A':
WriteTimer3 (0);
test_no++;
overflow = 0;
sprintf (buf, "B\0");
LMTTransmit( 0,(unsigned char *) buf, 1, 0, 0 );
break;

case 'C':
T3Time = ReadTimer3();
sprintf (buf, "D=");
LMTTransmit( 0,(unsigned char *) buf, 2, 0, 0 );
sprintf (buf, "%4x=%4x=%4x=%6x\n\r\0", Time,
   T3Time,overflow , test_no);
LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
overflow = 0;
break;
```

The transmission time (Tt), which was recorded by the Toothpick was logged by the PC and stored in a data file. The figure below shows the transmission time over a period of time.



Average Tt (ms)	125.6087
Minimum Tt (ms)	94.197
Maximum Tt (ms)	242.5152

8.1.3.2 Test 2

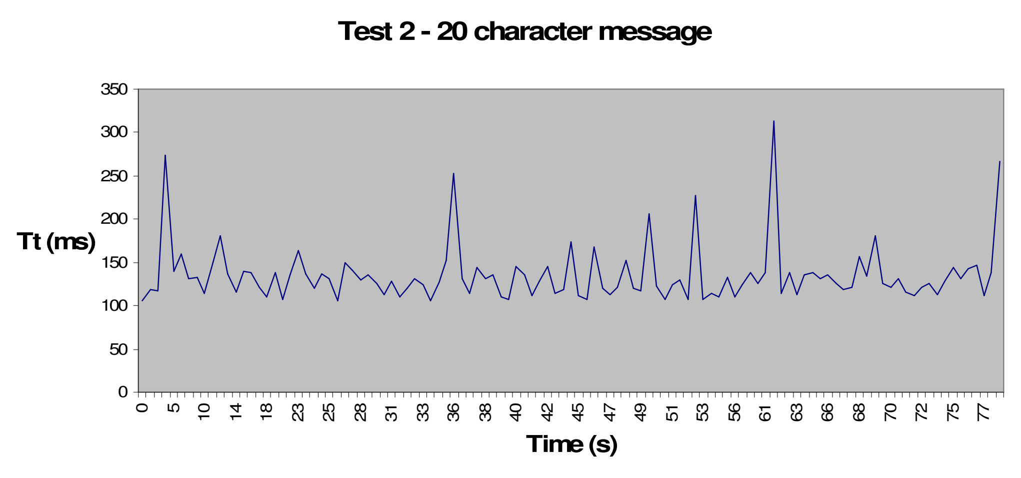
This test consists of three sub-tests and measures the time to transmit 20, 100 and 200 characters (each one of these form one of the subtests) from the Toothpick to the PC after receiving the initiate character ‘A’ from the PC. The communication steps are summed up as follows:

1. PC sends ‘A’ (initiate the test)
2. Toothpick receives the ‘A’ character, clears the hardware timers and transmits 20, 100 or 200 characters, ending with “\n\rY”.
3. PC receives the whole message and when the ‘Y’ character is received, ‘C’ is transmitted as fast as possible.
4. Toothpick receives the ’C’ character, captures the values of the hardware timers, transmits ‘D’ and the hardware timer values to the PC.

See the figure below for the coding of Test 2 (see that it shows the case when 200 characters is transmitted).

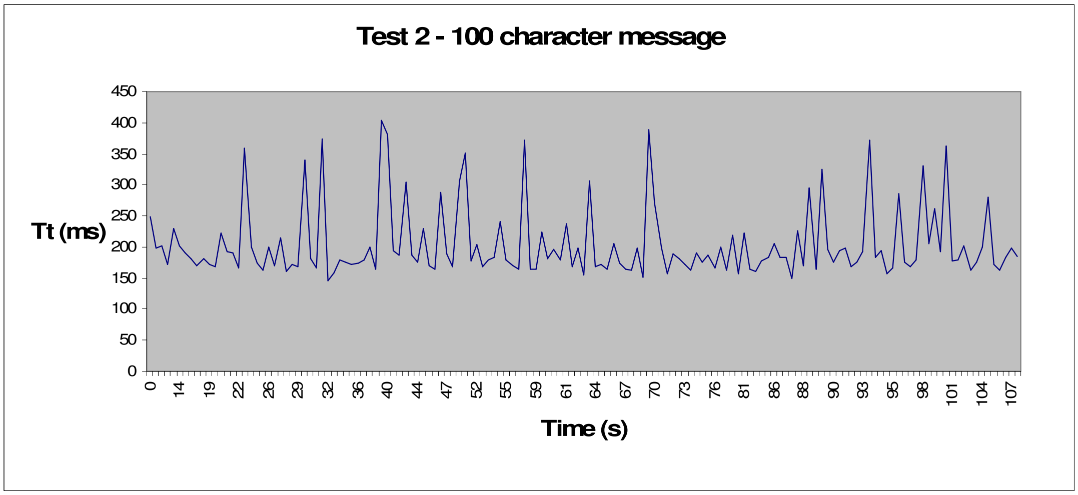
```
case 'A':
WriteTimer3 (0);
test_no++;
overflow = 0;
sprintf (buf,
   "123456789012345678901234567890123456789012345678901234
   56789012345678901234567890123456789012345678901234567890123456789
   0123456789012345678901234567890123456789012345678901234
   5678901234567890123456789012345 67\n\rY\0");
   LMTTransmit( 0,(unsigned char *) buf, 200, 0, 0 );
break;

case 'C':
   T3Time = ReadTimer3();
   sprintf (buf, "D=\n\r\0");
   LMTTransmit( 0,(unsigned char *) buf, 2, 0, 0 );
   sprintf (buf, "%4x=%4x=%4x=%6x\n\r\0", Time, T3Time,
   overflow , test_no); //WRITES LOOPTIME TO LABVIEW
   LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
   overflow = 0;
break;
```

By comparing the different sub-tests of Test 2, the performance of the Toothpick can be determined.

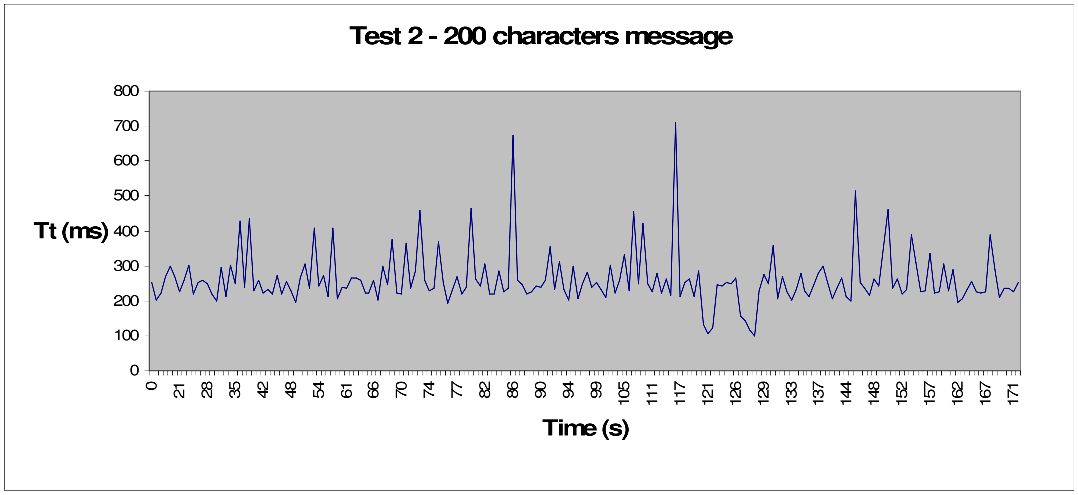


Average Tt (ms)	135.4046

Minimum Tt (ms)	105.1777
Maximum Tt (ms)	312.8115



Average Tt (ms)	260.9002
Minimum Tt (ms)	100.036
Maximum Tt (ms)	710.4293



Average Tt (ms)	260.9002
Minimum Tt (ms)	100.036

Maximum Tt (ms)	710.4293

8.1.3.3 Test 3

This test uses the batch transmission method to transmit 200 characters from the Toothpick to the PC after receiving the initiate character ‘A’ from the PC. By comparing Test 3 with Test 2 it can be determined which transfer method provides the best performance.

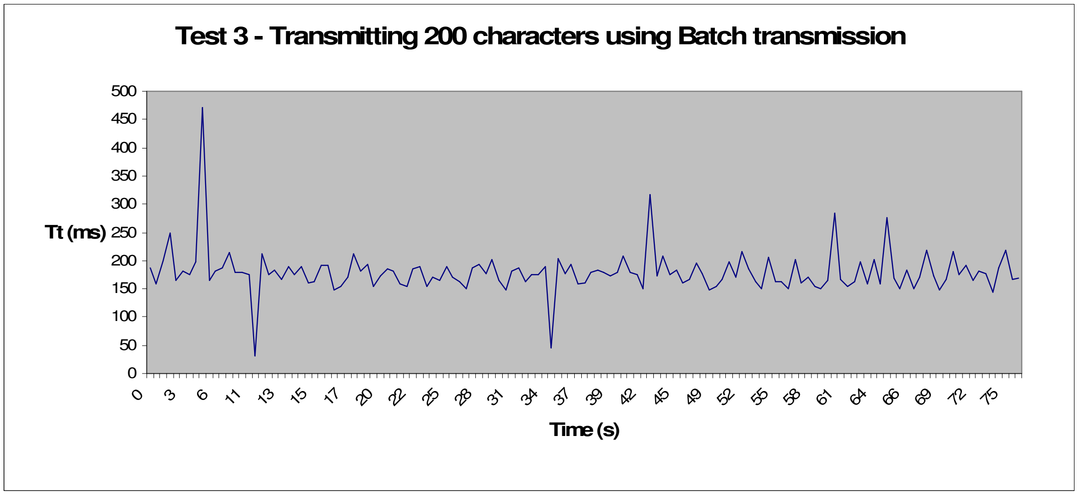
A summing up of the communications steps follow:

1. PC sends ‘A’ to the toothpick;
2. Toothpick receives the ‘A’ character, clears the hardware timers and transmits 200 characters, ending with “\n\rY”;
3. PC receives the whole message and as soon as it receives the ‘Y’ character, transmits ‘C’;
4. Toothpick receives the ‘C’ character, captures the values of the hardware timers, and transmits ‘D’ and the hardware timer values to the PC.

See the figure below for the coding of Test 3.

```
case 'A':
   WriteTimer3 (0);
   test_no++;
   overflow = 0;
   AN2Pin = 1;  //AN2 GOES ON
   LMTTransmit( 0, 0, 201, 0, 0 );//batch transmission
   cmdsprintf (buf,
   "123456789012345678901234567890123456789012345678901234
   5678901234567890123456789012345678901234567890123456789
   0123456789012345678901234567890123456789012345678901234
   567890123456789012345678901234567\n\rY\0");
   for (i=0; i<201; i++)
   {
      LMTTransmit( 0,(unsigned char *)buf+i, 1, 0, 0 );
   }

break;

case 'C':
   T3Time = ReadTimer3();
   AN2Pin = 0;  //AN2 GOES OFF
   sprintf (buf, "D=\n\r\0");
   LMTTransmit( 0,(unsigned char *) buf, 2, 0, 0 );
   sprintf (buf, "%4x=%4x=%4x=%6x\n\r\0", Time, T3Time,
   overflow , test_no);
   AwaitLMTComplete();
   LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
   overflow = 0;
break;
```

This test gives an indication of the transmission time of the system when sending 200 characters using the batch transmission method.


Test 3 - Transmitting 200 characters using Batch transmission

Average Tt (ms)	179.3378
Minimum Tt (ms)	30.42683
Maximum Tt (ms)	471.2725

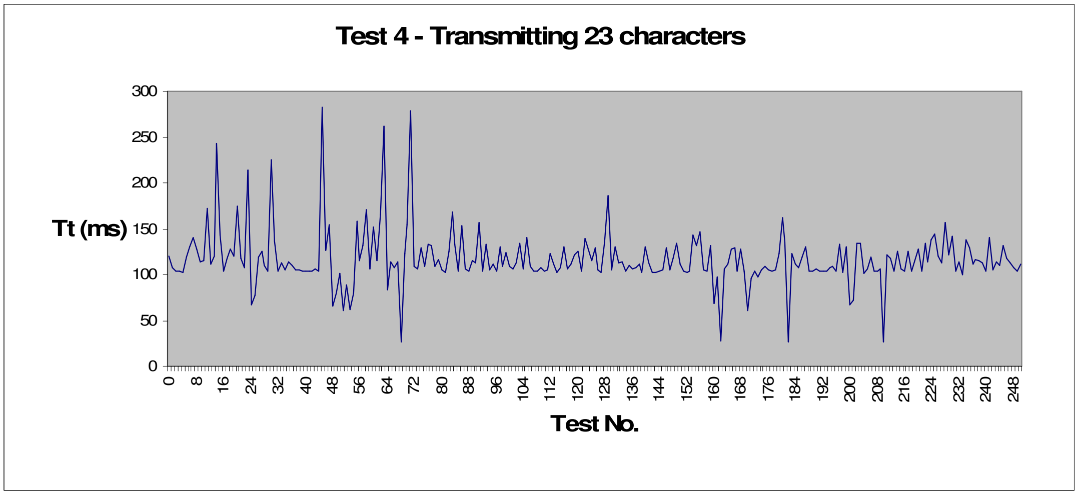
8.1.3.4 Tests 4 and 5

These tests measure the time taken to transmit 23 characters (test 4) or 103 characters (test 5) from the Toothpick to the PC. These tests start with the PC sending the initiate character ‘A’. These tests differ from the previous tests mainly from the use of the LMTTransmit service. Different parameters are used for the LMTTransmit service in order to transmit the data. Here is the communication steps summed up:

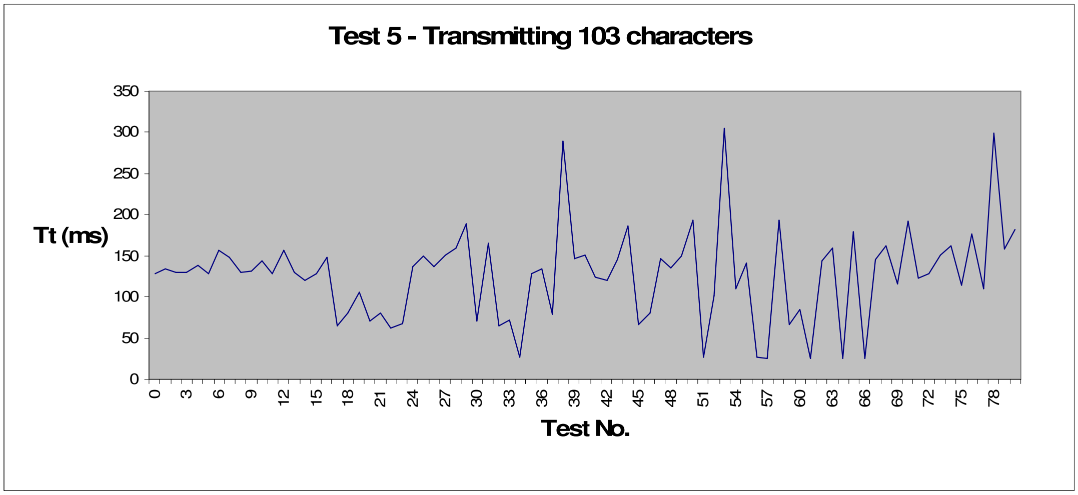
1. PC sends ‘A’ to the Toothpick;
2. Toothpick receives it, clears the hardware timers and transmits 23 characters, ending with “Y\n\r\”;
3. PC receives it, and as soon as it receives the last character, transmits ‘C’;
4. Toothpick receives it, captures the values of the hardware timers, and transmits ‘D’ and the hardware timer values to the PC.

See the figure hereunder for the coding of Test 4 followed by the test results.

```
case 'A':
   WriteTimer3 (0);
   test_no++;
   overflow = 0;
   sprintf (buf, "12345678901234567890Y\n\r\0");
   LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
break;

case 'C':
   T3Time = ReadTimer3();
   Time = 0;
   sprintf (buf, "D=%2x=%4x=%4x=%6x\n\r\0", Time, T3Time,
   overflow , test_no);
   LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
   overflow = 0;
break;
```



Average Tt (ms)	116.7836
Minimum Tt (ms)	26.0725
Maximum Tt (ms)	282.6138



Average Tt (ms)	127.1239
Minimum Tt (ms)	25.38467
Maximum Tt (ms)	304.9162

8.2 THE PHYSICAL TESTS

The physical tests were used to see how the spinning environment affected the communication link. Each test comprises six sub-tests (cases), which tests different attributes which might affect the communication link, and was performed at a higher RPM. The RPM was gradually increased in steps.

8.2.1 The Attributes of the Physical Tests

All the Physical tests shared the following attributes:

- They used the same sub-tests. (See Case ‘C’, Case ‘F’, Case ‘K’, Case ‘L’, Case ‘X’ and Case ‘Z’).
- The Toothpick experimental setup was securely mounted inside a Morei Seiki milling machine on a FSW chuck.

- The angle between the fixed PC's Bluetooth antennae and the rotating experimental setup was changed during some of the tests, that is the PC's Bluetooth antennae was moved from directly below the spinning experimental setup to directly next to the spinning experimental setup.
- The rotational speed was increased with fixed steps for each test. Two of the tests required that the rotational speed increase (ramp up) during the test.
- The Toothpick's performance was tested for one character received (from the PC) and a fixed number of characters which were sent back (to the PC).

8.2.2 The Aims of the Physical Tests

The aims of the physical tests are to determine how the performance of the Toothpick is affected by changing the physical conditions of the communication link. The Toothpick experimental setup was spun at various revolutions per minute (RPM) and the angle between the Toothpick experimental setup and the PC's Bluetooth antenna was changed during some of the tests. The BER and RSSI were logged in order to see how the quality of the Bluetooth link was affected. The logging of the hardware timers was also done in order to determine if there was any noticeable decrease in the Toothpick's performance (character throughput and transmitting speed).

8.2.3 Explanation of the Physical Tests

These tests were performed with the Toothpick experimental setup securely fastened to a FSW chuck in a Morei Seiki milling machine. In Table 1 below, the different tests are described.

Table 1: Description of the different Physical Tests:

Test 6:	This test was performed with the chuck stationary but not rotating. It is a static test and forms the basis of comparison for all the other tests. The position of the Bluetooth antennae was changed between 0° (directly below the chuck) and 90° (directly next to the chuck) in order to determine whether the Bluetooth link quality was affected by the position of the antennae.
Test 7:	This test was performed with the chuck rotating at 100 RPM and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 8:	This test was performed with the chuck rotating at 200 RPM and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 9:	This test was performed with the chuck rotating at 500 RPM and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 10:	This test was performed with the chuck rotating at 1000 RPM and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 11:	This test was performed with the chuck rotating at 1400 RPM and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 12:	This test was performed with the chuck rotating at 500 RPM increasing until the Bluetooth link disconnected and with the PC's Bluetooth antennae at 0° (or directly below the chuck).
Test 13:	This test was performed with the chuck rotating at 500 RPM increasing until it was spinning at 1500 RPM. The PC's Bluetooth antennae were moved the whole time. This was done to determine if the position of the antennae would affect the Bluetooth link quality.

For each test, the same sub-tests were performed as described below. In all cases the PC initiates the communication by transmitting a character (representing a certain sub-test) to the Toothpick system. Once received, the Toothpick will initiate the required sub-test. After completing the sub-test the PC sends the character which will initiate the next sub-test. This process is repeated until all the sub-tests have been performed for each test case. The different characters used to initiate the sub-tests are described in the following sections.

8.2.3.1 Case ‘C’:

This subtest reads and transfers the ADC value of the onboard Sensorboard. Its coding is almost the same as Case ‘Z’. The subtest differs mainly in the usage of the batch transmission command. A software loop (160 loops) is used to transmit data continuously. During each loop, the batch transmission command will be executed before each normal execution of LMTTransmit command. (During the execution of Case ‘Z’ the batch transmission command is used only once, before entering the loop)By comparing the results of Case ‘C’ with the results of Case ‘Z’ we can determine which communication method has the highest performance.

Table 2: Frimware for Case 'C'

```
case 'C':
  WriteTimer3 (0);
  Time = 0;
overflow = 0;
for (i_loop=0; i_loop<160; i_loop++)
  {
     AN2Pin = ~AN2Pin;
     AnalogVal_13bit = read_adc();
     T3Time = ReadTimer3();
     LMTTransmit( 0,0, 21, 0, 0 ); //expect 21 characters

     sprintf (buf, "%2x=%4x=%4x=%4x\n\r\0this is junk",
     i_loop, T3Time, overflow, AnalogVal_13bit);

     WriteTimer3 (0);
     overflow = 0;
     LMTTransmit( 0,(unsigned char *) buf, 21, 0, 0 );
}
```

This code transmits 21 characters continuously 160 times. Data is transmitted in the following format: *i_loop=T3Time=overflow=AnalogVal_13bit.* The '=' character is used to separate the values of the different variables. Each line is terminated with the '\n' and '\r' characters in order to move the data logging to the beginning of a new line (in the data file). The batch transmit method is used to transmit the data. Before each transmission, the LMTTransmit service is called with the parameters (0, 0, 21, 0, 0), telling it to expect 21 characters. This is done before each normal execution of the LMTTransmit service see Chapter 7, sub heading 7.3.3 The Firmware Explained). Thus within each loop (of the 160 loops) a batch transmission command is called.

8.2.3.2 Case 'F':

This subtest reads and transfers the ADC value of the onboard Sensorboard. A software loop (160 loops) is used to transmit data continuously. Its coding differs from Case ‘Z’ in the number of characters sent with each execution of the loop. Whilst Case ‘Z’ transmits 19 characters during each loop, Case ‘F’ transmits 10 characters. Thus, by comparing results obtained from Case ‘F’ with Case ‘Z’ we are able to determine the effect of the amount of characters sent with regard to the throughput performance.

Table 3: Frimware for Case 'F'

```
case 'F':
WriteTimer3 (0);
Time = 0;
overflow = 0;
LMTTransmit( 0, 0, 1600, 0, 0 );  // expect 1600 bytes
for (i_loop=0; i_loop<160; i_loop++)
{
     AN2Pin = ~AN2Pin;
     AnalogVal_13bit = read_adc();
     T3Time = ReadTimer3();
     sprintf (buf, "=%4x=%2x\n\r\0this is junk", i_loop,
     T3Time, overflow);
     WriteTimer3 (0);
     overflow = 0;
     LMTTransmit( 0,(unsigned char *) buf, 10, 0, 0 );
}
```

This code transmits 10 characters continuously 160 times. This test is used primarily to test a different batch transmission method and transmits data in the following format: i_loop=T3Time=overflow. The ‘=’ character is used to separate the values of different variables. Each line is terminated with ‘\n’ and ‘\r’ characters in order to move data logging to the beginning of a new line (in the data file). The batch transmit method is also used to transmit the data.

Before entering the 160 loops, a LMTTransmit service is called with the parameters (0, 0, 1600, 0, 0), telling the service to expect 1600 characters. Then the loops are executed, with each loop transmitting 10 characters.

8.2.3.3 Case ‘K’:

This subtest is used to receive the Toothpick’s RSSI value.

Table 4: Frimware for Case 'K'

```
case 'K':
Time = 0;
WriteTimer3 (0);
overflow = 0;

for (i_loop=0; i_loop<160; i_loop++)
{
     AwaitLMTComplete();
     LMTCommand( LMTC_Generic, 0, (rom void *) "RSSI 0"
     );
     while (!getRSSI);
     T3Time = ReadTimer3();
     LMTTransmit( 0,0, 23, 0, 0 ); //expect 23 chars
     sprintf (buf, "%4x=%4x=%4x=%6s\n\r\0",
     i_loop,T3Time, overflow , blinkRSSI);
     LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
     getRSSI = 0;
     WriteTimer3 (0);
     overflow = 0;
}
```

This code transmits 23 characters continuously 160 times. A software loop (160 loops) is used to transmit data continuously. The data is transmitted in the following format: i_loop=T3Time=overflow=blinkRSSI. The ‘=’ character is used to separate the values of the different variables. Each line is terminated with the '\n' and '\r' characters in order to move the data logging to the beginning of a new line (in the data file). The batch transmit method is used to transmit the data. Before each transmission, the LMTTransmit service is called with the parameters (0, 0, 23, 0, 0), telling it to expect 23 characters. This is carried out before each normal execution of the LMTTransmit service. Thus within each loop a batch transmission command is transmitted (that is 160 times).

8.2.3.4 Case 'L':

This subtest is primarily used to transmit the BER value of the Toothpick.

Table 5: Firmware of Case 'L'

```
case 'L':
Time = 0;
WriteTimer3 (0);
overflow = 0;

for (i_loop=0; i_loop<160; i_loop++)
{
      AwaitLMTComplete();
      LMTCommand( LMTC_Generic, 0, (rom void *) "BER 0" );
      while (!getBER);
      T3Time = ReadTimer3();
      LMTTransmit( 0,0, 23, 0, 0 ); //expect 23 chars
      sprintf (buf, "%4x=%4x=%4x=%6s\n\r\0", i_loop,
      T3Time, overflow , blinkBER);
      LMTTransmit( 0,(unsigned char *) buf, 23, 0, 0 );
      getBER = 0;
      WriteTimer3 (0);
      overflow = 0;
}
```

This code transmits 23 characters continuously 160 times. A software loop (160 loops) is used to transmit data continuously. The data is transmitted in the following format: i_loop=T3Time=overflow=blinkBER. The '=' character is used to separate the values of the different variables. Each line is terminated with the '\n' and '\r' characters in order to move the data logging to the beginning of a new line (in the data file). The batch transmit method is used to transmit the data. Before each transmission, the LMTTransmit service is called with the parameters (0, 0, 23, 0, 0), telling it to expect 23 characters. This is done before each normal execution of the LMTTransmit service. Thus within each loop a batch transmission command is used (that is 160 times).

8.2.3.5 Case 'X':

This subtest is basically the same as Case 'Z'. A software loop (160 loops) is used to transmit data continuously. The only difference between these tests is that Case 'X' does not actually read the value of the sensorboard's ADC. The same amount of characters is transmitted. By comparing Case 'X' with Case 'Z' we can determine how much the characters' throughput performance is affected by reading the sensorboard's ADC value.

Table 6: Firmware of Case 'X'

```
case 'X':
WriteTimer3 (0);
Time = 0;
overflow = 0;
test_no = 0;
LMTTransmit( 0, 0, 3040, 0, 0 );  // expect 3040 bytes
for (i_loop=0; i_loop<160; i_loop++)
{
     AN2Pin = ~AN2Pin;
     T3Time = ReadTimer3();
     sprintf (buf, "%2x=%4x=%4x=%4x\n\r\0this is junk",
     i_loop , T3Time, overflow , AnalogVal_13bit);
     WriteTimer3 (0);
     overflow = 0;
     LMTTransmit( 0,(unsigned char *) buf, 19, 0, 0 );
}
```

This code transmits 19 characters continuously 160 times (3040 characters). This test is primarily used to transmit the sensorboard’s ADC value continuously without actually reading it. The data is transmitted in the following format: i_loop=T3Time=overflow=AnalogVal_13bit. The ‘=’ character is used to separate the values of the different variables. Each line is terminated with the ‘\n’ and ‘\r’ characters in order to move the data logging to the beginning of a new line (in the data file). The batch transmit method is used to transmit the data. Before entering the 160 loops, a LMTTransmit service is called with the parameters (0, 0, 3040, 0, 0), telling the service to expect 3040 characters. Then the loops are executed, with each loop transmitting 19 characters.

8.2.3.6 Case ‘Z’:

This subtest is almost the same as subheading 8.2.3.5Case ‘above. A software loop (160 loops) is used to transmit data continuously. The only difference between these tests is that during the execution of this subtest, the value of the sensorboard’s ADC is read. By comparing Case ‘X’ with Case ‘Z’ we can determine the amount of time which was required to read the ADC value from the Sensorboard.

Table 7: Firmware of Case 'Z'

```
case 'Z':
  WriteTimer3 (0);
  Time = 0;
  overflow = 0;
  LMTTransmit( 0, 0, 3040, 0, 0 );  // expect 3040 bytes
  for (i_loop=0; i_loop<160; i_loop++)
  {
     AN2Pin = ~AN2Pin;
     AnalogVal_13bit = read_adc();
     T3Time = ReadTimer3();

     sprintf (buf, "%2x=%4x=%4x=%4x\n\r\0this is junk",
     i_loop, T3Time, overflow , AnalogVal_13bit);
     WriteTimer3 (0);
     overflow = 0;
     LMTTransmit( 0,(unsigned char *) buf, 19, 0, 0 );
  }
```

This code transmits 19 characters continuously 160 times (3040 characters). This test is primarily used to transmit the sensorboard’s ADC value continuously and rapidly after reading it. The data is transmitted in the following format: i_loop=T3Time=overflow=AnalogVal_13bit. The ‘=’ character is used to separate the values of the different variables. Each line is terminated with the ‘\n’ and ‘\r’ characters in order to move the data logging to the beginning of a new line (in the data file). The batch transmit method is used to transmit the data. Before entering the 160 loops, a LMTTransmit service is called with the parameters (0, 0, 3040, 0, 0), telling the service to expect 3040 characters. Then the loops are executed, each loop transmitting 19 characters.

8.3 CONCLUSION

Chapter 8 attempted to describe the different experiments which were performed and their results. The outcome of the performance tests was used to write the firmware for the physical tests. The physical tests show how the communication link is affected by the spinning environment. The results of the tests are explained in Chapter 9.

CHAPTER 9

THE RESULTS

9.1 OVERVIEW

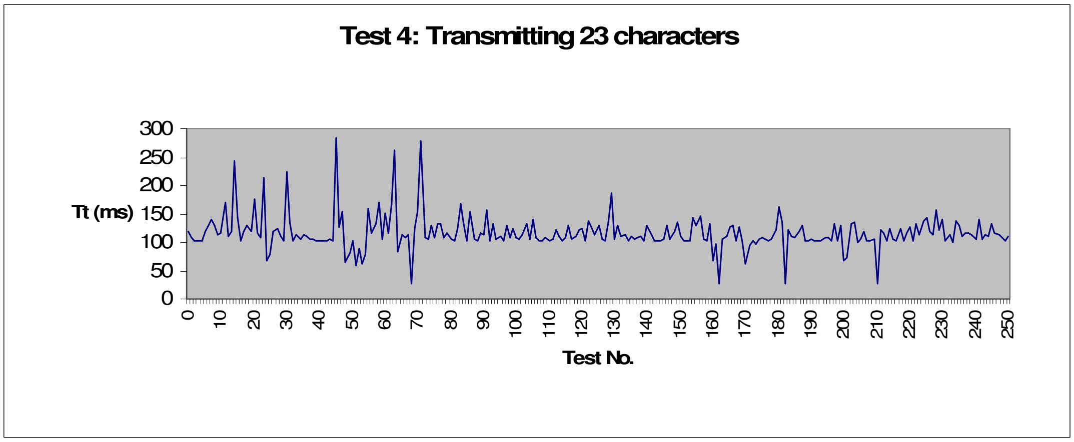
This chapter compares the results of certain tests, showing how the performance of the Bluetooth experimental system was affected or changed by changing its firmware (see 9.2 The Performance Tests) or the physical environment (see 9.3 The Physical Tests).

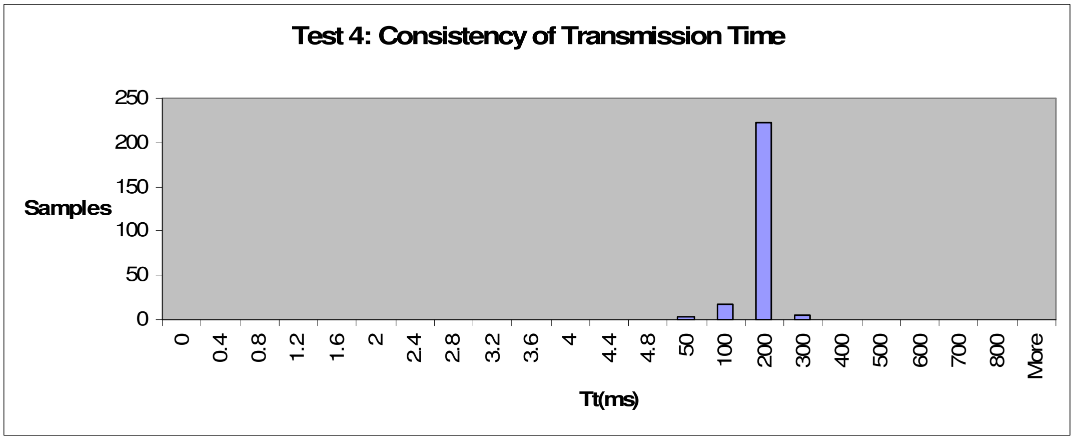
9.2 THE PERFORMANCE TESTS

9.2.1 Transmitting 23 characters continuously

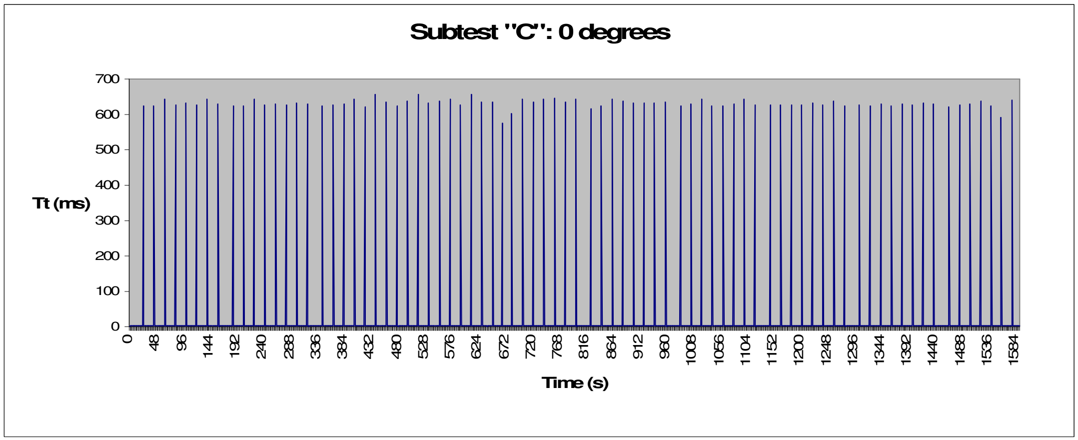
Three different methods are compared. Test 4 transmitted 23 characters without using batch transmission. The average transmission time is 116.7835637 ms. The histogram shows that most of the messages were below 200 ms. Both Cases “C” and “Z” has higher performance. These cases use batch processing and transmit 19 characters consecutively. They differ in the number of batch commands sent to the LinkMatik module. For example, both cases transmit 3040 characters. During Case “C” each batch transmission command informs the Linkmatik that it wants to transmit19 characters and then the characters is sent. This requires 160 batch commands in total. Case “Z”, however, informs the LinkMatik that it wants to send 3040 characters, and then 19 characters are sent consecutively, 160 times.

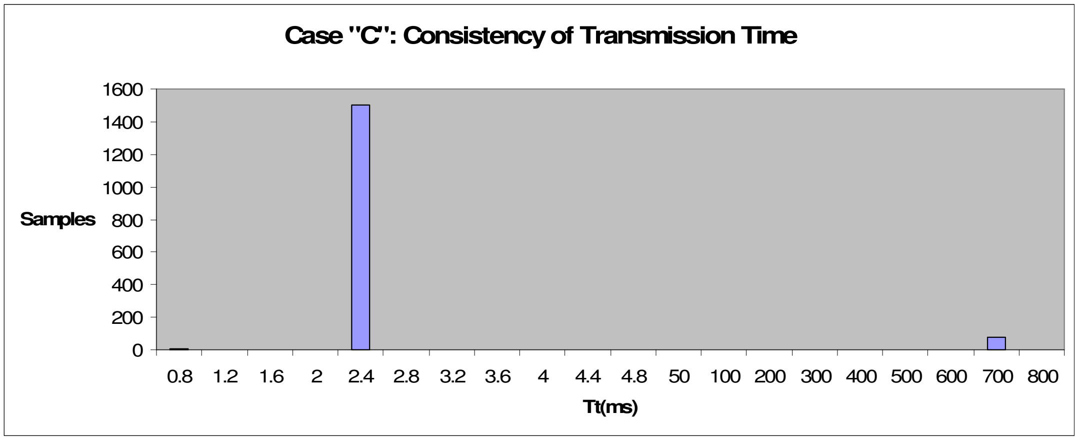
9.2.2 Test 4



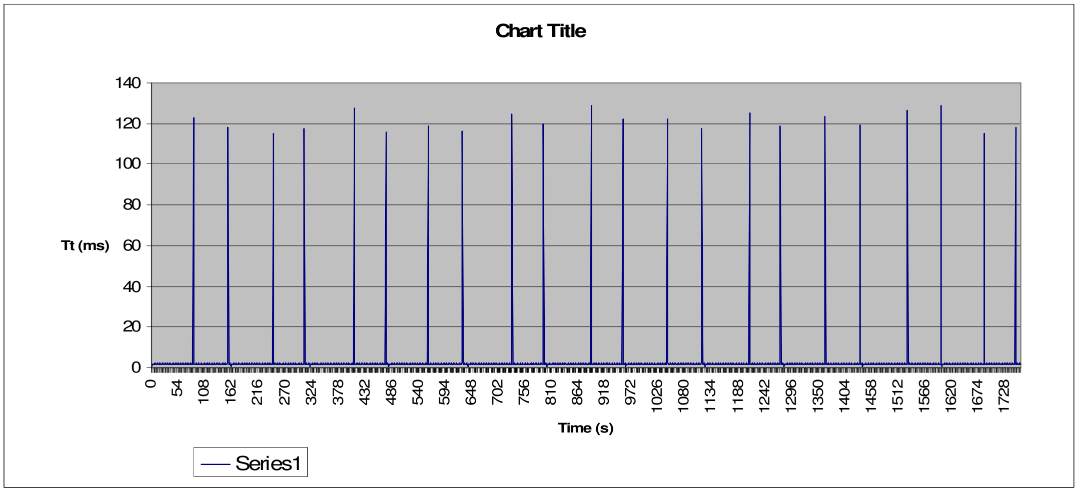


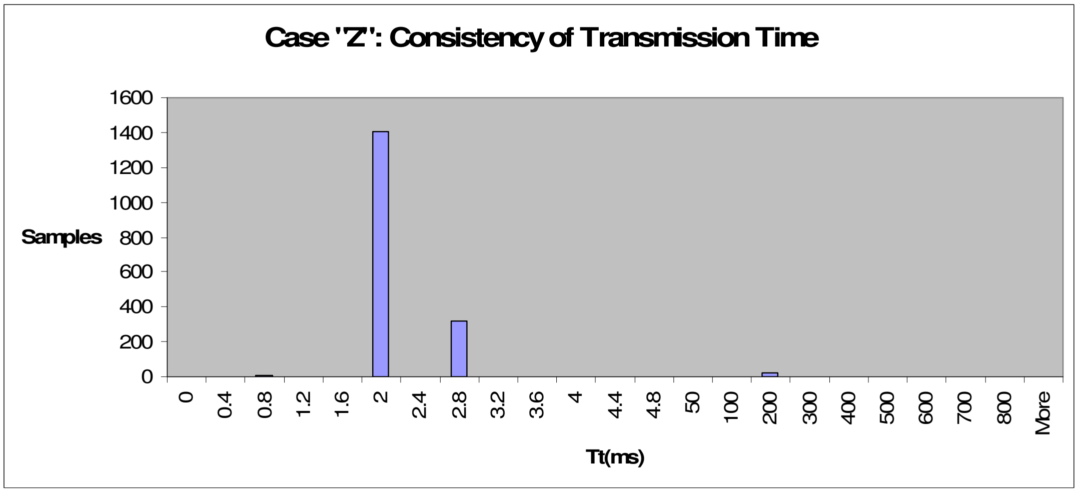
9.2.3 Case “C”


Subtest "C": 0 degrees
Tt (ms)
0
100
200
300
400
500
600
700
Time (s)
0
48
96
144
192
240
288
336
384
432
480
528
576
624
672
720
768
816
864
912
960
1008
1056
1104
1152
1200
1248
1296
1344
1392
1440
1488
1536
1584


Case "C": Consistency of Transmission Time
Samples
0
200
400
600
800
1000
1200
1400
1600
Tt(ms)
0.8
1.2
1.6
2
2.4
2.8
3.2
3.6
4
4.4
4.8
50
100
200
300
400
500
600
700
800

9.2.4 Case “Z”





9.2.5 Conclusion

By comparing these tests, it is seen that using the software for Case “Z” provides the best performance. The transmission time (Tt) for Case “Z” usually varies between 2 and 2.8ms, worst case takes 200 ms. Case “C” performs similarly, but some of the samples take 700ms to transmit. The transmission times of Test 4 are grouped more closely, varying between 100 – 300 ms.

For Case "C", a noticeable and higher than average transmission delay occurs almost every 40 ms or after the 18th (or 25th) transmission. This delay can be as large as 657.224 ms. Case "Z" has the same, higher than average transmission delay, albeit smaller (as long as 128.8725 ms). This occurs after every 63rd transmission or roughly every 140 ms. Case "Z" has the best performance.

9.3 PHYSICAL TESTS

The physical tests are performed to determine how the communication link between the experimental setup and the PC is affected by changing various physical properties.

9.3.1 Effect of changing the RPM of the spindle on Transmission Time

Case 'C', Case 'X' and Case 'Z' was used to determine what the effect of spindle rotation speed had on the transmission time. For each test, the spindle speed was kept constant while the experimental setup transmitted data (first Case 'C', then Case 'X' and finally Case 'Z'), after which the spindle speed was increased to a new constant value and the various Cases was executed again.

9.3.1.1 Case “C”

Table 8: Test 6 (Static test at 0RPM)

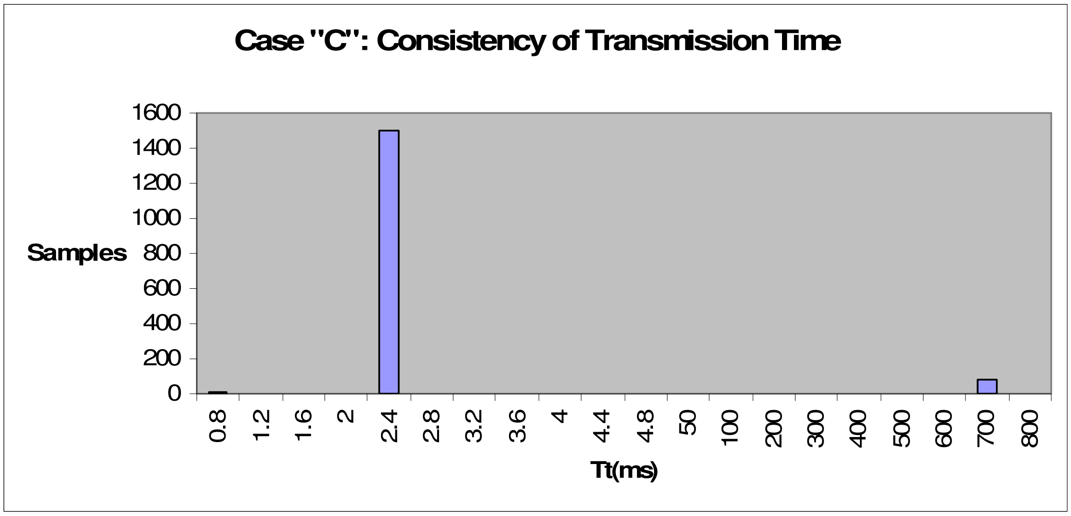


Table 9: Test 7 (taken at 100RPM)

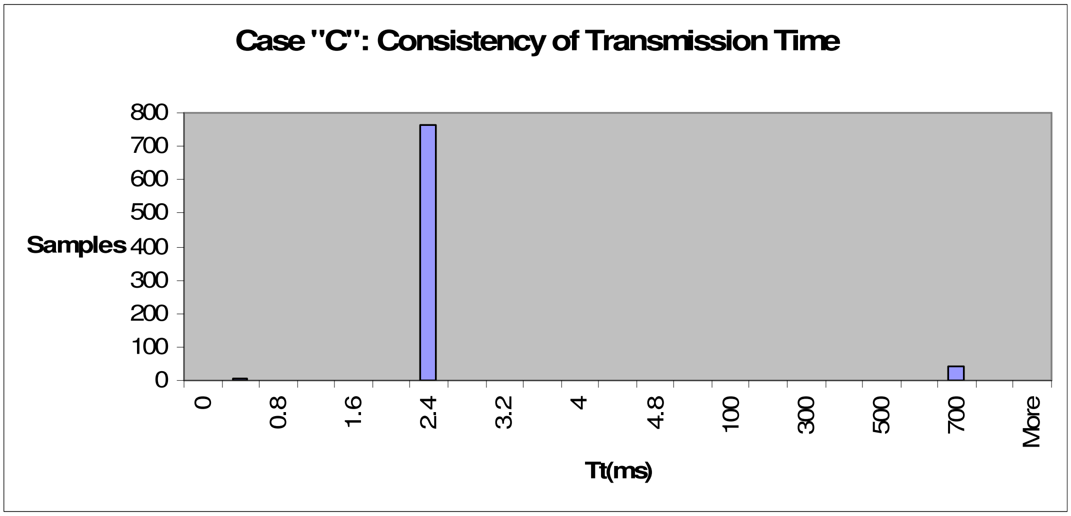


Table 10: Test 8 (taken at 200RPM)

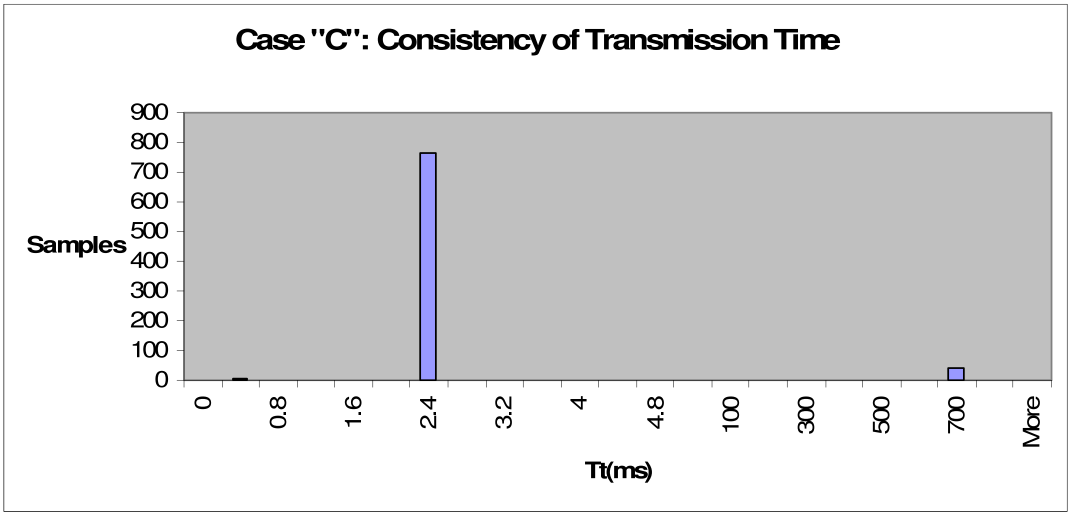


Table 11: Test 9 (taken at 500RPM)

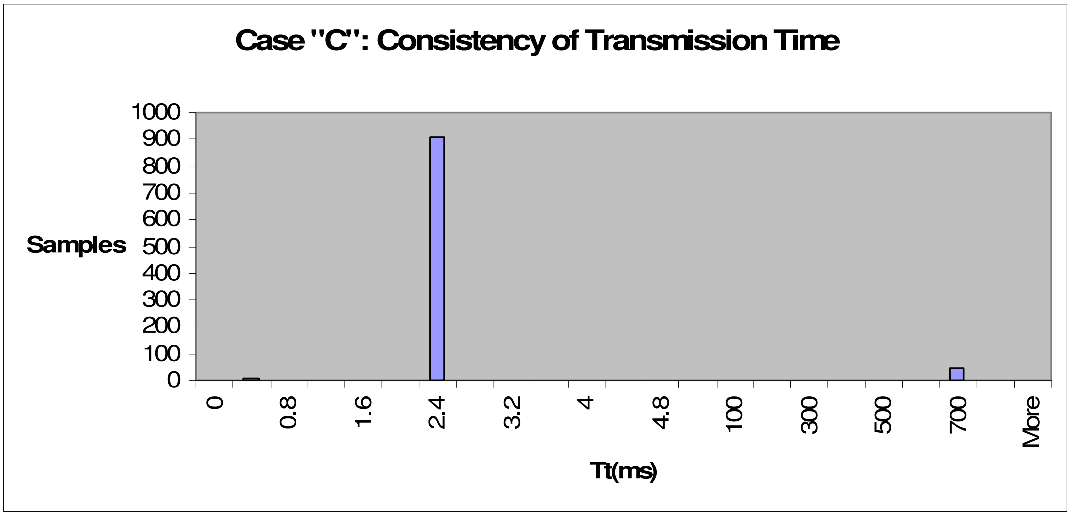


Table 12: Test 10 (taken at 1000RPM)

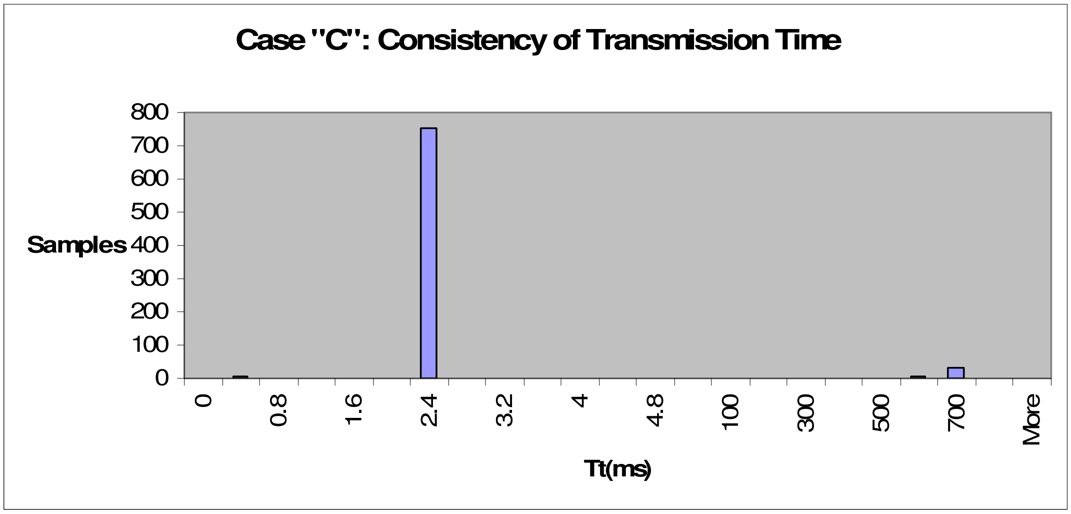
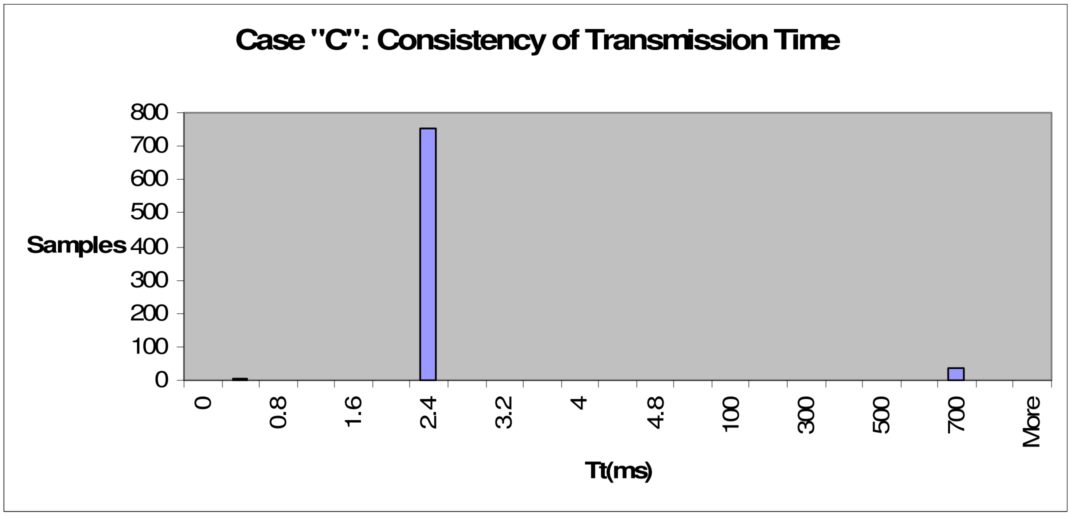


Table 13: Test 11 (taken at 1400RPM)



9.3.1.2 Case “X”

Table 14: Test 6 (Static test at 0RPM)

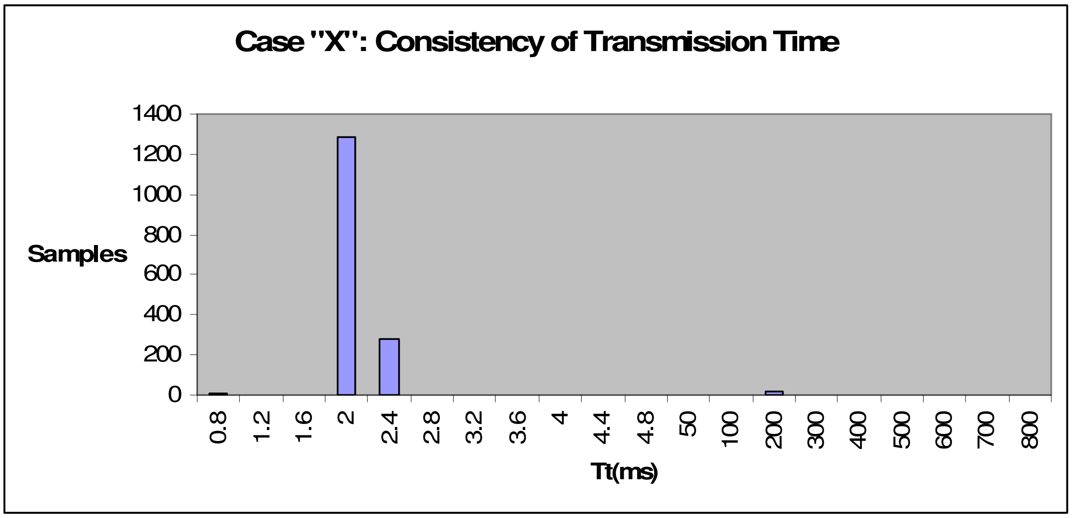


Table 15: Test 7 (taken at 100RPM)

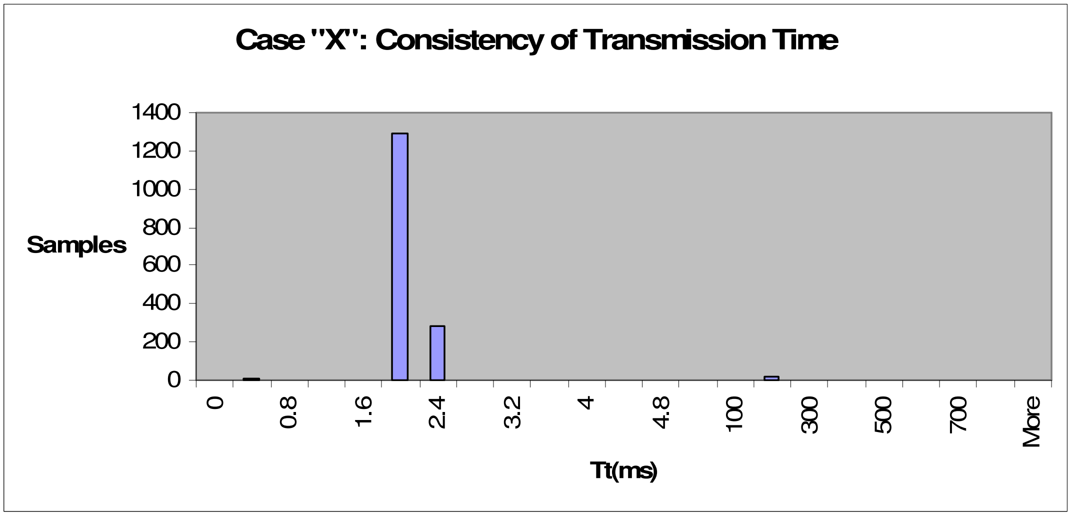


Table 16: Test 8 (taken at 200RPM)

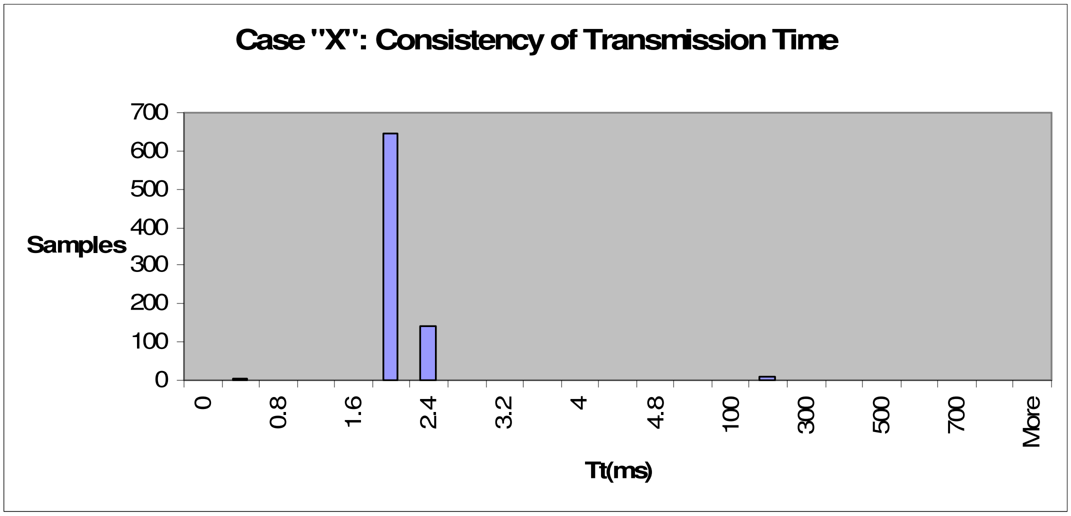


Table 17: Test 9 (taken at 500RPM)

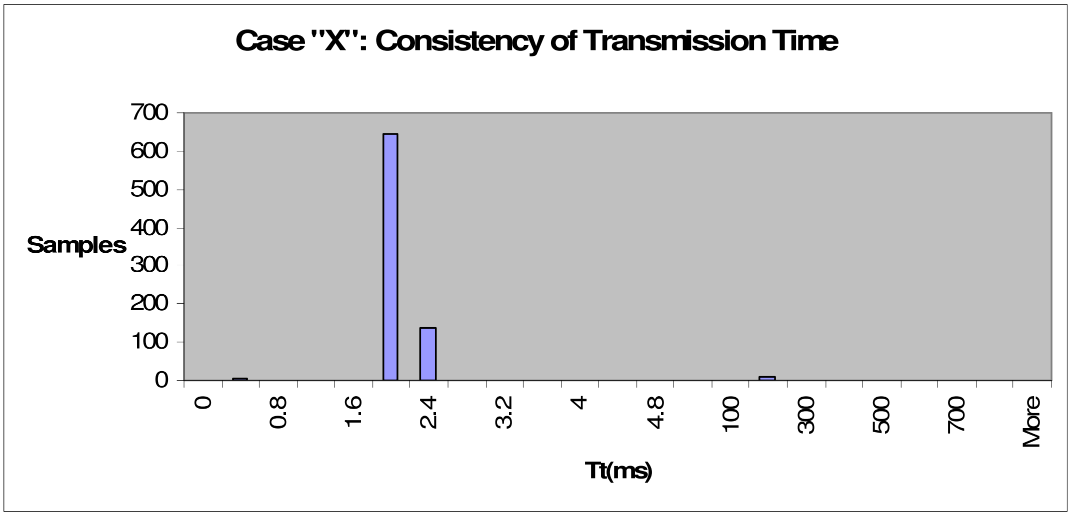


Table 18: Test 10 (taken at 1000RPM)

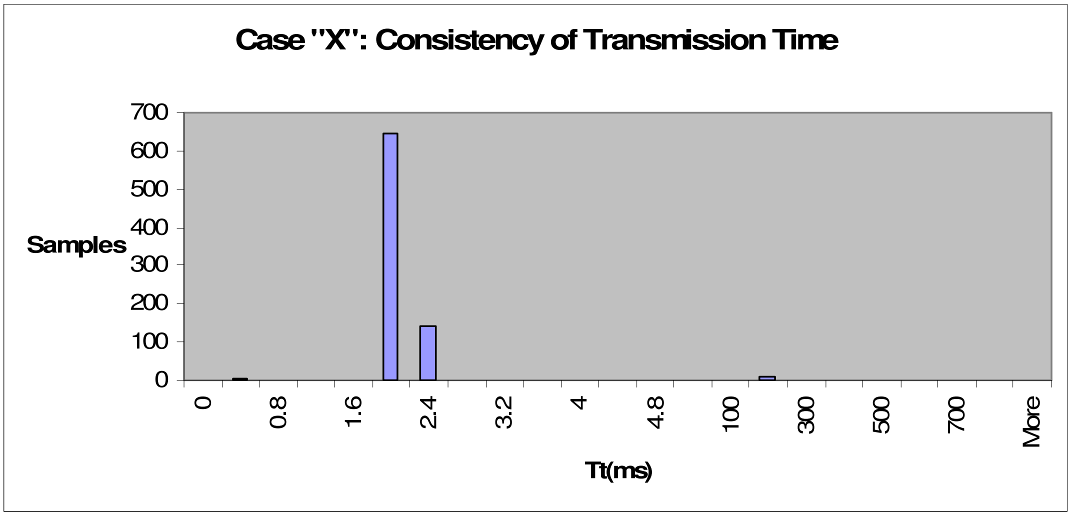
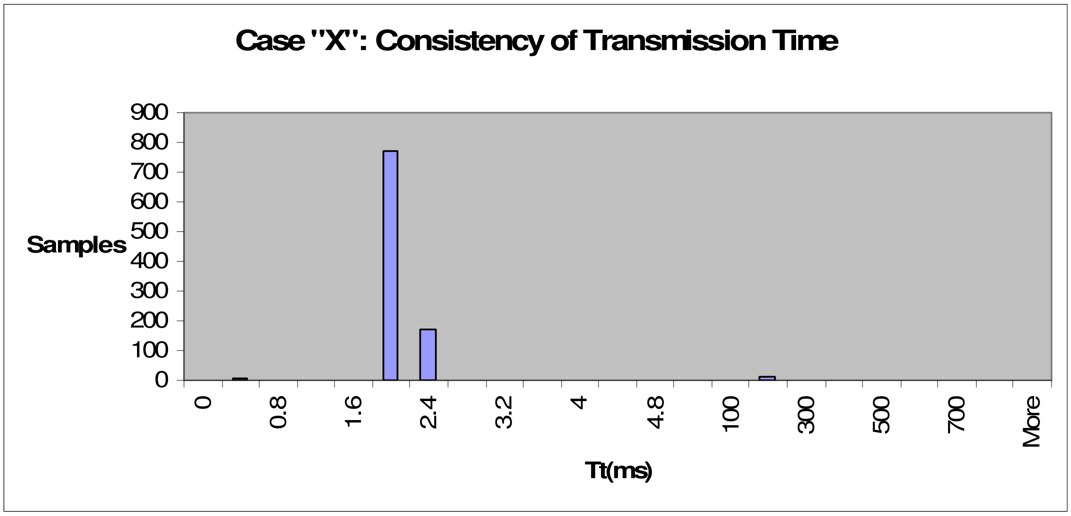


Table 19: Test 11 (taken at 1400RPM)



9.3.1.3 Case “Z”

Table 20: Test 6 (Static test at 0RPM)

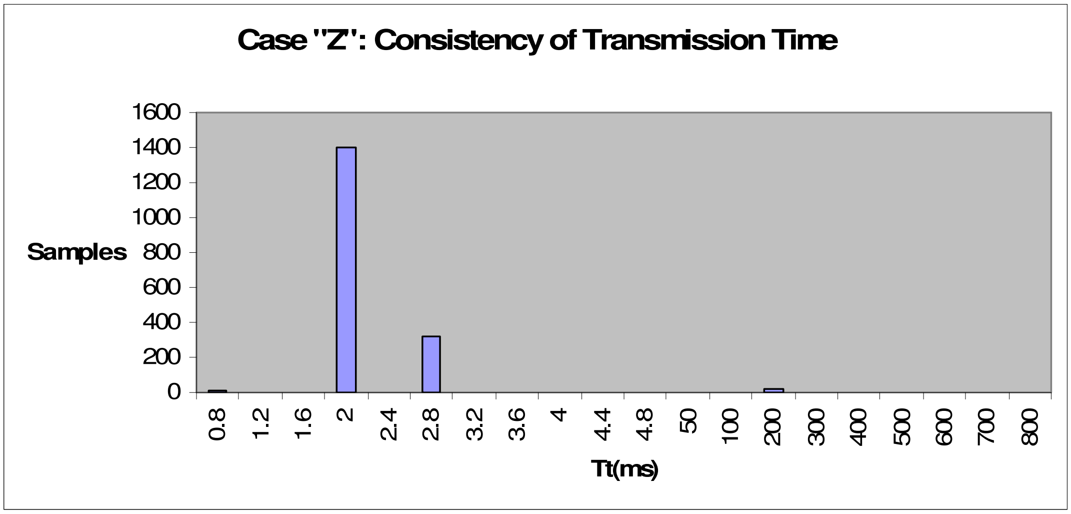


Table 21: Test 7 (taken at 100RPM)

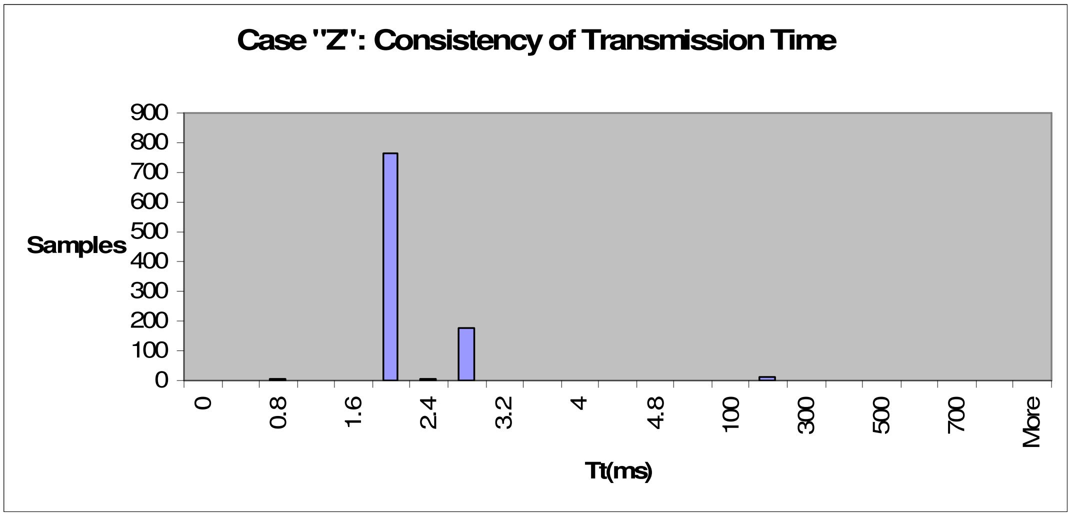


Table 22: Test 8 (taken at 200RPM)

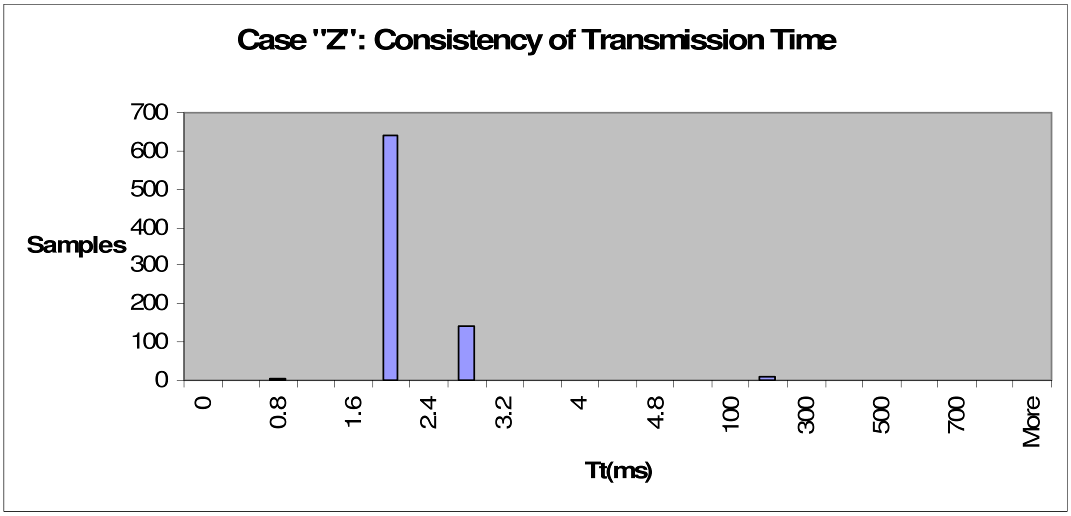


Table 23: Test 9 (taken at 500RPM)

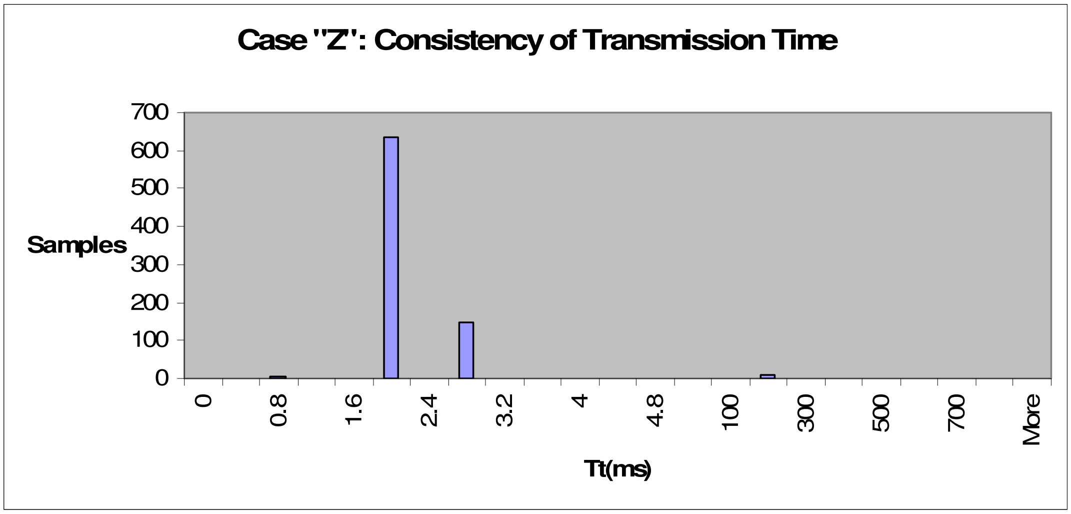


Table 24: Test 10 (taken at 1000RPM)

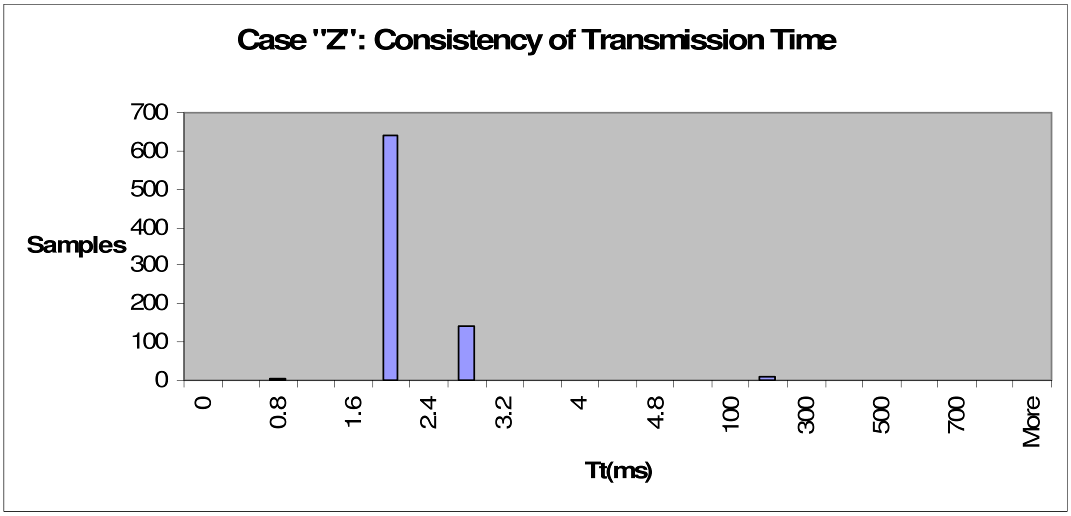
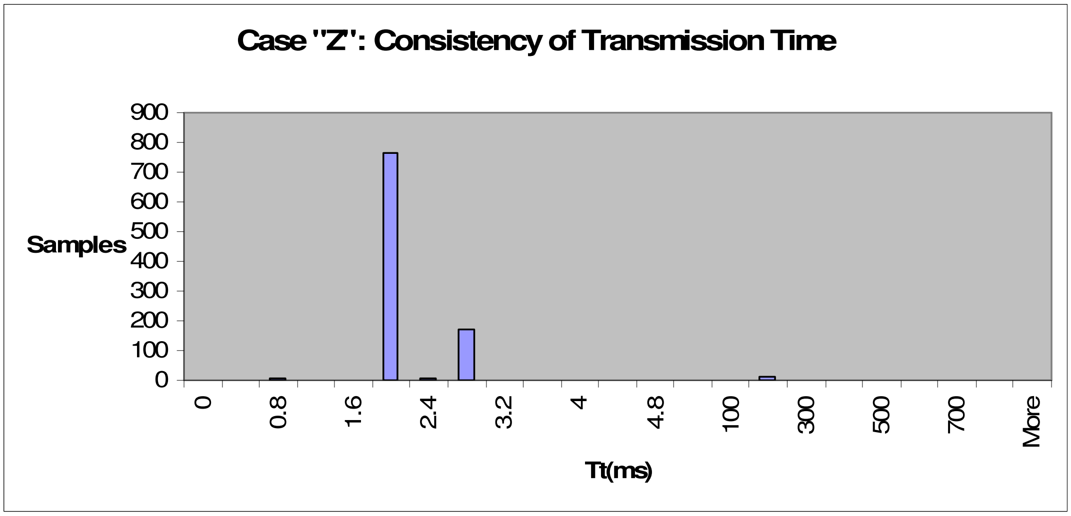


Table 25: Test 11 (taken at 1400RPM)



9.3.1.4 Conclusion

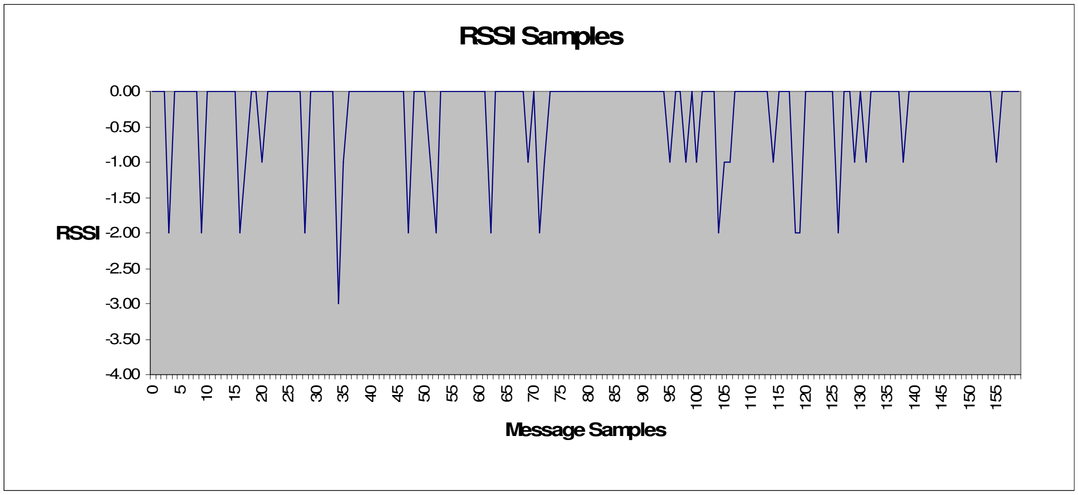
It can be seen from these graphs that the Transmission Time (Tt) is not affected by the spinning speed of the spindle. If it were affected, the Transmission time would show a definite increase or decrease.

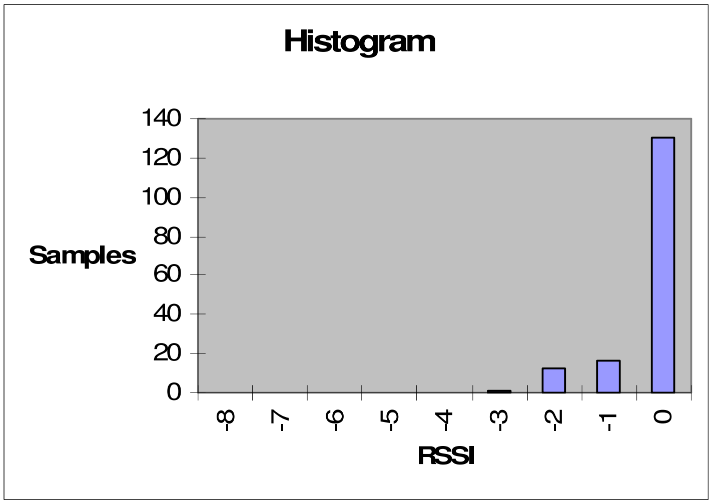
9.3.2 The effect of changing the RPM on the Communication Channel

Case ‘K’ and Case ‘L’ was used to determine if spindle rotation speed affects the Bluetooth communication link quality. For each test, the spindle speed was kept constant while the experimental setup transmitted data (first Case ‘K’, then Case ‘L’), after which the spindle speed was increased to a new constant value and the various Cases was executed again.

9.3.2.1 Case "K"

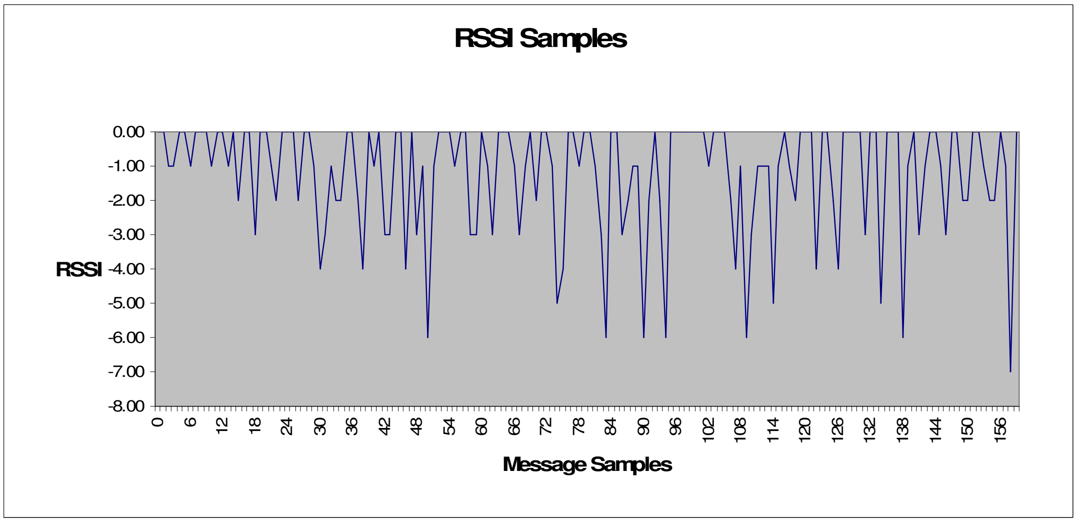
Table 26: Test 6 (Static taken at 0RPM)

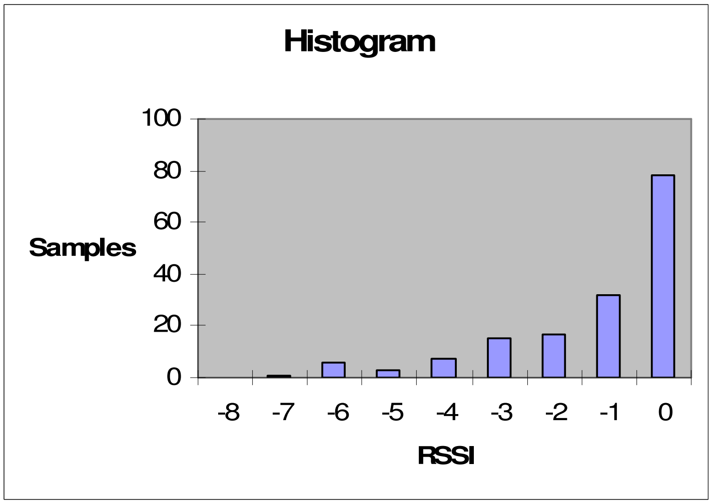




RSSI	Frequency
-8	0
-7	0
-6	0
-5	0
-4	0
-3	1
-2	12
-1	16
0	130
More	0

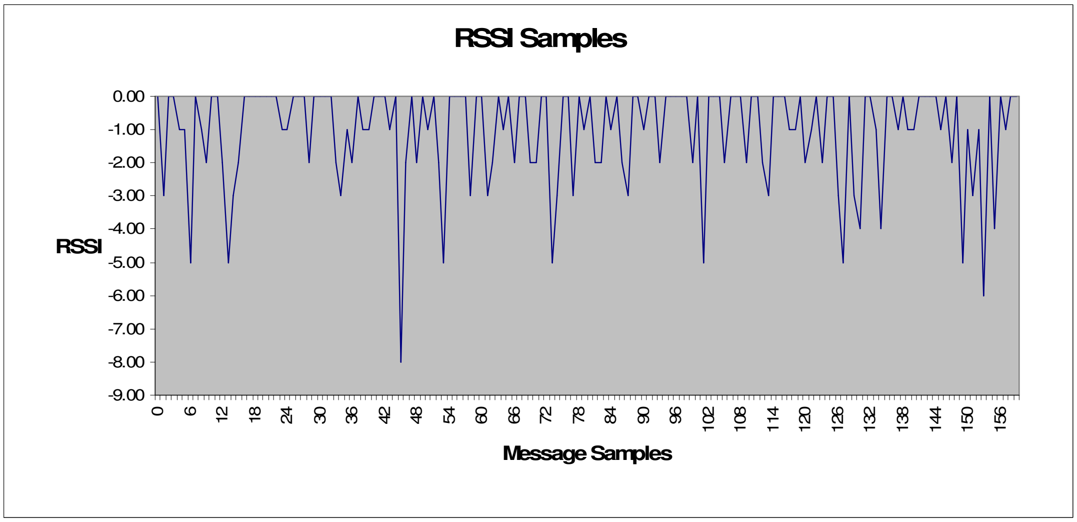
Table 27: Test 7 (taken at 100RPM)

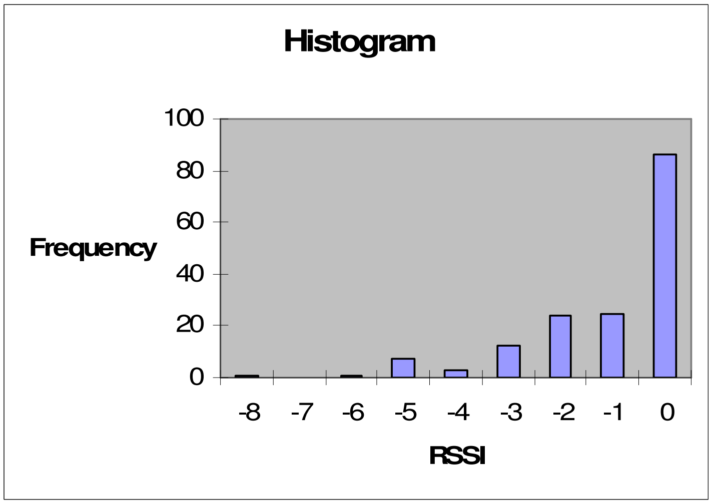




RSSI	Samples
-8	0
-7	1
-6	6
-5	3
-4	7
-3	15
-2	17
-1	32
0	78
More	0

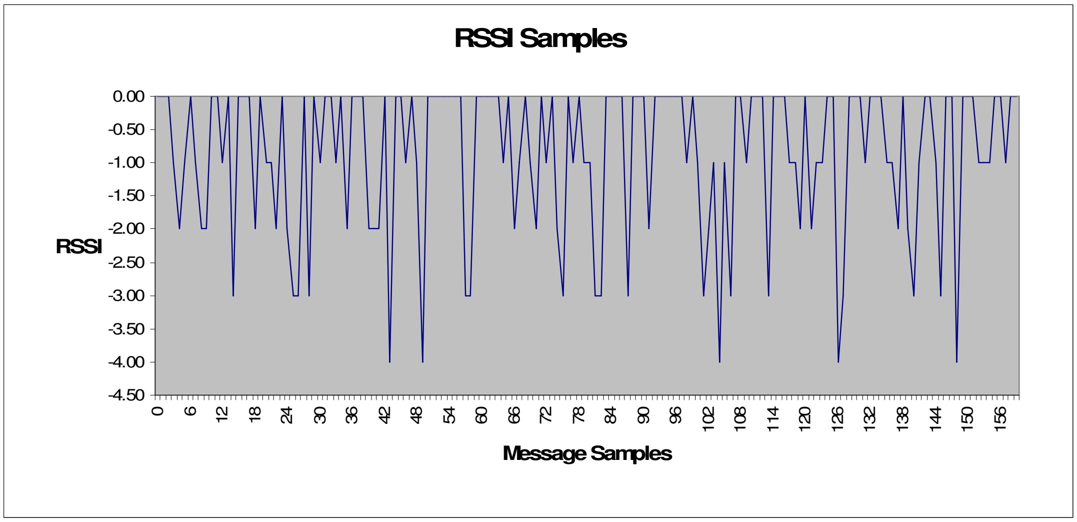
Table 28: Test 8 (taken at 200RPM)

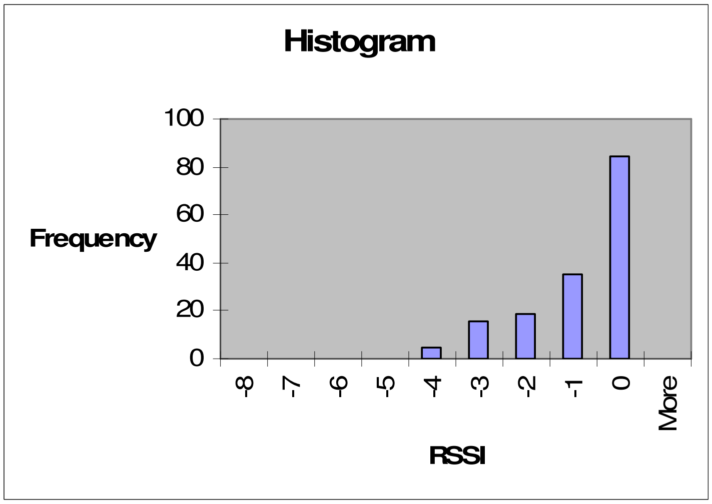




RSSI	Frequency
-8	1
-7	0
-6	1
-5	7
-4	3
-3	12
-2	24
-1	25
0	86
More	0

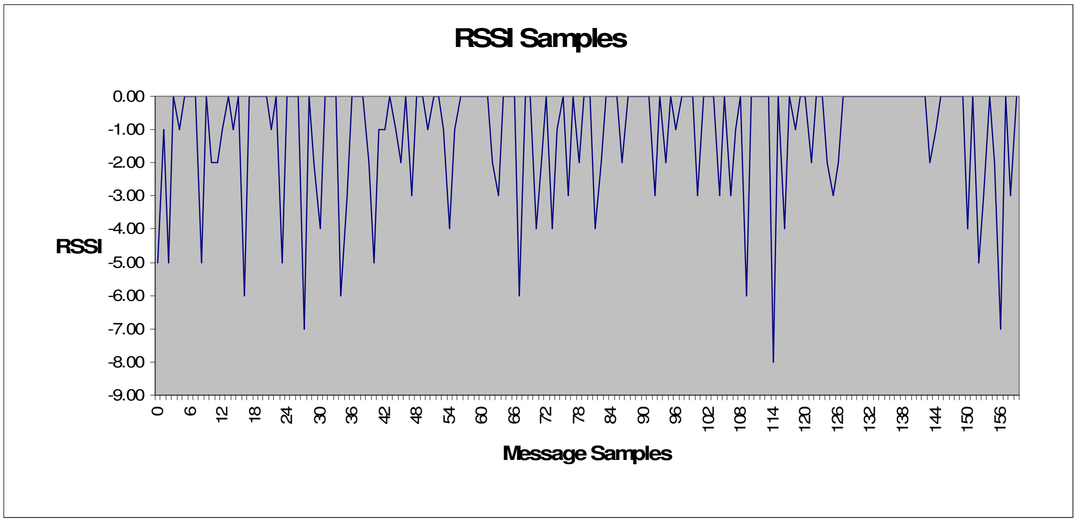
Table 29: Test 9 (taken at 500RPM)

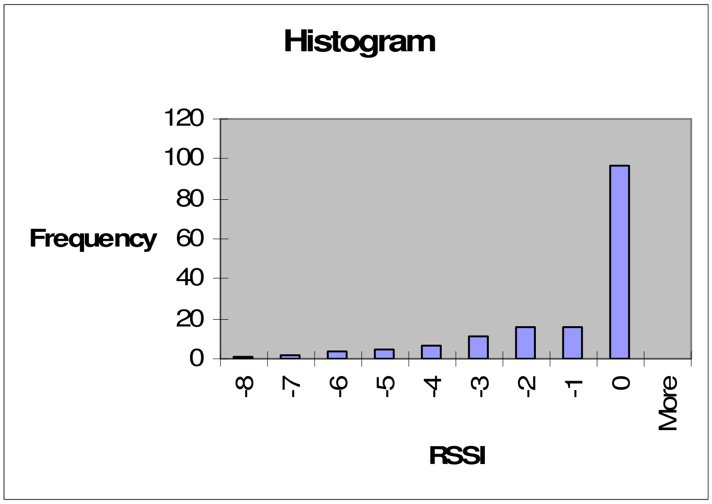




RSSI	Frequency
-8	0
-7	0
-6	0
-5	0
-4	5
-3	16
-2	19
-1	35
0	84
More	0

Table 30: Test 11 (taken at 1400RPM)





RSSI	Frequency
-8	1
-7	2
-6	4
-5	5
-4	7
-3	11
-2	16
-1	16
0	97
More	0

9.3.2.2 Case “L”

Table 31: Test 6 (Static test taken at 0RPM)

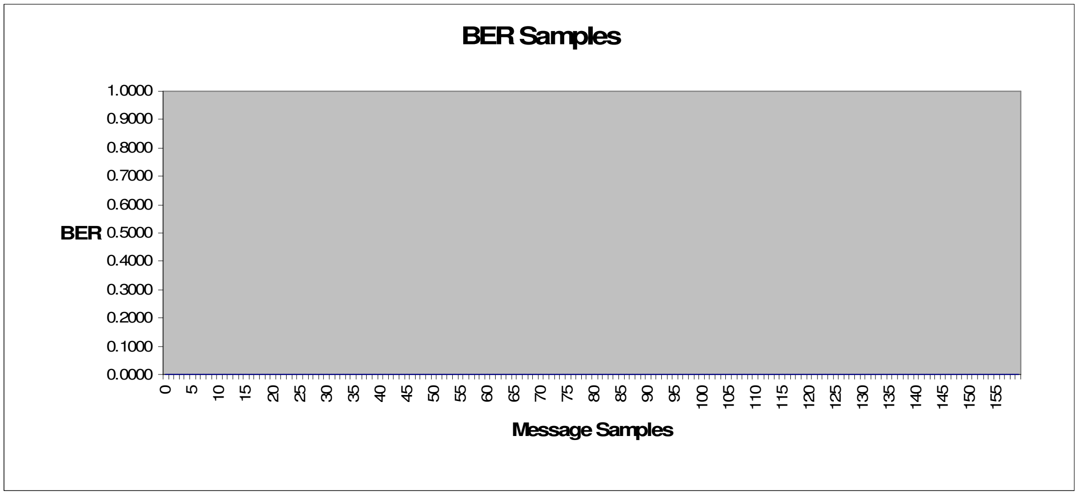


Table 32: Test 7 (taken at 100RPM)

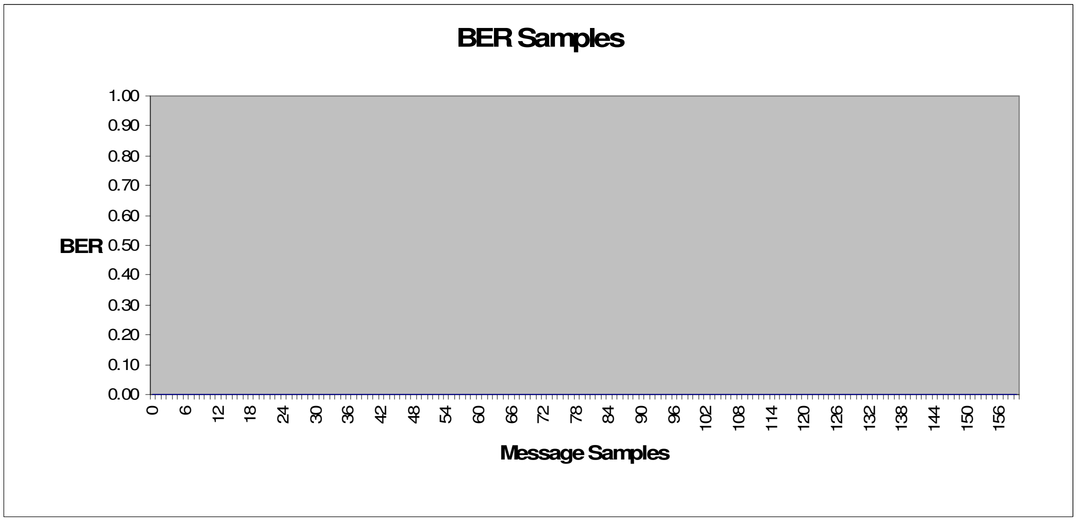


Table 33: Test 8 (taken at 200RPM)

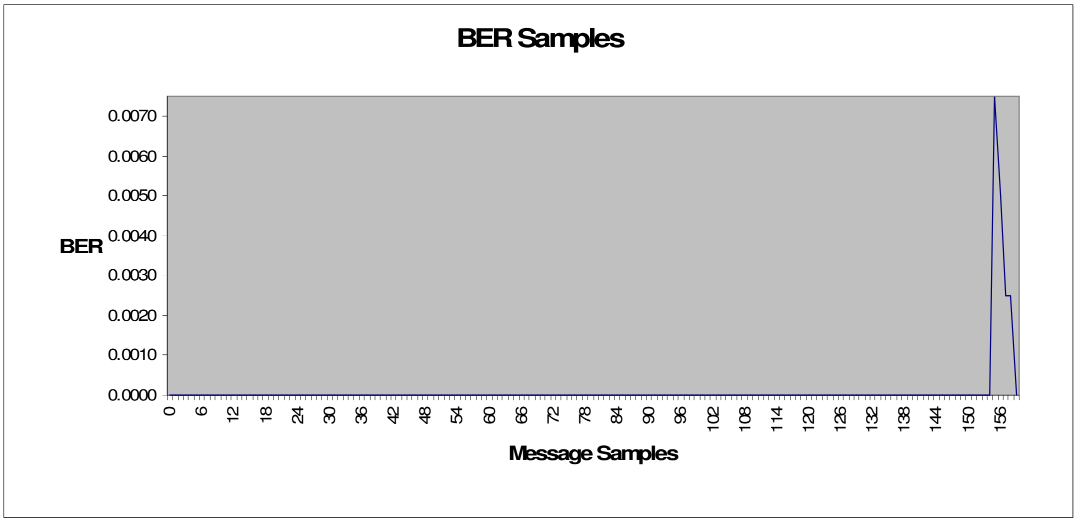


Table 34: Test 9 (taken at 500RPM)

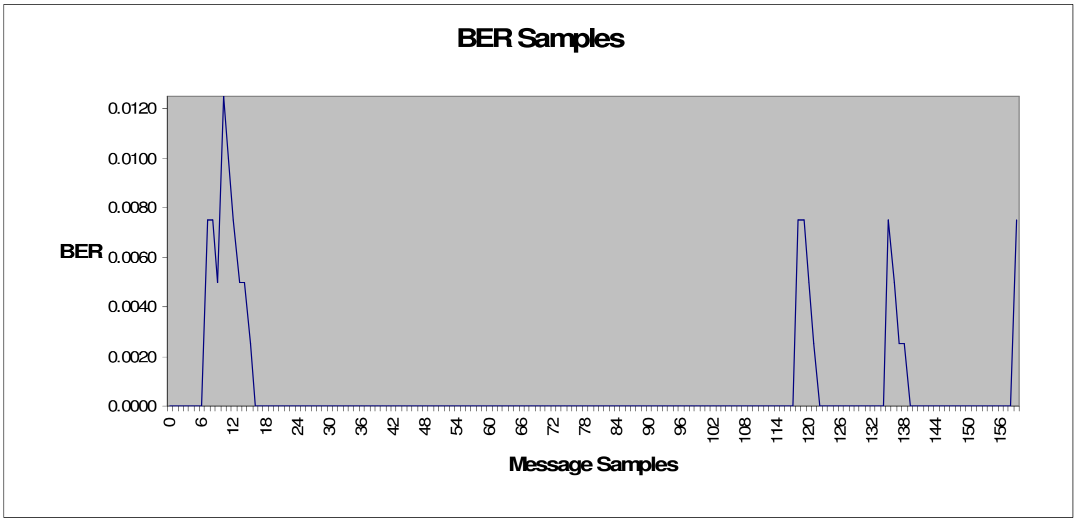
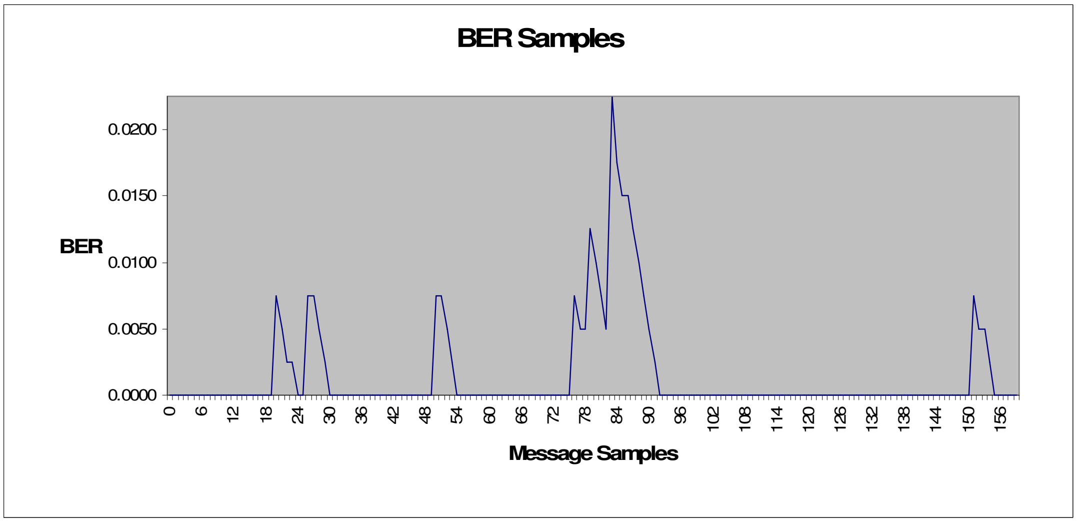


Table 35: Test 11 (taken at 1400RPM)



9.3.2.3 Conclusion

From these tests in Tables 26 - 30 it can be seen that the RSSI decreases with an increase in spindle RPM (reaching a peak of -8 at an RPM of 1400). Furthermore, in Tables 31 – 35 it can also be seen that the BER increases as the spindle RPM increases. A peak of 0.02 is reached at an RPM of 1400. Thus, it can be concluded that the Bluetooth communication's link quality decreases as the spindle's RPM increases.

9.3.3 The effect of the angle between the two Bluetooth-connected devices

It was found that by adjusting the angle between two communicating Bluetooth devices had a considerable effect on the link quality of their connection. Bellow are two graphical representations of the RSSI value of the Toothpick, which was logged by the PC. Naught degrees (0º) mean that the

PC's Bluetooth transceiver was placed directly below the chuck, which contained the Toothpick – they are vertically aligned. Ninety degrees (90º) means that the PC's Bluetooth transceiver is horizontally aligned with the chuck.

9.3.3.1 Case “K”

Table 36: RSSI samples taken at 0 degrees

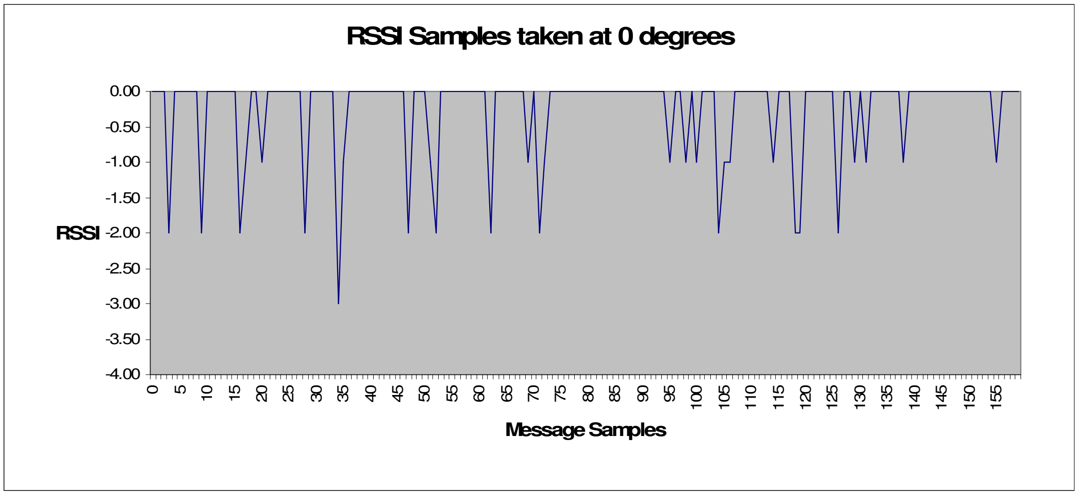
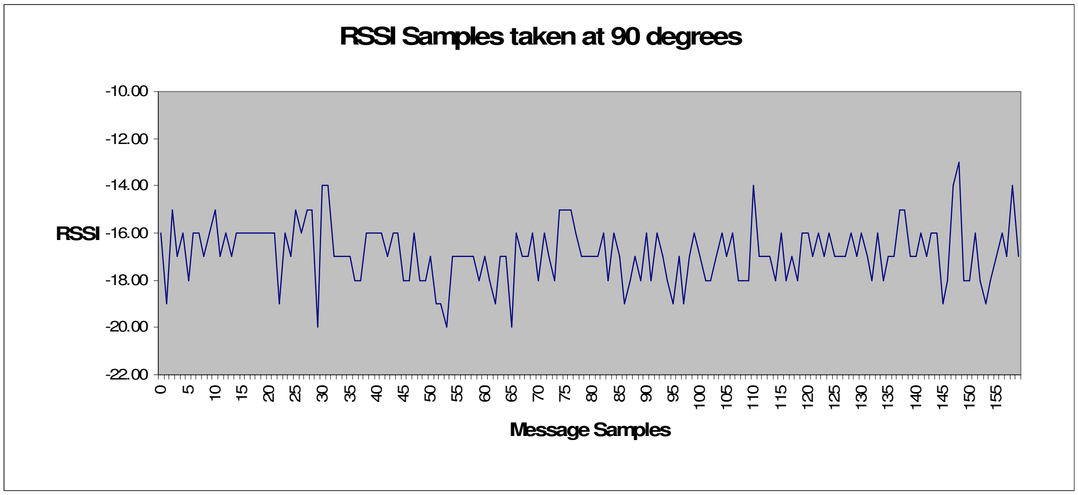
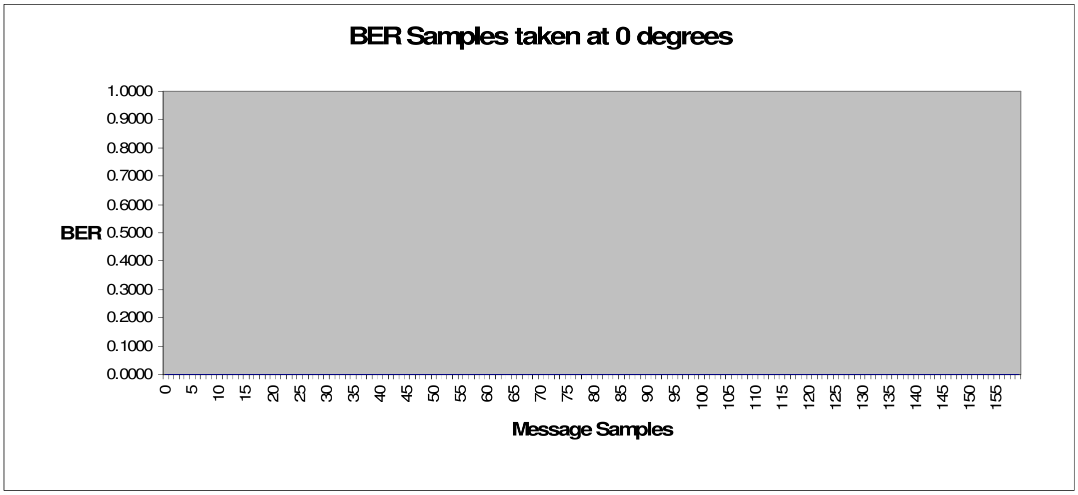


Table 37: RSSI samples taken at 90 degrees



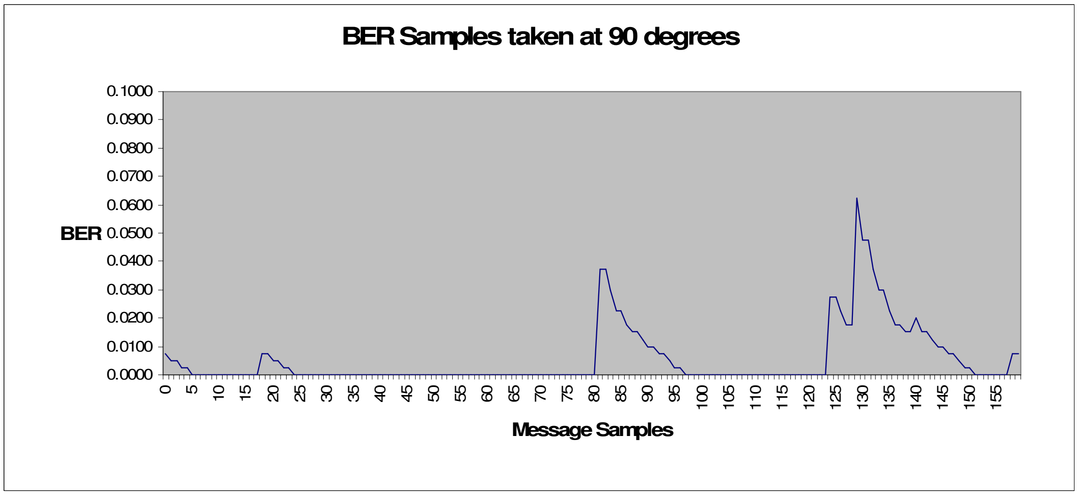
9.3.3.2 Case “L”

Table 38: BER samples taken at 0 degrees



Average BER	0.0000
Minimum BER	0.0000
Maximum BER	0.0000

Table 39: BER samples taken at 90 degrees



Average BER	0.0055
Minimum BER	0.0000
Maximum BER	0.0625

Table 40: Test 11 (taken at 1400RPM)

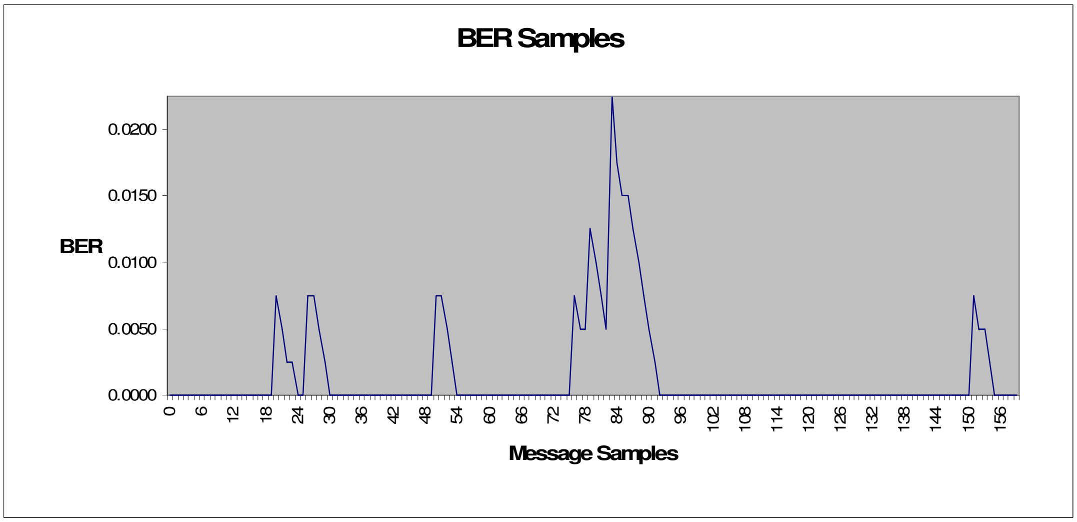
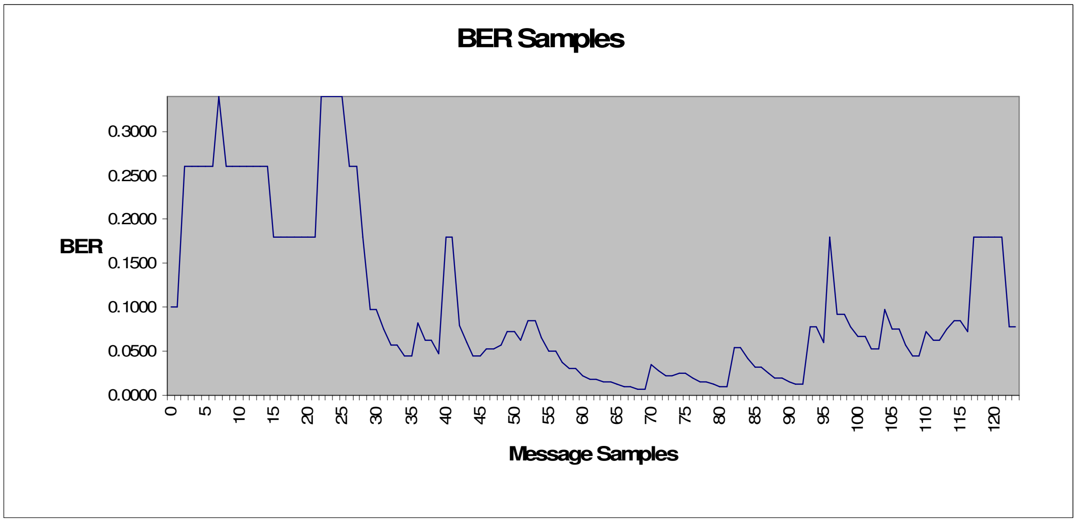


Table 41: Test 13 (taken at 1400RPM and changing the position of angle between the two Bluetooth-connected devices)



9.3.3.3 Conclusion

As can be seen from the Table 36 (0 degrees) and 37 (90 degrees), the RSSI value decreased dramatically from around 0 (at 0 degrees) and minimum spikes at -3, to -16 (at 90 degrees) and minimum spikes of -20, just by changing the angle between the two Bluetooth devices.

Looking at Table 38 and 39, shows that the BER values increases and more frequent spikes are obtained, indicating that the link quality is decreasing. The average BER increased from 0 (when measured at 0 degrees) to 0.0055 (when measured at 90 degrees). Table 40 was taken at 1400 RPM with the PC's Bluetooth antennae directly below (0 degrees) the spinning experimental setup. Table 41 on the other hand shows what happens if the PC's Bluetooth antenna is moved around (and not left directly below the spinning experimental setup). The increased interference in the Bluetooth link is apparent, thus decreasing the communications link.

CHAPTER 10

DISCUSSION, CURRENT LIMITATIONS, FUTURE DEVELOPMENTS AND CONCLUSION

10.1 INTRODUCTION

This chapter attempts to sum up the experimental setup's capabilities and limitations. The design objectives are listed and explained. The various problems which had to be overcome are discussed.

10.2 OBJECTIVES ACHIEVED

An experimental setup, capable of reading data from a sensor and transmitting that data via Bluetooth was developed. The proposed sub-problems which needed to be overcome included:

- The development and implementation cost of the experimental system must be kept low.
- The product must be capable of handling high spindle speeds (higher than 10000 RPM).
- The product must comply with the Bluetooth standard.
- The data must be sent wirelessly via the Bluetooth protocol; other means (such as wireless-Ethernet) will also be investigated.
- The communication system's channels will consist of:
    i. two force channels
    ii. one or two temperature channels
    iii. one accelerometer channel

- Each channel will consist of 16 bits. Therefore: 16 x 8 (8 channels max) = 128 bits.
- Bluetooth v1.2 can send data at 723.2 kbps; or roughly 1,4µs/bit
- Time required for 8 channels of data: 128 x 1.4 µs = 179.2 µs per package.

The spinning experimental setup which was designed included a Toothpick Bluetooth module. This module includes a PIC microcontroller and a LinkMatik Bluetooth Radio combination, and was loaded with firmware developed using MPLAB C18. The Toothpick was programmed to:

- Write configuration data to an AN221E04 dynamically reconfigurable FPAA sensorboard. This would configure the sensorboard to amplify/filter/sample an analog sensor (like for example a thermocouple).
- Read the sensor's (type – K thermocouple) value from a 13-bit ADC chip using SPI.
- Transmit this sensor's data as well as other variables to the PC via Bluetooth (in order to be logged).
- Receive data from the PC via Bluetooth.

The spinning experimental setup is fully compatible with the standard Bluetooth specification. The spinning experimental setup connects to any Bluetooth-enabled PC via a virtual COM port. It can also be spun at high spindle-speeds. The highest spindle-speed tested was 1500 RPM. The experimental setup was mounted in a round plastic board. Pieces were milled out for the boards and the cables to fit perfectly inside the plastic board. A stainless steel plate was fastened to the plastic board and covered the electronics. This plate provided a secure platform to connect the electronics to the spindle. The plate was connected to the tool chuck using a specially machined clamp. Epoxy was used to secure the Toothpick's Bluetooth antennae to the project board. The developed system was mounted and tested in an industrial milling machine. It can be easily modified to fit any spindle.

The communication link between the PC and the Toothpick failed at speeds above 1450 RPM. Increasing the RPM of the Bluetooth-enabled spindle causes the quality of the communication link to deteriorate.

The data packet which was used to transmit the data was 19 characters (bytes) long. The packet's transmission time was transmitted and logged using LabVIEW 8.3.In practise these 19 characters can be used to transmit various data. Reading the thermocouple's temperature via the sensorboard was the actual bottleneck, and not the Bluetooth Communications Link.

10.3 PROBLEMS EXPERIENCED DURING THE RESEARCH

The experimental system had to be as small as possible, as it was fitted to a rotating spindle. When an object is spun around a fixed centre, a force starts pulling the object, not towards but away from the centre. This effect is referred to as a centripetal force, which is actually a manifestation of the object attempting to travel in a straight line due to inertia. Because the object constantly changes direction (as it is moving in a circular path), the object has an acceleration which is called centripetal acceleration. It is expressed as

$$\begin{aligned}
&a_c = \frac{v}{r} \\
&where \\
&a_c = centripetal\ accelration\ (m/s^2) \\
&v = velocity\ (m/s) \\
&r = circular\ radius\ (m)
\end{aligned}$$

According to Newton's 2nd law of motion the centripetal force can be expressed as

$$\begin{aligned}
&F_c = ma_c \\
&\quad = m\frac{v^2}{r} \\
&where \\
&F_c = centripetal\ force\ (N) \\
&m = mass\ (kg)
\end{aligned}$$

From this, it can be seen that the mass of the rotating object as well as the radius from the centre point should be kept to a minimum in order to minimise the centripetal force experienced by the spinning electronic equipment. It is also preferable to use surface-mounted (SMT technologies) components instead of through-hole (DIP packages). Another point that needs considering is to equally balance the weights of the components mounted on the mounting board, otherwise the spindle will experience unbalanced forces, causing it to wobble which might cause damage to the workpiece.

Four 1.5V rechargeable batteries were used to power the experimental setup. These might be heavy compared to other batteries, but they proved adequate.

The biggest problem encountered was the loss of the Bluetooth communication link between the experimental setup and the PC. The highest RPM achieved with a working communication link was 1450 RPM. Once the RPM was increased above this RPM the communication's link failed. This is attributed to the increase in the BER with RPM (reaching a high of 0.02) and a decrease in the RSSI with RPM (reaching a low of -8).

The angle between the experimental setup also affects the quality of the Bluetooth communication link. It was found that the highest quality was achieved with the PC's Bluetooth antennae directly below (0°) the experimental setup which is not ideal because the workpiece (being worked on) would be mounted there.

Different methods of transmitting data were tested and compared. Batch processing provides the best performance. The time to transmit (Tt) was used to compare the different results. The transmission time was gained by resetting the Toothpick's hardware timers as soon as it received the command. These timers are then used to see how fast the transmission command is completed. The value of these timers are transmitted and logged using a LabView program running on Windows XP.

The following points highlight what could have been done better or evaluated:

- The effect of a higher or lower baud rate on the quality of the Bluetooth link at higher RPM.

- The effect of gasses or smoke between the experimental setup and the PC's Bluetooth antennae.
- The effect of changing the output power of the Bluetooth's power on the quality and distance of the Bluetooth link.
- Placement of the experimental setup's Bluetooth antennae – horizontal or vertically.
- How distance affects the quality of the Bluetooth link at higher RPM.
- Developing an application to log data using a cell phone.
- Developing a control algorithm to use the data obtained in a viable control application.
- Using different features of the Bluetooth stack like streaming of audio.

10.4 CURRENT LIMITATIONS

Owing to the physical nature of the project, some limitations exist. The project was developed to be as generic as possible (requiring only a small modification to fit any rotational process), battery operated, low-cost, Bluetooth-enabled, monitoring device capable of being spun at high RPMs. Some of the limitations are:

- The firmware, which was developed, uses FlexiPanel's standard services to communicate data. By modifying these services the performance can be altered. Once altered the software will need to be evaluated and re-qualified with the Bluetooth standard.

- The physical limitations which arise with spinning electronic components like PCBs, Bluetooth antennae and batteries, limit the performance of the system.
- Not having a Bluetooth module, specifically designed for rotational applications.

10.5 FUTURE DEVELOPMENTS AND IMPROVEMENTS

Wireless rotational process monitoring systems have obvious advantages due to their adaptability to various manufacturing processes (such as FSW, FSP, FSSW, drilling and milling). They could also be incorporated into closed-loop control systems for these processes, as they could provide valuable information, by being able to measure as close as possible to the actual process.

Some suggestions for future developments and improvements are to:

- use the next, faster Bluetooth standard v3.0 + HS for the communication between the spindle and the PC.
- develop a control algorithm for implementation in a real industrial control application.
- use actual Bluetooth signal monitoring hardware to accurately log the signals strength and quality. The Toothpick's reading of its own BER and RSSI was used in this project.
- use a Bluetooth radio with smaller communication latency (between µP to µP it should be less than 8ms).

- use the audio streaming capability to transmit data. The PCM (Pulse Code Modulation) audio connections provide an audio interface when used with a suitable audio codec.
- use the RF characteristics of the environment surrounding the Bluetooth module. Little or no metal should be used if possible for the casing of the experimental setup.

10.6 CONCLUSION

This project has shown how a Bluetooth-enabled monitoring system was designed for a rotational process. This system has the ability to monitor a system without being physically connected to it. An 8-bit microcontroller is used to read the sensor's data and to transmit it via Bluetooth to a PC. The microcontroller and the sensor can be reprogrammed to suit the application. The microcontroller can be reprogrammed wirelessly.

The aim of this research was to design a generic, low-cost and robust monitoring system for a rotational system. According to the tests undertaken, this aim has been achieved.

APPENDIX A:

REFERENCES

1. A. Rivero, L.N. López de Lacalle and Ma Luz Penalva (2008). Tool wear detection in dry high-speed milling based upon the analysis of machine internal signals, *Mechatronics* 18, 627–633
2. Arul, S., Kruger, G.H., Miller, S.F., Pan, T. & Shih, A.J. (2008). Spot friction weld strength improvement through in-process metal matrix formation, *Proceedings of the 2008 International Manufacturing Science and Engineering Conference* x, 07-Jan
3. Benny Endelt, Karl Brian Nielsen and Joachim Danckert (2006). New framework for on-line feedback control of a deep-drawing operation, *Journal of Materials Processing Technology* 177, 426–429
4. Blignault C., Hattingh, D.G.,Kruger, G.H.,van Niekerk, T.I.,James, M.N., (2006). Friction stir weld process evaluation by multi-axial transducer, *Measurement*, 41, 32-43
5. Buffa, G., Donati, L., Fratini, L. & Tomesani, L. (2006). Solid state bonding in extrusion and FSW: Process mechanics and analogies, *Journal of Materials Processing Technology* 177, 344-347
6. Buffa, G., Fratini, L. & Piacentini, M. (2008). On the influence of tool path in friction stir spot welding of aluminum alloys, *Journal of materials processing technology* 208, 309-317
7. D. Cuppini, G.D'errico and G. Rutelli (1990). Tool Wear Monitoring Based On Cutting Power Measurement, *Wear* 139, 303-311
8. D.E. Dimla Snr. (2000). Sensor signals for tool-wear monitoring in metal cutting operations—a review of methods, *International Journal of Machine Tools & Manufacture* 40, 1073–1098
9. Dae Kyun Baek, Tae Jo Ko and Hee Sool Kim (2000). Real time monitoring of tool breakage in a milling operation using a digital signal processor, *Journal of Materials Processing Technology* 100, 266-272
10. De Silva, C.W. (2007). "Sensors and Actuators - Control System Instrumentation". USA: CRC Press
11. Dunn, W.C. (2006). "Introduction to Instrumentation, Sensors, and Process Controll". Norwood, USA: Artech House, Inc.
12. Ellis, P. (1999). "Pic Trix - An introduction to PIC Microcontrollers Using the 16F877". USA: P. Ellis
13. Erkki Jantunen (2002). A summary of methods applied to tool condition monitoring in drilling, *International Journal of Machine Tools & Manufacture* 42, 997–1010
14. Erwin Schoitsch (1986). Software Engineering Aspects Of Real-Time Programming Concepts, *Computer Physics Communications* 41, 327—361
15. Fiona Zhao, Xun Xu and Shane Xie (2008). STEP-NC enabled on-line inspection in support of closed-loop machining, *Robotics and Computer-Integrated Manufacturing* 24, 200–216
16. Fraden, J. (2004). "Handbook of Modern Sensors - Physics, Designs, and Applications". New York, USA: Springer
17. Fratini, L., Buffaa, G. & Shivpuri, R. (2007). Improving friction stir welding of blanks of different thicknesses, *Materials Science and Engineering* 459, 209-215
18. G. Byrne, D. Dornfeld, I. Inasaki, G. Ketteler, W. Konig and R. Teti (1995). Tool Condition Monitoring (TCM) - The Status and Industrial Application, *Research Annals of the CIRP* 44, 541-566
19. G.E. D'Errico (1998). An adaptive system for turning process control based on tool temperature feedback, *Journal of Materials Processing Technology* 78, 43–47

20. Gi-Bum Jeonga, Dong Hwan Kima and Dong Young Jangb (2005). Real time monitoring and diagnosis system development in turning through measuring a roundness error based on three-point method, *International Journal of Machine Tools & Manufacture* 45, 1494–1503
21. H. Hadeby and G. Sohlenius (1989). Increasing Availability and Efficiency by Monitoring the Cutting Process in a Lathe During Production with Limited Manpower, *Annals of the CIRP* 38, 393-397
22. Hakan Hjalmarsson (2005). From experiment design to closed-loop control, *Automatica* 41, 393 – 438
23. Hovanski,Y., Santellab,M.L. & Grant, G.J. (2007). Friction stir spot welding of hot-stamped boron steel, *Scripta Materialia* 57, 873-876.
24. Huddleston, C. (2007). "Intelligent Sensor Design - Using the dsPIC". Burlington, USA: Elsevier
25. I.D. Landau (2001). Identification in closed loop: a powerful design tool (better design models, simpler controllers), *Control Engineering Practice* 9, 51-65
26. I.N. Tansel, W.Y. Bao, N.S. Reen & C.V. Kropas-Hughes(2005). Genetic tool monitor (GTM) for micro-end-milling operations, *International Journal of Machine Tools & Manufacture* 45, 293–299
27. Jian-Qing Sua, Tracy W. Nelson & Colin J. Sterling (2005). Microstructure evolution during FSW/FSP of high strength aluminum alloys, *Materials Science and Engineering* 405, 277-286
28. Jong-Geun Choi and Min-Yang Yang (1999). In-process prediction of cutting depths in end milling, *International Journal of Machine Tools & Manufacture* 39, 705–721
29. Kam-Yiu Lam, Tei-Wei Kuo, Ben Kao, Tony S.H. Lee, Reynold Cheng (2002). Evaluation of concurrency control strategies for mixed soft real-time database systems, *Information Systems* 27, 123-146
30. Kim, Y.G., Fujii, H., Tsumura , T., Komazaki , T. & Nakata , K. (2006). Effect of welding parameters on microstructure in the stir zone of FSW joints of aluminum die casting alloy, *Materials Letters* 60, 3830-3837
31. Krishnamachari, B. (2005). "Networking Wireless Sensors". United Kingdom: Cambridge University Press
32. Lathabai, S., Painter, M.J., Cantin, G.M.D. & Tyagi, V.K. (2006). Friction spot joining of an extruded Al–Mg–Si alloy, *Scripta Materialia* 55, 899-902
33. Lombard, H., Hattingh, D.G., Steuwer, A. & James, M.N. (2007). Optimising FSW process parameters to minimise defects and maximise fatigue life in 5083-H321 aluminium alloy,*Engineering Fracture Mechanics*75, 341-354
34. M. De la Sen (2009). Robust adaptive control of linear time-delay systems with point time-varying delays via multiestimation, *Applied Mathematical Modelling* 33, 959–977
35. M. Milfelner, F. Cus and J. Balic (2005). An overview of data acquisition system for cutting force measuring and optimization in milling, *Journal of Materials Processing Technology* 164, 1281–1288
36. Mandal, S., Rice, J. & Elmustafa, A.A. (2008). Experimental and numerical investigation of the plunge stage in friction stir welding, *Journal of materials processing technology* 203, 411-419
37. Mathieu Ritou, Sebastien Garnier, Benoit Furet and Jean-Yves Hascoet (2006). A new versatile in-process monitoring system for milling, *International Journal of Machine Tools & Manufacture* 46, 2026–2035
38. Mathon, M.H., Klosek, V., de Carlan, Y. & Forest, L. (2009). Study of PM2000 microstructure evolution following FSW process, *Journal of Nuclear Materials* x x-x
39. Mitlin, D.Radmilovic, V., Panc, T., Chena, J., Feng, Z., Santella, M.L. (2005). Structure–properties relations in spot friction welded (also known as friction stir spot welded) 6111 aluminum, *Materials Science and Engineering* 441, 79-96
40. Mohamed F. Younis, Mohamed Aboutabl, Daeyoung Kim (2004). Software environment for integrating critical real-time control systems, *Journal of Systems Architecture* 50, 649–674

41. Mohieddine Jelali (2006). An overview of control performance assessment technology and industrial applications, *Control Engineering Practice* 14, 441–466
42. Moreira, P.M.G.P., de Figueiredo, M.A.V. & de Castro, P.M.S.T. (2007). Fatigue behaviour of FSW and MIG weldments for two aluminium alloys, *Theoretical and Applied Fracture Mechanics* 48, 169-177
43. Ph. Poignet, M. Gautier, W. Khalil and M.T. Pham (2002). Modeling, simulation and control of high speed machine tools using robotics formalism, *Mechatronics* 12, 461–487
44. R. Nandan, T. DebRoy & H.K.D.H. Bhadeshia (2008). Recent advances in friction-stir welding – Process, weldment structure and properties, *Progress in Materials Science* 53, 980-1023
45. R.S. Mishraa & Z.Y. Mab (2005). Friction stir welding and processing, *Materials Science and Engineering* 50, Jan-78
46. Raghavendra, C.S., Sivalingam, K.M. & Znati,T. (2006). "Wireless Sensor Networks". New York, USA: Springer
47. Reynolds, A.P. (2008). Flow visualization and simulation in FSW, *Scripta Materialia* 58, 338-342
48. Ricardo Campa, Rafael Kelly and Victor Santibanez (2004). Windows-based real-time control of direct-drive mechanisms: platform description and experiments, *Mechatronics* 14, 1021-1036
49. Rodolfo Haber-Haber, Rodolfo Haber, Michael Schmittdiel and Raul M. del Toro (2007). A classic solution for the control of a high-performance drilling process, *International Journal of Machine Tools & Manufacture* 47, 2290–2297
50. S. Lakshiminarayanan, G. Emoto, S. Ebara, K. Tomida and Sirish L. Shah (2001). Closed loop identification and control loop reconfiguration: an industrial case study, *Journal of Process Control* 11, 587-599
51. S. Scott Collis, Ronald D. Joslin, Avi Seifert and Vassilis Theofilis(2004). Issues in active flow control: theory, control, simulation, and experiment, *Progress in Aerospace Sciences* 40, 237–289
52. T, W. Rauber, M. M. Barata and A. S. Steiger-Garcao (1994). Learning of Specific Process Monitors in Machine Tool Supervision, *Artificial Intdligence in Real Time Control* 105-110
53. Threadgill, P.L. & Nunn, M.E. (2003). A review of friction stir welding: Part 1, process review, Cambridge: The Welding Institute, United Kingdom
54. Xavier Desforges, Bernard Archime`de (2006). Multi-agent framework based on smart sensors/actuators for machine tools control and monitoring, *Engineering Applications of Artificial Intelligence* 19, 641–655
55. Xu, W., Liu, J., Luan, G. & Dong, C. (2008). Temperature evolution, microstructure and mechanical properties of friction stir welded thick 2219-O aluminum alloy joints, *Materials and Design* x, x-x
56. Yao-Wen Hsueh and Chan-Yun Yang (2009). Tool breakage diagnosis in face milling by support vector machine, *Journal of materials processing technology* 209, 145–152
57. Zhang, Z. & Zhang, H.W. (2008). Numerical studies on controlling of process parameters in friction stir welding, *Journal of materials processing technology* 209, 241-270