INTELLIGENT MONITORING AND CONTROL SYSTEM FOR A FRICTION STIR WELDING PROCESS

By

Grant Kruger, B.Tech. Elec.Eng.

Submitted in compliance with the requirements for the

Magister Technologiae (Electrical Engineering) Degree

at Port Elizabeth Technikon

Date of Submission	Promoters:
20 January 2003	Dr T.I. van Niekerk
	Dr D.G. Hattingh
	Mr F Adlam

Grant Kruger hereby declares that:

- At no time during the registration for the degree of Magister Technologiae has the author been registered for any other university degree.
- The work done in the thesis is his own; and
- All sources used or referred to have been documented and recognized.

	20/01/2003
Signature	Date

I would like to acknowledge the following for their contributions:

- My promoters, Dr van Niekerk, Dr Hattingh and Mr Adlam for their tireless support, guidance and encouragement

- All the team members in the Manufacturing Technology Research Center

- The National Research Foundation and research committee of the Port Elizabeth Technikon for their financial assistance

- My family and friends for their understanding, support, encouragement and love

- The Lord Jesus Christ for giving me strengh, knowledge, will power, guidance and assistance throughout my life

Abstract

A Friction Stir Welding machine is proposed and built to allow future research into the process and to provide a framework from which the application of intelligent manufacturing to industrial processes can be investigated. Initially a literature survey was conducted upon which the design of the machine could be based. The conversion of a conventional milling machine into a Friction Stir Welding machine by applying modern monitoring and control systems is then presented. Complete digital control was used to drive actuators and monitor sensors. A wireless chuck mounted monitoring system was implemented, enabling forces, torques, temperature and speed of the tool to be obtained directly from the process. Software based on a hierarchical Open Systems Architectural design, incorporating modularity, interoperability, portability and extensibility is implemented. This experimental setup is used to analyze the Friction Stir Welding process by performing data analysis using statistical methods. Three independent variables (weld speed, spindle speed and plunge depth) were varied and the independent variables (forces, torques, power, temperature, speed, etc) recorded using the implemented software. The statistical analysis includes the analysis of variants, regression analysis and the creation of surface plots. Using these results, certain linguistic rules for process control are proposed. An intelligent controller is designed and discussed, using the derived rules to improve and optimize certain aspects of the process encountered during the experimental phase of the research.

Table of Contents

Abbreviations

AC	Adaptive control
AI	Artificial Intelligence
ANOVA	Analysis of variants
ANSI	American National Standards Institute
API	Application program interface
ASCII	American Standard Code for Information Interchange
bps	Bits per second
CAD	Computer aided design
CAM	Computer aided manufacturing
CIM	Computer integrated manufacturing
CNC	Computer numerical controller
CoA	Center of area
CoG	Center of gravity
COID	Connection identification
CoM	Center of maximum
CPU	Central processing unit
DAQ	Data acquisition
DNS	Domain name service
DOM	Document object model
DoM	Degree of membership

DoS	Degree of support
EMF	Electromotive force
FET	Field effect transistor
FIFO	First in first out
FLC	Fuzzy logic controller
FLIPS	Fuzzy logic inferences per second
FSW	Friction Stir Welding
GUI	Graphical user interface
HAZ	Heat affected zone
HTL	High Threshold Logic
HTML	Hypertext markup language
I/O	Input/output
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPC	Inter-process communication
IRQ	Interrupt request
IS	Information systems
ISO	International Organization for Standardization
IT	Information technology
LAN	Local area network
LCD	Liquid crystal display
LED	Light emitting diode

LMI	Libra Measuring Instruments
MBF	Membership function
MIG	Metal Inert Gas
MIMO	Multiple input multiple output
MMU	Memory management unit
MoM	Mean of Maximum
NC	Numerical control
OA	Orthogonal analysis
OEM	Original Equipment Manufacturer
OO	Object orientated
OOP	Object orientated programming
OS	Operating system
OSA	Open Systems Architecture
OSI	Open Systems Interconnection
PCI	Peripheral Component Interconnect
PID	Proportional integral derivative
POSIX	Portable operating system interface
PTC	Positive Temperature Coefficient
RAM	Random access memory
RPM	Revolutions per minute
SISO	Single input single output
SMP	Symmetrical multiprocessor
TCP	Transport control protocol

TTL	Transistor-transistor logic
TWI	The Welding Institute
UART	Universal asynchronous receiver transmitter
UI	User interface
UML	Unified Modeling Language
URL	Universal resource locator
VAC	Volts alternating current
VME	Versa Module European
XML	Extensible Markup Language
YAG	Yttrium aluminum garnet

Glossary of Terms

Abstract machine model	Used to model the interfacing of sub-systems. Organizes the system into a set of layers (or abstract machines) each of which provides a set of services. This supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.
Abstraction	A model or simplification of a physical object or concept by ignoring the details of a problem and concentrating on the solution.
ActionScript	Macromedia Flash uses the ActionScript scripting language to add interactivity to a movie. Similar to JavaScript, ActionScript is an object-oriented programming language.
Adaptive control (AC)	A control system that senses one or more parameters and utilizes sensor data to adjust control signals to meet performance criteria. This is done by automatically changing the type or influence of control parameters to improve overall control systems performance.

Artificial intelligence (AI)	AI is the mimicking of human thought and cognitive processes to solve complex problems automatically. It uses techniques for writing computer code to represent and manipulate knowledge. Different techniques mimic the different ways that people think and reason. AI applications can be either stand-alone software, such as decision support software, or embedded within larger software or hardware systems.
Automatic control (AC)	Control whereby an intelligent system calculates and commands a control action by the process.
Architectural design	An early stage of the system design process. Representing the link between specification and design processes. It is often carried out in parallel with some specification activities and involves identifying major system components and their communications.
American Standard Code for Information Interchange (ASCII)	This is the worldwide standard for the code numbers used by computers to represent all the upper and lower-case letters, numbers, punctuation, etc. There are 128 standard ASCII codes each of which can be represented by a 7 digit binary number: 0000000 through 1111111.
Backing plate	A layer of material that is placed below the joint interface of the materials to be welded. It provides a surface to oppose

	the vertical downward force on the material and protects the machines bed.
Backlash	Rotational movement due to clearance between transmission elements. In reducers, upon reversing direction of the input shaft, there is no immediate rotation of the output shaft.
Baud rate	Baud rate is the number of times per second that a carrier signal shifts value.
Blocking	A means for threads to synchronize with other threads or events. In the blocking state, a thread doesn't consume any CPU time. When the event occurs the thread was waiting for, it is unblocked and is able to consume CPU time again.
Computer aided design (CAD)	A computer technology which allows two and three-dimensional designs to be produced on a computer screen where they can be quickly and easily modified. It also allows computerized simulation for testing of designs.
Computer aided manufacture	The extension of a CAD design to include the determining of the quantities of materials needed and the shop-floor

(CAM)	instructions required for production of an item.
Closed-loop control	A control system in which a sensor is used to provide feedback to a controller, informing it how well a process is performing. This is information is used to adjust some variable of the process, therefore "closing the feedback loop"
Computer Integrated Manufacturing (CIM)	Manufacturing has reached a stage where design, planning, simulation, machining, sensing, control, monitoring and assembly, as well as maintenance, management and customer service are integrated by means of computers and their network systems.
Computerized Numerical Control (CNC)	Enhanced numerical control (NC) machines have a programmable computer built into them giving them greater flexibility. This allows for a range of different programs to be used and for these programs to be changed easily, resulting in computerized numerical control.
Contactor	A contactor is an electrically operated switch usually used in control circuits, whose contacts are considered high amperage, compared to a relay.

Context switches	The process by which the operating system goes through to change from one process being the current/active one to another.
Correlation coefficient	The correlation coefficient (r) provides an index of the degree to which a pair of measured variables co-varies in a linear fashion.
Data server	A computer, or a software package, that provides specific information to client software running on other computers.
Defuzzification	The process of transforming the fuzzy output of a fuzzy inference system into a crisp output.
Degree of membership (DoM)	It is the degree to which a value satisfies the definition of the linguistic variable, which a membership function describes. The output value of a membership function is always limited to between 0 and 1. It is also known as a membership value or membership grade.
Degree of support (DoS)	The degree to which the antecedent part of a fuzzy rule is satisfied. The degree of support may be the result of an

	AND or an OR operation, and it shapes the output function for the rule. Also known as degree of fulfillment.
Dependent variable	The outcome that we are trying to explain in a regression analysis. Any change in the dependent variable is viewed as a function of changes in the independent variable(s).
Dwell period	A time period in the FSW process after the plunging of the tool's pin is required to ensure the material to be welded is sufficiently plasticized before the weld traverse begins.
Dynamometer	An apparatus for measuring force or power developed by a motor, or that is required to operate machinery.
Electromagnetic interference (EMI)	EMI consists of low frequency electromagnetic waves that emanate from electromechanical devices. An electromagnetic disturbance caused by such radiating and transmitting sources, such as heavy-duty motors and power lines, can induce unwanted voltages in electronic circuits, damage components and cause malfunctions.
Encapsulation	Dividing a program up into separate compartments by enclosing related data, routines and definitions into a

	modular unit.
Expert system	Systems in which human expertise is held in the form of rules, which enable the system to diagnose situations without the human expert being present.
Fatigue strength	Maximum stress that a material will endure without failure for a specified number of load cycles.
Firewall	A firewall protects a computer network from unauthorized access. Firewalls may be hardware devices, software programs, or a combination of the two. A firewall typically guards an internal network against malicious access from the outside; however, firewalls may also be configured to limit access to the outside from internal users.
Friction Stir Welding (FSW)	Friction Stir Welding was developed by The Welding Institute (Cambridge, UK) in 1991. With this process, a rotating pin is pressed into the interface of the components to be joined. Frictional heat causes the material to soften without reaching the melting point. The material extrudes around the pin as it traverses the joint and is forged by downward pressure from the pin’s shoulder.

Full-duplex	Full duplex communication allows information to be carried in both directions over a communications channel simultaneously.
Fuzzification	The conversion of a numerical (crisp) value into degrees of membership for the membership functions defined for the linguistic variable.
Fuzzy logic	A superset of conventional (Boolean) logic that recognizes more than simple true and false values. With fuzzy logic, members of a set have a "degree of membership" that (typically) ranges from 0-100%. This is used to transform crisp values into linguistic terms, by a process called fuzzyfication. These terms are then processed using a rule base. The linguistic output is then converted to crisp values once again using a process called defuzzification. Dr. Lotfi Zadeh of UC/Berkeley introduced fuzzy logic in the 1960's as a means to model the uncertainty of natural language.
Fuzzy rule base	A group of IF-THEN rules using linguistic variables for representing the knowledge of a system.

Fuzzy set	In traditional logic, sets are “crisp”, either you belong 100% to a set or you do not. A set of tall people may consist of all people over six feet tall, anyone less than six feet is “short” (or more appropriately “not tall”). Fuzzy logic allows sets to be “fuzzy” so anyone over six feet tall may have 100% membership in the “tall” set, but may also have 20% membership in the “medium height” set.
Genetic Algorithms	Search algorithms used in machine learning which involve iteratively generating new candidate solutions by combining two high scoring earlier (or parent) solutions in a search for a better solution. So named because of its reliance on ideas drawn from biological evolution.
Grain refinement	The reducing of the crystalline structure of a material as observed by the eye or under a microscope.
Graphical user interface (GUI)	A type of user interface where the user controls the operation of a piece of software by interacting with graphical elements on the display. These graphical elements may include menus and icons. Other important features of a graphical user interface are window techniques and the use of a mouse or other pointing device. Graphical user interfaces are used by most modern software.
Hard real-time	The term "hard real-time system" specifies that the

	response from a system has to be within the deadline 100% of the time.
Half-duplex	Pertains to communications in which data can be sent in only one direction at a time.
Hard-stop	A protection method typically used with moving machinery to detect and prevent out of range motion, protecting operators and equipment, in the event of a failure or malfunction.
Heat affected zone (HAZ)	This area in a FSW weld will lie close to the weld center and has experienced a thermal cycle, which has modified the microstructure and/or the mechanical properties of the material. There is no plastic deformation occurring in this area.
High level language	A programming language that utilizes macroinstructions and statements that closely resembles human language or mathematical notation to describe the problem to be solved or the procedure to be used. Also called a compiler language.

Hooke's law	Stress is directly proportional to strain. Hooke's law assumes perfectly elastic behavior and does not take into account plastic or dynamic loss properties.
Hypertext Markup Language (HTML)	A popular page description language for creating hypertext and hypermedia documents for the World Wide Web and intranet websites.
Independent variable	The factor that is used as a predictor in a regression analysis. The independent variable is conceptualized as accounting for changes in the dependent variable.
Induction motor	Abbreviation for three-phase induction motor, also known as the asynchronous motor. It is the electrical machine most commonly used in drive systems. The stator iron and the rotor core are both made from magnetic sheet material. The motor shows magnetic symmetry because stator has a three-phase symmetric winding and the rotor construction is also symmetric. Depending on the rotor design, there are squirrel-cage and wound rotor (or slip-ring) motors.
Inference engine	The part of an expert system responsible for drawing new conclusions from the current data and rules. The inference engine is a portion of the reusable part of an expert system (along with the

	user interface, a knowledge base editor, and an explanation system) that will work with different sets of case-specific data and knowledge bases.
Instrumented chuck	A specially designed chuck fitted with various sensors, sampling and communication electronics to feedback information regarding the machining process, such as force, speed, torque and temperature during a process.
Intelligent system	An automated system designed to process information and make decisions using written rules that mimic the way a human would work. Intelligent systems can be used to monitor physical processes in real time and make critical decisions in the absence of human interaction. They are also used to help humans decide on a course of action based on a number of existing conditions, such as in medical diagnoses. Some systems are programmed to learn from errors.
Intelligent manufacturing system (IMS)	An intelligent manufacturing system takes intellectual activities in manufacturing and uses them to better harmonize human beings and intelligent machines integrating the entire corporation from marketing through design, production and distribution, in a flexible manner that improves productivity. IMS's possess the innate ability to

	respond, promptly and correctly, to changes in requirements. They differ from conventional manufacturing systems - even advanced ones - in their inherent capability to adapt to changes without external intervention.
Interface cards	Common term used to describe a circuit board designed to plug into a computer’s expansion slot to allow the connection to another peripheral.
Interlock	A safety device or system that prevents certain conditions occurring until required events or states are met.
Interoperability	The ability of two or more systems to exchange information and to mutually use the information that has been exchanged.
Inter-process communication (IPC)	The exchange of data between one process and another, within a computer or between networked computers.
Inverter	A term commonly used for an AC adjustable frequency drives. An inverter is also a term used to describe a

	particular section of an AC driver. This section uses the DC voltage form a previous circuit stage (DC link) to produce an AC current.
Internet Protocol address (IP address)	A means of referring to locations on the Internet. Composed of four numbers from 0 through 255, separated by decimal points. All machines on the Internet have one, often assigned by the Internet service provider at connection time.
Interrupt request (IRQ)	One of several control lines into a computer's CPU to provide a means for hardware components such as disk controllers, printers, and modems to gain the attention of the CPU.
Joint interfaces	This is the area formed between the butted surfaces of the two materials that are to be welded.
Knowledge based systems (KBS)	A computer based system in which there is a symbolic representation of the knowledge in the system and it is separated from the inferencing mechanism in the system. A knowledge-based system has a typical architecture, consisting of a knowledge base (where the knowledge is

	stored), an inferencing engine and a working memory where the initial data and intermediate results are stored.
Linguistic variable	A set of linguistic interpretations describing a range of values (a universe of discourse). The range of values described is termed the base variable.
Local area network (LAN)	Linked computers and peripherals, usually in the same building, using generally a single network standard and owned and controlled by the organization that uses it. The main advantages are the ability to share resources, communications, and transfer data.
Machine intelligence	Comprise various intelligent techniques (fuzzy logic, neural networks etc) and expert criteria (operator knowledge), with one of more higher resolution levels (hierarchical levels), which together manipulate process conditions resulting in a change of the process parameters.
Machine limitations	A machines mechanical design and structure will impose certain physical limitations. Exceeding these limitations could result in damage to the machine.

Macromedia Flash	Flash is a multimedia format developed specifically for use on the Web. It uses vector graphics and allows for complete control and synchronization of sound and animation. A company called Macromedia created Flash.
Macrostructure	The general crystalline structure of a metal and the distribution of impurities seen on a polished or etched surface by either the naked eye or under low magnification of less than x10.
Mechanical properties	Those properties that reveal the reaction, either elastic or plastic, of a material to an applied stress. Tensile strength, yield strength, elongation, and reduction of area, hardness, impact strength, and bend ability are mechanical properties.
Membership function (MBF)	A function that specifies the degree to which a given input belongs to a set or is related to a concept.
Message loop	A message loop will get a message from a message queue, process it, get the next message, process it and so on. When the input queue is empty, the execution of the message loop thread will be suspended until a message

	becomes available.
Message-passing	A parcel of bytes is passed from one process to another. Only the sender and receiver processes need to understand the meaning of the bytes. Message passing provides a means for synchronizing the execution of several processes.
Metal inert gas welding (MIG welding)	An arc welding process that joins metals by heating them with an electric arc. The arc is between a continuously fed filler metal (consumable) electrode and the workpiece. Externally supplied gas or gas mixtures provide shielding. Common MIG welding is also referred to as short circuit transfer. Metal is deposited only when the wire actually touches the work. No metal is transferred across the arc. Another method of MIG welding, spray transfer moves a stream of tiny molten droplets across the arc from the electrode to the weld puddle.
Milling machine	A machine that uses a spinning tool to cut shapes and patterns into a material. Either the material or the tool is moved.

Motor cowling	A removable metal covering, located on the back of the motors of the FSW machine, used to protect the optical encoder and guide airflow through the motor's cooling vanes.
Multithreading	A style of programming that allows many separate threads of control inside one process. Multithreading is often used to minimize overhead when a number of closely-cooperating, concurrently-executing, program fragments do not need the full weight of normal inter-process protection from the OS, and indeed where the overhead of providing such protection is prohibitive.
Neural-fuzzy	Neural fuzzy systems are artificial neural networks with fuzzy input/output information.
Neural network	Neural networks are an approach to machine learning, which developed out of attempts to model the processing that, occurs within the neurons of the brain. By using simple processing units (neurons), organized in a layered and highly parallel architecture, it is possible to perform arbitrarily complex calculations. Learning is achieved through repeated minor modifications to selected neurons, which results in a very powerful classification system.
Neutrino	QNX Neutrino is a microkernel real-time operating system that makes fault-resilient, field-upgradeable systems much easier to design and implement.

Numerical control (NC)	Numerically controlled machines are operated by a series of coded instructions often on paper or magnetic tape.
Object-Oriented Design (OOD)	The design principle that uses classes, objects, inheritance, encapsulation, and data hiding.
Object-Orientated Programming (OOP)	Programming using classes, objects, inheritance, encapsulation, and data hiding.
Open architecture	An architecture whose specifications are public. This includes officially approved standards as well as privately designed architectures whose specifications are made public by the designers. The opposite of open is closed or proprietary.
Optical encoder	This feedback device is used to detect rotary or linear position and convert it to an electrical output. A light source, usually either an LED or a laser, is projected through thin slits in a rotary disc for rotary encoders, or a

	thin tape scale for linear. The LED is adequate for most applications, although the laser has found niches in several high precision, high-resolution applications. The disk and tape can either be made of covered glass with thin etchings in the cover, or thin metal with etchings through it. Each has appropriate applications. As light is transmitted, a photoreceptor on the opposite side of the disc or tape detects the light and converts it to an electrical output.
Orthogonal array experiment	A balanced evaluation whereby the average effect of a factor is determined while the levels of all other factors in the design are systematically changed.
Orthogonal array	A table consisting of rows and columns with the property that for any pair of columns (factors) all combinations of values (levels) occur, and further, all combinations occur the same number of times.
Parent metal	It may have experienced a thermal cycle from the weld, but is not affected by the heat in terms of microstructure or mechanical properties
PCI bus	This is a 32-bit bus architecture (64-bit with multiplexing)

	that was developed by DEC, IBM, Intel, and others. It is widely used in Pentium-based PCs. A PCI bus provides a high-bandwidth data channel between system board components such as the CPU and devices such as hard disks and video adapters.
Proportional, integral, derivative (PID) controller	This is the most accurate type of basic control. If set correctly (either manually or by auto tuning) the controller will aim to get the process variable to the setpoint as quickly and smoothly as possible, whilst also maintaining a stable control at setpoint value, with a fast response to any deviation. A correctly set/tuned PID controller will have a limited over- and undershoot and minimum error from setpoint.
Plasma-keyhole weld	This process uses a CO_2 laser to produce keyhole welds. Key factors are beam mode quality and laser power. A low M^2 beam can be focused to produce maximum power density in the center of the beam. With sufficient laser power, the power density overcomes losses from reflectivity and thermal diffusivity and a keyhole weld is begun. During laser welding plasma of metal vapor forms above the

	area where the focused beam impinges on the metal surface. This plasma absorbs the laser energy, and reduces the energy transmitted to the surface of the material. As laser power levels increase, it becomes necessary to suppress the plasma reaction with a shielding gas of helium or a mixture of helium and argon. In addition, the shielding gas prevents oxidation and pinholes in the weld zone, a requirement for high strength, hermetic welds.
Porosity	Voids or pores occurring in materials resulting from trapped gas, or shrinkage during solidification.
Portability	The same system components must operate on different hardware platforms. This allows the system to move to whichever hardware platform best suits the application, while using the same system components.
Portable Operating System Interface (POSIX)	The main goal being to support application portability at the source-code level. It covers various aspects of operating systems and is continually evolving and is currently being standardized by the Computer Society of IEEE as the IEEE standard P1003. The ISO/IEC is also busy with the international standard ISO/IEC-9945.

Probability	A probability provides a quantitative description of the likely occurrence of a particular event. Probability is conventionally expressed on a scale from 0 to 1; a rare event has a probability close to 0, a very common event has a probability close to 1.
Process	1) An instance of a program that is being executed. It is a non-schedulable thread container that occupies memory. Each process has a unique PID, which is that process's entry in the kernel's process table. Also referred to as a 'task'. 2) A series of steps resulting in an output (e.g. a product).
Process control	A system of measurements and actions within a process intended to insure the output of the process conforms to pertinent specifications.
Process manager	The process manager is provided by the operating system and it responsible for creating new processes in the system and managing the most fundamental resources associated with a process. These services are all provided via

	messages.
Protocol	A protocol is the standard or set of rules that two devices use to communicate with each other. Also known as a communications protocol. Any product that uses a given protocol should work with any other product using the same protocol.
Pulse	A pulse is a special notification mechanism used in the QNX operating system whereby a sender sends a message to a receiving process without waiting for acknowledgment.
RAM disk	A section of RAM set up to function like a hard disk however the information stored on the RAM disk will be lost during re-boot or power out.
Ram spindle	The convention-milling machine used to create the Friction Stir Welding machine has two spindles, the main spindle and the ram spindle. The ram spindle is used for vertical milling operations.
Real-time	The characteristic of determinism applied to computer hardware and/or software. A real-time process must

	perform a task in a determined length of time.
Real-time system	An application or group of applications with real-time requirements.
Real-time operating system (RTOS)	An operating system implementing components and services that explicitly offer deterministic responses, and therefore allow the creation of hard real-time systems. An RTOS is characterized by the richness of the services it provides, the performance characteristics of those services, and the degree to which those performance characteristics can be controlled by the application engineer (to satisfy the requirements of the application).
Resource	Any component of a computing machine that can be utilized by software. Examples include: RAM, disk space, CPU time, real-world time, serial devices, network devices, and other hardware, as well as OS objects such as semaphores, timers, file descriptors, files, etc.
Repeatability	A measure of the ability of a transducer to repeatedly produce the same measurement signal for the same measurand.

Residual stresses	Macroscopic stresses that are set up within a metal as the result of non-uniform plastic deformation. This deformation may be caused by cold working or by drastic gradients of temperature from quenching or welding.
Resource manager	In QNX, a resource manager is a user-level server program that accepts messages from other programs and performs some form of processing. Resource managers are used to extend the services of the operating system.
Router	Connected between networks, a router controls the flow packets, directing traffic according to routing tables it has been programmed with. It reads the addresses of each packet coming in, and decides what interface to send that packet onto. It can also block packets based on rules.
Server	A computer program or system that serves a client (another program or system) with information or processing power.
Scalability	Allows the systems functionality to be increased or decreased by modifying specific system components, avoiding the cost of replacing the entire system.

Scheduling	Determining when a given process or thread is to be run, within a multitasking environment. Real-time operating systems often use priority-driven preemptive scheduling. Other types of scheduling include round-robin (every process gets an equal-length turn) and FIFO.
Scripting language	A scripting language is a simple programming language used to write an executable list of commands, called a script. It is a high-level command language that is interpreted rather than compiled, and is translated on the fly rather than first translated entirely.
Soft real-time	Basically the equivalent of a hard real-time system, except that an occasional slightly late response is acceptable. A system that misses an occasional deadline does not really fit into what is perceived as real-time, but it might meet all of the actual time-response requirements of the system.
Signal	Signals are software interrupts supported by POSIX compliant operating systems. A signal can be generated at any time (asynchronously) or at specified times (synchronously). They are generated by the kernel as a

	result of an exceptional event (floating point error, hang-up etc), from the terminal/shell (when `^Z`, `^C`, or `kill` are typed) or from processes. Signals have almost no information content.
Signal conditioning	Amplifying, filtering, or otherwise processing a real-world signal to make it suitable for further processing by the system.
Soft-stops	Limit switches are placed at the maximum and minimum allowed safe travel positions. These are used to inform the control software if the bed is in its home position or exceeding its travel range. If either of these conditions occurs, the control software can take the appropriate action before the hard-stops are activated.
Software architecture	The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design. The output of this design process is a description of the software architecture.
Space delimited string	A standard ASCII human readable text string containing fields separated by spaces.

Squirrel cage induction motor	A type of rotor winding in which heavy conductors are imbedded in the rotor body. The conductors are shorted together at the ends by continuous rings. No insulation is required between the windings and the core. This type of winding is rugged, easily manufactured, and practically maintenance free. It is widely applied in AC induction motors. Physically, it appears as a rotating squirrel-cage, thus the name.
Statistics	Statistics is the mathematical science of utilizing data about a population in order to describe it usefully and to draw conclusions and make decisions. The population may be a community, an organization, a production line, a service counter, or a phenomenon such as the weather. Statisticians develop models based on probability theory. They determine which probability model is correct for a given type of problem and they decide what kinds of data should be collected and examined.
Strain gauge	A device with electrical resistance that is a function of the applied strain.

Switch	In telecommunications, a switch is a network device that selects a path or circuit for sending a unit of data to its next destination. In general, a switch is a simpler and faster mechanism than a router, which requires knowledge about the network and how to determine the route.
Synchronization services	Services offered by an operating system that allow ability to synchronize the activities of various threads.
System modules	A unit of a system designed for use with other modules, also called a component.
Transmission Control Protocol/Internet Protocol (TCP/IP)	This is the suite of protocols that defines the Internet. Originally designed for the UNIX operating system, TCP/IP software is now included with every major kind of computer operating system. To be truly on the Internet, your computer must have TCP/IP software.
Tensile strength	The maximum stress in uniaxial tension testing which a material will withstand prior to fracture. The ultimate tensile strength is calculated from the maximum load applied during the test divided by the original cross-sectional area.

Thermocouple	A pair of dissimilar conductors so joined at two points that an electromotive force is developed by the thermoelectric effect when the junctions are at different temperatures.
Thermo-mechanically affected zone (TMAZ)	The heat generated by the welding tool during the FSW process has plastically deformed the material in this region, influencing the materials properties.
Thin client	A "thin processing" client in a client/server environment that performs very little data processing. The client processes only keyboard input and screen output, and all application processing is done in the server. Examples are X Window terminals and Windows terminals.
Thread	A thread is a single, schedulable flow of execution.
Thread priority	A number assigned to a task or thread in a multitasking system that sets the scheduling priority for that task. This determines in what order processes or threads are run, within a multitasking environment.

Tool eccentricity	Deviation from parallelism between the actual and required center axes along the length of the tool. This results in a slight oscillatory movement of the tool pin.
Tool plunge	The rotating tool is forced vertically into the material to be welded along the joint interface. This process should be controlled to ensure tool or machine limits are not exceeded. After the correct plunge depth into the material is reached, plunging ceases and the dwell period begins.
LMI TS1000 interface unit	The interface unit developed by Libra Measurement Instruments, which provides an interface between the computer and instrumented chuck. This interface handles the calibration of sensors, signal conditioning of the sensor data received and provides a serial interface for sampling the data by a computer system.
Torque control	A method of using current limit circuitry to regulate torque instead of speed of an actuator (motor, shaft etc).
Unified Modeling Language (UML)	A standard for specifying the properties of, and interactions between, the elements of a software system. UML is

	essentially a graphical technique, although there is an underlying formal semantics.
Universe of discourse	The universe of discourse is the range of all possible values for an input to a fuzzy system.
User interface (UI)	A basic operating system function that manages the accomplishment of the computing tasks of users by a computer system.
Vector control	A control method where the position and speed information of the rotor are used in the control circuitry to achieve similar performances as that of a DC servo drives.
Voids	FSW welds can contain areas within the weld nugget where improper consolidation of the plasticized material occurred. This leaves cavities in or along the surface of the weld, greatly reducing the mechanical properties of the weld.
Web browser	A software package that provides the user interface for accessing Internet, intranet, and extranet websites. Browsers are becoming multifunction universal clients for sending and receiving e-mail, downloading files, accessing

	Java applets, participating in discussion groups, developing web pages, and other Internet, intranet, and extranet applications.
Web server	A web server is a computer program running on a computer that is continually connected to the Internet. Specifically it 'hosts' websites, which is to say that it has them stored on its disks. A single web server can host many sites each with their own address. Anyone with a web browser will be able to use this address to connect to the web server and see the websites it hosts.
Weld hardening	During welding the material is heated followed by a cooling period. This is the same as tempering, resulting in the hardening of the material.
Weld interface	See joint interface.
Weld nugget	This is the recrystallised area in the TMAZ in aluminum alloys after they have been welded and has traditionally been called the nugget.
Weld sequence	The FSW welding process consists of a certain distinct

	sequence of steps. These steps form the weld sequence as follows: move to start of weld, start the tool rotation, plunge tool’s pin into material, wait for dwell period, traverse the weld path and extract the tool’s pin.
Wormhole	When an elongated sub-surface void is formed, due to incorrect process parameters, it is termed a wormhole.
YAG welding	This is a laser welding process, similar to plasma-keyhole welding. However, YAG welding uses a different type of lazer with a shorter wavelength.

List of Figures

List of Tables

CHAPTER 1

Introduction

Friction Stir Welding (FSW) is a relatively new process, where frictional forces applied by a rotating tool cause the material around the weld area to become viscous and mix. This process emerged during the last decade and research is needed to advance the technology to reach its full potential.

The Manufacturing Technology Research Center (MTRC) has acquired a conventional milling machine from a Pretoria based company and want to convert it into a friction stir welder for research purposes.

Aim

To design and implement an advanced monitoring and control system for the Friction Stir Welding process to perform research and development into intelligent manufacturing.

Objectives

Design and implement a computer based control system for the friction stir welding process by studying information pertaining to various modern control system techniques.

Apply sensors and amplifiers to measure forces, temperature and other parameters of the system to ensure sufficient feedback is available to facilitate intelligent machining.

Develop software using object-orientated techniques to provide a platform to perform investigation into intelligent manufacturing. The software should provide a human machine interface, a method to control the process variables (speed, feed, force, etc), sample and process sensor data and provide a means to test and implement intelligent system components.

Perform data analysis of the multi-variable FSW process to understand the fundamentals for process control purposes.

Identify a critical process control variable (temperature or force) and analyze its data linguistically by generating a simple fuzzy rule-based control system.

Hypothesis

The design and implementation of an industrial closed loop control and monitoring system with intelligent system components for the Friction Stir Welding process will serve as a research tool for further study of this process and intelligent manufacturing in greater detail.

Delimitation

A new machine will not be designed, rather the Correa milling machine will be modified to suit the process and provide a base framework for further development.

The system will not attempt to improve or modify the friction stir welding process, but rather only provide a platform to intelligently control and monitor the process to aid further research.

The data analysis for intelligence will include the generation and discussion of possible rules for intelligent control. Implementation and testing is for future research purposes.

The existing controls of the milling machine will be removed and an x86-based computer with the necessary interface cards will be used to control the system.

The software will be written to ensure maintainability, extensibility and flexibility.

The mechanical structure of the milling machine will not be altered in any significant way, rather the process parameters (weld material thickness, feed rate, etc) may need to be slightly adjusted to suit the machines limitations.

Only linear welds will be considered.

Significance of the Research

Within the Technikon

- Opening of a new research area, enabling further research projects for postgraduate students.

General

- Creating a foundation for the further exploration and research of the friction stir welding process and the growth of engineering expertise in this emerging technology within South Africa.

- The control system will also serve as a platform to demonstrate the advantage of intelligent monitoring and control of process parameters to aid research efforts.

Summary of Related Literature

Over the last 60 years the use of automatic control theory and technology has allowed many industrial processes to operate automatically under certain operating conditions. Most machining processes are stochastic, nonlinear, complex and ill defined and are open to control by means of intelligent systems. Process automation tasks are performed in hierarchical levels, including [Isermann, 1998]:

- **Process level**: Measurement of the input variables and manipulation of the output variables require a fast reaction time.
- **Control level**: Feedback and feedforward control where various variables are adjusted according to conditions or reference variables.
- **Supervision**: Indicate undesired or unpermitted process states and to take appropriate actions such as fail-safe, shut-down, or retriggering of redundancy or reconfiguration schemes.
- **Management**: Perform optimisation, coordination of general management in order to meet economic demands or scheduling and dedicated to tasks that do not require fast responses and act globally.

As the vagueness in automation generally increases with increasing hierarchical level the potential for fuzzy logic approaches grows. Fuzzy control in the basic level may be attractive for processes where only qualitative knowledge is available and strongly nonlinear properties exist. This is true of machining processes used in manufacturing, including milling, turning, drilling and grinding. Friction stir welding incorporates factors from all these processes, thereby lending it to intelligent monitoring and control.

The major independent variables in FSW are: tool material (coatings and condition), tool shape (surface finish and profile), workpiece material (type and temperature), feed parameters (speed, and vertical force), characteristics of machine tools (stiffness and damping), workholding (fixturing, etc) and dependent variables which are influenced by changes in the independent variables. The latter include: Force and energy dissipated in the welding process, temperature rise in the workpiece, tool wear, failure of the tool, quality (strength, residual stress) and surface finish produced on the workpiece after welding.

When unacceptable conditions result from welding operations, the manufacturing engineer must empirically determine the cause of the problem. If, for example, the quality of the workpiece being welded is poor and unacceptable, which of the independent variables should be changed first? The angle of the tool? If so, should it be increased or decreased? If the tool does not generate sufficient plasticity, should the weld speed, the vertical force or the tool profile be changed?

Intelligent machining is a procedure comprising various intelligent techniques (fuzzy logic, neural networks etc) and expert criteria (operator knowledge), with one of more higher resolution levels (hierarchical levels), which together manipulate the welding conditions resulting in a change of the frictional force between the tool and workpiece. Conditions include (tool rotation speed, vertical force, feed rate, etc). Various parameters relating to the tool and workpiece are monitored. This forms an intelligent closed loop system, allowing increased productivity through faster and higher quality welds.

Manufacturing has reached a stage where design, planning, simulation, machining, sensing, control, monitoring and assembly, as well as maintenance, management and customer service are integrated by means of computers and their network systems - hence Computer Integrated Manufacturing (CIM). More recently, sensor-based manufacturing coupled with artificial intelligence technology has resulted in on-line machining and intelligent monitoring. Intelligent systems for machining may include:

- Fuzzy decision support to select cutting parameters by considering material hardness, depth of cut, carbon content (data handbook) quality of machining, rigidity of machine-tool-workpiece and desired tolerances (from experience) [Balazinski, Bellerose, 1994].

- Hierarchical neural-fuzzy systems that combine the decision making capability of fuzzy logic with the learning capability of neural networks to implement adaptive advanced fuzzy control (from using a self-regulating output factor scaling system to changing the feed and speed for varying cutting depths to maintain a constant cutting force) schemes [Balzinski, Czogala, Sadowski, 1993; Trang, Yen, Nian, 1996; Yen, Trang, Nian, 1995].

- Self-learning algorithms (genetic algorithms) capable of modifying the rule base (neural-fuzzy system), based on the systems performance; for example, to use genetic algorithms to change the membership functions and control rule base to obtain an optimal control performance in machining [Hsu, Fann, 1996].

- Knowledge based systems to reliably and timeously diagnose machining process states, such as tool breakage, severe wear and chip hazard, and with the ability to detect and diagnose developing and sudden faults so that such faults are then prevented from re-occurring [Isermann, 1993; Fang, 1993; Andreasen, De Chiffre, 1993; Reltz, Elbestawi, 1993].

Numerical Control (NC) and Computerized Numerical Control (CNC) machine tools have been widely applied in industry. Productivity and production quality has been increased accordingly by means of these facilities. However, conventional CNC machines have the following limitations because of their closed architecture [Altintas, Newell, Ito, 1996; Alitintas, Munasinghe, 1996].

- They cannot efficiently provide real-time monitoring of a machining process by means of sensor feedback.

- The control of the machining process is not achieved adaptively in terms of on-line sensory data.

- The integration of task planning with control activities, and optimization of system performance, are not realized efficiently.

Open architecture is a competitive area of manufacturing and it will meet manufacturing requirements in supplying more competitive products for the global market. It has the following characteristics [Schofield, Wright, 1998; Altintas, 1994]:

- Sensor based. The combination of multiple sensors makes it possible to reflect the complex manufacturing process. Sensory data are not only for control, but also for process modeling, real-time simulation and performance monitoring and evaluation.

- Knowledge based. Human expertise, work experience, and testing experiment. Fuzzy logic is powerful in modeling human expertise and experience knowledge, as well as the highly non-linear manufacturing process. Since fuzzy knowledge inference is embedded within the modeling, monitoring and control, system flexibility and intelligence would be much enhanced.

- Integration. System integration is realized from different points of view. The processes of modeling, monitoring and control are integrated. On the other hand, sensory data and knowledge inference are integrated for on-line monitoring and remote decision-making via the Internet.

- Modular. A modular design is achieved in the interface and control of the system. It may be extended to other parts of the system, such as the inference algorithm.

- Openness. Systems developed incorporating those features mentioned above would be open to changes in respect of machine setup, machining process, and control algorithm and operation.

Organization of Thesis

Chapter 2 presents an introduction to Friction Stir Welding, describes the key components, fundamental concepts of the process operation and important variables that influence the weld.

Chapter 3 describes the hardware resources used to complete the experimental setup, providing a foundation for the development of the system’s software and intimately the creation of a test bed, used to investigate intelligent machining of the process.

Chapter 4 explains the software architecture, operating system architecture and key concepts employed when constructing the FSW software architecture.

Chapter 5 provides a detailed explanation of the implementation of each software module used to build the monitoring and control software.

Chapter 6 presents the aim, method and conclusion of welding tests run on the machine and the analysis of the results, including the calibration of the transducers.

Chapter 7 discusses the use of fuzzy logic controllers for intelligent process monitoring and control.

Chapter 8 concludes by providing some conclusions regarding the process and provides some suggestions for further research.

CHAPTER 2

Fundamental Concepts of the Friction Stir Welding Process

Friction Stir Welding (FSW) is a solid-state process containing multiple inputs and outputs, where a non-consumable shouldered pin tool is rotated in a weld material [Aota et al., 2001; Johnson, 2001]. The pin length of the tool is slightly less than the thickness of the material for butt joining. The tool is plunged into the material until sufficient contact is made between the tool face and material [Shinoda et al., 2001, Kallee, 2001]. This process on joining can be seen in Figure 2.1.

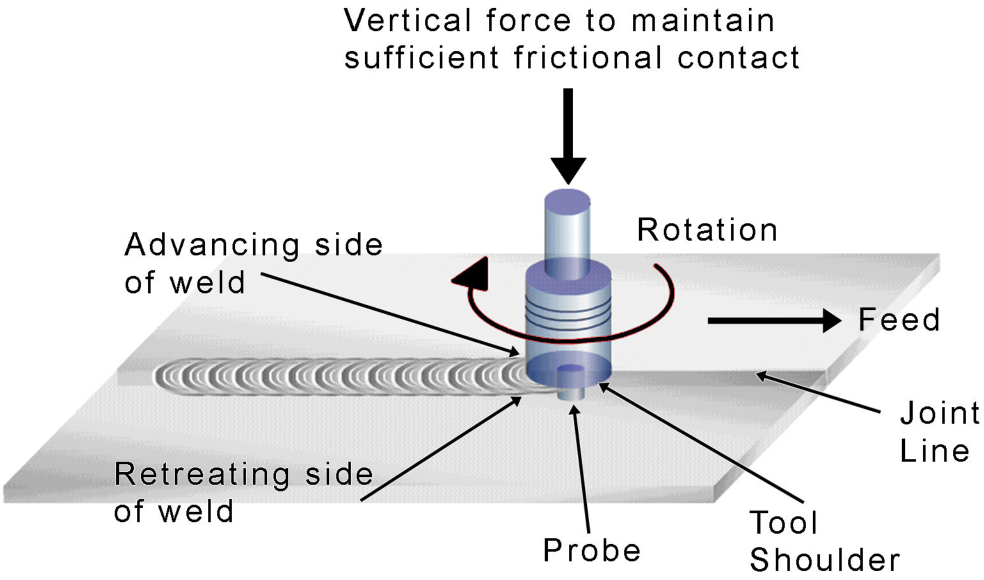
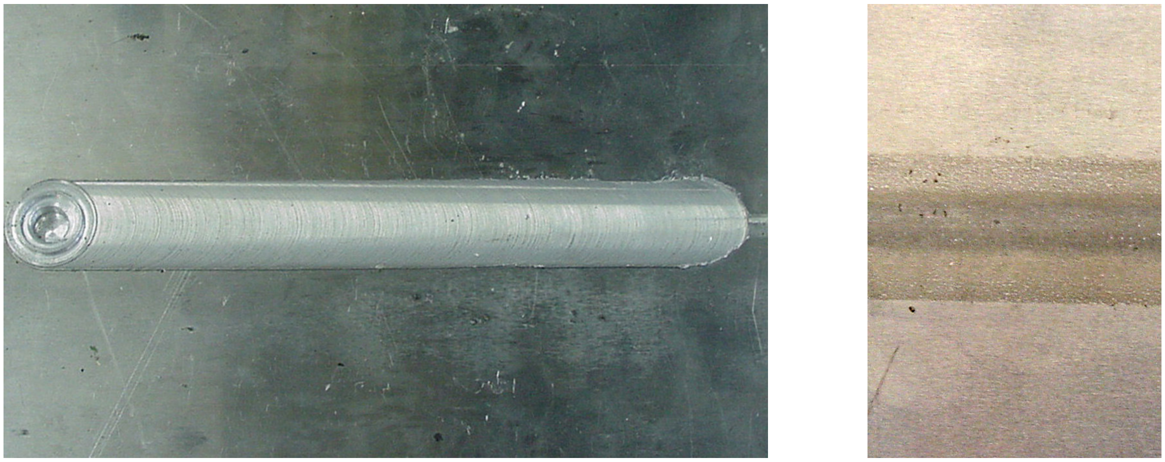


Figure 2.1: Diagram of the Friction Stir Welding process [Kallee et al., 2002]

Tool rotation causes friction, heating the material adjacent to the pin as it moves through the material. The material enters its plastic region and is swept from the front to the back of the tool, where it consolidates. The two materials to be joined are clamped, while in contact, on a backing plate before welding commences [Shinoda et al., 2001]. During the welding process, as the tool moves along the joining path, joint interfaces are destroyed and replaced by plasticised material, as shown in Figure 2.2. This is similar to a constrained internal extrusion [Johnson, 2001].

Welds are made below the melting point, in the solid phase of the material. Due to this low heat input excellent mechanical properties and low distortion are exhibited [Kallee, 2001].



(i) Top view (ii) Bottom view

Figure 2.2: Aluminum plate welded at the Port Elizabeth Technikon

There are three distinct stages in a typical FSW process cycle, which are:

- **Tool plunge** – The process of forcing the tool's pin into the material that is to be welded.
- **Dwell period** – After the plunge, a certain time period elapses where the tool rotates in contact with the plate, but with no traverse. This generates the initial heat to plasticise the material before the traverse is started.
- **Welding** – The traverse is then started and the tool moves along the joint line, welding the materials together.

The FSW process is a multiple-input variable multiple-output variable process. The following subsection will attempt to describe the effect and interaction between these variables. Information used for this section was obtained from available literature and resources. This information formed the knowledge on which decisions were based regarding the research conducted at the Port

Elizabeth Technikon. There are very few fundamental equations available to describe the FSW process.

2.1 Tool Design

When welding thin sheets of material, the main source of heat is from the shoulder of the tool. As the material thickness increases, more heat must be supplied by the friction between the rotating pin and the material. In addition, the main function of the pin is to ensure sufficient working of the material at the weld joint and control the flow of the material around the tool, in order to form a quality weld [Thomas et al., 2001].

Various shoulder profiles of the tool are used to improve coupling between the tool shoulder and the work piece by trapping plasticised material within special re-entrant features. This provides like-to-like frictional contact and improves the closure of the weld, by preventing plasticised material expulsion [Thomas et al., 2001]. Figure 2.3 shows some examples of shoulder geometries used by TWI.

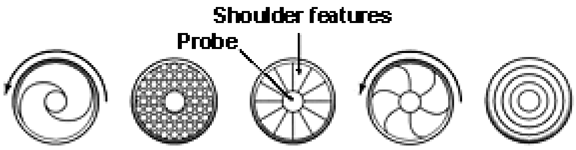


Figure 2.3: Some shoulder geometries used by TWI [Thomas et al., 2001]

The tool should be carefully designed to minimize horizontal traverse forces. This will reduce the amount energy necessary, allowing faster traverse rates to be used, resulting in lower thermal cycles [Johnson, 2001]. This is very important when welding hardened material, as heat input will cause softening and therefore loss of material strength [Johnson, 2001].

The diameter of the tool's shoulder is proportional to the torque at a constant rotational speed. As the tool shoulder diameter increases, so does the torque during welding. Different pin diameters have virtually no effect on torque values [Johnson, 2001]. Increasing the diameter of the shoulder, has practical limitations and tends to produce side flash on the weld surface, as can be seen in Figure 2.4 [Blignault, 2002].

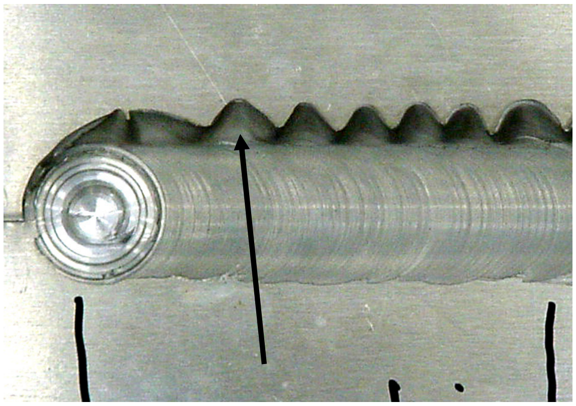


Figure 2.4: Side flash obtained during a trial weld at the Port Elizabeth Technikon

Experiments also show that as shoulder diameter increases, so does the M-type shape of residual stress distribution in the transverse section through the weld length [Donne et al., 2001]. This M-type distribution can be seen in Figure 2.5. Residual stresses will be discussed further in weld quality and weld rates Section 2.3.

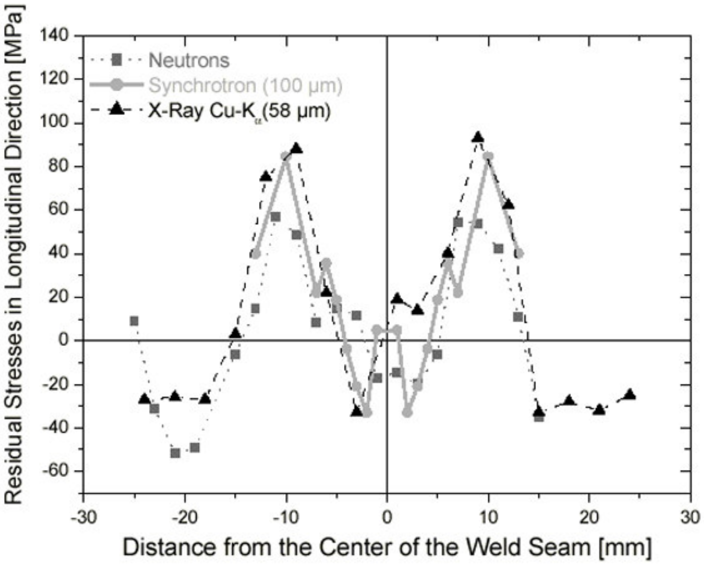


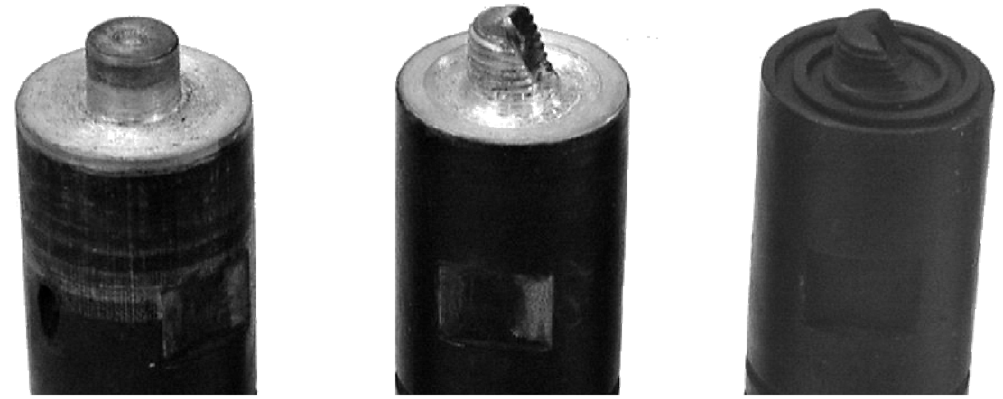
Figure 2.5: Example graph showing M-type distribution of residual stresses, through a 6013-T4 aluminum FSW weld [Donne et al., 2001]

Analyzing horizontal forces applied on the tool during welding is important for evaluating and improving tool designs. By attempting to minimize the increase in horizontal force with traverse speed, the process efficiency can be increased. Tool wear and the pickup of material on the tool contribute to a deviation of efficiency from its optimum value [Johnson, 2001].

Heat input to the process is proportional to the tool area and peripheral velocity of the tool. The cube of tool diameter can be related to the heat input. To reduce the energy of the process, the minimum tool diameter for a particular material should be found, which also maintains weld quality [Johnson, 2001].

Dawes has shown that by using a more efficient tool design and higher processing speeds, the tensile strength of a FSW weld in 7075-T7531 aluminum alloy can be increased from about 60% to 90% of the parent material.

The tool pin usually has a profiled surface to facilitate a downward auguring effect, which can be described as the tool gripping the plasticised material and pulling it in a downward direction [Blignault, 2002]. Examples of these profiles can be seen in Figure 2.6. Tools Y1 and Y2 were used to conduct the experimental analysis discussed in Chapter 6.



Non-profiled tool (Y1)

Threaded pin and simple shoulder profile (Y2)

Threaded pin and spiral shoulder profile

Figure 2.6: Examples of various tool pin profiles used for the FSW process at the Port Elizabeth Technikon

If the pin of the tool is too short for the actual material thickness to be welded, the work pieces are forged together and the oxide layers are not stirred up. This results in flaws, which are difficult to detect without destructive testing [Kallee, 2001].

2.1.1 Shape

Figures 2.6 shows some of the tools developed at the Port Elizabeth Technikon, used to obtain the test results later in this document. All the tools have a 20mm shoulder and a 5mm pin diameter. The following tools were designed and made:

- Non-profiled tool with a pin length of 5.5mm (Y1)
- Threaded pin with simple shoulder profile and a pin length of 5.5mm (Y2)
- Non-profiled tool with a pin length of 7.5mm
- Threaded pin with simple shoulder profile and a pin length of 7.5mm
- Threaded pin with spiral shoulder profile and a pin length of 5.5mm

2.1.2 Vertical Force

The Z-axis vertical force necessary for welding is much greater than the horizontal forces during the welding process. The vertical force is used by many FSW machine manufactures to control to process during welding [Johnson, 2001].

In the graph below, Fz, T and Fx represent the vertical force on the tool, tool torque and horizontal force on the tool respectively. Due to the rotation on the tool, Fx will be sinusoidal in nature [Johnson, 2001].

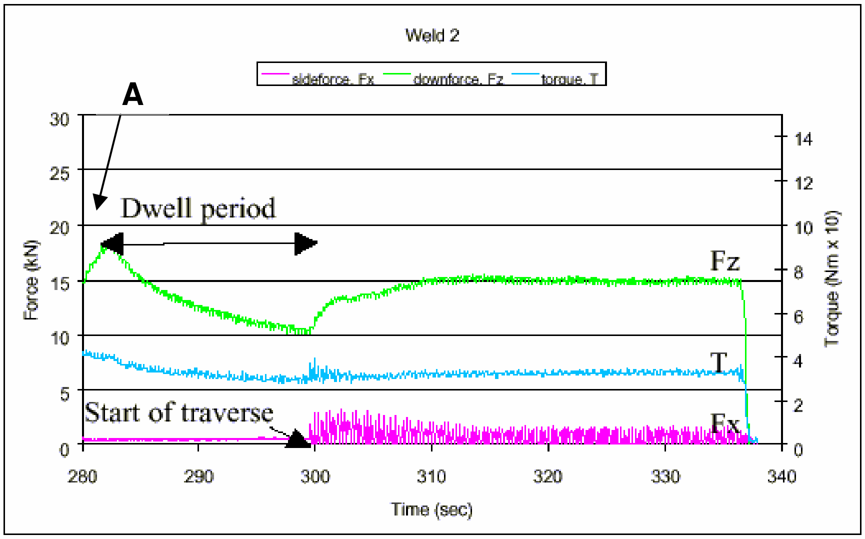

Weld 2
sideforce, Fx
downforce, Fz
torque, T
A
Dwell period
Fz
T
Start of traverse
Fx
Force (kN)
Torque (Nm x 10)
Time (sec)
30
25
20
15
10
5
0
14
12
10
8
6
4
2
0
280
290
300
310
320
330
340

Figure 2.8: Graph showing process variable values occurring in a full FSW cycle [Johnson, 2001]

Plunging too deep or fast can cause high vertical force values. However sufficient vertical force during welding is necessary to prevent running voids [Johnson, 2001]. This initial jump in vertical force can be seen in Figure 2.8 at point A.

2.1.3 Horizontal Force

The horizontal force exerted on the tool in the direction of traverse increases with traverse speed. It also increases less consistently and to a smaller extent with an increase in rotational speed, if the traverse speed is kept constant [Johnson, 2001].

A force at 90° to the welding direction exists due to the rotation of the FSW tool through the material. This force is about 5-10% lower than the force opposing the forward movement of the tool through the material [Johnson, 2001].

2.1.4 Torque

Torque is dependent on the rotational speed of the tool. Its value is almost constant at a particular rotational speed and is governed more by the size of the tool's shoulder as opposed to the pin diameter [Johnson, 2001].

2.1.5 Heat Input

Because FSW is a solid-state process, the heat input is low, about half that of MIG welding. The heat input will result in deformation and heat hardening, affecting the tensile strength and stress properties of the welded plate [Aota et al., 2001]. This is especially true of 6000 series aluminum, which gains its strength from a heat treatment. In the event of the heat input becoming too large, the strength of the heat-treated alloy is reduced in the heat-affected zone (HAZ) and distortion problems could occur [Kallee, 2001].

Heat input will decrease in proportion to the welding feed rate [Aota et al., 2001]. Due to the lower heat input the width of the nugget and HAZ in FSW will be narrower than that in MIG welding. With FSW and MIG welding the position of minimum hardness will occur in the HAZ, however the minimum hardness of FSW is higher than MIG welding. Also, in FSW the position of minimum hardness follows the advancing side of the tool [Aota et al., 2001].

Comparisons have been conducted between the deformation of FSW, MIG and YAG welded plates and it was concluded that FSW showed the least deformation [Shinoda, 2001].

2.1.6 Sensing

Typical methods used to sense process variables, include thermocouples and infrared imaging to measure temperature and dynamometers, mounted on the spindle, to monitor applied forces.

2.2 Friction Stir Welding Automation

Each new material and thickness, which is to be welded, needs different parameter settings and possibly tool design [Shinoda, 2001].

Distinct trends can be seen between the process data for horizontal and vertical force and temperature, which can be related to the process parameters (spindle speed, traverse speed and plunge depth). These trends appear to show a good level of reproducibility [Johnson, 2001].

Performing friction stir welds through solid plates (not along a joint line), produced forces and torques almost identical to welds performed along joint lines. This can be used to conserve resources, when planning experiments to measure force and torque levels [Johnson, 2001].

The drop in the vertical force during the dwell period after plunging can be calibrated to allow for an automatic start of the horizontal traverse. This delay, will allow sufficient, but not too much softening of the material. The dwell period could also be manipulated to prevent the initial horizontal force at the start of traverse from rising above its steady state value, which could lead to tool failure [Johnson, 2001].

Excessive penetration of the FSW tool into the material can be avoided by constantly monitoring the vertical force. In the event of the tool coming into contact with the backing plate the steady state vertical force values become unstable. Vertical force control can therefore be employed to maintain the perfect plunge depth [Johnson, 2001].

By monitoring temperature, overheating during welding may be detected. In the event of overheating, all the force traces will become unstable. This could be used as a measure of weld quality during production, when various rotation speeds are used and an incorrect setting may be selected for an application [Johnson, 2001]. There is also a minimum required contact time between the tool and the base material because the material must be in its plastic state, not in its liquid state [Shinoda, 2001].

The creation of running voids may be detectable by monitoring the relationship between the horizontal force in the traverse direction and the traverse speed. Running voids could cause data points to fall outside the expected trend [Johnson, 2001].

Reducing the forces generated during a FSW weld will increase the process efficiency. This will allow smaller machines to be used. The generation of heat and thermal cycling of the material being welded will also be reduced [Johnson, 2001].

The following subsection will discuss process parameters which lend themselves to process automation. The following parameters can be used with the characteristics discussed in Section 2.1.2 to 2.1.5 to implement automation of the FSW process.

2.2.1 Plunging the Tool

The Z-axis downward force should be minimized while plunging the tool into the material [Johnson, 2001].

A dwell period is necessary, after plunging is stopped, but before the traverse is started. This dwell period will allow the material around the FSW tool to fully plasticise. The duration of this period is currently usually determined by operator experience.

At the start of a dwell, the vertical force will initially drop rapidly, to a value slightly below its steady state value, after which a far more gradual decrease can be seen in Figure 2.8 [Johnson, 2001].

To avoid the formation of running voids during welding, it is essential that the tool be correctly plunged into the material.

2.2.2 Weld Rate

After the dwell period, the traverse will be started. Peak values of torque and X-axis force will be attained at or immediately after this event. The vertical force will climb, while the torque will decay from its dwell period value to its steady state value. The initial jump in the X-axis force will also decay slightly to its steady state value [Johnson, 2001].

Tests conducted on various types of 6mm aluminum plate, revealed that the magnitude of the tensile residual stresses reached in the HAZ decreases with decreasing weld speed, tool rotational speed and increasing dwell time. Increasing the dwell time caused a widening of the heat-affected zone due to increased heat input. Tests also have shown that weld strength increased with increasing weld speed in 6013-aluminum [Donne et al., 2001]. Tests conducted with 5083-aluminium show the fatigue strength also increases as tool travel speed increases [James et al.].

2.2.3 Tool Angle

The angle of the tool's pin will affect the two material flow patterns present in a weld, the bottom flow and the surface flow. The tool's angle will affect the intersection of these two flows, which results in a stagnant material flow [Shinoda, 2001]. This area exists at the bottom of the weld, when using a 0^{o} tool tilt and could result in weld defects [Shinoda, 2001]. Increasing this angle can make this stagnant point move upwards, even above to plate surface, possibly removing defects. During welding, the material in the weld zone rises and is extruded around the pin to the back of the tool [Shinoda, 2001].

2.3 Weld Quality

The macro-section of a weld interior typically exhibits a high degree of continuity and lacks defects such as porosity or cracks [Shinoda, 2001]. A good quality weld nugget should display an onion ring like structure, with no porosity or internal voids detectable [Kallee, 2001].

Defects can occur if incorrect processing parameters are selected for welding. The width of the weld joint will widen towards the surface of the material, due to extreme deformation caused by frictional heating between the tool shoulder and material surface [Johnson, 2001]. This characteristic can be seen in Figure 2.9 and 2.10.

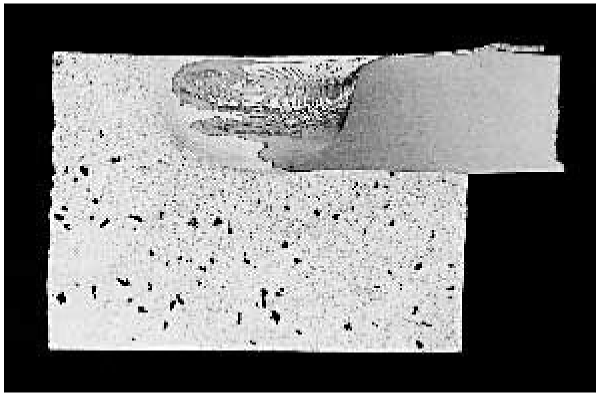


Figure 2.9: Transverse macro-section of 6mm thick wrought aluminum welded to cast aluminum [Kallee et al., 2002]

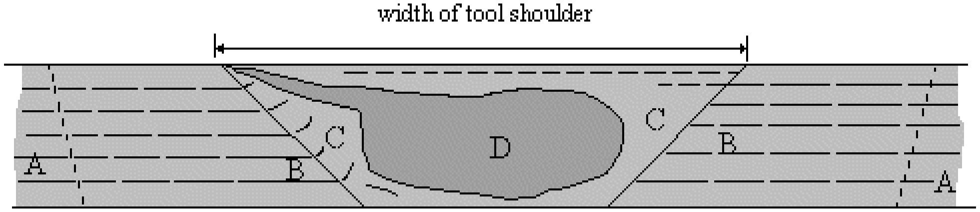


Figure 2.10: Diagram describing transverse section of weld [Kallee et al., 2002]

The various areas visible in Figure 2.10, will now be explained:

A. Unaffected material or parent metal: This is material, which has not been deformed as it is away from the weld. It may have experienced a thermal cycle from the weld, but is not affected by the heat in terms of microstructure or mechanical properties [Kallee et al., 2002].

B. Heat affected zone (HAZ): This area will lie closer to the weld center and has experienced a thermal cycle, which has modified the microstructure and/or the mechanical properties of the material. There is no plastic deformation occurring in this area [Kallee et al., 2002].

C. Thermo-mechanically affected zone (TMAZ): The material has been plastically deformed by the heat from the welding tool and will have had some influence on the material [Kallee et al., 2002].

D. Weld Nugget: This is the recrystallised area in the TMAZ in aluminum alloys and has traditionally been called the nugget [Kallee et al., 2002].

In macro-sections of good quality welds, a well-developed nugget can be seen at the center of the weld. In the thermo-mechanically affected zone, just outside the weld nugget, severe plastic deformation and some areas of partial grain refinement can be found. The process parameters and weld material, will determine the shape of this weld nugget [Kallee, 2001].

For a good quality weld, there should be no wormholes or surface defects. This is caused where the material does not consolidate properly behind the tool and a surface or sub-surface elongated cavity remains [Johnson, 2001]. Poor consolidation can be a result of incorrect FSW processing parameters or tool geometry [Johnson, 2001]. Photographs of common weld defects can be seen in Figures 2.11, 2.12 and 2.13.

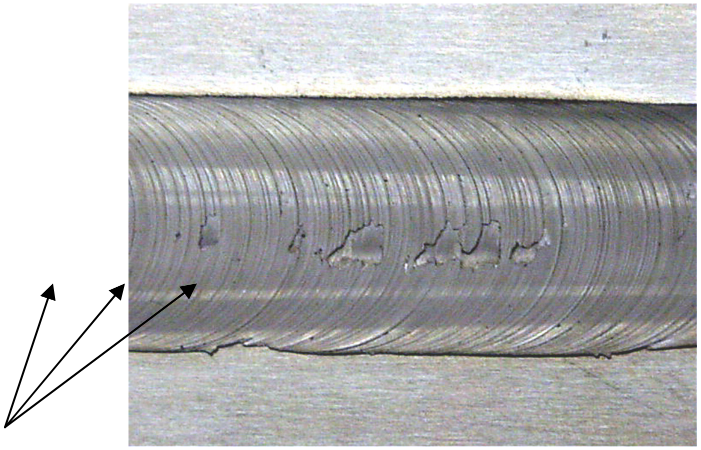


Figure 2.11: Small surface defects

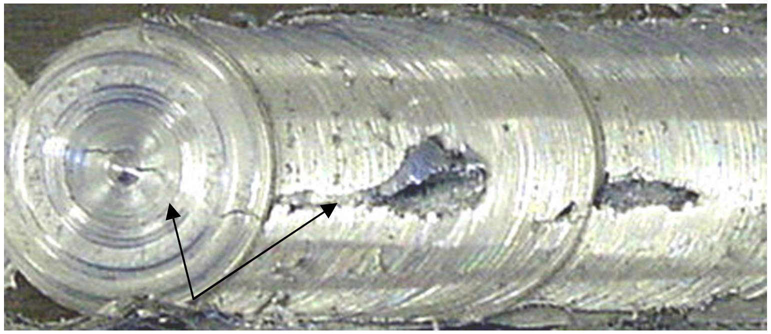


Figure 2.12: Large surface cavities

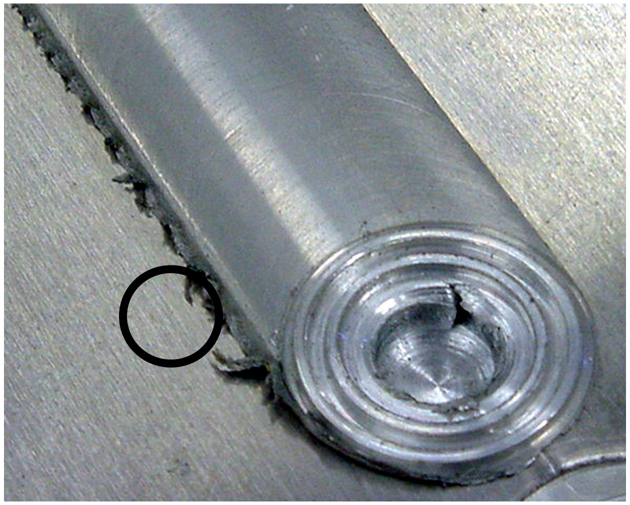


Figure 2.13: A sub-surface wormhole visible at the end of the weld run

When tensile strength of FSW joints are compared to that of MIG welded joints, it is found that all mechanical properties are superior to MIG. There is a much smaller softening zone width for FSW joints as opposed to MIG joints [Shinoda, 2001].

For FSW, the material hardness in the center of the weld is higher than that of MIG welds. Weld hardness of the stirred area was higher than that of the material near the weld interface [Shinoda, 2001].

The FSW weld exhibits a fatigue life similar to the base materials and greater than that of a MIG or Plasma-keyhole weld. The fatigue strength of FSW welds seem to be affected by the ratio of travel speed (W) to tool rotational speed (R), denoted as the W/R ratio. As this ratio increased it appears that the fatigue strength of the joint decreases. A critical value for this ratio has been found to be 0.2, above which results in severe fatigue endurance declination [Shinoda, 2001].

During welding, the vertical, longitudinal and lateral forces will attempt to force the clamped materials apart and lift them. A gap between the two materials of up to 10% of the sheet thickness can be tolerated before weld quality is affected [Kallee, 2001].

Maximum values of tensile stresses are located in the heat-affected zone of the weld. Tests also show that independent of weld speed, tool rotational speed and pin diameter; residual stresses are higher in the longitudinal direction as opposed to the transverse direction [Donne et al., 2001].

2.4 Friction Stir Welding Applications

FSW is suitable for the following industrial sectors:

- **Marine Industries** – Panels, extrusions, hulls, platforms, bulkheads, masts
- **Aerospace Industry** – Wings, fuselages, fuel tanks, rockets
- **Railway Industry** – high speed trains, rolling stock, tankers, containers
- **Automotive Industry** – engine and chassis cradles, wheel rims, vehicle bodies, armor plating vehicles, fuel tankers, caravans, mobile cranes, busses
- **Construction Industry** – aluminum bridges, window frames, façade panels, heat exchangers, pipe fabrication
- **Electrical Industry** – Electric motor housings, busbars, connectors, encapsulation of electronics
- **Other Industries** – Refrigeration panels, cooking equipment, gas tanks, furniture

A visit was conducted to the University of South Carolina, in order to form research links and also obtain additional exposure to the FSW process. The university has a hydraulic powered FSW machine, which is controlled via a computer. It has a dynamometer mounted on the spindle, used to measure vertical and horizontal forces, and tool torques. A thermal imaging camera is also available to measure temperature distribution during the welding process.

The university has conducted numerous aluminum welds. Thickness of butt-welded plate up to about 30mm was seen. A demonstration was also conducted of the FSW of mild steel. During the weld, the tool's temperature rose to such a degree that the tool glowed white. This weld was done in an argon shield to prevent combustion.

2.5 Summary

A literature survey has been completed with regard to process control of the FSW process. It can be seen that this process can benefit from using a multiple-input multiple-output controller to optimize the process and the weld quality.

Friction Stir Welding is a solid-state process, where a non-consumable shouldered pin tool is rotated along the joint line of the material to be welded. This results in heat being generated causing the material to enter its plastic state and being stirred into a solid joint.

The FSW tool is responsible for the heat input to the process and the stirring of the material. Tool design is therefore crucial. By using a good tool design, the mechanical requirements of the machine can be reduced, increasing the efficiency of the process.

The vertical force exerted by the tool on the material is far greater than the horizontal force and is commonly used to control the plunge depth of the tool. The horizontal, vertical and torque forces have constant characteristics, if plotted on a graph. Different weld parameters tend to affect the size of the forces, rather than their shape.

The heat input to the process is an important quantity, due to its influence on the mechanical properties of the material being welded.

Selection of weld parameters is critical to ensure a good weld. Each new material, material thickness or tool design requires a new set of process parameters. Distinct trends with good reproducibility exist between process parameters. It may be possible to relate these trends to the occurrence of weld defects.

The welding process consists of three main sections, plunging the tool, the dwell period and the weld traverse. The dwell period is essential to ensure the material has plasticised sufficiently before the traverse begins.

Two flows of plasticised material exist around the tool during the weld. The angle of the tool affects the intersection of these flows. The intersection point of the two flows forms a stagnant flow, which can cause weld defects.

There are distinct regions of the weld, which can be seen in its macro-section. These include the unaffected material, heat affected zone, thermo-mechanically affected zone and the weld nugget. The quality of the weld is a measure of its internal mechanical structure and its surface finish and is affected by the selected weld parameters.

FSW can be used in a variety of industrial sectors, such as the marine, aerospace, railway, automotive, construction and electrical industries.

The Manufacturing Technology Research Center (MTRC) at the Port Elizabeth Technikon successfully completed the first Friction Stir Weld in South Africa, by joining two 6mm aluminum plates together. This was done, using a conventional milling machine, modified to form a Friction Stir Welding machine, discussed in this document.

The following chapter will discuss the mechanical modifications, hardware design and design implementation undertaken to convert a conventional milling machine into a Friction Stir Welding machine. This will enable research and development into intelligent manufacturing and the FSW process.

CHAPTER 3

Experimental Setup: Machine Controls, Sensors and Hardware Integration

This chapter will discuss the hardware aspects investigated during the conversion of a conventional milling machine into a Friction Stir Welding machine. The machine will be used as a foundation for experimentation into the FSW process.

3.1 Description of Milling Machine

A conventional milling machine was purchased, which was to be converted into a FSW machine. Figure 3.1 shows the condition of the milling machine upon arrival.

The machine's characteristics were as follows [Nicolas Correa]:

- 3-axis bed movement, with only one axis moving at a time

- 5.5kW 3-phase squirrel cage induction motor used for the ram spindle
- 1.5kW 3-phase squirrel cage induction motor for bed movement
- 0.18kW creep feed motor used for very slow feed rates
- 1kW ram feed motor used to control the position of the ram
- 7.5kW primary spindle motor
- Trip dogs and stop-dogs were used to set the required milling geometry
- The machine made extensive use of contactors for control

Bed feed rates and spindle speeds had to be set using mechanical selectors. To change speeds the motor needed to be stopped and a new setting selected.

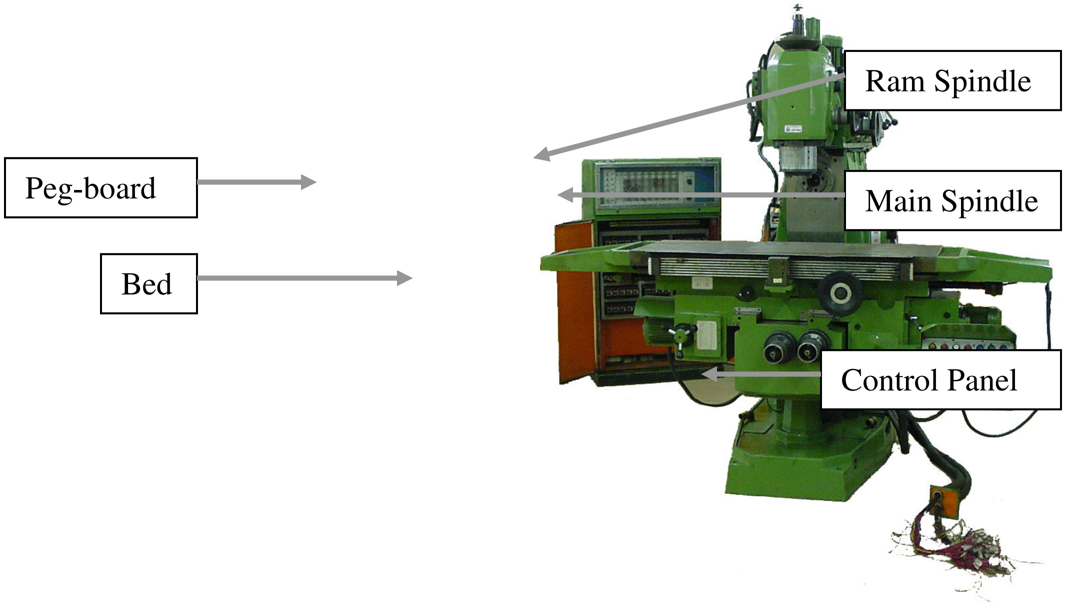


Figure 3.1: Initial condition of conventional milling machine purchased

The machine was not in a working condition prior to the start of this project. An attempt was made to repair the machine, but after an initial inspection, it was decided that this was not necessary and would only result in lost resources.

3.2 General Overview of Mechanical Operation

The milling machine consisted of two control panels. Figures 3.2a and 3.2b show the main control panel, housing numerous contactors and a peg-board. The contactors perform interlocking and machining sequence control. The peg-board is used to “program” the sequence of machining steps during the milling process. This panel is located next to the machine and is connected by two bundles of wires housed in flexible metallic shields [Nicolas Correa].

(a) Back (b) Front

Figure 3.2: Initial peg-board panel with contactors for machine control

Figure 3.3 shows another control panel, which is mounted on the back of the milling machine. It is much smaller than the previously mentioned panel and performs functions such as motor control and protection (starting, star-delta switching, motor temperature and over current protection) [Nicolas Correa].

Figure 3.3: Control panel mounted on back of machine before redesign

The required machine bed feed is engaged with certain combinations of 5 electromagnetic clutches. Three brakes are used to lock inactive axes. Two of the clutches are used to select normal feedrate or rapid feedrate, the other three are used to select the axis to be moved. Only one bed feed axis can be active a time and brakes must never be engaged for an axis that is to be intentionally moved [Nicolas Correa].

3.3 Implementation of System Architecture

This section will explain the requirements, design and implementation of the hardware necessary to facilitate the functionality required by a FSW machine, adhering to the physical constraints of the milling machines mechanical design.

The hardware must provide the following functionality to enable the requirements of an intelligent monitoring and control system for the FSW process to be met:

- Adjustment of the machines spindle speed, without having to stop or interrupt the machining process.

- The machine must have the ability to move the bed to a commanded position with a required accuracy. Positioning accuracy will be discussed further in the software implementation chapter (Chapter 5), it will be a function of feedrate, inverter communication baud rate and number of parameters monitored.
- The feed rate at which the bed moves to the required position must be able to be set before the process begins and then adjusted, without stopping the process.
- Spindle status must be available for further processing and recording. Status information must include:
  - Actual spindle speed
  - Motor temperature
  - Torque
  - Current
  - Power
  - Fault conditions
- Feed status must also be available for further processing and recording. The status information must include:
  - Actual bed motor shaft speed
  - Motor temperature
  - Torque

  - Current
  - Power
  - Number of motor shaft revolutions since the feed was started. This information will be used for positioning.
  - Fault conditions
- Process data from the FSW tool must be recorded and transferred to the computer for further processing. The following process variables need to be monitored on the tool:
  - Vertical force on the tool
  - Forces present in a 360-degree distribution around the tool in the horizontal plane, viz. force footprint
  - Tool temperature
  - Chuck electronics temperature (to prevent damage in the event of the heat from the tool reaching the electronics)
  - Tool torque
- The clutches and brakes, used to engage feed axes, gear ratios or hold positions, must be able to be individually controlled by the computer system. The software will handle selection and interlocking.

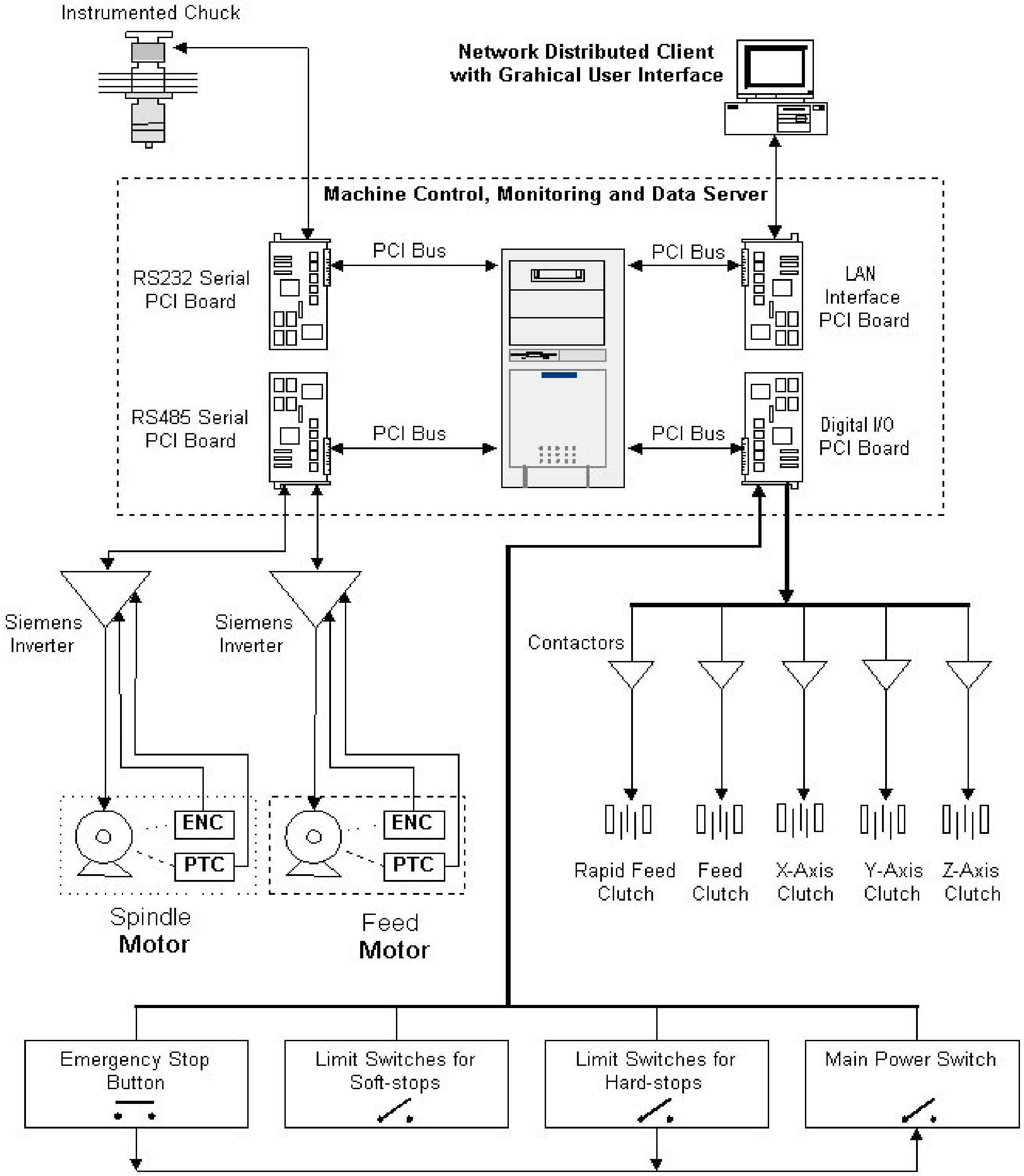
- The limit switches used to provide soft-stops, hard-stops and home positions must be interfaced to the computer, which will monitor their states and make control decisions.

The large amounts of process variables, which need to be monitored, are necessary to provide thorough investigation of the characteristics of the machine process.

The Figure 3.4 outlines the general hardware structure that was implemented. It provides a machine fulfilling all requirements of a system used for preliminary research into the FSW process.

The system is designed to facilitate adaptive control (AC) in a research environment. AC systems measure certain output process variables and use them to control speed and/or feed. They are designed to compensate for the changing environment by monitoring its performance and altering, accordingly some aspect of its control mechanism to achieve optimal or near-optimal performance. AC systems should be used in applications exhibiting the following conditions [Groover, 1987]:

- The in-process time consumes a significant portion to the total production time.
- Significant sources of variability exist in the process for which AC can compensate [Groover, 1987].



Instrumented Chuck
Network Distributed Client with Grahical User Interface
Machine Control, Monitoring and Data Server
RS232 Serial PCI Board
PCI Bus
PCI Bus
LAN Interface PCI Board
RS485 Serial PCI Board
PCI Bus
PCI Bus
Digital I/O PCI Board
Siemens Inverter
Siemens Inverter
Contactors
ENC
PTC
ENC
PTC
Spindle Motor
Feed Motor
Rapid Feed Clutch
Feed Clutch
X-Axis Clutch
Y-Axis Clutch
Z-Axis Clutch
Emergency Stop Button
Limit Switches for Soft-stops
Limit Switches for Hard-stops
Main Power Switch

Figure 3.4: FSW Machine Hardware Interconnection Diagram Overview

(Appendix A contains a detailed electrical wiring diagram of the machine.)*

FSW is therefore a prime candidate for AC. To facilitate AC the machines hardware must be capable of measuring output variables of the process. This was a key factor in the design and selection of the hardware for the FSW machine.

The central core of the design is the computer system with its software used to monitor and control the FSW process by communicating with the machines sensors and actuators. There are four primary PCI bus devices used to communicate with the machine. The digital IO, RS485 and RS232 are used to interface directly with the machine and the local area network (LAN) IO board is used to present and manage the distributed user interface using the TCP/IP protocol.

The design uses two motors each with an optical encoder (ENC) and thermistor (PTC). This is sufficient for the in-process control of linear welds. If profile welding is required an additional inverter could be installed for the ram feed motor. Each motor is controlled by an inverter, which is used to protect, monitor and control the motor’s operation to a user specified setpoint. The inverters use the RS485 communication specification to communicate with the computer system.

The existing bed limit switches are used to provide hard and soft stops for the bed movement. Soft-stops are used to prevent bed feeds outside an allowed area and as a marker when homing the machine. The hard-stops are connected directly to the feed motor inverter and used to generate a fault if triggered. Only in the event of a severe hardware or software malfunction will the hard-stops be necessary. An emergency stop button has been added for safety. When it is pressed, power to the motor inverters is immediately removed, thereby stopping the machine.

Each inverter has a dedicated RS485 connection to a 16650 UART in the computer, which allows higher data transfer rates as opposed to both inverters sharing the same communication medium. An advantage of using RS485 for the system as opposed to analog signals is that it allows many motor/inverter parameters to be written and read directly from the inverter by the computer. No problems due to analog noise or large bundles of wires exist. A disadvantage is that due to communication baud rates, write and read delays could become quite significant, affecting the response time of the control system.

An instrumented chuck has been built to allow detailed information to be obtained about process variables, allowing greater investigation into the process. The electronics mounted on the chuck is interfaced using the RS232 standard to the computer system.

Digital outputs are used to activate or deactivate electromechanical clutches and brakes on the machine, used to select or hold required feed axes.

The machine may be controlled from any client attached to the same network as the machine, unless access is limited by using the security features of 3rd party web servers, firewalls, routers or switches. Only one client may be connected at a time, preventing contention for control of the machine. When a client connects to the machine's server using a web browser, the server sends the GUI to the client. Requests for information and control messages are then continually sent between the client and server, until the client breaks the connection.

3.4 Selection and Explanation of Controls

The following subsections will discuss the hardware selected and installed to enable the conventional milling machine to become a FSW machine able to display characteristics of agile manufacturing. Figure 3.5 shows the hardware components used inside the computer to communicate with the machine and user interface.

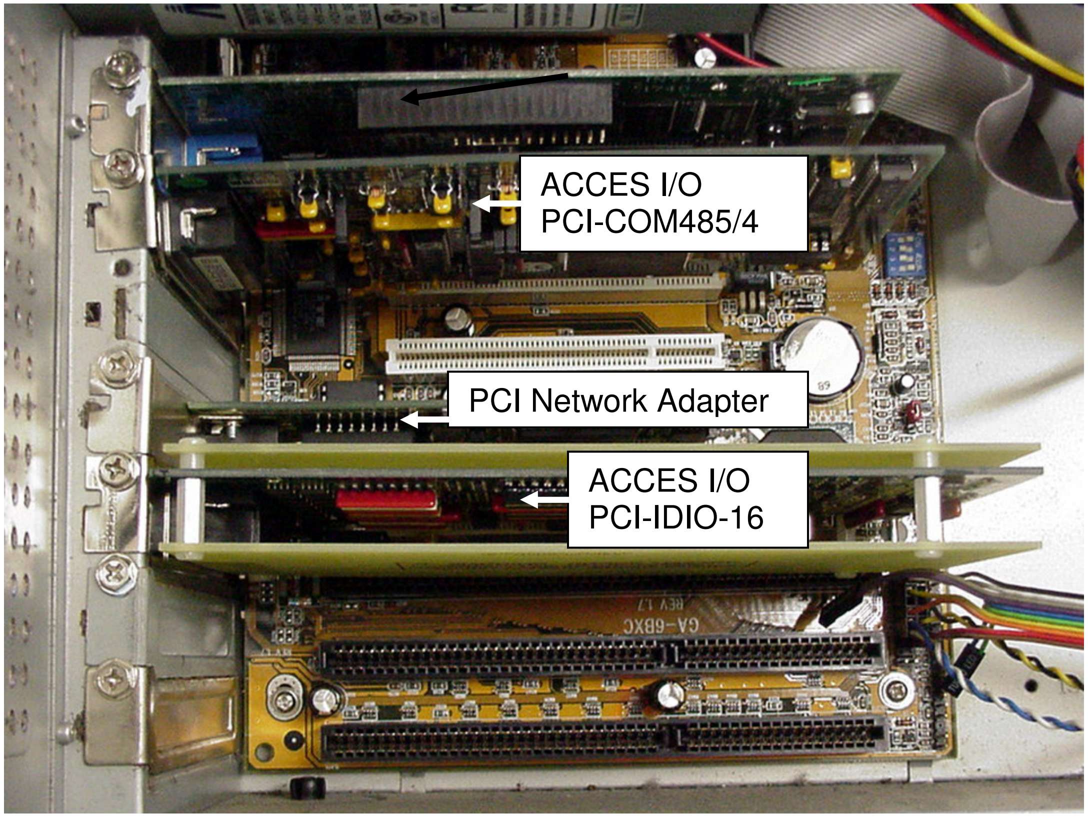


Figure 3.5: Computer interface boards housed inside server

3.4.1 PCI Digital IO Interface Card

A digital IO card was necessary for control of the clutches and brakes via contactors and the monitoring of limit and user interface switches. The contactors were necessary to overcome problems such as high inrush currents and high demagnetization EMFs, which could damage the interface card. The switches located on the machine interfaced to the computer are the main power isolator, motor inverter power switch, emergency stop button and x-axis, y-axis, z-axis soft stop limit switches. There is also a jog button for the spindle motor and hard-stop limit switches for bed positioning safety. These are connected directly to the motor inverter, which handles their operation.

The PCI-IDIO-16 is a half size board that provides an isolated digital input and output interface for PCI-bus computers. The card has sixteen optically isolated digital inputs for AC or DC control signals and sixteen solid-state switch outputs. An interrupt can be generated when any of the inputs change state eliminating the need to poll the board using software [Access I/O, EDIO-16 PCI Board User Manual, 2002].

3.4.2 PCI RS485 4 Port Interface Card

The RS485 PCI card selected allows the full range of baud rates to be used and includes a 16550 USART chip, making it compatible with standard PC comport software drivers. It also features auto-RTS transceiver control, which allows automatic switching of the line driver between transmit and receive modes. Only two wires are used, therefore only half-duplex communication is possible [Access I/O, RS485-4 PCI Board User Manual, 2002].

The Siemens motor inverters support the RS-485 communication standard, which is commonly used for factory environments. Other communication possibilities existed (Profibus, RS232, etc), however additional costs were either prohibitive due to additional modules having to be purchased, reduced the monitoring flexibility or required increased software development times for implementation.

The four RS485 ports allow the flexibility of adding additional motor controllers in the future if the system needs to be expanded (e.g. to add more degrees of freedom). Each inverter has a dedicated RS485 cable and UART linked to the computer. This enables communication to take place almost simultaneously between inverters and the computer. This allows higher data rates as opposed to using a single cable.

Both RS485 cables between the computer and machine are shielded and grounded to reduce the effects of electromagnetic interference (EMI) on the transferred data. Without proper shielding, errors occurred in the communications. This reduced the amount of data able to be written/read in a certain time period. This will reduce the systems response time [Access I/O, RS485-4 PCI Board User Manual, 2002; Perrin, 1999]. The RS485 network is terminated on the PCI board by the use of a jumper [Access I/O, RS485-4 PCI Board User Manual, 2002].

Complete rewiring of the machine was necessary in order to allow all the required hardware to be interconnected and provide a singular interface point to the computer system, shown in Figure 3.6.

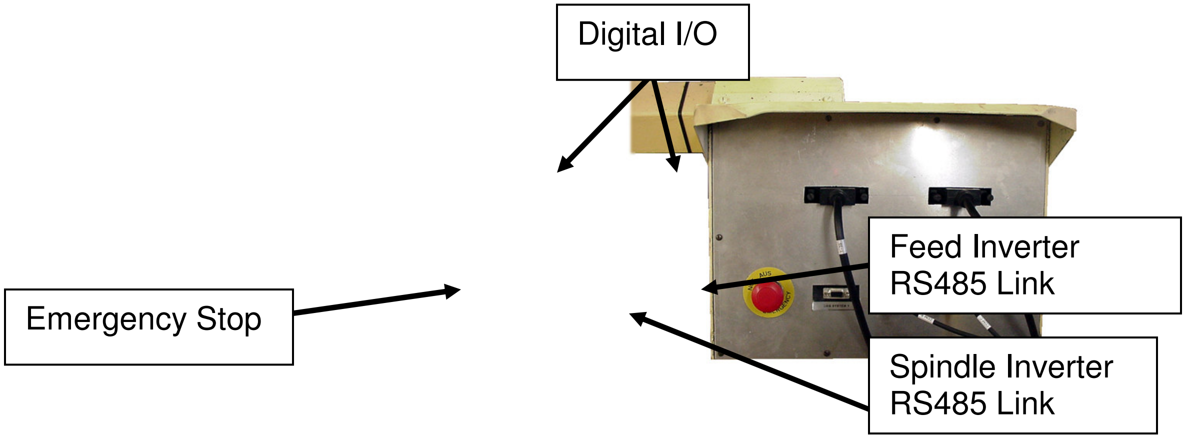


Figure 3.6: Interface point between FSW machine and computer

The original wire from the machine was reused, adhering to current versus conductor cross-sectional area limitations. Each wire was numbered and this was used to trace and wire the system correctly.

3.4.3 Motor Inverters

The two main pieces of hardware used to control the machine, besides the computer, are the spindle and feed inverters. Their correct selection was therefore crucial to ensure the required control and monitoring capabilities were supported.

Inverters from three different manufactures were evaluated during the hardware selection process. The Siemens Micromaster 440 Inverter was selected to control the spindle and feed motors on the machine, because of its flexibility, large amount of features and its capability-cost ratio.

The mounting positions of the Siemens MM440 Inverters in the control panel can be seen in Figure 3.7. Specifications in the user manual for the clearance of Siemens inverters, chokes and power supply, were adhered to. These specifications ensure that the equipment will receive sufficient ventilation to prevent overheating related problems [Siemens, MM440 Operating Instructions, 2002].

3.4.4 Optical Encoders

Leine & Linde 610/632 industrial incremental optical encoders were fitted to the rotors of the spindle and feed motors. The feed motor encoder provided feedback, which could be related to bed position and the spindle motor encoder allows more accurate motor speed control by the inverter, as opposed to other sensorless monitoring methods (sensorless vector control). A 512 pulse per revolution encoder with an unbreakable acrylic glass code disc was selected, due to its wide availability, low lead-time, reduced cost and suitability for the required application [Leine & Linde, 2002].

A standard low backlash coupling was used to link the encoder shaft to the motor's rotor [Leine & Linde, 2002]. The encoder is interfaced to the Siemens MM440 Inverter by an encoder module, which attaches to the face of the inverter. This module allows the use of single channel encoders and quadrature encoders with HTL or TTL logic [Siemens, Encoder Module Operating Instructions, 2002].

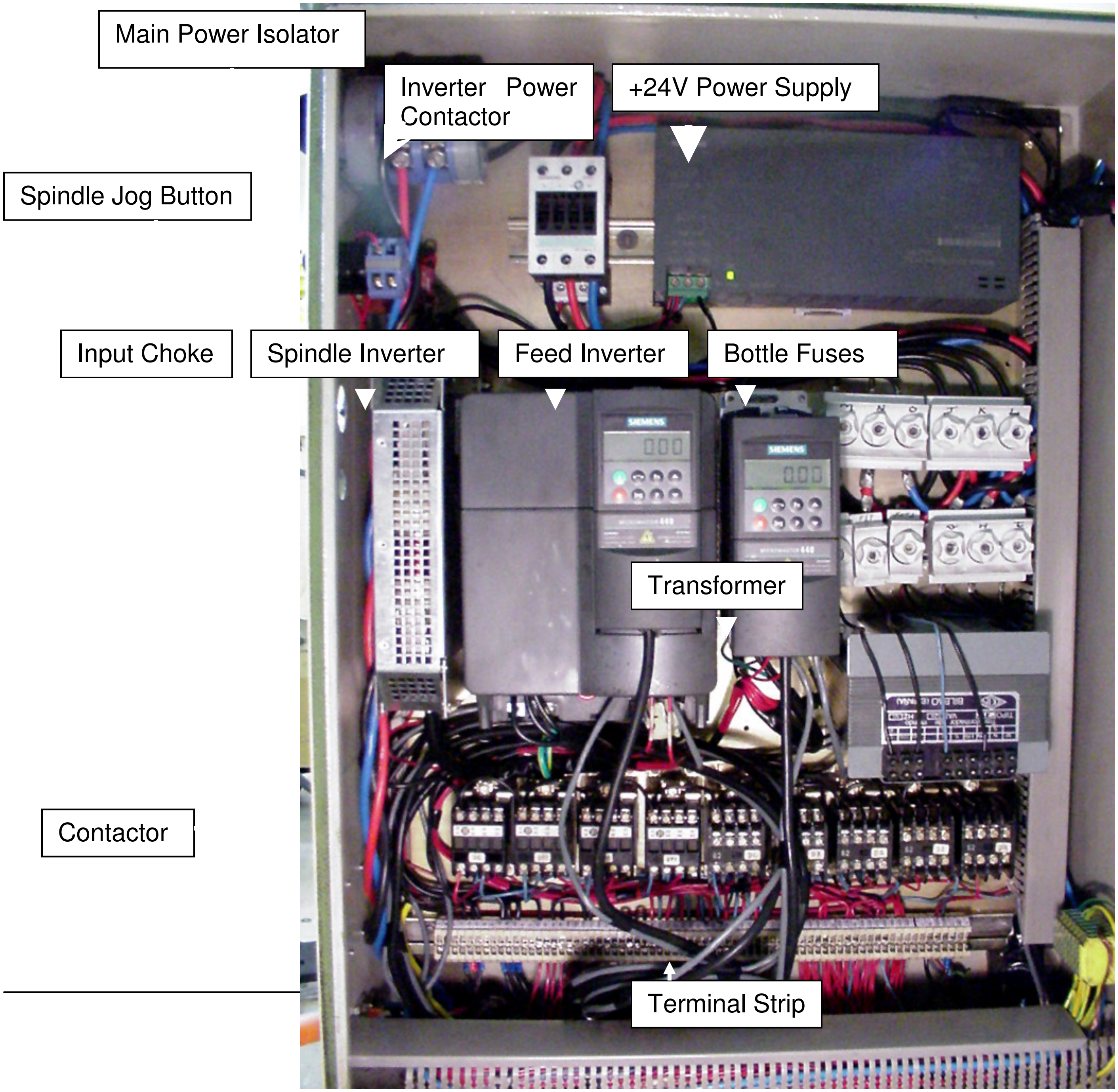

Main Power Isolator
Inverter Power Contactor
+24V Power Supply
Spindle Jog Button
Input Choke
Spindle Inverter
Feed Inverter
Bottle Fuses
Transformer
Contactor
Terminal Strip
SIEMENS
SIEMENS

Figure 3.7: Redesigned FSW Machine Control Panel

The only limitation is that the frequency of the encoder pulses generated must be below a certain threshold (f_{max}). Formula 3.1 can be used to calculate the frequency (*f*) of the pulses generated by the encoder [Siemens, Encoder Module Operation Instructions, 2002].

$$f_{\max} > f = \frac{Pulses\_pre\_revolution \times rpm}{60} \qquad (3.1)$$

At a maximum motor speed of 2175 RPM (maximum motor speed of 1450 RPM boosted by the inverters to 150%), f = 18.560 kHz using Formula 3.1. f_{max} for the Siemens encoder module is 300kHz, therefore the encoder will operate correctly with the encoder module.

3.4.5 Modifications to Induction Motors

The following three modifications, that can be seen in Figure 3.8, were carried out on the spindle and feed motors:

- When variable speed inverters are used care must be taken to avoid overheating of the motor at low speeds. This occurs due to the reduction of airflow through the motor's cooling vanes when the fan mounted on the rotor turns slower than it does at maximum rotor speed. If prolonged periods of slowed rotation will be experienced, forced cooling must be added to ensure sufficient airflow is maintained. Therefore 220VAC single-phase brushless cooling fans were added. The step-down transformer in the control panel reduces one 380VAC phase of the three-phase supply to 220VAC single phase for the fans.
- PTC temperature sensors were embedded into the stator windings of the motors that allow their temperatures to be monitored enabling thermal protection of the motors, due to insufficient cooling or overload conditions. This was necessary due to the unknown stress the FSW process would exert on the machine and its motors.
- Incremental optical encoders were attached to the rotors of the spindle and feed motors, which has already been discussed in Section 3.4.4.
- Cowls were added to the backs of each motor to provide for the following:
    - Guidance of the airflow through the motor's cooling vanes
    - Mechanical protection of the optical encoders
    - A support to mount the cooling fans

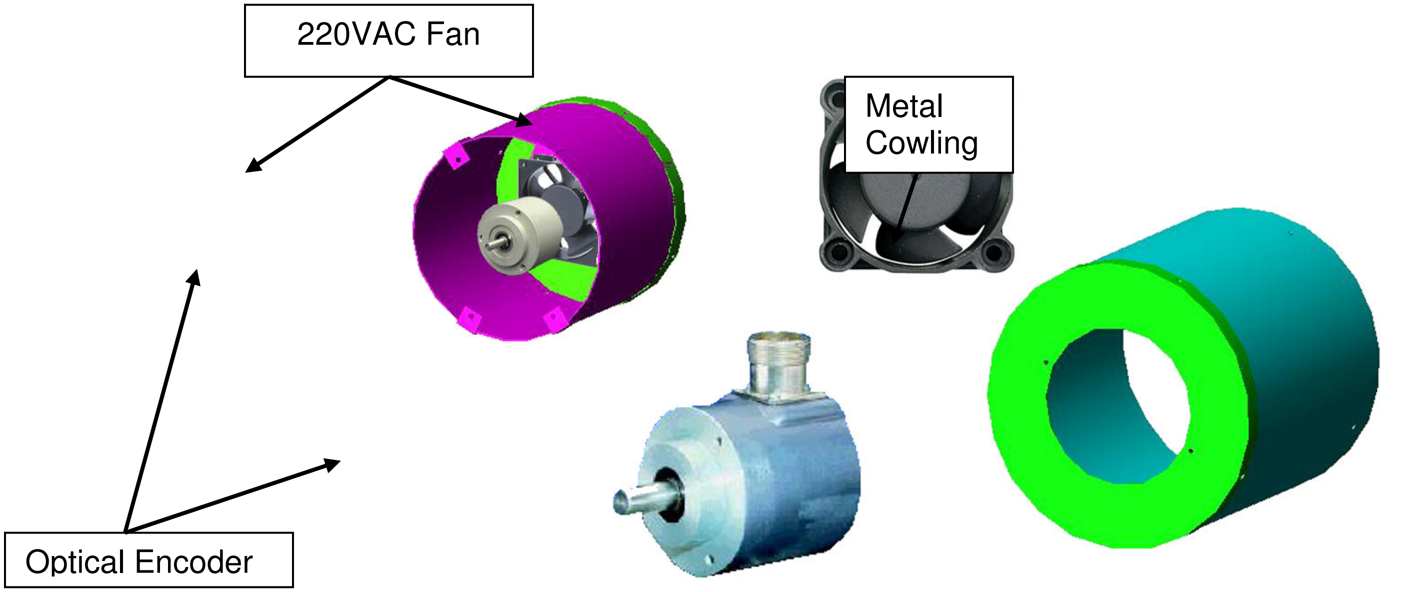
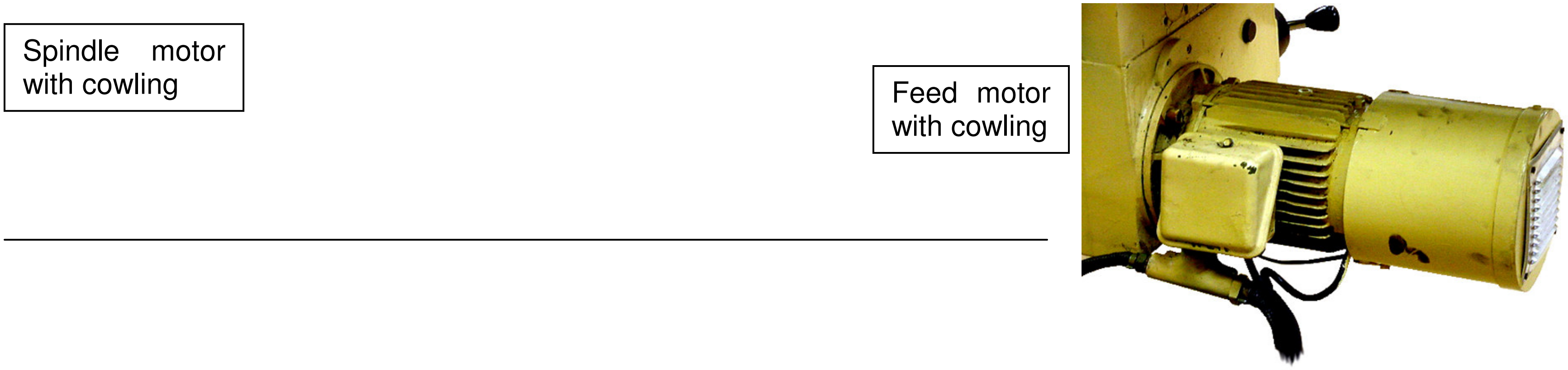


Figure 3.8: Contents of motor cowlings [Blignault, 2002; Leine & Linde, 2002]

Figure 3.9 shows the actual implementation of the spindle and feed motor cowlings mounted on the backs of their respective motors.



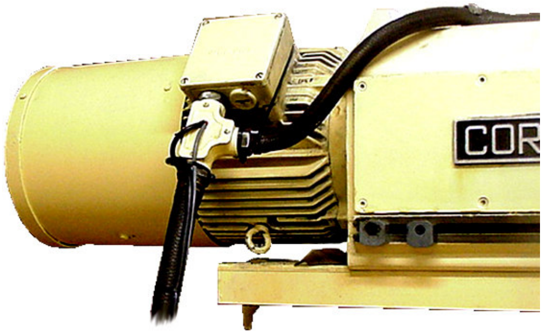


Figure 3.9: Modifications to spindle and feed induction motors

3.4.6 Contactors

Contactors were reused from the existing machine to switch its electromechanical clutches and brakes. Each clutch or brake draws approximately 1.5 amps steady state DC with a potential difference of 24 volts. The contactors are used as an additional layer of protection for the computer interface against large inrush currents and back EMFs generated on magnetization and demagnetization of the electromagnetic actuators. Also, the computer interface board can only switch a maximum current of 1 amp. The contactors require less current for switching, and therefore less electrical stress is exerted on expensive critical systems. In other words, it is much cheaper and easier to replace one contactor, than a computer interface board.

3.5 Telemetry Monitoring System

A method was necessary to obtain real-time in-process information regarding the welding process to promote further investigation into the process. This data can then be used to create a control system used to optimize the process. The following section will discuss the system implemented.

3.5.1 Reasons for Monitoring the Tool

It is necessary to monitor tool variables during the process for the following reasons:

- Direct measurement of process variables is preferred to indirect measurement methods [Hordeski, 1992].
- When the tool is busy traversing the weld path, continuous data is obtained as opposed to mounting discrete sensors on the material or bed. The latter method will only return a few discrete points of useful data along the weld path.

3.5.2 Requirements of the Monitoring System

The monitoring system must provide the following functionality to provide the information necessary to study the FSW process:

- Measurement of the horizontal forces (X force) on the tool
- Measurement of the vertical force (Z force) exerted on the tool
- Measurement of the torque exerted on the tool
- Measurement of the tool’s pin temperature
- A sufficiently high sample rate of horizontal force vectors to enable a 2D polar diagram to be constructed

- A sufficiently high sample rate of process variables allowing an acceptable response time for the AC of the process

3.5.3 System Operation

Figure 3.10 illustrates the basic operation of the tool monitoring system, which was built by Libra Measuring Instruments (LMI) [Libra Measurement Instruments, 2002] and installed on a custom designed chuck designed by a Technikon mechanical engineering masters student [Blignault, 2002].

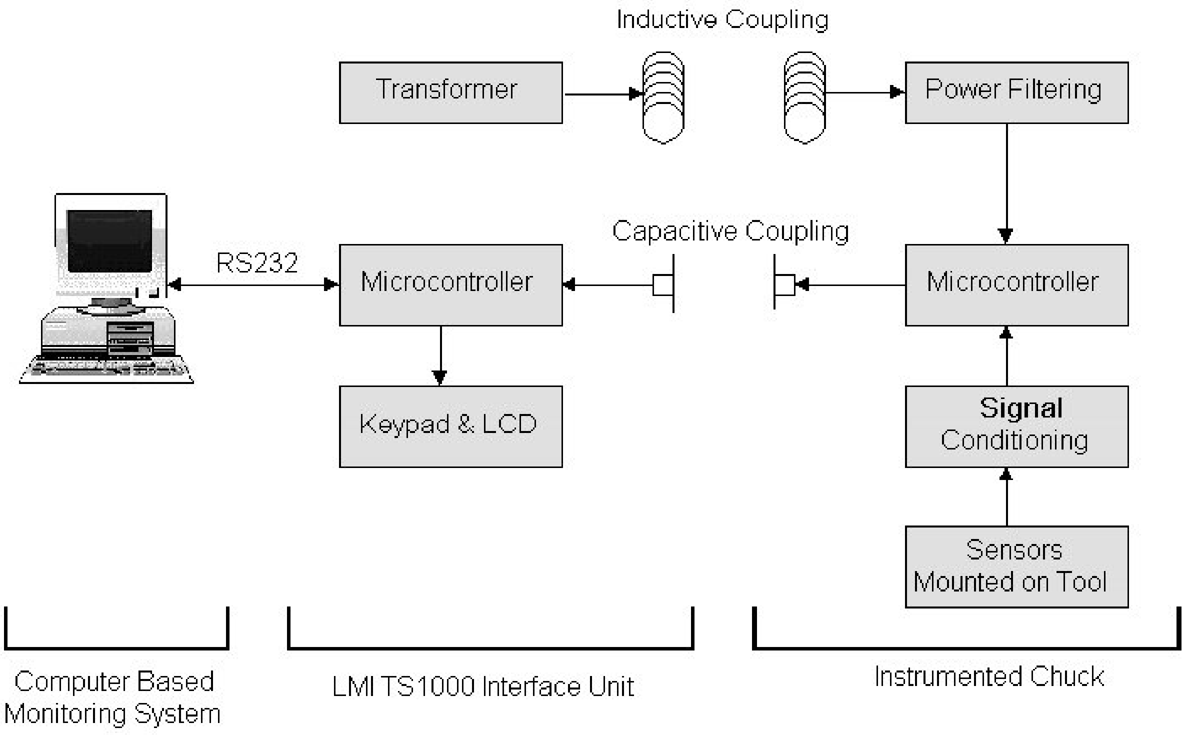


Figure 3.10: Block diagram illustrating monitoring system operation [Libra Measurement Instruments, 2002]

The electronics mounted on the chuck allow all the required variables to be sampled, the raw sensor data is signal conditioned and passed to a microprocessor, where it is prepared for transmission to the TS1000 interface unit. Electrical power is transferred to the chuck using induction and the sampled data is transferred off the chuck in digital form, using a capacitive technique. The power coil and capacitive receiver can be seen on Figure 3.10. There are two of the above systems present on the chuck, as each system can only support 4 channels of sampled data. Both systems share the same power coil and capacitive receiver. Table 3.1 lists the channel assignments used:

	Monitoring System 1	**Monitoring System 2**
Channel 1	Tool Pin Temperature	Tool Torque
Channel 2	Z Force	X Force
Channel 3	Tool Speed	Unused
Channel 4	Gauge Temperature	Unused

Table 3.1: Monitoring system channel distribution

The channels of sampled data are received by the microcontroller housed in the interface unit, for processing. When the interface unit receives a request for data, it transfers the information via the RS232 serial interface to the requesting device (the computer) [Libra Measurement Instruments, 2002; Husselmann, 2002].

Figure 3.11 shows the completed monitoring system mounted on the custom chuck attached to the machine's spindle. During the FSW process the tool temperature may reach very high temperatures, which may damage the electronics mounted on the chuck or cause them to produce erroneous data. A thermal insulator, a Tufnol disk (behind the power coil in the figure) and heatsink promote heat dissipation, preventing damage to the electronics. Another safety measure is the gauge temperature sensor embedded near the electronics, used to monitor the temperature rise. The inductive sensor is used to measure the tool's rotational velocity [Husselmann, 2002].

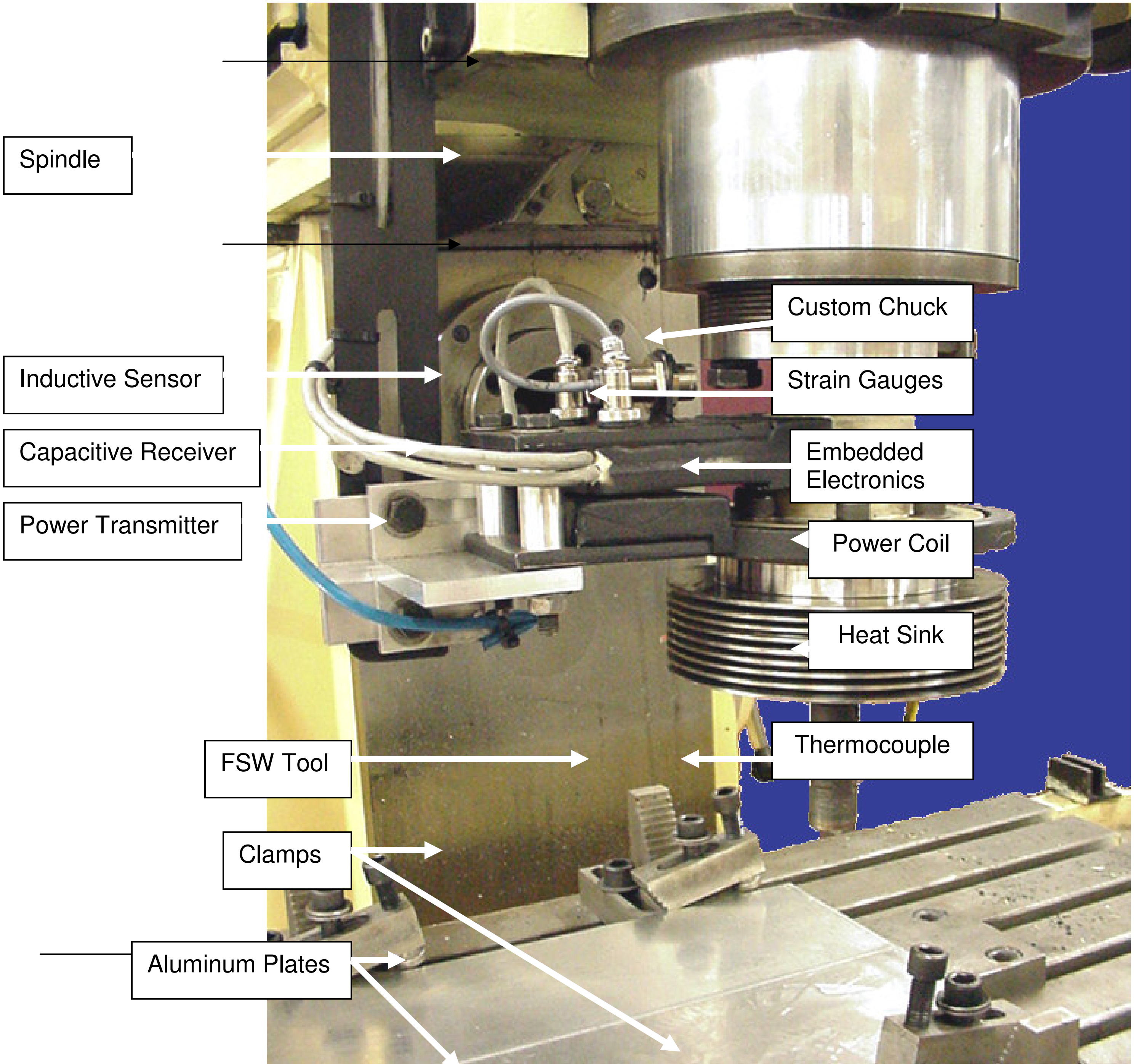

Spindle
Custom Chuck
Inductive Sensor
Strain Gauges
Capacitive Receiver
Embedded Electronics
Power Transmitter
Power Coil
Heat Sink
Thermocouple
FSW Tool
Clamps
Aluminum Plates

Figure 3.11: Tool monitoring system

The two TS1000 interface units are shown in Figure 3.12. Their user interfaces consist of an LCD display and keypad for entering calibration data. The LCD also displays the most recent measured values.

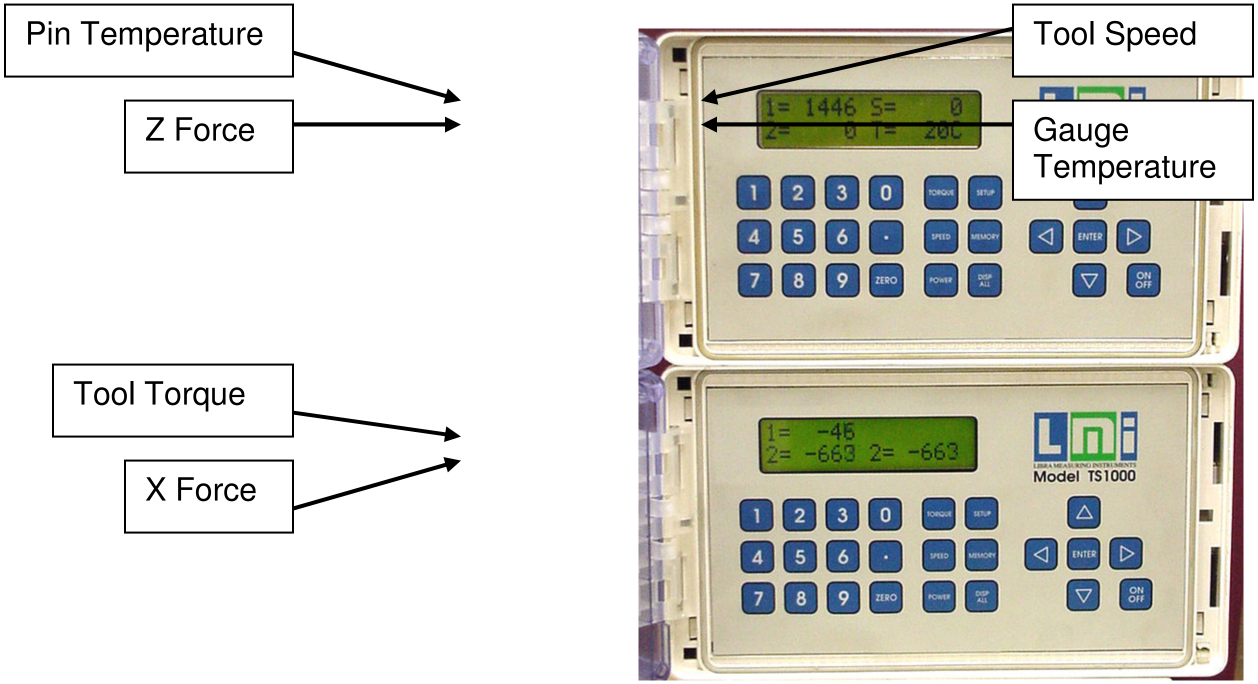


Figure 3.12: Tool monitoring system interface units

Each interface unit has a dedicated RS232 serial port for transferring the sampled data of the channels it is responsible for. The values displayed and transferred are the “real” values measured in kilograms, degrees centigrade and Newton-meters.

3.5.4 Installation of Monitoring System

A bracket had to be constructed to hold the power transmitter and capacitive receiver at the correct distance from the rotating chuck. This was necessary to ensure sufficient power was transferred to the chuck and process data could be received.

3.5.5 Monitoring System Limitations

The following are limitations of the monitoring system, resulting in sampled variables differing from the actual process variables read by the computer:

- **Machine Vibration** - Excess machine vibration during the weld trials caused instability of the power transmitter and capacitive receiver units. This resulted in voltage fluctuations producing transmission errors or loss in communication. In order to overcome this problem a more rigid mounting bracket for the system should be used as well as a more stable machine base. The machines base should also be secured to the ground.

- **Limited Sampling Rate** - The number of samples possible during one revolution of the tool is inversely proportional to the rotational speed of the tool. The maximum sample period allowed by the measuring system is 5ms. This is because each interface unit has two channels that must be alternately sampled at a baud rate of 28800bps.

$$R_{CH1} + R_{CH2} + D_{IU} + D_C \approx P_S$$

$$2.08ms + 2.08ms + 0.75ms + 0.1ms \approx 4.91ms$$

Where:

R_{CH1} and R_{CH2} are the transmission times to and from the interface unit at 28800 baud.

D_{IU} and D_C are the approximate processing delays imposed by the interface unit and computer software.

P_S is therefore approximate sampling period.

- Fluctuation of communication response times **- Response times vary when data is transferred between the measurement system and the computer, due to the design of the interface unit. This is unavoidable and results in deviations of expected sampling periods.**

3.5.6 Machine and Equipment Limitations

Following are limitations imposed by the machine on the data recorded from the telemetry system:

- **Excess vibration -** Machine bed vibration could cause measured tool forces to deviate from expected values.
- **Tool eccentricity –** Results in major deviations in force profiles if the tool is not correctly inserted in the tool holder.
- **Tool slippage -** Excessive forces on the tool can result in slippage of the tool in the chuck causing the 2D polar force plot to shift orientation.
- **Tool Orientation -** Although the profile and magnitude of the horizontal polar force plot stays constant during tool replacement, its angle of orientation varies. This is due to characteristics of the tool's geometry lying in different orientations to the strain gauges mounted on the chuck.

3.6 Installation of Network Distributed Client

Figure 3.13 shows the two computers used for the experimental setup. The client can be located anywhere on the Internet and used to control and monitor the FSW machine. Access can be limited by using standard information technology (IT) tools if required.

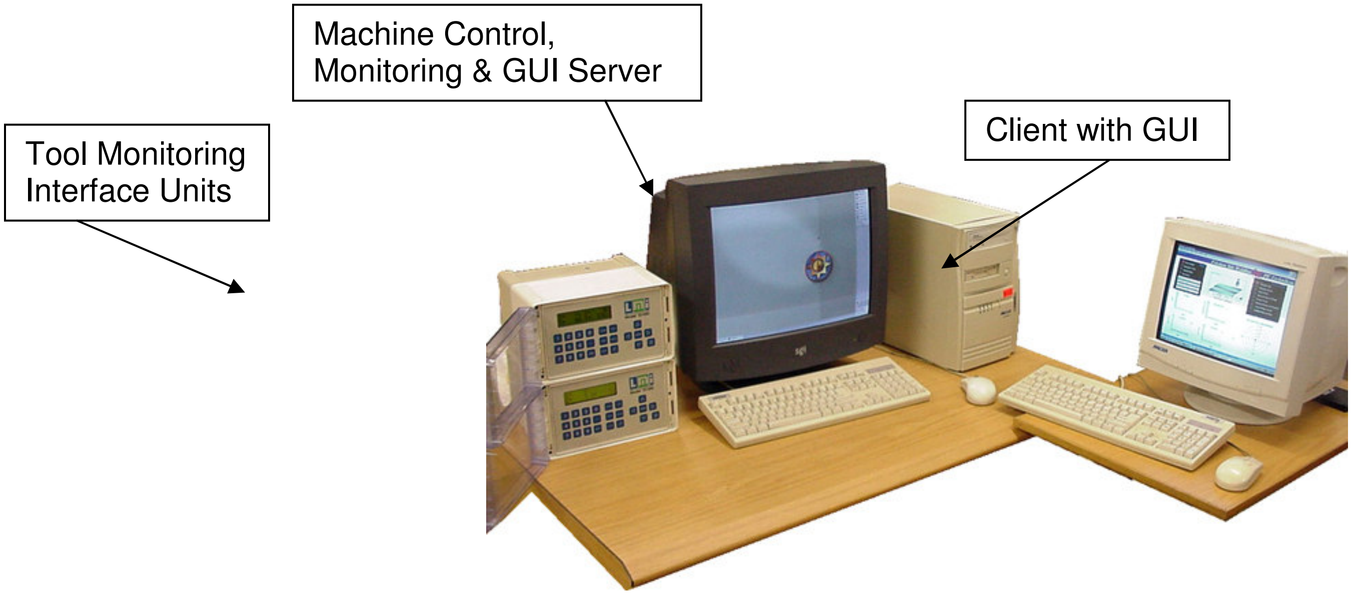


Figure 3.13: Machine control computer next to GUI client

An IEEE 802.3 100Base-T network was used as the communications backbone between the FSW machine server and client computer [Held, 1994]. The TCP/IP protocol is used, enabling the client to be located almost anywhere in the world. This is because this protocol is used for the Internet, which spans most of the planet. A constraint of this statement is that the client must be connected to the Internet, without restrictions to the required IP address or ports used to connect to the FSW machine server.

3.7 Summary

Figure 3.14 shows the completed experimental setup, consisting of the FSW machine, its control and monitoring computer and network-distributed client.

All the monitoring and control software was written, using this setup. In addition all the weld tests used to obtain experimental data were conducted on this system.

A conventional milling machine, using contactors and a pegboard for control, was upgraded and converted into a FSW machine, which can be used for research purposes.

The FSW machine has four degrees of freedom, the machine beds x, y and z-axes and the ram spindle. The entire control system for the milling machine was removed and replaced with inverters, used to control the motors.

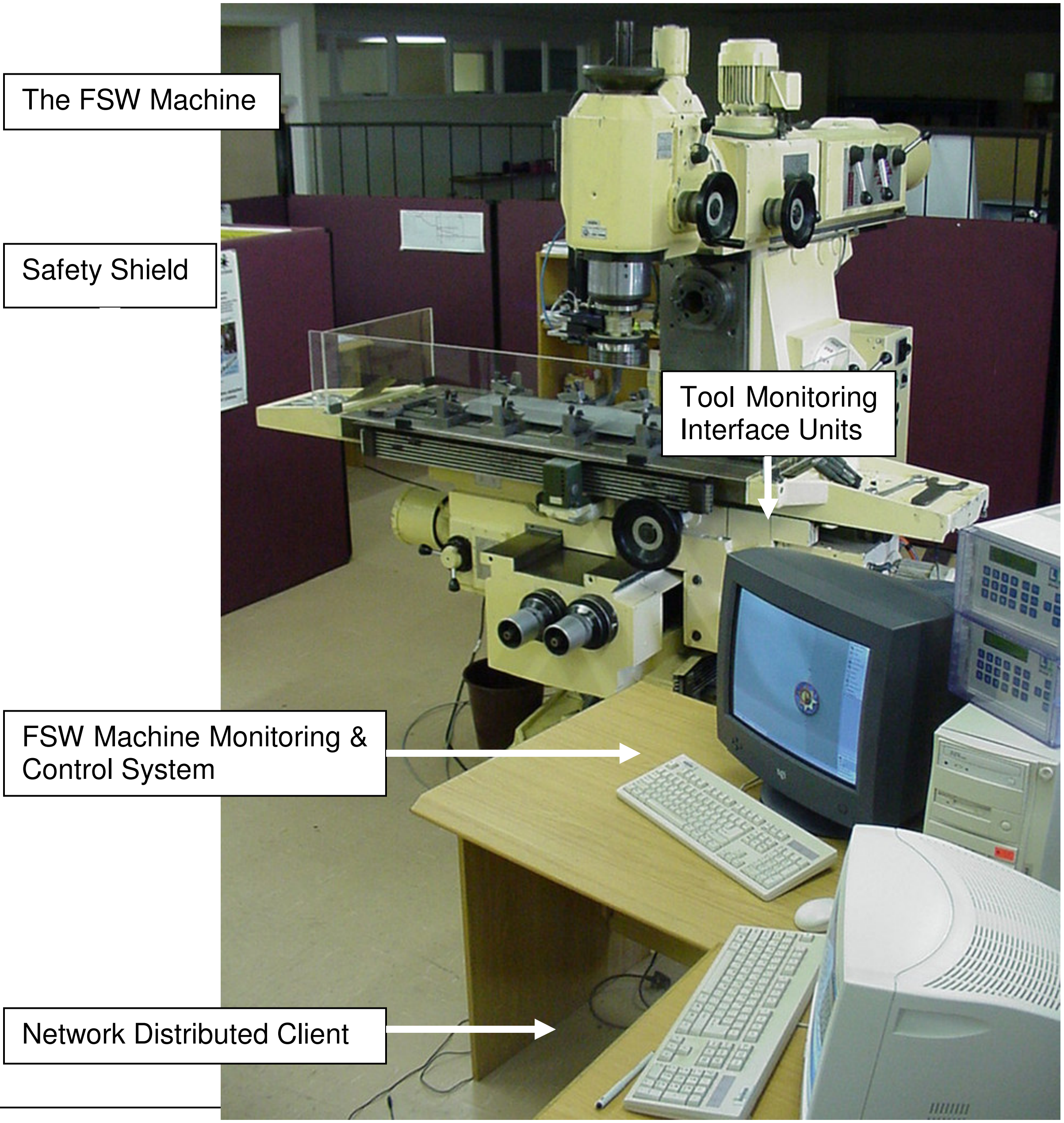

The FSW Machine
Safety Shield
Tool Monitoring Interface Units
FSW Machine Monitoring & Control System
Network Distributed Client

Figure 3.14: Complete experimental setup

The machines inverters, clutches, brakes and limit switches were connected to a computer system using PCI interface cards. The machine was designed to facilitate adaptive control of the process.

An instrumented chuck was designed and built. This allowed monitoring of the process by an operator or closed-loop control of the process. The chuck measures the 2D force distribution, the vertical force, torque and the temperature of the tool.

The interfaces present on the computer and used to communicate with the machine are two RS232 serial ports, two RS485 serial ports, digital I/O board and LAN interface board.

Optical encoders and temperature sensors were fitted to the feed and spindle motors. Fans were mounted in cowlings, secured to the ends of the motors, to provide forced cooling. This is necessary because at slower rotor speeds the standard motor cooling fan, becomes ineffective.

A network-distributed client can be used to control the machine over the Internet. This is done, using the LAN interface board in the FSW machine's computer.

The next chapter will discuss the overview of the software architecture by discussing the concepts, methods and services used to design and implement the FSW software.

CHAPTER 4

Fundamental Concepts used for FSW Software Architecture

The FSW machine intelligent monitoring and control software is required to provide the following functionality:

- Recording of process variables at periodic intervals
- Suitable response times for in process control decisions
- Fast and efficient access to recorded data
- Maintainability, enabling simple, fast future upgrades and modifications
- Scalability ensuring system can expand as the requirements of the research grow
- Reliability, reducing research delays or loss of test data

2.6 Open System Architecture Design (OSA)

“An open architecture controller is flexible in hardware and software. It allows hardware changes to its basic configuration and software changes at all levels of control; this enables the use of advanced monitoring and control techniques” [Schofield, Wright, 1998].

It allows the addition or modification of hardware and software at a later time [Yellowley, Potter, 1992]. Expansion of the system and flexibility for the control system to grow as knowledge about the machining process grows.

Most users of machine tools today are faced with closed, relatively unspecified “control-boxes”. Generally current machine tools are not open to changes in hardware, and are normally only programmable in a fairly inflexible numerical control language. This greatly complicates computer-aided design (CAD) to computer-aided manufacturing (CAM) processing or the integration of sensors, advanced calibration and adaptive control algorithms [Schofield, Wright, 1998].

Adhering to the OSA model is not a guarantee that openness itself will lead to a suitable solution to the intelligent control of manufacturing systems. It will however minimize programmer and operator effort in the setting of limits and appropriate constants in the various models used for process monitoring and control. A system should have a responsive hardware structure, technologically based high level language (C/C++) and suitable process models, which automate much of the technological calculation [Yellowley, Potter, 1992].

Open system controllers are economical, maintainable, modular and scaleable. Therefore they are economical as costs incurred due to lack of available hardware or software is averted [Owen, 1995].

The NGC-SOSAS established a standard virtual controller infrastructure, integrating software-based “services” and “applications” under the supervision of a master controller. This provides for the following [Schofield, Wright, 1998]:

- **Interoperability** – System components must be able to function together in a cooperative manner. This is accomplished by using a “system integrator” which combines a series of independent modules, providing the systems functionality [Schofield, Wright, 1998].
- **Portability** – The same system components must operate on different hardware platforms. This allows the system to move to whichever hardware platform best suits the application, while using the same system components. System flexibility and potential future growth are also increased [Schofield, Wright, 1998].

- **Scalability** – Allows the systems functionality to be increased or decreased by modifying specific system components, avoiding the cost of replacing the entire system.
- **Interchangeability** – System components may be interchanged with functionally identical components, which possibly provide greater capabilities, cost advantages, reliability or performance. This enables key components to be replaced without replacing the whole system [Schofield, Wright, 1998].

An open architecture controller must meet the following requirements:

- Standard hardware architectures must be used, for example, VME or PCI bus and standard processors like the Intel x86 or PowerPC [Owen, 1995].
- Standard operating systems should be used, for example, Unix, Linux, QNX or Windows NT [Owen, 1995].
- It must be programmed in standard languages like Microsoft Visual Basic, Java, C/C++ or X-Windows/Motif [Owen, 1995].
- Its control software must be extendable to let the OEM or user integrate custom control algorithms [Owen, 1995].

The advantages of using an Open System Architecture design are:

- The capacity to expand the system allows more axes to be added, network distributed processing, additional monitoring systems, co-operating machine units, etc promoting system growth and optimization [Yellowley, Potter, 1992].
- A totally open system allows future research to be uninhibited by previous investments in capital equipment and software, as only the modules affected by the change need by altered or replaced [Yellowley, Potter, 1992].
- Also the extension of the control strategy, modification of robust operator interfaces or other planned, but unimplemented additions can be easily completed at a later stage [Yellowley, Potter, 1992].

2.7 Portable Operating System Interface (POSIX)

POSIX is an acronym for Portable Operating System Interface, the main goal being to support application portability at the source-code level. It covers various aspects of operating systems and is continually evolving and is currently being standardized by the Computer Society of IEEE as the IEEE standard P1003. The ISO/IEC is also busy with the international standard ISO/IEC-9945 [Harbour].

The group of POSIX standards can by grouped into three categories:

- **Base standards** – Define operating system related interfaces, which allow application programs to access operating system services. Implementation is however not described [Harbour].
- **Language bindings** – Provide interfaces for various programming languages, which include C, Ada, Fortran 77 and Fortran 90 [Harbour].
- **Open systems environment** – Include a guide to the POSIX environment and application profiles. A list of POSIX standards, their options and parameters whose support is necessary for a particular application environment, is termed an application profile. They are necessary as a means of achieving a small number of well-defined types of operating system implementations appropriate for certain application environments [Harbour].

POSIX is a proposed operating system interface standard based on the UNIX operating system and is necessary because there are many differences between implementations of UNIX based operating systems. The portability of applications is therefore reduced. This is especially pronounced when porting real-time applications to a different platform, due to the vast number of real-time operating systems. Because of this a real-time POSIX extension was developed, which includes the following standards [Harbour]:

- **POSIX.4: Real-time extensions** – Defines interfaces supporting the portability of applications with real-time requirements [Harbour].
- **POSIX.4a: Threads extension** – Defines interfaces to allow multiple control threads support within each POSIX process [Harbour].
- **POSIX.4b: Additional real-time extensions** – Defines interfaces for additional real-time services [Harbour].
- **POSIX.13 – Real-time application environment profiles** – Each profile lists the services necessary for a certain application environment [Harbour].

The POSIX.1b real-time extension defines realtime as “the ability of the operating system to provide a required level of service in a bounded response time [Butenhof, 1997].”

Whatever applies to the operating system, applies to the application. A bounded response time is the ability for the operating system to provide a predictable response, not necessarily a fast response. Realtime programming can be divided into “Hard real-time” and “Soft real-time” [Butenhof, 1997].

In a “Hard real-time” system the required level of service must be provided 100% of the time as the bounded response time is defined by a physical quantity or something equally unyielding. Consequences of a system that fail to meet the schedule are severe and can result in severe damage, injury or a fatality [Butenhof, 1997].

A “Soft real-time” system does not have catastrophic failure modes; therefore a less rigorous approach to their design can usually be followed [Kopetz, 1997].

Realtime means carefully giving precedence to parts of a program so that external response time can be limited [Butenhof, 1997].

2.8 The QNX Real-time Operating System

QNX is an embeddable real-time POSIX operating system (OS) in a robust, scalable form, providing open systems standards, wide scalability and high reliability, suitable for a wide range of applications. These include tiny, resource-constrained systems to high-end distributed computing environments [QNX software systems, Architecture Guide, 2002]. This OS was selected for the development of the FSW software to enable the goals of the software to be achieved.

The QNX OS uses a “microkernel” architecture, in which a tiny kernel provides the basic services used by a team of optional cooperating processes (file systems, etc). These in turn provide the higher-level OS functionality [QNX software systems, Architecture Guide, 2002].

The OS consists of the small Neutrino microkernel managing a group of cooperating processes, where each process provides a service, as shown in Figure 4.1. Each user written process can serve either as an application or a service to other applications, making the OS “open” and easily extensible. This allows the OS functionality to be extended for industry-specific applications [QNX software systems, Architecture Guide, 2002].

QNX utilizes the memory management unit (MMU) to provide each process complete memory protection for user and OS component processes.

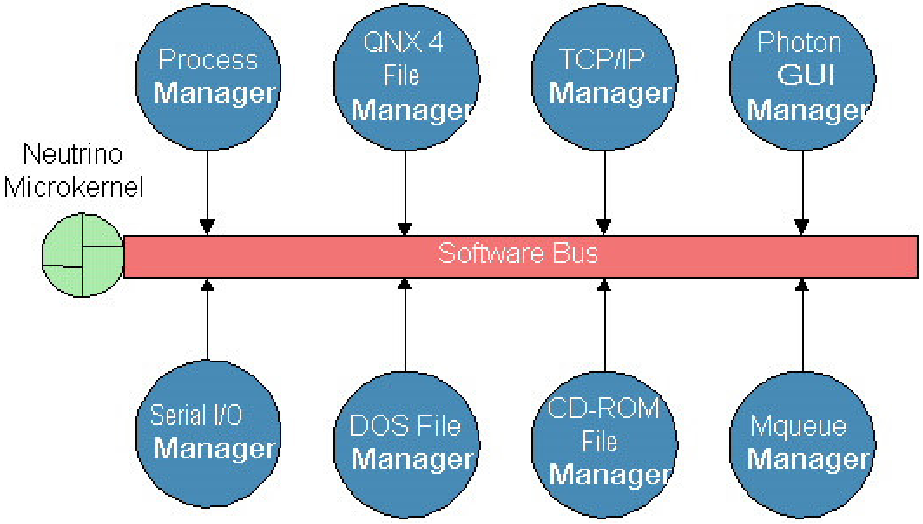


Figure 4.1: The QNX architecture [Krten, 1999]

Figure 4.1 shows how cooperating processes are managed by the Neutrino microkernel. Managers can be dynamically added or removed as needed [QNX software systems, Architecture Guide, 2002]. QNX message passing forms a virtual software bus that allows modules to be added (or removed) to deliver services. This results in modularity, allowing the creation of highly reliable, serviceable and scaleable systems [QNX Software Systems, 2002].

The Neutrino microkernel proves the following fundamental services:

- **Thread services** – Used for creating POSIX threads

- **Signal services** – Neutrino provides POSIX signal primitives
- **Message-passing services** – Message routing between threads running on the system
- **Synchronization services** – Used for POSIX thread synchronization.
- **Scheduling services** – Thread execution scheduling, using POSIX real-time scheduling algorithms
- **Clock, timer and interrupt services** – POSIX services
- **Process management services** – For managing processes, memory and pathname space

All of the above services were used during the design and implementation of the FSW software.

Several threads run concurrently in typical real-time multitasking environments and inter-process communication (IPC) must be used to allow cooperation to form a complete system. QNX implements its IPC as message passing, where a parcel of bytes is passed from one process to another. Only the sender and receiver processes need to understand the meaning of the bytes. Message passing also provides a means for synchronizing the execution of several processes. Real-time applications require a dependable form of IPC, because processes forming the system are strongly interrelated.

Neutrinos message-passing design brings order and reliability to applications [QNX software systems, Architecture Guide, 2002].

Threads existing on the system can be in any of the following three general states:

- **Running** – The thread is actively consuming CPU time. On a symmetrical multi-processing system (SMP) multiple threads could be running, but on a single-processor system only one thread will be running [Krten, 1999].
- **Ready** – A thread in this state is able to run right now, but is not, because a thread of equal or higher priority is currently running on the CPU [Krten, 1999].
- **Blocked** – A blocked thread is waiting for a resource to become available or an event to occur. The thread, while blocked, will not consume and CPU time [Krten, 1999].

2.9 Processes and Threads

Threads are extensively used for the FSW software and are encapsulated by its processes, forming the software architecture.

A thread is a single, schedulable flow of execution. Their functionality is implemented directly within the Neutrino kernel and adheres to the POSIX convention. Synchronization techniques can be used to control the execution of multiple threads with regard to each other. The following synchronization techniques can be used to control thread execution with other threads:

- **Mutual Exclusion Lock (mutex)** – A Mutual Exclusion lock only allows one thread at a time access to the resource defined by the mutex.
- **Condition variable (condvar)** – Condition variables are used to synchronize multiple threads. Threads will wait at a condvar until a signal is given to allow them all to continue execution.
- **Barrier** – A barrier is also used to synchronize multiple threads. Each thread will wait at a barrier, until the specified number of threads is waiting, at which time all the waiting thread will be allowed to continue execution.
- **Semaphore** – A semaphore is similar to a barrier, except that threads can either increase or decrease a count. When this count is zero, threads will wait until the count becomes non-zero before continuing execution [Krten, 1999].
- **Sleepon locks** – This thread synchronization method allows a thread to block, until a specified state change occurs elsewhere in the system (for example the completion of a data transfer by another thread) [Krten, 1999].

- **Reader/writer locks** – One thread is allowed to exclusively lock a shared data area so it can write to the data. Other threads are allowed to simultaneously lock the data for read access. This ensures only one thread may write to the data at an instant, later multiple threads may read the data simultaneously if the data is not being written [Butenhof, 1997].
- **Spinlocks** – A spinlock is the most primitive but fastest synchronization mechanism available (similar to a mutex), except the thread will not block if another thread has the lock. The thread will continually try to gain access, until the thread holding the lock releases it or the threads time-slice runs out [Butenhof, 1997].

Thread execution can be scheduled in any of the following modes:

- **FIFO scheduling** – A thread will continue executing on the CPU until a higher priority thread is ready to run, or until another thread gives up the CPU. If neither of these conditions occur, the thread will continue executing indefinitely.
- **Round robin scheduling** – Similar to FIFO scheduling, except the thread will also voluntarily give up the CPU if its time-slice expires and another thread at the same priority is ready to run [Krten, 1999].

- **Adaptive scheduling** – If a thread consumes its time-slice, its priority is reduced by 1, called priority decay. Its priority will only drop one level below its original priority, even if it consumes another time-slice. When the thread blocks (another thread executes), its priority will be reverted immediately to its original priority.
- **Sporadic scheduling** – The thread will continue executing until it blocks or is preempted by a higher-priority thread. Its priority will then drop, as with adaptive scheduling, however more precise control is possible [QNX software systems, Architecture Guide, 2002].

Round robin scheduling was primarily used for the FSW software to ensure all threads were able to gain access to CPU resources when necessary.

Threads are useful for all types of realtime programming as programming for a predictable response is much easier when operations can by kept separate [Butenhof, 1997].

A process is a non-schedulable container entity for threads, which occupies memory. Processes work together to form a system, which performs some goal [Krten, 1999].

A hierarchical series of processes was used to build the FSW software. The main reasons for using processes to break up system were:

- **Decoupling and modularity** – Solving many small problems is easier than solving one large problem. Processes do not need to rely on each other much and can focus on one particular responsibility. The only “reliance” the modules have on each other is through a small number of well-defined interfaces.
- **Maintainability** – Because each module is separate to the other modules and its purpose is well defined, modifications or corrections are relatively easy.
- **Reliability** – Each process’s memory address space is maintained and enforced by Neutrino’s process manager module. This means two threads, running in different processes are isolated from each other. Figure 4.2 illustrates this protection. The process manager module will prevent processes inadvertently accessing each other’s memory space, thereby increasing the reliability of the system [Krten, 1999].

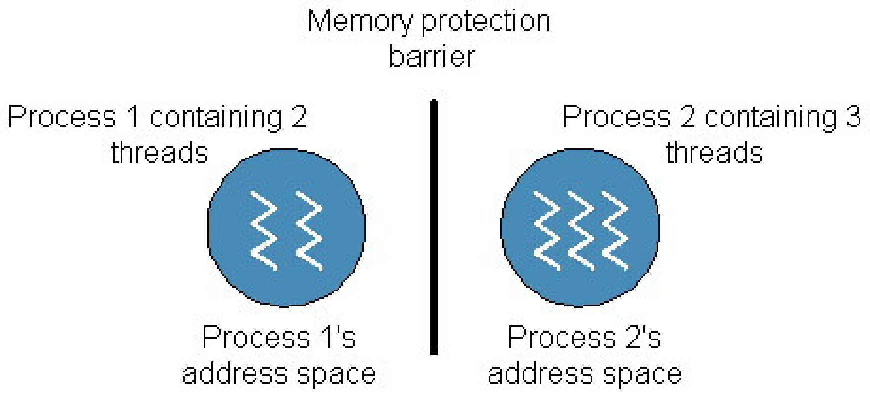


Figure 4.2: Example of memory protection [Krten, 1999]

The benefits of the multithreaded software model, used for the FSW software, are as follows:

- Program parallelism can be exploited on multiprocessor hardware [Butenhof, 1997].
- A program's natural concurrency can be more efficiently utilized by continuing to perform computations while waiting for slow I/O operations to complete [Butenhof, 1997].
- Multithreaded software follows a modular programming model where relationships between independent "events" with a program can be expressed [Butenhof, 1997].

Multithreading also introduces some challenges:

- It may lead to performance losses because the multithreaded implementation adds thread synchronization and scheduling overhead to the execution time of the program [Butenhof, 1997].
- Good sequential code is not necessarily good threaded code. Bad threaded code can cause problems that are more difficult to locate and repair. Threaded programming requires more detailed work to ensure it executes correctly and efficiently [Butenhof, 1997].
- Multithreaded code is harder to debug than sequential code. Debugging requires special techniques that must be learnt. It also changes the timing of events, which may prevent the error for reoccurring. Errors can also be caused by a transgression of locking protocol and in many cases a code review is most useful [Butenhof, 1997].

Generally, most systems can be abstracted into three general categories, which include the input module, the processing module and the output module. This input/process/output model can be applied to a problem in various ways.

This generalization can be seen in Figure 4.3, where the arrows represent some form of communication path with its related protocol. Possibilities include, but are not limited to: pipes, POSIX message queues or native Neutrino message passing. This form of the model is the most loosely coupled, meaning each process is not affected by the implementation of the other processes. The only reliance they share is the communication path and its protocol [Krten, 1999].

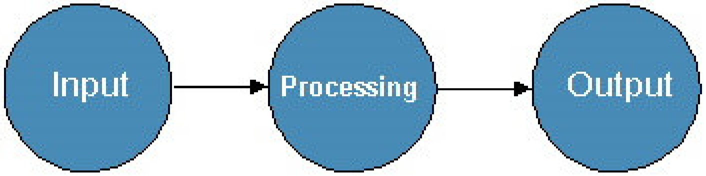


Figure 4.3: Multiple processes performing multiple operations [Krten, 1999]

The advantages of this implementation are:

- Process can be easily replaced by loading a different process and not having to recompile or redesign.
- The system can be distributed among multiple nodes in a Neutrino network. Neutrino supports network distributed IPC, so the communication path can be abstracted over a network using the QNET protocol. This provides excellent scalability to the design [Krten, 1999].

This method was used to combine the co-operating processes used to form the FSW software architecture.

Figure 4.4 is another form of the model, which can be used to optimize the communication path, depending on the volume of information needing to be transferred. This implementation increases the coupling of the processes, by using a shared memory scheme. This scheme results in faster more efficient data flow, which is shown by the thick arrows. The thinner arrows represent the flow of low bandwidth control information.

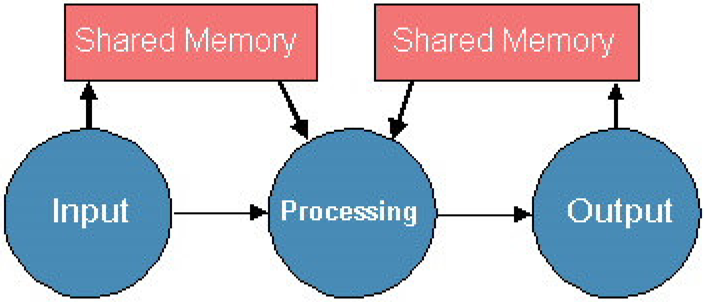


Figure 4.4: Multiple processes communicating via shared memory [Krten, 1999]

This system cannot be network distributed, as Neutrino does not support network-distributed shared memory objects. It is still easy to change or modify processes, but a constraint has been added where all processes must adhere to the shared memory model [Krten, 1999]. This model would be used where high data transfer rates are necessary. An example of such a situation would be the application of a camera to monitor the process.

The use of one process containing multiple threads, as shown in Figure 4.5, represents the tightest form of coupling, where threads share data areas implicitly. Control information can be sent using various forms of IPC or via thread synchronization primitives [Krten, 1999].

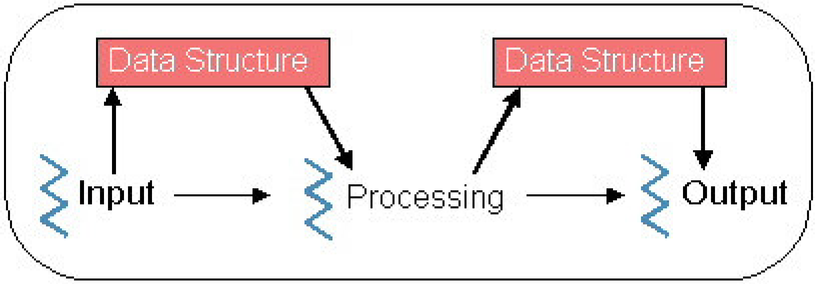


Figure 4.5: Multiple threads performing multiple operations within a process [Krten, 1999]

Each of the processes forming the FSW software architecture uses this model for their internal structure. This system has, however, lost the configurability and scalability advantages mentioned for Figures 4.3 and 4.4. It is however the fastest as there are no thread-to-thread context switches necessary for threads in different processes. Shared memory regions or forms of IPC do not need to be set up or used. Simple thread synchronization primitives can be used. Another advantage is that at startup all necessary information and resources are available, there is no possibility that a process manager necessary later has not been loaded. This system is also likely to occupy the least memory, because multiple copies of process information do not need to be stored [Krten, 1999].

Thread pools are used in applications requiring a certain number of threads performing the same function, where the behavior of the threads needs to be controlled within certain limits. A common application requiring this technique would be a server handling requests from multiple clients, such as a resource manager or web server. This allows multiple requests to be processed without clients waiting for other clients' requests to complete. Each resource manager constituting the FSW software uses a thread pool to service client requests.

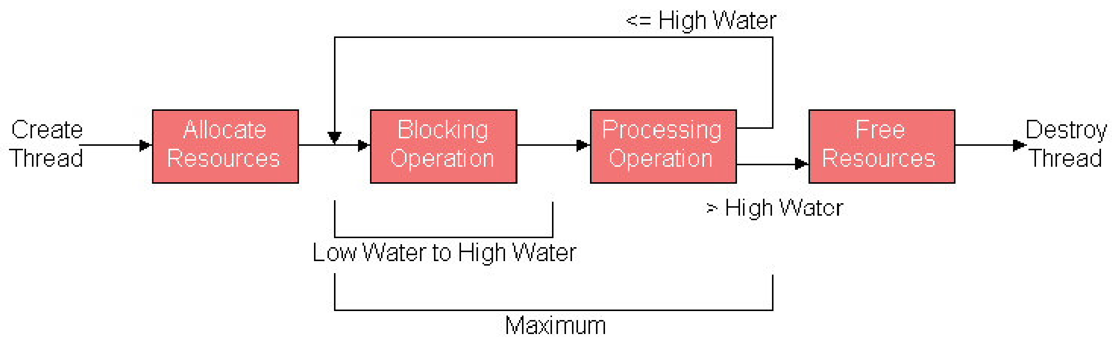


Figure 4.6: Thread flow when using thread pools [Krten, 1999]

Figure 4.6 shows the regulation of the thread pool between the low water and high water marks. If the numbers of threads in the blocking state are below the low water mark, n number of threads are created, where n is a specified increment. Threads will continue to be created, until the number of threads exceeds the maximum limit. The maximum limit sets the total number of blocked and running threads allowed in the pool. When a thread finishes it checks high water limit and if there are too many blocked threads, it kills itself. The high and low water limits allow an operating range to be specified where threads are not necessarily continually created and destroyed [Krten, 1999].

The priority of threads can be set. If two threads are capable of using the CPU, the thread with the highest priority will be executed first. In the event of a thread executing and a thread with a higher priority become ready to use the CPU, the kernel will context-switch to the higher priority thread. This process is called preemption [Krten, 1999].

There are three main reasons why rescheduling of a thread occurs:

- Hardware interrupts (timers or other hardware)
- A kernel call (timer functions, message functions, thread primitives)
- A fault (exception) [Krten, 1999]

Priority inversion is a severe problem with fixed priority scheduling and occurs when a low-priority thread prevents a high-priority thread from running due to interaction between scheduling and synchronization [Butenhof, 1997].

Assume a system containing three threads, with low, medium and high priorities. The high priority thread begins executing, but blocks on a mutex shared by the low priority thread. The medium priority thread now executes and the low priority thread cannot release the mutex until the medium priority thread finished using the CPU. The executing of the high priority thread is delayed due to a lower priority thread [Kopetz, 1997]. This is an example of priority inversion.

2.10 Message Passing

All services in the Neutrino operating system are provided in a synchronous manner by passing messages from client to server. The client will send a message to the server and then block. The server will receive a message from the client; perform some processing, and the reply to the client's message, which will unblock the client [Krten, 1999].

There are three possible models, which can be used with message passing:

- **Client/server model of message passing**
- **Multiple threads**
- **Server/sub server**

For Client/server message passing, the server is initially received blocked. When a message is received the server enters the ready state and can run on the CPU if it is the highest priority ready process [Krten, 1999]. This model is used to service client requests to the resource managers forming the FSW software.

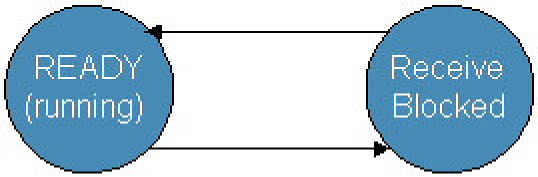


Figure 4.7: Server state transitions [Krten, 1999]

The server analyses the received message and perform the requested operation, after which a reply will be sent to the client. Figure 4.7 shows this server state change when processing a message [Krten, 1999].

Figure 4.8 shows the client's state changes when communicating with a server. When the client sends the server a message, its state changes from ready to send-blocked or reply-blocked depending of the state of the server.

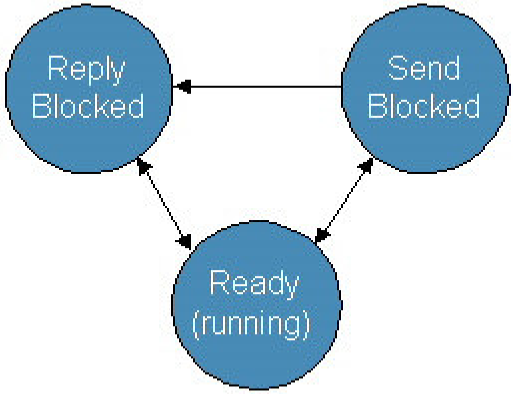


Figure 4.8: Client state transitions [Krten, 1999]

A send-blocked client means that the server has not received the message yet or that the server is busy servicing another client. The reply-blocked state is more common and occurs while the client is waiting for the server to complete the processing of the request and respond. This model is termed send-driven because processing begins when the client sends a request [Krten, 1999].

The multiple threads model allows a pool of threads to be created, where each thread in the pool is capable of handling a message from the client. This model allows the kernel to multitask to server among various clients, without the server performing the actual multitasking. This is especially powerful on SMP machines [Krten, 1999].

In the server/sub-server message passing model a server creates a series of sub-servers, unknown to the client. These sub-servers send a message requesting work from the server. Only when the server receives a message for the client, will it reply to the sub-server's request. The server/sub-server model is especially useful for network distributed processing applications. This model is termed reply-driven as the request is only processed once the reply to the sub-server is sent [Krten, 1999].

The message passing discussed thus far causes the client to block when a request is sent. Another form of message is termed a pulse and is non-blocking for the sender. It contains a small payload and can be received using standard techniques. A pulse is usually used for sending small asynchronous signal messages from a server to a client so not to break the send hierarchy discussed earlier [Krten, 1999]. The FSW software uses pulses to inform higher-level processes of changes in the state of the digital inputs.

2.11 Resource Managers

A resource manager is a server process, which provides certain well-defined file-descriptor-based services to arbitrary clients (e.g. The file //dev/fd0 can be used to access the floppy disk controller). An abstraction exists between the client and the server, as both do not need to know each other's internal implementation. The client uses standard POSIX function calls (fprintf, fgets, lseek, etc) to interact with the server. Neutrino converts these POSIX function calls into messages, which are sent between client and server. All a server needs to do is handle certain well-defined messages, which the client has generated [Krten, 1999].

Resource managers are used extensively for the FSW software to provide modularity, reliability and to extend the functionality of the OS for the FSW process.

Figure 4.9 and 4.10 show the stages necessary for a client to connect to a resource manager. Firstly, the client queries the resource manager with the path name of the required process manager (1). The process manager will then reply with the process ID, channel ID and handle to the requested resource manager (2).

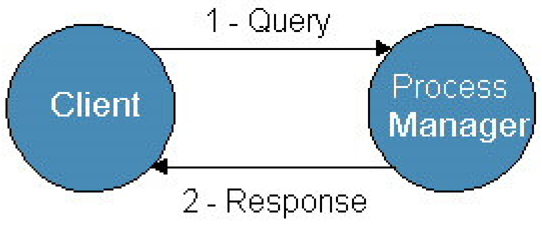


Figure 4.9: First stage of resource manager name resolution [Krten, 1999]

This is used to send a connect message to the process manager (3). The process manager performs validation of the connection request, checking file attributes and allocating resources. A reply is the send back to the client containing a connection ID, which can be used to send messages in the future.

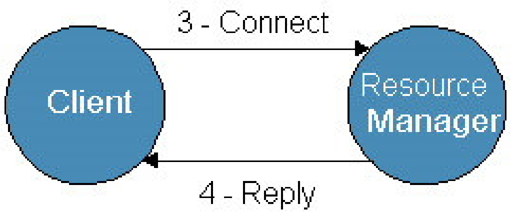


Figure 4.10: Client attempting to connect to resource manager [Krten, 1999]

The standard C open () function handles this name resolution process transparently and simply returns the connection ID for the requested resource [Krten, 1999].

The general structure of a resource manager can be seen in Figure 4.11. A thread pool allows multiple requests from multiple clients to be processed. The resource manager must register with the process manager, informing it which pathnames are under their domain of authority. The resource manager resolves the path name to the node descriptor, process ID and channel ID of that process manager [Krten, 1999].

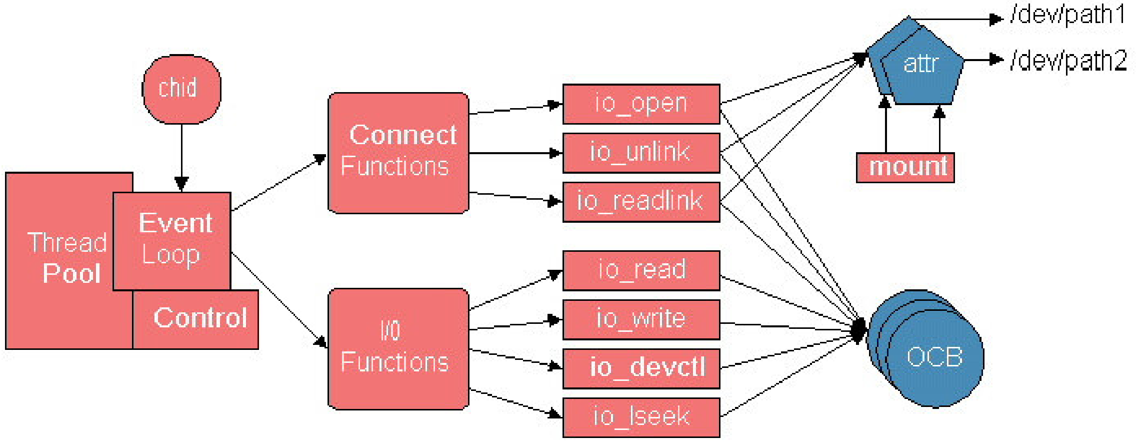


Figure 4.11: General architecture of a resource manager [Krten, 1999]

Once the path names have been registered, the resource manager can prepare to receive messages from clients, using standard POSIX message passing techniques. Resource managers have a set of well-defined message types they can handle, which can be divided into two groups:

- **Connect messages** – Always contain a pathname and the handler for the connect messages used to establish a context for further I/O messages. An example would be the C function open () function [Krten, 1999].
- **I/O messages** – Use the context created during the handling of a prior connect message, to communicate with the resource manager. The read and write C function calls are good examples of I/O messages [Krten, 1999].

2.12 FSW Machine Monitoring and Control Software Architecture

The following section will discuss the use of multiple processes performing multiple operations resulting in the formation the FSW system, shown in Figure 4.12.

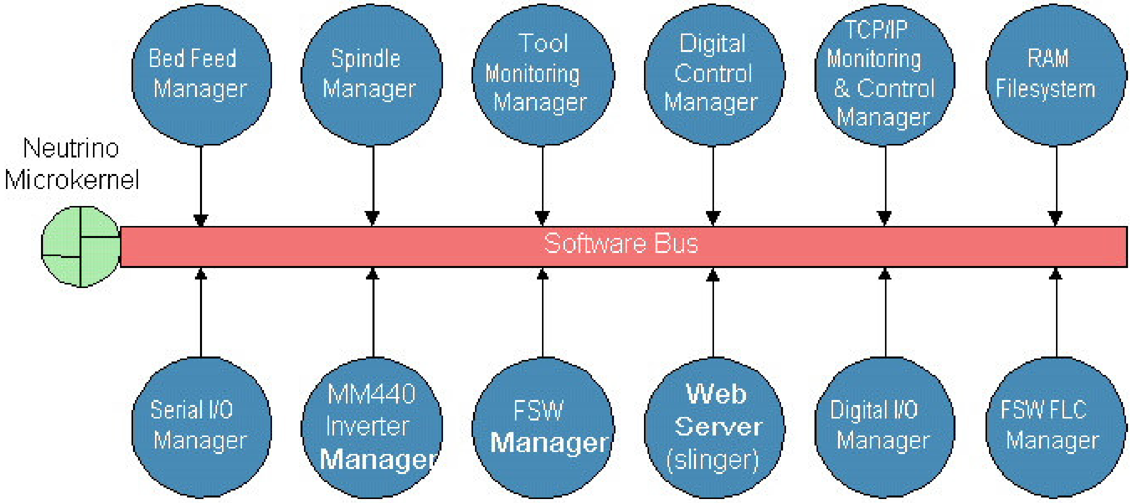


Figure 4.12 Processes forming the FSW Intelligent Monitoring and Control System

Each process is a resource manager, expanding the services of the Neutrino Microkernel. The basic services offered by each of the process managers in Figure 4.12 are as follows:

- **Digital I/O manager** – This process forms the interface between higher-level managers and the hardware (physical layer). It locates the address and interrupts of the Access IO digital IO PCI card on the PCI bus. These are used to read the state of the digital inputs or set the state of the digital outputs. Digital inputs include the state of the power contactor, bed limit switches, e-stop state and power supply output. Digital outputs control the state of the clutches, brakes and power contactor.
- **Serial I/O manager** – The serial manager (devc-ser8250) is supplied with the QNX operating system. It provides the interface between other processes and any serial UART controllers existing in the system. This driver is used to provide access to the two RS232 ports and four RS485 ports on the system.
- **MM440 inverter manager** – This manager handles the monitoring of parameters and inverter status, including the sending of control commands to the Micromaster 440 inverter. It is responsible for the encoding and decoding of data packets using the USS protocol specification. These packets are then transmitted and received by the serial I/O process manager. This manager can handle multiple inverters connected to the system via multiple serial ports.
- **Digital control manager** – The digital control manager provides interlocking of the brakes and clutches and transforms the raw 16bit integers returned by the digital I/O manager into a more meaningful format.

- **Spindle manager** – The spindle manager customizes the service provided by the MM440 inverter manager for monitoring and recording the parameters and control of the spindle motor.
- **Feed manager** – This manager customizes the service provided by the MM440 inverter manager for monitoring and recording the parameters and control of the machine’s bed feed motor and associated clutches and brakes. This manager also provides relative positioning and basic feed functions.
- **Tool manager** – Recording and processing of tool parameters from the chuck are conducted by this manager for the higher level processes.
- **Ram file system** – A simple RAM-based file system that allows the read/write of files is supplied with the QNX operating system [QNX software systems, Architecture Guide, 2002].
- **FSW manager** – This manager controls the welding process, collecting all process data from the lower levels and using it to implement machine protection and safety limit checking. It also performs the machine sequences necessary to complete a Friction Stir Weld and makes process data available to higher-level processes.
- **FSW FLC manager** – A fuzzy logic controller could be implemented, allowing multiple process variables to be controlled during the welding sequence. Intelligence implemented in a rule base is used to make control decisions.

- **Web server** – Neutrino is supplied with its own small memory footprint HTTP web server, called Slinger. A comma delimited ASCII text file containing the recorded process data and the in-process monitoring and control provided by the Macromedia Flash GUI, can be accessed via this web server.
- **TCP/IP monitoring & control manager** – Process variables and machine status are obtained from this FSW manager and are made available to remote client requests over a network using the TCP/IP protocol. The manager also accepts commands to enable user initiated process control. This manager also provides the data the GUI will use to display process information.

One module not shown on figure 4.12 is the actual graphical user interface (GUI), which is implemented using Macromedia Flash 5 and provides the front end of the monitoring and control system. This front end consists of the visual display of process variables and enables end-user control of the weld sequence. The Flash GUI is embedded into an Internet web page, which is made available by the Slinger web server.

2.13 Summary

The FSW software architecture has been selected, based on popular software design concepts. These concepts include Open System Design, the Portable Operating System Interchange specification, soft-realtime execution and modular construction based on functionally abstracted layers.

The FSW control software must be able to record process variables, have a good response time, access recorded data, allow software modification and expansion and be reliable. This was achieved using a real-time operating system that supports the POSIX standard and designing the software based on Open System Architecture principles.

The QNX operating system was selected as a platform for the development of the software. This operating system uses a microkernel architecture. This allows each of the FSW software modules to be memory protected and scaleable. Message passing is used to enable inter-process communications.

There are three models that can be used with message passing, which are the client/server, multiple thread and server/sub-server. The first and second models are used for inter-process and internal process communication respectively.

The operating system provides various services that were used to implement the FSW software. These services include: thread, signal, message-passing, synchronization, scheduling, timer, interrupt and process management.

The real-time ability of the operating system, allows predictable response times from the software.

Processes are non-schedulable container entities for threads. The FSW software uses multi-processes implemented as resource managers to extend the functionality of the operating system, suiting it to the FSW process. Processes consist of threads that implement this functionality. Access by the threads to resources need to be synchronized by using mutual exclusion locks, condition variables, barriers, semaphores, sleepon locks, reader/writer locks and spinlocks. Thread execution can be scheduled using the following modes: FIFO, round robin, adaptive and sporadic. QNX only supports FIFO and round robin. The latter was used for the FSW software to ensure equal access to resources by all the threads.

The software uses thread pools to service multiple client requests simultaneously, increasing system response time. Thread priority can be set allowing a hierarchical structure to service threads requiring CPU time.

The FSW software consists of the following processes:

- Digital I/O resource manager
- Serial I/O resource manager
- Siemens Micromaster MM440 resource manager
- Control resource manager
- Spindle resource manager
- Feed bed resource manager
- Tool monitoring resource manager
- RAM file system resource manager
- FSW sequence resource manager
- Slinger web server process
- TCP/IP FSW data server process

The following chapter will discuss the implementation and internal structure of the FSW software, using the techniques discussed in this chapter. The code to create a resource manager is initially presented, as most of the software modules are resource managers. Then each software module is discussed in depth pertaining to its function, structure and interface.

CHAPTER 5

Software Implementation

This chapter will discuss the implementation of the monitoring and control software for the FSW machine to enable FSW welds to be performed.

ANSI C was predominantly used due to the following reasons:

- It is widely recognized with a large following.
- Relatively simple to learn and has a large knowledge base.
- Adding classes and OOP techniques to the already many small modules could have reduced code readability.
- The operating system promotes an OO-type programming structure using resource managers.
- Most of the instructional and sample code provided with the OS was written in C.

- The C language is extensively used with embedded systems; thereby future embedding of the software into a controller would be relatively simple.

Some C++ was used, but only where its use would result in a large benefit to the software.

Architectural design is the process of decomposing large systems into sub-systems to provide some, related set of services and results in a system's software architecture. System architecture affects its performance, robustness, distributability and maintainability [Sommerville, 2001].

The process of architectural design can be broken into the following processes:

- **System structuring** – A system is structured into a number of principle sub-systems where a sub-system is an independent software unit. Communications between sub-systems are identified [Sommerville, 2001].
- **Control modeling** – A general model to the control relationships between the parts of the system is established [Sommerville, 2001].
- **Modular decomposition** – Each identified sub-system is decomposed into modules. The types of modules and their interconnections must be decided upon [Sommerville, 2001].

A sub-systems operation does not depend on services provided by other sub-systems and is composed of modules, with defined interfaces for communication with other sub-systems [Sommerville, 2001].

A module is comprised from a number of simpler system components, providing one of more services to other modules. It makes use of services provided by other modules and is not usually considered an independent system [Sommerville, 2001].

The abstract machine model (sometimes called the layered model) was used for the FSW systems software architecture. The system is organized into a series of layers, each of which provides a set of services. Each layer defines an abstract machine whose machine language (services) is used to implement the next level of abstract machine [Sommerville, 2001]. This layered model is shown below, in Figure 5.1.

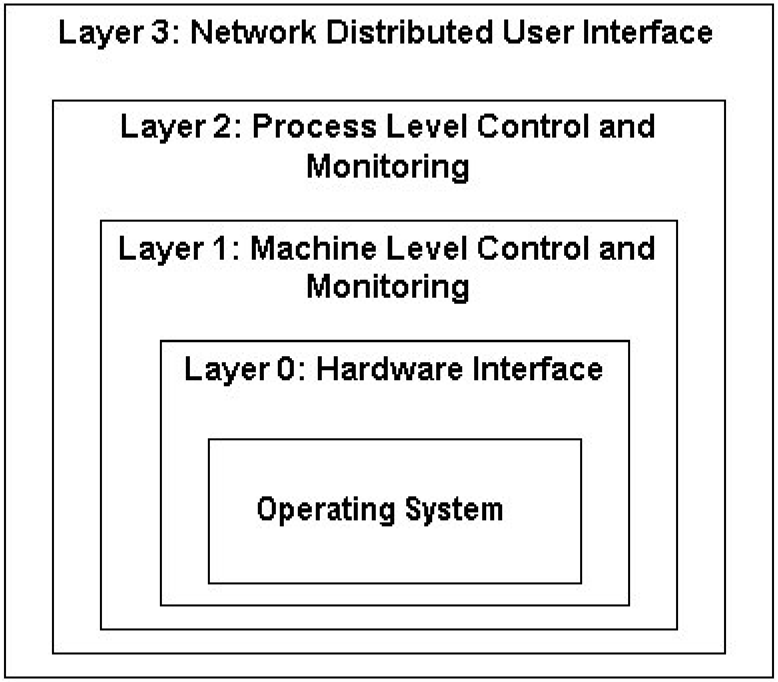


Figure 5.1: Layered architecture of monitoring and control software

A well-known example of this approach is the OSI reference model of network protocols [Sommerville, 2001].

The layered approach supports incremental system development. This architecture is changeable and portable. If the interface is preserved, another layer can replace a layer. When layer interfaces change, only the adjacent layer is affected. Because layered systems localize machine dependencies in inner layers, they can by implemented on other computers relatively cheaply. Only the inner, machine-dependent layers need be re-implemented [Sommerville, 2001].

A disadvantage of this architecture is that certain abstract machines may need access to basic facilities, many layers below. This subverts the model, as the outermost layer is not dependent on its immediate predecessor. Care must be taken to prevent this. Performance can also be a problem due to the multiple levels of command interpretation possibly required [Sommerville, 2001]. Figure 5.2 presents the complete UML static object diagram showing the modular architecture of the FSW software and the modules associated interfaces.

The FSW software uses an event-driven control model. An event, in this context, can take a range of values. The difference between an event and a simple input is that an event’s timing is outside the control of the event handler [Sommerville, 2001].

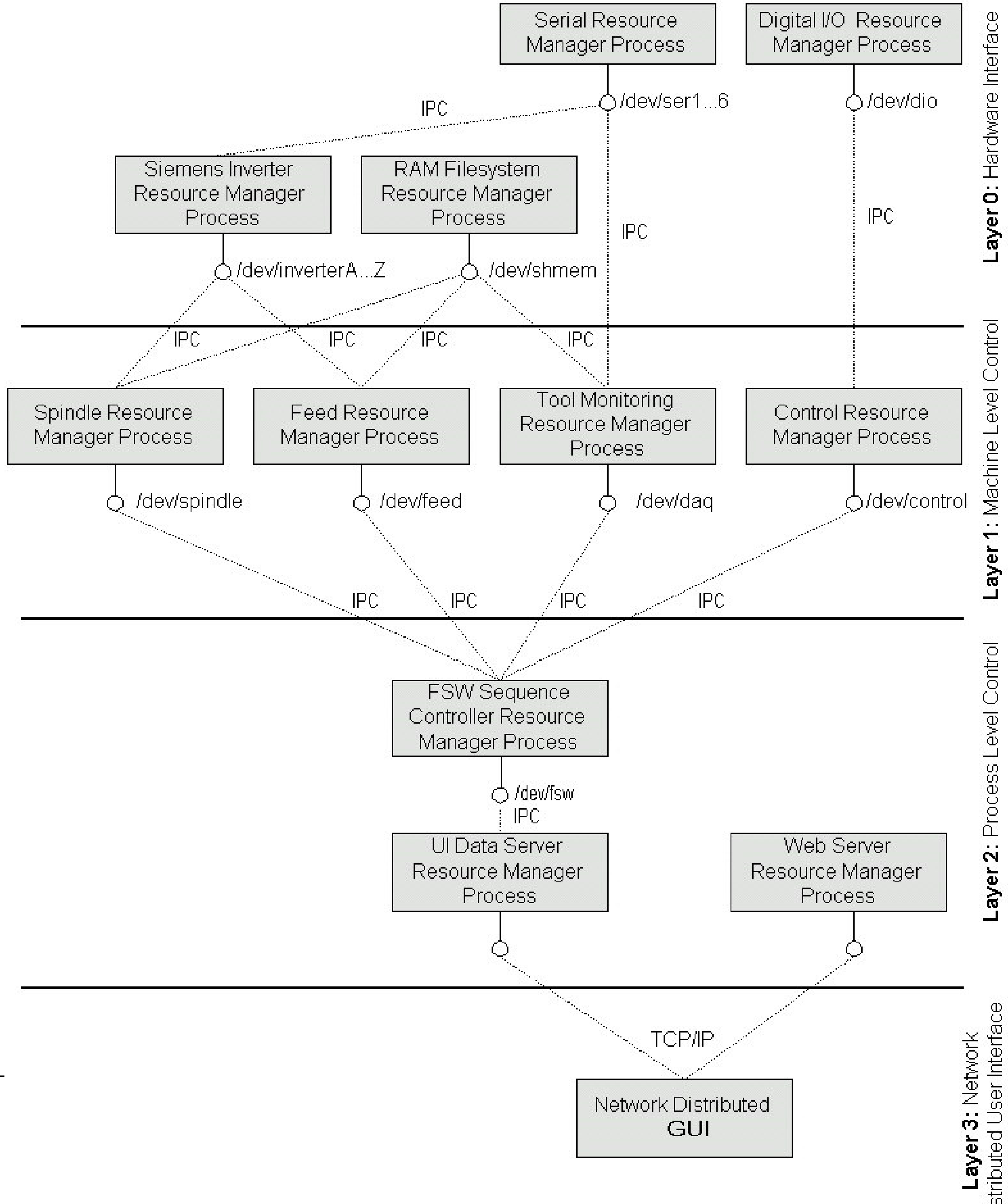

Serial Resource Manager Process
Digital I/O Resource Manager Process
/dev/ser1...6
/dev/dio
IPC
Siemens Inverter Resource Manager Process
RAM Filesystem Resource Manager Process
IPC
IPC
/dev/inverterA...Z
/dev/shmem
IPC
IPC
IPC
IPC
Spindle Resource Manager Process
Feed Resource Manager Process
Tool Monitoring Resource Manager Process
Control Resource Manager Process
/dev/spindle
/dev/feed
/dev/daq
/dev/control
IPC
IPC
IPC
IPC
FSW Sequence Controller Resource Manager Process
/dev/fsw
IPC
UI Data Server Resource Manager Process
Web Server Resource Manager Process
TCP/IP
Network Distributed GUI
Layer 0: Hardware Interface
Layer 1: Machine Level Control
Layer 2: Process Level Control
Layer 3: Network
stributed User Interface

Figure 5.2: UML static object diagram of modules and associated interfaces constituting the FSW software

The data-flow model was used when considering the modular decomposition of the FSW software. In this model, functional transformations process their inputs and produce outputs. Data flows from one to another and is transformed from an input to an output, as it moves through the sequence. The transformations for the FSW software are represented as separate processes (pipe model). A pipe forms a data conduit where a set of commands performs functional transformations [Sommerville, 2001].

Advantages of the data-flow model architecture are [Sommerville, 2001]:

- Reuse of transformations are supported.
- It is intuitive that many people think of their work in terms of input and output processing.
- It is usually straightforward to evolve the system by adding new transformations.
- It is simple to implement either as a concurrent or a sequential system.

The main disadvantage of the model is the need for a common format for data transfer, which can be recognized by all transformations. Either a standard format for all transformations must be decided upon, or each transformation must agree with its communicating transformation [Sommerville, 2001].

The FSW software architecture makes extensive use of resource managers to extend the functionality of the operating system. Therefore, the following section will discuss the critical code necessary to initialize and implement a resource manager. Eight of the ten system modules are resource managers, incorporating the code that is to be discussed. Listing 5.1 shows the general stages used to configure a resource manager.

```
dpp = dispatch_create();                     //(1) initialize dispatch interface

memset (&resmgr_attr, 0, sizeof resmgr_attr);//initialize resource manager attributes
resmgr_attr.npartx_max = 1;
resmgr_attr.msg_max_size = 2048;

iofunc_func_init (_RESMGR_CONNECT_NFUNCTS, &connect_funcs,  //(2) initialize functions
                      RESMGRIO_NFUNCTS, &io_functs);        //for handling messages
io_funcs.read = my_io_read;                  //give address of read, write and devctl
io_funcs.write = my_io_write;                //handler functions which resource manager
io_funcs.devctl = my_io_devctl;                      //must use.

iofunc_attr_init (&attr, S_IFNAM | 0666, 0, 0);             //(3) initialize device
               //attribute structure

id = resmgr_attach (  dpp,                   //(4) dispatch handle
                      &resmgr_attr,          //resource manager attributes
```

```
                       “/dev/myname”,        //device name
                       _FTYPE_ANY,           //open type
                       0,                    //flags
                       &connect_functs,      //connect routines
                       &io_functs,           //I/O routines
                       &attr);               //handle

ctp = dispatch_context_alloc (dpp);          //(5) allocate a context structure
```

Listing 5.1: Device resource manager initialization skeleton [QNX software systems, Programmers Guide, 2002]

From the above listing, the required stages are as follows:

- **Initialize the resource manager’s attributes (1) –** This is used to set the number of structures available for server replies and the minimum received message buffer size [QNX software systems, Programmers Guide, 2002].
- **Initialize functions used to handle messages (2)** – Pointers to the functions used to handle connect and I/O requests are stored into two separate tables (connect_funcs and io_funcs) [QNX software systems, Programmers Guide, 2002].
- **Initialize the attribute structure used by the device (3)** – This structure contains the permissions and type of device, including the devices owner and group ID [QNX software systems, Programmers Guide, 2002].

- **Put a name into the namespace (4)** – The resource manager needs to inform other programs, via the process manager, which pathnames it is responsible for. This allows programs to find and connect to the resource manger so messages can be passed [QNX software systems, Programmers Guide, 2002].
- **Allocate the context structure (5)** – Creates a buffer for incoming and outgoing messages, according to the initialization of the attribute structure [QNX software systems, Programmers Guide, 2002].
- **Start the resource manager message loop** – The message loop must wait for incoming messages from programs and then provide the required service, if possible. Two possible types of message loops could be used, as shown in Listing 5.2 and 5.3.

```
while(1) {                              //start the resource manager message loop
       ctp = dispatch_block (ctp);      //block (1) - waiting for a event
       dispatch_handler (ctp);          //handle event (2)
}
```

Listing 5.2: Message loop for single-threaded device resource manager [QNX software systems, Programmers Guide, 2002]

The loop, in the above listing, will block (1), waiting for a message from a program requiring a connection or I/O function. When this message is received, the dispatch_handler is called, which will in turn execute the required handler function. Only one message can be handled at a time, using this approach.

Listing 5.3 contains the listing, where a thread pool is used to handle multiple messages simultaneously. This will prevent clients, waiting for messages to be accepted by the server, while it is busy processing another message.

```
memset (&pool_attr, 0, sizeof pool_attr);    //initialize thread pool attributes
pool_attr.handle = dpp;
pool_attr.context_alloc = dispatch_context_alloc;
pool_attr.block_func = dispatch_block;
pool_attr.handle_func = dispatch_handler;
pool_attr.context_free = dispatch_context_free;
pool_attr.lo_water = 2;
pool_attr.hi_water = 4;
pool_attr.increment = 1;
pool_attr.maximum = 50;

tpp = thread_pool_create (     &pool_attr,            //allocate a thread pool handle
                               POOL_FLAG_EXIT_SELF);

thread_pool_start (tpp);               //start the threads (will not return)
```

Listing 5.3: Thread pool for multi-threaded device resource manager [QNX software systems, Programmers Guide, 2002]

Listing 5.4 shows the basic source code used to implement a message handler used to handle data written to the resource manager by a client.

```
int my_io_write (resmgr_context_t *ctp, io_write_t *msg, RESMGR_OCB_T *ocb)
{
        status = iofunc_write_verify (ctp, msg       //ensure client has permissions
                                      , ocb, NULL);  //to write to resource

        IO_SET_WRITE_NBYTES (ctp, msg->i.nbytes);    //set number of bytes returned
                                                     //by client's write

        buf = (char *) malloc (msg->i.nbytes);       //allocate a buffer
        resmgr_msgread (ctp, buf->i.nbytes, sizeof (msg->i));       //re-read the
                                                                    //message

         - - - Process the message here - - -

        return (E_OK);                               //return success
}
```

Listing 5.4: Basic client write request handler skeleton [QNX software systems, Programmers Guide, 2002]

All the resource managers used for the FSW software, following the structures shown in listing 5.4, 5.5 and 5.6 to provide interfaces between software modules. Only the modules function (processing) varies from one module to the next.

```
int my_io_read (resmgr_context_t *ctp, io_read_t *msg, RESMGR_OCB_T *ocb)
{
        status = iofunc_read_verify (ctp, msg //ensure client has permissions
                                      , ocb, NULL);  //to read resource


        - - Construct the data to be sent to the client - -


        SETIOV (ctp->iov, buffer + ocb->offset, nbytes);     //set up the return
                                                             //data structure
        _IO_SET_READ_NBYTES (ctp, nbytes)            //set number of bytes to return

        ocb->offset += nbytes; //advance offset by number of bytes returned
```

```
        return (_RESMGR_NPARTS (nparts));             //return the data to the client
}
```

Listing 5.5: Basic client read request handler skeleton [QNX software systems, Programmers Guide, 2002]

Listing 5.5 and 5.6 show the basic source code used to implement a message handler used to handle read data requests sent by a client.

```
int my_io_devctl (resmgr_context_t *ctp, io_devctl_t *msg, RESMGR_OCB_T *ocb)
{
        status = iofunc_devctl_default (ctp, msg, ocb);      //let system handle
                                                             //unsupported commands
        rx_data = DEVCTLDATA(msg->i); //read in data from client

        switch (msg->i.dcmd) {        //list of commands handled by this RM
        case MY_DEVCTL_SETVAL:
                global_integer = rx_data->data32;     //set the variables value
                nbytes = 0;                           //return no bytes
                break;

        case MY_DEVCTL_GETVAL:
                rx_data->data32 = global_integer;     //read return data
                nbytes = sizeof (rxdata->data32);     //size of returned data
                break;

        default:
                return (ENOSYS);                      //unknown command
        }
        memset (&msg->o, 0, sizeof (msg->o));  //clear start of return message
        msg->o.ret_val = status;                      //specify return value
        msg->o.nbytes = nbytes;                       //number of bytes to return

        return (_RESMGR_PTR (ctp, &msg->o, sizeof (msg->o) + nbytes) );
}
```

Listing 5.6: Basic client devctl request handler skeleton [QNX software systems, Programmers Guide, 2002]

2.14 Software Layer 1: Hardware Interface

This layer is responsible for the low-level interface between higher-level modules and the physical hardware of the computer. The computer contains four relevant hardware devices used to communicate with the FSW machine. These include the RS232, RS485, digital I/O and LAN interfaces. If the hardware architecture is changed, then only this layer will need to be modified.

Three resource managers (modules) are used to provide the hardware interface to the real world:

- Serial interface driver (to motor inverters)
- Digital I/O driver (to limit switches and contactors)
- LAN interface driver (to distributed user interface)

2.14.1 Devc-ser8250 Serial Interface Driver

Serial communication channels are managed by the devc-ser* family of driver processes. These drivers can manage more than one physical channel and can map character devices to the pathname space, with names such as /dev/ser1, /dev/ser2 [QNX software systems, Architecture Guide, 2002].

When devc-ser* is started, command-line arguments can specify which, and how many serial ports are installed. The devc-ser* driver directly supports most nonintelligent multiport serial cards [QNX software systems, Architecture Guide, 2002].

The stty command is used to configure the serial ports baud rate, stop bits, data bits and parity. The following configuration was used for the FSW system:

Chuck Interface Unit 1	stty baud=28800 data=8 stop=1 parity=n < /dev/ser1
Chuck Interface Unit 2	stty baud=28800 data=8 stop=1 parity=n < /dev/ser2
Spindle Inverter	stty baud=57600 data=8 stop=1 parity=n < /dev/ser3
Feed Inverter	stty baud=57600 data=8 stop=1 parity=n < /dev/ser4

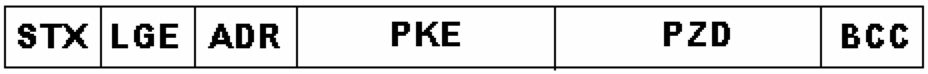
Table 5.1: FSW system devc-ser8250 serial driver configuration

Serial communication hardware generally contains a UART. This device changes incoming parallel information to serial data, which can be sent and received over a communication medium. A UART performs all the tasks, timing, parity checking, etc needed for communication. These devices contain registers, which the software drivers can access in order to configure the communication parameters (baudrate, type of parity check and signaling). A UART removes processor overhead by handing the transmission and reception instead of the CPU. When the task is complete the CPU is signaled using an IRQ.

2.14.2 Dev-inv Inverter Siemens MM440 Driver

This resource manager forms an extension to the low-level serial resource manager.

The Siemens Micromaster 440 inverters use the Siemens USS protocol to provide remote monitoring and control of the inverter. The protocol defines the access technique according to the master-slave principle for communications via a serial bus. One master and a maximum of 31 slaves can be connected to the bus. The master via an address character in the telegram selects the individual slaves. A slave can never transmit without first being requested to do so, and direct message transfer between the individual slaves is not possible [Siemens, Using the USS Protocol, 2002]. This forms part of the RS485 standard, which the inverters use for control and monitoring.



Where: **STX** is the delimiter used to signal the start of a telegram. Its value is always 0x02 hex.

LGE is the length of the telegram in bytes.

ADR is the target address or source address of another node on the network.

PKE contains information used to read or modify inverter parameters.

PZD is used to set the motors set point.

BCC is a checksum, used by the receiver to verify the telegram was transferred unaltered from the sender.

Figure 5.3: A USS telegram as implemented for the FSW machine [Siemens, Using the USS Protocol, 2002]

Figure 5.3 shows the structure of the USS telegram that is transferred between the inverter and remote system. The USS protocol allows variable PKE and PZD field lengths to allow the protocol to be customized for the application.

For the FSW machine, a fixed four word PKE and a fixed two word PZD field length were chosen in order to allow rapid construction and post-reception processing of the packet and ensure access to the required functionality would be able to be obtained. A C++ class was written to handle the construction and analysis of incoming and outgoing telegrams. This allows the details of the protocol to be “hidden” to higher-level software and also promote software reuse. The Siemens MM440 inverter resource manager’s architecture is shown in Figure 5.4.

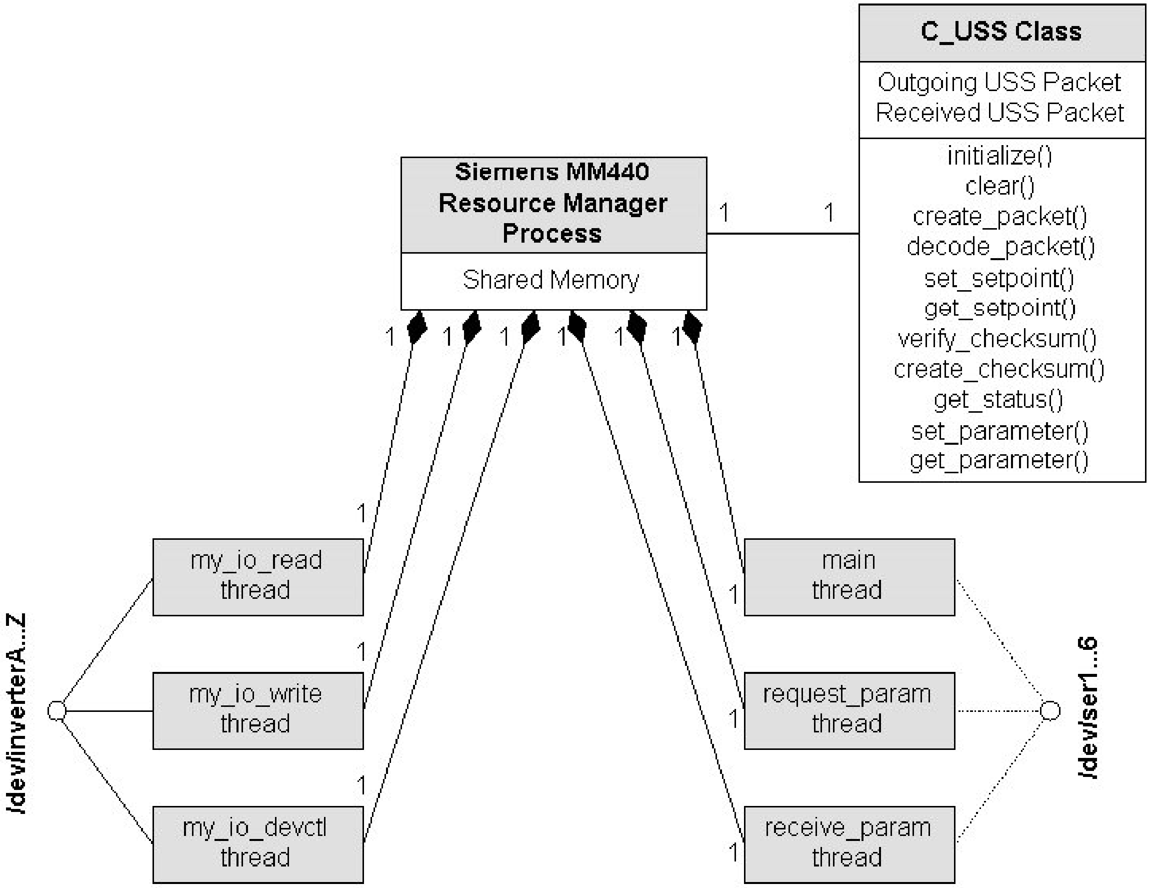


Figure 5.4: UML static object diagram of the MM440 inverter resource manager

Using the USS protocol, the resource manager is responsible for the following functions:

- Starting/stopping the motor (my_io_devctl thread)
- Setting the motor's speed (specified in Hz, where 50Hz is the speed rated on the nameplate – my_io_write thread)
- Maintaining a client-specified list of parameters, which must continually be read to ensure the list, is current (request/receive_param threads).
- Acknowledging inverter warnings and faults (my_io_devctl thread)
- Returning the inverter information when it is requested by a client (my_io_read thread)
- Modifying the list, when a client requests another set of parameters to monitor (my_io_write thread)

Array Index	**Parameter Number**	**Parameter Value**
1	35	48
2	45	112
3	1198	0

Table 5.2: Example parameter list

Table 5.2 shows an example of the table stored in the resource manager's shared memory area. It is in the form of an array, where the user can specify a list of parameter numbers and the request-receive thread pair is responsible for filling in the parameters value field.

Two wire RS485 is used to communicate with the inverters, which only allows half-duplex communication. Therefore, when a command is sent or parameter is requested, the software must wait for a response from the inverter before transmitting again. Figure 5.5 shows this timing between the two threads responsible for ensuring half-duplex communication.

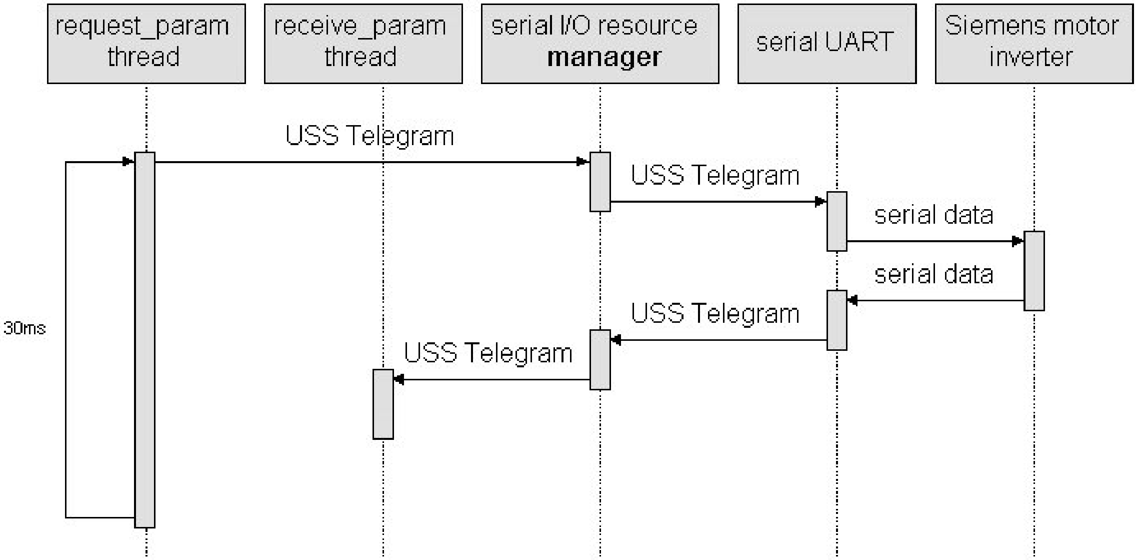


Figure 5.5: Timing of parameter update procedure

After conducing testing, is was found that the inverter will only process another telegram after a certain delay, resulting in a 28ms round-trip time per telegram. Therefore a 30ms delay was used, which also allows a predictable response time of the system to be established. This method may not produce optimum communication efficiency, but for this application a known “sampling” rate was preferred.

By sending write requests to the resource manager, the parameters to be monitored from the associated inverter can be specified. An ASCII space delimited string with the following format can be used:

$$\mathbf{P_{n1}\ P_{i1}\ P_{n2}\ P_{i2}\ P_{n3}\ P_{i3}\ \ldots\ P_{nx}\ P_{ix}}$$

Where: $\mathbf{P_n}$ is the number of the parameter between 0 and 4000.

$\mathbf{P_i}$ is the parameters index between 0 and 255.

Issuing a read request after the above-mentioned procedure, will yield the required values in the following format:

$$\mathbf{P_{n1}\ P_{v1}\ P_{n2}\ P_{v2}\ P_{n3}\ P_{v3}\ \ldots\ P_{nx}\ P_{vx}}$$

Where: $\mathbf{P_v}$ is the value read from the inverter

The devctl handler is used to perform the following functions:

- Setting the inverter’s frequency set point
- Starting and Stopping the motor associated with the inverter
- Acknowledging faults and warnings

2.14.3 Dev-dio Digital IO Interface Driver

This resource manager forms the link between the physical digital I/O PCI card and the higher-level software modules. The PCI interface card contains software addressable registers that provide access to the physical hardware input and outputs.

The purpose of the digital I/O resource manager is to:

- Provide an interface capable of turning on or off the FET output switches
- Allow higher-level software modules to read the state of the digital inputs
- Monitor the generation of asynchronous changes of the digital input states and inform higher-level modules of these changes

The Figure 5.6 shows the structure of this resource manager. A C++ class has been developed to handle the low level communication to the PCI hardware. This allows fast, easy modification to the resource manager should a different digital I/O card need to be implemented in the future.

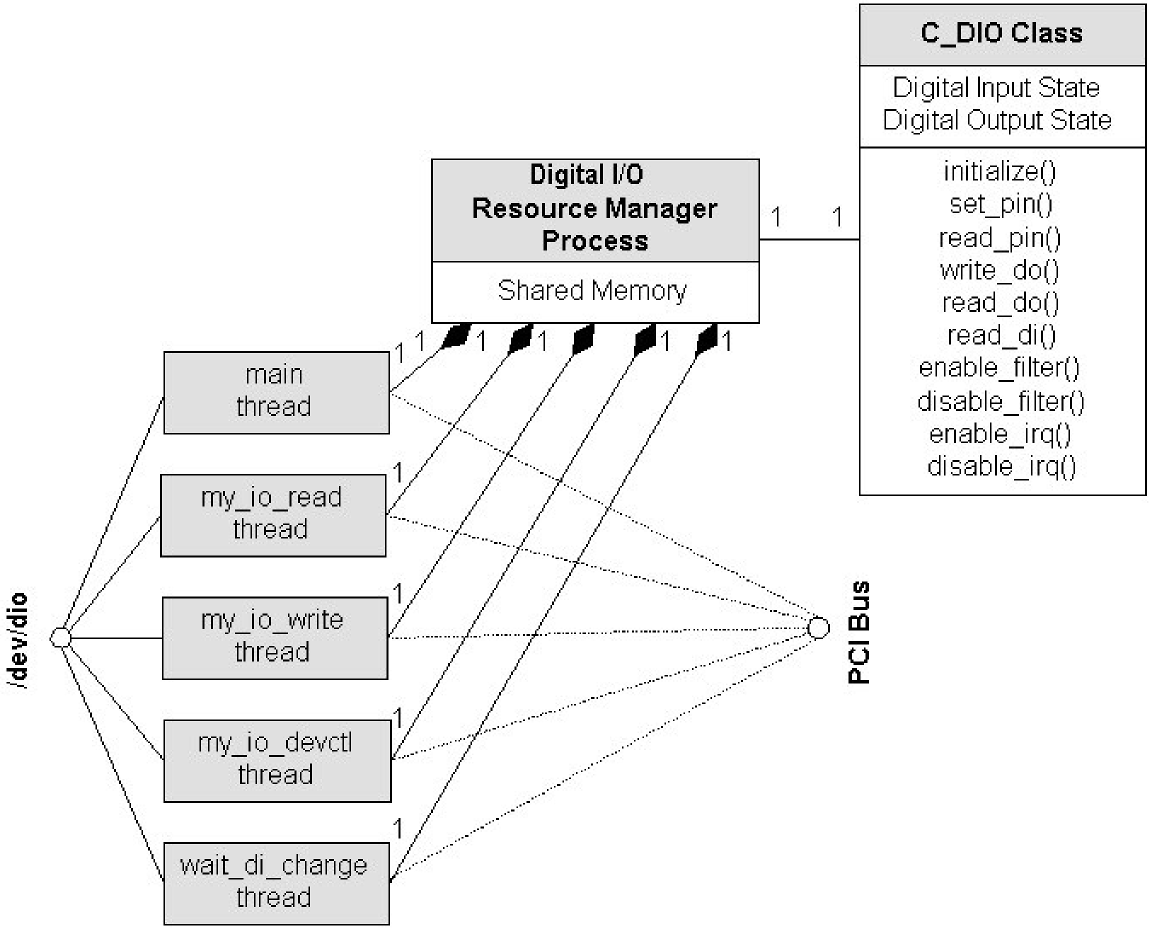


Figure 5.6: UML static object diagram of the AccessIO digital I/O resource manager

The I/O card has eight one byte wide registers that are implemented as follows:

I/O Address	**Read Register**	**Write Register**
Base + 0	FET Drive Outputs 0-7	FET Drive Outputs 0-7
Base + 1	Isolated Inputs 0-7	Clear Interrupt
Base + 2	Enable IRQ	Disable IRQ
Base + 3	Activate Filter	Deactivate Filter
Base + 4	FET Drive Outputs 8-15	FET Drive Outputs 8-15
Base + 5	Isolated Inputs 8-15	Unused
Base + 6	Interrupt Status	Unused
Base + 7	Filter Status	Unused

Table 5.3: PCI-IDIO 16 software accessible registers

The 5ms filter can be used to slow the response time of the inputs to eliminate noise spikes on DC inputs in industrial environments. The card's IRQ makes it unnecessary to poll inputs to detect state changes and its interrupt level is assigned by the BIOS or plug-and-play operating system [Access I/O, EDIO-16 PCI Board User Manual, 2002]. This reduces the processor overhead associated with polling.

Listing 5.7 shows the detection of the DIO card on the PCI bus and its initialization.

By invoking the devctl handler, the clients may add themselves to a list of clients to notify should an IRQ be generated from the change of an input. This notification is done using pulses, discussed in Chapter 4. The read and write handlers are used to directly access the inputs or outputs of the card.

```
//****************************************************************************************//
//** Description: Constructor to detect ACCESS DIO card, initialize and            **//
//**                                         find its base addr & IRQ              **//
//** Arguments: None                                                               **//
//** Returns: Nothing                                                              **//
//***************************************************************************************//
C_DIO::C_DIO (void) {

        ThreadCtl(_NTO_TCTL_IO, 0);   //attempt to obtain I/O permissions for thread
```

```
        memset(&DIO_Info, 0, sizeof (DIO_Info));     //Clear DIO card info structures
        memset(&info, 0, sizeof (info));

        if (pci_attach(0) < 0) {
               perror("PCI_attach error!");
               DIO_Info.Card_Exists = FALSE;
        }

        // Fill in the Vendor and Device ID for ACCESS DIO PCI Card
        info.VendorId = ACCESSIO_VENDOR_ID;          //ACCESS IO
        info.DeviceId = ACCESSIO_DEVICE_ID;          //DIO CARD

        if ((hdl = pci_attach_device(0, PCI_SHARE|PCI_INIT_ALL, 0, &info)) == 0) {
                perror("PCI_attach_device error!");
               DIO_Info.Card_Exists = FALSE;
        }

        //find the DIO cards I/O base address
        iobase = (mmap_device_io(info.BaseAddressSize[2], info.CpuBaseAddress[2])-1);

        out8 ( iobase+DIO_FET_DRIVE_OP_0_7,  0 );           //clear digital outputs (0-7)
        out8 ( iobase+DIO_FET_DRIVE_OP_8_15,  0 );   //clear digital outputs (0-8)

        DIO_Info.Card_Exists = TRUE;
}
```

Listing 5.7: DIO PCI board detection and initialization

2.15 Software Layer 2: Machine Level Control and Monitoring

This software layer is responsible for providing basic machine functionality. This includes positioning of the bed, control of the spindle and the recording of data from the motor inverters and chuck monitoring system. This layer provides an abstraction from the low-level control provided by the hardware drivers.

2.15.1 Dev-control: Digital Input Monitoring, Clutch, Brake and Contactor Control

This resource manager is responsible for the following functions:

- Extending the functionality of the digital I/O resource manager
- Interlocking of the feed axes selection clutches and feed axes brakes
- Detection of faults with the limit switches & other digital inputs
- Engaging the selected feed axis and feed-type clutches and axes brakes
- Software selection of inverter power state (on/off)
- Limit switch, PSU, main power switch and ESTOP status monitoring
- Informing higher-level modules of changes in the state of the digital inputs

The UML diagram below shows the structure of this resource manager.

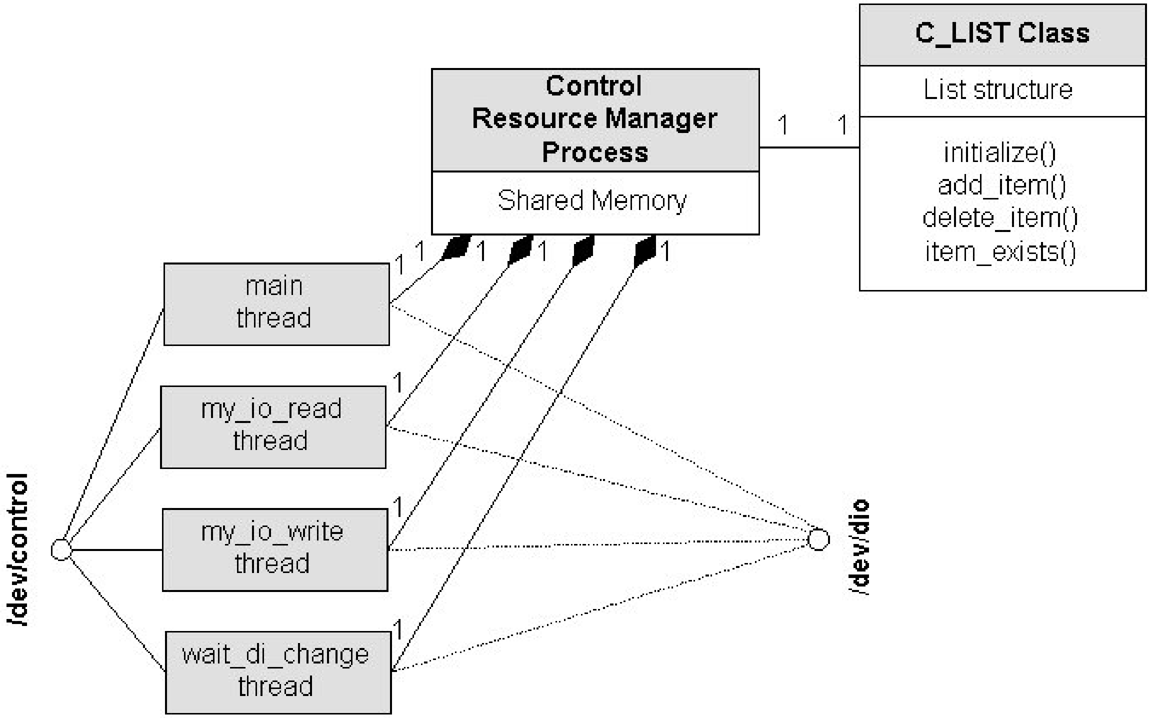


Figure 5.7: UML static object diagram of control resource manager

By checking for certain impossible conditions, such as the depression of two limits switches on the same axis simultaneously, basic fault checking has been implemented.

This resource manager can inform higher-level modules of changes in the state of the digital inputs. A simple list class has been developed to facilitate this. Clients of this resource manager, can enter their PID and COID numbers into the list, so the resource manager can send a pulse informing them of any change. A maximum of up to 10 clients can be entered into the list. The wait_di_change thread, shown is Figure 5.7, waits for a message from the digital I/O resource manager. Once this message is received, a pulse is sent to all the clients in the list.

This resource manager also converts the binary values returned from the digital I/O resource manager into a more easily readable and intuitive format. For example:

When a read request is issued the resource manager returns the following:

P1S1E0FNDXL000

Where:

P = inverter power: **1** = on / yes
0 = off / no

S = +24V power supply output

E = ESTOP pressed

F = feed type: **N** = normal feed-type

R = rapid feed-type

D = feed axis type: **X** = x-axis

Y = y-axis

Z = z-axis

L = limit switches: **000** represents the x, y and z feed axes states:

s = minimum soft-stop position

S = maximum soft-stop position

H = hard-stop position (user intervention required)

! = limit switch fault

0 = not triggered

The above string can also be written to the resource manager to effect changes to the feed-type, feed-axis and inverter power state. This string can be contrasted to the digital I/O resource manager's output, which is just two hexadecimal integers, proving very little other information.

2.15.2 Dev-daq: Data Acquisition System Monitoring

The following three resource managers discussed contain a “capture” thread. This thread handles the in process recording of machine parameters. For the dev-daq resource manager, the “capture” thread records the raw data received from the chuck at specified periods to the RAM file system (/dev/shmem). This data is then processed and displayed after the weld. For record keeping purposes, this data can manually be copied to a permanent storage medium.

This resource manager is responsible for continually reading the data from the instrumented chuck, ensuring up to date data is available for higher-level client requests. Figure 5.8 shows the internal structure of the dev-daq resource manager. By reading from the resource manager a space delimited string is returned to the client, containing the current data received from the chuck. There is no write handler as the communications to the chuck are only unidirectional. The devctl handler is used to start, stop, reset and set the sampling rate of the data recording process. For each chuck interface unit attached to the system, a request-receive pair of threads is created and is used to transfer the data from the interface unit to the computer.

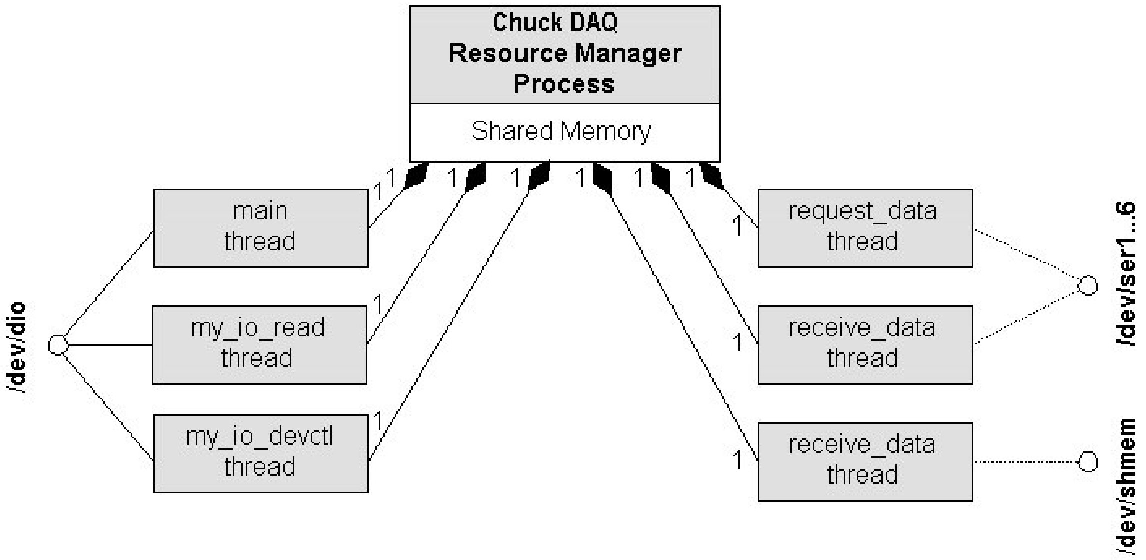
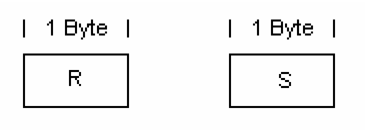


Figure 5.8: UML static object diagram of chuck DAQ resource manager

The chuck interface units operate at a baud rate of 28800bps. The communications operate over RS232, which is full duplex and uses a basic request-respond protocol. The client sends a request character, as shown in Figure 5.9a and b, and the interface unit will respond by sending a response as shown in Figure 5.10. Each interface unit has two channels and the data for each channel must be requested separately. By sending an “**R**” the data from channel 1 is requested and by sending an “**S**” the data for channel 2 is requested. This allows channel to have different sample rates.

The marker field in the response data is only valid from the interface unit attached to the inductive proximity sensor. When the sensor is triggered, the marker field contains the ASCII character “1”. The next time a request for data is sent to the unit, the marker field is automatically reset and will contain a “0”.



(a) (b)

Figure 5.9: Data to send to the interface unit

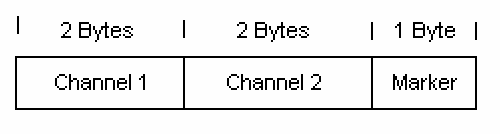


Figure 5.10: Data received from interface unit

At a baud rate of 28800bps each bit transmitted takes 34.722us until is it received. If only one channel is sampled a total of six bytes need to be transferred (one request and five response bytes). This results in sixty bits being exchanged, consuming 2.083ms.

While the tool is rotating, the horizontal strain gauge is repeatedly sampled, so a strain pattern around the tool can be obtained. The strain measured by the strain gauge has been signal conditioned by the interface unit, so as to provide the actual force exerted on the pin of the FSW tool.

The sample rate for the horizontal strain gauge of 5ms has been selected; this allows sufficient time to sample the other channel between force samples.

A typical spindle speed used during the FSW tests was 500 RPM. At this speed, the tool takes 120ms to complete one revolution. This results in a maximum of 24 samples per revolution for that spindle speed. The number of samples per revolution is inversely proportional to the spindle speed (S_s), as shown in formula 5.1.

$$N_s \propto \frac{1}{S_s} \qquad (5.1)$$

The graph in Figure 5.11 illustrates this relationship graphically.

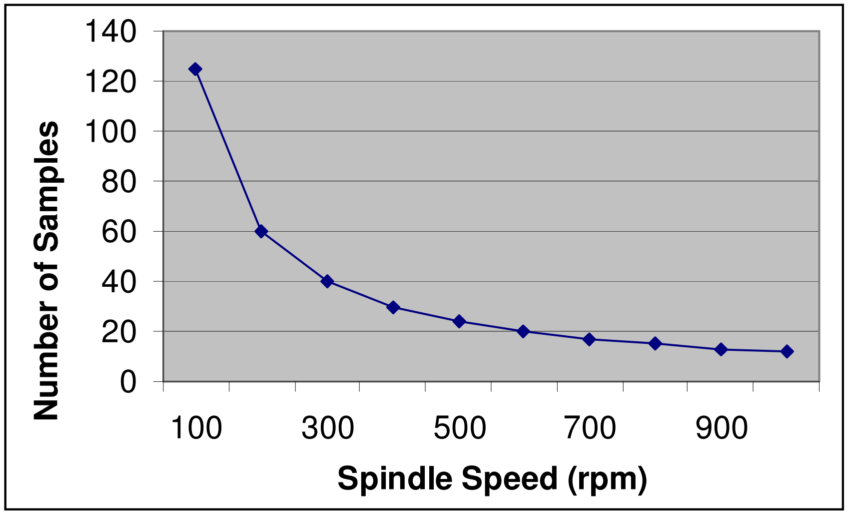


Figure 5.11: Relationship between number of samples possible per revolution and spindle speed

The number of samples mentioned above is the maximum possible, however in practice this number is usually a few counts lower. The following reasons have been found to influence this value:

- Variations in spindle speed during welding
- Varying communication response times from the LMI interface unit

The receive_param thread tags each sample received with a 64bit value, obtained from a hardware counter, and places this pair into an array. The counter is located on the computer's motherboard and is used to count the number of CPU instructions executed. Once the marker is set to 1, indicating a completed revolution, the array of magnitude-count pairs is evaluated. At the beginning (C_b) and end (C_e) of each revolution the counter is read, to obtain references. So that:

$$Angle/Count(AC) = \frac{360^o}{C_e - C_b} \quad (5.2)$$

The above formula is used to obtain the relationship between counts and degrees. The value (AC) is obtained and then multiplied by the difference between the beginning count and the sample's count, which was recorded. Using the following formula:

$$\mathrm{A_i} = (\mathrm{C_i} - C_b) \times AC \quad (5.3)$$

Where: **i** is the sample number for which the angle is required and can be any integer between 1 and the maximum number of samples according to formula 5.1.

The variable A_i contains the angle of the sample number i. By processing the array for each revolution an array can be obtained, containing the resultant force vectors present on the FSW tools pin.

Due to the variation in the number of samples obtained per revolution, a threshold feature has been added to the software. This feature predicts the maximum number of samples that should be obtained for a particular spindle speed. The actual spindle speed is obtained from the inductive proximity sensor via the interface unit. If the number of actual samples is lower than a certain threshold below the predicted value, the horizontal force data for that revolution is discarded and the data for the previous revolution is used. The horizontal force changes during a FSW weld are rather slow, therefore a short delay is not critical to the monitoring of the process.

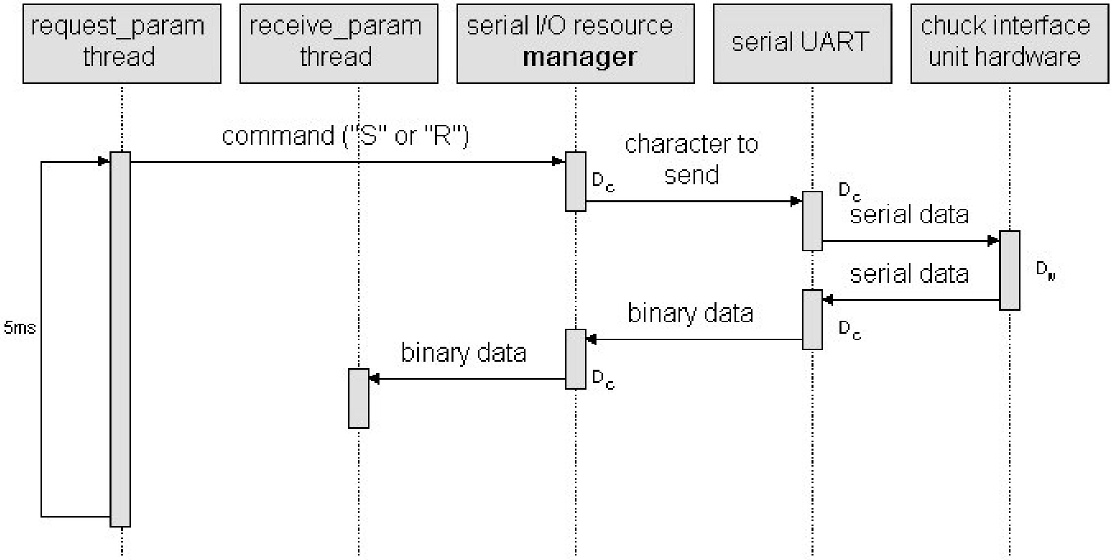


Figure 5.12: UML sequence diagram of thread controlled communication

The UML diagram in Figure 5.12 shows the timing of the request-receive thread pairs used to obtain data from the instrumented chuck.

The interface unit is calibrated in kilograms; therefore this resource manager must convert the unit to Newtons, which is a standard when specifying forces.

When the client issues a read request to the resource manager, a human readable space delimited ASCII string is returned. This string contains all the information measured from the chuck, in the format as shown below:

$$\mathbf{F_x\ \ F_z\ \ T_t\ \ T_g\ \ T\ \ S_s\ \ N\ \ A_1\ \ M_1\ \ A_2\ \ M_2\ \ A_3\ \ M_3 \ldots A_n\ \ M_n}$$

Where: $\mathbf{F_x}$ is the instantaneous horizontal force applied to the pin of the tool perpendicular to the strain gauge (Newtons)

$\mathbf{F_z}$ is the vertical force present on the tool (kilo-Newtons)

$\mathbf{T_t}$ is the temperature of the tool’s pin (degrees Celsius)

T_g is the temperature of the electronics mounted on the chuck (degrees Celsius)

S_s is the current spindle speed (RPM)

N is the number of F_x values obtained for the last revolution

A_n is the nth sample's angle of the variable F_x (degrees)

M_n is the nth sample's magnitude of the variable F_x (Newtons)

2.15.3 Dev-spindle: Spindle Control and Monitoring

The inverter resource manager deals with the monitoring and control of the Siemens motor inverter and its associated communications. The spindle resource manager extends this interface and provides control of the machines spindle, using the functionality provided by the inverter resource manager. This resource manager provides for the following:

- Setting the spindle speed by specifying an RPM value, instead of a frequency
- Starting and stopping the spindle
- Informing the inverter resource manager of which of the inverters parameters, relevant to the spindle, should be monitored

- Starting, stopping and customizing the recording of the parameters to the RAM file system (sample rate, record path, filename, etc)
- Recording the inverters parameters at periodic intervals to the RAM file system
- Sending higher-level clients the information obtained from the inverter

The spindle resource manager’s internal structure is shown in Figure 5.13. The monitor thread periodically submits a read request to the inverter process manager, which responds with the most recent list of parameter values. These are stored in the processes shared memory area as a structure. This structure is in turn read periodically at the user specified sample rate and written to the RAM file system by the capture thread.

On initialization of this resource manager, it informs the inverter process manager which parameters should by continually monitored. The spindle resource manager monitors the parameters listed in Table 5.4.

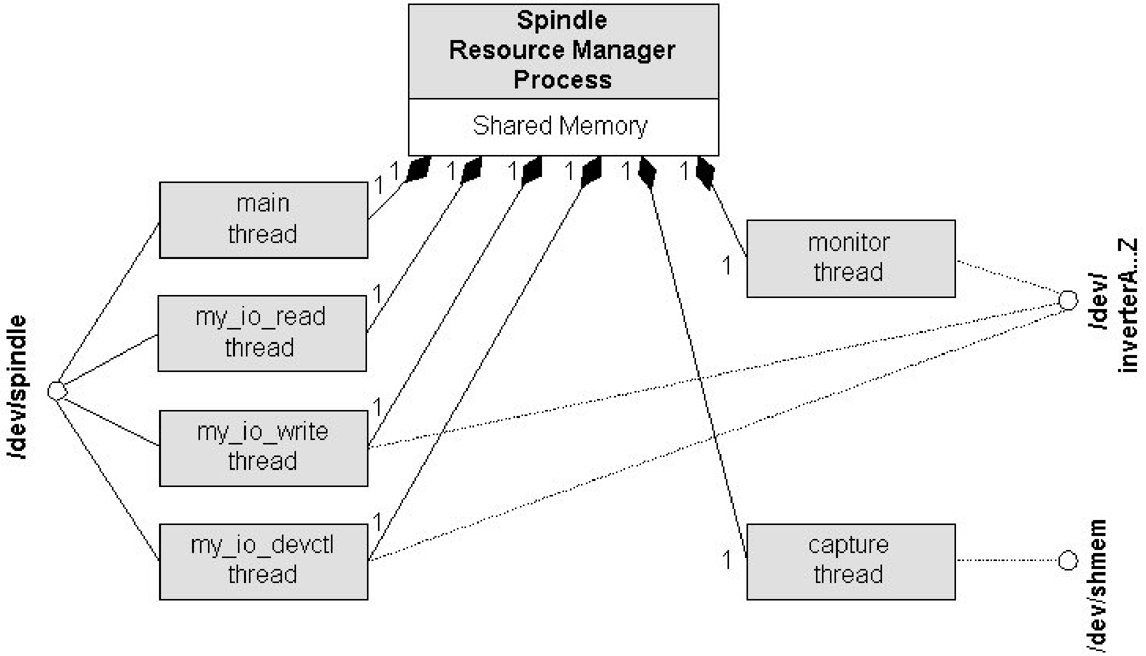


Figure 5.13: UML static object diagram of spindle resource manager

Parameter Description	**Number**
Inverter status word (running, stopped, fault, warning, etc)	r0052
Actual motor frequency (which is converted to RPM)	r0061

RMS motor current	r0068
RMS motor voltage (used with current to calculate RMS motor power)	r0025
Stator windings temperature (for safety and fault finding)	r0035
Inverter electronics temperature (for safety and fault finding)	r0037
Spindle motor torque	r0080

Table 5.4: Spindle motor inverter parameters monitored

** (When parameter numbers are discussed, assume parameters indices are 0, unless specified)*

The time required by the inverter resource manager to obtain one parameter from the list is 30ms. Therefore every 210ms all the parameters in the list will be refreshed. The resource manager's monitor thread is responsible for this update process. Should a higher response time be needed, fewer parameters must be monitored. If higher-level modules read from this resource manager it can supply the most recent information in the lowest possible time. This is because of the mirrored data from the inverter resource manager being stored in this resource manager. An advantage of using this method is that the sample rate for the capture thread can have a higher interval, without affecting this dataflow. Also, using a separate thread for recording is that should the media fail (out of RAM, disk full, etc) the systems dataflow is not compromised.

A read request from the spindle resource manager, will return a space delimited human readable string in the following format:

$$\mathbf{S \quad St \quad T_s \quad S_s \quad T \quad T_m \quad T_i \quad V_{rms} \quad I_{rms}}$$

Where: **S** is the status of the inverter. Possible values include F (fault), W (warning) and N (normal operation).

St is a 16bit hexadecimal value providing inverter condition information (running, stopped, direction, faults, brake, etc)

Ts is the number of milliseconds elapsed since the start of recording.

Ss is the spindle speed in RPM.

Tm is the temperature of the spindle motor, in degrees Celsius.

Ti is the temperature of the inverters electronics, in degrees Celsius.

VRMS is the RMS voltage applied to the spindle motor.

IRMS is the RPM current applied to the spindle motor.

By sending a write request to the resource manager, the spindle speed can be set. The physical gearbox setting on the machine must be entered, to allow the conversion between motor frequency and RPM to be obtained. The written message should be in the following format:

<Gearbox_setting> <Required_motor_speed>

The devctl interface is used to acknowledge faults, start or stop the spindle and configure data recording. It is also capable of switching to an alternate set of parameters to monitor. This alternate set can be used for troubleshooting. Table 5.5 lists the parameters that are monitored in this mode.

Parameter Description	**Number**
Inverter control word 1	r0054
Inverter status word 1	r0056
Encoder status	r0403
Inverter monitoring word 1	r2197
Inverter monitoring word 2	r2198
Last fault code 1	r0947
Last fault code 2 (index 1)	r0947
Warning number 1	r2110
Warning number 2 (index 1)	r2110

Table 5.5: Spindle motor inverter alternate parameters monitored

2.15.4 Dev-feed: Bed Feed Control and Monitoring

The internal structure of the resource manager responsible for the movement and positioning of the bed is shown in Figure 5.14.

The function and implementation of the monitor, capture, read, write and devctl threads are almost identical. The only difference is the parameters that are monitored. The monitored parameters for this resource manager are listed in Table 5.6.

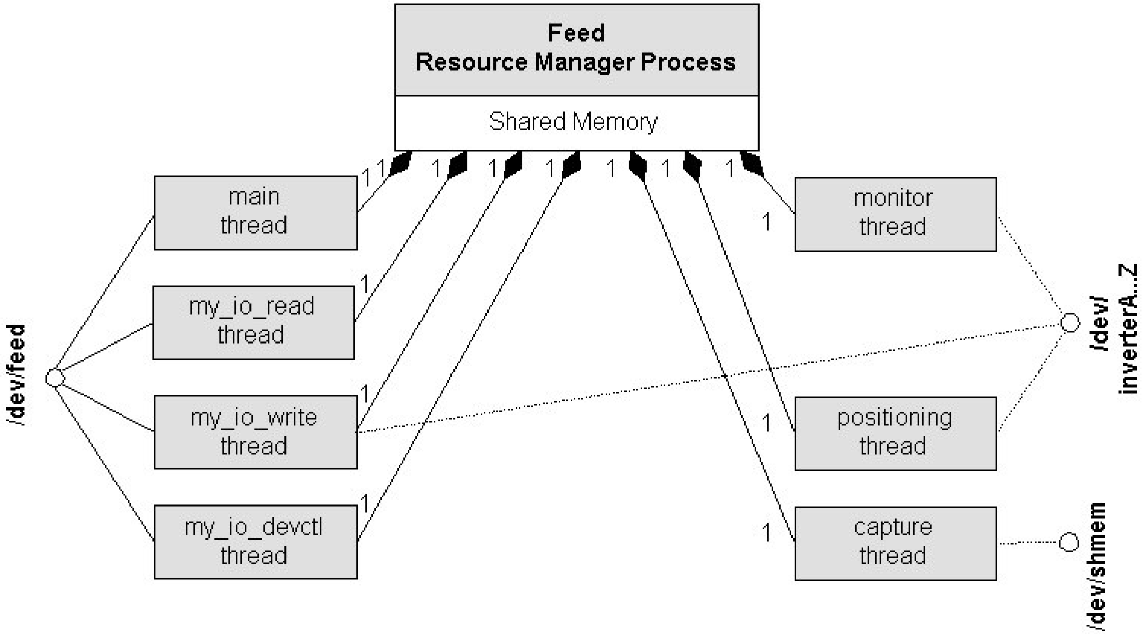


Figure 5.14: UML static object diagram of feed resource manager

Parameter Description	**Number**
Inverter status word (running, stopped, fault, warning, etc)	r0052
Actual motor frequency (which is converted to RPM)	r0061
RMS motor current	r0068
RMS motor voltage (used with current to calculate RMS motor power)	r0025
Stator windings temperature (for safety and fault finding)	r0035
Inverter electronics temperature (for safety and fault finding)	r0037
Spindle motor torque	r0080
Number of motor shaft revolutions since last stop command	r2489

Table 5.6: Feed motor inverter parameters monitored

** (This resource manager uses the same alternate parameters to monitor as the spindle resource manager as shown in Table 5.5)*

The wait_limitsw thread, simply waits (in the blocked state) for a message from the control resource manager. When the feed resource manager is initialized, it requests the control resource manager to inform it of any changes in digital input condition. In the event of a change, the control resource manager will send a pulse to the feed resource manager, resulting in the unblocking of the wait_limitsw thread. The thread then sets a software flag, which is monitored by the position thread.

The position thread handles the sequencing of the movement and positioning of the bed. Two positioning modes have been implemented:

- Move the indicated axis to its home position
- Move the indicated axis a certain distance from its current position

Before positioning can commence, the client must specify the feed parameters that describe the type of movement required. This is done by writing a space-delimited string to this resource manager. The format of this string is as follows:

<gear_selection> <feedrate> <feed_axis> <distance>

Where: **feedrate** is measured in mm/min and can be any integer below the selected feedrate on the gearbox. The sign of this integer will determine the direction of traverse.

feed_axis can be either X, Y or Z.

distance is measured in millimeters to move from current position. A distance of 9999 will result in the homing of the selected axis.

After the feed parameters have been specified, the start feed devctl option can be used to execute the desired feed.

The following sequence of steps is executed that will result in the bed being moved to its home position:

- Rapid feed with the maximum negative feedrate
- Wait for the home limit switch to be depressed
- Reduce the feedrate to a very low value
- Wait for the home limit switch to be released
- Rapidly ramp down the feed until the bed is stopped
- Normal feed with the maximum possible positive feedrate, until the home limit switch is depressed

- Move the bed just less than the length of the limit switch at the same feedrate
- Use the slowest feedrate possible to continue the traverse
- Wait for the limit switch to be released
- Rapidly stop the feed

This sequence will result in the rapid homing of the specified axis. When the bed rapid feeds, instantaneous deceleration is not possible and could damage the machine. Therefore once the limit switch is depressed, the bed is decelerated gradually to a stop. This, in most cases, results in the limit switch being overrun and therefore the slower feed in the opposite direction to find the exact starting point.

When moving a specified distance, no limit switches are used and accuracy of the distance moved becomes important. The inverter controls the ramp-up and speed control of the motor at the start and during the traverse. The resource manager controls the stopping. Because the process is a welding process, extreme accuracy is not vital. Millimeter accuracy is sufficient, except when plunging the tool into the material. Incremental optical encoders, connected to the inverters, are used for positioning. The encoder count is accessed via a parameter on the inverter, which must be sampled by the computer system, as discussed in Section 5.1.2. This makes positioning accuracy a function of sample time. This relationship can be described using the derived formula below:

$$r_{ff} = \frac{T_s . a . f_f}{60} \qquad (5.4)$$

Where: **f_f** is the frequency of the supply to the motor as controlled by the inverter, just before the required position is reached.

T_s is the time between updates of the encoder count parameter, as discussed in Section 5.1.2.

a is the ratio of selected gearbox feedrate to the maximum frequency of the inverter.

The value for **a** can be calculated using the following formula:

$$a = \frac{G_f}{f_{\max}} \qquad (5.5)$$

Where: **G_f** is the feedrate as selected on the gearbox; and

$\mathbf{f_{max}}$ is the maximum frequency the inverter will allow, as set by parameter p1080.

This resource manager monitors eight parameters; therefore each parameter will be refreshed every 270ms. This will be the value for $T_{s.}$ Using formula 5.4, the positioning accuracy for a gearbox setting of 180mm/min and a positing seeking motor frequency of 15Hz, will be 0.24mm.

In order to traverse at the specified rate and bring the bed to a stop at the correct position, the resource manager must determine the point to begin decelerating. Figure 5.15 shows a velocity-time graph of the positioning process.

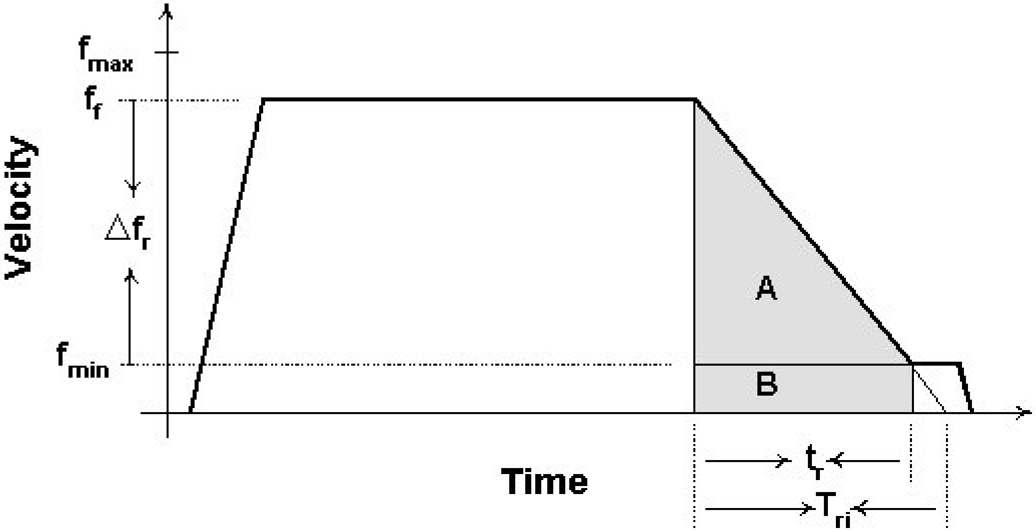


Figure 5.15: Velocity-time graph of the positioning process

With regard to the Figure 5.15, the inverter controls the complete process, until the shaded area is reached. This point must be determined, in order to ramp the bed's velocity down to an acceptable level. This level will ensure the bed can be stopped accurately when the correct position is reached. To determine the distance the bed will move during the ramp-down process, the integral of the velocity with respect to time is taken. This is the same as calculating the area under the graph. The hatched area on the graph, marked A and B, is calculated using formulas 5.6 and 5.7 respectively:

$$d_{sq} = t_r f_{\min} \frac{a}{60} \qquad (5.6)$$

F_{min} is the minimum frequency at which the motor can operate, without causing the feed to stick. This value will depend on many factors, the most prominent being the feed gearbox selection, which relates to available torque. The resource manager is capable of detecting this condition and gradually increasing the feed frequency, to a point where sticking does not occur. Formula 5.7 is used to calculate the combined area of regions A and B.

$$d_r = (\frac{1}{2} t_r . \frac{\Delta f_r}{60}) + d_{sq} \qquad (5.7)$$

Where: d_r is the distance at which the ramp down process must begin

T_r is the time interval required to ramp the velocity from f_f to f_{min}

Delta f_r is the change in feedrate from f_f to f_{min}, measured in mm/min. The relationship between feedrate and inverter feed frequency is shown in formula 5.8:

(5.8)

$$\Delta f_r = a\Delta f_f$$

With regard to Figure 5.15, T_{ri} can be set using an inverter parameter, which specifies the time taken to ramp the motor from f_{max} to zero. Its value will depend on the inertia and other characteristics of the load. This value was selected to be 2 seconds. After selecting T_{ri}, t_r can be calculated which is the time required to ramp the motor from its feed frequency (f_f) to f_{min}, using formula 5.9.

$$\mathrm{t}_r = \frac{T_{ri}}{f_{\max}}\Delta f_f \qquad (5.9)$$

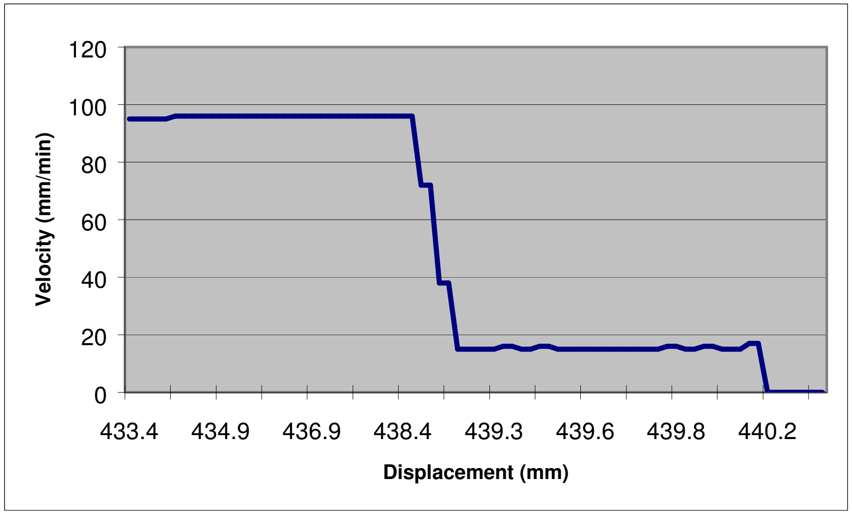


Figure 5.16: Actual positioning velocity-displacement graph obtained

Figure 5.16 shows the practical results of the positioning method discussed, as implemented for the feed resource manager.

Mechanical errors will cause deviations in the actual and expected positions. During development of the resource manager three types of mechanical error were encountered:

- Slippage of the feed clutches under severe loading conditions
- Backlash of gears
- Errors due to the machining error during the manufacture of the lead screw and related components

The last error was investigated, by doing a series of positioning tests. It was found that the positioning error increased linearly to the distance moved. This was compensated for using a simple straight-line equation. The only crucial stage of the welding process for positioning is during the plunging of the tool into the material. Due to the method used to plunge the tool this error remains constant. A constant can therefore be used to compensate for it.

The slippage however could not be compensated because of its intricate composition of factors influencing it. It is recommended that another more direct method be used to obtain the position of the bed, independent of the clutches. Currently slippage may result in errors as large as 0.2mm under heavily loaded situations.

Backlash was not compensated for, as it does not influence the welding process because of the sequence of movements used.

The sequence of steps used to move the bed a specified distance from its current position is as follows:

- Select axis to move and normal feedrate clutch.
- Set required feedrate set point and direction using inverter.
- Start the feed.
- Wait for the bed to move the distance d_r, using formula 5.7.
- Ramp the feedrate down to $f_{min.}$
- Continue the movement, until the correct distance has been reached.
- Immediately stop the feed.

A read request from the feed resource manager, will return a space delimited human readable string in the following format:

$$\mathbf{S\ \ St\ \ T_s\ \ S_s\ \ T\ \ T_m\ \ T_i\ \ V_{rms}\ \ I_{rms}\ \ P_X\ P_Y\ P_Z}$$

The format is the same as discussed for the spindle resource manager in Section 5.2.3, except that three more parameters have been added. P_x, P_y and P_z provide the position of the bed in relation to its home position. They are reset each time an axis reaches its home limit switch. Their values are derived using the gearbox ratio, feed motor RPM and encoder count monitored from the inverter.

2.16 Software Layer 3: Process Level Control and Monitoring

This section will discuss the convergence of all the functionality discussed to this point to form a common point for the co-ordination, control and monitoring of the machine.

The FSW sequencer and the network distributed user interface along with its data server will be discussed.

2.16.1 Sequence control system (/dev/fsw)

The following section will discuss the FSW sequence controller. This resource manager forms the core of the software, by forming a central module responsible for the control and monitoring of the FSW process. This resource manager relies on all the layers below to provide the functionality necessary to complete its task. Figure 5.17 shows the internal structure of this resource manager.

There are three distinguishing features for this resource manager, which are:

- The my_io_write handler accepts English-like commands, which are then used to set parameters or initiate machine actions. Table 5.7 provides some examples of this interface. Certain process parameters can be modified during the process (Spindle speed, feedrate, etc) allowing the process to be controlled by higher-level clients.

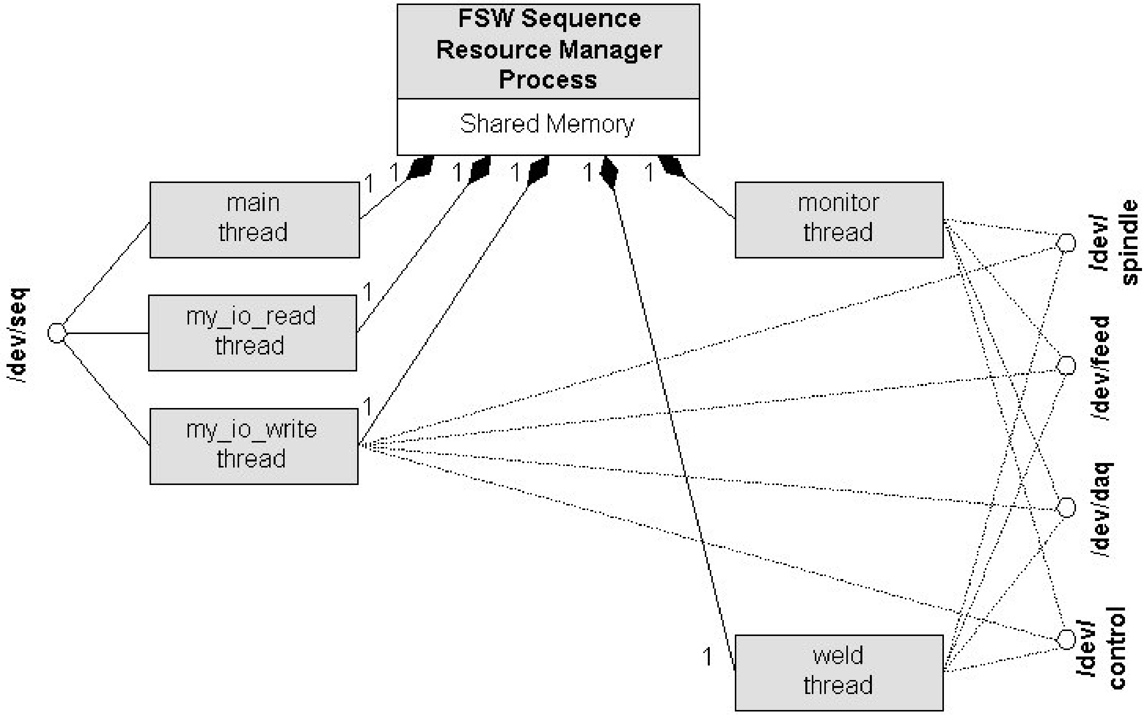


Figure 5.17: UML static object diagram of FSW sequence controller resource manager

Command	Example Value	Description
start_weld	--	Start the FSW weld
stop_weld	--	Stop the FSW weld
machine_home	--	Home the three axes
plunge_rate	10	Set the rate to plunge the tool's pin (mm/min)
weld_rate	100	Set the weld traverse speed (mm/min)

weld_spindle_speed	560	Set the tool speed to weld at (RPM)
max_pin_temperature	700	Set the maximum limit for the thermocouple mounted on the tool (^{o}C)

Table 5.7: FSW sequence command examples

- The monitor thread, reads data from the spindle, feed, daq and control resource managers and then compares the values to maximum and minimum limits that the user can set if desired. This can be used for protection against out of range conditions.
- The FSW weld thread is created; each time the machine is given a command that involves motion (e.g. start_weld, machine_home, etc). The thread co-ordinates all the stages of the FSW process. After the task is complete the thread dies.

The FSW process can be divided into the following stages:

- Move to the weld start position (X and Y axes).
- Start the spindle at the user specified plunge spindle speed.
- Find the surface of the material by feeding the Z-axis and monitoring the vertical force present on the tool.
- Plunge tool pin to required depth then stop the vertical feed and wait for the dwell period.
- Traverse the weld path at the selected weld spindle speed.
- Wait for the correct distance to be traversed and then stop the feed.
- Extract the pin from the material.
- Home the machine.

This resource manager provides the following services:

- Monitoring of machine maximum and minimum limits
- Controlling and FSW welding process
- Processing real-time process data from the spindle, feed, DAQ and control resource managers into a single block of data. This data is sent to a client when a read request is issued. The user interface data server, uses this information to update the GUI.

With regard to the last point, the processing of the 2D polar force plot needs to be further discussed. The 2D force data obtained from the DAQ resource manager is not ideal for graphic display. This is because each revolution will produce a different set of angles, causing the plot to appear unstable. Linear interpolation is therefore used to align the acquired angles with a set of user-defined angles providing a more readable graphic display. A standard linear interpolation formula was used, as shown in Formula 5.10.

$$Y_3 = Y_1 + \frac{x_3 - x_1}{x_2 - x_1}(Y_2 - Y_1) \qquad (5.10)$$

2.16.2 User Interface Data Server

The user interface data server is used to send machine and process information to the distributed user interface and to receive user specified commands and configuration information.

This is implemented using a thin-client, client-server architectural model, where all of the application processing and data management is carried out on the server. The client is simply responsible for running the presentation software [Sommerville, 2001].

A disadvantage of this model is that heavy processing load is placed on the server and network [Sommerville, 2001]. This however, isn’t a major problem currently with the FSW software, as process variables change slowly so a UI update of 500ms is sufficient. Data sent from the UI to the server are only periodic commands when the user changes a parameter. If further monitoring is necessary, such as a video feed, network bandwidth limitations may become apparent.

The UI data server is not a resource manager like the rest of the modules, because it needs to provide functionality via an Ethernet network as apposed to being pathname mapped. Its internal structure is shown in Figure 5.18.

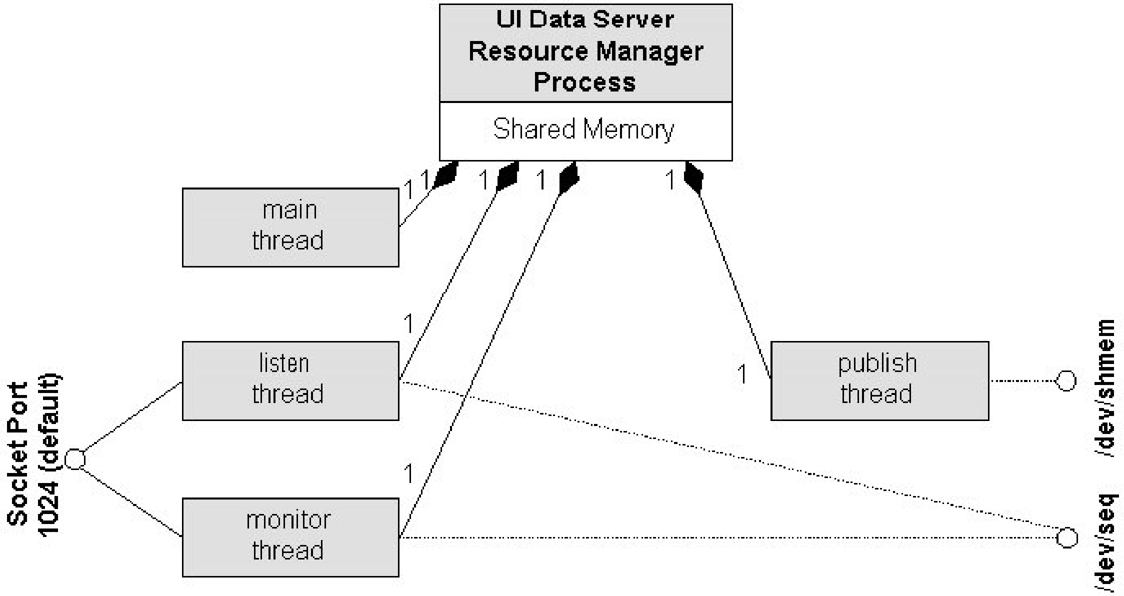


Figure 5.18: UML static object diagram of GUI network data server

2.16.2.1. Aim of the Server

The UI data server must send packets of data at regular intervals in order to keep the clients GUI updated with the current state of the FSW machine and welding process. This information must be read periodically from the FSW sequence resource manager (/dev/fsw), processed and converted to the XML data format before it is sent over the network. The client sends commands to this server asynchronously, which must be processed and sent to the FSW sequence resource manager, shown in Table 5.8

The server supports the following commands sent to it by the GUI:

R	Request the data server to periodically send information to update the user interface.
S	Stop sending periodic user interface information.
P	Convert the raw binary files containing the recorded process data into a comma-delimited format and publish it on the web page.
D	Inform data server to stop sending periodic information and close the connection to the user interface

Table 5.8: Commands supported by user interface data server

2.16.2.2. Explanation of Operation

The operating system features a full implementation of the industry standard TCP/IP protocol stack. The QNX socket IP manager supports a variety of protocols, including TCP, UDP and IP. It also supports a wide range of network hardware. TCP/IP for QNX comes with a Berkeley socket API that was used to establish peer-to-peer communications over a TCP/IP connection [QNX Software Systems, 1999]. This provides a connection-orientated service for the FSW software to enable interaction between the UI data server and client GUI.

QNX is supplied with a web server, called Slinger, which communicates over TCP/IP. The server supports CGI 1.1, HTTP 1.1, and dynamic HTML [QNX software systems, Architecture Guide, 2002].

The Macromedia Flash player, necessary to display the GUI, is embedded as an object in a HTML web page. To access the GUI the following procedure, shown in Figure 5.19, is necessary [Shay, 1999]:

- The client loads a web browser (Opera, Internet Explorer, etc) with the URL of the FSW GUI web server (e.g. fsw.petech.ac.za).
- The web browser then performs a DNS lookup to determine the IP address associated with the URL.

- The client’s web browser sends a request to the web server (slinger) for the web page associated with the URL.
- The web server sends the HTML file to the client.
- The client sends a request to the web server for the macromedia flash media file, because the HTML file specified that this file must be retrieved.

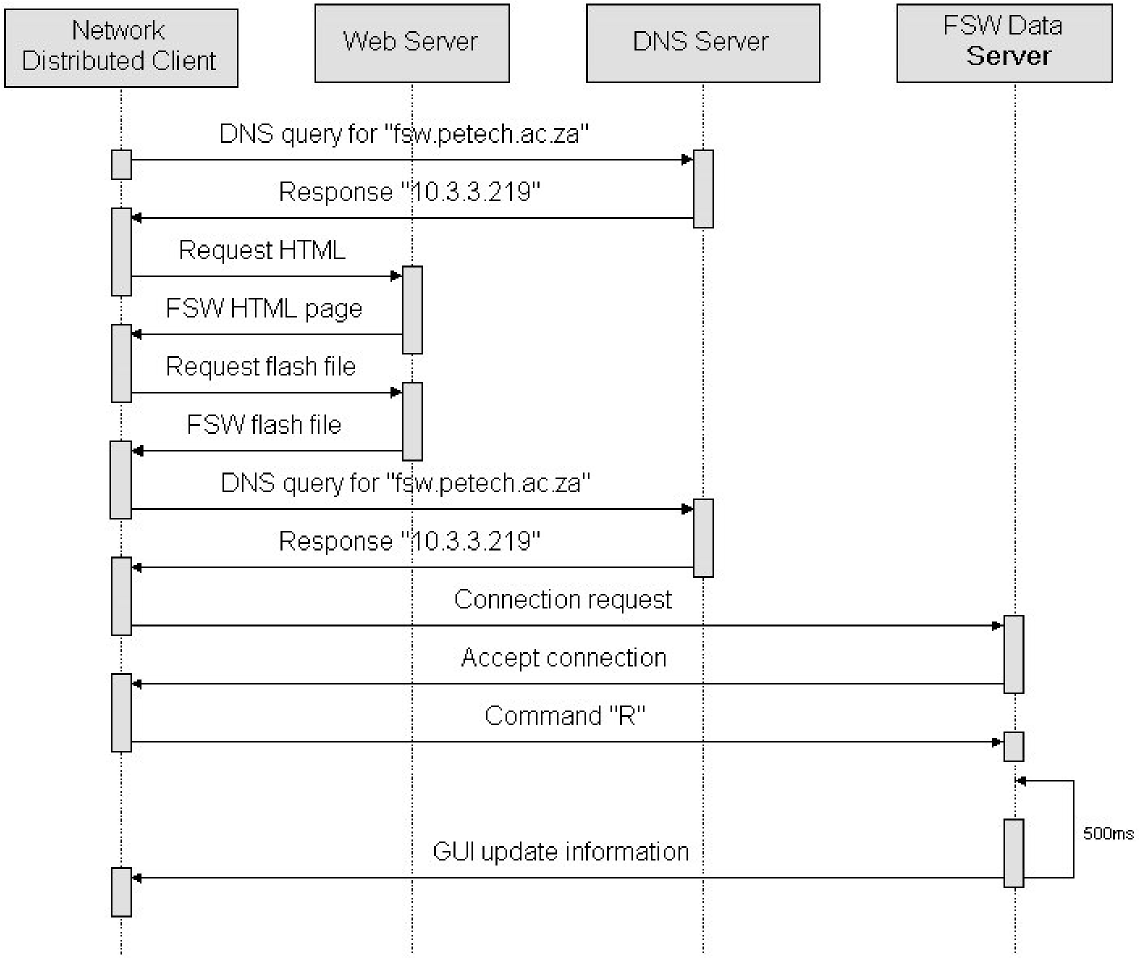


Figure 5.19: UML sequence diagram of client-server communication initiation

- The web server sends the media file to the client.
- Upon reception, the client browser loads its embedded flash media player and displays the GUI within the web browser.
- Once the FSW GUI media file is loaded, the action script contained within the media file, attempts to connect to the FSW data server.
- If this is successful, the FSW GUI will send a command to the server requesting the periodic transmission of machine status and process data in the XML format.
- The FSW data server will then proceed to send this information to the client, until requested it to stop or a network error occurs.

XML is becoming the standard for the interchange of structured data in Internet applications. In XML, as with HTML, you can use tags to markup, or specify, a body of text. The difference being that XML tags can be user-defined to specify a piece of data (e.g. <username>myname</username>). XML separates the structure of the information from the way it is displayed. This allows the same XML document to be used and reused in different environments.

Each XML tag is called a node, or an element. Each node has a type (1-XML element, or 3-text node) and elements may be assigned attributes. A node nested in a node is called a child or a child node. This hierarchical tree structure is called the XML DOM [Macromedia, 2000].

In addition to sending data and receiving commands, the FSW data server is also responsible to publishing the data recorded to the RAM disk, from the spindle, feed and chuck resource managers. The publishing process consists of:

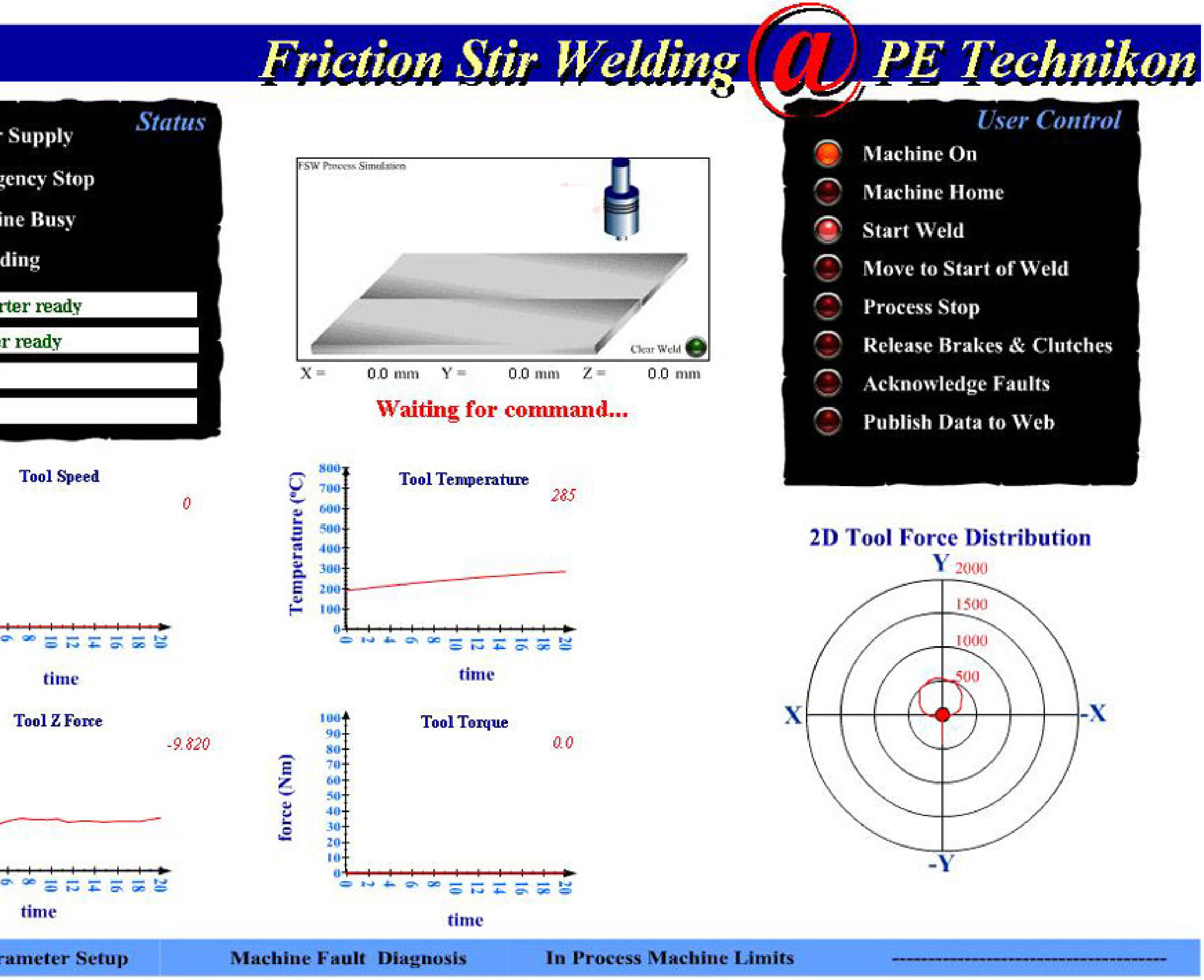
- Reading the binary data from the three ram disk files (spindle.dat, feed.dat and tool.dat)
- Processing it into a usable format
- Storing the processed data as comma delimited ASCII files in a location where the web server can access them when a client web browser requests them

This process is only executed when requested by the user. The “publish data to web” button on the GUI must be clicked, causing the “P” command to sent to the UI data server.

2.16.3 User Interface

Because of increased time constraints, increased productivity and quality, industry is adopting information technology (IT) at growing rates. Methods and tools available for production systems development are less advanced than software-engineering tools used for information systems development (IS). The IS tools have become standardized and easier to use, allowing increased efficiency. Usually software for a production system (PS) was developed for each controller using controller-specific tools designed to be similar to the tools that those in charge of software development for computer-controlled processes were used to [Edan, Pliskin, 2002].

The user interface for the FSW machine was created using Macromedia Flash, an Internet based standard used for interactive presentations and animations.



Waiting for command...

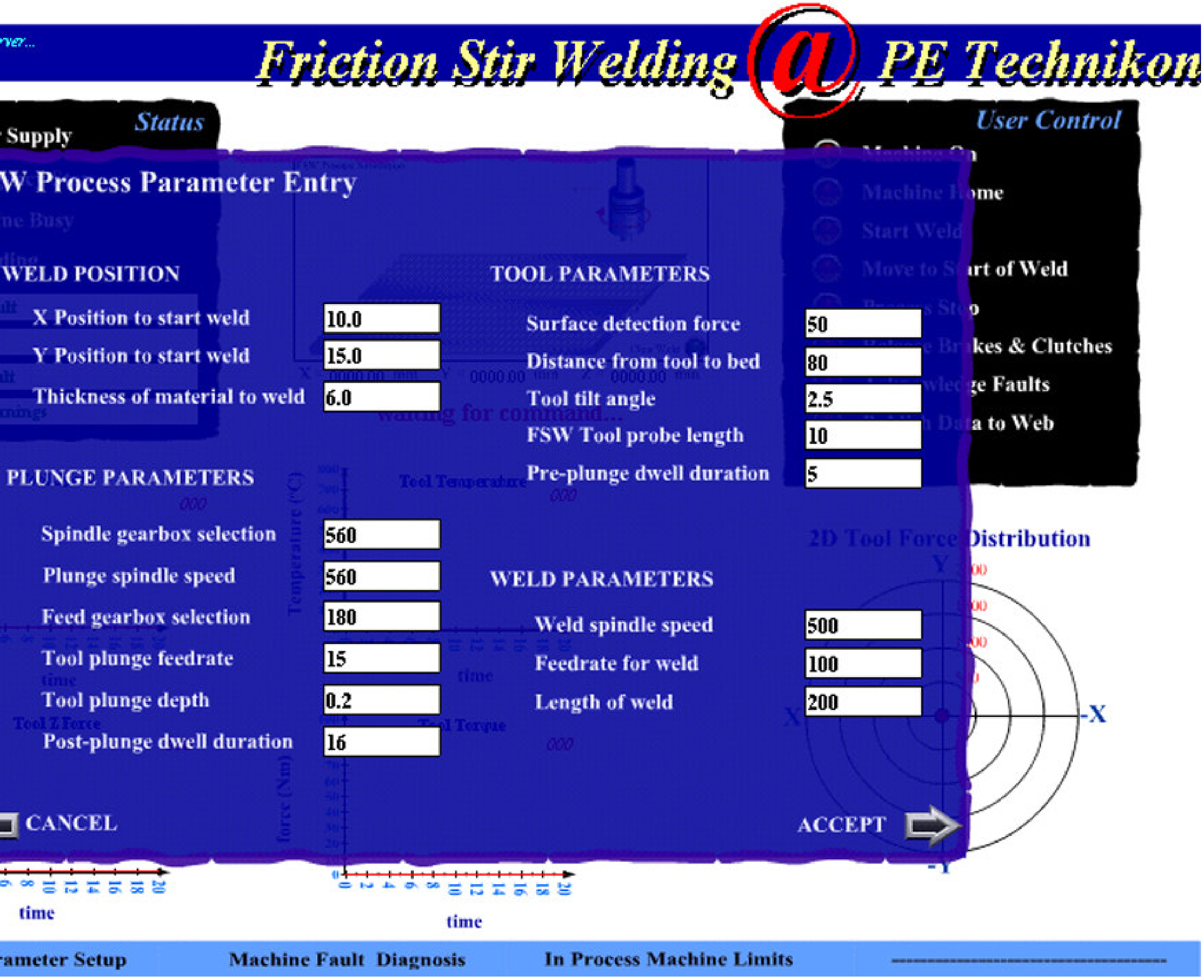
Figure 5.20: Screen shot of the network distributed graphical user interface

Figure 5.20 shows the GUI that was developed. The black box at the top right, allows the user to control the machine or process. The illustration in the center provides a simulation of the machine's actual state, showing the welds progression. At the top left, machine status information is displayed, such as faults, warnings, power state, etc. Under this, a series of graphs can be seen, which display relevant in process information, such as:

- Vertical force exerted on the tool (lower left graph)
- Torque exerted on the tool (lower right graph)
- Tool pin temperature (upper right graph)
- Tool speed (upper left graph)

The graph on the far right provides a polar plot of the resultant forces exerted on the tool's pin. The menu along the bottom of the screen provides a means to:

- Enter FSW process parameters to use during welding. These include plunge depth, feed rate, spindle speed, etc. These are shown in Figure 5.21.
- View the status of the inverters to assist in fault diagnosis, shown in Figure 5.22.
- Set or adjust maximum and minimum machine limits, to prevent damage to the machine or tool certain constraints should not be exceeded. Figure 5.23 shows this window.



W Process Parameter Entry

WELD POSITION

X Position to start weld: 10.0
Y Position to start weld: 15.0
Thickness of material to weld: 6.0

PLUNGE PARAMETERS

Spindle gearbox selection: 560
Plunge spindle speed: 560
Feed gearbox selection: 180
Tool plunge feedrate: 15
Tool plunge depth: 0.2
Post-plunge dwell duration: 16

TOOL PARAMETERS

Surface detection force: 50
Distance from tool to bed: 80
Tool tilt angle: 2.5
FSW Tool probe length: 10
Pre-plunge dwell duration: 5

WELD PARAMETERS

Weld spindle speed: 500
Feedrate for weld: 100
Length of weld: 200

Figure 5.21: Window used to configure FSW process parameters

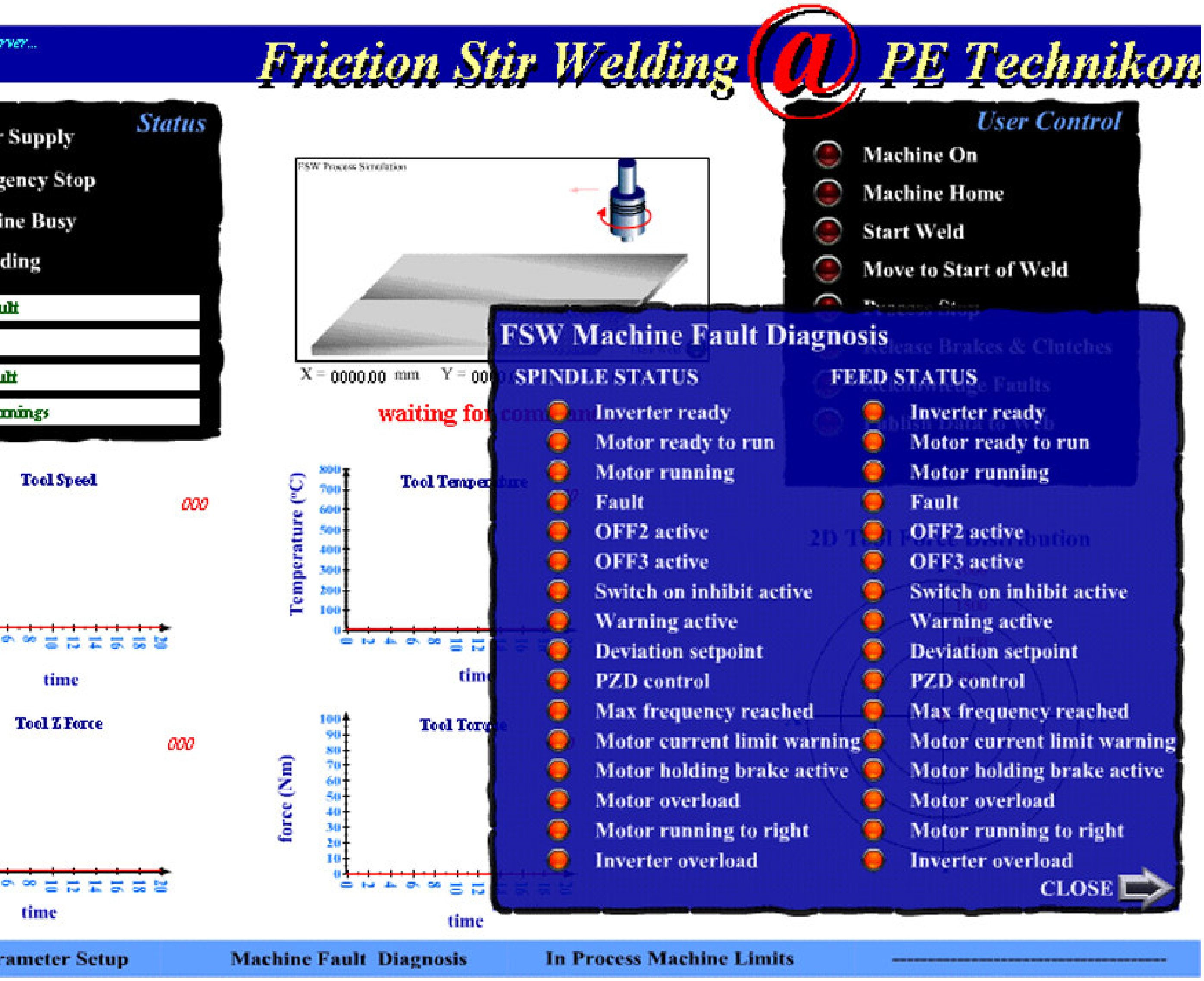
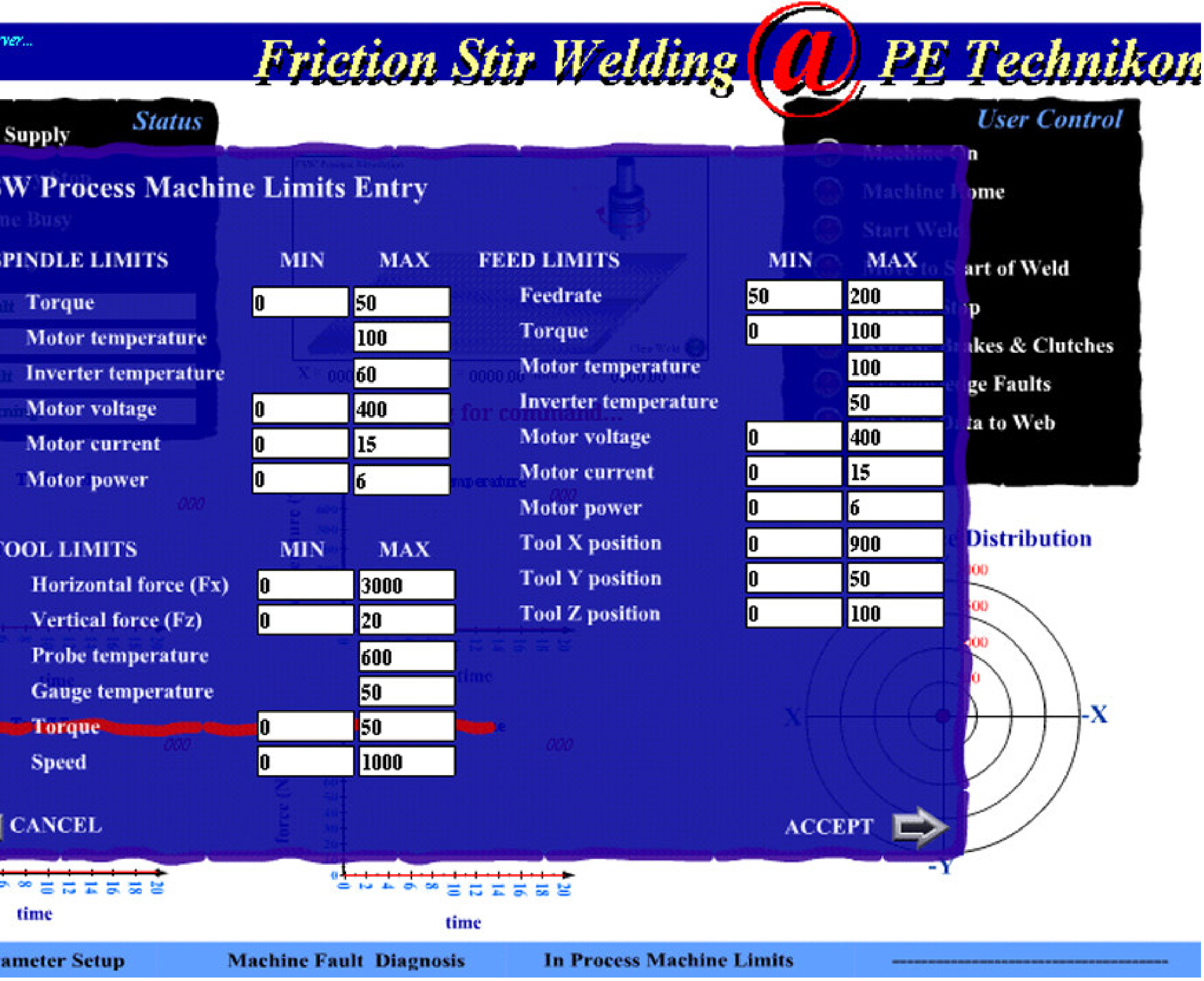

Friction Stir Welding @ PE Technikon
User Control
Machine On
Machine Home
Start Weld
Move to Start of Weld
Status
Supply
gency Stop
ine Busy
ding
ult
ult
rnings
FSW Process Simulation
X = 0000.00 mm
Y = 00
waiting fo
Tool Speed
000
time
Tool Z Force
000
time
Tool Temper
Temperature (°C)
time
Tool Tor
force (Nm)
time
FSW Machine Fault Diagnosis
SPINDLE STATUS
Inverter ready
Motor ready to run
Motor running
Fault
OFF2 active
OFF3 active
Switch on inhibit active
Warning active
Deviation setpoint
PZD control
Max frequency reached
Motor current limit warning
Motor holding brake active
Motor overload
Motor running to right
Inverter overload
FEED STATUS
Inverter ready
Motor ready to run
Motor running
Fault
OFF2 active
OFF3 active
Switch on inhibit active
Warning active
Deviation setpoint
PZD control
Max frequency reached
Motor current limit warning
Motor holding brake active
Motor overload
Motor running to right
Inverter overload
CLOSE
rameter Setup
Machine Fault Diagnosis
In Process Machine Limits

Figure 5.22: Window used to diagnose machine faults



W Process Machine Limits Entry

PINDLE LIMITS	MIN	MAX
Torque	0	50
Motor temperature		100
Inverter temperature		60
Motor voltage	0	400
Motor current	0	15
Motor power	0	6

OOL LIMITS	MIN	MAX
Horizontal force (Fx)	0	3000
Vertical force (Fz)	0	20
Probe temperature		600
Gauge temperature		50
Torque	0	50
Speed	0	1000

FEED LIMITS	MIN	MAX
Feedrate	50	200
Torque	0	100
Motor temperature		100
Inverter temperature		50
Motor voltage	0	400
Motor current	0	15
Motor power	0	6
Tool X position	0	900
Tool Y position	0	50
Tool Z position	0	100

Figure 5.23: Window used to adjust FSW machine limits

The indicator at the extreme top left of the screen will flash while data is being transferred between the GUI and server. This can be used to ensure communications are operating correctly as the indicator should not stop flashing. Should communications fail, the machine will complete its current operation, adhering to all machining limits and parameters set by the user before the communications failure.

2.16.3.1. Aim of the Graphical User Interface

The aim of using a GUI is that they are relatively easy to learn and use. Users with little computing experience can learn to use the interface after a brief training session. Also using multiple forms prevents the user-losing track of the information generated [Sommerville, 2001].

This GUI provides a simple, easy interface for controlling the FSW machine. The UI is network distributed, allowing one user to access and control the machine from anywhere on the LAN. User may bring their computer or laptops to the machine and connect it to the network. This allows them to complete an FSW weld and save the FSW weld data directly to their computer, without having to uses disks or shared folders. All the user needs to have installed is a web browser, with the Macromeda Flash 5 player. This comes standard with Windows 98 and above. Otherwise it is easily downloadable from the Internet.

Figure 5.24 shows the composition of the GUI movie.

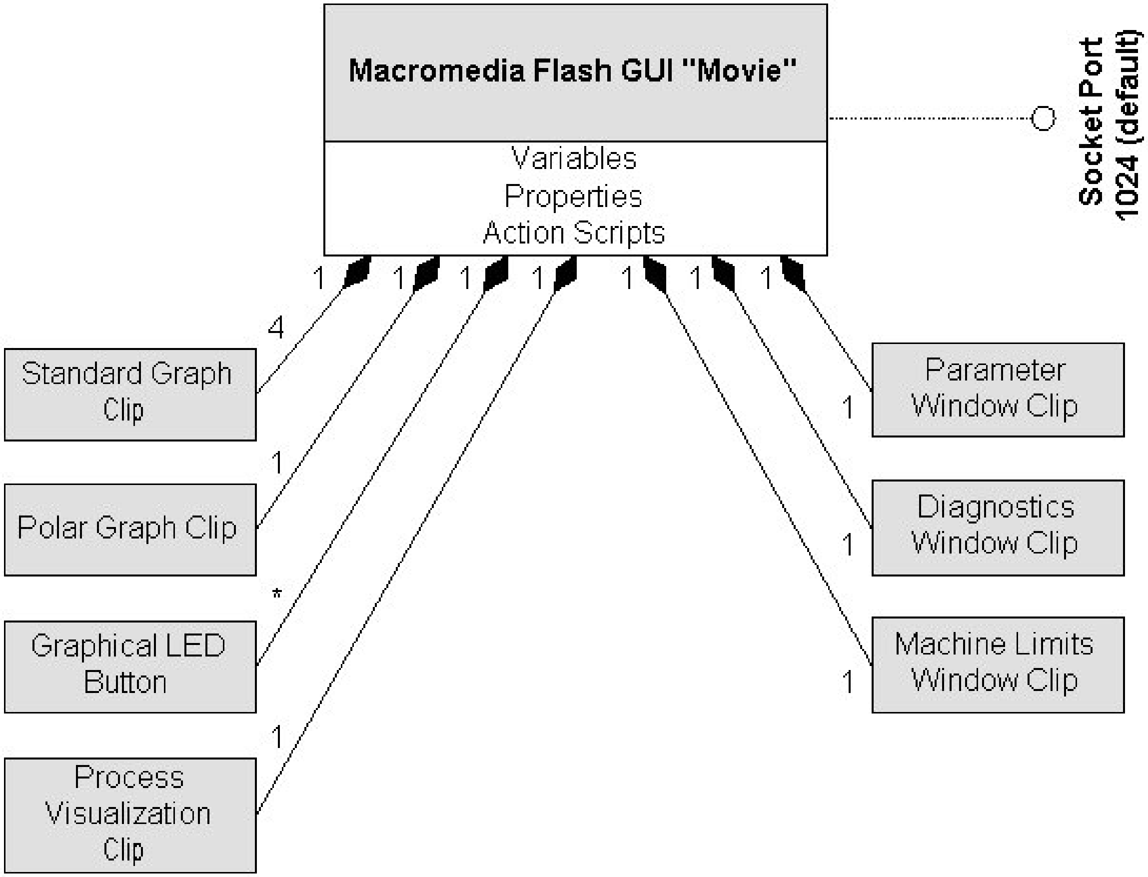


Figure 5.24: UML static object diagram of the FSW GUI movie

2.16.3.2. Explanation of Operation

The GUI's functionality was programmed using Macromedia Flash Action-scripts, which is very similar to ANSI C. Action-scripts are blocks of code that are executed when a specific event occurs. This scripting language comes with a large support function base.

The complete page is called a "movie", containing clips, graphics and buttons. A graphic is just a simple image displayed on the movie. Clips can contain Action-scripts and have the same properties as a movie.

Action-scripts support XML sockets which are used to connect the UI data server and the network distributed GUI. The XML children form variables within the movie when the GUI receives the XML packet. These variables can then be used directly to update the display.

2.17 Summary

The chapter begins by explaining the reason the C programming language was predominantly used for the project. The layered architectural design using the data-flow model is then discussed. Resource managers are used extensively in the software architecture and so the basic code for implementing a resource manager is explained. After this, each layer of the hierarchical architecture is mentioned. The remainder of this chapter discusses each software module constituting the architecture and its layers, with regard to its internal structure, function and interface to other modules.

The following chapter uses the machine developed in this and previous chapters to perform FSW weld tests. Statistical analysis is then performed on the resulting data in order to analyze the process. This analysis is used to build control rules, with a view to designing an intelligent controller for the FSW process.

CHAPTER 6

Calibration and Data Analysis of the FSW Process

This chapter discusses the calibration of the machine's measurement transducers and the statistical analysis of process data to enable the generation of control rules.

6.1 Calibration of Transducers

There are two main sources of sampled data used to monitor and control the process, which are the LMI chuck-monitoring unit and the Siemens MM440 inverters.

LMI calibrated the chuck interface unit so that the correct range of output values is sent to the computer for a specified range of input values from the sensors mounted on the chuck [Husselmann, 2002].

The measured strain readings from the strain gauges mounted on the chuck were compared to applied loadings measured using a loadcell in order to verify the calibration of the system [Blignault, 2002].

6.1.1 Tool Horizontal Force Strain Gauge

To verify the measurements obtained from the horizontal force strain gauge, the calibration setup in Figure 6.1 was used. A custom-made two-quarter bridge diagonal loadcell was used to obtain the values. Care was taken to ensure that the strain gauge mounted on the chuck was aligned parallel to the applied force. The force measured by the LMI interface unit was then compared to the force measured using a strain gauge amplifier.

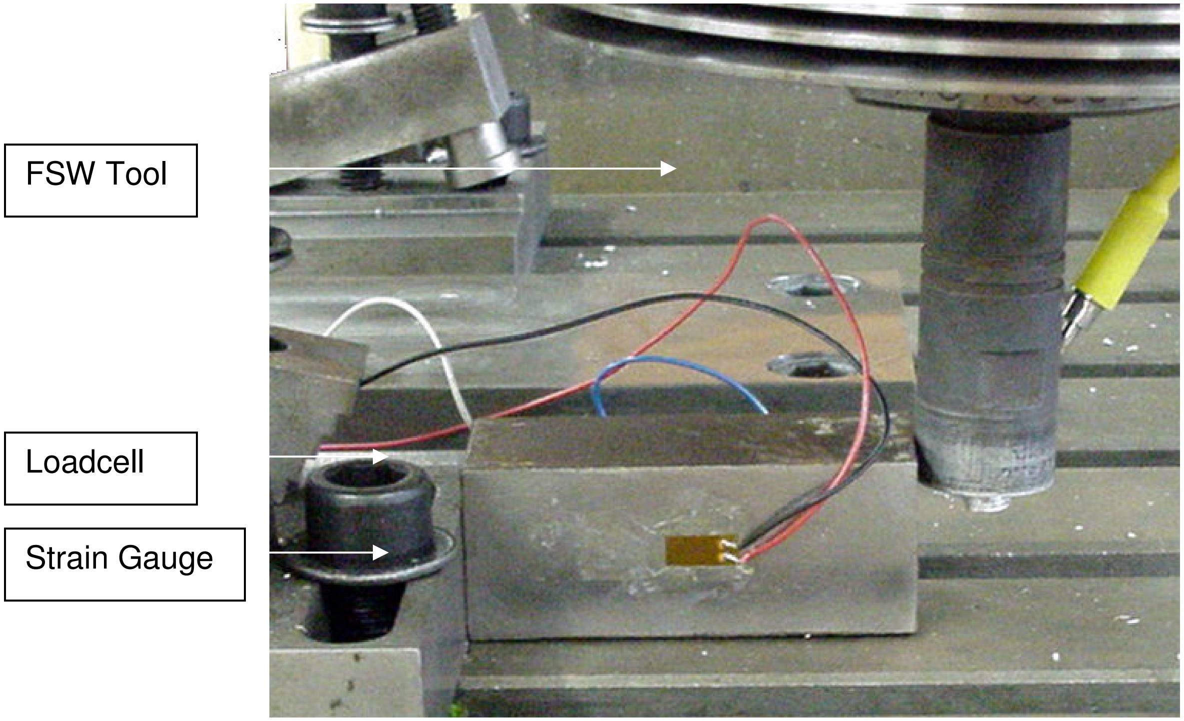


Figure 6.1: Loadcell used to verify calibration

During the calibration procedure the most difficult obstacle was to accurately calibrate the 360° polar force pattern. A constant force was applied to a point on the tool while it was rotating and the horizontal force (x-force) was recorded. These results were then compared to an ideal force distribution plotted using a mathematical formula.

Since the measuring is done by means of strain gauges on a rotating chuck arm, it can be closely related to the existing theory relating to a bar (square or tubular) in tension or compression. The maximum tension or compressive strains exist in the direction of the applied force. In all other directions the strain is smaller and follow the relationship as shown in Figure 6.2 [Hoffmann, 1989]. Strain was used for this discussion, as opposed to stress, as a strain gauge measures strain.

There are two defined strain directions, one in the active direction of the force (0^o) and the other perpendicular to it (90^o). Starting from the principal strain $\varepsilon_0 = \varepsilon_1$, the strains ε_ϕ, which occur at the angle $0^o < \phi < 90^o$ to the direction, can be calculated according to Formula 6.1 [Hoffmann, 1989]:

$$\varepsilon = f(\phi) = \frac{1}{2}\varepsilon_1[1 - \nu + \cos 2\phi \cdot (1 + \nu)] \qquad (6.1)$$

The relationship between the two principal strains is expressed by the transverse strain factor ‘m’ or its reciprocal, Poison’s ratio ν.

$$\varepsilon_2 = -\nu . \varepsilon_1 \qquad (6.2)$$

In Figures 6.2 and 6.3 the relationship is shown for a tension bar and a compression bar respectively. These diagrams are drawn for a material with a Poison’s ratio $\nu = 0.3$. In this case the strain becomes zero at an angle $\phi = 61^{o}$, i.e. the zero crossover between the positive and the negative strain regions.

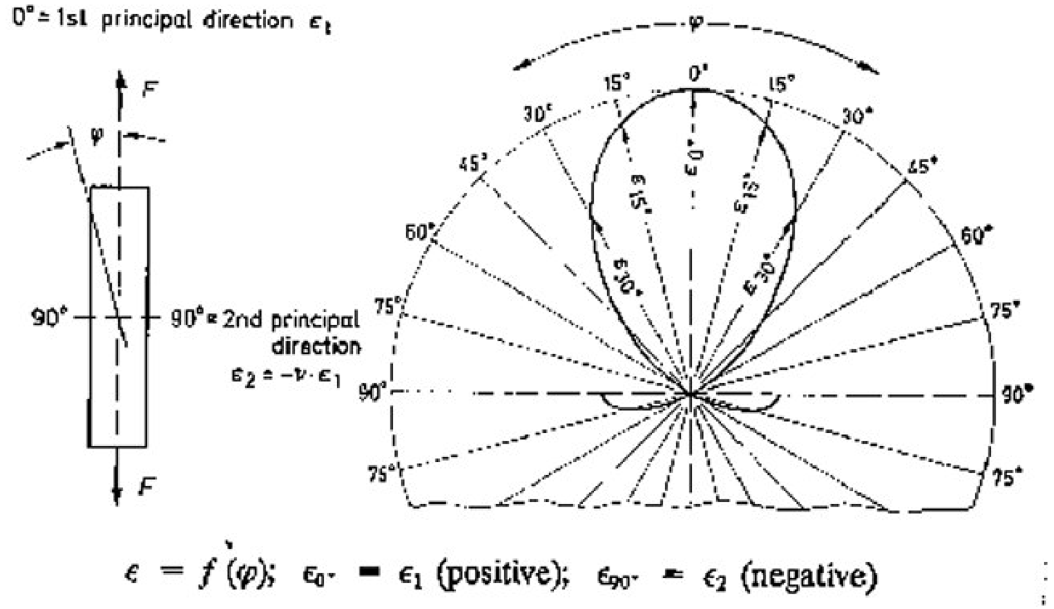
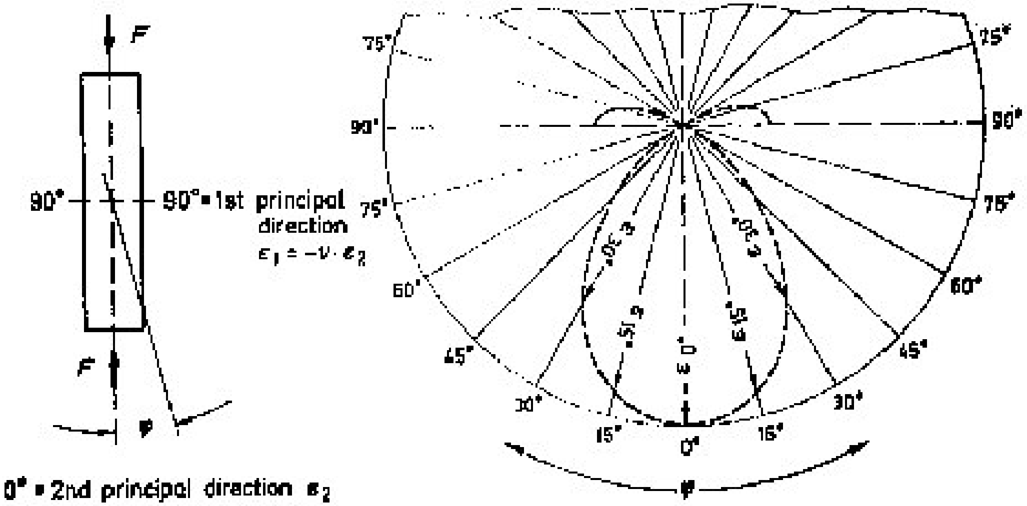


Figure 6.2: Strain distribution on the tension bar [Hoffmann, 1989]

The difference between the stress distribution and the strain distribution with regard to their dependence on the active direction of the force, leads to an extremely important conclusion: The material stress σ should be calculated from the measured strain only according to Hooks's Law for the uniaxial stress state as shown in Formula 6.3.

(6.3)

$$\sigma = \varepsilon . E$$



$$\varepsilon = f(\varphi); \quad \varepsilon_{0^\circ} = \varepsilon_2 \text{ (negative)}; \quad \varepsilon_{90^\circ} = \varepsilon_1 \text{ (positive)}$$

Figure 6.3: Strain distribution on the compressive bar [Hoffmann, 1989]

In the transverse direction (90^{o} direction) there is no material stress present despite the measurable strain (transverse contraction, transverse dilation). Therefore for reliable results the active direction of the force must be known and the strain must be measured in this direction.

The calibration of the polar force pattern is based on exactly the same theory that was explained above. A force was applied in a known direction on a non-profiled tool, to ensure symmetrical readings. The strain was recorded from the measuring unit via two RS-232 links to the computer software program. The software then further processed the samples. Firstly a unidirectional force was applied on the tool shoulder and plotted as shown in Figure 6.4. Also shown in this figure is the corresponding force profile of the calculated data obtained when using Formula 6.2.

A mathematical definition of stress is shown in Formula 6.4. By applying this relationship to Formula 6.3, the applied force can be derived for a given strain.

$$\sigma = F / A \qquad (6.4)$$

This results in Formula 6.5, which was used to plot Figure 6.4, showing the relationship between the theoretical and practical forces obtained.

$$F = \varepsilon \cdot E \cdot A \tag{6.5}$$

The variables E and A are constants, representing Young's modulus and the cross-sectional area of the shaft on which the strain gauge is mounted. Therefore it can be seen that force and strain are directly proportional.

Calibration Plot - Non profile Tool

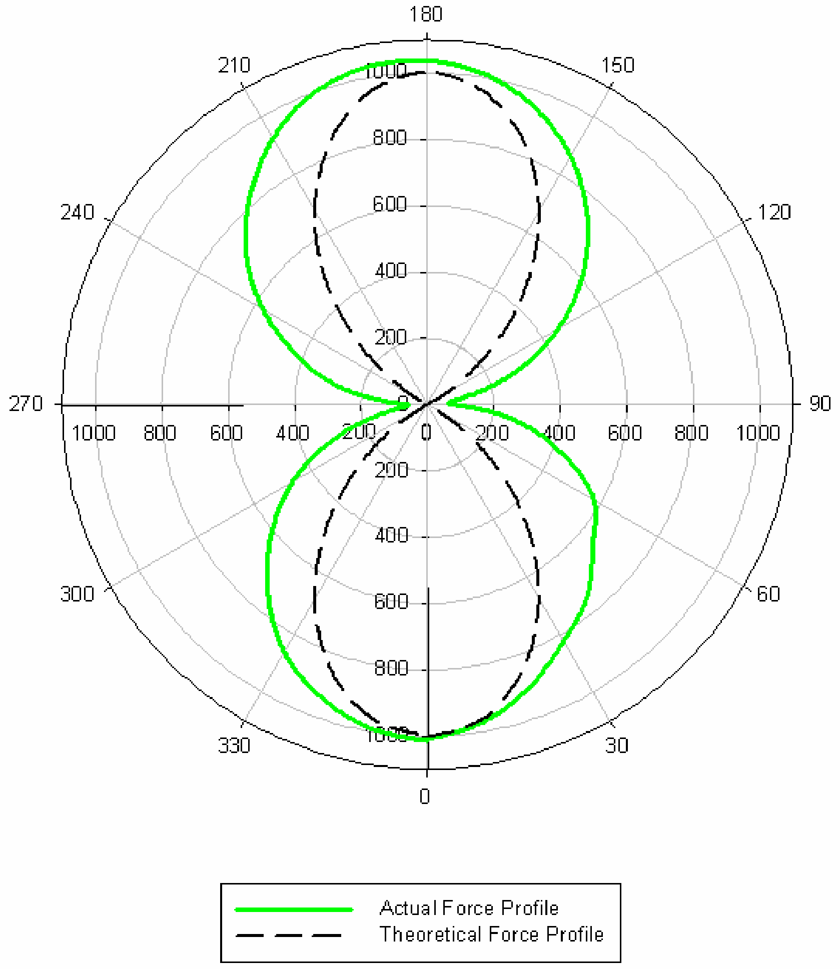


Figure 6.4: Polar force plots for calibration purposes

A deduction can be made from Figure 6.4 illustrating that the force trend lines are of the same nature and the transition zones are at the same interval. The narrower gradient of the calculated curve (dashed) is due to the fact that a sharper point load is applied on a flat theoretical surface. The more rounded gradient of the actual force plot (solid) is due to the load that is applied on a round, rotating, surface giving a more dampened gradient as shown in Figure 6.5.

(a) Practical (b) Ideal

Figure 6.5: Practical and ideal point forces applied to a bar

6.1.2 Tool Vertical Force Strain Gauge

The z-force was verified using the same custom-made two-quarter bridge diagonal loadcell, used to verify the x-force response [Blignault, 2002]. Figure 6.6 shows the procedure used to measure the vertical force. The same procedure used to verify the x-force values was followed. In both cases, a linear relationship was obtained between the applied force and the measured force.

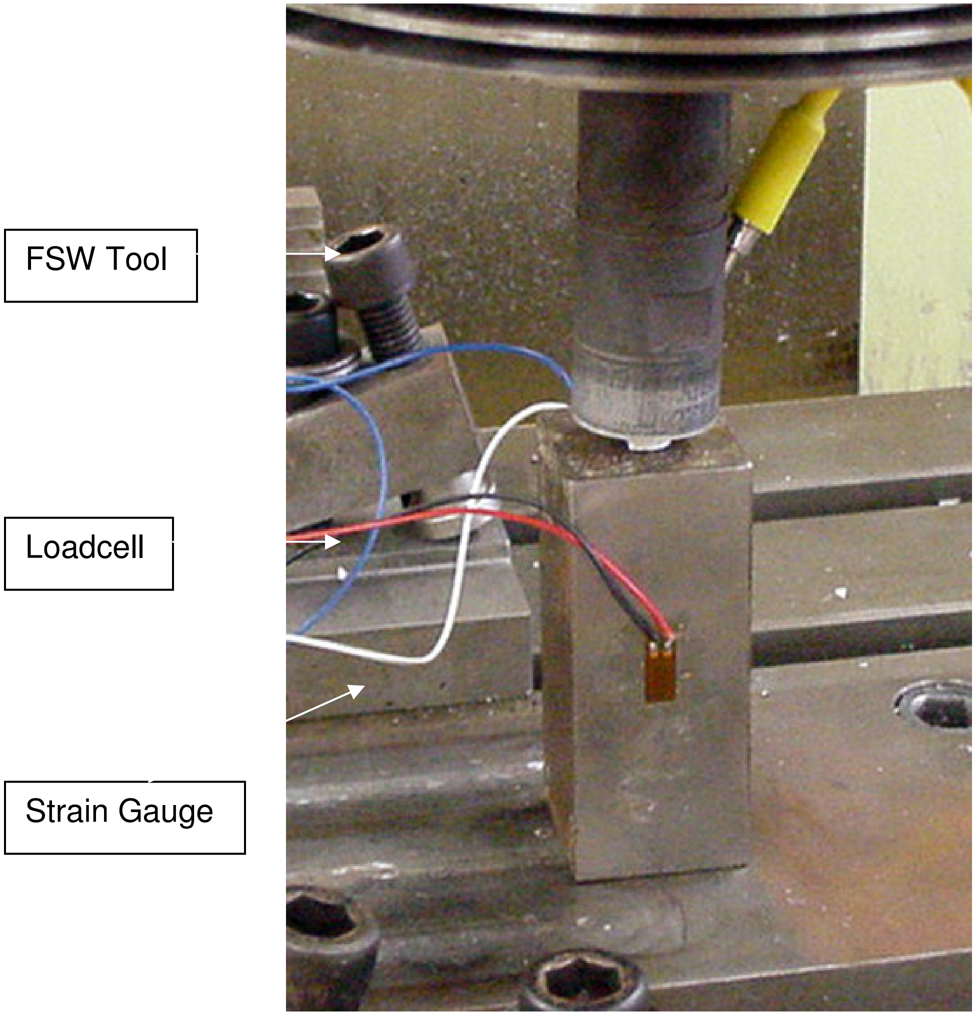


Figure 6.6: Loadcell used to verify calibration

6.1.3 Tool Thermocouple

When the chuck-monitoring unit was designed, the temperature to voltage response graph was supplied to LMI and used to calibrate the tool's custom build thermocouple. This graph was obtained from the company that designed to thermocouple to suit the application. The calibration was verified using a portable handheld precision digital thermometer, which also uses a thermocouple. Both thermocouples were heated simultaneously and the resulting values compared.

6.1.4 Siemens MM440 Inverters

After installation of the inverters, an automatic motor identification process was executed. This allowed the feed and spindle inverters to detect the motor parameters and tune themselves. The control mode for both inverters was set to vector control mode, because of the following:

- The speed of the rotor can be controlled with inherent slip compensation
- High torque
- Transient response is improved
- Excellent speed holding
- Improved torque is obtained at low frequencies

The inverters handle the calibration of parameters measured from the motors [Siemens, MM440 Operating Manual, 2002].

6.2 Repeatability Tests

After the calibration of the measuring unit, three tests were performed to verify the repeatability of the system. The three tests conducted were all done at 560 RPM, 0.1 mm plunge depth and with a feed rate of 80 mm/min with a non-profiled tool. The results for the three 2D force plots repeatability tests can be found in Appendix C. These plots show the same orientation and profile of forces around the tool in all three tests conducted. The readings do fluctuate slightly due to the high gain setting used during signal conditional by the chuck interface unit set by LMI. The accuracy and repeatability is estimated to be in the range of 100 to 150N for the X and Y forces and 200N for the z-force. Other possible reasons for the small discrepancies in magnitudes between tensile and compressive forces are the fact that the tool is rotating and a frictional (rubbing velocity) vector could cause an additional force opposing the direction of travel. The tool eccentricity factor may also have an influence in non-symmetrical profiles. This scenario must still be further investigated to verify the validity of these statements. The objective of this report is not to investigate all these forces, therefore comment on their effects are limited. All tests performed used aluminum alloy 5083.

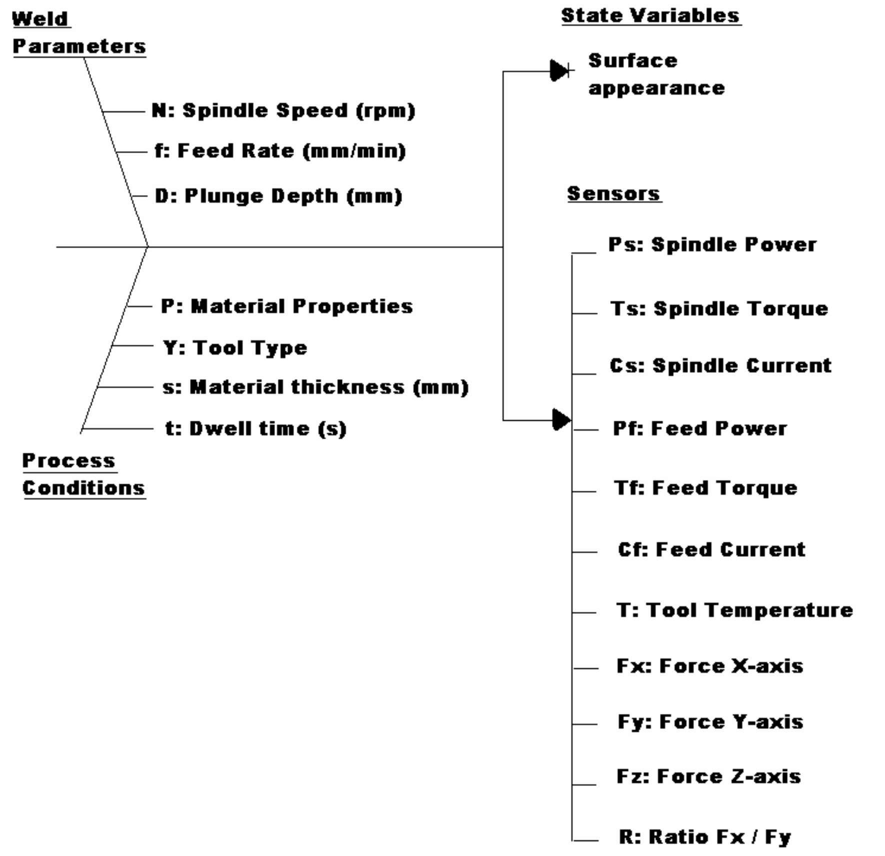
6.3 Data Analysis Experimental Procedure

There are various methods that can be used when performing data analysis, which include evaluating the effects of changing one parameter at a time or using a factorial design. The latter option results in a large number of tests, which may be impractical depending on the number of parameters [Azouzi, Guillot, 1997].

For FSW, multiple short welds may be conducted on the same set of plates. However, a cooling period must be exercised to prevent residual heat in the tool or plate affecting the next weld. From practical experience, it was found that only one test could be completed every 30 minutes at best.

An efficient testing strategy, called orthogonal arrays (OA) was developed by Taguchi to minimize the number of tests required. This is a fractional factorial design technique. An advantage of an OA design is its equal representation of all factors, some combinations of factors and factor levels are tested which otherwise would have not been investigated. Accordingly, the OA will be used for the design of experiments and models. The OA that best fits this FSW experimentation is the L_{16} with a total of 16 tests [Azouzi, Guillot, 1997].

Figure 6.7 shows the design cause-effect diagram used. Parameters N, f and D were assigned four different levels varying from 300 to 600 RPM, 40 to 100 mm/min and 0.1 to 0.4mm plunge depth, respectively. These ranges were chosen to ensure maximum deviation of the dependent variables. Appendix D, Table D.1 contains the complete list of tests performed and their results. The tests were not conducted sequentially, so the effect of systematic errors could be reduced. By randomly performing tests, systematic errors become random errors, which be dealt with using statistics. Appendix E contains images of the surface finish of each weld conducted during the tests.



Material Properties:	Tool Type:	Material thickness:	Dwell time:

P1: Oxide layer	**Y1: Profiled**	**s1: 6mm**	**t1: 16s**
P2: No-oxide layer	**Y2: Non-profiled**	**s2: 8mm**	**t2: 8s**

Figure 6.7: Weld Process cause-effect diagram and factor levels

6.4 Statistical Data Analysis

Statistical analysis software was used to analyze the results of the tests performed for the following reasons [Zeelie, 2002]:

- Clear and simple presentation of the test results
- Extraction of the maximum information for a given set of experiments
- Derivation of the correct conclusions in spite of the variability in the experimental results

There are three types of error that must be avoided to ensure experiments give a true response [Pietersen, 2002]:

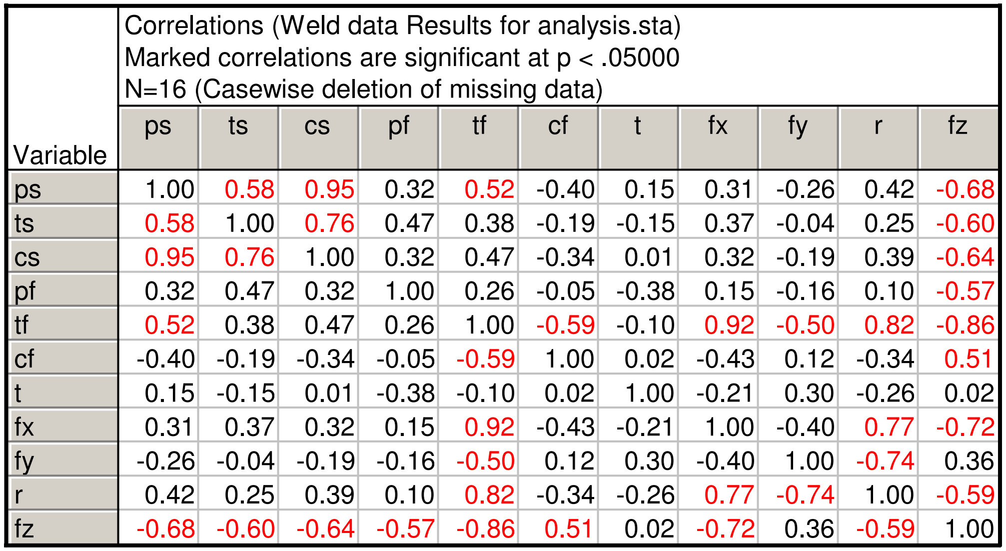
- **Gross Errors** – The existence of these errors requires the experiment to be restarted. Causes of these errors include instrumentation malfunction or vital error in the conduct of the experiment.
- **Systematic Errors** – These errors give results that are always lower or higher than the true value. Causes include poor equipment calibration or inaccurate control of the test.
- **Random Errors** – These errors give test results that fall randomly on both sides of the true value. Causes include slight variations in process conditions outside the control of the researcher.

Each weld was 100mm in length and the average of the data recorded between 80mm and 90mm was used for the statistical analysis. This is because it was practically found that the process variables have settled and are more stable near the end of the weld than at the start. The data obtained from the weld tests and used for the statistical analysis can be found in Appendix D, Table D.2.

6.4.1 Analysis of Variants (ANOVA)

ANOVA is an extremely powerful statistical technique, which can be used to separate and estimate the different causes of variation, i.e. variation from random error and variation from deliberately changing experimental conditions [Zeelie, 2002].

The ANOVA technique was applied to dependent variables of FSW test result data using Statistica, which is a Windows based statistical analysis software package. Table 6.1 shows us the correlation between each of the independent variables.



Correlations (Weld data Results for analysis.sta)
Marked correlations are significant at p < .05000
N=16 (Casewise deletion of missing data)

Variable	ps	ts	cs	pf	tf	cf	t	fx	fy	r	fz
ps	1.00	0.58	0.95	0.32	0.52	-0.40	0.15	0.31	-0.26	0.42	-0.68
ts	0.58	1.00	0.76	0.47	0.38	-0.19	-0.15	0.37	-0.04	0.25	-0.60
cs	0.95	0.76	1.00	0.32	0.47	-0.34	0.01	0.32	-0.19	0.39	-0.64
pf	0.32	0.47	0.32	1.00	0.26	-0.05	-0.38	0.15	-0.16	0.10	-0.57
tf	0.52	0.38	0.47	0.26	1.00	-0.59	-0.10	0.92	-0.50	0.82	-0.86
cf	-0.40	-0.19	-0.34	-0.05	-0.59	1.00	0.02	-0.43	0.12	-0.34	0.51
t	0.15	-0.15	0.01	-0.38	-0.10	0.02	1.00	-0.21	0.30	-0.26	0.02
fx	0.31	0.37	0.32	0.15	0.92	-0.43	-0.21	1.00	-0.40	0.77	-0.72
fy	-0.26	-0.04	-0.19	-0.16	-0.50	0.12	0.30	-0.40	1.00	-0.74	0.36
r	0.42	0.25	0.39	0.10	0.82	-0.34	-0.26	0.77	-0.74	1.00	-0.59
fz	-0.68	-0.60	-0.64	-0.57	-0.86	0.51	0.02	-0.72	0.36	-0.59	1.00

Table 6.1: Correlation between independent variables

Table 6.1 lists the correlation factor relating the interaction between each dependent variable measures during the FSW process. The definitions of each variable can be found in Figure 6.7. As the values approach unity the correlation becomes stronger. A correlation factor of 1.0 shows a direct correlation, as can be seen by observing the intersection between two identical variables. All the dark values relate to significant values with a 95% certainty, the lighter values indicate the certainty lower than the 95% threshold.

Figure 6.8 shows a matrix plot of Table 6.1 representing the correlations graphically. The diagonal axis contains the dependent variables with their associated sample distribution. By finding the intersection point of two variables, the correlation between the two variables may be found. The area to the top-right of the diagonal is the mirror image of the lower-left area across the diagonal axis.

The graph marked A on Figure 6.8, shows the correlation between spindle power and spindle current. It can be seen that as spindle current increases, so does the spindle power in a linear fashion. This example illustrates the use of the matrix plot.

Correlation information may be used when defining rules for an intelligent controller. Actual correlation values may, for example be used as independent variable's certainty factors, which in turn is used to decide if a control variable should by active.

Λ ↘

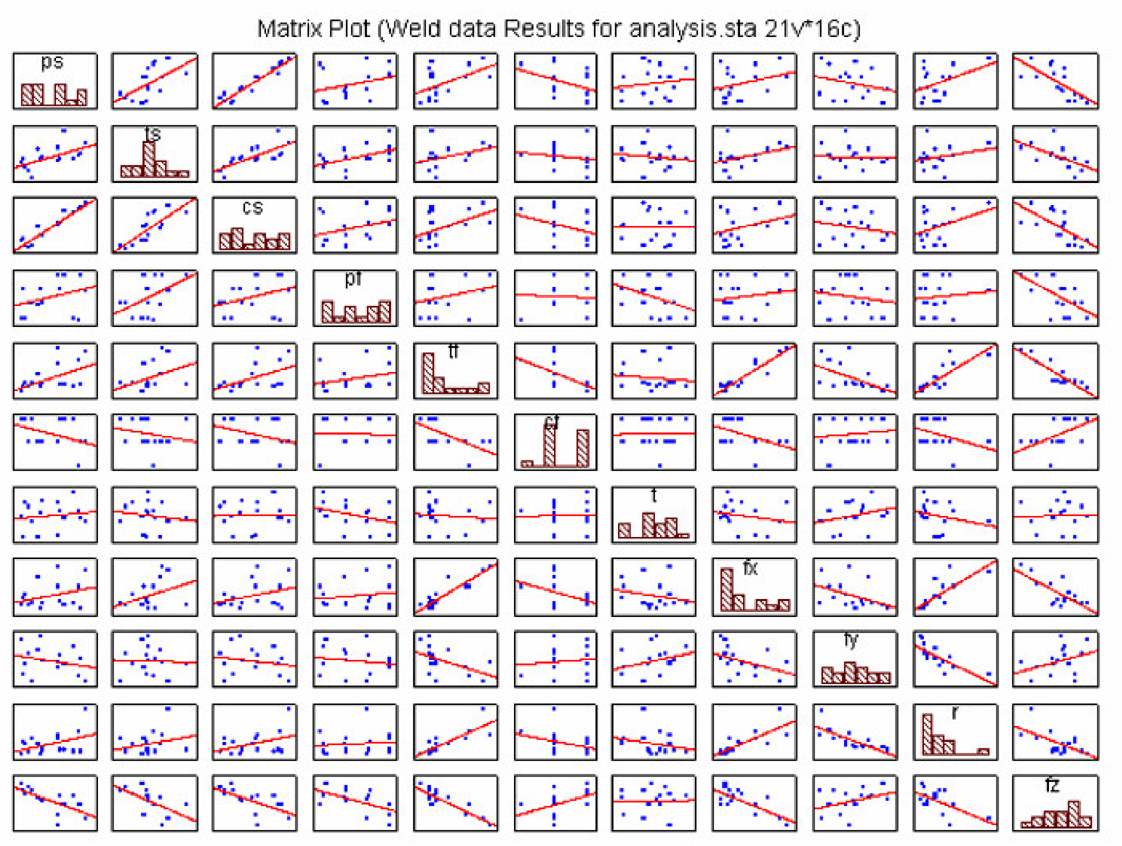
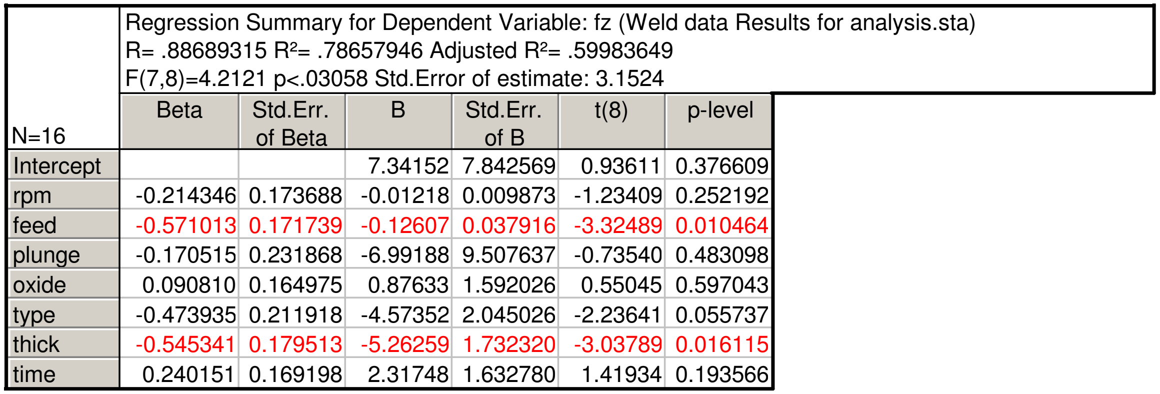


Figure 6.8: Matrix plot showing correlation between independent variables

6.4.2 Regression Analysis

Basic multiple regression analysis was performed on the test data in order to derive linear equations relating the dependent with the independent variables of the FSW process [van Niekerk, 2001]. Appendix D, Table D.2 contains the data used for the multiple regression analysis. The regression summary for the dependent variable fz, used for explanation purposes can be seen in Table 6.2.

The probability-level (p-level) column is the probability that the observed difference between the derived value and actual value (Beta) is solely the result of random error (chance). The lower the probability, the less likely it is that the difference occurred is by chance [Zeelie, 2002]. The dark values on Table 6.2 fall outside the 0.05 (5%) probability level, meaning that the possibility of an effective influence on the independent variable is low.



Regression Summary for Dependent Variable: fz (Weld data Results for analysis.sta)
R= .88689315 R^2= .78657946 Adjusted R^2= .59983649
F(7,8)=4.2121 p<.03058 Std.Error of estimate: 3.1524

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			7.34152	7.842569	0.93611	0.376609
rpm	-0.214346	0.173688	-0.01218	0.009873	-1.23409	0.252192
feed	-0.571013	0.171739	-0.12607	0.037916	-3.32489	0.010464
plunge	-0.170515	0.231868	-6.99188	9.507637	-0.73540	0.483098
oxide	0.090810	0.164975	0.87633	1.592026	0.55045	0.597043
type	-0.473935	0.211918	-4.57352	2.045026	-2.23641	0.055737
thick	-0.545341	0.179513	-5.26259	1.732320	-3.03789	0.016115
time	0.240151	0.169198	2.31748	1.632780	1.41934	0.193566

Table 6.2: Regression summary for dependent variable fz

If more tests are conducted, the p-levels may be slightly reduced. This is due to the small sample taken from the population of possible tests affecting the probability prediction [Pietersen, 2002].

The R^2 value at the top of the table is an indication of how well the actual data fits the derived linear equation. Its value can also give an indication of the percentage the dependent variable is not accounted for by the listed independent variables [van Niekerk, 2001]. The adjusted R^2 value can be regarded as a penalty due to the ratio of independent variables to number of tests conducted. By increasing the sample of tests or reducing the number of independent variables, the adjusted R^2 value will move closer to the unadjusted R^2 value [Pietersen, 2002].

The beta column can be used to create an equation, used to predict the dependent variable (y), when the independent variables (x_i) are specified. The generalized equation, Formula 6.4, is shown below.

$$y = B_1x_1 + B_2x_2 + B_3x_3 + \ldots + B_nx_n + B_0 \quad (6.4)$$

Only beta values whose p-levels have or could have a significant effect on the dependent variable need to be used for the formulas. For the formulas in Table 6.3, some beta values were selected above the usual p-level of 0.05 because the p-level may drop slightly if more tests are conducted. It is important not to discard any variables that may have an important influence when using a larger test sample [Pietersen, 2002].

By applying this general equation to the regression summaries obtained from the multiple regression analysis, the formulas in Table 6.3 are obtained.

Formulas Derived from Multiple Regression Analysis	**Units**

ps = -675.597+(5.016.rpm)+(15.726.feed)+(1372.913.thick)-(525.528.time)	**W**
ts = 33.11654-(0.04039.rpm)+(0.1638.feed)+(8.42024.thick)-(1.32626.time)	**N.m**
cs = 4.03186+(0.03666.feed)+(3.10006.thick)-(1.03066.time)	**A**
pf = 30.44096-(0.01084.rpm)+(5.34155.feed)+(1.79081.oxide)-(4.48396.thick)+(5.08189.time)	**W**
tf = -1.1805+(0.00218.rpm)+(0.00994.feed)+(1.03749.type)+(0.74876.thick)-(0.5996.time)	**N.m**
cf = 2.662-(0.000224.rpm)-(0.1552.plunge)-(0.0456.type)-(0.0412.thick)+(0.0861.time)	**A**
t = 406.775+(0.1371.rpm)-(0.6646.feed)+(106.455.plunge)+(26.8745.thick)	**°C**
fx = -341.862+(3.479.feed)-(121.73.oxide)+(658.081.type)+(404.503.thick)-(273.969.time)	**N**
fy = 589.722-(0.212.rpm)-(211.222.plunge)-(118.356.type)-(30.836.thick)-(23.352.time)	**N**
r = -16.806+(0.0136.rpm)+(9.3476.type)+(4.3665.thick)-(2.856.time)	
fz = 6.34152-(0.01218.rpm)-(0.12606.feed)-(4.57352.type)-(5.26259.thick)+(2.31748.time)	**kN**

Table 6.3: Formulas obtained from multiple regression analysis

The above formulas can be used to predict dependent process variables, if the independent variables are supplied. This is of great benefit for future test/weld planning and perhaps for simulating the welding process.

6.4.3 Analysis of Surface Plots for FSW Process Control

The following section will present a series of surface plots, which graphically relate the effect on a dependent variable of altering an independent variable. Directly measured process variables, such as tool temperature, torque, vertical and horizontal forces will be discussed. These process variables will be used in the development of for intelligent control modules used to solve specific problems that occur during the welding process. The complete set of statistical data can be found in Appendix F.

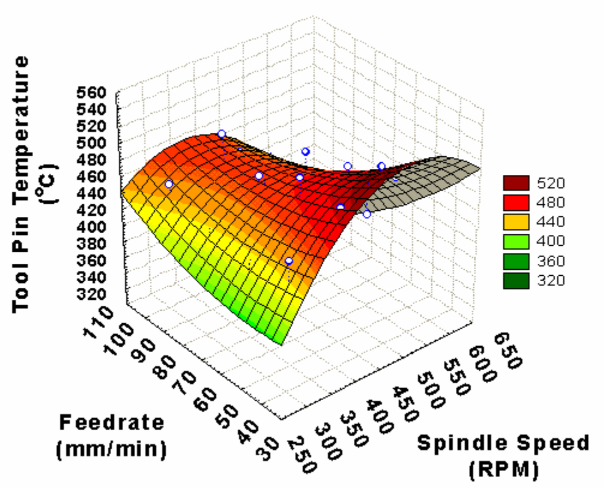


Figure 6.9: Surface plot relating spindle speed, feedrate and tool temperature

The approximate optimum temperature to ensure plasticity of the material to be welded is 0.6 of the material melting point. For aluminum this relates to about 400^{o}C. From the above graph, Figure 6.9, it can be seen that this condition can be reached at either a high RPM or low RPM. In order to maximize the weld speed, while maintaining the optimum temperature for plasticity, the former combination of spindle speed ideally should be used. The effect of feedrate on temperature only becomes prominent at lower feedrates.

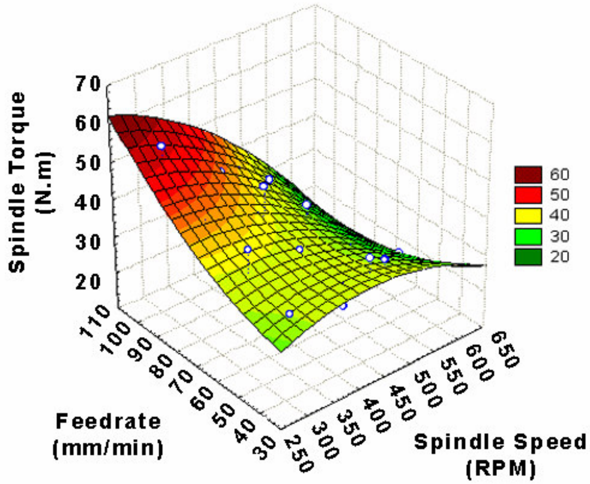


Figure 6.10: Surface plot relating spindle speed, feedrate and spindle torque

From Figure 6.10 it is noted that the highest spindle torques are obtained at high feedrates and low spindle speeds. At higher spindle speeds, the effect of feedrate becomes less of an influence on spindle torque.

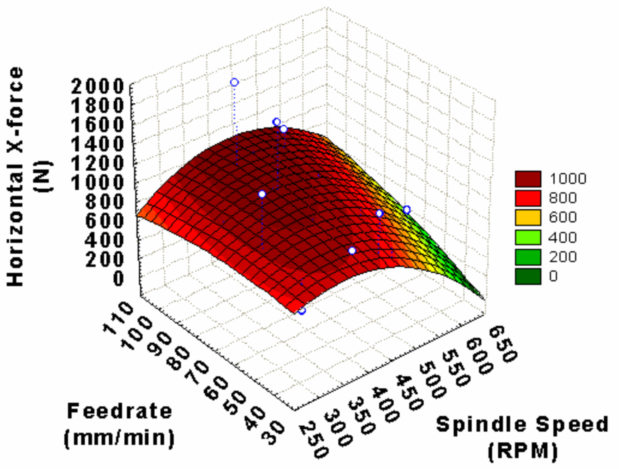


Figure 6.11: Surface plot relating spindle speed, feedrate and maximum horizontal force exerted on the tool

From Figure 6.11, it can be seen that the effect of feedrate and spindle speed on the horizontal force (Fx) exerted on the tool is fairly insignificant. Only at high spindle speeds and low feedrates, does the forces drop off. This could be of vital importance in the design of a portable, hand-held or robotic FSW unit.

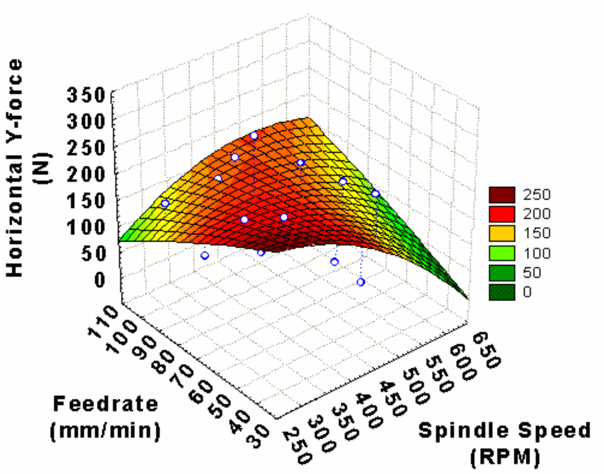


Figure 6.12: Surface plot relating spindle speed, feedrate and maximum vertical force exerted on the tool

Figure 6.12 shows the force measured at 90° to the maximum horizontal force on the FSW tool. In a similar fashion to the maximum force, by using a high spindle speed and low feedrate, the force exerted on the tool may be lowered. If a low spindle speed and high feedrate are used a similar result is obtained.

In Figure 6.13 the dark areas represent low compressive force exerted on the tool. By welding with high spindle speeds or low spindle speeds and high feedrates, this force can be minimized reducing the power requirement of the machine and minimizing the weld time.

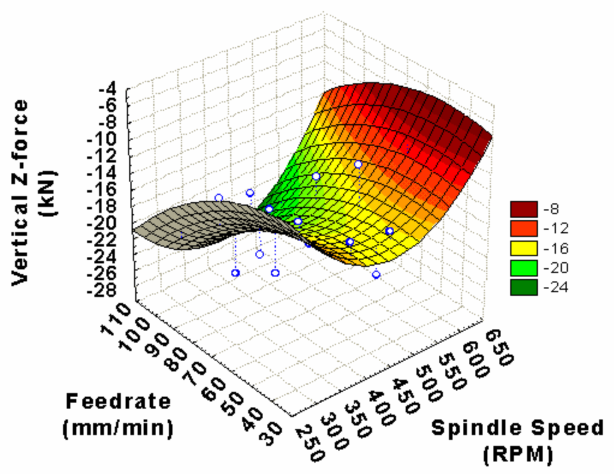


Figure 6.13: Surface plot relating spindle speed, feedrate and compressive vertical force applied on the tool

It is apparent from the graph that high spindle speeds will produce the lowest forces.

6.5 Practical Verification of Statistical Results

The derived equations were used to calculate the required feedrate and spindle speed necessary to obtain a weld temperature of 420°C, with a vertical z-force of 20kN. A profiled tool was used at a plunge depth of 0.2mm, with an initial dwell-period of 16s. The oxide layer of a 6mm aluminum plate will not be removed for this test.

$t = 406.775+(0.1371.rpm)-(0.6646.feed)+(106.455.plunge)+(26.8745.thick)$

$fz = 6.34152-(0.01218.rpm)-(0.12606.feed)-(4.57352.type)-(5.26259.thick)+(2.31748.time)$

By solving the two formulas (from Table 6.2) above simultaneously, the required spindle speed and feed rate can be obtained. This gives a required feedrate of 167 mm/min and a spindle speed of 471 RPM.

A test was conducted using the predicted values. The data was sampled 80mm into the weld, the same position where the initial test data was extracted. This revealed a tool temperature of 479^{o}C, giving an error of 59^{o}C. The expected error of estimate was only 28^{o}C. However, only approximately 30% of the factors influencing the tool temperature are being considered in the analysis. This can be seen by the adjusted R^2 value for the “regression summary for the dependent variable t” in Appendix F.

The vertical force for the tool was also recorded 80mm into the weld and a value of –19.4kN was obtained. This gives a deviation of –0.6kN, which is well within the expected error of estimate for fz being 3.15kN. Figure 6.14 shows the surface finish obtained from the test.

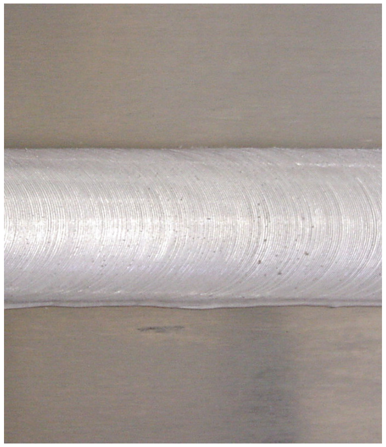


Figure 6.14: Resulting surface finish using calculated feedrate and spindle speed

6.6 Conclusions

The influence of process parameters on quality factors, such as weld strength, fatigue life, residual stress, etc, was not investigated. Further research is necessary into this area, as the effect these process parameters on weld quality is vital to enable online prediction of the weld quality during the welding process. Quality discussions in the conclusion will be restricted to the observed surface finish (voids, roughness, flash, etc) of the welds performed.

By analyzing the experimental results, it seems that there are two possibilities regarding the capacity of the FSW machine:

- **Low capacity machine** - By welding at lower feedrates and high spindle speeds, high temperatures are obtained. There seems to be a correlation between high temperatures (> 500^{o}C) and the formation of post-weld surface cracks, resulting in poor weld quality [Blignault, 2002]. Further investigation is necessary to prove the validity of the above statements. The advantage of using a low capacity machine is that less power is required for a weld, resulting in smaller motors, less rigid machine supports, etc and therefore lowering the cost of the machinery. This may be the key to improving robot mounted or portable friction stir weld heads, which are very advantageous, especially in the automotive industry [Smith, 2000].
- **High capacity machine** – Using high feedrates and high spindle speeds, the temperature during the weld is kept lower. This situation is very beneficial as discussed in Chapter 2, as it results in improved weld quality. Most institutions conducting research into the FSW process have used this method.

To improve the statistical analysis results obtained the following is proposed:

- A larger number tests can be conducted, which could lower the p-values of the independent variables for the regression analysis.
- Fewer independent variables could be used. Results obtained can be used to eliminate variables that do not have a significant effect on the dependent variable under investigation.
- Insignificant independent variables could be replaced with possibly more significant variables as more sensors are added to the machine.

The tests conducted focused on the lower region of possible feedrates and spindle speeds. Further tests should be performed at feedrates and spindle speeds above the sample used in these tests. This will provide a wider “view” of the relationship between the independent and dependent variables, allowing more “insight” into the FSW process.

6.7 Summary

Sensors were mounted to monitor the process from two sources, the instrumented chuck and the inverters. Sensor calibration was performed to ensure reliable data was obtained. Both horizontal and vertical strain gauges were calibrated using a loadcell and the thermocouple was calibrated using a digital thermometer. The Siemens inverters allowed automatic calibration for the motors used.

The instrumented chuck providing a 2D resultant force distribution pattern around the tool was calibrated. Calibration was achieved using a point load and applying the existing theory relating a square or tubular bar in tension or compression. Theoretical and practical results differed due to the variation between theoretical and practical point forces.

Once the system was calibrated, repeatability tests were done to ensure acceptable data would be obtained from the process investigation tests, which were to be done.

A L16 orthogonal analysis test design was selected to minimize the number of tests required and also maximize the information obtained from them. The experiments were then conducted; statistical analysis was performed on the resultant data.

Analysis of variants, regression analysis and surface plots were obtained for analysis. The analysis of variants provided the correlation between each of the independent variables obtained from the FSW test data. Regression analysis provided equations that can be used to predict the values of the independent variables if the dependent variables are supplied for possible design and simulation purposes. The surface plots provide a graphical relationship between the dependent variables spindle speed and feedrate and an independent variable. These plots will be used in the design of rules used to solve specific in –process control problems (Chapter 7).

From studying the results it appears two possibilities regarding an FSW machine’s capacity can be found. Either a low capacity machine can be used with its related disadvantages or a high capacity machine can be used. Low capacity machines may be necessary in physically constrained applications, such a portable or robot-mounted FSW machines.

More tests are however vital to provide further insight into the interrelated factors affecting the FSW process.

The next chapter will discuss fuzzy logic, an intelligent systems technique, and its application to manufacturing systems, the FSW process in particular. The application of the knowledge obtained from practical experience and the analysis of the FSW process surface plots will be used to implement basic intelligent control modules, solving inherent problematic characteristics with the process. Note that the design and integration of all the control modules will lead into a complete advanced control system for the FSW process. This is beyond the scope of this dissertation, however, the work in Chapter 7 forms the basis for such and advanced structure.

Table 6.1: Correlation between independent variables

CHAPTER 7

Intelligent Control for Process Optimization

Intelligent systems are related to artificial intelligence (AI) techniques such as fuzzy logic, neural networks, expert systems and others. These techniques are applied to complex systems whose behavior is difficult to describe mathematically because of the complexity of the physical processes. This complexity can be due to severe non-linearity (forces, temperatures, vibrations, etc) and the variation of process parameters (tool wear, workpiece machinability, differences in tool geometry) [Elias et al., 1998].

Adaptive control allows for the control of a process by online calculation and determination of optimal parameters subject to machining constraints. Such controllers have been used extensively in milling and turning systems to improve the productivity of the manufactured components [Masory et al., 1980].

Machining under optimal conditions can theoretically be achieved by using adaptive controlled machine tools. In practice, adaptive control is essential for improvements regarding optimization of machine costs [Lundholm, 1991]. However adaptive control systems are limited with regard to their operating range for non-linear systems. More flexibility in the control of the process is required. In addition, physical limitations of a machine can be critical during a machining operation owing to lack of power, torque, excessive force, bad surface finish or feed/speed range constraints. To prevent tool breakage, machining conditions are usually very conservative and safe, resulting in an uneconomical situation [Lundholm, 1991]. Intelligent system techniques are able to provide a solution to non-linear process control problems as well as expanded to solve physical limitations of the process with an integrated scheme.

A rule-based fuzzy logic control method, rather than a model-based method, does not need an exact process model and is robust for disturbances, allowing a large uncertainty and variation in the process behavior [Koren, 1997].

Fuzzy logic allows the experience of human operators to be included into the process control design. This has great practical application for industrial control, manufacturing and man-machine communications [Shaw, 1997].

This chapter will briefly discuss the theory behind fuzzy logic for the implementation of a generalized fuzzy logic controller (FLC) to solve basic process control problems intelligently.

2.18 Fuzzy Logic versus Conventional Control Methods for Process Control

Conventional control requires idealized mathematical models of the controlled process. These models can be extremely complex, which results in certain assumptions such as linearity and time invariance. To compensate for these assumptions, tuning of the controller is required. Processes can become so complex in advanced control theory that even though assumptions are used, the models become so complex that their accuracy, operating range and reliability become questionable [Shaw, 1997].

Fuzzy logic provides a rigorous method, capable of expressing imprecise, vague and ill-defined quantities in a systematic manner, by using the experiential knowledge of a trained human operator [Shaw, 1997].

Over 80% of all industrial plant and process controllers are of the PID type, which are not suited to controlling non-linear and time-variant processes. Also, many machine operators are not able to periodically tune the controller sufficiently in order to optimize the control of the process [Shaw, 1997].

Fuzzy logic can be used to control non-linear systems where the operating point can shift over a considerable input range [Shaw, 1997]. An important feature of fuzzy logic control is its ability to handle multi-objective control, where many possibly conflicting requirements can be evaluated to provide a compromise in control strategy [Shaw, 1997]. FSW presents multi-objective control problems and therefore is suited to a fuzzy logic solution.

2.19 Overview of Fuzzy Logic Control

A control system is a closed-loop system that typically controls a machine to achieve a particular desired response, given a number of environmental inputs. A fuzzy control system is a closed-loop system that uses the process of fuzzification to generate fuzzy inputs to an inference engine, which is a knowledge base of actions to take. The inverse process, defuzzification, is also used in a fuzzy control system to create crisp, real values to apply to the machine or process under control. Figure 7.1 shows the major parts of a fuzzy logic control system [Rao et al., 1995].

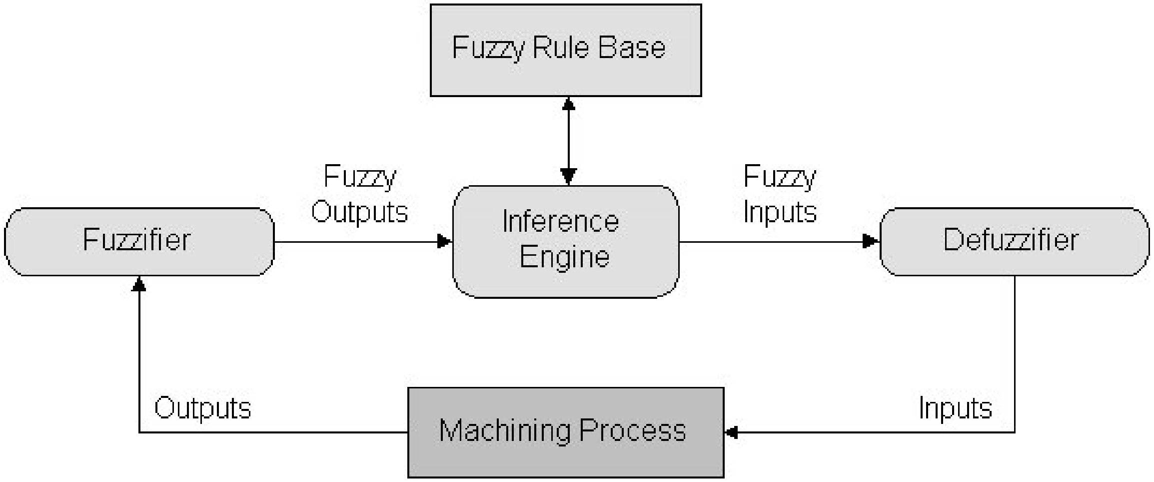


Figure 7.1: Basic diagram of a fuzzy control system [Rao et al., 1995]

The parts in the above diagram are as follows:

- **Process –** The process that is being controlled.
- **Outputs –** These are the measured quantities from the process.
- **Fuzzy outputs –** These are the outputs after being passed through a fuzzifier.
- **Inference engine/Fuzzy rule base –** The fuzzy outputs are tested against fuzzy rules in a fuzzy rule base to determine the output action to take.

- **Fuzzy inputs –** These are the fuzzy actions to perform.
- **Inputs –** These are the crisp values used to control the process.

The fuzzy rule base must be constructed in a manner to yield the desired response from the system. Knowledge about the problem is used to construct the fuzzy rules, allowing a robust control system to be developed in a short period [Rao et al., 1995].

2.20 Components of a fuzzy logic system

The following section will discuss Figure 7.1 in greater detail, by expanding the mentioned areas.

2.20.1 Input Membership Functions

All information in a fuzzy set is described by its membership function (MBF). Figure 7.2 shows the areas of such a membership function.

The core of a membership function is the region of the universe of discourse that is characterized by complete and full membership. The support of a membership function is the region of the universe, which has non-zero membership.

Boundaries of a membership function are the regions of the universe containing elements that have non-zero membership, but not full membership [Ross].

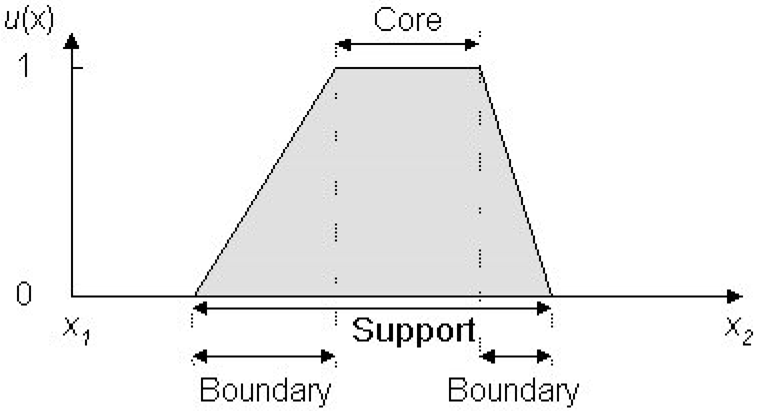


Figure 7.2: Structure of a fuzzy set [Ross]

A linguistic variable is a set of linguistic interpretations describing a range of values (a universe of discourse). The range of values described is termed the base variable. This can be seen in Figure 7.2, where a mathematical function can describe the degree of membership (*u*) of a value (x), to the defined area within the universe of discourse (x_1 -> x_2) [von Altrock, 1995].

Figure 7.3 shows an example of a linguistic variable with three membership functions (low, medium and high). The degree to which a value within the universe of discourse satisfies the linguistic concept of the term of a linguistic variable is called degree of membership (DoM). This is represented by a "*u*" symbol.

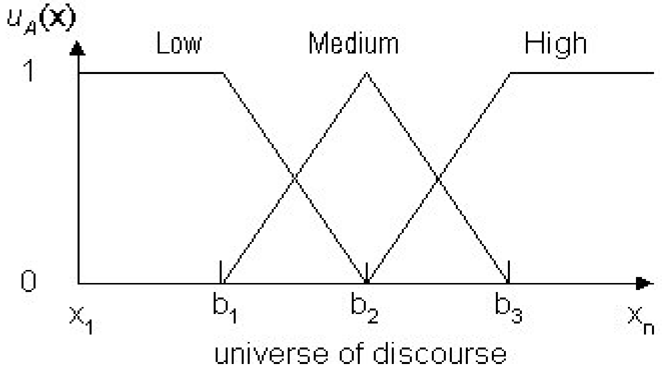


Figure 7.3: Membership functions for a fuzzy variable [Ross]

A = {low, medium, high}

Low, medium and high are the linguistic terms of the linguistic variable A in the above example [Shaw, 1997].

For a continuous variable, the degree of membership is expressed by a mathematical function called the membership function. Membership functions map each value with the universe of discourse to the membership degree in the linguistic terms [von Altrock, 1995].

The formulas below are used to map the degree of membership of a crisp value to a linguistic variable (A). Formula 7.1, 7.2 and 7.3 describe the DoM of x to either the lower, middle or high fuzzy set.

The associated mathematical functions used to describe each of the membership functions in Figure 7.3, would be:

$$\mu_{ALOW}(x)=\begin{cases} 1 & x<b_1 \\ \dfrac{x-b_2}{b_1-b_2} & b_1 \leq x \leq b_2 \\ 0 & x>b_2 \end{cases} \tag{7.1}$$

$$\mu_{AMEDIUM}(x) = \begin{cases} 0 & x < x_1 \\ \dfrac{x-b_1}{b_2-b_1} & b_1 \le x \le b_2 \\ \dfrac{x-b_3}{b_2-b_3} & b_2 \le x \le b_3 \\ 0 & x > b_3 \end{cases} \tag{7.2}$$

$$\mu_{AHIGH}(x) = \begin{cases} 0 & x < b_2 \\ \dfrac{x-b_2}{b_3-b_2} & b_2 \le x \le b_3 \\ 1 & x > b_3 \end{cases} \tag{7.3}$$

There are many different forms of membership functions. Figure 7.4 shows some commonly used standard membership functions.

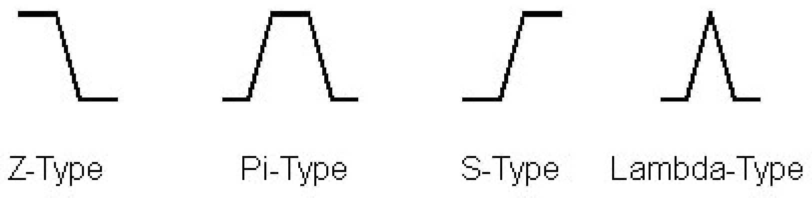
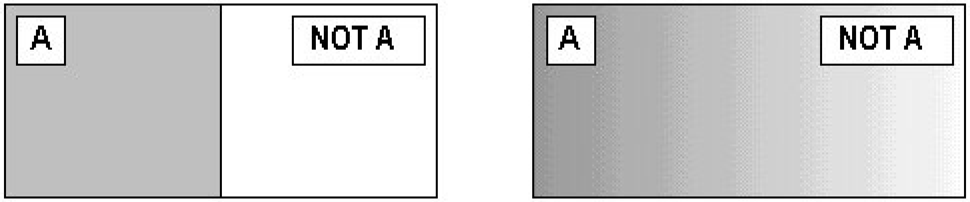


Figure 7.4: Standard membership functions [von Altrock, 1995]

Membership functions do not have to be equal. Their density can be varied to allow greater sensitivity near a set point [Shaw, 1997]. Using standard MBF's are easy to interpret and will suffice for most applications as opposed to other more complex, computation intensive MBF's [von Altrock, 1995]. Each membership function constituting a linguistic variable is described using mathematical formulas, which describe the membership functions association with the universe of discourse.

2.20.2 Fuzzy Sets as a Rule base

Bivalence is predominant in mathematics, however multi-valence exists extensively in physical reality. The fuzzy principle states that everything is a matter of degree and therefore multi-valent logic is used [Shaw, 1997].

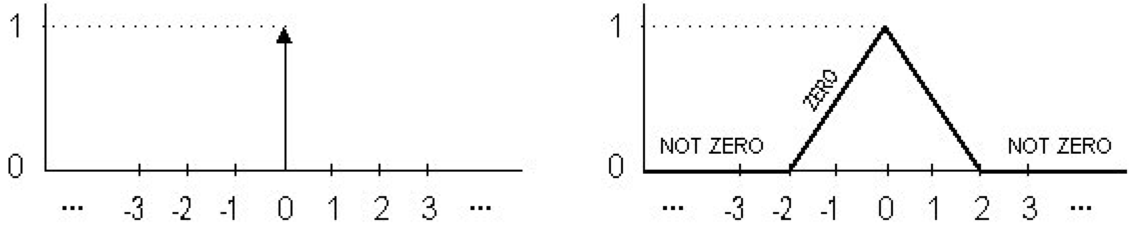


(a) Bivalent logic (b) Multivalent logic

Figure 7.5: Diagram bivalent and multivalent logic [Shaw, 1997]

The Figure 7.5 shows how bivalent logic compares to multivalent logic. Multi-valence has an infinite spectrum of possibilities [Shaw, 1997].

Elements exist in a fuzzy set by a certain degree as opposed to conventional set theory where and element either exists in a set or it does not. Figure 7.6 shows the comparison between a non-fuzzy set and a fuzzy set [Shaw, 1997].



(a) Non-fuzzy set (b) Fuzzy set

Figure 7.6: Comparison between non-fuzzy and fuzzy sets [Shaw, 1997]

Figure 7.6a is a non-fuzzy set as it can be seen that only the element 0 belongs to the set 0. Figure 7.6b shows how the numbers either side of zero belong to the set 0, but with a lesser degree of association. Element 0 is still however, the most prominent element in set 0 [Shaw, 1997].

2.20.3 Principles of Uncertainty

Formulae are used to describe deterministic processes, where no uncertainty exists. However, physical processes virtually always contain uncertainty. This arises from inaccurate measurements, lack of knowledge, vagueness or imprecision. The inability to make the best decision can be attributed to uncertainty, this could lead to a biased or inaccurate decision. Human operators have the ability to overcome uncertainties while controlling processes. They may not know the technical details of the process, however they know what action to take in the event of certain process conditions displayed on instrumentation [van Niekerk, 2001].

According to the classic probability concept, if there are N equally likely possibilities, of which one must occur and N_S is regarded as a "success", then the probability of a "success" is given by the ratio N_S / N, the relative frequency. If the probability of an event occurring is P and the probability that it will not occur is Q, then P + Q = 1 [Shaw, 1997].

2.20.4 The Rule Base

The rules of a fuzzy logic system represent the knowledge of the system. They use linguistic variables as the vocabulary to express the control strategy of a fuzzy logic controller [von Altrock, 1995].

IF <condition> THEN < conclusion>

IF A = a OR B = b THEN C = c

Many fuzzy logic controllers only allow normalized rule bases. A normalized rule does not contain any logical operator such as AND, OR, etc. If the above rule is normalized it becomes two rules [von Altrock, 1995]:

IF A = a THEN C = c

IF B = b THEN C = c

Normalized rules allow complex rule bases to remain understandable, where standard rule bases would become difficult to read [von Altrock, 1995].

A fuzzy system fires each rule in parallel, but to a different degree to generate an output. The degree depends on a weight, which is assigned to each rule. Fuzzy logic systems reason with linguistic fuzzy sets as opposed to bivalent logical operations used in conventional control strategies [Shaw, 1997].

In most decision-making situations in real life, contributing factors often have different importance. Formation of rules in a weighted manner is often encountered in inference engines. Importance weights can be assigned to rules in the following situations [Berkan, Trubatch, 1997]:

- The antecedents involved have different importance in terms of their contribution to the final decision.
- The designers of the rules are different experts with varying levels of expertise [Berkan et al., 1997].

The weight of each rule in a rule base is called the rules degree of support (DoS) and is representative of the certainty of the correctness of the rule [von Altrock, 1995].

The speed of a fuzzy logic system is measured in FLIPS (fuzzy logic inferences per second) [Shaw, 1997].

2.20.5 Inferencing Mechanisms

The purpose of defuzzification is to produce a real crisp value from the linguistic output variable, inferred by the fuzzy controller. There are various types of defuzzification techniques, however the best one for the application must be selected. Two different linguistic meanings of defuzzification have practical importance [Shaw, 1997]:

I. Determine the “best compromise” method:
   - **Center-of-Maximum (CoM)** – determines the most typical value for each term, then computes the best compromise of the fuzzy logic inference result [von Altrock, 1995]. This will balance out the results for each of the terms of the output linguistic variable. To determine the most typical value of each term of the possibility vector, the maximum point for each membership function is found. If the membership function has a maximum interval, then the median of the maximum interval is chosen. Following this, the weights of the

possibility vectors for each membership function of the output linguistic variable are calculated. The crisp compromise value is then determined by balancing these weights [Shaw, 1997]. The following simplified formula can be used to calculate the crisp output value:

$$r = \frac{\sum_i \left[u_{I_i} . \max_X \left(u_{R_i} \right) . \arg \left(\max_X \left(u_{R_i} \right) \right) \right]}{\sum_i u_{I_i}} \qquad (7.4)$$

In the above formula, R is the linguistic variable to the defuzzified, and

u_{R_i} is the membership functions of all linguistic terms I defined within the universe of discourse X ($x \in X$). u_{I_i} is the fuzzy inference result for every term I and r is the crisp output value.

The above formula assumes the maxima of the membership functions are always 1 and only the non-zero inferred values contribute to the crisp outputs [Shaw, 1997].

This method has a fairly high computational efficiency and displays continuity [von Altrock, 1995].

- **Center-of-Area (CoA) / Center-of-Gravity (CoG) –** This method is used where overlapping membership functions are found. The areas of the membership functions constituting the output linguistic variable are cut at the degree of truth of all the fuzzy terms. The areas of the resulting functions of all terms are then superimposed. Finally, the compromise value is obtained by balancing the resulting area.

$$CoG = \frac{\int_a^b u(x) \cdot x dx}{\int_a^b u(x) dx} \quad (7.5)$$

This method has some disadvantages [Shaw, 1997]:

- CoA requires high computational effort as it utilizes numerical integration (possibly 1000x that of CoM).
- Vastly different sized areas can have a distorting influence on defuzzification.

This method lacks computational efficiency, however it promotes continuity [von Altrock, 1995].

II. Determine the "most plausible result" method:

There are situations where the best compromise method will not work and the “most plausible result” method must be used [Shaw, 1997].

- **Mean-of-Maximum (MoM) –** does not use balancing, but selects the typical value of the term that is most valid. The following formula demonstrates this [Shaw, 1997]:

$$r = \frac{\max(u_{I_i})}{l} \quad (7.6)$$

In the above formula $\max(u_{I_i})$ is the inferred fuzzy output term with the highest degree of truth and l is the integer number of such terms [Shaw, 1997].

The computational efficiency of this method is very high, but does not promote continuity [von Altrock, 1995].

2.20.6 Verification and Stability

Stability theory of conventional control completely suffices for fuzzy logic systems. A possible mathematical definition for a fuzzy logic system is a mapping of an input space into an output space with the following properties [von Altrock, 1995]:

- **Deterministic** – The same input condition always results in the same ouput condition.
- **Time Invariant** – Time does not affect the mapping of the transfer function.
- **Nonlinear** – There is no linear relationship between input and output variables [von Altrock, 1995].

This type of system is defined by control theory as a “nonlinear multivariable controller” or “multiband controller”. Therefore, all stability analysis methods that apply to these controllers also apply to fuzzy logic controllers [von Altrock, 1995].

Due to complex nonlinearities, an analytical solution is often impossible. However, process simulation is available using software packages such as Mathlab/Simulink, VisSim and Matrixx, including others that support mathematical stability analysis [von Altrock, 1995].

2.21 Closed Loop Controllers

This research project did not consist of the implementation of a full closed loop controller. However, the basic ideas of such a controller are used to solve problem areas with the FSW process.

The following section will discuss various closed-loop controllers used for industrial control.

2.21.1 Conventional PID control

PID control is used where control over steady state and transient error is necessary. It is the combination of three basic controllers [Franklin, Powell, Emami-Naeini, 1994]:

- **Proportional** – The size of the controller output is proportional to the size of the error [Bolton, 1995].
- **Integral** - The rate of change of the controller output is proportional to the input error signal (*e*) [Bolton, 1995].
- **Derivative** – The change of controller output from the setpoint value is proportional to the rate of change with time of the error signal [Bolton, 1995].

By combining these modes of control, it enables a controller to the produced that has no offset error and reduces the tendency for oscillations. Figure 7.7 shows the basic structure of a PID controlled system.

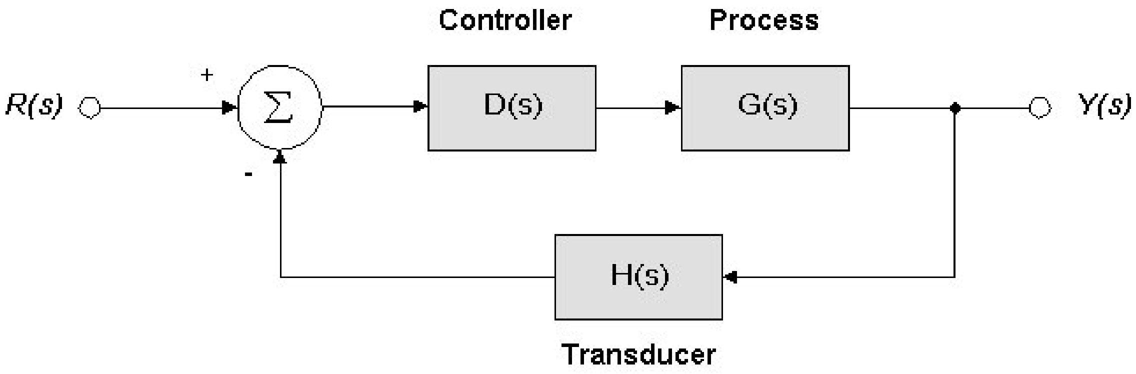


Figure 7.7: General Conventional Feedback Control System [Bolton, 1995]

Formula 7.7 provides the mathematical transfer function for PID control, which is placed in the controller section of the above diagram.

$$D(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right) \qquad (7.7)$$

The PID combination is sometimes able to provide an acceptable degree of error reduction simultaneously with acceptable stability and damping and is used extensively for industrial process control. Controller tuning by an operator is required, for optimal performance, by adjustment of the gains of the three terms (K, T_I, T_D) [Franklin, Powell, Emami-Naeini, 1994].

In process control, PID controllers work very well in keeping single parameters of a process, such a temperatures, flows, and pressures, at a constant level. However, PID controllers cannot control the operating point of the entire process, as this is a multivariable control problem. Past applications have shown the fuzzy logic is successful for such problems [Rao et al., 1995]. Some situations where PID control is not successful display the follow characteristics:

- Non-linear control tasks
- Processes whose characteristics vary over time (gusts of wind, etc)
- Multi-variable control problems
- Processes with unknown or inexact mathematical models

A fuzzy logic controller alone controlling a process, is more the exception than the rule. Usually, the fuzzy logic controller output proceeds into a conventional controller as opposed to driving the actuator directly. The advantage of using the method is that the controller exhibits the following features [von Altrock, 1995]:

- The controller is nonlinear.
- Linguistic rules express the control strategy rather than gains.
- The controller does not need a set point; a decision to increase or decrease the output is only necessary.
- It is easy to extend the controller and scale it to suit new process demands.
- The controller can control the operating point of an entire process, using multiple inputs and multiple outputs.
- A mathematical model does not need to exist for the process.

Some applications require preprocessing of the measured variables of the process. This preprocessing could be in the form of statistical analysis, filtering or Fourier transforms and could be an off-line process as opposed to a closed-loop application [von Altrock, 1995].

2.21.2 Static Fuzzy PID Controller

PID controllers can be implemented using static fuzzy controllers, whose task is to control constant-coefficient unknown dynamic processes, including nonlinear ones. Figure 7.8 shows a PID controller implemented using a static fuzzy controller. The proportional, derivative and integral input terms and the output terms gain setting can be adjusted as with a conventional PID controller [Shaw, 1997]. Therefore this type of controller also requires tuning. However, by starting from the conventional PID controller's settings, much of the tedious tuning of parameters can be alleviated [TATA Consultancy Services].

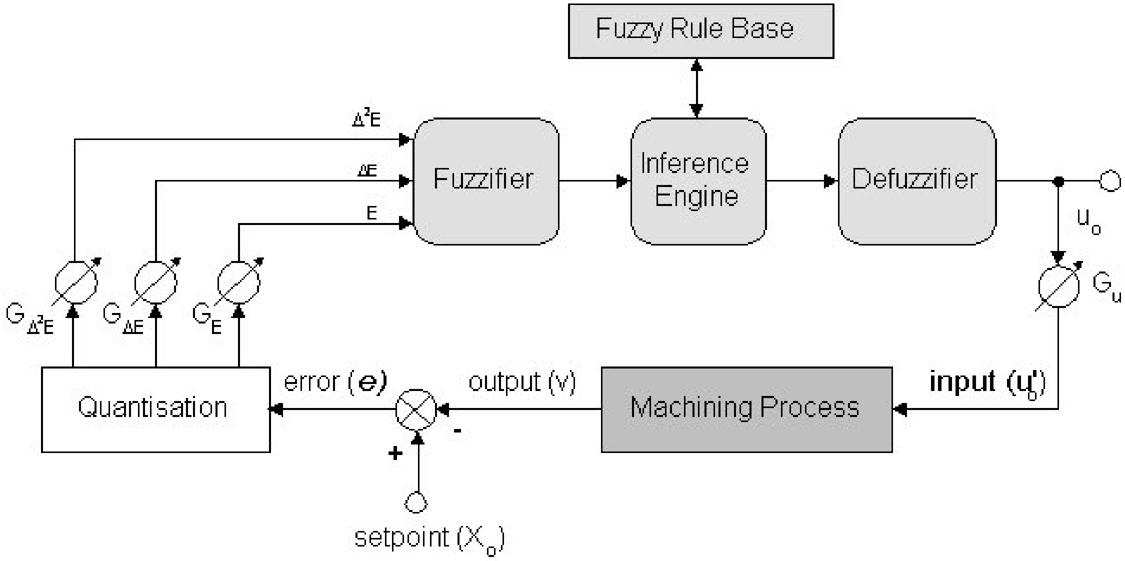


Figure 7.8: PID Controller implementation using fuzzy logic [Shaw, 1997]

The advantages of using a static fuzzy PID are [Jantzen, 1998]:

- Relative simplicity.
- Less overshoot is experienced than with conventional PID controllers.
- Steady state error is removed.
- A smooth control signal is obtained.
- The addition of local non-linearities is possible [TATA Consultancy Services].

However, some disadvantages do exist, such as windup and derivative kick [Jantzen, 1998].

2.21.3 Traditional Multiple-input/Multiple-output Controller

Conventional control theory is difficult to apply to multiple-input/multiple-output (MIMO) systems, especially to obtain a solution quickly. This is due to the interaction of control loops in a multivariable system, each single-input/single-output (SISO) transfer function can have acceptable step response and robustness properties, but the overall control of the system can fail to meet the requirements necessary [Lewis, 1992].

Therefore, MIMO designs require painstaking effort using the approach of closing one loop at a time using graphical techniques. A root locus, for instance, should be plotted for each gain element, taking into account the gains previously selected. This is a trial-and-error procedure that may require multiple iterations, and does not guarantee good results, or even closed-loop stability [Lewis, 1992].

The following diagram shows a MIMO feedback controller, which is based on a mathematical model of the process to be controlled. The system consists of three feedback functions, H_1(s), H_2(s) and H_3(s), which are connected in parallel. G(s) is the transfer function of the physical process and D_1(s) and D_2(s) are two controller transfer functions proving a dual output for control of two separate actuators [Nise, 1995].

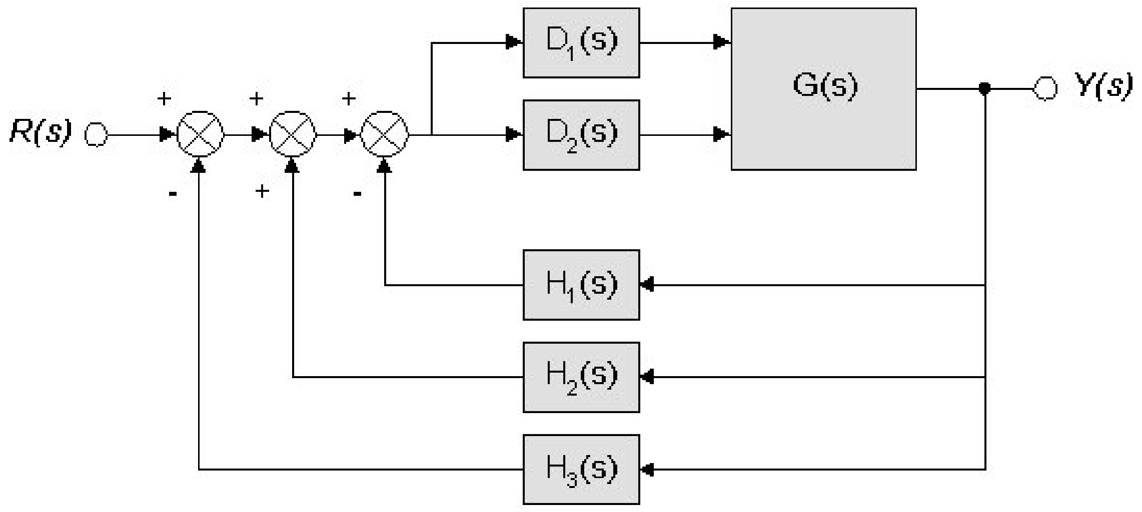


Figure 7.9 General diagram of a traditional MIMO controller [Nise, 1995]

2.21.4 Fuzzy Logic Based MIMO Control

A fuzzy controller is essentially a MIMO type controller, since several variables can be tied together using connectives like “AND“ and “OR” [TATA Consultancy Services]. Figure 7.10 shows an example of a fuzzy logic MIMO controller using the force, temperature and vibration measured from a process to adjust the feedrate and spindle speed. This was done in an attempt to optimize a machining process.

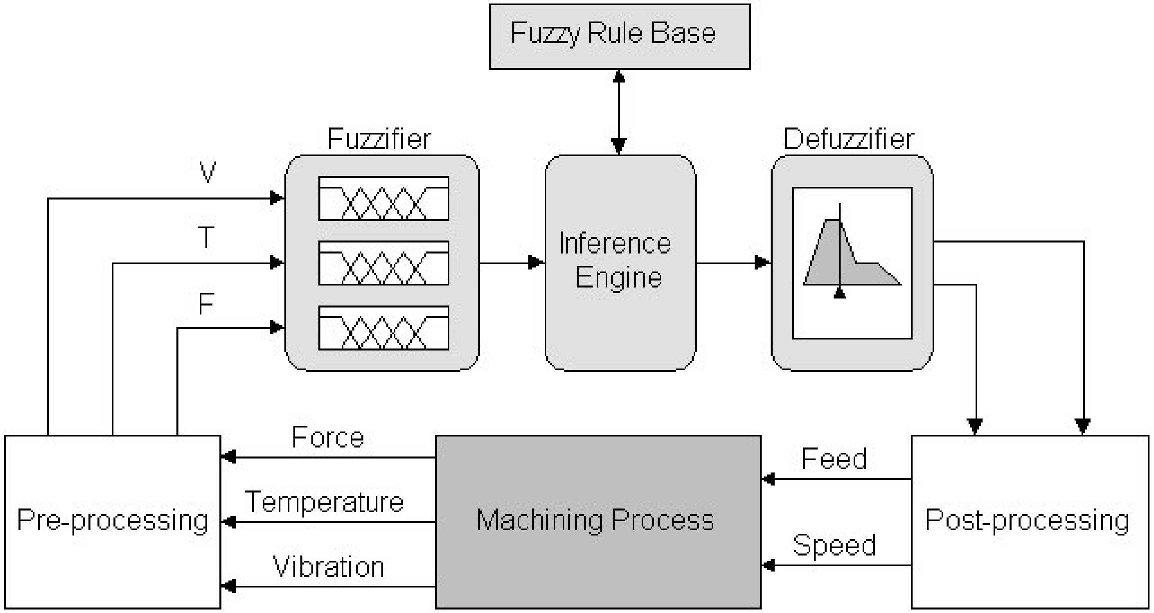


Figure 7.10: An example of a fuzzy logic MIMO controller [von Altrock, 1995]

Table 7.1 illustrates the combination of the measured process variables in order to derive an optimum feedrate and spindle speed for the process. This forms the fuzzy rule base, used during the inference process.

IF force = low THEN feed = high

IF force = low THEN speed = high

IF temperature = high THEN feed = low

IF vibration = high THEN feed = medium

IF vibration = high THEN speed = medium

Table 7.1: Example rules for a fuzzy logic controller

2.22 Fuzzy Logic Control of the FSW Process

This section will discuss the design of a base fuzzy logic controller to optimize and regulate the FSW process. The statistical data, more particularly the surface plots, from the welding tests will be used with practical process experience gained to derive a linguistic rule base. A very basic FLC is proposed as a starting point for the intelligent control of the FSW process. It is designed to improve /solve basic process control problems experienced during the course of the FSW research. In time, as more research is conducted, the knowledge relating to the process will grow. This knowledge can then be incorporated in the controller, expanding its contribution to the control of the FSW process.

Input variables to the FLC controller, measured from the FSW process, are as follows:

Variable	Description	Units	Maximum Value
t	Tool pin temperature	^{o}C	800
ts	Spindle motor torque	N.m	65

Table 7.2: Crisp input variables used as input for the controller

Table 7.2 lists process variables that will be used as inputs to the fuzzy logic controller. The abbreviations listed in the variable column will become the names of the associated linguistic variable after fuzzification, becoming the “vocabulary” of the system in which the rules work [von Altrock, 1995].

Each linguistic variable will contain three terms as a starting point, with the possibility of later adding more terms to improve the controllers response [Rao et al., 1995]. Initially normalized standard membership functions will be used (Z-Type, Lambda-Type and S-Type) as they are the sufficient for most designs, allow for easy interpretation of the design and are most computationally efficient [von Altrock, 1995]. Using this method a set of membership functions for the linguistic variables (Table 7.2) appearing as in Figure 7.3 is obtained. If this is insufficient for the control of the process, other membership functions can be tried at a later stage, if necessary [von Altrock, 1995].

The FLC design will aim to fulfill the following criteria:

- Maintaining the material in its plastic state during the welding process
- Preventing the spindle from stalling during the weld
- Optimizing the processes productivity, by maximizing the feedrate

Using the linguistic variables and the surface plots discussed in Section 6.4.3, the following set of rules have been created to accomplish the proposed goals, as listed in Table 7.3.

The degree of support (DoS) for each rule will be set to 1 initially. If it is found to be too high for a particular rule, then it can be reduced at a later stage [von Altrock, 1995].

#	Rule				Source Surface Plot
1	IF	t = low	THEN	rpm = medium	temperature (t)

2	IF	t = low	THEN	feed = medium	temperature (t)
3	IF	t = high	THEN	rpm = high	temperature (t)
4	IF	t = high	THEN	feed = high	temperature (t)
5	IF	ts = high	THEN	feed = medium	torque (ts)
6	IF	ts = medium	THEN	feed = high	torque (ts)
7	IF	ts = low	THEN	feed = high	torque (ts)
8	IF	ts = medium	THEN	rpm = high	torque (ts)
9	IF	ts = low	THEN	rpm = high	torque (ts)

Table 7.3: Linguistic rules derived from FSW process surface plots

Center-of-Maximum (CoM) defuzzification will be used because it is suited to closed-loop control applications. It has a high computational efficiency and provides a continuous output. Small changes of input values will not produce rapid fluctuation of the output. However, jumps in the output could cause instabilities and oscillations.

Figure 7.11 shows the complete structure of the fuzzy logic controller. The two outputs (speed and feed) will be sent into their respective conventional PID controllers located inside the Siemens MM440 inverters in order to accurately maintain their FLC specified set points. The measured process variables will undergo processing (filtering, averaging, etc) before being fed into the FLC.

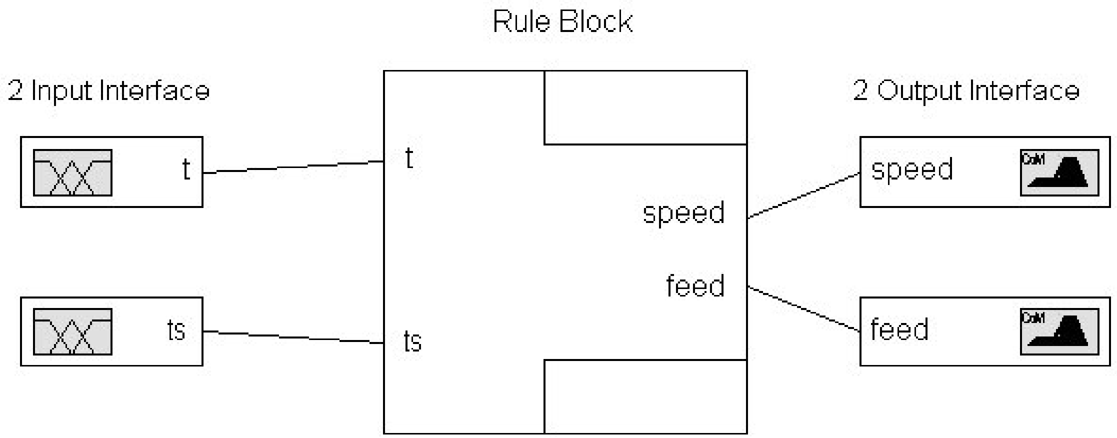


Figure 7.11: Fuzzy logic controller for the FSW process

2.22.1 Maintaining Plasticity

For a successful FSW weld, the material around the tool should be kept within its plastic region. The temperature of the material will determine its level of plasticity. The linguistic variable **t** relates to the temperature at the pin of the tool, which is in contact with the material. The membership functions of this variable will be such that when the material is within its plastic range, the value of linguistic variable **t** will be medium.

Should the temperature increase into the high range, rule 3 and 4 will fire, increasing the feedrate and spindle speed. Figure 6.9 shows this will result in a decrease in temperature.

If the temperature decreases into the low range, rule 1 and 2 will fire, decreasing the feedrate and spindle speed. The surface plot in Figure 6.9 shows that this will result in a temperature increase.

2.22.2 Spindle Stall Prevention

A situation practically encountered for welds greater than 300mm, ultimately results in the spindle motor stalling. This is characterized by a decrease in tool pin temperature, increase in spindle torque and an increase in vertical force. This interaction can be seen from the surface plots in Figures 6.9, 6.10 and 6.13. A possible cause of this event is the tool leaving the “heat island” created during the dwell period and not generating sufficient heat to maintain material plasticity.

Using Table 7.3 shows that as the spindle torque becomes high, rule 5 will fire reducing the feedrate. Figure 6.10 indicates that this will cause the torque to drop. Rule 6 and 8 will now fire, increasing the feedrate and spindle speed. This allows the spindle torque to avoid the high region indicated by the red (dark) area in Figure 6.10. The spindle speed and feedrate once again are able to resume their optimum production points.

2.22.3 Productivity Optimization

It can be seen from the surface plots that there are two possible combinations of feedrate and spindle speed, which can be used to control the temperature. These points are either at a low spindle speed and feedrate or at a high spindle speed and feedrate. The latter combination was selected, as it will optimize the welding speed. Further testing and analysis is necessary to determine the effect of feedrate and spindle speed on the mechanical properties the physical weld.

If the spindle torque is low, then rules 7 and 9 will fire, causing the spindle speed and feedrate to increase. This change will not negatively affect to spindle's torque, as can be seen in Figure 6.10, but it will increase the weld speed.

All the rules are designed to result in the feedrate and speed linguistic variables obtaining high values under normal operating conditions.

2.23 Summary

Intelligent control techniques can be applied to systems as they can describe non-linear and physically complex processes, resulting in the possible optimization of the process.

Fuzzy logic is a form of intelligent control, which is used for processes exhibiting disturbances, large uncertainties and variations in behavior. This is because it does not need an exact process model as with other model-based control methods.

A fuzzy logic system consists of a fuzzifier, inference engine, fuzzy rule base, defuzzifier and the machining process to control.

Fuzzy logic uses fuzzy sets to convert crisp values into linguistic variables. This is done using membership functions, which define certain ranges of values to their associated linguistic variables.

A fuzzy set differs from conventional sets by allowing members of the set to have various degrees of membership as opposed to discrete values.

A fuzzy logic system uses rules to represent the knowledge of the system. Rules are generated using standard basic conditional operators (IF, THEN, =) and linguistic variables to express the control strategy. Rules are fired in parallel, but to different degrees depending on the certainty of the rule, expressed as the certainty factor. The certainty factor is a measure of the correctness of the rule. The inference engine is responsible for the processing of the rules.

The output of the inference engine must be converted back to crisp values by using the defuzzification method best suited to the application. Common methods include: center-of-maximum, center-of-area/center-of-gravity and mean-of-maximum.

Stability theory of conventional controllers can be used for fuzzy logic systems, because a fuzzy logic controller can be defined as a “multiband controller”.

Other forms of closed loop controllers include the conventional PID controller and traditional multiple-input/multiple-output controller. A fuzzy logic controller can be used to implement either of the previously mentioned control strategies, with the added benefits of the fuzzy logic method.

Fuzzy logic is suited to the control of the FSW process, due the many influences on the independent variables by process variables. The data obtained was used to propose a fuzzy logic based controller to ensure material plasticity is maintained, overloading of the spindle is prevented and the weld speed is optimized. Verification and further tuning are however necessary. The controller may be expanded to include variables controlling weld quality. Ultimately the goal of an intelligent controller would be to optimize this quantity regardless of the tool design and material properties. This is proposed as a doctoral topic. As knowledge about the FSW process grows, it can be implemented by expanding the fuzzy logic controller's rule base, resulting in improved welds.

The following chapter will present some conclusions regarding this work done with the FSW process and propose some ideas for future researchers continuing this work.

CHAPTER 8

Conclusions and Future Development

Friction Stir Welding is about 10 years old, therefore it is still a relatively new process. Much research has been conducted into the mechanical properties of the weld, however little documentation could by found on the control and automation of the process. It appears that it is still largely driven by the experience of human operators.

The goal of applying intelligence to this process would be to remove this operator dependence, improving manufacturing productivity and quality.

Conventional CNC machines have limitations due to their closed architecture. They are therefore not suited to the application of intelligent manufacturing. The FSW machine designed and developed during this research project can be used as a platform to investigate intelligent manufacturing, by applying its methodology to the Friction Stir Welding process. The project presents components used in monitoring and control for intelligent machining.

Flexibility was one of the key aspects when designing the machine to allow various intelligent components to be easily implemented and tested. Hardware can be re-configured as required with only minimal software changes. The hierarchical modular system hardware and software design assures re-configurability. Minimizing cost and complexity of the maintenance are the results of this architecture.

Accomplishments of this study can be summarized as follows:

- A Literature survey of the Friction Stir Welding process with a view to its intelligent control was completed.
- Redesign and implementation of the control and monitoring system for a conventional milling machine was completed.
- Monitoring and control software for the FSW process was designed and implemented.

- Sensors were calibrated, experiments performed and statistical data analysis was conducted on the experimental results.
- Intelligent control was investigated and the results of the data analysis used to build linguistic control rules for the FSW process and propose a fuzzy logic controller to implement these rules.

Researchers continuing this investigation, should take note of the following guidlines:

- More information is needed into the FSW process, in particular relating the weld quality or material plasticity to the dependent or independent process variables.
- Vertical force control should be implemented.
- A more powerful machine or thinner material than used in this research should be investigated. The current machines operating ranges are very limited, when welding thicker plate (6mm and 8mm).
- The 2D force plot should be improved, by generating it using the electronics mounted on the chuck. This would remove the effect of transmission delays.
- More research is necessary to relate the 2D force plot to in-process material flow around the tool.

- Many more experiments need to be performed with possibly reduced dependent variables so as to improve the p-levels obtained from the statistical analysis.
- The temperature profile around the tool should be investigated and an attempt should be made to relate it to material plasticity or weld quality.
- An investigation should be conducted to determine the effect on material plasticity and weld quality by the dependent process variables.
- The effect of tool design on the dependent variables should be investigated.
- All of the above points will increase the knowledge into the Friction Stir Welding process, enabling larger, more comprehensive rule bases to be developed, which in turn will allow further optimization of the process.
- This process can benefit from using a self-learning intelligent controller that can possibly relate graph characteristics and ratios to weld quality, allowing the FSW machine to compensate and adapt to new unknown tools and materials. Problems related to the determination of optimum welding parameters each time a new material or tool is used can therefore be alleviated.

It is strongly recommended that this project be continued as it provides an entry point for the continued investigation of the Friction Stir Welding process and re-configurable intelligent manufacturing systems.

References

Access I/O, 2002, Access I/O EDIO-16 PCI Board User Manual.

Access I/O, 2002, Access I/O RS485-4 PCI board user manual.

Altintas Y., Munasinghe W.K., 1996, Modular CNC Design for Intelligent Machining, Part 2: Modular Integration of Sensor Based Milling Process Monitoring and Control Tasks, ASME Journal of Manufacturing Science and Engineering, Vol. 118, No. 4, pp. 514-521.

Altintas Y., Newell N., Ito M., 1996, Modular CNC Design for Intelligent Machining, Part 1: Design of a Hierarchical Motion Control Module for CNC Machine Tools, ASME Journal of Manufacturing Science and Engineering, Vol. 118, No. 4, pp. 506-513.

Andreasen J.L., De Chiffre L., 1993, Automatic Chip-Breaking Detection in Turning by Frequency Analysis of Cutting Force, Annals of the CIRP, Vol. 42/1, pp. 45-48.

Aota K., Okamura H., Masakuni M. and Takai H., 2001, Heat Inputs and mechanical properties in Friction Stir Welding, Hitachi Ltd.

Azouzi R., Guillot M., 1997, On-line Prediction of Surface Finish and Dimension Deviation in Turning using Neural Network Based Sensor Fusion. International Journal of Machine Tools Manufacturing, Elsevier Science Ltd, Vol. 37, No. 9, pp. 120-1217.

Balazinski M., Bellerose M., 1994, Application of Fuzzy Logic Techniques to the Selection of Cutting Parameters in Machining Processes, Fuzzy Sets and Systems, North Holland, March, Vol. 1, No. 3.

Balazinski M., Czogala, Sadowski T, 1993, Control of Metal-Cutting Process Using Neural Fuzzy Controller, IEEE.

Berkan R.C., Trubatch S.L., 1997, Fuzzy Systems Design Principles, IEEE Press.

Blignault C., 2002, Masters Thesis, Port Elizabeth Technikon.

Bolton W., 1995, Mechatronics: Electronic Control Systems in Mechanical Engineering, Longman.

Butenhof D.R., 1997, Programming with POSIX Threads, Addison-Wesley Professional Computing Series.

Donne D.C., Lima E., Wegener J., Pyralla A., Buslaps T., 2001, Investigations on Residual Stresses in Friction Stir Welds, Institute of Material Research, German Aerospace Center, Hahn-Meitner-Institute Berlin, European Synchrotron Radiation Facility.

Edan Y., Pliskin N., 2002, Transfer of Software Engineering Tools from Information Systems to Production Systems. Industrial Engineering and Management Ben-Gurion University of the Negev Israel, August 20.

Elias R., Peres C.R., Aligue A., Ros S., Gonzalez C., Alique J.R., 1998, Toward Intelligent Machining: Hierarchial Fuzzy Control for the End Milling Process, IEEE Transactions on Control Systems Technology Vol. 6. No. 2, March.

Franklin G.F., Powell D.J., Emami-Naeini A., 1994, Feedback Control of Dynamic Systems: Third Edition, Addison-Wesley Publishing Company.

Groover M.P., 1987, Automation, Production Systems and Computer-Integrated Manufacturing, Prentice Hall.

Harbour M.G., Real-time POSIX: An overview, Departmento da Electronica, Universidad de Cantabria, Avda, Los Castros s/n.Spain.

Held G., 1994, Understanding Data Communications: Fourth Edition, Sams Publishing.

Hoffmann K., 1989, An Introduction to Measurements using Strain Gauges, Hottinger Baldwin Messtechnik Gmbh, Darmstadt.

Hordeski M.F., 1992, Control Technology and Personal Computers: System Design and Implementation, Van Nostrand Reinhold.

Hsu P., Fann W., 1996, Fuzzy Adaptive Control of Machining Processes with Self-Learning Algorithm, Transactions of the ASME, Vol. 118.

Husselmann M. (B. Eng Electronics), 2002, Libra Measurement Instruments, interview.

Isermann R., 1993, Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing: Tutorial Paper, Automatica, Vol. 29, No. 4, pp. 815-835, Pergamon Press.

Isermann R., 1998, On Fuzzy Logic Applications for Automatic Control, Supervision, and Fault Diagnosis, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 28, No.2.

James N.M., Hattingh D.G., 2002, Influence of travel speed on fatigue life of Friction Stir Welds in 5083 aluminium.

Jantzen J., 1998, Tuning of Fuzzy PID Controllers, Technical University of Denmark, Department of Automation, 30 September.

Johnson R., 2001, Forces in Friction Stir Welding of Aluminium Alloys, TWI Ltd.

Kallee S., Nicholas D., 2002, Friction Stir Welding at TWI, Paper available from www.twi.co.uk.

Kallee S.W., 2001, Friction Stir Welding – How to Weld Aluminum without melting it, TWI Ltd.

Kopetz H., 1997, Real-time systems: Design principles for Distributed Embedded Applications, Kluwer Academic Publishers.

Koren Y., Control of Machine Tools, The Univeristy of Michigan, Journal of Manufacturing Science and Engineering, November, Vol. 119, Transaction of the ASME.

Krten R., 1999, Getting Started with QNX Neutrino 2: A Guide for Realtime Programmers, Parse Software Devices.

Leine & Linde, 2002, 610/632 Optical Encoder product datasheet, Datasheet available from www.leinelinde.de.

Lewis F.L., 1992, Applied Optimal Control and Estimation, Prentice-Hall.

Libra Measurement Instruments, 2002, TS1000 Torque Sensor Operating Manual.

Lundholm T., 1991, A Flexible Real-Time Adaptive Control System for Turning, Annals of the CIRP, Vol. 40/1/1991.

Macromedia, 2000, Integrating Flash with Web Applications: Overview, Macromedia Flash 5 help.

Masory O., Koren Y., 1980, Adaptive Control System for Turning, Technion-Israel Institute of Technology, Annals of the CIRP Vol. 29/1/1980.

Nicolas Correa, S. A, Correa milling machine technical manual.

Nise N.S., 1995, Control Systems Engineering: Second Edition, The Benjamin/Cummings Publishing Company, Inc.

Owen J.V., 1995, Opening up Controls Architecture, Manufacturing Engineering, November.

Perrin B., 1999, The Art and Science of RS-485, Circuit Cellar Online, July, Paper available at www.circuitcellar.com.

Pietersen J., 2002, Port Elizabeth Technikon Department of Mathematics, 12th October, interview.

QNX Software Systems, 1999. TCP/IP for QNX Product Brief. Brief available from www.qnx.com.

QNX Software Systems, 2002, QNX Neutrino RTOS Product Brief, Brief available from www.qnx.com/products/ps_neutrino/prodbrief.html

QNX software systems, 2002, QNX Programmers Guide, Document available from www.qnx.com.

QNX software systems, 2002, QNX System Architecture Guide, Document available from www.qnx.com.

Rao V.B., Rao H.V., 1995, C++ Neural Networks and Fuzzy Logic: Second Edition, MIS: Press.

Ross T.J., Engineering Applications: International Edition.

Schofield S. (Hewlett Packard), Wright P. (University of California at Berkeley), 1998, Open Architecture Controllers for Machine Tools, Part 1: Design Principles, May, Vol. 120, Journal of Manufacturing Science and Engineering.

Shaw I.S., 1997, The Theory and Application of Fuzzy Logic to Industrial Process Control, Industrial Electronic Technology Research Group, Faculty of Engineering, Rand Afrikaans University.

Shay W.A., 1999, Understanding Data Communications & Networks: Second Edition, Brooks/Cole Publishing Company.

Shinoda T., 2001, Effect of Tool Angle on Metal Flow Phenomenon in Friction Stir Welds, Department of Materials Processing, Nagoya University.

Shinoda T., Tokisue H., Enomoto M., Hori H., Koga S., Kumagai M., Matsumoto K., Okamura H., Tsuchiya K., 2001, Recent Trends of Research and Development of FSW Technology in Japan.

Siemens, 2002, Micromaster 440 Operating Instructions, March.

Siemens, 2002, Micromaster Encoder Module Operation Instructions, February.

Siemens, 2002, Using the USS Protocol with Micromaster MM440, Note available at www.siemens.com.

Smith C.B., 2000, Robotic Friction Stir Welding using a Standard Industrial Robot, Tower Automotive, Milwaukee, WI.

Sommerville, Ian, 2001. Software Engineering:6th edition. Addison-Wesley Publishers. 2001.

TATA Consultancy Services, 2002, Manufacturing Practice: Fuzzy Control.

Thomas M.W., Nicholas E.D., and Smith S.D., 2001, Friction Stir Welding – tool developments, TWI Ltd.

Trang Y., Yen Z., Nian C., 1996, Genetic Synthesis of Fuzzy Logic Controllers in Turning, Fuzzy Sets and Systems, Elseviier Science, pp. 301-310.

van Niekerk T.I., 2001, Monitoring and Diagnosis for Control of an Intelligent Machining Process.

von Altrock C., 1995, Fuzzy Logic & Neurofuzzy Applications Explained, Prentice Hall.

Yellowley and Potter P.R., 1992, The Integration of Processes and Geometry within an Open Architecture Machine Tool Controller, Dept Mechanical Engineering, University of British Columbia, Vancouver, Canada.

Yen Z., Trang Y., Nian C., 1995, A Self-Organizing Neural Fuzzy Logic Controller for Turning Operations, International Journal of Machine Tools Manufacturing, Vol. 35, No. 10, pp. 1363-1374, Elsevier Science Ltd.

Zeelie B, 2002, Experimental Design and Analysis for Scientists and Engineers, Deparment of Chemistry, Port Elizabeth Technikon.

Access I/O, 2002, Access I/O EDIO-16 PCI Board User Manual.

Access I/O, 2002, Access I/O RS485-4 PCI board user manual.

Altintas Y., Munasinghe W.K., 1996, Modular CNC Design for Intelligent Machining, Part 2: Modular Integration of Sensor Based Milling Process Monitoring and Control Tasks, ASME Journal of Manufacturing Science and Engineering, Vol. 118, No. 4, pp. 514-521.

Altintas Y., Newell N., Ito M., 1996, Modular CNC Design for Intelligent Machining, Part 1: Design of a Hierarchical Motion Control Module for CNC Machine Tools, ASME Journal of Manufacturing Science and Engineering, Vol. 118, No. 4, pp. 506-513.

Andreasen J.L., De Chiffre L., 1993, Automatic Chip-Breaking Detection in Turning by Frequency Analysis of Cutting Force, Annals of the CIRP, Vol. 42/1, pp. 45-48.

Aota K., Okamura H., Masakuni M. and Takai H., 2001, Heat Inputs and mechanical properties in Friction Stir Welding, Hitachi Ltd.

Azouzi R., Guillot M., 1997, On-line Prediction of Surface Finish and Dimension Deviation in Turning using Neural Network Based Sensor Fusion. International Journal of Machine Tools Manufacturing, Elsevier Science Ltd, Vol. 37, No. 9, pp. 120-1217.

Balazinski M., Bellerose M., 1994, Application of Fuzzy Logic Techniques to the Selection of Cutting Parameters in Machining Processes, Fuzzy Sets and Systems, North Holland, March, Vol. 1, No. 3.

Balazinski M., Czogala, Sadowski T, 1993, Control of Metal-Cutting Process Using Neural Fuzzy Controller, IEEE.

Berkan R.C., Trubatch S.L., 1997, Fuzzy Systems Design Principles, IEEE Press.

Blignault C., 2002, Masters Thesis, Port Elizabeth Technikon.

Bolton W., 1995, Mechatronics: Electronic Control Systems in Mechanical Engineering, Longman.

Butenhof D.R., 1997, Programming with POSIX Threads, Addison-Wesley Professional Computing Series.

Donne D.C., Lima E., Wegener J., Pyralla A., Buslaps T., 2001, Investigations on Residual Stresses in Friction Stir Welds, Institute of Material Research, German Aerospace Center, Hahn-Meitner-Institute Berlin, European Synchrotron Radiation Facility.

Edan Y., Pliskin N., 2002, Transfer of Software Engineering Tools from Information Systems to Production Systems. Industrial Engineering and Management Ben-Gurion University of the Negev Israel, August 20.

Elias R., Peres C.R., Aligue A., Ros S., Gonzalez C., Alique J.R., 1998, Toward Intelligent Machining: Hierarchial Fuzzy Control for the End Milling Process, IEEE Transactions on Control Systems Technology Vol. 6. No. 2, March.

Franklin G.F., Powell D.J., Emami-Naeini A., 1994, Feedback Control of Dynamic Systems: Third Edition, Addison-Wesley Publishing Company.

Groover M.P., 1987, Automation, Production Systems and Computer-Integrated Manufacturing, Prentice Hall.

Harbour M.G., Real-time POSIX: An overview, Departmento da Electronica, Universidad de Cantabria, Avda, Los Castros s/n.Spain.

Held G., 1994, Understanding Data Communications: Fourth Edition, Sams Publishing.

Hoffmann K., 1989, An Introduction to Measurements using Strain Gauges, Hottinger Baldwin Messtechnik Gmbh, Darmstadt.

Hordeski M.F., 1992, Control Technology and Personal Computers: System Design and Implementation, Van Nostrand Reinhold.

Hsu P., Fann W., 1996, Fuzzy Adaptive Control of Machining Processes with Self-Learning Algorithm, Transactions of the ASME, Vol. 118.

Husselmann M. (B. Eng Electronics), 2002, Libra Measurement Instruments, interview.

Isermann R., 1993, Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing: Tutorial Paper, Automatica, Vol. 29, No. 4, pp. 815-835, Pergamon Press.

Isermann R., 1998, On Fuzzy Logic Applications for Automatic Control, Supervision, and Fault Diagnosis, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 28, No.2.

James N.M., Hattingh D.G., 2002, Influence of travel speed on fatigue life of Friction Stir Welds in 5083 aluminium.

Jantzen J., 1998, Tuning of Fuzzy PID Controllers, Technical University of Denmark, Department of Automation, 30 September.

Johnson R., 2001, Forces in Friction Stir Welding of Aluminium Alloys, TWI Ltd.

Kallee S., Nicholas D., 2002, Friction Stir Welding at TWI, Paper available from www.twi.co.uk.

Kallee S.W., 2001, Friction Stir Welding – How to Weld Aluminum without melting it, TWI Ltd.

Kopetz H., 1997, Real-time systems: Design principles for Distributed Embedded Applications, Kluwer Academic Publishers.

Koren Y., Control of Machine Tools, The Univeristy of Michigan, Journal of Manufacturing Science and Engineering, November, Vol. 119, Transaction of the ASME.

Krten R., 1999, Getting Started with QNX Neutrino 2: A Guide for Realtime Programmers, Parse Software Devices.

Leine & Linde, 2002, 610/632 Optical Encoder product datasheet, Datasheet available from www.leinelinde.de.

Lewis F.L., 1992, Applied Optimal Control and Estimation, Prentice-Hall.

Libra Measurement Instruments, 2002, TS1000 Torque Sensor Operating Manual.

Lundholm T., 1991, A Flexible Real-Time Adaptive Control System for Turning, Annals of the CIRP, Vol. 40/1/1991.

Macromedia, 2000, Integrating Flash with Web Applications: Overview, Macromedia Flash 5 help.

Masory O., Koren Y., 1980, Adaptive Control System for Turning, Technion-Israel Institute of Technology, Annals of the CIRP Vol. 29/1/1980.

Nicolas Correa, S. A, Correa milling machine technical manual.

Nise N.S., 1995, Control Systems Engineering: Second Edition, The Benjamin/Cummings Publishing Company, Inc.

Owen J.V., 1995, Opening up Controls Architecture, Manufacturing Engineering, November.

Perrin B., 1999, The Art and Science of RS-485, Circuit Cellar Online, July, Paper available at www.circuitcellar.com.

Pietersen J., 2002, Port Elizabeth Technikon Department of Mathematics, 12th October, interview.

QNX Software Systems, 1999. TCP/IP for QNX Product Brief. Brief available from www.qnx.com.

QNX Software Systems, 2002, QNX Neutrino RTOS Product Brief, Brief available from www.qnx.com/products/ps_neutrino/prodbrief.html

QNX software systems, 2002, QNX Programmers Guide, Document available from www.qnx.com.

QNX software systems, 2002, QNX System Architecture Guide, Document available from www.qnx.com.

Rao V.B., Rao H.V., 1995, C++ Neural Networks and Fuzzy Logic: Second Edition, MIS: Press.

Ross T.J., Engineering Applications: International Edition.

Schofield S. (Hewlett Packard), Wright P. (University of California at Berkeley), 1998, Open Architecture Controllers for Machine Tools, Part 1: Design Principles, May, Vol. 120, Journal of Manufacturing Science and Engineering.

Shaw I.S., 1997, The Theory and Application of Fuzzy Logic to Industrial Process Control, Industrial Electronic Technology Research Group, Faculty of Engineering, Rand Afrikaans University.

Shay W.A., 1999, Understanding Data Communications & Networks: Second Edition, Brooks/Cole Publishing Company.

Shinoda T., 2001, Effect of Tool Angle on Metal Flow Phenomenon in Friction Stir Welds, Department of Materials Processing, Nagoya University.

Shinoda T., Tokisue H., Enomoto M., Hori H., Koga S., Kumagai M., Matsumoto K., Okamura H., Tsuchiya K., 2001, Recent Trends of Research and Development of FSW Technology in Japan.

Siemens, 2002, Micromaster 440 Operating Instructions, March.

Siemens, 2002, Micromaster Encoder Module Operation Instructions, February.

Siemens, 2002, Using the USS Protocol with Micromaster MM440, Note available at www.siemens.com.

Smith C.B., 2000, Robotic Friction Stir Welding using a Standard Industrial Robot, Tower Automotive, Milwaukee, WI.

Sommerville, Ian, 2001. Software Engineering:6th edition. Addison-Wesley Publishers. 2001.

TATA Consultancy Services, 2002, Manufacturing Practice: Fuzzy Control.

Thomas M.W., Nicholas E.D., and Smith S.D., 2001, Friction Stir Welding – tool developments, TWI Ltd.

Trang Y., Yen Z., Nian C., 1996, Genetic Synthesis of Fuzzy Logic Controllers in Turning, Fuzzy Sets and Systems, Elseviier Science, pp. 301-310.

van Niekerk T.I., 2001, Monitoring and Diagnosis for Control of an Intelligent Machining Process.

von Altrock C., 1995, Fuzzy Logic & Neurofuzzy Applications Explained, Prentice Hall.

Yellowley and Potter P.R., 1992, The Integration of Processes and Geometry within an Open Architecture Machine Tool Controller, Dept Mechanical Engineering, University of British Columbia, Vancouver, Canada.

Yen Z., Trang Y., Nian C., 1995, A Self-Organizing Neural Fuzzy Logic Controller for Turning Operations, International Journal of Machine Tools Manufacturing, Vol. 35, No. 10, pp. 1363-1374, Elsevier Science Ltd.

Zeelie B, 2002, Experimental Design and Analysis for Scientists and Engineers, Deparment of Chemistry, Port Elizabeth Technikon.

Appendix B

B.1: Spindle Motor Inverter Configuration

Parameter	Parameter Description	Value	Explaination
P0003	User access level	3	Expert access level
P0010	Start quick commisioning	1	Ready to run
P0100	Operation for Europe/N.America	0	Power in KW; default 50Hz
P0205	Inverter application	0	Constant torque
P0300	Select motor type	1	Asyncronous motor type
P0304	Rated motor voltage	380V	As specified on motor name plate
P0306	Rated motor current	11.7A	As specified on motor name plate
P0307	Rated motor power	5.5kW	As specified on motor name plate
P0308	Rated motor cosPhi	0.85	As specified on motor name plate
P0310	Rated motor frequency	50Hz	As specified on motor name plate
P0311	Rated motor speed	1450RPM	As specified on motor name plate
P0320	Motor magnetizing current	0	Automatic calculation
P0335	Motor cooling	1	Force-cooled
P0700	Selection of command source	1	BOP/AOP

P1000	Selection of frequence setpoint	1	Motor potentiometer setpoint
P1080	Minimum motor frequency	0Hz	
P1082	Maximum motor frequency	50Hz	To prevent damage motor and machine (above 60Hz machine bearings make noises)
P1120	Ramp-up Time	2s	Sufficient to prevent inertia related problems
P1121	Ramp-down Time	2s	
P1135	OFF3 ramp-down time	2s	If process is aborted
P1300	Control mode	21	Vector control with sensor (gives best torque holding results)
P1500	Selection of torque setpoint	0	No main setpoint
P1910	Select motor data identification	1	Identification of all parameters with parameter change
P3900	End Quick Commissioning	1	End Quick Commissioning with motor calculation and factory reset
P0290	Inverter overload reaction	1	Trip (F0004)
P0400	Select encoder Type	2	Quadrature encoder without zero

			pulse
P0408	Encoder pulses per revolution	512	As per encoder specification
P0491	Reaction on speed signal loss	0	Do not change to SLVC
P0492	Allowed speed difference	10Hz	Under heavy loading conditions
P0500	Technological application	0	Constant torque
P0531	Unit Conversion	0	No conversion of units
P0601	Motor temperature sensor	1	PTC thermistor
P1000	Selection of frequency setpoint	5	USS on a COM link
P1134	Rounding type	0	Continuous smoothing (spindle)
P1800	Pulse frequence	4kHz	To ensure full output current is obtained
P1960	Speed control optimisation	1	Enable autotuning of controller
P2009	USS normalization	1	Enabled (setpoint interpreted as absolute value)
P2010	USS baudrate	9	57600bps
P2011	USS address	8	Selected as default for software
P2012	USS PZD length	2	2 words selected as default for

			software
P2013	USS PKW length	4	4 words selected as default for software
P2014	USS telegram off time	2000	2s timeout before trip selected for saftty

B.2: Feed Motor Inverter Configuration

Parameter	Parameter Description	Value	Explanation
P0003	User access level	3	Expert access level
P0010	Start quick commissioning	1	Ready to run
P0100	Operation for Europe/N.America	0	Power in KW; F default 50Hz
P0205	Inverter application	0	Constant torque
P0300	Select motor type	1	Asynchronous motor type
P0304	Rated motor voltage	380V	As specified on motor name plate
P0306	Rated motor current	3.7A	As specified on motor name plate
P0307	Rated motor power	1.5kW	As specified on motor name plate

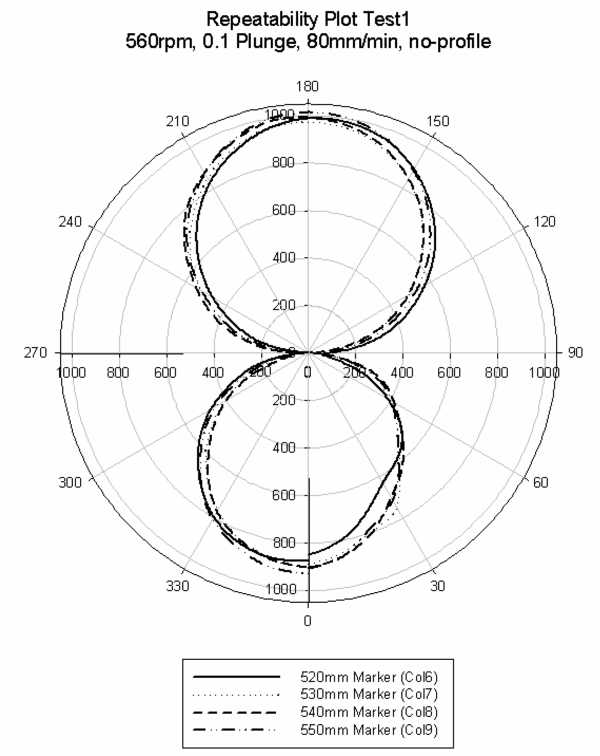
P0309	Rated motor efficiency	0.76	As specified on motor name plate
P0310	Rated motor frequency	50Hz	As specified on motor name plate
P0311	Rated motor speed	1420RPM	As specified on motor name plate
P0320	Motor magnetizing current	0	Automatic calculation
P0335	Motor cooling	1	Force-cooled
P0700	Selection of command source	1	BOP/AOP
P1000	Selection of frequency setpoint	1	Motor potentiometer setpoint
P1080	Minimum motor frequency	0Hz	
P1082	Maximum motor frequency	50Hz	To prevent damage motor and machine (above 60Hz machine bearings make noises)
P1120	Ramp-up Time	2s	Sufficient to prevent inertia related problems
P1121	Ramp-down Time	1s	The ramp down is controlled by software
P1135	OFF3 ramp-down time	2s	If the weld is aborted
P1300	Control mode	21	Vector control with sensor (gives best torque holding results)

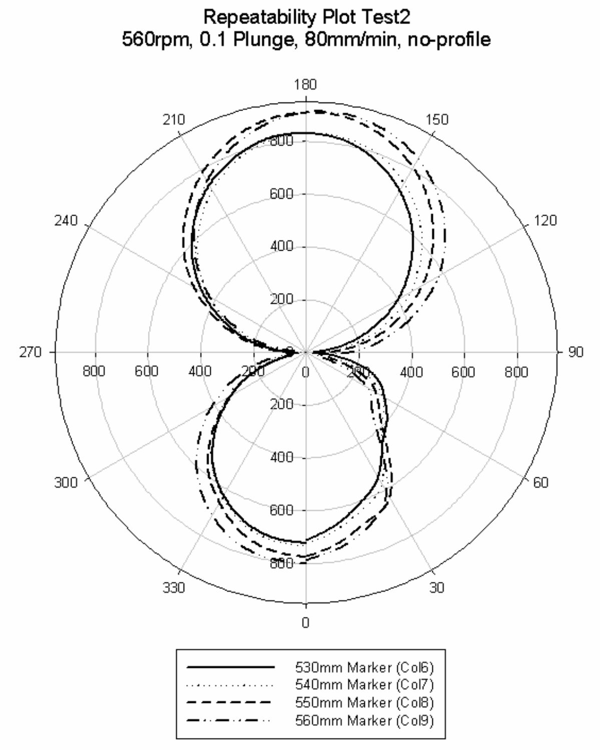
P1500	Selection of torque setpoint	0	No main setpoint
P1910	Select motor data identification	1	Identification of all parameters with parameter change
P3900	End Quick Commissioning	1	End Quick Commissioning with motor calculation and factory reset
P0290	Inverter overload reaction	1	Trip (F0004)
P0400	Select encoder Type	2	Quadrature encoder without zero pulse
P0408	Encoder pulses per revolution	512	As per encoder specification
P0491	Reaction on speed signal loss	0	Do not change to SLVC
P0492	Allowed speed difference	10Hz	Under heavy loading conditions
P0500	Technological application	3	Simple positioning
P0531	Unit Conversion	0	No conversion of units
P0601	Motor temperature sensor	1	PTC thermistor
P1000	Selection of frequency setpoint	5	USS on a COM link
P1134	Rounding type	1	Discontinuous smoothing (for feed calculation accuracy)

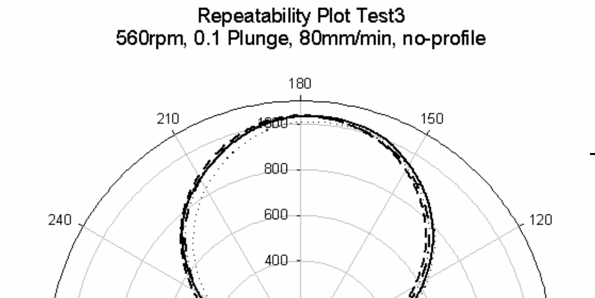
P1800	Pulse frequency	4kHz	To ensure full output current is obtained
P1960	Speed control optimization	1	Enable autotuning of controller
P2009	USS normalization	1	Enabled (setpoint interpreted as absolute value)
P2010	USS baudrate	9	57600bps
P2011	USS address	8	Selected as default for software
P2012	USS PZD length	2	2 words selected as default for software
P2013	USS PKW length	4	4 words selected as default for software
P2014	USS telegram off time	2000	2s timeout before trip selected for safety
P2106	BI: External fault	r2829	Specify BICO source of inverted digital input
P2480	Position mode	1	Open loop positioning
P2482	No. of shaft turns = 1 Unit	1	Software required shaft turns
P2800	Enable free function blocks (FFBs)	1	Enable

P2801[9]	Activate FFBs	1	Assign NOT1 to level 1
P2828	BI:NOT 1	r0701	Input via digital input 1 for bed hard stops

Appendix C


Repeatability Plot Test1
560rpm, 0.1 Plunge, 80mm/min, no-profile


Repeatability Plot Test2
560rpm, 0.1 Plunge, 80mm/min, no-profile


Repeatability Plot Test3
560rpm, 0.1 Plunge, 80mm/min, no-profile

Appendix D

D.1: Design of Experiments Used for Statistical Analysis

Welding Parameters					Process conditions			
Test No	N (rpm)	F (mm/min)	D (mm)		Oxide Layer (P)	Tool type (Y)	Material thickness (s)	Dwell time (t)
1	300	40	0.1		P1	Y1	s1	t1
2	600	100	0.4		P1	Y1	s1	t1
3	400	60	0.4		P1	Y1	s2	t2
4	500	80	0.1		P1	Y1	s2	t2
5	300	80	0.4		P3	Y2	s1	t2
6	600	60	0.1		P3	Y2	s1	t2
7	400	100	0.1		P3	Y2	s2	t1
8	500	40	0.4		P3	Y2	s2	t1
9	500	100	0.2		P1	Y2	s1	t2
10	400	40	0.3		P1	Y2	s1	t2
11	600	80	0.3		P1	Y2	s2	t1
12	300	60	0.2		P1	Y2	s2	t1
13	500	60	0.3		P3	Y1	s1	t1
14	400	80	0.2		P3	Y1	s1	t1
15	600	40	0.2		P3	Y1	s2	t2
16	300	100	0.3		P3	Y1	s2	t2

D.3: Experimental Results for Tests T1 to T16

EXPERIMENTAL RESULTS TEST 1 (T1)

Feed X Position	310.0	320.0	330.0	340.0	350.0	360.0	370.0	380.0	390.0	400.0
Spindle Speed	495.6	300.1	299.3	300.2	300.0	299.9	300.1	299.7	300.0	299.8
Spindle Power	1172.9	2352.1	2340.1	2318.1	2353.9	2380.5	2432.2	2470.9	2460.2	2447.8
Spindle Torque	11.0	32.9	33.1	32.7	32.9	33.4	33.9	34.4	34.1	34.1
Spindle Current	3.5	8.3	8.2	8.2	8.3	8.3	8.5	8.6	8.6	8.5
Feedrate	7.5	38.5	38.4	38.4	38.4	38.4	38.4	38.4	38.4	33.4
Feed Power	94.4	241.3	240.4	240.7	240.1	239.5	240.0	238.5	240.4	216.1
Feed Torque	1.2	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.4
Feed Current	1.4	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Tool Temp.	51.9	235.4	279.0	309.0	330.6	347.2	361.3	373.2	384.2	394.1
Tool Fx/Fy	1.8	11.0	35.4	53.3	44.8	49.5	2.1	175.8	145.4	18.5
Tool Fz	-1.5	-7.9	-8.4	-8.8	-9.0	-9.2	-9.7	-10.0	-10.2	-10.3
0 Degrees	139.1	382.0	371.6	349.9	344.1	351.3	553.8	360.6	345.3	350.3
10 Degrees	135.8	363.8	352.1	328.2	324.5	331.6	526.8	335.4	327.6	328.0
20 Degrees	127.7	314.9	297.5	274.4	270.5	273.9	491.9	275.5	271.4	276.8
30 Degrees	118.3	256.8	235.5	212.3	208.6	209.5	454.4	210.3	210.2	219.1
40 Degrees	109.1	198.9	174.0	150.0	147.8	147.1	410.0	148.7	149.4	164.1
50 Degrees	100.6	142.8	115.1	90.7	89.1	88.5	370.0	91.9	94.2	111.8
60 Degrees	92.9	91.2	61.8	38.1	35.6	36.6	336.8	41.1	44.9	65.3
70 Degrees	86.6	44.0	13.8	8.8	10.6	9.7	276.5	2.8	3.3	25.6
80 Degrees	81.7	2.7	27.2	49.4	50.5	50.1	212.9	39.5	31.0	6.1
90 Degrees	78.2	33.5	61.8	83.3	83.3	83.9	151.5	68.4	58.0	30.9
100 Degrees	76.5	62.2	88.2	108.8	108.0	107.7	94.2	88.2	76.0	46.6
110 Degrees	76.3	82.4	106.1	123.5	121.5	121.0	42.9	95.8	82.1	51.6
120 Degrees	77.9	91.6	112.7	127.2	123.6	121.9	1.4	92.1	78.0	45.9
130 Degrees	81.3	90.0	107.6	118.9	114.3	111.0	39.4	77.6	62.5	30.1
140 Degrees	86.2	76.9	90.6	100.4	94.4	88.9	70.3	51.5	36.8	4.7
150 Degrees	92.3	53.6	63.2	70.3	63.9	56.0	92.0	16.0	2.1	28.6
160 Degrees	99.5	20.3	25.8	30.2	24.2	14.5	102.3	28.8	42.2	70.2
170 Degrees	107.2	21.0	19.6	17.4	23.7	34.7	101.0	81.1	93.9	119.6
180 Degrees	115.7	68.7	72.8	72.3	79.5	90.4	88.0	138.4	149.3	171.8
190 Degrees	124.3	120.9	129.8	130.4	137.5	149.6	64.4	197.8	207.9	227.4
200 Degrees	132.9	175.6	188.0	190.3	197.6	210.8	31.3	259.7	267.3	283.3

210 Degrees	141.0	233.1	246.3	250.5	257.1	271.4	11.2	318.5	326.1	338.8
220 Degrees	148.7	289.6	303.2	308.7	315.7	330.9	61.6	376.4	381.8	390.5
230 Degrees	155.7	343.6	358.5	363.7	370.6	387.1	117.9	429.6	433.5	438.5
240 Degrees	161.9	393.6	409.7	414.1	422.2	438.5	174.9	478.6	478.9	480.5
250 Degrees	167.1	438.4	456.5	458.2	466.9	483.5	234.2	520.6	517.4	516.2
260 Degrees	171.2	476.2	495.3	495.4	503.7	519.8	292.9	553.4	547.6	542.8
270 Degrees	174.1	506.5	525.7	523.5	531.2	547.0	351.8	577.1	569.1	561.6
280 Degrees	175.8	528.0	546.3	542.8	549.9	563.9	406.7	590.9	581.0	571.7
290 Degrees	176.2	540.4	557.1	552.1	558.8	571.7	456.7	595.5	584.2	574.4
300 Degrees	175.3	544.9	560.0	553.2	560.0	571.4	499.4	592.2	580.2	569.8
310 Degrees	173.4	541.7	555.2	545.6	552.2	562.5	534.4	579.7	567.6	557.3
320 Degrees	170.1	530.0	540.9	528.4	535.0	543.6	560.0	558.4	545.8	536.6
330 Degrees	165.5	510.6	518.7	502.7	508.8	516.0	575.9	528.6	516.4	508.0
340 Degrees	160.0	482.6	487.9	469.1	475.0	480.4	582.2	492.3	479.9	473.5
350 Degrees	153.9	449.8	453.5	434.1	438.2	441.0	581.4	452.1	440.1	436.0

EXPERIMENTAL RESULTS TEST 2 (T2)

Feed X Position	440.0	450.0	460.0	470.0	480.0	490.0	500.0	510.0	520.0	530.0
Spindle Speed	537.5	495.1	484.1	477.6	472.2	457.4	443.5	427.3	421.7	417.6
Spindle Power	1812.7	5057.4	5072.7	5118.7	5166.2	5229.0	5279.8	5402.9	5438.0	5430.7
Spindle Torque	12.6	35.7	36.7	37.3	37.9	39.3	40.7	42.6	43.1	43.6
Spindle Current	4.8	13.3	13.3	13.4	13.5	13.7	13.9	14.3	14.3	14.3
Feedrate	8.1	96.4	96.6	96.7	96.7	96.7	96.7	96.7	96.7	67.8
Feed Power	100.1	543.8	542.0	543.6	543.7	545.5	544.1	548.2	544.9	409.0
Feed Torque	1.3	2.0	1.8	1.7	1.7	1.7	1.7	1.7	1.7	1.8
Feed Current	1.3	2.5	2.5	2.5	2.5	2.5	2.5	2.6	2.5	2.6
Tool Temp.	49.6	274.3	299.2	319.0	334.6	348.2	360.4	370.4	379.7	388.6
Tool Fx/Fy	1.3	1.8	3.1	6.4	9.2	4.7	2.8	6.2	6.2	11.3
Tool Fz	-2.9	-16.9	-16.9	-17.5	-18.6	-19.4	-19.5	-19.5	-19.6	-19.7
0 Degrees	114.9	528.1	538.2	430.7	446.2	398.0	511.8	444.6	442.7	401.7
10 Degrees	112.8	504.2	516.7	410.7	410.8	371.5	504.9	416.7	417.6	386.3
20 Degrees	109.7	450.1	483.0	364.4	358.8	327.4	484.9	367.3	366.1	346.3
30 Degrees	105.3	356.0	405.5	301.5	301.5	271.0	458.1	314.7	309.6	297.5
40 Degrees	101.4	256.5	318.9	240.2	240.3	208.5	430.0	250.7	246.3	244.0
50 Degrees	97.9	155.1	231.3	179.1	183.3	150.0	405.5	189.3	182.2	187.2
60 Degrees	95.1	53.5	144.8	118.0	128.1	94.5	385.1	126.3	118.4	133.2
70 Degrees	92.9	43.5	58.3	61.8	74.1	40.3	343.3	64.7	58.6	82.5
80 Degrees	91.8	134.1	26.3	11.2	27.7	6.3	291.2	12.1	5.2	36.3
90 Degrees	91.4	217.4	104.4	32.1	13.4	48.2	238.0	35.3	38.5	3.5
100 Degrees	91.8	288.9	174.9	67.3	48.7	84.0	184.0	71.4	71.2	35.4
110 Degrees	93.6	345.0	235.6	95.2	73.1	107.7	130.8	97.2	95.2	57.7
120 Degrees	96.2	383.2	282.9	112.4	88.3	122.4	80.0	111.1	108.4	70.6
130 Degrees	99.4	405.9	315.6	118.2	90.5	125.8	32.8	114.1	109.4	73.4
140 Degrees	103.6	411.2	335.1	114.1	84.2	116.6	9.4	106.1	99.3	65.3
150 Degrees	108.3	396.5	336.6	98.8	68.9	98.6	44.8	86.1	79.7	49.9
160 Degrees	113.6	366.5	323.3	72.7	42.2	70.5	67.5	57.6	53.2	26.6
170 Degrees	119.0	320.6	297.1	39.7	9.9	34.2	81.6	20.9	15.3	4.2
180 Degrees	124.3	259.0	254.8	0.3	29.1	7.2	86.6	22.4	26.7	42.3
190 Degrees	129.5	186.4	201.3	47.0	76.8	54.4	80.4	70.8	73.1	83.7
200 Degrees	134.3	104.3	139.2	97.8	125.7	105.1	64.9	124.1	127.2	128.9
210 Degrees	138.6	11.8	67.9	152.4	178.8	159.9	41.7	180.1	181.1	178.3
220 Degrees	142.1	81.9	10.1	211.2	237.0	216.6	11.6	235.4	236.3	225.4

230 Degrees	144.9	178.0	90.5	269.3	291.5	271.9	25.6	291.6	292.3	273.1
240 Degrees	146.8	277.3	172.2	326.9	344.7	325.7	67.4	345.1	344.7	320.1
250 Degrees	148.0	371.7	255.7	382.3	397.3	376.3	113.6	392.3	395.9	362.8
260 Degrees	148.3	458.7	333.6	431.5	441.5	421.8	161.9	437.4	440.7	403.7
270 Degrees	147.9	540.1	406.0	475.4	481.6	461.3	211.2	474.3	479.8	438.8
280 Degrees	146.9	611.7	475.0	512.9	514.8	492.5	261.0	504.6	513.3	468.2
290 Degrees	145.3	668.0	532.9	540.7	537.8	516.2	310.1	528.6	539.8	492.7
300 Degrees	142.5	709.1	580.9	560.1	554.5	532.2	355.3	542.0	556.3	508.1
310 Degrees	139.4	737.3	617.5	570.8	561.5	537.8	396.0	546.1	563.7	515.3
320 Degrees	135.9	747.3	633.1	570.1	559.2	536.8	432.6	541.2	563.2	516.0
330 Degrees	132.0	739.3	638.1	562.2	548.7	527.1	461.7	528.7	552.5	508.9
340 Degrees	128.0	717.5	627.6	542.3	528.4	504.9	485.2	499.4	528.7	490.2
350 Degrees	124.0	683.1	602.3	522.2	502.5	477.9	502.2	464.6	502.7	471.7

EXPERIMENTAL RESULTS TEST 3 (T3)

Feed X Position	60.0	70.0	80.0	90.0	100.0	110.0	120.0	130.0	140.0	150.0
Spindle Speed	522.2	400.1	399.9	399.9	400.2	399.7	400.3	399.8	399.9	399.7
Spindle Power	1644.6	3557.0	3496.9	3443.9	3452.0	3398.4	3408.4	3438.1	3453.0	3464.0
Spindle Torque	13.8	38.6	37.9	37.6	37.4	37.2	37.1	37.2	37.7	37.4
Spindle Current	4.4	9.9	9.8	9.7	9.7	9.5	9.6	9.6	9.7	9.7
Feedrate	8.7	57.7	57.7	57.7	57.7	57.7	57.6	57.6	57.6	45.9
Feed Power	110.6	336.9	337.6	338.7	338.1	336.9	336.2	334.5	336.2	281.0
Feed Torque	1.4	1.7	1.6	1.6	1.6	1.7	1.8	1.8	1.8	1.9
Feed Current	1.4	2.6	2.6	2.6	2.6	2.6	2.5	2.5	2.5	2.6
Tool Temp.	57.8	291.1	329.7	355.8	375.1	389.9	401.7	411.3	418.9	426.6
Tool Fx/Fy	0.5	3.4	3.3	3.8	3.3	3.6	4.0	2.6	2.4	2.7
Tool Fz	-3.7	-17.2	-17.3	-17.5	-17.5	-17.6	-18.1	-18.3	-18.6	-18.6
0 Degrees	23.5	191.6	179.1	186.8	207.1	232.9	302.8	223.8	222.4	180.1
10 Degrees	22.9	184.4	168.3	180.4	196.6	224.5	312.7	213.8	210.6	172.5
20 Degrees	21.3	147.7	132.8	148.6	158.1	185.8	306.7	162.1	161.2	127.6
30 Degrees	19.2	98.4	84.0	99.6	106.6	132.6	290.4	94.1	88.8	63.4
40 Degrees	17.0	40.6	27.2	41.6	45.4	69.4	260.0	18.7	7.6	8.2
50 Degrees	14.3	23.2	36.2	23.9	23.5	2.1	235.0	63.9	80.5	86.0
60 Degrees	11.0	92.2	106.4	95.5	99.2	80.0	204.0	153.1	173.9	168.7
70 Degrees	8.1	164.5	179.9	172.2	178.8	163.4	157.8	245.2	270.0	254.1
80 Degrees	5.6	238.5	255.3	250.5	259.8	249.3	93.1	338.2	366.1	340.7
90 Degrees	3.4	311.0	329.6	326.9	341.0	332.5	21.1	427.5	458.7	422.3
100 Degrees	1.6	379.7	398.0	397.9	416.7	412.4	58.9	510.4	543.6	496.6
110 Degrees	0.2	442.5	457.4	462.4	485.0	484.5	142.4	581.9	617.5	561.6
120 Degrees	0.6	494.9	507.5	516.2	540.7	545.1	231.1	640.6	678.1	614.2
130 Degrees	0.7	535.4	546.4	560.7	585.1	595.2	320.0	684.2	723.0	653.5
140 Degrees	0.1	563.3	573.9	589.2	615.6	629.9	406.1	713.2	750.2	677.3
150 Degrees	1.4	576.9	587.3	604.3	630.7	650.2	485.9	722.5	757.7	684.9
160 Degrees	3.4	576.9	585.3	604.1	631.0	652.9	557.3	714.9	747.5	675.6
170 Degrees	6.0	562.0	569.4	589.3	614.3	639.6	614.0	688.9	717.3	650.7
180 Degrees	9.0	534.7	539.5	561.3	583.6	611.2	658.1	646.3	671.6	610.2
190 Degrees	12.0	493.9	497.7	519.9	538.8	568.0	684.6	589.9	608.9	557.2
200 Degrees	15.1	443.8	445.9	468.8	483.7	514.0	695.4	521.1	535.4	493.8
210 Degrees	18.1	385.3	385.7	409.0	419.1	449.5	688.6	443.9	453.3	422.1
220 Degrees	20.7	319.2	319.4	340.9	347.0	377.3	664.5	360.7	363.7	345.2

230 Degrees	23.1	250.8	249.7	269.7	272.3	299.8	624.6	274.3	272.7	263.2
240 Degrees	25.0	180.2	178.6	197.3	195.5	218.9	568.4	185.7	177.7	179.7
250 Degrees	26.5	111.0	107.8	125.1	119.2	138.6	502.5	96.6	85.2	97.2
260 Degrees	27.7	43.7	39.3	55.9	45.0	60.1	427.2	12.0	3.5	17.5
270 Degrees	28.5	21.2	24.5	8.8	26.8	13.8	345.6	67.7	87.7	57.4
280 Degrees	29.2	79.6	82.6	69.3	90.6	81.8	261.0	139.4	162.7	124.6
290 Degrees	29.6	132.4	133.3	122.9	147.5	142.9	175.1	200.2	226.8	181.0
300 Degrees	29.9	176.9	176.0	167.9	194.8	194.9	90.8	248.5	277.7	226.9
310 Degrees	30.0	212.1	209.4	204.4	231.8	236.4	9.4	285.7	314.2	260.7
320 Degrees	29.6	236.9	232.4	231.2	258.4	266.4	66.1	310.2	337.4	282.0
330 Degrees	29.1	249.5	242.7	245.3	272.5	284.1	136.0	319.6	346.0	289.0
340 Degrees	28.0	250.8	242.2	247.9	274.0	289.1	195.8	315.1	336.2	281.8
350 Degrees	27.0	242.6	232.7	241.6	265.6	283.4	243.9	301.6	318.4	267.7

EXPERIMENTAL RESULTS TEST 4 (T4)

Feed X Position	200.0	210.0	220.0	230.0	240.0	250.0	260.0	270.0	280.0	290.0
Spindle Speed	534.1	474.0	486.3	496.5	500.3	499.3	495.6	460.3	456.2	461.4
Spindle Power	1781.9	5212.9	5115.5	4331.8	3787.6	3897.8	4325.3	5230.0	5278.0	5237.4
Spindle Torque	13.4	39.8	38.1	35.0	33.0	33.7	35.6	40.7	41.1	40.2
Spindle Current	4.7	13.7	13.4	11.3	9.8	10.1	11.2	13.7	13.8	13.7
Feedrate	8.9	77.2	77.2	77.3	77.2	77.1	77.3	77.1	77.2	57.1
Feed Power	112.2	450.3	447.7	442.7	450.6	442.5	442.9	445.6	446.4	349.9
Feed Torque	1.4	1.6	1.7	1.6	1.8	1.8	1.7	1.8	1.7	1.8
Feed Current	1.4	2.6	2.6	2.5	2.6	2.5	2.6	2.6	2.6	2.6
Tool Temp.	46.7	262.0	301.1	328.6	348.6	364.4	377.4	387.7	397.4	406.6
Tool Fx/Fy	0.4	4.1	3.4	3.4	3.1	2.5	3.6	2.5	2.2	3.0
Tool Fz	-3.0	-15.9	-14.9	-14.5	-14.6	-15.1	-15.6	-16.6	-17.0	-16.8
0 Degrees	7.6	184.4	197.4	219.1	222.0	211.7	302.8	186.3	161.0	142.8
10 Degrees	7.8	171.0	179.7	192.8	193.2	182.5	313.3	155.0	134.9	124.8
20 Degrees	9.4	130.6	136.5	150.9	148.6	123.9	310.7	99.3	77.5	76.3
30 Degrees	12.3	75.1	83.5	89.2	82.8	51.6	299.8	29.6	2.6	12.0
40 Degrees	15.2	11.5	16.5	18.8	7.4	25.8	260.0	54.5	78.9	60.3
50 Degrees	17.8	56.3	61.5	55.5	72.9	110.1	225.0	139.3	164.9	137.6
60 Degrees	20.0	127.7	140.7	136.4	158.5	198.0	191.6	225.7	254.7	213.2
70 Degrees	22.0	205.7	223.9	221.3	245.8	285.3	144.5	319.9	342.2	292.8
80 Degrees	23.7	286.6	311.3	305.9	332.6	373.0	72.8	404.6	426.9	372.6
90 Degrees	25.0	365.5	390.4	388.1	413.8	456.9	6.0	485.2	508.6	447.0
100 Degrees	26.0	438.3	465.6	466.3	491.6	532.4	89.6	563.0	580.9	515.6
110 Degrees	26.6	503.3	534.5	535.9	559.8	597.3	178.9	623.1	636.9	574.7
120 Degrees	26.3	555.9	588.9	592.5	614.9	648.1	270.7	672.1	682.2	620.9
130 Degrees	25.5	597.1	631.4	638.4	658.8	684.6	360.7	706.8	712.6	654.5
140 Degrees	23.9	626.5	661.3	669.0	688.1	706.6	446.2	723.2	725.7	673.3
150 Degrees	21.4	641.4	674.3	683.0	701.1	711.8	526.4	724.9	723.7	678.2
160 Degrees	18.3	639.8	672.8	682.3	697.3	700.2	593.8	709.0	705.4	664.7
170 Degrees	14.7	625.2	654.4	665.4	678.2	673.3	649.2	679.8	669.9	638.0
180 Degrees	10.9	596.7	620.4	634.1	643.0	629.8	691.8	631.5	618.5	598.7
190 Degrees	6.6	551.4	572.9	586.3	592.0	573.3	717.3	569.5	555.0	542.0
200 Degrees	2.6	495.6	511.4	526.7	529.4	507.5	728.5	502.6	481.2	480.4
210 Degrees	0.9	434.0	443.4	458.7	458.0	432.0	719.8	425.6	398.2	413.7
220 Degrees	4.0	363.3	370.9	381.8	380.6	350.2	695.7	340.9	312.4	333.1

230 Degrees	6.6	288.4	290.5	301.1	295.9	262.8	658.2	253.4	224.8	254.9
240 Degrees	7.9	213.5	208.7	218.6	209.3	173.2	603.5	160.7	134.4	176.8
250 Degrees	8.9	135.0	127.8	133.6	123.6	86.3	537.4	73.2	48.1	93.7
260 Degrees	9.0	62.2	47.8	52.4	37.5	3.1	462.9	6.9	31.7	21.5
270 Degrees	8.4	3.3	27.0	22.7	40.9	72.8	380.2	82.7	105.5	45.7
280 Degrees	7.5	67.3	93.4	92.8	111.8	142.7	291.8	149.0	168.2	109.4
290 Degrees	6.2	121.6	151.6	152.9	170.6	199.7	202.2	201.1	219.0	158.6
300 Degrees	4.4	167.9	199.6	202.5	218.9	244.0	113.3	244.0	259.1	199.2
310 Degrees	2.2	205.4	234.3	240.4	258.2	277.7	27.5	273.5	284.2	227.9
320 Degrees	0.2	229.7	255.9	262.3	280.9	295.3	50.8	287.7	295.9	243.9
330 Degrees	1.7	244.7	265.6	273.8	290.0	300.9	125.4	291.1	295.6	248.4
340 Degrees	4.0	246.9	262.7	271.1	289.2	294.1	189.8	279.1	281.3	238.5
350 Degrees	6.1	242.0	254.1	259.5	277.7	278.8	239.5	259.8	254.4	220.9

EXPERIMENTAL RESULTS TEST 5 (T5)

Feed X Position	150.0	160.0	170.0	180.0	190.0	200.0	210.0	220.0	230.0	240.0
Spindle Speed	515.3	301.4	300.3	299.3	299.9	300.0	300.4	299.7	299.4	300.1
Spindle Power	1557.5	2753.4	2456.4	2439.0	2499.4	2517.5	2526.4	2550.3	2540.3	2451.1
Spindle Torque	13.4	38.3	34.7	34.9	35.5	35.8	35.8	35.8	35.9	34.7
Spindle Current	4.3	9.4	8.6	8.5	8.7	8.7	8.7	8.8	8.8	8.5
Feedrate	8.4	77.2	77.3	77.3	77.3	77.3	77.2	77.3	77.2	56.8
Feed Power	105.0	441.3	445.4	444.8	450.4	444.0	453.2	443.5	449.4	347.2
Feed Torque	1.6	1.7	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
Feed Current	1.4	2.5	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Tool Temp.	39.6	210.5	244.4	268.1	285.5	299.1	310.7	320.3	329.1	338.6
Tool Fx/Fy	5.7	1.4	1.2	1.3	1.3	1.3	7.9	1.3	1.2	1.5
Tool Fz	-4.7	-15.4	-12.9	-12.8	-13.0	-13.2	-13.2	-13.4	-13.2	-12.7
0 Degrees	16.2	443.9	472.0	503.5	499.3	518.6	602.3	491.8	495.1	438.9
10 Degrees	17.6	415.7	438.4	475.2	468.5	476.8	617.6	459.2	453.6	413.0
20 Degrees	21.4	336.5	355.9	391.6	384.9	388.0	612.0	378.5	369.6	345.1
30 Degrees	25.9	239.7	253.5	286.9	281.1	282.6	587.3	283.3	274.5	259.1
40 Degrees	30.8	134.3	144.9	172.8	167.6	167.2	560.0	169.5	169.8	160.0
50 Degrees	35.8	23.8	28.1	52.1	46.3	45.9	512.0	48.5	50.5	52.6
60 Degrees	40.7	87.5	88.4	72.3	78.8	80.2	466.6	78.3	77.7	60.7
70 Degrees	45.5	198.4	206.7	200.5	208.6	211.6	385.6	210.2	212.2	177.3
80 Degrees	49.8	307.8	324.6	326.8	338.6	342.2	280.6	341.8	342.5	292.3
90 Degrees	53.6	413.4	439.8	445.8	462.0	466.3	165.4	463.4	462.3	400.3
100 Degrees	56.8	514.2	544.8	555.0	568.1	576.2	44.3	572.2	570.0	499.6
110 Degrees	59.2	601.4	632.4	650.6	660.6	671.4	82.6	666.4	662.6	586.1
120 Degrees	60.5	671.8	703.8	730.7	737.9	747.4	215.3	743.0	736.6	658.7
130 Degrees	60.9	724.4	757.4	790.0	797.1	805.0	345.8	797.9	790.9	712.4
140 Degrees	60.3	757.8	789.8	826.5	832.7	840.4	468.7	830.1	823.1	747.8
150 Degrees	58.8	768.6	801.2	840.4	844.5	851.4	579.0	840.8	830.9	762.4
160 Degrees	56.2	759.6	787.6	829.3	833.7	838.8	674.7	827.5	817.3	757.4
170 Degrees	53.0	729.8	752.0	795.7	797.8	802.6	749.4	791.2	779.7	733.2
180 Degrees	49.1	680.4	695.2	741.3	741.2	744.9	805.4	734.8	722.6	688.9
190 Degrees	44.7	612.4	620.7	665.8	665.7	667.9	837.2	659.1	644.3	626.2
200 Degrees	40.0	527.8	532.4	577.1	572.5	573.8	847.5	566.6	550.8	547.6
210 Degrees	35.2	430.1	431.2	474.6	467.2	467.2	832.2	462.3	445.8	458.0
220 Degrees	30.3	324.5	322.1	364.3	351.4	351.7	796.3	347.2	331.4	357.9

230 Degrees	25.6	217.6	205.6	244.7	231.4	230.5	737.6	227.9	211.1	252.0
240 Degrees	21.3	107.4	88.7	120.9	106.6	106.4	661.7	103.5	86.2	140.8
250 Degrees	17.3	1.4	29.6	4.4	16.7	17.0	567.4	17.4	35.2	30.6
260 Degrees	13.9	108.5	143.8	127.3	140.0	138.4	461.1	135.6	156.0	79.0
270 Degrees	11.1	208.8	252.9	243.6	253.7	255.6	346.1	248.0	266.6	182.4
280 Degrees	8.9	301.4	349.9	348.6	358.6	360.1	226.9	350.2	367.9	275.9
290 Degrees	7.0	378.8	433.1	440.1	449.0	450.3	107.4	440.3	453.4	358.3
300 Degrees	5.8	444.0	499.1	516.4	523.8	524.4	10.6	513.2	523.2	426.7
310 Degrees	5.3	492.0	547.2	573.4	578.4	580.0	128.4	567.9	574.5	479.8
320 Degrees	5.5	524.1	576.7	611.3	614.3	615.8	243.7	602.2	605.3	516.5
330 Degrees	6.3	537.2	587.3	627.5	629.2	630.4	348.4	616.4	616.4	534.3
340 Degrees	8.0	533.6	579.8	624.8	624.9	623.6	438.3	611.1	607.4	534.0
350 Degrees	10.1	517.4	554.6	603.6	603.7	603.7	512.4	585.7	581.6	516.4

EXPERIMENTAL RESULTS TEST 6 (T6)

Feed X Position	280.0	290.0	300.0	310.0	320.0	330.0	340.0	350.0	360.0	370.0
Spindle Speed	556.2	600.1	599.9	600.1	600.1	599.8	600.1	599.9	600.1	600.1
Spindle Power	1474.0	3146.5	3110.3	3070.2	3026.4	3031.3	3012.7	3007.2	2980.6	2935.8
Spindle Torque	10.5	24.1	24.0	23.8	23.6	23.5	23.4	23.4	23.2	22.9
Spindle Current	3.8	8.1	8.0	7.9	7.8	7.8	7.7	7.7	7.6	7.5
Feedrate	8.4	57.5	57.6	57.6	57.6	57.5	57.6	57.6	57.6	45.2
Feed Power	101.4	340.7	340.8	341.4	341.4	342.2	341.9	342.2	342.9	282.6
Feed Torque	1.4	1.9	1.9	1.9	1.8	1.8	1.8	1.7	1.8	1.8
Feed Current	1.3	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Tool Temp.	37.8	202.1	251.0	287.8	314.4	334.3	349.6	362.3	372.7	382.9
Tool Fx/Fy	50.3	1.2	1.4	1.3	1.3	1.3	9.7	1.2	1.3	1.6
Tool Fz	-3.8	-13.8	-13.9	-13.7	-13.5	-13.5	-13.4	-13.4	-13.2	-12.9
0 Degrees	14.1	326.0	325.8	300.3	309.0	309.0	437.7	309.5	302.7	297.7
10 Degrees	15.2	296.2	301.9	273.9	286.4	279.8	434.8	279.5	271.9	273.5
20 Degrees	17.4	239.0	251.8	225.2	237.0	229.8	425.2	230.8	216.8	225.5
30 Degrees	21.3	157.8	175.0	149.2	156.7	155.0	403.3	157.1	143.2	156.7
40 Degrees	26.0	72.2	91.1	64.5	70.2	70.5	360.0	73.0	59.4	80.2
50 Degrees	31.2	18.7	3.0	27.4	22.1	21.3	330.0	16.5	32.2	4.3
60 Degrees	36.3	112.3	99.2	119.4	117.1	114.6	284.4	107.1	125.8	90.6
70 Degrees	41.2	209.0	195.3	214.9	213.7	211.5	227.2	199.0	219.2	178.7
80 Degrees	45.8	301.2	289.4	308.3	306.5	304.7	155.0	288.2	310.2	264.7
90 Degrees	49.9	389.7	379.5	394.8	392.0	389.9	72.9	370.1	394.8	346.4
100 Degrees	53.2	469.8	460.3	472.6	470.4	467.1	17.9	445.8	471.7	422.4
110 Degrees	55.8	538.9	530.4	539.8	539.2	534.1	109.8	511.0	536.5	488.4
120 Degrees	57.3	592.5	586.5	592.7	592.5	586.5	202.3	564.5	588.9	542.6
130 Degrees	57.8	633.3	630.6	631.1	631.7	625.1	292.8	604.1	626.0	583.9
140 Degrees	57.0	652.3	652.2	650.8	651.1	642.9	380.6	623.9	644.3	606.4
150 Degrees	55.6	657.4	660.6	655.8	655.9	647.7	461.0	629.9	647.7	616.2
160 Degrees	52.7	639.7	645.7	639.7	639.1	628.9	528.3	613.8	632.2	606.9
170 Degrees	49.4	608.5	617.9	611.2	609.9	600.6	582.4	586.3	602.9	585.1
180 Degrees	45.0	557.2	571.4	562.7	561.9	550.9	621.9	539.7	556.1	544.2
190 Degrees	40.5	497.4	515.1	503.5	503.1	492.7	641.8	484.0	499.7	495.8
200 Degrees	35.2	424.3	442.7	430.3	430.0	416.8	647.0	412.0	428.0	431.7
210 Degrees	30.1	340.1	363.4	350.1	349.7	337.7	631.1	332.4	347.4	358.1
220 Degrees	24.8	246.0	278.3	259.6	260.3	248.6	602.5	244.1	257.6	276.2

230 Degrees	19.5	151.2	188.3	165.1	166.5	156.3	555.2	152.8	166.4	193.7
240 Degrees	14.4	55.4	93.3	69.7	70.4	62.5	499.3	59.3	72.7	107.3
250 Degrees	10.2	40.7	1.0	21.7	23.4	27.0	425.8	32.4	19.6	20.5
260 Degrees	6.2	134.8	89.5	111.3	115.0	114.7	345.4	121.0	109.8	62.9
270 Degrees	3.1	218.5	174.7	192.9	198.0	196.5	254.9	201.7	190.4	138.5
280 Degrees	0.6	296.1	255.9	267.2	274.3	273.0	162.0	276.9	264.9	209.8
290 Degrees	0.7	356.8	317.2	327.8	334.3	331.0	66.4	336.2	323.6	268.0
300 Degrees	1.7	408.3	371.0	380.4	385.4	380.3	27.7	385.6	373.0	317.7
310 Degrees	1.4	440.9	405.7	412.4	417.0	410.1	118.9	416.0	404.1	351.7
320 Degrees	0.7	461.6	427.9	432.6	439.1	429.5	201.2	436.2	425.0	376.3
330 Degrees	1.2	459.6	430.5	434.1	438.6	425.8	276.3	436.2	423.1	380.9
340 Degrees	3.0	448.6	426.2	429.5	432.3	416.3	336.7	428.1	414.0	378.1
350 Degrees	4.8	425.1	409.7	413.1	416.9	395.1	387.8	410.5	396.9	364.9

EXPERIMENTAL RESULTS TEST 7 (T7)

Feed X Position	340.0	350.0	360.0	370.0	380.0	390.0	400.0	410.0	420.0	430.0
Spindle Speed	509.0	400.0	400.3	400.3	400.0	399.6	400.2	399.9	399.8	400.5
Spindle Power	2222.0	3964.8	4090.7	4134.6	4139.8	4165.4	4235.5	4237.4	4258.6	4258.4
Spindle Torque	18.7	41.9	42.8	42.9	43.2	43.4	43.7	43.8	44.0	43.8
Spindle Current	5.8	10.7	11.0	11.1	11.1	11.1	11.2	11.3	11.3	11.3
Feedrate	8.4	95.1	95.2	95.4	95.4	95.5	95.4	95.4	95.2	64.4
Feed Power	106.2	546.0	549.6	543.3	543.8	543.8	541.3	540.7	543.7	387.5
Feed Torque	1.7	4.0	3.8	3.8	3.6	3.7	3.8	3.7	3.8	3.7
Feed Current	1.3	2.5	2.6	2.5	2.5	2.5	2.5	2.5	2.5	2.5
Tool Temp.	54.1	284.9	325.4	351.3	370.1	384.8	396.4	406.6	415.0	422.0
Tool Fx/Fy	0.2	0.8	0.9	1.0	1.0	1.0	11.4	1.0	1.0	1.1
Tool Fz	-7.2	-27.1	-27.5	-27.3	-27.6	-27.8	-28.2	-28.2	-28.3	-27.4
0 Degrees	9.2	932.5	1025.3	1038.7	1033.7	1043.8	1827.7	1141.5	1036.7	1105.9
10 Degrees	7.0	871.1	942.2	993.6	997.9	986.9	1764.4	1012.6	968.4	1014.9
20 Degrees	0.8	602.5	690.8	772.1	754.4	762.3	1655.9	770.2	737.3	792.9
30 Degrees	8.1	278.9	377.1	469.3	449.5	456.1	1536.9	466.7	434.3	518.5
40 Degrees	16.2	58.3	49.9	144.8	123.6	134.3	1420.0	140.4	112.2	222.2
50 Degrees	23.4	391.1	278.3	195.6	204.5	197.5	1220.0	200.3	213.2	79.7
60 Degrees	29.1	715.5	612.4	529.9	533.1	528.8	1009.5	533.6	542.4	380.0
70 Degrees	33.4	1015.0	925.8	849.7	839.4	848.3	761.4	849.0	856.6	669.8
80 Degrees	36.0	1283.1	1206.1	1131.8	1120.5	1133.7	453.8	1130.8	1143.6	937.7
90 Degrees	37.2	1517.3	1452.3	1379.0	1370.3	1386.7	123.6	1381.0	1392.5	1175.9
100 Degrees	36.5	1704.7	1649.2	1580.9	1576.2	1594.4	214.5	1583.1	1596.3	1373.4
110 Degrees	34.0	1838.0	1784.3	1735.1	1725.9	1746.3	545.0	1738.2	1745.8	1531.5
120 Degrees	30.0	1914.2	1874.9	1835.1	1824.8	1845.9	860.8	1836.4	1840.0	1639.3
130 Degrees	24.8	1934.1	1902.2	1877.8	1871.4	1888.2	1142.8	1880.2	1881.8	1696.5
140 Degrees	18.1	1895.2	1882.4	1866.7	1859.2	1880.7	1391.5	1868.6	1864.3	1709.5
150 Degrees	10.2	1800.4	1800.6	1796.7	1795.1	1811.4	1592.1	1794.2	1790.3	1666.7
160 Degrees	1.5	1653.4	1662.6	1677.4	1677.6	1691.7	1745.7	1670.9	1664.1	1576.1
170 Degrees	7.8	1460.0	1482.8	1513.0	1512.0	1522.1	1846.0	1499.9	1493.5	1436.8
180 Degrees	17.4	1222.8	1257.7	1302.1	1303.3	1307.0	1891.3	1287.1	1278.8	1255.7
190 Degrees	27.2	946.7	1001.2	1056.4	1053.1	1062.0	1878.2	1037.6	1025.4	1041.4
200 Degrees	36.9	639.9	712.5	774.1	774.2	778.7	1807.9	755.6	743.7	796.0
210 Degrees	45.9	318.1	403.3	469.7	476.2	478.0	1687.1	453.7	438.3	529.9
220 Degrees	54.4	9.0	85.1	152.5	162.9	167.0	1516.8	135.7	122.4	248.5

230 Degrees	61.8	337.3	242.6	171.6	152.2	157.7	1302.8	186.7	196.2	41.9
240 Degrees	67.9	660.9	564.3	489.8	466.4	470.6	1053.8	504.5	505.8	332.6
250 Degrees	72.6	960.3	865.1	791.2	762.0	771.7	775.4	806.7	803.4	614.9
260 Degrees	75.4	1233.1	1141.8	1072.4	1034.3	1053.6	476.1	1083.2	1081.5	880.4
270 Degrees	76.7	1469.2	1381.9	1318.0	1274.7	1299.4	166.5	1328.4	1326.1	1122.2
280 Degrees	76.1	1658.6	1579.9	1520.9	1480.1	1512.4	158.2	1526.0	1529.9	1321.0
290 Degrees	73.8	1796.5	1730.2	1674.2	1640.5	1669.4	478.2	1685.6	1686.5	1482.0
300 Degrees	69.9	1876.8	1822.6	1780.0	1751.1	1773.6	779.5	1790.7	1795.8	1596.8
310 Degrees	64.6	1903.9	1865.5	1832.7	1811.2	1828.5	1063.1	1844.8	1846.0	1665.1
320 Degrees	57.7	1869.8	1849.9	1829.6	1813.2	1829.6	1306.7	1836.0	1838.0	1679.9
330 Degrees	49.8	1777.7	1773.7	1765.8	1756.3	1770.7	1511.3	1772.0	1776.4	1641.3
340 Degrees	40.9	1636.6	1651.1	1649.5	1648.7	1662.5	1669.9	1662.0	1661.4	1557.1
350 Degrees	33.2	1484.0	1520.8	1525.3	1514.2	1543.7	1775.8	1533.8	1522.9	1457.7

EXPERIMENTAL RESULTS TEST 8 (T8)

Feed X Position	200.0	210.0	220.0	230.0	240.0	250.0	260.0	270.0		
Spindle Speed	510.4	450.1	451.4	453.9	454.9	453.1	448.6	445.6		
Spindle Power	2436.0	5338.4	5288.8	5226.3	5189.4	5181.1	5187.6	5195.1		
Spindle Torque	19.4	40.8	40.1	39.6	39.1	39.2	39.3	39.4		
Spindle Current	6.4	14.0	13.9	13.7	13.6	13.6	13.6	13.6		
Feedrate	7.8	36.8	37.0	37.1	37.1	37.2	37.2	37.1		
Feed Power	99.2	219.0	218.2	219.1	219.3	220.6	219.7	218.8		
Feed Torque	1.8	3.6	3.3	3.2	3.1	3.1	3.1	3.1		
Feed Current	1.4	2.4	2.4	2.4	2.4	2.4	2.4	2.4		
Tool Temp.	46.7	262.0	301.1	328.6	348.6	364.4	377.4	387.7		
Tool Fx/Fy	0.7	0.8	1.0	1.0	1.0	1.0	12.8	1.0		
Tool Fz	-8.4	-22.2	-21.2	-20.5	-20.2	-19.9	-19.8	-19.6		
0 Degrees	4.7	639.8	690.0	712.6	689.4	684.7	1067.8	661.4		
10 Degrees	0.9	612.5	665.3	659.5	641.9	644.7	1056.5	641.4		
20 Degrees	10.0	464.7	543.4	521.4	513.0	520.6	1008.3	510.9		
30 Degrees	23.3	257.9	351.6	341.6	330.8	348.2	948.7	335.8		
40 Degrees	37.4	32.7	149.4	145.6	138.4	155.4	860.0	146.9		
50 Degrees	51.1	191.7	63.7	57.6	59.8	41.5	760.0	49.9		
60 Degrees	64.1	414.3	277.2	261.3	258.2	238.4	635.7	247.9		
70 Degrees	76.0	629.4	480.4	460.3	450.5	431.6	517.1	440.8		
80 Degrees	85.9	826.4	675.7	650.4	633.3	615.1	344.4	624.2		
90 Degrees	93.9	1000.6	857.1	821.8	801.4	783.2	153.2	790.3		
100 Degrees	99.3	1146.9	1010.9	968.9	943.5	930.3	42.2	936.6		
110 Degrees	102.7	1258.8	1137.2	1091.0	1060.8	1051.9	239.3	1057.7		
120 Degrees	102.9	1330.0	1235.7	1179.0	1147.4	1142.8	432.4	1146.6		
130 Degrees	100.5	1366.1	1288.3	1234.8	1200.8	1199.4	615.3	1203.6		
140 Degrees	95.7	1358.4	1305.6	1253.6	1219.2	1223.4	782.2	1224.8		
150 Degrees	87.9	1308.0	1289.8	1231.8	1201.6	1210.7	931.1	1210.4		
160 Degrees	78.5	1225.8	1227.9	1177.8	1148.9	1158.8	1050.9	1161.4		
170 Degrees	67.3	1103.4	1134.0	1089.6	1062.8	1076.5	1140.9	1077.9		
180 Degrees	54.2	954.5	1011.8	969.4	947.8	964.3	1200.7	964.8		
190 Degrees	40.5	782.4	857.9	825.1	807.0	825.1	1220.1	825.1		
200 Degrees	25.8	580.1	684.7	655.7	643.9	664.4	1205.1	662.4		
210 Degrees	10.9	364.2	492.3	467.5	464.2	485.8	1159.1	483.7		
220 Degrees	3.5	140.0	285.5	270.1	271.4	295.6	1076.2	290.9		

230 Degrees	17.5	89.4	76.2	65.2	70.8	97.1	964.3	93.6		
240 Degrees	30.0	314.0	139.8	138.5	131.9	104.1	827.1	106.8		
250 Degrees	41.0	528.5	348.8	333.9	328.3	299.2	665.1	304.3		
260 Degrees	50.2	727.7	543.6	520.0	512.4	486.1	487.1	490.7		
270 Degrees	57.2	900.1	724.4	690.3	682.2	651.0	296.1	658.2		
280 Degrees	62.3	1042.0	874.3	834.7	824.4	793.7	95.5	801.2		
290 Degrees	64.9	1154.6	1001.2	955.6	941.8	911.6	105.9	919.5		
300 Degrees	64.6	1227.0	1096.8	1044.6	1028.0	998.5	300.3	1005.8		
310 Degrees	61.8	1254.7	1147.5	1094.0	1074.8	1051.5	482.3	1055.9		
320 Degrees	56.8	1249.0	1169.0	1111.8	1094.7	1073.0	646.8	1075.2		
330 Degrees	49.8	1205.5	1149.8	1094.0	1075.1	1059.8	786.8	1059.7		
340 Degrees	40.6	1120.4	1094.8	1040.8	1021.2	1008.8	903.6	1008.4		
350 Degrees	32.4	1022.1	1028.1	973.7	956.7	945.5	994.8	938.4		

EXPERIMENTAL RESULTS TEST 9 (T9)

Feed X Position	420.0	430.0	440.0	450.0	460.0	470.0	480.0	490.0	500.0	510.0
Spindle Speed	534.8	499.7	499.8	499.8	499.7	499.8	500.3	500.1	484.8	479.8
Spindle Power	1726.7	3984.0	4166.3	4182.1	4344.0	4300.7	4238.3	4236.0	4948.7	5105.1
Spindle Torque	12.9	34.3	35.2	35.4	35.8	35.7	35.4	35.3	38.1	38.5
Spindle Current	4.5	10.3	10.7	10.8	11.2	11.1	11.0	11.0	12.9	13.4
Feedrate	8.5	96.1	95.9	96.1	95.9	96.1	96.0	95.8	96.1	65.5
Feed Power	106.4	549.0	543.9	540.7	551.0	539.0	545.8	547.1	544.0	398.6
Feed Torque	1.6	2.8	3.0	2.7	2.9	2.8	2.9	2.7	2.8	2.8
Feed Current	1.4	2.6	2.5	2.5	2.6	2.5	2.5	2.6	2.5	2.5
Tool Temp.	37.8	202.7	234.7	261.0	282.4	300.1	314.5	327.0	337.9	348.9
Tool Fx/Fy	0.5	0.9	0.9	0.9	0.8	0.8	6.1	0.9	0.9	1.1
Tool Fz	-4.6	-23.7	-24.0	-23.9	-23.9	-23.7	-23.7	-23.9	-23.6	-22.2
0 Degrees	36.6	494.9	464.7	486.7	408.1	426.6	875.8	355.5	393.0	462.2
10 Degrees	38.3	450.1	392.8	412.2	309.0	326.1	826.9	306.0	338.8	393.6
20 Degrees	41.0	332.6	281.8	294.0	175.0	168.3	756.8	208.2	239.7	280.0
30 Degrees	43.6	147.6	105.0	119.2	15.2	2.2	657.9	48.7	73.8	130.9
40 Degrees	46.6	33.3	77.3	70.4	167.1	177.0	580.0	128.6	107.8	35.2
50 Degrees	49.6	212.6	259.1	259.7	355.6	359.5	380.0	309.4	294.3	199.1
60 Degrees	52.2	388.3	441.7	440.7	524.0	530.0	214.1	483.0	477.4	361.8
70 Degrees	54.7	555.7	613.7	613.2	683.9	690.5	145.7	649.9	653.8	523.9
80 Degrees	56.5	717.4	774.7	772.3	836.7	841.9	20.5	804.4	815.9	671.8
90 Degrees	57.2	857.5	903.9	897.2	956.1	960.3	199.7	934.3	949.1	805.7
100 Degrees	57.5	970.7	1011.3	1006.9	1057.9	1059.7	378.8	1048.0	1063.2	923.1
110 Degrees	56.5	1057.0	1092.4	1091.0	1132.6	1132.6	546.8	1129.3	1148.6	1015.3
120 Degrees	54.5	1109.1	1138.4	1139.7	1171.0	1171.4	710.6	1180.0	1202.5	1082.3
130 Degrees	52.1	1131.5	1160.0	1161.5	1186.2	1183.5	865.5	1206.5	1224.7	1119.5
140 Degrees	48.7	1119.3	1143.1	1145.8	1157.6	1161.5	985.2	1194.6	1216.6	1131.3
150 Degrees	45.1	1069.1	1092.8	1100.0	1102.1	1105.2	1086.9	1152.4	1174.5	1113.5
160 Degrees	41.0	991.6	1018.6	1026.2	1023.9	1022.7	1157.2	1083.1	1101.3	1060.5
170 Degrees	36.4	885.9	913.2	921.2	907.7	909.5	1195.8	983.8	1003.7	986.0
180 Degrees	31.4	748.4	786.5	795.2	770.8	768.6	1210.0	855.7	874.0	888.9
190 Degrees	26.8	596.3	629.1	639.2	620.8	613.3	1186.8	714.5	717.9	760.5
200 Degrees	22.3	429.8	459.0	467.2	444.5	442.9	1135.6	559.4	556.1	618.0
210 Degrees	17.9	236.0	279.2	287.0	255.3	261.2	1054.0	374.6	377.8	465.0
220 Degrees	14.5	45.6	89.1	99.9	82.8	81.4	942.9	193.0	182.2	292.2

230 Degrees	11.5	140.5	95.8	91.0	99.4	100.7	811.2	16.9	1.4	125.3
240 Degrees	8.8	330.5	277.9	268.9	282.1	282.0	656.0	170.5	181.0	40.0
250 Degrees	7.5	503.3	449.3	438.5	443.9	447.6	488.3	340.3	362.8	209.4
260 Degrees	6.9	655.2	602.7	596.7	595.6	594.3	314.2	488.6	509.8	354.5
270 Degrees	7.2	788.4	730.2	722.6	722.3	717.8	132.9	623.3	643.4	484.8
280 Degrees	8.3	895.9	841.7	827.5	819.8	817.0	52.8	736.2	759.9	600.3
290 Degrees	10.2	968.4	925.2	909.6	890.8	882.5	230.7	815.6	830.2	688.7
300 Degrees	12.7	1007.7	963.4	947.9	926.4	914.1	394.9	863.3	879.0	754.1
310 Degrees	15.4	1025.1	985.1	965.5	939.0	922.3	547.3	887.1	901.4	789.6
320 Degrees	18.5	1015.8	976.4	946.8	915.7	896.9	669.2	885.5	888.0	796.4
330 Degrees	21.9	968.9	930.7	897.5	861.6	840.8	769.4	844.5	852.8	779.1
340 Degrees	25.6	897.4	871.3	840.6	786.6	765.1	840.5	776.8	786.4	732.1
350 Degrees	29.6	815.9	829.1	812.7	688.3	697.8	881.4	709.8	699.4	684.5

EXPERIMENTAL RESULTS TEST 10 (T10)

Feed X Position	550.0	560.0	570.0	580.0	590.0	600.0	610.0	620.0	630.0	640.0
Spindle Speed	517.6	399.9	400.0	400.0	399.9	399.9	400.0	399.9	400.0	400.0
Spindle Power	1746.9	2966.8	2771.4	2670.1	2616.0	2597.3	2595.1	2592.6	2593.8	2591.8
Spindle Torque	14.8	33.1	31.2	30.1	29.7	29.5	29.5	29.4	29.4	29.5
Spindle Current	4.7	8.6	8.1	7.8	7.7	7.7	7.6	7.6	7.6	7.6
Feedrate	7.8	38.0	38.2	38.1	38.2	38.1	38.1	38.1	38.0	33.5
Feed Power	99.8	233.7	237.9	236.7	236.9	236.3	234.0	236.5	233.6	213.5
Feed Torque	1.6	1.9	1.7	1.7	1.7	1.8	1.8	1.9	1.9	2.0
Feed Current	1.5	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Tool Temp.	54.1	284.9	325.4	351.3	370.1	384.8	396.4	406.6	415.0	422.0
Tool Fx/Fy	0.7	0.7	0.9	0.9	0.9	0.9	5.6	0.8	0.8	0.8
Tool Fz	-5.4	-16.7	-15.3	-14.6	-14.4	-14.5	-14.7	-14.7	-14.8	-14.8
0 Degrees	27.4	476.1	491.9	499.2	501.5	497.5	561.5	502.8	501.6	495.0
10 Degrees	29.1	455.3	470.3	482.0	485.5	487.3	594.1	490.2	481.9	479.0
20 Degrees	35.5	381.1	405.8	420.2	425.3	423.1	603.1	415.6	409.2	413.1
30 Degrees	43.8	275.9	318.0	335.0	339.9	332.2	594.1	316.9	308.6	315.0
40 Degrees	52.9	155.8	215.5	234.8	238.8	227.0	550.0	204.0	193.6	203.1
50 Degrees	62.3	28.7	99.6	122.3	125.3	110.6	500.0	79.4	67.1	81.1
60 Degrees	72.4	106.6	24.5	2.5	2.5	14.8	470.0	50.7	67.1	48.1
70 Degrees	82.3	245.5	154.6	124.9	125.5	145.9	417.8	186.6	205.7	182.6
80 Degrees	91.7	383.6	286.0	253.1	254.6	279.8	322.2	321.5	343.6	317.8
90 Degrees	100.6	515.3	414.7	380.6	382.1	410.8	212.4	452.6	477.4	450.7
100 Degrees	108.3	638.9	536.1	504.0	503.6	534.9	91.6	576.5	601.8	575.2
110 Degrees	114.6	749.1	646.2	616.4	614.6	649.1	34.8	687.0	713.6	687.5
120 Degrees	119.3	840.4	741.3	715.2	713.7	746.8	169.9	782.6	807.1	783.3
130 Degrees	122.2	911.5	818.0	796.5	795.6	827.2	304.7	858.8	881.4	861.2
140 Degrees	123.2	959.1	874.6	856.6	856.5	886.8	436.0	912.6	934.2	915.3
150 Degrees	122.1	979.6	908.1	894.8	896.9	920.7	561.6	943.2	960.0	945.0
160 Degrees	119.0	976.7	917.4	908.1	908.8	933.1	675.2	948.4	961.3	951.0
170 Degrees	114.2	947.1	902.5	899.0	900.2	919.4	773.7	928.6	936.3	931.1
180 Degrees	107.7	894.1	864.8	866.3	867.8	882.0	853.9	886.3	887.8	888.2
190 Degrees	99.7	819.1	805.9	812.0	814.4	823.4	911.0	819.4	818.2	822.7
200 Degrees	90.8	723.5	728.4	739.5	742.0	744.9	942.5	735.4	728.5	736.7
210 Degrees	81.3	614.3	634.7	648.0	650.9	651.7	949.8	632.9	624.9	636.3
220 Degrees	71.3	490.8	527.5	545.0	548.1	543.7	932.0	518.9	506.4	523.1

230 Degrees	61.3	358.2	409.4	431.1	432.6	425.2	892.4	394.6	379.2	398.4
240 Degrees	51.4	220.2	284.5	310.7	311.3	300.0	829.3	261.7	245.3	268.3
250 Degrees	42.0	82.3	156.6	187.1	186.3	169.7	746.5	128.3	109.4	135.3
260 Degrees	33.3	52.0	28.9	60.6	59.7	40.3	647.7	4.0	24.5	2.6
270 Degrees	25.5	182.0	93.9	61.5	60.9	84.8	536.2	131.5	155.5	126.3
280 Degrees	19.0	299.3	210.0	177.0	179.6	203.8	414.2	250.7	275.2	247.1
290 Degrees	14.2	402.2	314.1	284.3	286.1	309.7	284.7	357.6	380.6	354.8
300 Degrees	10.9	488.0	403.1	378.1	379.7	402.3	153.0	449.0	469.8	446.3
310 Degrees	8.9	555.2	478.7	456.5	459.3	478.7	21.0	521.9	540.7	520.3
320 Degrees	8.4	599.2	531.3	513.6	517.7	536.6	106.7	574.3	588.7	572.6
330 Degrees	9.7	619.1	563.9	549.9	556.0	571.3	227.9	602.5	612.9	600.6
340 Degrees	12.5	614.4	574.9	565.0	572.7	583.4	336.0	607.5	615.1	607.6
350 Degrees	15.6	595.7	567.7	562.5	571.4	576.7	429.0	595.1	600.5	596.5

EXPERIMENTAL RESULTS TEST 11 (T11)

Feed X Position	0.1	10.1	20.1	30.1	40.1	50.1	60.1	70.1	80.1	90.1
Spindle Speed	522.8	456.6	433.6	419.7	412.8	408.4	406.2	403.2	398.0	395.5
Spindle Power	2235.6	5300.8	5428.4	5581.0	5586.2	5620.1	5582.4	5644.9	5647.6	5585.9
Spindle Torque	17.2	41.1	43.7	45.5	46.2	46.6	46.6	46.8	47.3	47.3
Spindle Current	5.9	13.9	14.3	14.7	14.7	14.8	14.7	14.8	14.9	14.7
Feedrate	8.4	75.5	75.6	75.7	75.9	75.9	76.0	76.0	76.0	55.9
Feed Power	104.2	440.3	439.1	437.8	439.9	434.7	435.9	435.9	435.3	338.8
Feed Torque	1.8	4.1	3.9	3.8	3.7	3.6	3.6	3.6	3.6	3.7
Feed Current	1.3	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
Tool Temp.	37.8	202.1	251.0	287.8	314.4	334.3	349.6	362.3	372.7	382.9
Tool Fx/Fy	0.6	0.6	0.7	0.8	0.8	0.9	30.0	0.8	0.9	1.0
Tool Fz	-7.3	-24.4	-25.4	-25.6	-25.5	-25.3	-25.0	-25.0	-24.9	-24.1
0 Degrees	56.4	696.6	725.2	945.4	942.1	1059.9	1581.4	917.5	921.4	1051.9
10 Degrees	56.7	588.6	631.8	799.9	852.9	918.2	1510.3	752.6	814.5	840.0
20 Degrees	59.4	312.8	439.0	532.8	573.1	646.8	1356.4	542.0	554.4	596.5
30 Degrees	67.5	37.0	193.7	262.0	293.1	364.0	1230.0	266.2	282.7	338.1
40 Degrees	74.1	247.7	105.2	33.6	9.5	64.3	1140.0	30.5	19.5	48.1
50 Degrees	81.7	522.7	386.8	329.0	311.1	242.0	1030.0	311.1	319.9	228.9
60 Degrees	89.5	760.9	661.2	614.9	614.3	541.9	911.6	608.6	628.8	534.2
70 Degrees	95.7	994.5	936.1	896.8	892.8	810.5	639.6	878.5	905.7	805.3
80 Degrees	100.4	1208.2	1166.0	1142.3	1121.0	1057.8	360.9	1128.9	1152.7	1047.9
90 Degrees	103.1	1369.6	1362.9	1360.5	1333.5	1292.8	57.4	1352.4	1376.9	1281.6
100 Degrees	104.1	1497.0	1520.9	1522.9	1496.7	1459.3	242.2	1514.7	1542.9	1447.5
110 Degrees	103.9	1577.0	1620.5	1644.8	1623.5	1593.5	551.7	1638.5	1677.3	1587.0
120 Degrees	101.1	1607.1	1669.8	1718.4	1706.5	1680.0	819.6	1713.8	1761.0	1682.3
130 Degrees	97.0	1586.1	1673.0	1733.8	1731.4	1719.3	1063.9	1740.8	1794.5	1731.6
140 Degrees	91.6	1516.7	1633.6	1696.5	1695.1	1693.4	1305.2	1715.0	1767.2	1730.7
150 Degrees	84.3	1410.9	1528.4	1609.6	1613.4	1623.1	1466.0	1632.2	1692.5	1678.9
160 Degrees	76.2	1252.2	1387.6	1484.7	1490.4	1504.5	1599.8	1498.8	1571.5	1575.7
170 Degrees	67.8	1071.1	1215.4	1297.8	1302.7	1339.4	1684.3	1330.0	1410.6	1432.6
180 Degrees	57.5	855.0	988.8	1090.6	1101.6	1152.3	1724.5	1131.0	1214.8	1270.5
190 Degrees	47.4	597.8	742.2	852.6	850.3	902.1	1698.7	880.4	966.7	1048.5
200 Degrees	37.8	330.2	471.5	571.0	581.6	652.2	1627.7	622.9	700.8	821.9
210 Degrees	27.5	60.6	185.9	297.6	311.1	371.4	1504.7	338.3	421.5	561.7
220 Degrees	18.1	221.0	88.6	6.3	14.0	72.2	1338.9	46.3	130.9	281.4

230 Degrees	10.4	491.9	375.8	303.5	293.5	204.7	1156.5	236.2	140.1	17.3
240 Degrees	3.0	747.7	650.1	576.0	581.0	506.0	902.3	536.4	436.1	283.9
250 Degrees	3.2	988.4	907.8	850.9	853.5	766.8	650.3	785.9	709.6	544.4
260 Degrees	7.2	1195.1	1131.8	1082.4	1084.3	1006.6	368.9	1020.2	947.7	797.8
270 Degrees	9.9	1364.7	1317.9	1289.5	1304.2	1226.8	70.9	1227.2	1171.3	1031.2
280 Degrees	11.1	1478.2	1469.4	1465.6	1450.0	1386.6	202.7	1386.2	1336.0	1213.0
290 Degrees	9.8	1553.9	1573.0	1575.1	1558.7	1519.7	504.9	1490.3	1462.5	1371.6
300 Degrees	7.1	1590.2	1627.3	1648.5	1631.3	1596.9	763.1	1572.3	1551.0	1476.7
310 Degrees	3.2	1562.6	1626.7	1665.7	1677.3	1632.6	1004.9	1576.3	1585.5	1527.2
320 Degrees	2.5	1495.1	1583.4	1643.1	1617.2	1604.6	1224.3	1536.7	1563.3	1530.7
330 Degrees	8.8	1396.8	1490.1	1551.1	1517.1	1531.1	1383.2	1468.1	1497.2	1524.8
340 Degrees	16.8	1222.4	1338.4	1422.5	1371.6	1385.3	1514.5	1320.9	1369.0	1412.6
350 Degrees	23.1	1079.0	1161.7	1303.5	1250.7	1221.8	1587.3	1151.1	1211.3	1297.8

EXPERIMENTAL RESULTS TEST 12 (T12)

Feed X Position	640.0	650.0	660.0	670.0	680.0	690.0	700.0	710.0	720.0	730.0
Spindle Speed	491.7	300.1	300.3	300.2	299.8	299.9	300.0	299.9	300.1	300.3
Spindle Power	2117.8	3793.9	3502.4	3294.9	3242.7	3247.9	3268.1	3272.8	3256.7	3192.3
Spindle Torque	20.0	48.2	45.4	43.7	43.2	43.2	43.3	43.4	43.1	42.5
Spindle Current	5.9	11.9	11.2	10.7	10.6	10.6	10.6	10.6	10.6	10.4
Feedrate	8.1	57.0	57.2	57.3	57.3	57.3	57.3	57.3	57.3	45.3
Feed Power	101.1	327.1	326.8	328.3	327.2	329.3	328.0	328.5	328.6	273.4
Feed Torque	1.7	2.6	2.4	2.3	2.3	2.3	2.3	2.3	2.2	2.3
Feed Current	1.3	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
Tool Temp.	39.6	210.5	244.4	268.1	285.5	299.1	310.7	320.3	329.1	338.6
Tool Fx/Fy	0.3	0.6	0.8	0.7	0.7	0.7	5.9	0.7	0.7	0.8
Tool Fz	-7.5	-19.9	-18.1	-16.3	-16.0	-16.0	-16.5	-16.4	-16.1	-15.6
0 Degrees	15.7	1021.8	1078.2	883.2	849.6	866.6	1225.5	864.2	852.1	851.9
10 Degrees	10.2	942.4	996.9	824.8	788.6	811.2	1222.5	813.5	795.1	796.3
20 Degrees	0.9	725.3	816.0	654.9	618.8	633.8	1179.6	640.7	620.6	646.1
30 Degrees	12.1	477.7	588.1	448.6	411.4	426.3	1107.4	431.4	411.1	457.1
40 Degrees	23.1	208.3	341.9	226.6	185.0	199.2	1000.0	200.2	187.9	249.9
50 Degrees	33.3	62.1	84.8	9.5	51.5	33.0	900.0	40.3	53.1	26.8
60 Degrees	42.3	328.5	177.1	247.0	289.4	273.7	817.7	281.8	291.4	194.6
70 Degrees	49.9	587.8	438.7	479.2	520.9	506.7	646.0	515.3	522.7	416.6
80 Degrees	55.9	840.3	684.9	701.2	739.9	728.2	443.0	735.0	734.8	627.8
90 Degrees	59.9	1067.8	908.4	902.3	936.6	928.3	226.2	938.1	930.2	824.6
100 Degrees	61.6	1266.7	1108.1	1081.3	1111.7	1106.4	10.1	1116.0	1101.3	997.9
110 Degrees	61.4	1426.8	1279.1	1229.5	1255.7	1254.1	250.8	1266.3	1248.5	1144.8
120 Degrees	58.7	1546.2	1420.3	1339.8	1364.0	1365.6	486.7	1381.7	1362.7	1260.0
130 Degrees	54.3	1621.3	1519.0	1414.7	1435.6	1440.4	713.4	1456.4	1436.1	1341.6
140 Degrees	47.4	1647.2	1575.5	1451.3	1465.1	1471.5	919.4	1493.6	1469.4	1385.1
150 Degrees	39.2	1627.5	1585.6	1450.1	1457.0	1462.0	1101.9	1485.1	1460.1	1391.4
160 Degrees	29.4	1557.7	1549.7	1404.7	1406.2	1412.5	1253.3	1438.3	1411.4	1356.2
170 Degrees	18.8	1445.6	1469.9	1323.4	1316.8	1323.3	1368.6	1348.6	1322.4	1285.7
180 Degrees	7.0	1290.4	1344.9	1202.4	1190.3	1197.5	1447.9	1226.9	1199.3	1177.7
190 Degrees	5.0	1107.0	1185.0	1051.0	1033.6	1039.2	1482.3	1074.5	1045.2	1040.9
200 Degrees	17.1	891.7	993.3	871.8	850.7	854.8	1478.1	897.4	865.5	876.0
210 Degrees	28.8	651.7	779.4	670.6	644.9	648.8	1429.0	690.7	664.6	691.4
220 Degrees	39.9	389.6	545.6	453.1	423.0	425.6	1344.4	465.4	443.8	491.0

230 Degrees	49.9	118.1	300.1	229.6	193.3	192.5	1222.9	230.1	214.1	279.3
240 Degrees	58.4	152.7	48.1	5.1	36.4	40.5	1069.4	5.7	17.6	61.7
250 Degrees	65.4	417.0	207.4	216.2	264.9	269.0	888.2	239.6	247.5	154.2
260 Degrees	70.6	667.1	454.3	432.2	480.7	487.8	680.9	461.8	464.2	361.6
270 Degrees	73.8	894.5	685.3	633.5	680.2	687.6	457.6	666.1	666.4	556.6
280 Degrees	74.9	1092.1	890.8	813.4	853.4	859.8	223.8	845.2	842.2	728.6
290 Degrees	74.4	1255.2	1066.6	964.7	1000.7	1002.1	9.6	993.5	987.6	874.9
300 Degrees	71.9	1376.7	1207.9	1081.6	1110.8	1112.3	240.5	1109.9	1102.3	991.4
310 Degrees	67.7	1453.2	1307.0	1161.5	1184.1	1186.6	460.8	1185.4	1178.8	1077.4
320 Degrees	61.5	1483.9	1366.2	1203.4	1218.2	1220.5	662.9	1223.8	1219.9	1123.9
330 Degrees	53.6	1466.6	1378.8	1202.8	1210.5	1211.9	842.7	1221.4	1216.3	1132.1
340 Degrees	44.4	1402.6	1348.1	1163.6	1164.8	1165.6	991.1	1178.4	1172.5	1103.6
350 Degrees	34.9	1304.9	1279.4	1094.8	1089.4	1090.0	1107.5	1106.2	1100.4	1044.5

EXPERIMENTAL RESULTS TEST 13 (T13)

Feed X Position	580.0	590.0	600.0	610.0	620.0	630.0	640.0	650.0	660.0	670.0
Spindle Speed	539.6	499.3	499.8	500.5	499.9	500.1	500.0	501.2	499.3	500.7
Spindle Power	1391.8	2798.8	2821.7	2857.6	2819.8	2801.2	2808.0	2788.3	2778.6	2772.4
Spindle Torque	11.0	26.4	26.6	26.9	26.5	26.3	26.3	26.3	26.3	26.1
Spindle Current	3.6	7.2	7.2	7.3	7.2	7.2	7.2	7.1	7.1	7.1
Feedrate	7.9	57.8	57.9	57.8	57.8	57.8	57.7	57.8	57.7	45.5
Feed Power	95.9	338.2	341.6	336.3	337.7	336.8	335.9	338.1	336.0	280.9
Feed Torque	1.2	1.4	1.4	1.5	1.5	1.5	1.6	1.5	1.6	1.6
Feed Current	1.3	2.6	2.6	2.5	2.6	2.5	2.5	2.6	2.5	2.6
Tool Temp.	55.3	281.9	316.1	341.7	361.4	376.9	389.0	399.4	407.9	416.0
Tool Fx/Fy	1.0	0.8	0.8	0.8	0.8	0.8	1.3	0.8	0.8	0.8
Tool Fz	-2.0	-9.5	-10.5	-11.1	-11.1	-11.2	-11.3	-11.4	-11.1	-11.0
0 Degrees	124.5	459.9	471.3	492.3	484.9	489.7	580.5	456.5	459.7	436.3
10 Degrees	123.6	440.1	447.1	464.9	460.3	460.3	578.9	438.6	437.0	415.7
20 Degrees	121.8	401.9	404.1	414.8	414.9	409.8	565.7	407.3	393.0	383.2
30 Degrees	118.7	346.6	343.9	354.8	356.0	345.9	547.5	348.5	333.1	333.8
40 Degrees	115.4	285.2	280.8	292.2	290.9	278.0	510.0	282.7	270.2	280.5
50 Degrees	112.0	223.5	217.3	227.3	220.7	209.2	470.0	213.6	205.1	225.4
60 Degrees	108.5	163.4	151.8	157.2	150.1	141.7	445.4	143.7	139.5	170.4
70 Degrees	105.4	103.4	88.3	89.6	82.4	77.1	405.2	77.1	78.1	117.0
80 Degrees	102.3	46.2	30.6	25.2	17.5	15.6	346.4	14.3	23.7	66.2
90 Degrees	99.7	2.8	19.7	29.5	38.0	39.1	280.0	39.7	25.0	21.9
100 Degrees	97.5	46.4	64.7	77.9	88.2	88.0	211.3	88.2	67.5	17.5
110 Degrees	96.0	82.5	100.0	116.9	130.0	127.8	141.2	128.8	102.5	50.1
120 Degrees	95.3	106.2	124.5	145.0	158.5	154.6	74.2	156.5	127.0	72.5
130 Degrees	95.1	121.8	139.9	162.7	177.2	171.2	10.9	175.4	142.3	87.7
140 Degrees	95.6	125.9	142.9	165.9	182.2	175.3	45.0	182.0	146.5	92.6
150 Degrees	96.7	118.6	134.3	158.3	175.4	166.6	94.1	175.5	138.9	87.7
160 Degrees	98.3	101.7	114.4	139.0	157.4	148.1	135.8	158.3	120.0	74.8
170 Degrees	100.2	74.0	84.1	108.9	125.8	119.2	163.2	130.1	91.3	52.3
180 Degrees	102.4	38.4	47.4	68.9	86.0	79.4	181.3	92.4	52.9	22.3
190 Degrees	104.7	2.4	2.1	20.9	39.5	32.6	188.2	45.8	8.0	13.5
200 Degrees	107.2	50.0	50.6	33.7	16.0	20.7	180.7	8.0	42.1	54.8
210 Degrees	109.5	101.4	106.5	95.7	76.8	79.1	163.5	65.3	99.3	100.4
220 Degrees	111.8	156.3	167.3	159.0	139.2	141.8	133.8	128.5	157.9	149.7

230 Degrees	114.0	216.5	230.4	223.0	206.5	207.7	95.0	194.5	217.0	201.6
240 Degrees	116.4	275.7	289.2	288.6	274.3	271.8	50.3	257.2	277.9	254.1
250 Degrees	118.6	331.6	346.4	351.0	337.3	333.8	3.4	319.5	336.4	304.9
260 Degrees	120.9	386.3	401.4	409.5	398.1	391.4	62.6	380.9	390.2	353.1
270 Degrees	122.7	433.4	449.1	462.0	453.1	442.6	125.7	433.7	441.0	395.9
280 Degrees	124.5	474.7	491.4	508.4	500.6	488.0	193.6	481.6	485.7	434.1
290 Degrees	126.1	510.4	526.0	545.5	539.3	522.8	259.2	521.8	519.7	465.7
300 Degrees	127.1	535.0	550.0	572.1	567.6	548.4	322.4	550.4	545.3	489.6
310 Degrees	128.0	552.6	566.3	591.2	587.3	565.1	381.7	571.5	562.6	506.5
320 Degrees	128.4	558.9	569.7	596.8	595.3	569.8	435.6	579.2	567.3	513.9
330 Degrees	128.3	555.4	564.7	591.5	590.7	564.0	484.0	577.0	563.4	513.0
340 Degrees	127.7	546.2	553.0	578.0	576.3	549.3	523.3	566.0	549.2	504.7
350 Degrees	126.7	525.3	534.9	561.2	553.0	527.6	553.0	545.1	528.2	491.6

EXPERIMENTAL RESULTS TEST 14 (T14)

Feed X Position	0.1	10.1	20.1	30.1	40.1	50.1	60.1	70.1	80.1	90.1
Spindle Speed	522.0	399.8	399.9	399.9	399.9	399.8	400.1	400.4	399.9	400.3
Spindle Power	1420.6	3018.4	3195.0	3331.4	3381.0	3341.2	3358.5	3298.6	3355.2	3331.0
Spindle Torque	12.0	33.9	35.4	36.6	36.9	36.8	36.6	36.3	36.6	36.5
Spindle Current	3.9	8.7	9.1	9.4	9.5	9.4	9.5	9.3	9.5	9.4
Feedrate	7.9	77.2	77.2	77.2	77.3	77.2	77.2	77.2	77.2	56.3
Feed Power	97.0	443.2	449.7	444.0	442.9	444.2	441.8	447.5	442.1	351.2
Feed Torque	1.3	1.8	1.8	1.7	1.6	1.7	1.6	1.7	1.7	1.7
Feed Current	1.3	2.6	2.6	2.6	2.6	2.6	2.5	2.6	2.5	2.7
Tool Temp.	52.8	276.7	305.4	328.2	346.4	361.4	372.7	382.6	391.8	400.9
Tool Fx/Fy	0.9	1.2	1.1	1.2	1.5	1.3	1.8	1.3	1.9	2.0
Tool Fz	-2.5	-14.0	-15.9	-17.5	-17.2	-17.2	-17.1	-17.8	-17.3	-16.9
0 Degrees	129.6	463.4	497.6	442.9	434.7	379.8	473.6	444.3	487.2	487.9
10 Degrees	127.8	458.9	468.8	431.7	423.4	372.6	459.5	441.1	471.7	470.9
20 Degrees	125.3	432.2	415.8	401.0	393.4	338.7	431.6	403.7	444.5	445.0
30 Degrees	123.2	359.8	351.7	329.3	352.6	297.6	402.0	341.4	409.7	413.9
40 Degrees	120.9	286.1	282.2	256.9	310.5	250.6	382.9	275.4	362.7	369.7
50 Degrees	118.4	206.4	212.6	179.9	258.0	197.7	370.0	205.9	312.1	322.2
60 Degrees	116.4	131.3	140.4	109.1	208.8	149.6	368.0	136.0	254.2	266.8
70 Degrees	114.8	58.1	75.3	43.4	155.8	103.4	328.9	69.4	195.8	210.4
80 Degrees	113.3	13.2	13.6	18.1	104.6	57.1	288.2	11.3	139.9	156.0
90 Degrees	112.3	67.9	39.0	64.9	58.8	18.7	238.9	35.4	80.8	96.7
100 Degrees	111.6	115.9	79.6	105.9	16.4	9.3	196.0	81.2	33.6	50.7
110 Degrees	111.3	151.0	111.9	131.9	14.9	32.0	154.6	111.4	7.9	8.9
120 Degrees	111.6	174.2	130.2	147.3	40.5	48.2	117.5	132.6	41.6	25.2
130 Degrees	112.4	184.0	137.2	149.6	57.4	55.6	85.8	143.5	67.1	51.5
140 Degrees	113.9	177.1	130.2	140.6	65.6	54.0	62.7	141.6	85.5	71.6
150 Degrees	116.0	156.1	111.7	118.8	63.4	43.8	48.3	125.0	92.2	80.5
160 Degrees	118.5	124.6	80.3	88.4	53.7	27.4	39.3	98.4	91.0	82.0
170 Degrees	121.3	83.1	42.5	49.4	36.5	4.7	37.2	62.2	80.3	74.2
180 Degrees	124.2	27.7	6.6	0.7	11.0	23.5	41.3	15.1	61.9	59.3
190 Degrees	127.1	28.0	61.0	52.5	17.5	58.5	55.3	31.7	31.9	31.4
200 Degrees	129.9	94.4	119.0	113.6	56.4	98.0	72.2	89.3	1.0	2.0
210 Degrees	132.9	163.6	183.9	175.4	96.1	138.5	97.2	147.0	40.1	34.3
220 Degrees	135.5	229.8	247.5	236.3	140.2	182.3	127.6	206.0	84.8	76.6

230 Degrees	137.6	311.1	316.2	298.3	188.1	225.3	159.5	272.7	128.9	117.0
240 Degrees	139.2	380.1	378.5	356.9	233.1	266.7	198.3	331.6	182.2	168.7
250 Degrees	140.4	447.5	435.7	413.9	280.4	307.1	231.8	393.4	231.3	215.2
260 Degrees	141.3	511.0	490.5	465.9	322.6	342.8	268.1	448.7	280.8	262.8
270 Degrees	141.7	564.7	532.2	510.7	362.5	376.2	304.7	496.0	331.4	312.6
280 Degrees	141.4	613.8	570.0	547.8	402.0	404.3	339.1	542.2	374.0	353.6
290 Degrees	140.2	648.6	597.7	576.3	433.5	426.6	375.5	572.8	415.5	395.5
300 Degrees	138.9	675.8	615.9	596.9	459.4	443.9	403.5	598.5	456.8	437.5
310 Degrees	137.8	692.4	623.0	606.5	479.5	455.9	429.6	613.7	489.4	471.3
320 Degrees	136.5	696.1	620.0	604.9	494.1	460.9	451.6	619.0	511.7	494.7
330 Degrees	135.1	685.2	602.1	591.4	500.4	456.9	468.0	611.7	532.5	518.0
340 Degrees	133.0	662.4	573.4	570.2	501.2	445.4	479.5	592.5	536.2	523.7
350 Degrees	131.2	638.6	533.2	542.1	499.1	430.1	485.1	579.7	540.2	528.9

EXPERIMENTAL RESULTS TEST 15 (T15)

Feed X Position	325.0	335.0	345.0	355.0	365.0	375.0	385.0	395.0	405.0	415.0
Spindle Speed	525.7	466.6	475.8	478.9	477.5	479.3	481.1	482.1	482.5	484.8
Spindle Power	2001.1	5214.8	5106.5	5055.4	5037.6	5002.1	4964.1	4932.6	4928.3	4900.7
Spindle Torque	15.4	39.9	38.3	37.6	37.5	37.1	36.5	36.2	35.9	35.4
Spindle Current	5.2	13.7	13.4	13.3	13.2	13.1	13.0	12.9	12.9	12.8
Feedrate	8.4	38.3	38.3	38.3	38.2	38.2	38.2	38.3	38.3	33.1
Feed Power	107.6	240.4	240.2	239.7	239.1	238.2	237.4	238.4	238.5	214.2
Feed Torque	1.4	1.5	1.5	1.5	1.6	1.6	1.6	1.6	1.6	1.7
Feed Current	1.5	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
Tool Temp.	57.7	315.7	355.8	381.0	398.2	412.4	423.7	432.9	440.3	446.8
Tool Fx/Fy	0.2	1.1	1.1	1.1	1.0	1.1	4.1	1.0	1.1	1.1
Tool Fz	-4.0	-16.4	-16.3	-15.7	-15.9	-15.8	-15.6	-15.5	-15.5	-14.8
0 Degrees	1.2	71.1	99.8	133.4	126.4	169.4	195.3	162.7	174.2	159.3
10 Degrees	0.0	62.4	90.2	114.2	112.0	155.0	209.0	134.9	156.7	142.5
20 Degrees	1.5	44.0	68.6	79.8	78.7	124.0	207.1	96.0	116.6	107.1
30 Degrees	3.4	7.4	26.3	33.8	31.1	78.5	197.7	42.4	67.9	55.6
40 Degrees	5.7	35.1	23.7	17.1	28.0	21.2	175.0	21.1	8.4	4.9
50 Degrees	8.2	81.9	80.7	73.5	92.9	43.2	155.0	88.1	57.6	69.8
60 Degrees	10.5	135.9	144.1	134.7	160.4	114.4	135.3	159.1	125.0	141.1
70 Degrees	13.1	188.2	208.6	198.3	229.2	189.5	101.4	235.4	198.7	215.7
80 Degrees	15.0	241.6	274.6	263.3	298.3	262.7	49.2	310.8	275.0	289.6
90 Degrees	16.8	297.6	338.9	329.5	364.2	335.7	14.3	382.9	345.3	361.1
100 Degrees	18.4	347.1	397.3	391.2	427.6	407.0	82.8	449.9	412.3	428.3
110 Degrees	19.2	394.2	453.3	448.2	486.4	470.0	155.9	510.7	472.5	489.5
120 Degrees	19.5	437.5	502.4	496.5	536.4	525.1	231.8	561.9	522.2	542.6
130 Degrees	18.9	469.9	541.8	536.0	576.2	570.4	306.1	602.3	565.2	585.6
140 Degrees	17.3	494.3	573.0	564.8	604.9	603.4	377.7	632.6	597.7	617.1
150 Degrees	14.9	509.6	589.9	580.6	619.6	624.0	444.3	648.0	615.3	634.3
160 Degrees	11.8	512.8	593.7	583.0	620.8	629.3	504.2	647.1	620.2	638.0
170 Degrees	8.1	506.2	587.4	572.3	609.6	621.1	555.0	634.0	610.2	628.4
180 Degrees	3.9	490.6	566.6	550.5	585.9	600.4	595.0	607.4	588.7	604.4
190 Degrees	0.3	465.2	536.4	517.2	549.9	564.2	624.1	568.4	554.3	568.3
200 Degrees	4.4	431.7	497.6	473.1	502.4	519.0	640.4	519.0	506.7	522.3
210 Degrees	8.4	391.6	448.4	423.0	447.5	465.9	639.6	459.8	453.1	466.6
220 Degrees	11.7	345.0	393.7	364.3	386.5	401.1	626.7	393.4	390.9	402.7

230 Degrees	14.3	293.5	333.3	300.8	319.7	333.8	601.2	321.4	321.9	335.6
240 Degrees	16.5	240.4	269.2	237.0	251.7	264.5	560.6	248.4	253.7	265.7
250 Degrees	18.0	186.9	206.2	170.7	182.6	190.4	511.0	174.0	182.6	194.0
260 Degrees	18.6	133.4	143.7	107.6	114.6	120.8	453.6	99.9	109.9	124.6
270 Degrees	18.8	83.8	83.2	48.6	50.8	56.0	385.0	31.7	43.9	58.5
280 Degrees	18.1	37.0	27.5	6.7	8.2	7.3	314.2	30.8	16.9	6.3
290 Degrees	17.2	1.9	22.7	54.9	59.8	59.9	242.2	86.2	72.8	62.4
300 Degrees	16.0	34.3	66.8	96.2	103.9	106.4	166.3	132.1	120.3	109.2
310 Degrees	14.7	62.7	102.6	129.4	139.7	146.1	94.3	167.4	158.4	148.3
320 Degrees	13.2	84.5	129.0	150.0	163.3	172.1	27.5	190.2	185.1	176.2
330 Degrees	10.9	101.5	146.9	162.2	177.6	190.0	36.0	203.6	202.4	191.6
340 Degrees	8.4	104.4	154.1	162.5	183.4	195.8	90.2	206.7	207.8	197.1
350 Degrees	6.6	100.2	152.5	154.1	178.1	190.9	136.4	198.0	203.8	190.7

EXPERIMENTAL RESULTS TEST 16 (T16)

Feed X Position	470.0	480.0	490.0	500.0	510.0	520.0	530.0	540.0	550.0	560.0
Spindle Speed	511.1	302.1	299.1	299.2	301.0	299.1	299.8	300.9	298.9	300.0
Spindle Power	1737.3	4513.5	4496.0	4414.2	4426.4	4505.6	4521.0	4643.3	4671.2	4747.4
Spindle Torque	15.7	54.8	55.2	54.9	54.4	55.2	55.5	55.7	56.6	57.0
Spindle Current	4.7	13.5	13.5	13.2	13.2	13.4	13.4	13.6	13.7	13.8
Feedrate	9.0	96.8	96.8	96.7	96.8	96.7	96.7	96.8	96.7	66.7
Feed Power	114.4	545.8	549.2	547.4	548.2	548.4	547.1	551.6	546.6	407.6
Feed Torque	1.4	1.7	1.6	1.7	1.7	1.7	1.7	1.7	1.8	1.7
Feed Current	1.4	2.5	2.6	2.6	2.6	2.6	2.6	2.6	2.5	2.6
Tool Temp.	47.7	259.2	288.9	312.7	329.8	344.9	356.7	366.4	375.1	384.3
Tool Fx/Fy	0.3	1.1	0.9	0.9	0.9	0.9	3.1	0.9	0.9	0.9
Tool Fz	-3.6	-19.5	-19.3	-19.4	-19.3	-19.0	-19.2	-19.9	-20.1	-20.3
0 Degrees	1.2	86.1	105.0	117.5	122.5	116.7	104.8	110.8	103.1	147.1
10 Degrees	0.9	93.5	124.5	138.8	140.0	144.8	63.5	127.7	128.1	161.7
20 Degrees	0.1	141.0	175.7	193.0	189.5	200.8	18.9	176.0	181.5	198.1
30 Degrees	1.0	198.5	233.4	253.7	251.1	263.0	25.7	236.9	241.9	238.6
40 Degrees	2.2	259.8	293.5	316.0	315.6	327.8	75.0	301.9	307.7	280.1
50 Degrees	3.7	321.4	351.3	376.8	378.1	391.3	100.0	366.4	373.2	321.3
60 Degrees	5.2	378.2	404.5	431.6	435.0	449.3	141.1	424.4	430.4	359.9
70 Degrees	6.6	428.7	451.4	480.5	484.3	498.2	195.5	474.5	480.3	393.2
80 Degrees	7.9	472.4	490.9	520.4	525.9	540.0	252.9	516.8	521.4	422.7
90 Degrees	8.9	509.5	522.4	553.2	559.5	572.0	320.4	551.4	555.1	446.6
100 Degrees	9.5	538.9	544.0	574.9	581.6	593.4	384.7	576.1	578.3	463.4
110 Degrees	9.6	556.9	555.1	585.5	591.7	602.5	444.4	587.3	587.8	471.8
120 Degrees	8.9	562.5	553.3	583.0	588.7	596.8	494.6	583.0	584.4	470.4
130 Degrees	7.4	554.9	539.4	567.7	572.0	580.1	536.5	566.8	566.4	458.9
140 Degrees	4.9	534.8	513.3	538.5	542.5	548.3	570.0	539.1	535.1	437.2
150 Degrees	1.7	501.3	475.6	497.7	499.1	505.7	593.1	498.3	493.6	405.9
160 Degrees	2.1	456.0	426.2	445.5	445.1	447.1	602.4	444.8	440.7	366.2
170 Degrees	6.1	400.3	369.1	384.7	380.2	381.9	597.2	382.4	378.8	319.9
180 Degrees	10.2	338.3	305.0	317.3	310.5	308.8	578.2	312.8	307.5	268.9
190 Degrees	13.9	271.3	239.4	247.5	238.3	234.7	547.3	241.6	234.4	216.1
200 Degrees	17.3	204.2	173.6	177.4	166.5	157.9	502.8	168.5	160.3	163.5
210 Degrees	20.0	137.1	109.8	109.0	96.5	85.9	445.9	98.2	90.3	112.3
220 Degrees	22.1	74.8	50.8	46.5	31.6	19.0	379.1	32.1	24.1	65.9

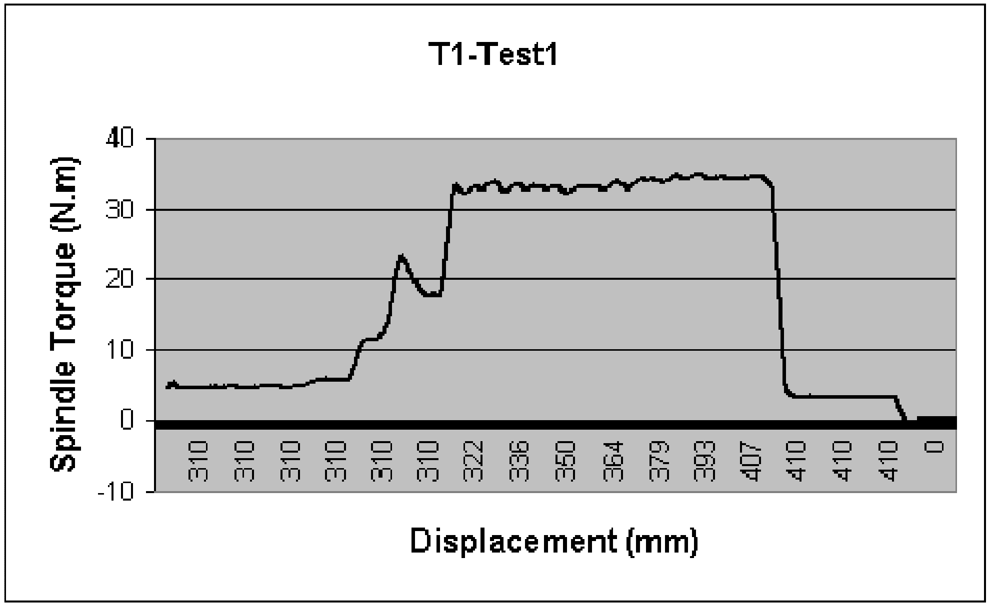
230 Degrees	23.3	15.7	2.1	8.3	25.9	39.4	305.9	25.3	33.3	25.2
240 Degrees	23.6	35.0	47.5	54.2	74.3	90.1	230.8	76.3	83.1	9.0
250 Degrees	22.9	78.2	84.8	92.7	113.4	129.2	154.8	116.2	122.3	36.5
260 Degrees	21.5	111.2	112.2	121.6	142.0	158.1	81.9	145.2	150.6	55.6
270 Degrees	19.6	135.0	129.0	142.4	161.0	175.7	13.8	164.1	168.5	65.5
280 Degrees	17.4	148.7	135.5	150.6	169.3	181.6	45.4	172.3	174.0	67.5
290 Degrees	15.0	153.1	132.8	148.6	164.7	176.4	95.9	169.0	169.1	61.9
300 Degrees	12.6	146.9	119.9	137.5	150.1	160.1	135.8	154.9	154.9	49.2
310 Degrees	10.5	130.8	98.2	115.1	126.1	133.8	164.3	134.0	129.0	29.5
320 Degrees	8.4	108.1	68.8	84.3	93.4	98.8	182.3	102.1	95.0	4.2
330 Degrees	6.7	76.6	35.5	46.5	54.3	57.5	188.4	64.6	53.8	25.2
340 Degrees	5.2	40.1	1.0	4.7	10.8	11.0	182.6	27.6	9.9	55.2
350 Degrees	3.9	0.2	35.7	37.0	29.0	36.3	165.2	11.4	34.8	81.2

D.2: Experimental Results used for Statistical Analysis

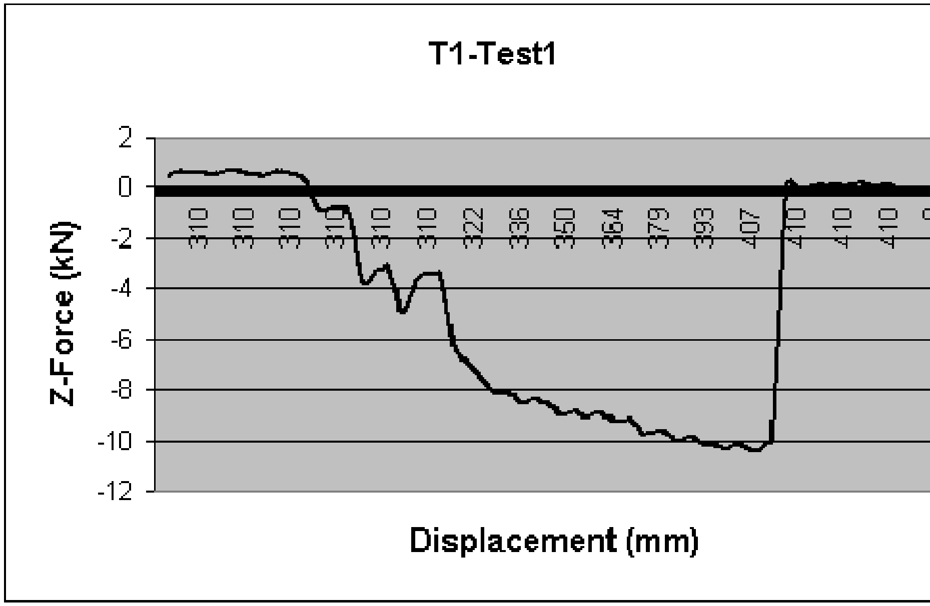
Test No	Spindle Power (W) [Ps]	Spindle Torque (N.m) [Ts]	Spindle Current (A) [Cs]	Feed Power (W) [Pf]	Feed Torque (N.m) [Tf]	Feed Current (A) [Cf]	Tool Temp (oC) [T]	Force X (N) [Fx]	Force Y (N) [Fy]	Fx / Fy [R]	Z-Force (kN) [Fz]
1	2432.20	33.90	8.50	240.00	1.30	2.60	461.30	582.20	276.50	2.10	-9.70
2	5279.80	40.70	13.90	544.10	1.70	2.50	460.40	511.80	184.00	2.78	-19.50
3	3408.40	37.10	9.60	336.20	1.80	2.50	501.70	695.40	175.10	3.97	-18.10
4	4325.30	35.60	11.20	442.90	1.70	2.60	477.40	728.50	202.20	3.60	-15.60

5	2526.40	35.80	8.70	453.20	1.60	2.60	410.70	847.50	107.40	7.89	-13.20
6	3012.70	23.40	7.70	341.90	1.80	2.60	449.60	647.00	66.40	9.74	-13.40
7	4235.50	43.70	11.20	541.30	3.80	2.50	496.40	1891.30	166.50	11.40	-28.20
8	5187.60	39.30	13.60	219.70	3.10	2.40	477.40	1220.10	95.50	12.78	-19.80
9	4238.30	35.40	11.00	545.80	2.90	2.50	414.50	1210.00	132.90	9.10	-23.70
10	2595.10	29.50	7.60	234.00	1.80	2.60	496.40	949.80	153.00	6.21	-14.70
11	5582.40	46.60	14.70	435.90	3.60	2.50	449.60	1724.50	70.90	24.30	-25.00
12	3268.10	43.30	10.60	328.00	2.30	2.50	410.70	1482.30	223.80	6.62	-16.50
13	2808.00	26.30	7.20	335.90	1.60	2.50	489.00	580.50	211.30	2.75	-11.30
14	3358.50	36.60	9.50	441.80	1.60	2.50	472.70	485.10	288.20	1.68	-17.10
15	4964.10	36.50	13.00	237.40	1.60	2.60	523.70	640.40	242.20	2.64	-15.60
16	4521.00	55.50	13.40	547.10	1.70	2.60	456.70	602.40	154.80	3.89	-19.20

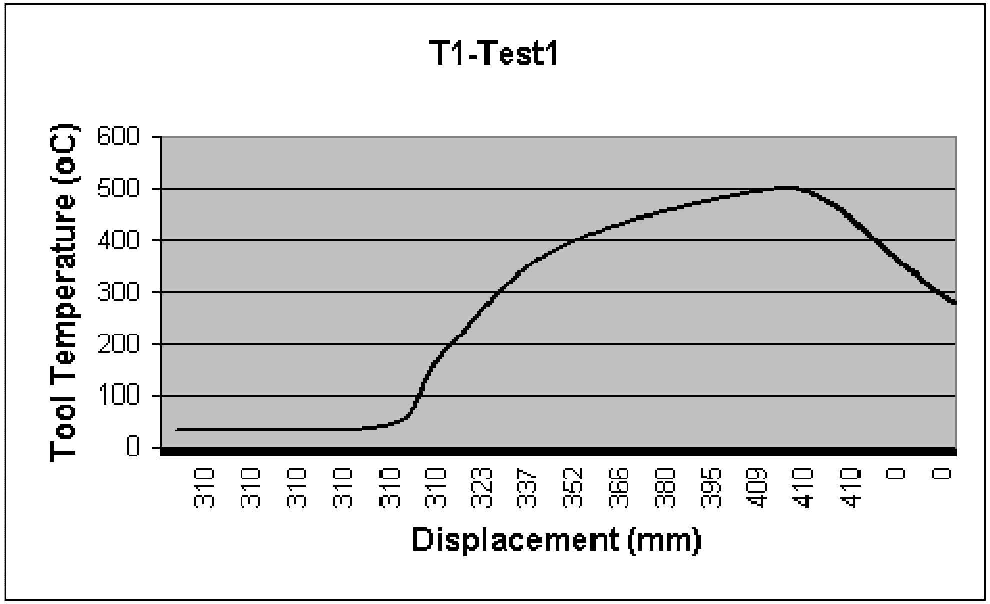
*** All welds made with a tool tilt angle of 2.5 degrees.**



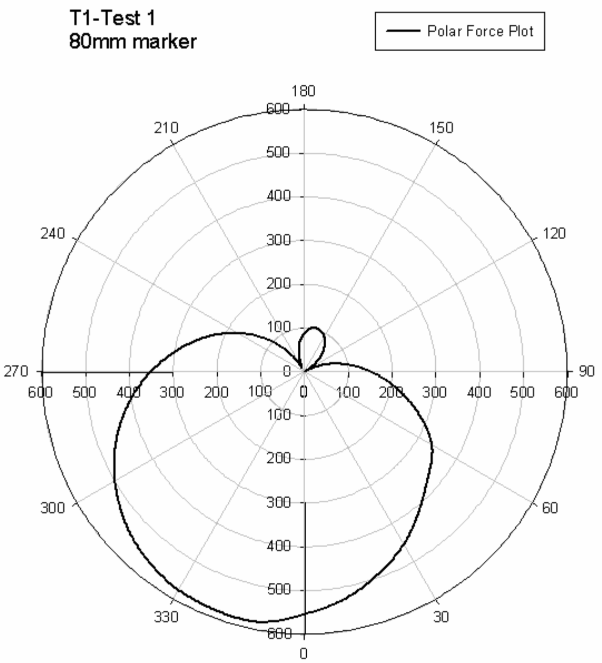
T1-Test1



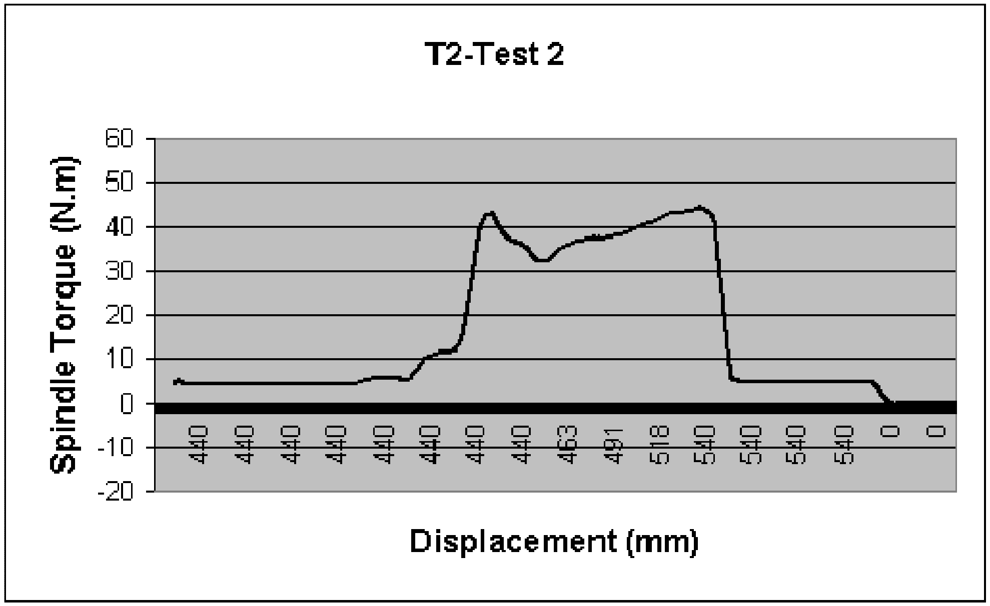
T1-Test1

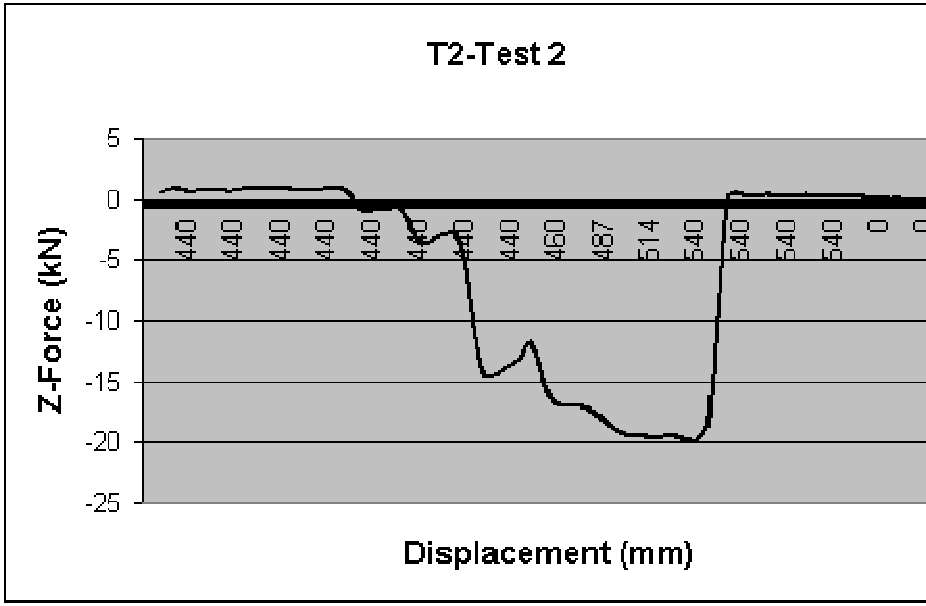


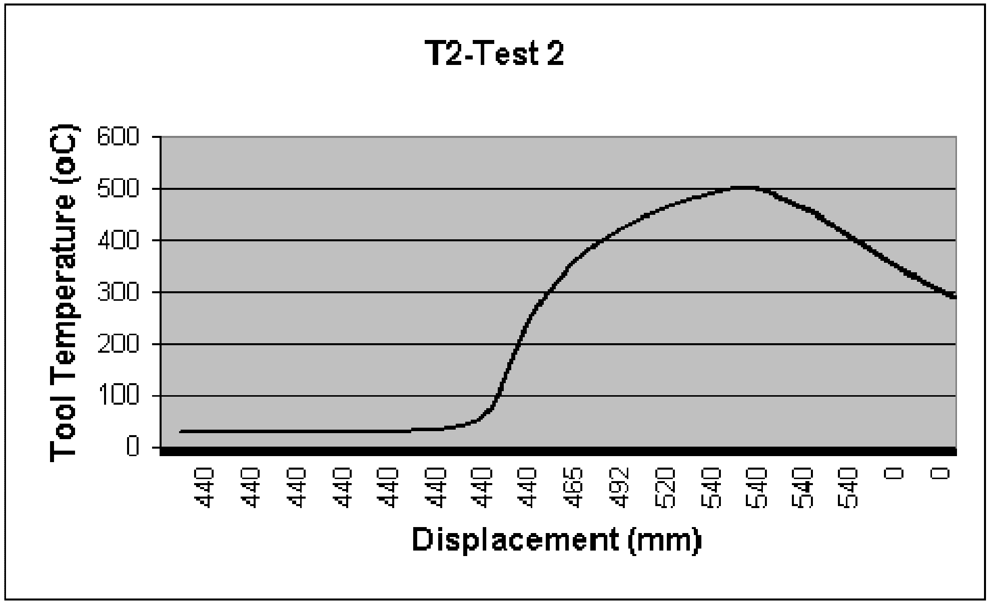
T1-Test1

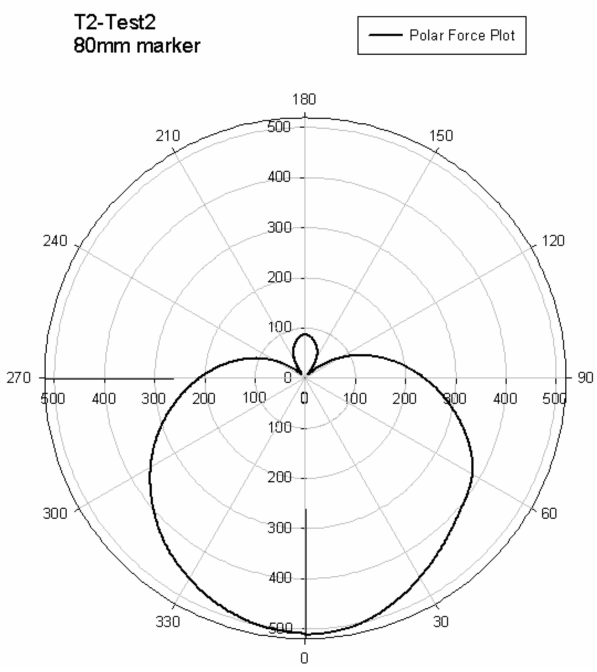


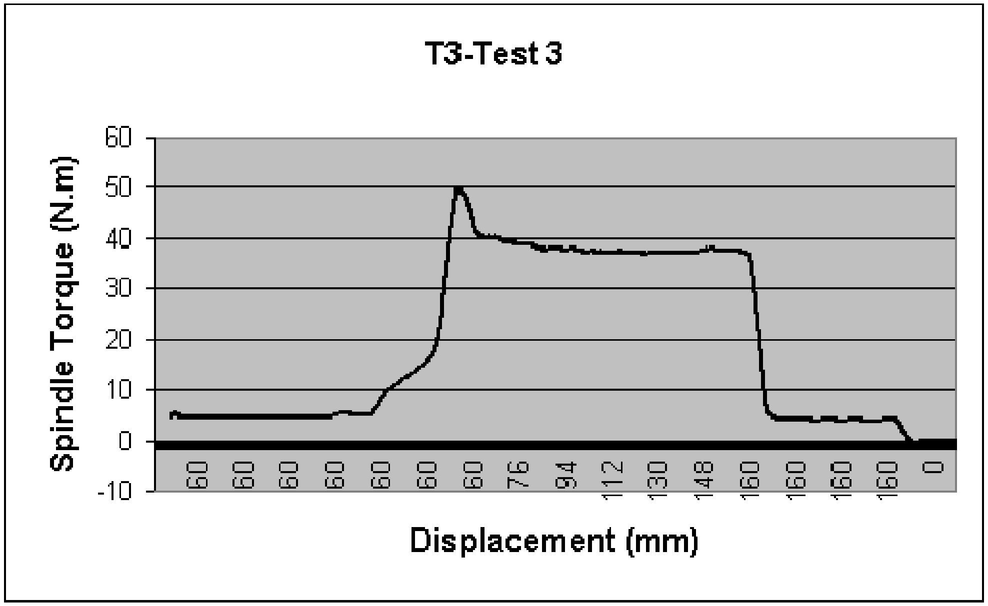
T1-Test 1
80mm marker

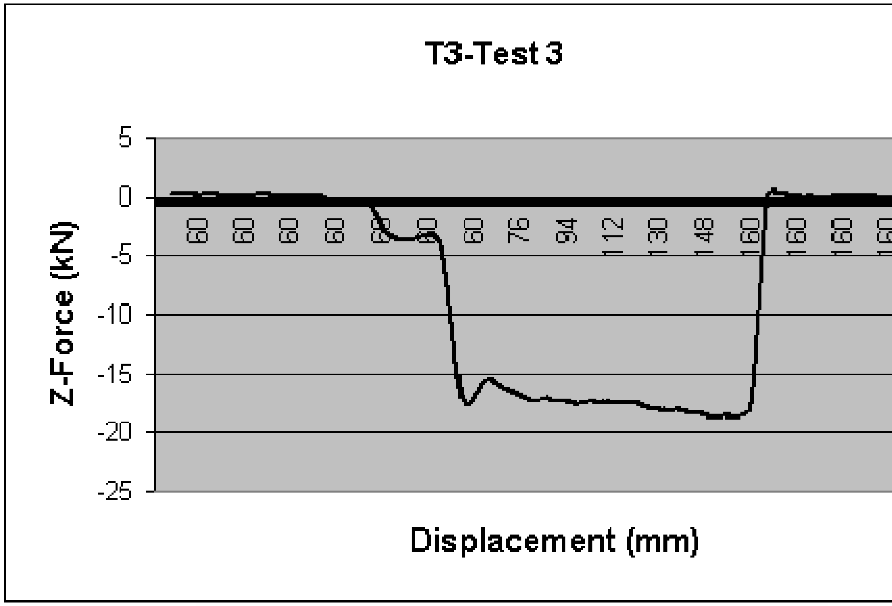

T2-Test 2
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
-20
440
440
440
440
440
440
440
440
463
491
518
540
540
540
540
0
0
Displacement (mm)

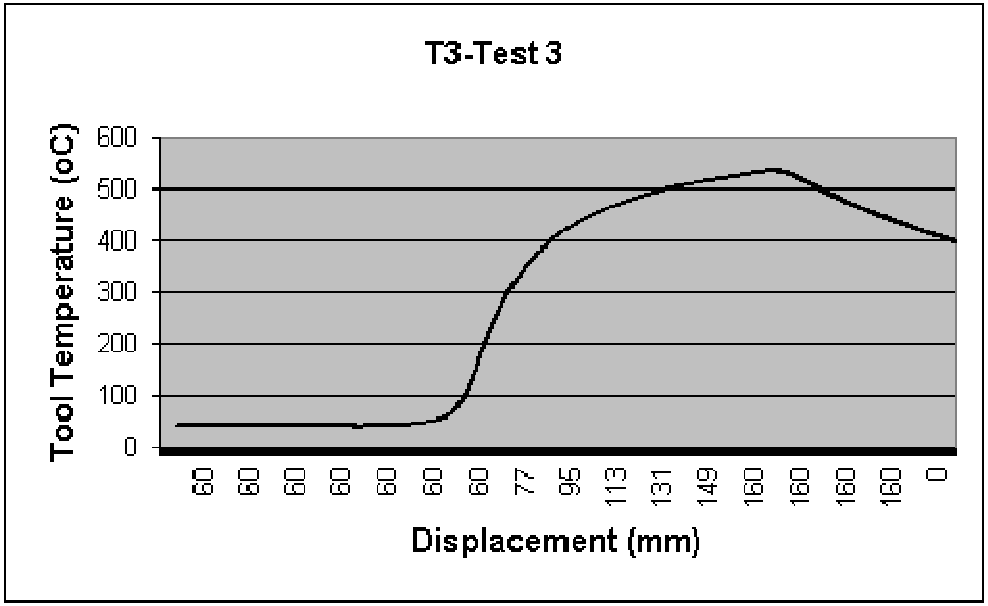

T2-Test 2
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
440
440
440
440
440
440
440
440
460
487
514
540
540
540
540
0
Displacement (mm)

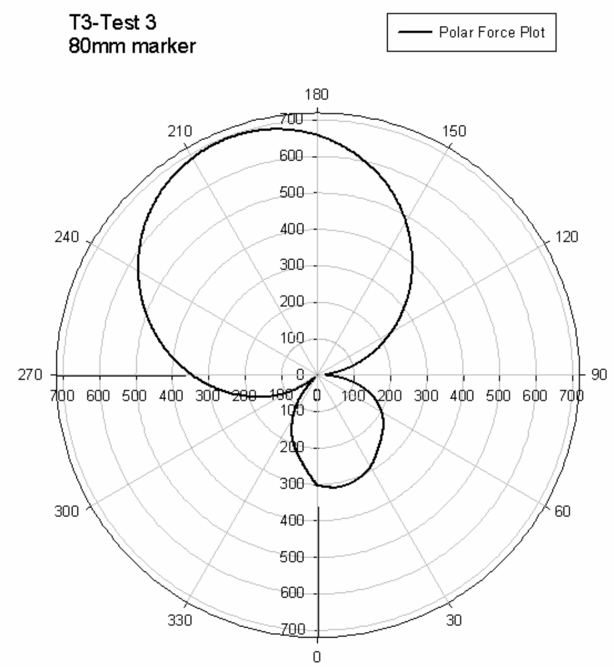

T2-Test 2
Tool Temperature (oC)
600
500
400
300
200
100
0
440
440
440
440
440
440
440
440
465
492
520
540
540
540
540
0
0
Displacement (mm)

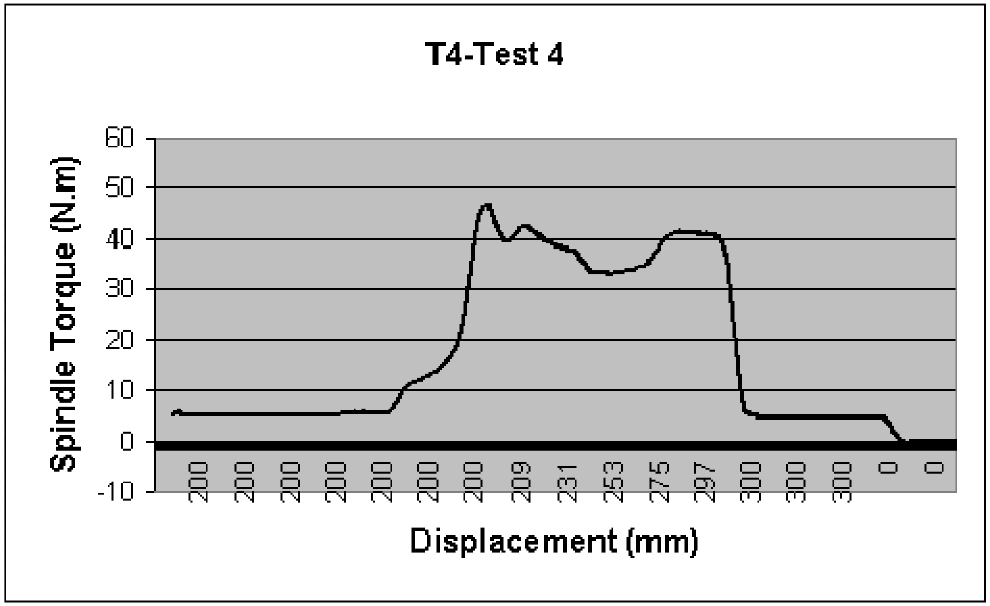

T2-Test2
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
500
400
300
200
100
0

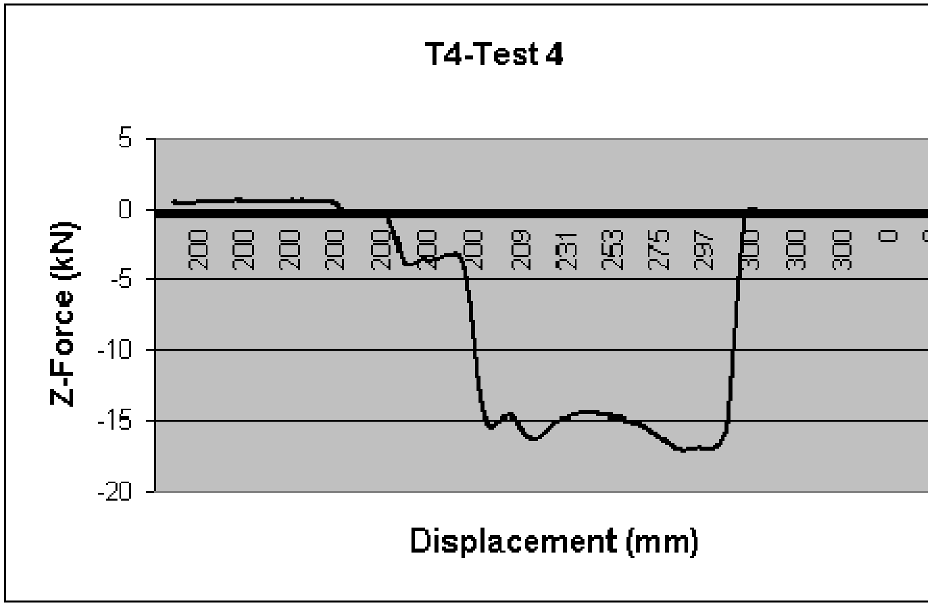

T3-Test 3
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
60
60
60
60
60
60
60
76
94
112
130
148
160
160
160
160
0
Displacement (mm)

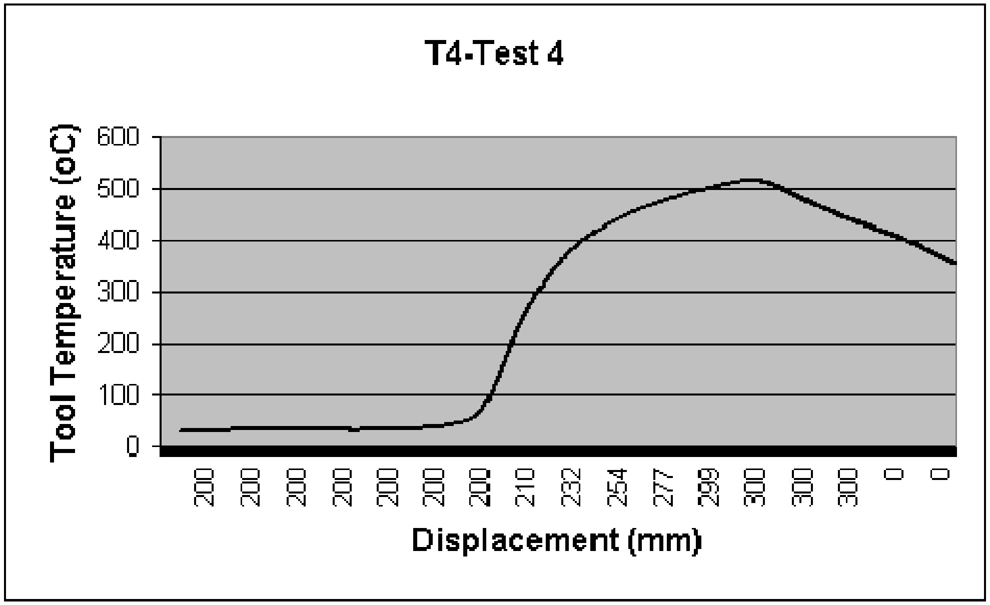

T3-Test 3
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
60
60
60
60
60
60
60
76
94
112
130
148
160
160
160
160
Displacement (mm)

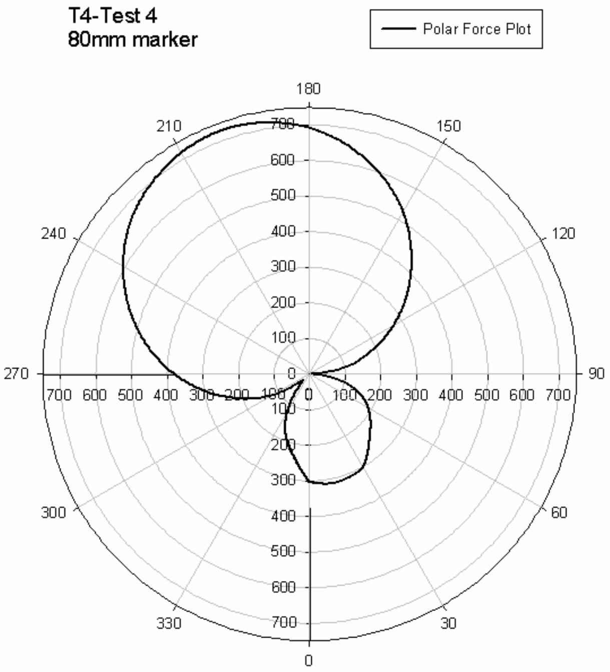

T3-Test 3
Tool Temperature (oC)
600
500
400
300
200
100
0
60
60
60
60
60
60
60
77
95
113
131
149
160
160
160
160
0
Displacement (mm)

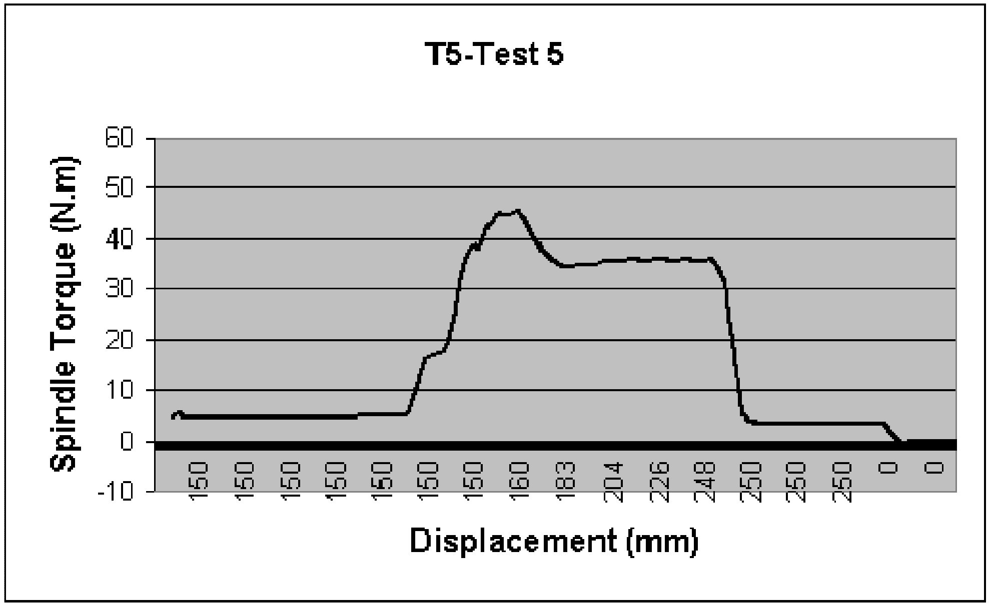

T3-Test 3
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
700
600
500
400
300
200
100
0
100
200
300
400
500
600
700

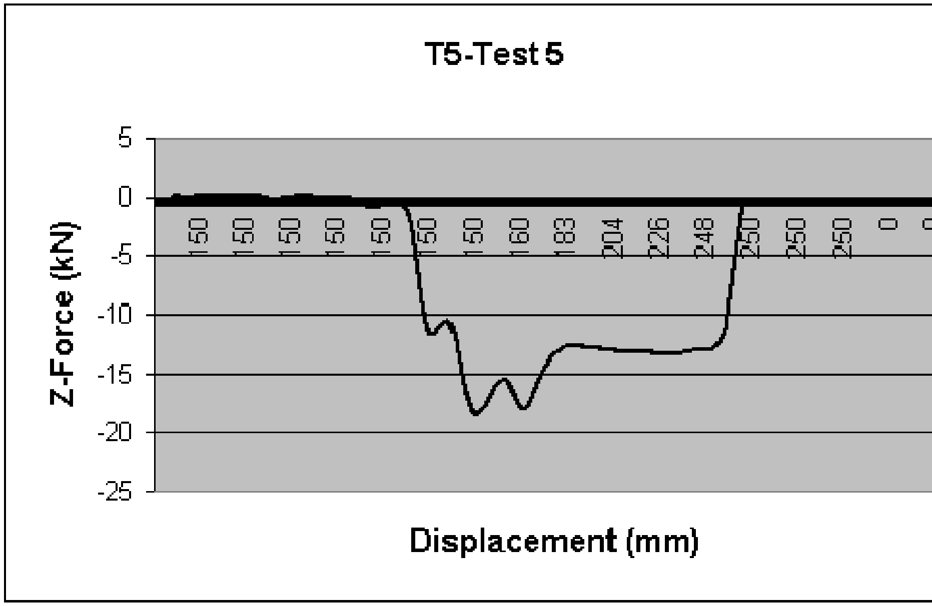

T4-Test 4

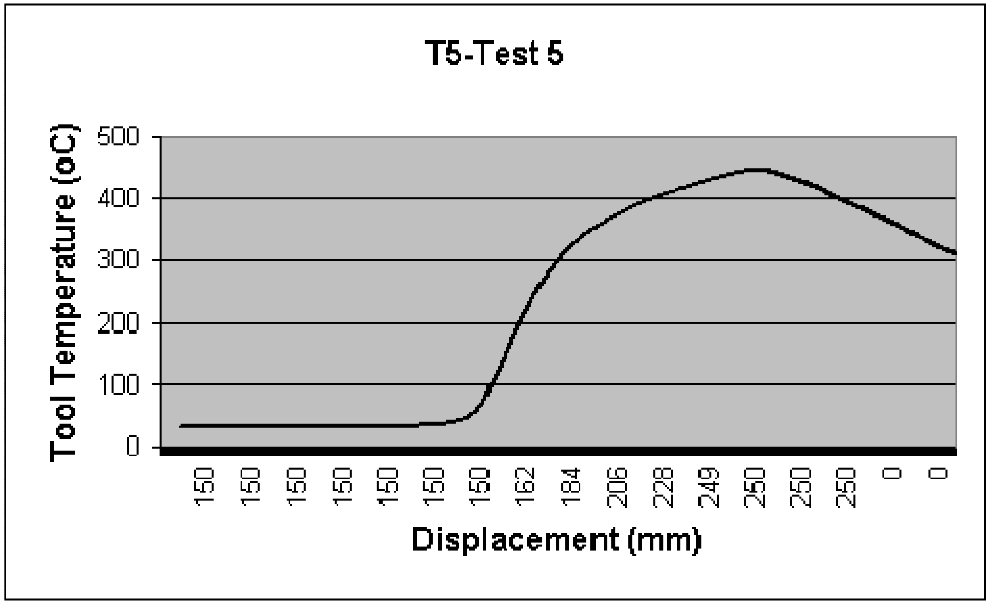

T4-Test 4

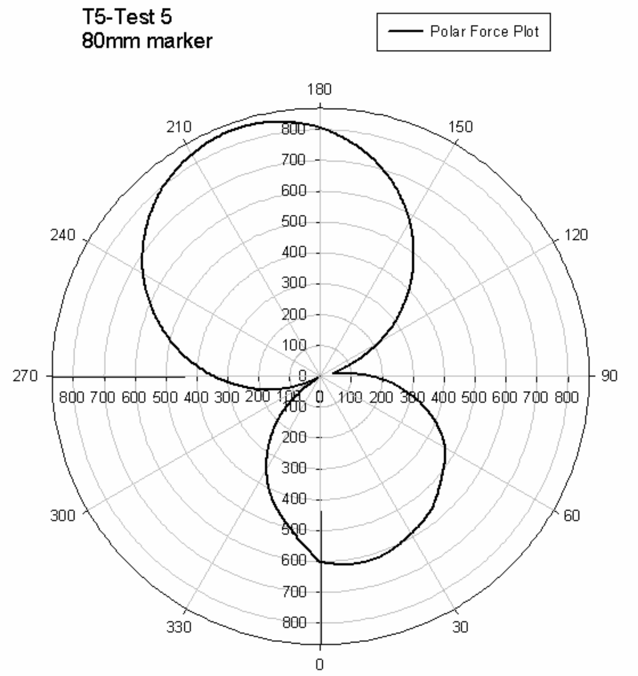

T4-Test 4

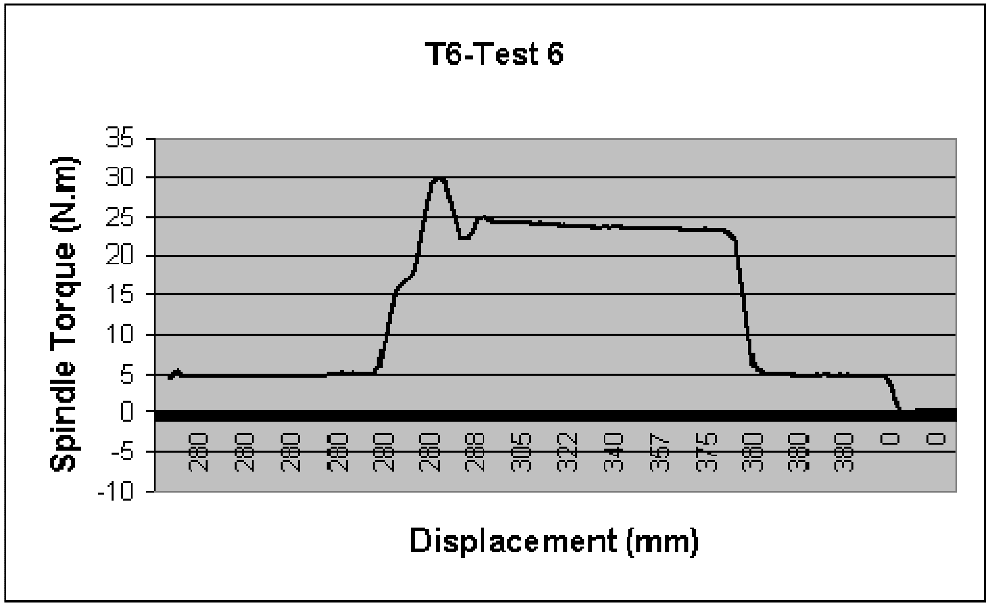

T4-Test 4
80mm marker

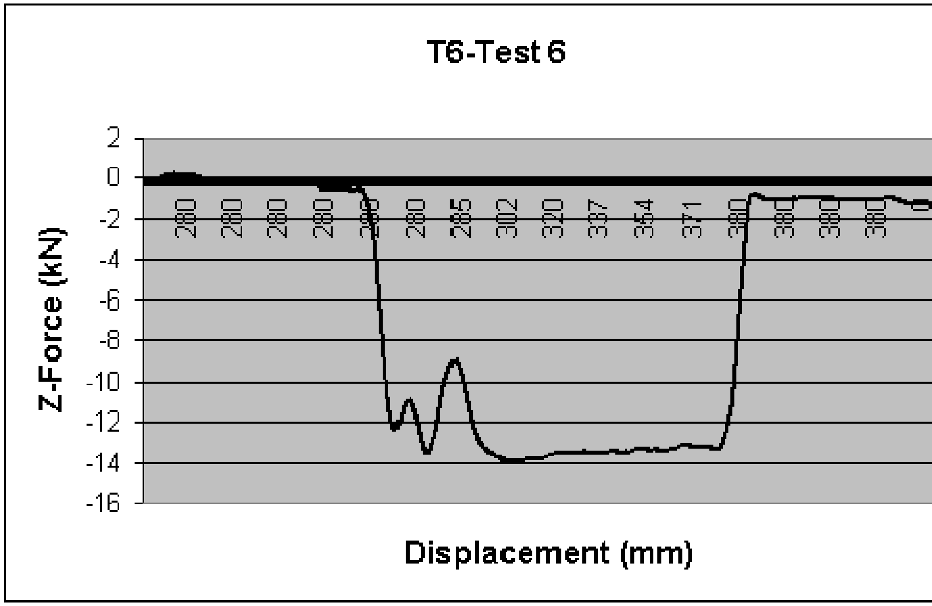

T5-Test 5
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
150
150
150
150
150
150
150
160
183
204
226
248
250
250
250
0
0
Displacement (mm)

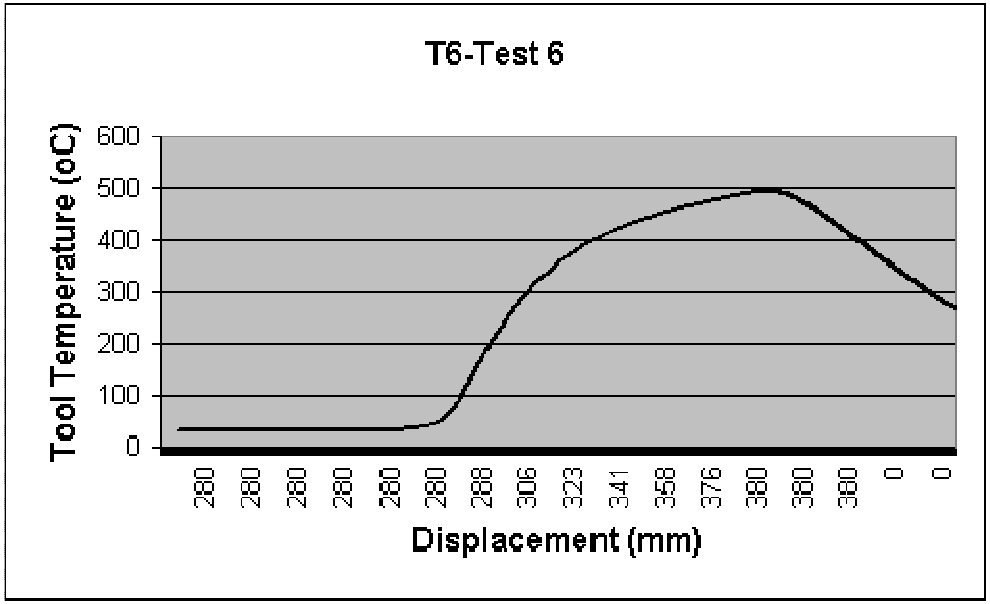

T5-Test 5
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
150
150
150
150
150
150
150
160
183
204
226
248
250
250
250
0
Displacement (mm)

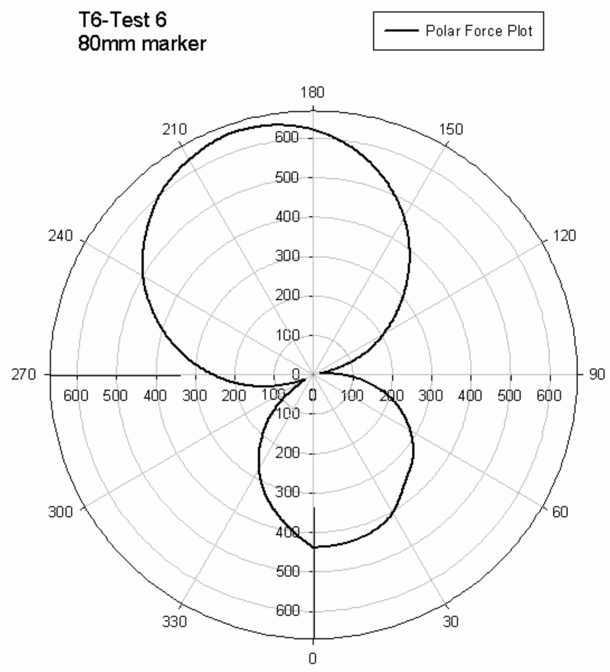

T5-Test 5
Tool Temperature (oC)
500
400
300
200
100
0
150
150
150
150
150
150
150
162
184
206
228
249
250
250
250
0
0
Displacement (mm)

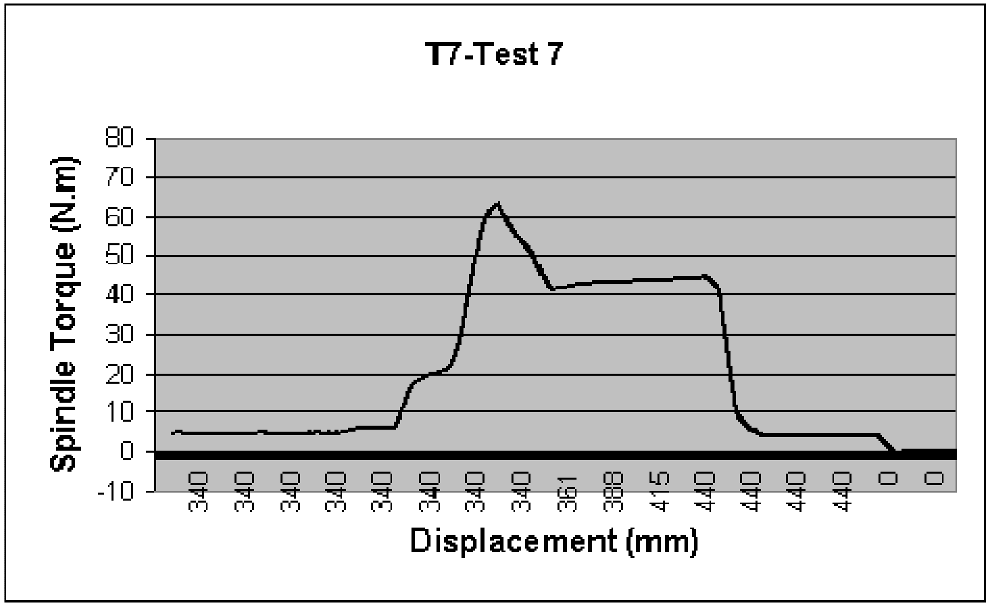

T5-Test 5
80mm marker
Polar Force Plot
180
210
150
240
120
270
90
300
60
330
30
0
800
700
600
500
400
300
200
100
0

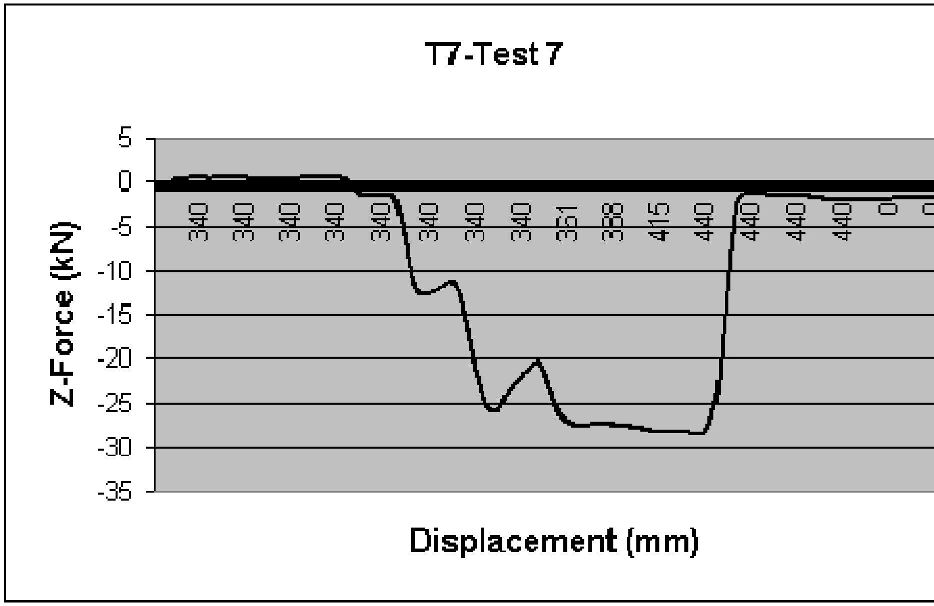

T6-Test 6
Spindle Torque (N.m)
35
30
25
20
15
10
5
0
-5
-10
280
280
280
280
280
280
288
305
322
340
357
375
380
380
380
0
0
Displacement (mm)

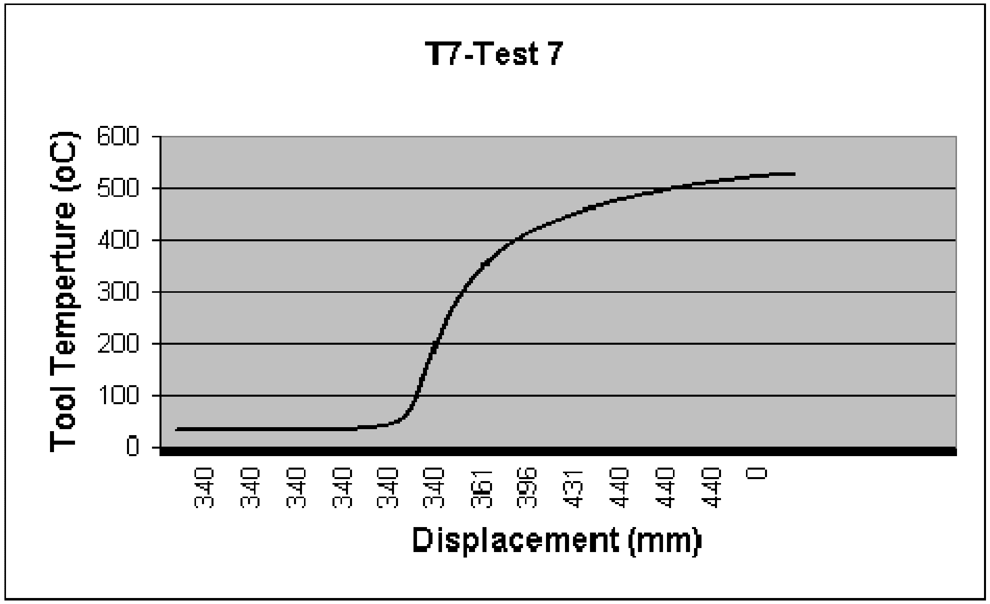

T6-Test 6
Z-Force (kN)
2
0
-2
-4
-6
-8
-10
-12
-14
-16
280
280
280
280
280
285
302
320
337
354
371
380
380
380
380
Displacement (mm)

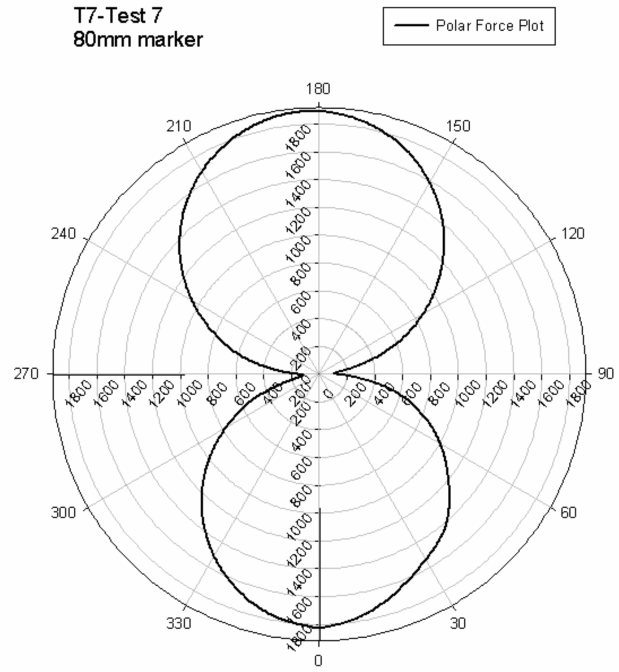

T6-Test 6
Tool Temperature (oC)
600
500
400
300
200
100
0
280
280
280
280
280
280
288
306
323
341
358
376
380
380
380
0
0
Displacement (mm)

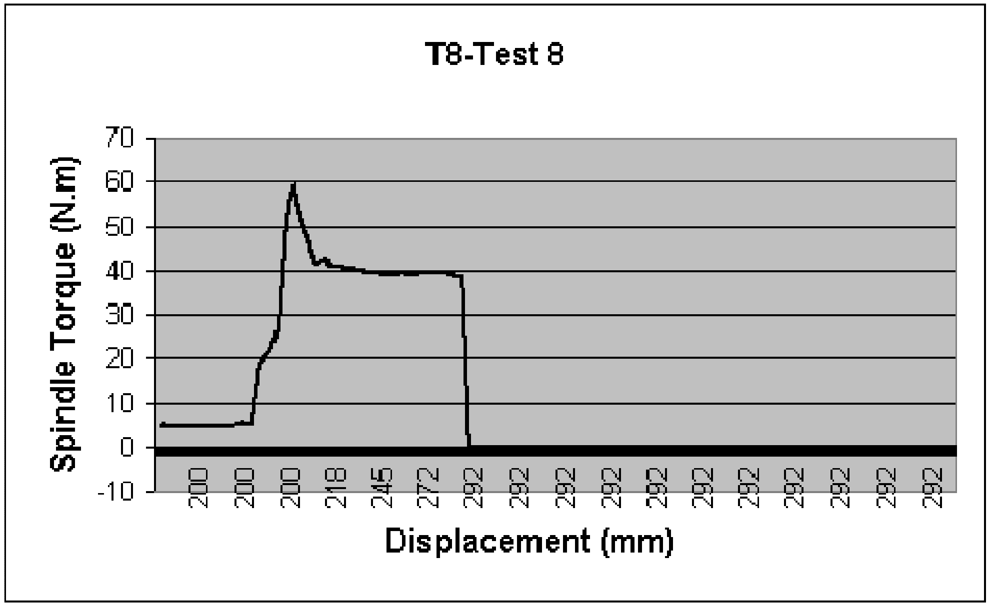

T6-Test 6
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
600
500
400
300
200
100
0
100
200
300
400
500
600

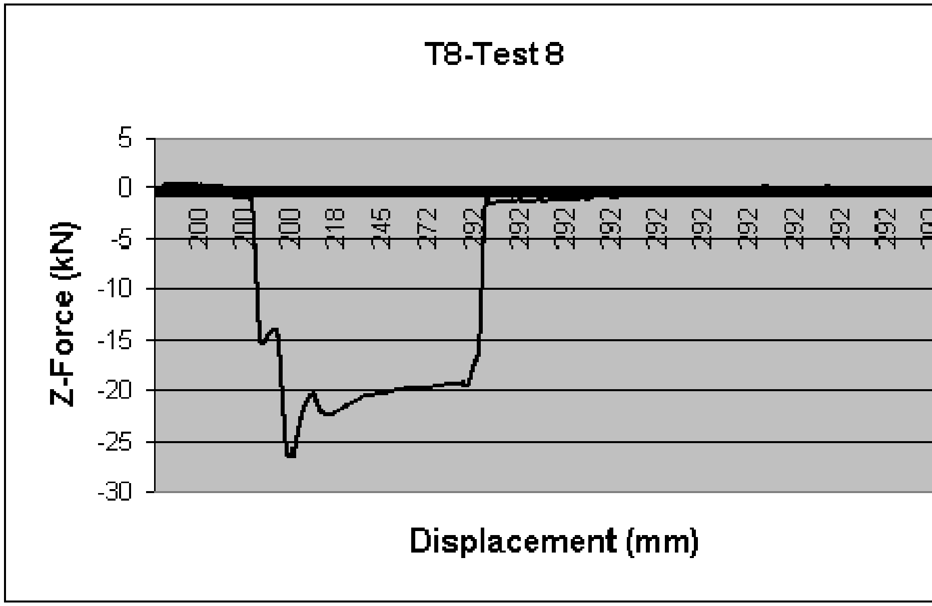

T7-Test 7
Spindle Torque (N.m)
80
70
60
50
40
30
20
10
0
-10
340
340
340
340
340
340
340
340
361
388
415
440
440
440
440
0
0
Displacement (mm)

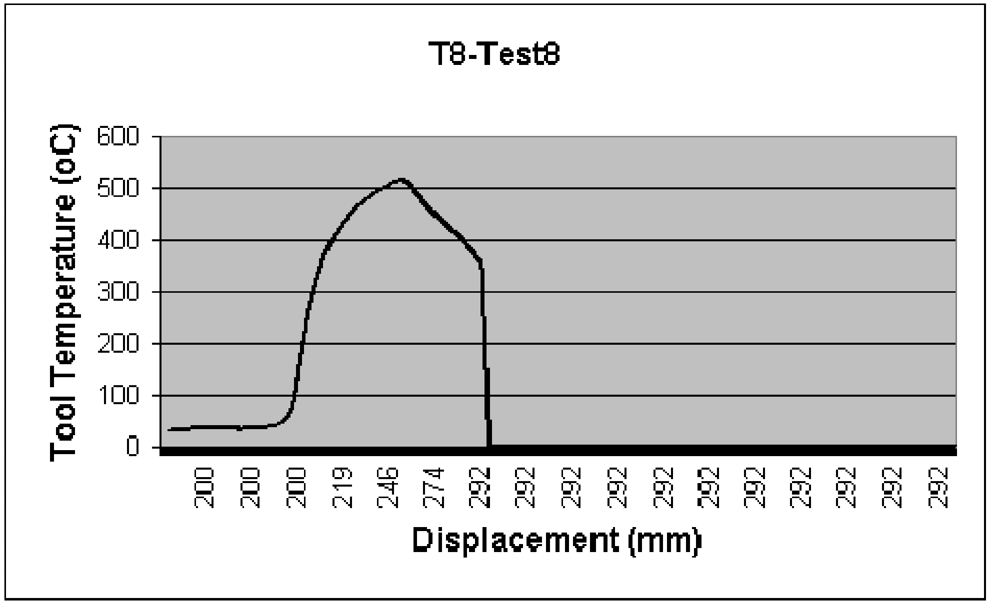

T7-Test 7
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
-30
-35
340
340
340
340
340
340
340
340
351
388
415
440
440
440
440
0
Displacement (mm)

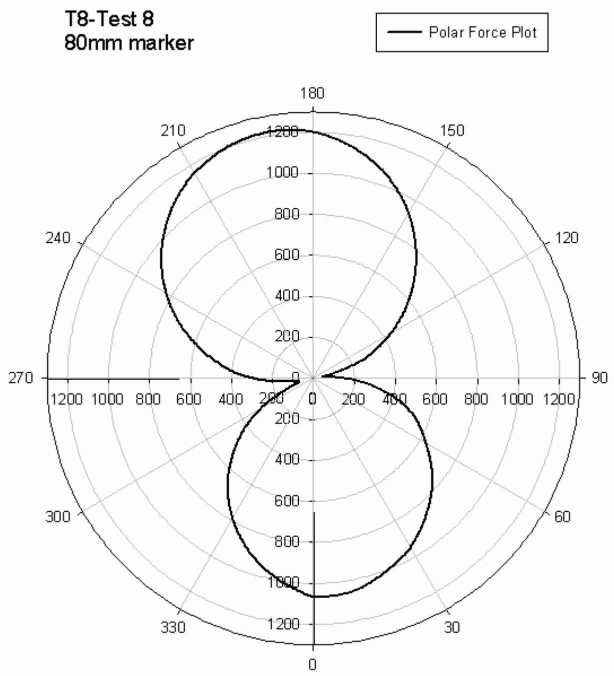

T7-Test 7
Tool Temperture (oC)
600
500
400
300
200
100
0
340
340
340
340
340
340
361
396
431
440
440
440
0
Displacement (mm)

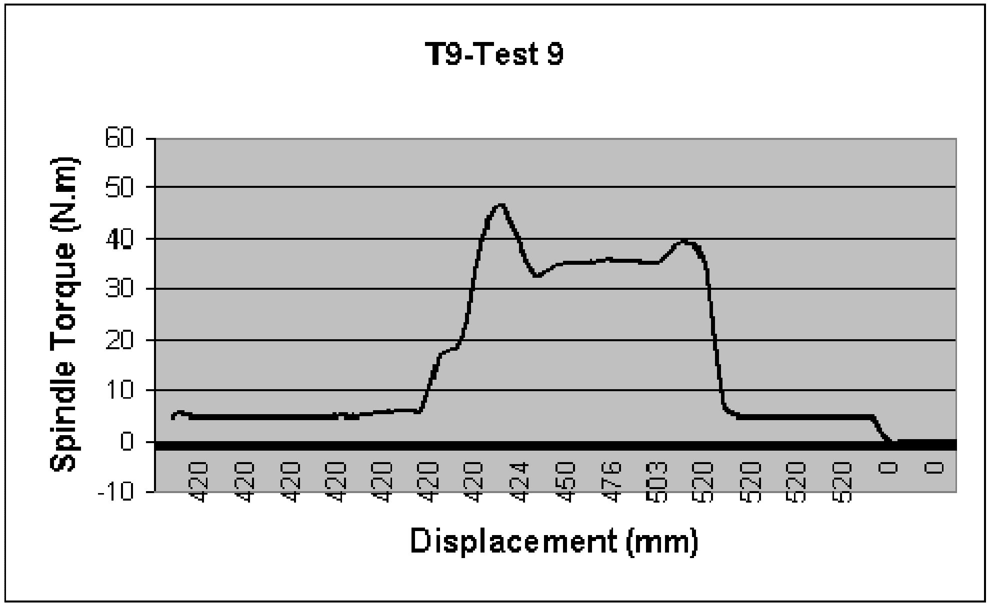

T7-Test 7
80mm marker
Polar Force Plot
180
210
150
240
120
270
90
300
60
330
30
0
1800
1600
1400
1200
1000
800
600
400
200
0

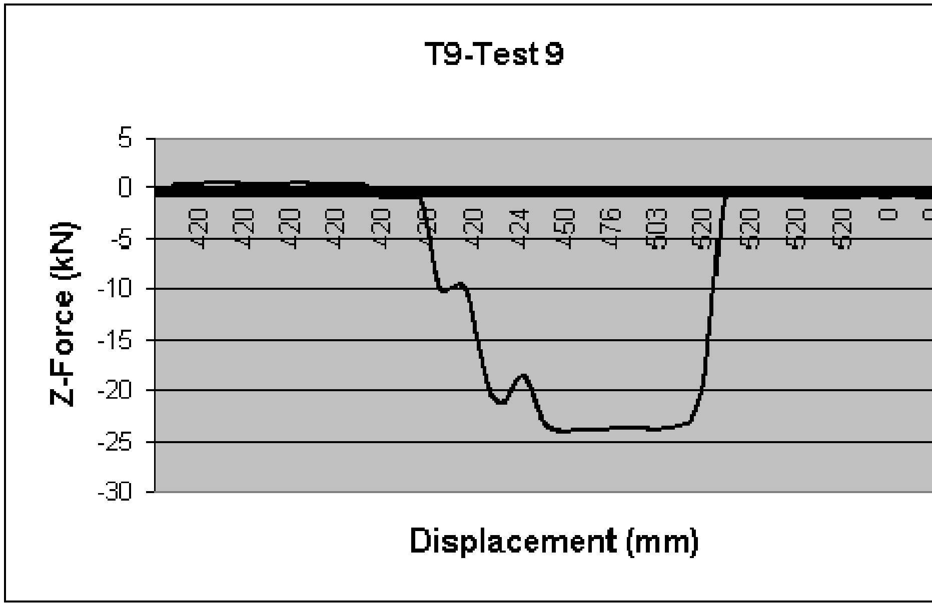

T8-Test 8
Spindle Torque (N.m)
70
60
50
40
30
20
10
0
-10
200
200
200
218
245
272
292
292
292
292
292
292
292
292
292
292
292
Displacement (mm)

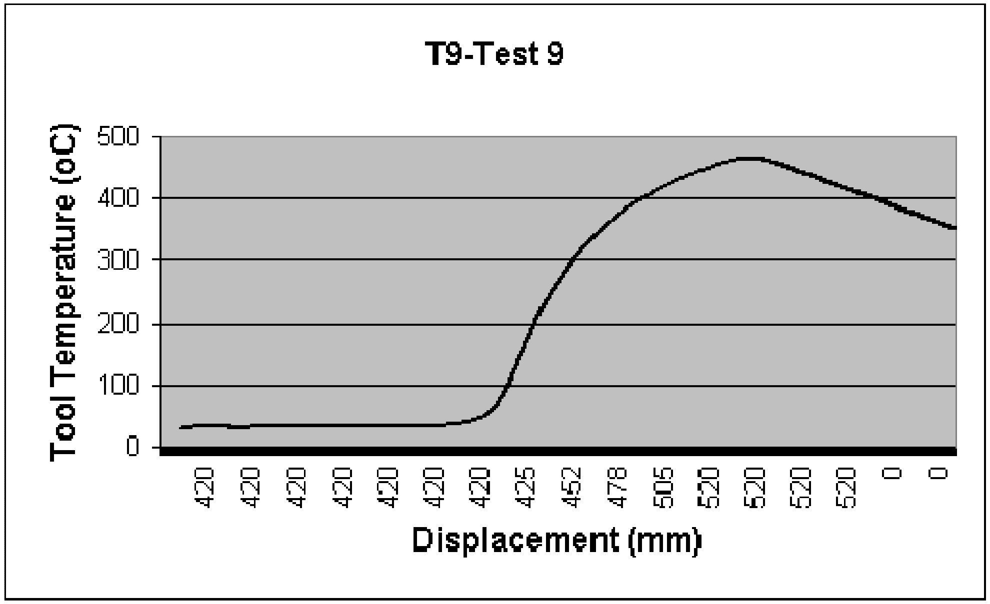

T8-Test 8
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
-30
200
200
200
218
245
272
292
292
292
292
292
292
292
292
292
292
Displacement (mm)

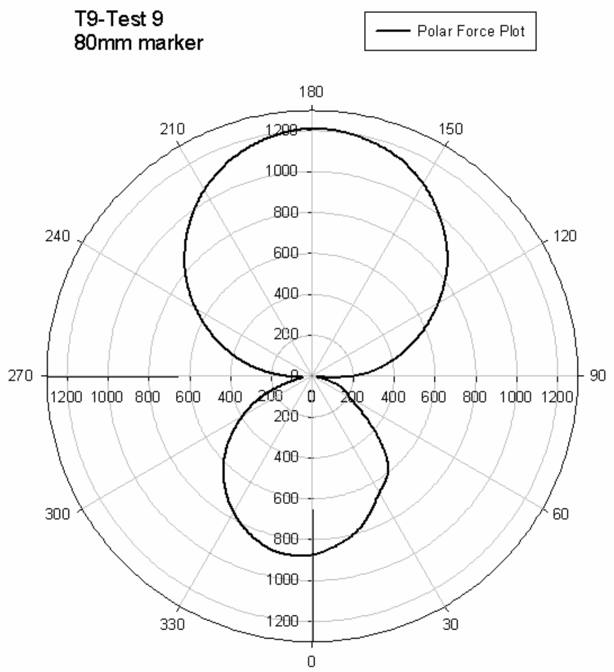

T8-Test8
Tool Temperature (oC)
600
500
400
300
200
100
0
200
200
200
219
246
274
292
292
292
292
292
292
292
292
292
292
292
Displacement (mm)

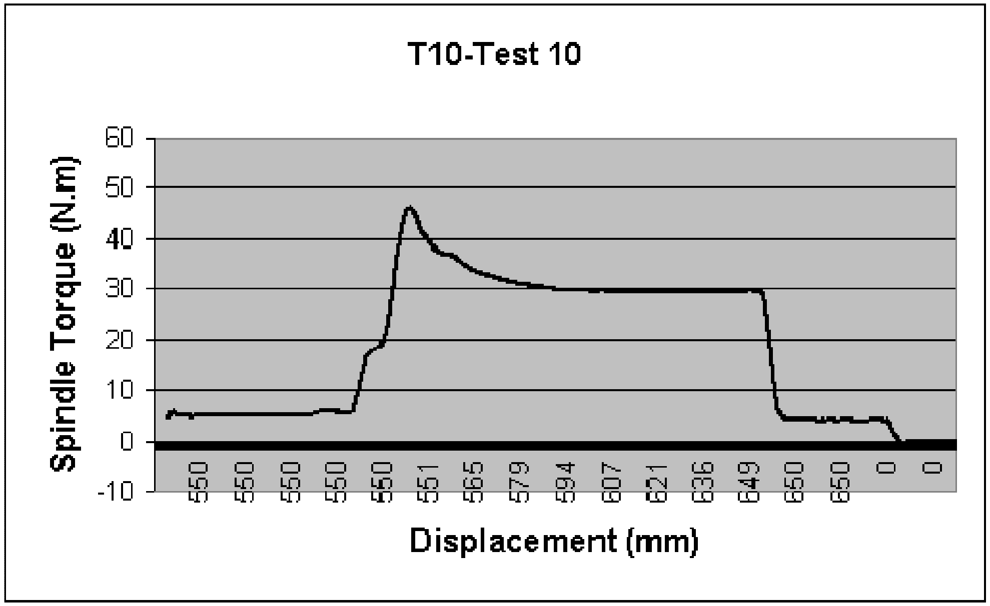

T8-Test 8
80mm marker
Polar Force Plot
180
210
150
240
120
270
90
300
60
330
30
0
1200
1000
800
600
400
200
0
200
400
600
800
1000
1200

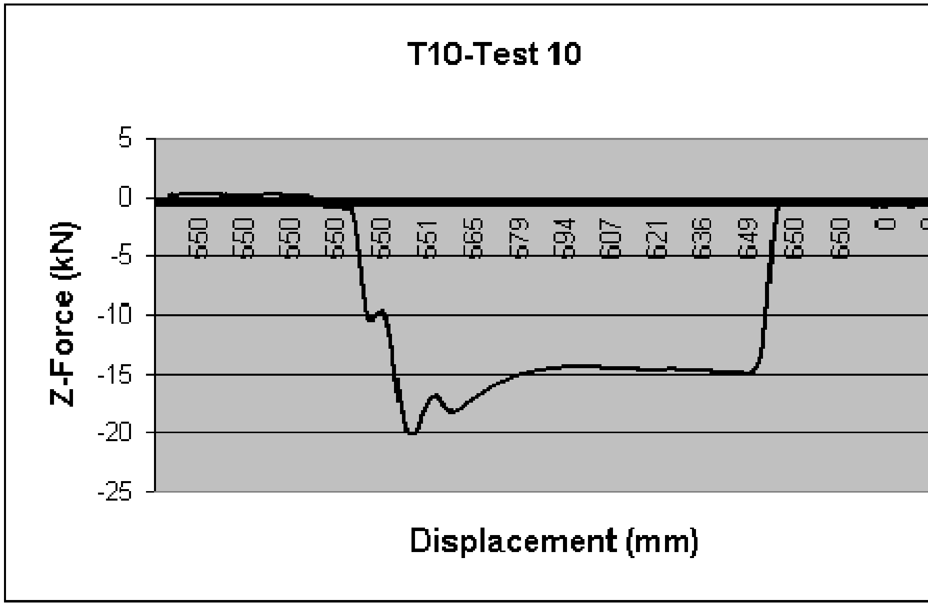

T9-Test 9
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
420
420
420
420
420
420
420
424
450
476
503
520
520
520
520
0
0
Displacement (mm)

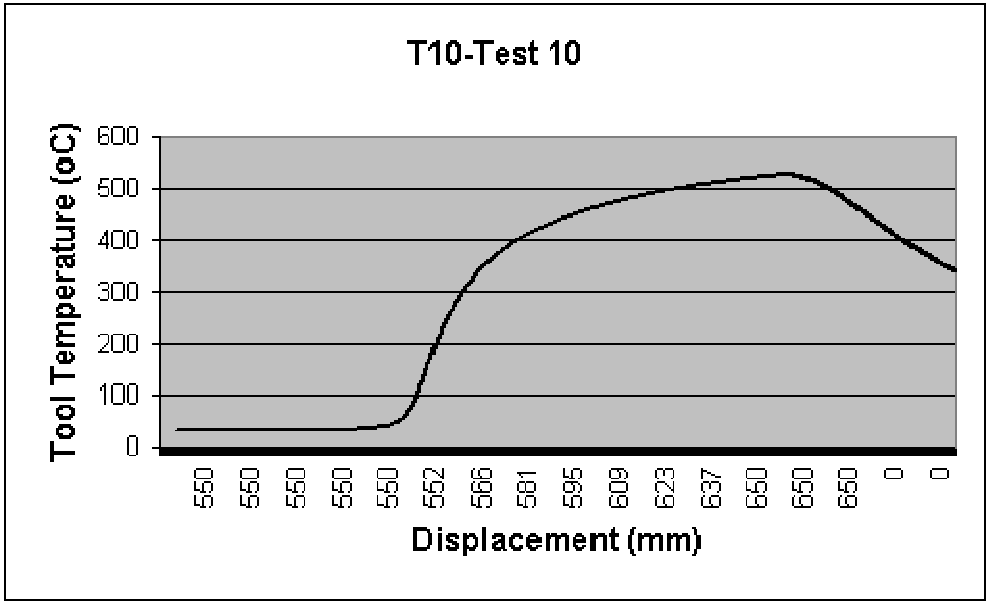

T9-Test 9
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
-30
420
420
420
420
420
420
420
424
450
476
503
520
520
520
520
0
Displacement (mm)

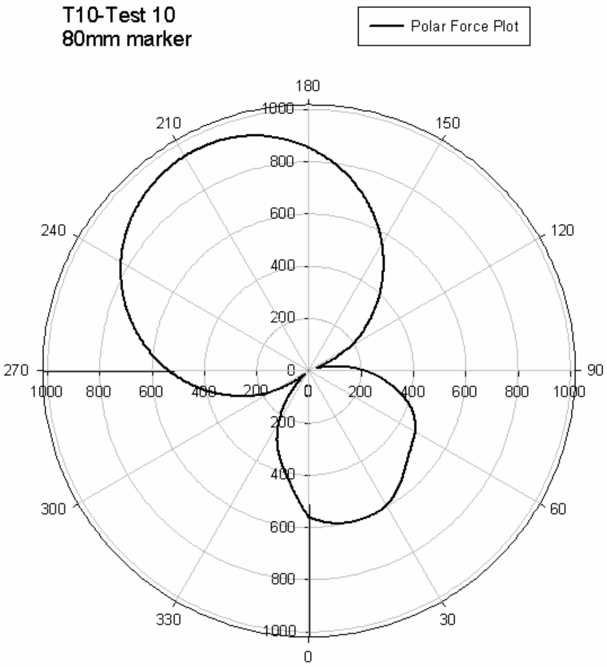

T9-Test 9
Tool Temperature (oC)
500
400
300
200
100
0
420
420
420
420
420
420
420
425
452
478
505
520
520
520
520
0
0
Displacement (mm)

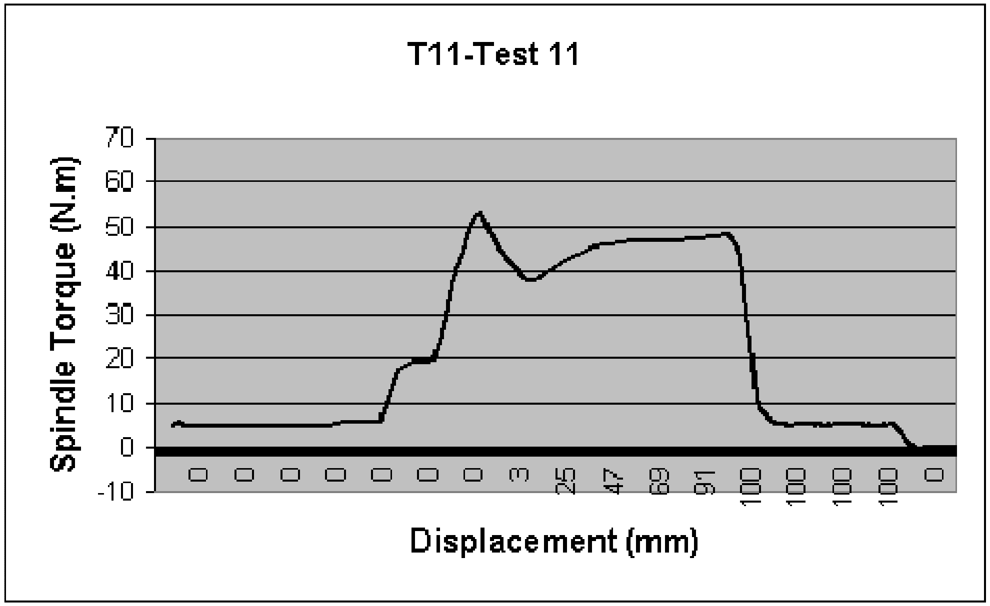

T9-Test 9
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
1200
1000
800
600
400
200
0

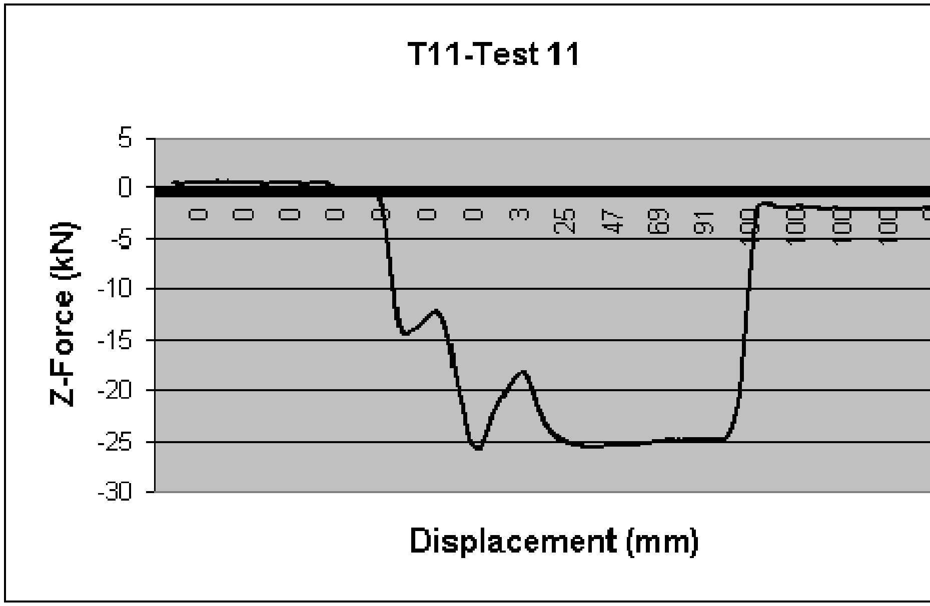

T10-Test 10
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
550
550
550
550
550
551
565
579
594
607
621
636
649
650
650
0
0
Displacement (mm)

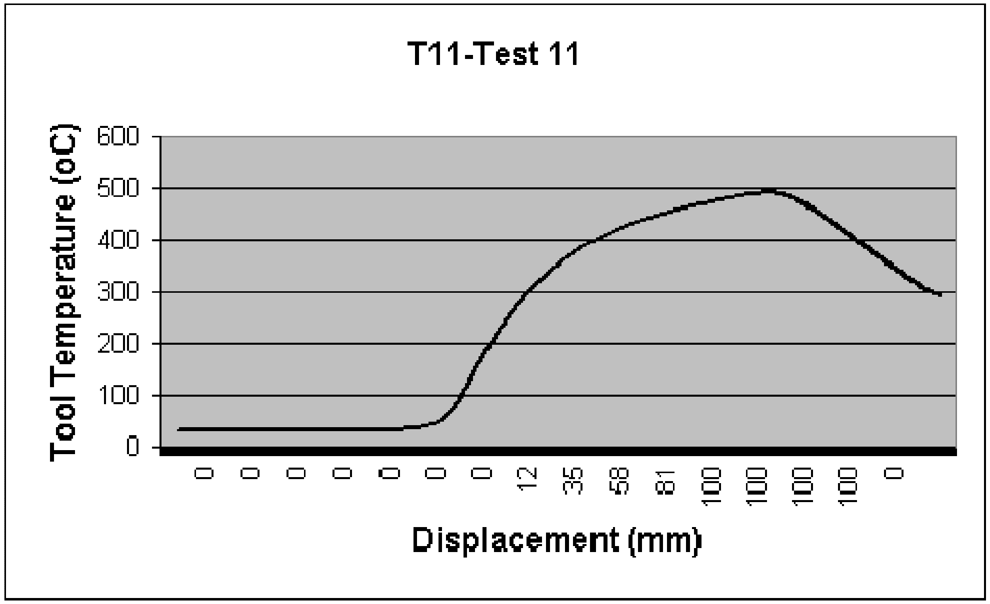

T10-Test 10
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
550
550
550
550
550
551
565
579
594
607
621
636
649
650
650
0
Displacement (mm)

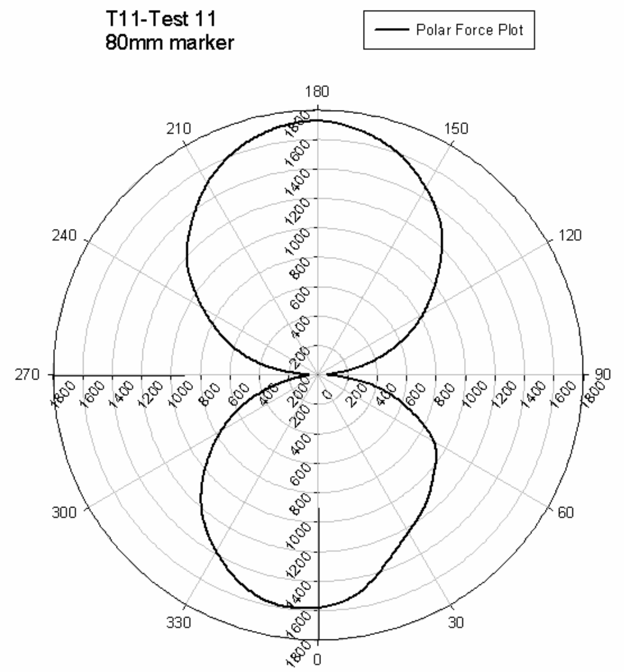

T10-Test 10
Tool Temperature (oC)
600
500
400
300
200
100
0
550
550
550
550
550
552
566
581
595
609
623
637
650
650
650
0
0
Displacement (mm)

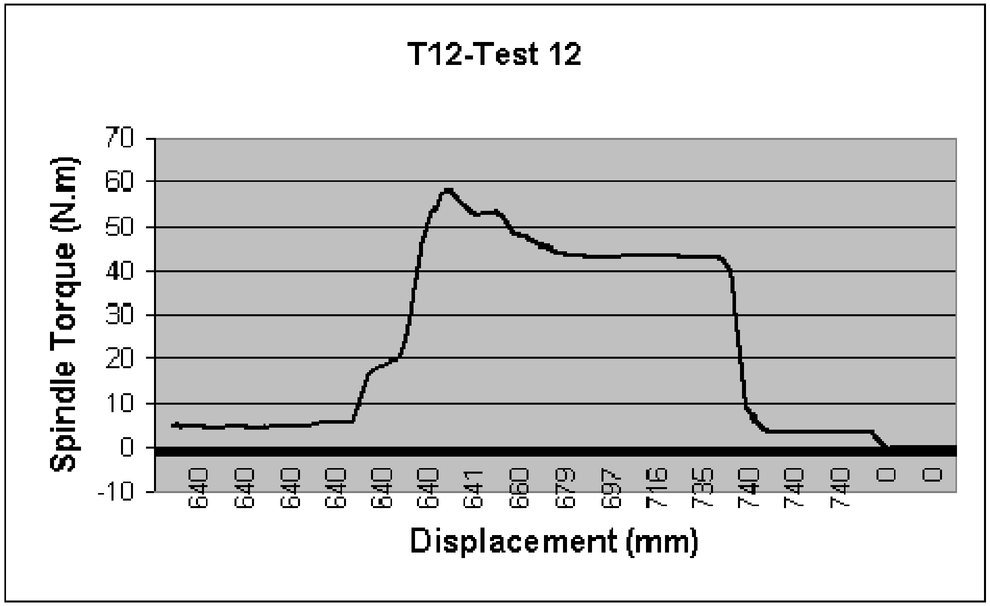

T10-Test 10
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
1000
800
600
400
200
0
200
400
600
800
1000

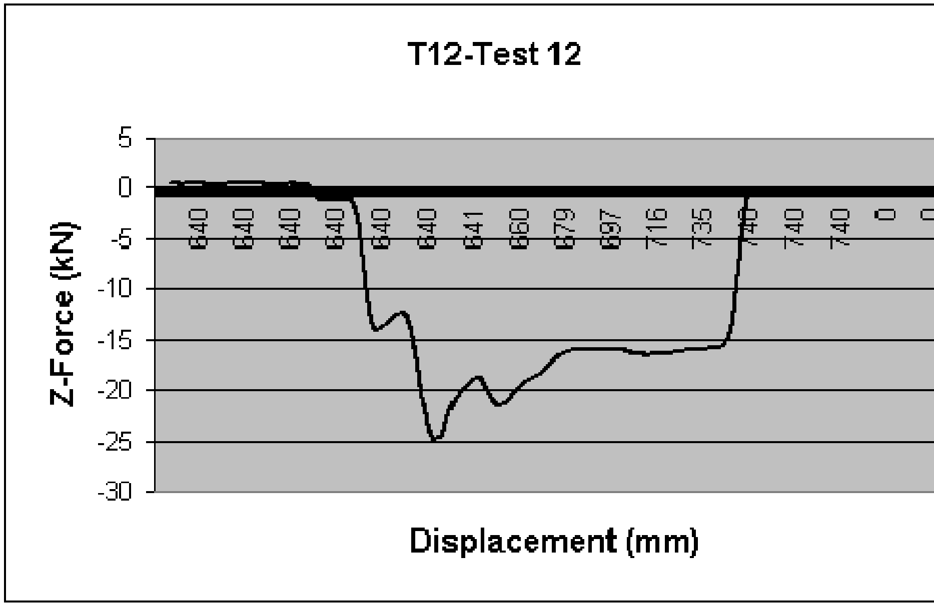

T11-Test 11
Spindle Torque (N.m)
70
60
50
40
30
20
10
0
-10
0 0 0 0 0 0 0 3 25 47 69 91 100 100 100 100 0
Displacement (mm)

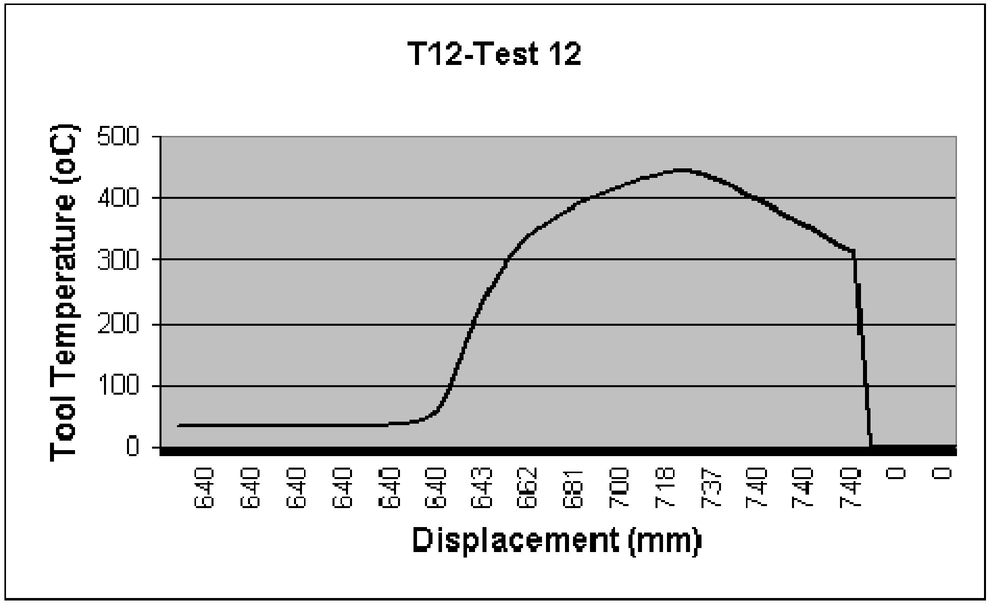

T11-Test 11
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
-30
0 0 0 0 0 0 0 3 25 47 69 91 100 100 100 100
Displacement (mm)

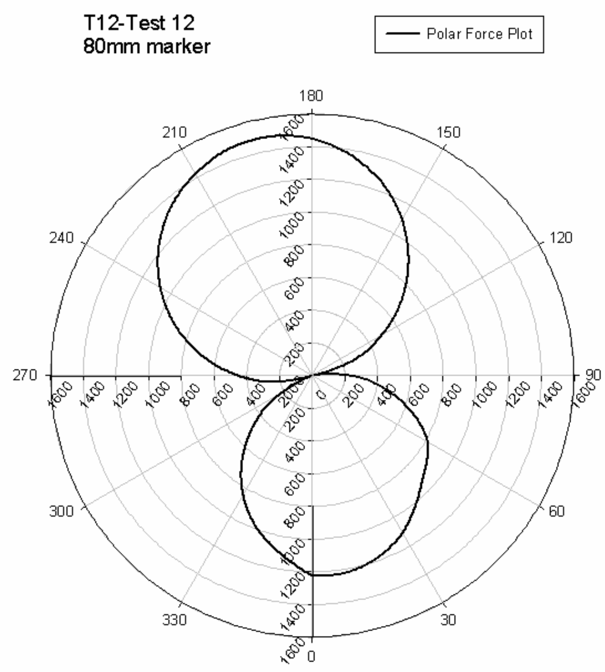

T11-Test 11
Tool Temperature (oC)
600
500
400
300
200
100
0
0 0 0 0 0 0 0 12 35 58 81 100 100 100 100 0
Displacement (mm)

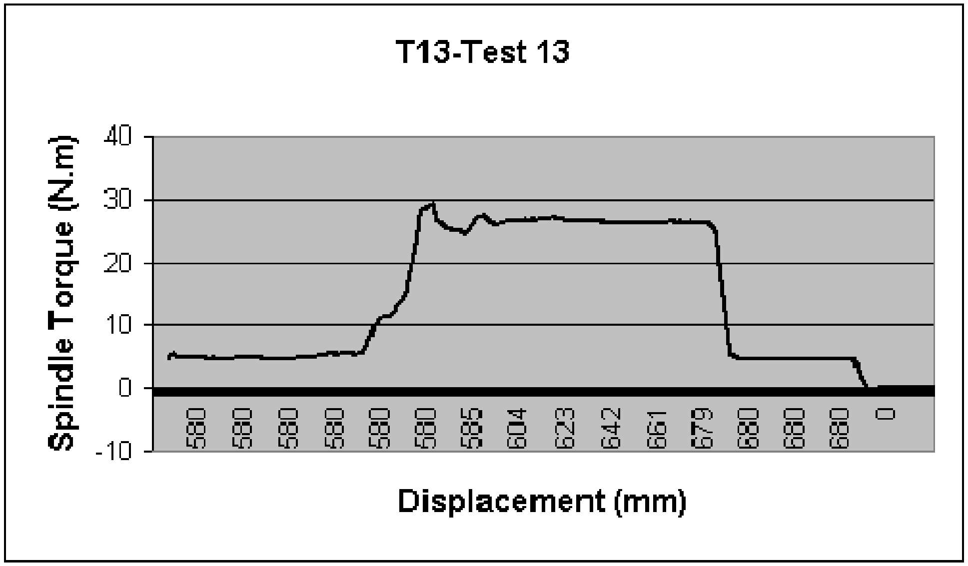

T11-Test 11
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
0
200
400
600
800
1000
1200
1400
1600
1800

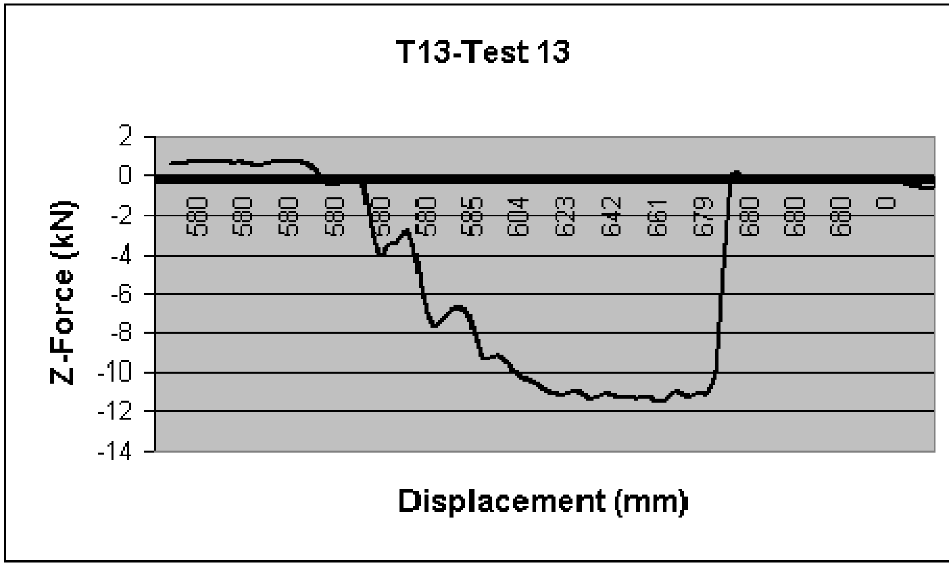

T12-Test 12
Spindle Torque (N.m)
70
60
50
40
30
20
10
0
-10
640
640
640
640
640
640
641
660
679
697
716
735
740
740
740
0
0
Displacement (mm)

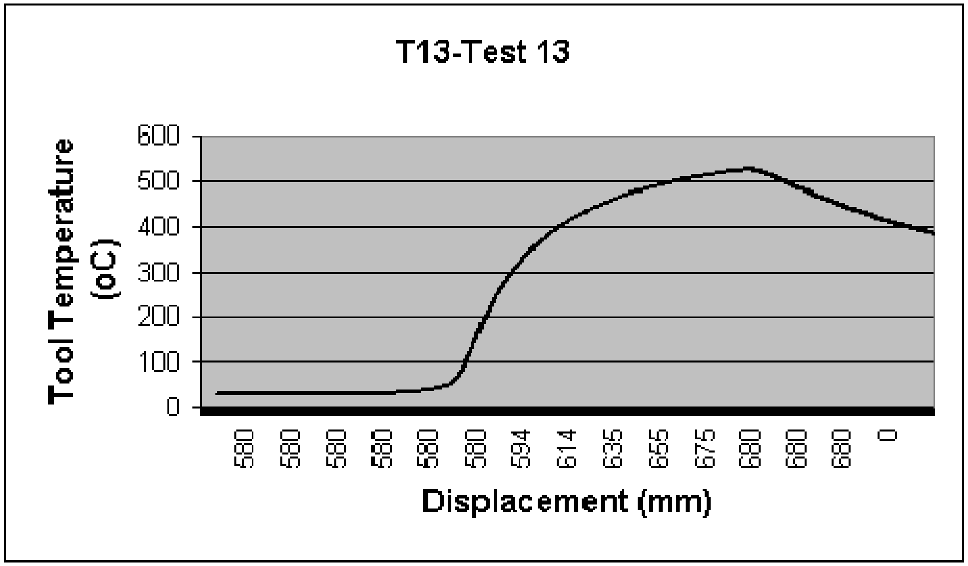

T12-Test 12
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
-30
640
640
640
640
640
640
641
660
679
697
716
735
740
740
740
0
Displacement (mm)

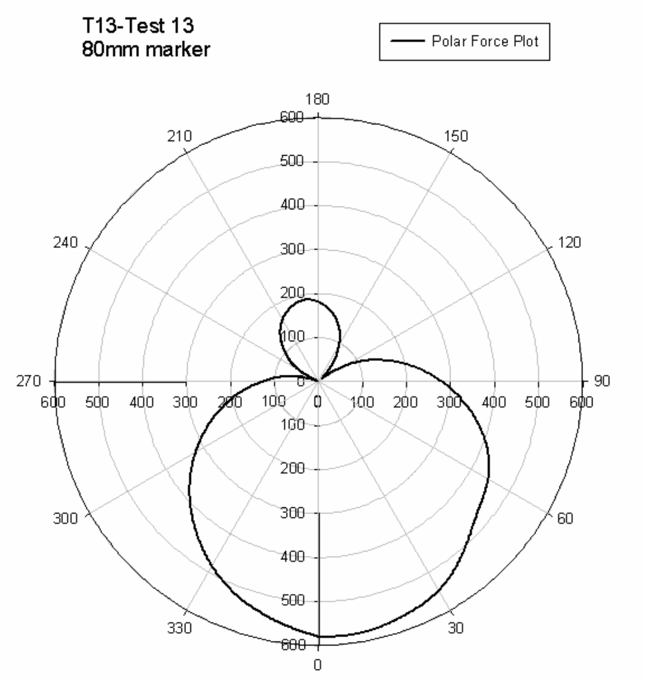

T12-Test 12
Tool Temperature (oC)
500
400
300
200
100
0
640
640
640
640
640
640
643
662
681
700
718
737
740
740
740
0
0
Displacement (mm)

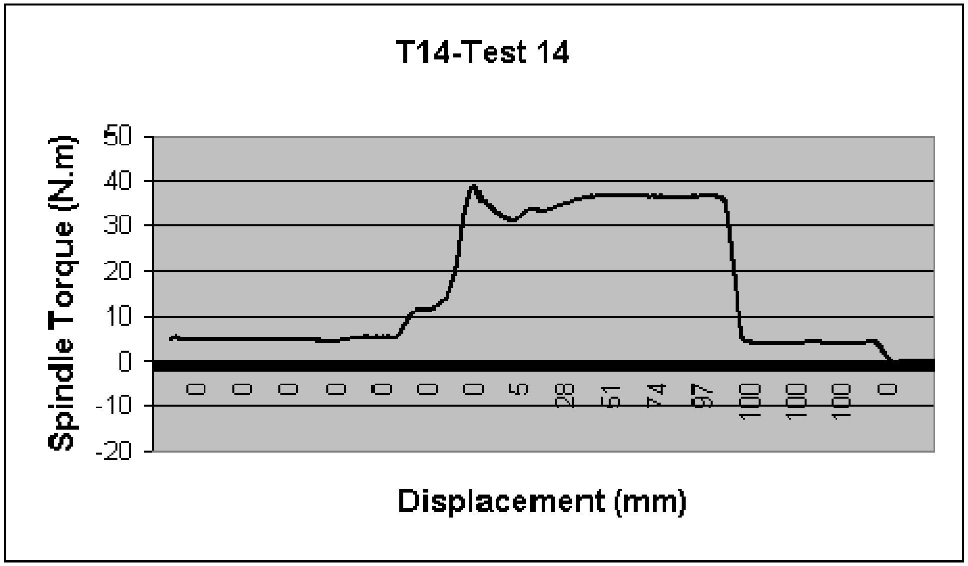

T12-Test 12
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
0
200
400
600
800
1000
1200
1400
1600

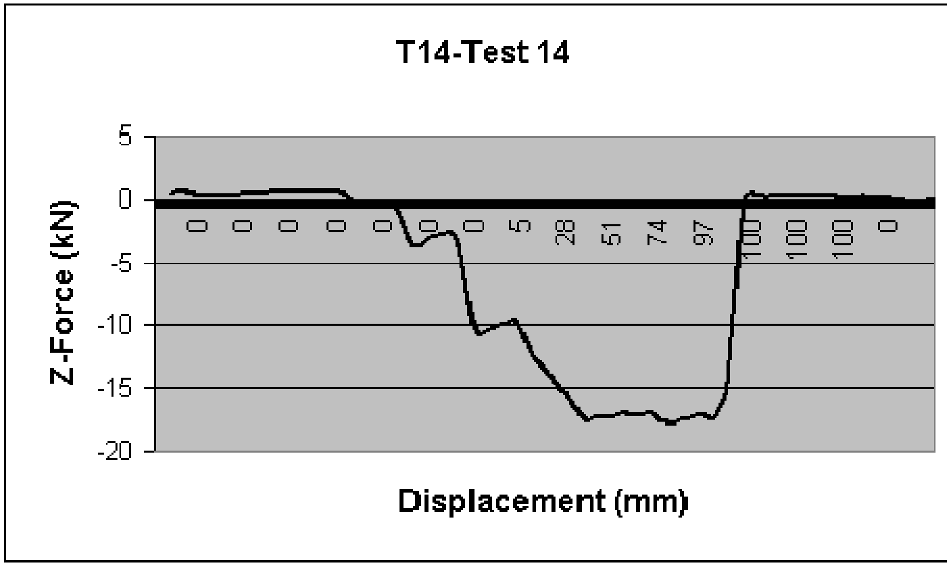

T13-Test 13
Spindle Torque (N.m)
40
30
20
10
0
-10
580
580
580
580
580
580
585
604
623
642
661
679
680
680
680
0
Displacement (mm)

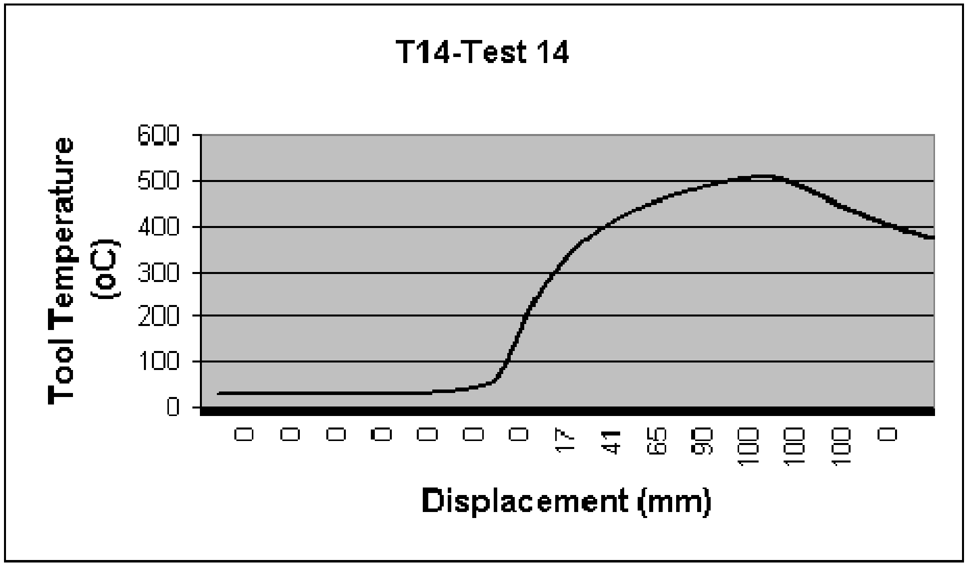

T13-Test 13
Z-Force (kN)
2
0
-2
-4
-6
-8
-10
-12
-14
580
580
580
580
580
580
585
604
623
642
661
679
680
680
680
0
Displacement (mm)

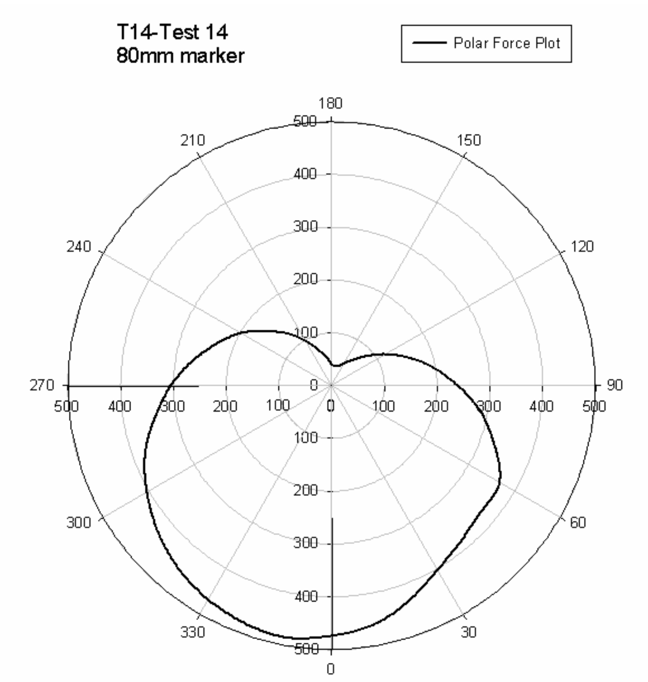

T13-Test 13
Tool Temperature (oC)
600
500
400
300
200
100
0
580
580
580
580
580
580
594
614
635
655
675
680
680
680
0
Displacement (mm)

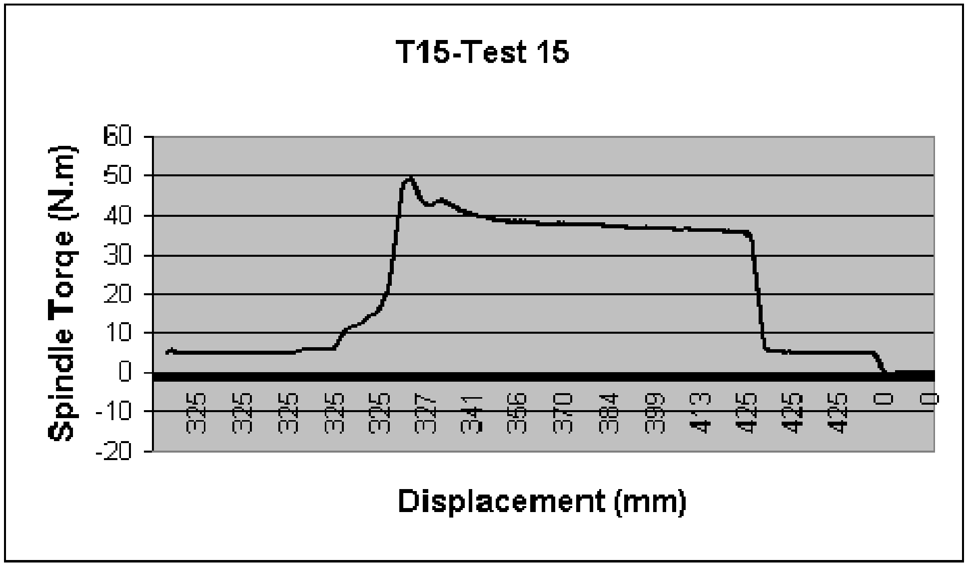

T13-Test 13
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
600
500
400
300
200
100
0
100
200
300
400
500
600

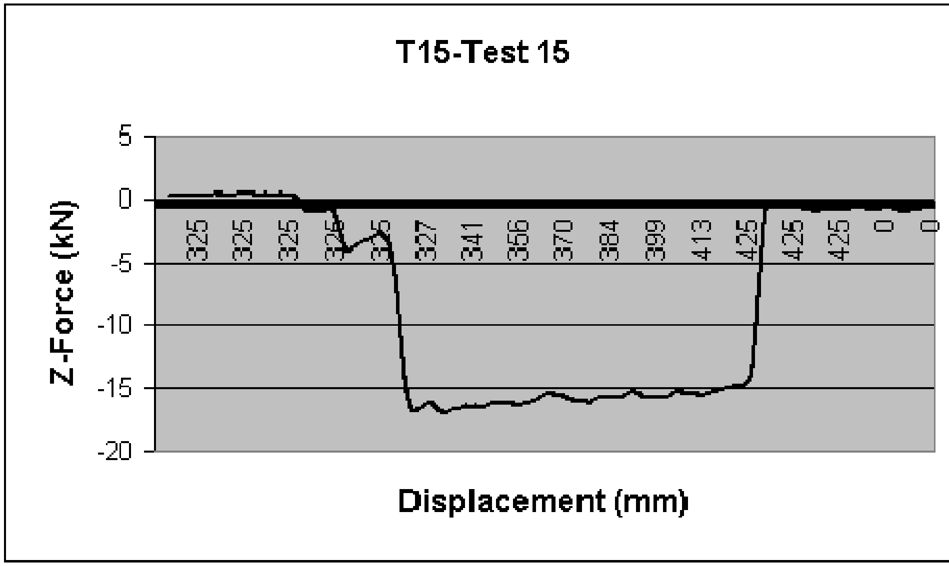

T14-Test 14

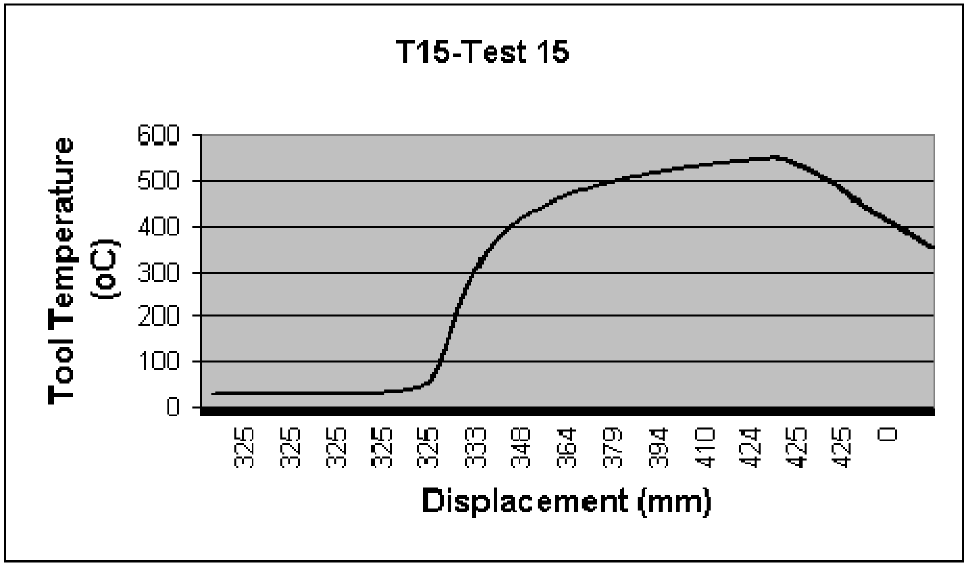

T14-Test 14

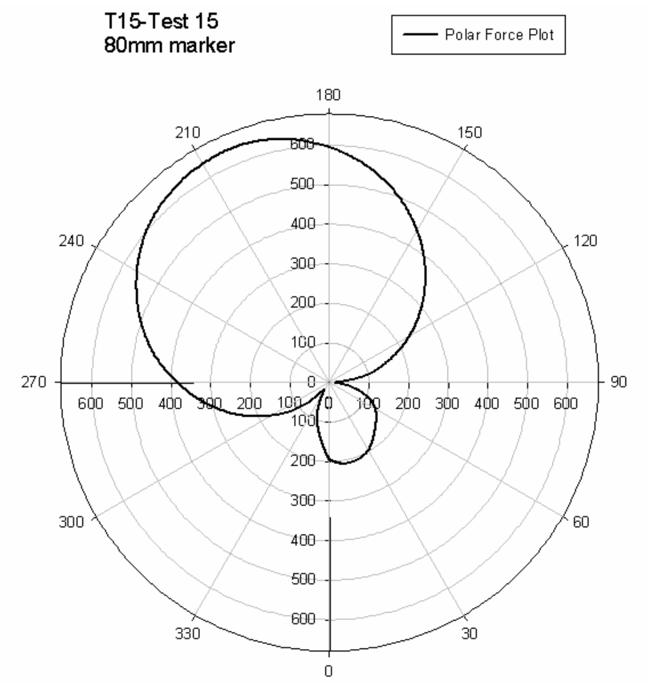

T14-Test 14

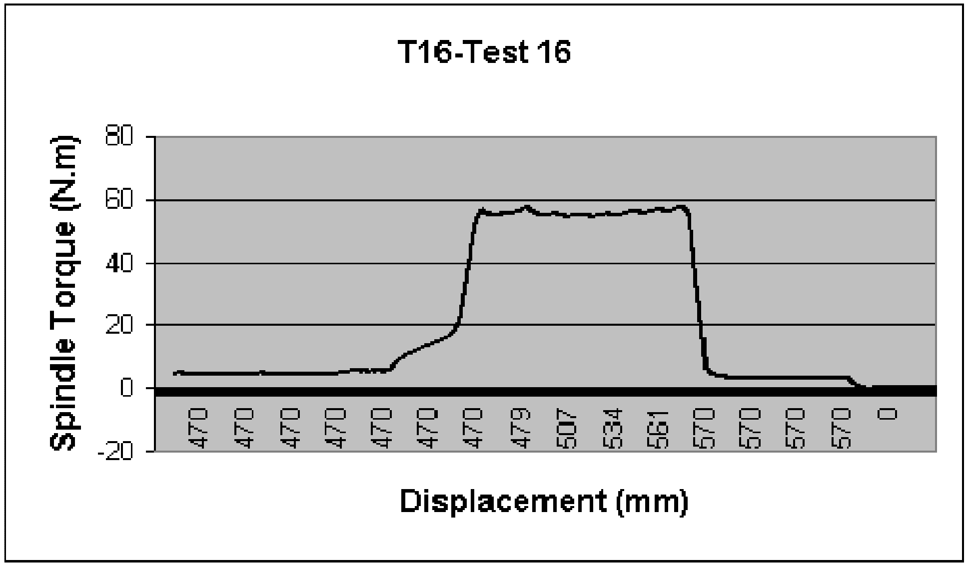

T14-Test 14
80mm marker

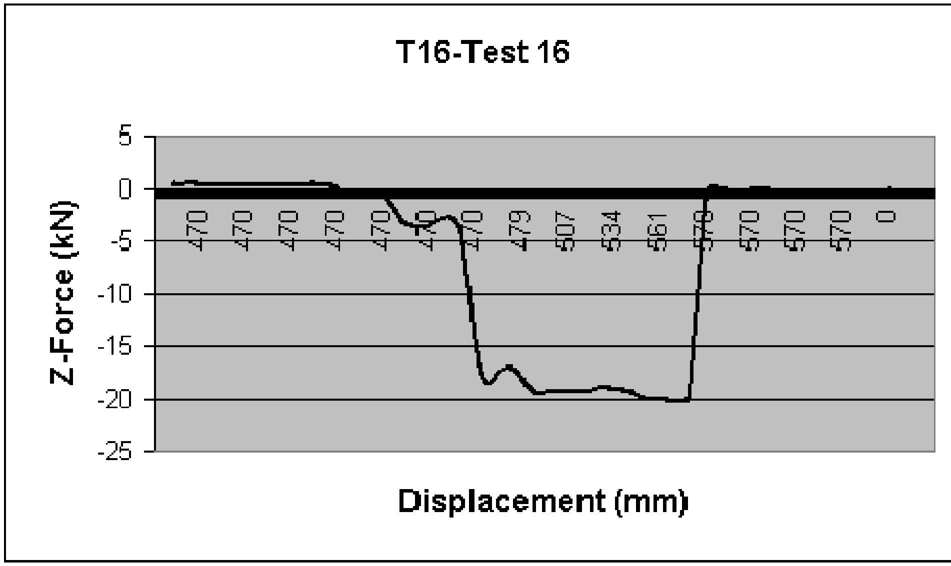

T15-Test 15
Spindle Torque (N.m)
60
50
40
30
20
10
0
-10
-20
325
325
325
325
325
327
341
356
370
384
399
413
425
425
425
0
0
Displacement (mm)

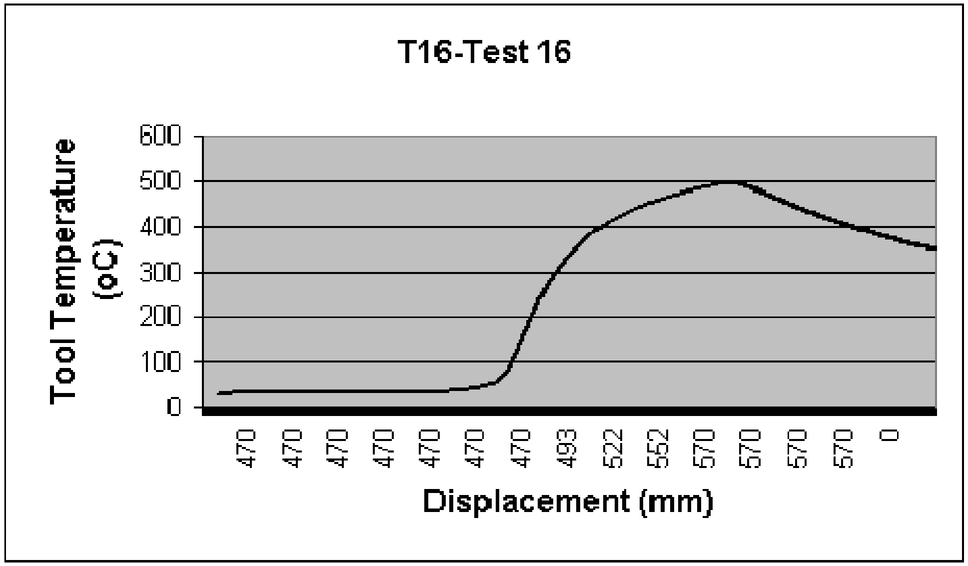

T15-Test 15
Z-Force (kN)
5
0
-5
-10
-15
-20
325
325
325
325
325
327
341
356
370
384
399
413
425
425
425
0
0
Displacement (mm)

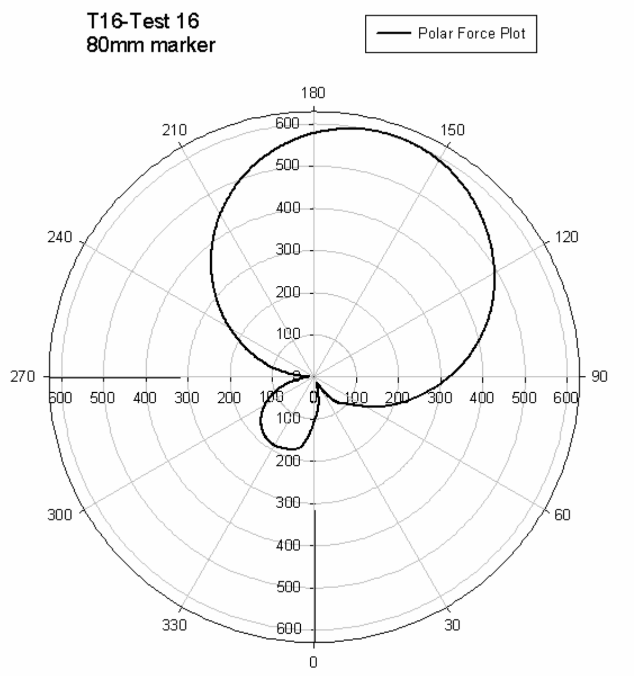

T15-Test 15
Tool Temperature (oC)
600
500
400
300
200
100
0
325
325
325
325
325
333
348
364
379
394
410
424
425
425
0
Displacement (mm)


T15-Test 15
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
600
500
400
300
200
100
0


T16-Test 16
Spindle Torque (N.m)
80
60
40
20
0
-20
470
470
470
470
470
470
470
479
507
534
561
570
570
570
570
0
Displacement (mm)

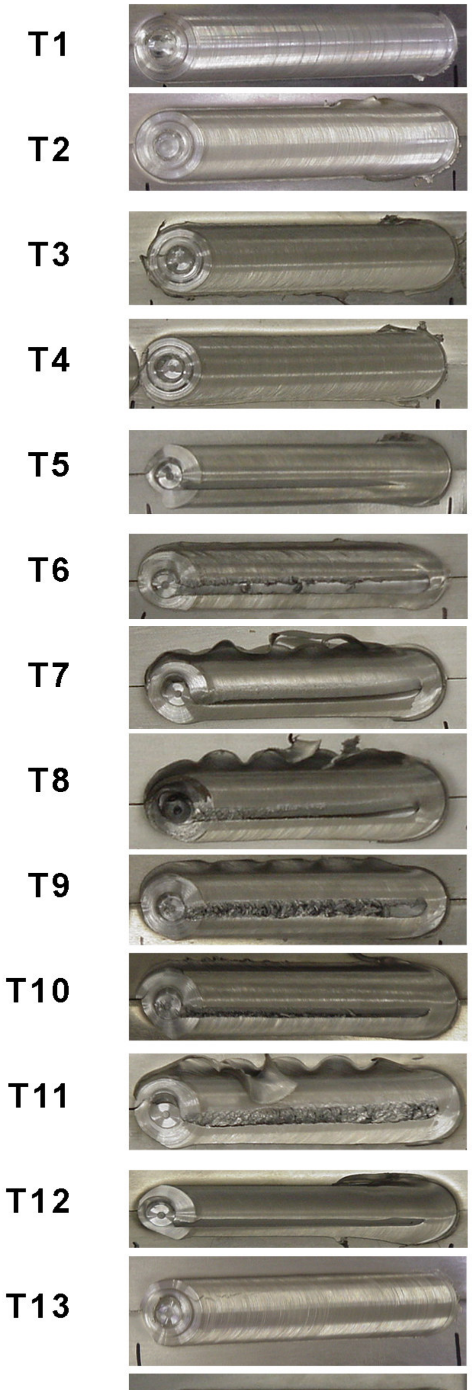

T16-Test 16
Z-Force (kN)
5
0
-5
-10
-15
-20
-25
470
470
470
470
470
470
470
479
507
534
561
570
570
570
570
0
Displacement (mm)


T16-Test 16
Tool Temperature (oC)
600
500
400
300
200
100
0
470
470
470
470
470
470
470
493
522
552
570
570
570
570
0
Displacement (mm)


T16-Test 16
80mm marker
Polar Force Plot
180
150
120
90
60
30
0
330
300
270
240
210
600
500
400
300
200
100
0
100
200
300
400
500
600

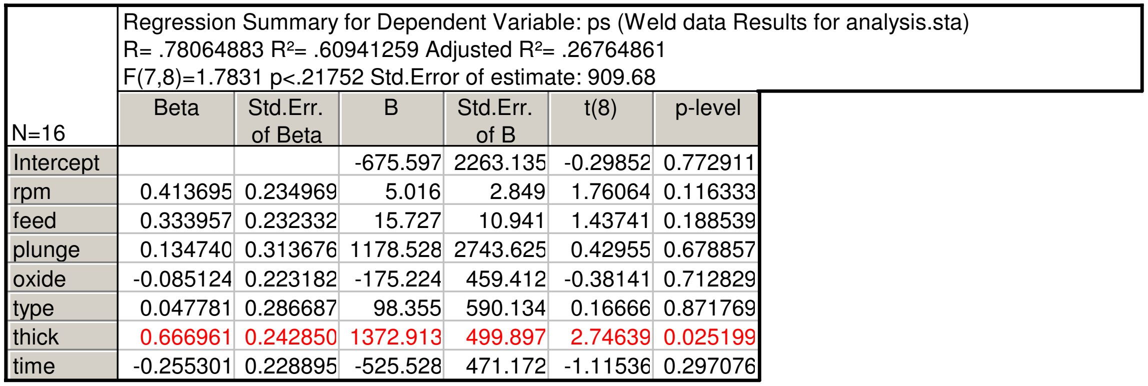
Appendix E

Surface Finishes of FSW Weld Tests



Appendix F

F.1: Regression Summary for Independent Variables



Regression Summary for Dependent Variable: ps (Weld data Results for analysis.sta)
R= .78064883 R²= .60941259 Adjusted R²= .26764861
F(7,8)=1.7831 p<.21752 Std.Error of estimate: 909.68

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			-675.597	2263.135	-0.29852	0.772911
rpm	0.413695	0.234969	5.016	2.849	1.76064	0.116333
feed	0.333957	0.232332	15.727	10.941	1.43741	0.188539
plunge	0.134740	0.313676	1178.528	2743.625	0.42955	0.678857
oxide	-0.085124	0.223182	-175.224	459.412	-0.38141	0.712829
type	0.047781	0.286687	98.355	590.134	0.16666	0.871769
thick	0.666961	0.242850	1372.913	499.897	2.74639	0.025199
time	-0.255301	0.228895	-525.528	471.172	-1.11536	0.297076

Regression Summary for Dependent Variable: ts (Weld data Results for analysis.sta)
R= .93237423 R²= .86932170 Adjusted R²= .75497818
F(7,8)=7.6027 p<.00520 Std.Error of estimate: 3.8479

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			33.11654	9.57293	3.45939	0.008575
rpm	-0.455522	0.135910	-0.04039	0.01205	-3.35164	0.010056
feed	0.475613	0.134386	0.16380	0.04628	3.53917	0.007629
plunge	-0.016836	0.181436	-1.07690	11.60538	-0.09279	0.928350
oxide	0.020301	0.129093	0.30560	1.94329	0.15726	0.878938
type	-0.032017	0.165825	-0.48196	2.49623	-0.19307	0.851712
thick	0.559358	0.140469	8.42024	2.11453	3.98208	0.004050
time	-0.088104	0.132397	-1.32626	1.99303	-0.66545	0.524481

Regression Summary for Dependent Variable: cs (Weld data Results for analysis.sta)
R= .73986555 R²= .54740103 Adjusted R²= .15137693
F(7,8)=1.3822 p<.32818 Std.Error of estimate: 2.2587

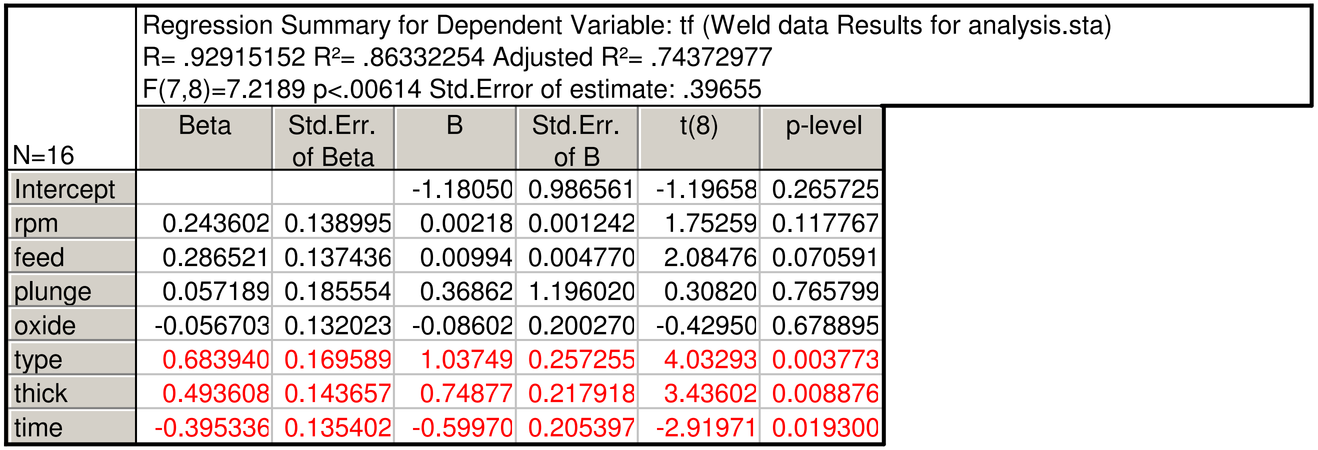
N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			4.03186	5.619328	0.717498	0.493482
rpm	0.113189	0.252934	0.00317	0.007074	0.447503	0.666381
feed	0.337462	0.250096	0.03666	0.027167	1.349329	0.214173
plunge	0.075732	0.337660	1.52791	6.812377	0.224284	0.828158
oxide	-0.087674	0.240247	-0.41629	1.140713	-0.364934	0.724619
type	0.016678	0.308607	0.07919	1.465294	0.054042	0.958227
thick	0.652906	0.261418	3.10006	1.241235	2.497558	0.037083
time	-0.217069	0.246397	-1.03066	1.169914	-0.880975	0.404021

Regression Summary for Dependent Variable: pf (Weld data Results for analysis.sta)
R= .99947192 R^2= .99894413 Adjusted R^2= .99802024
F(7,8)=1081.2 p<.00000 Std.Error of estimate: 5.3673

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			30.44096	13.35294	2.27972	0.052094
rpm	-0.007876	0.012217	-0.01084	0.01681	-0.64469	0.537170
feed	0.999503	0.012080	5.34155	0.06456	82.74221	0.000000
plunge	-0.004675	0.016309	-4.64065	16.18793	-0.28667	0.781647
oxide	0.007666	0.011604	1.79081	2.71062	0.66066	0.527391
type	0.003186	0.014906	0.74425	3.48191	0.21375	0.836091
thick	-0.019196	0.012627	-4.48396	2.94949	-1.52025	0.166937
time	0.021755	0.011901	5.08189	2.78001	1.82801	0.104956

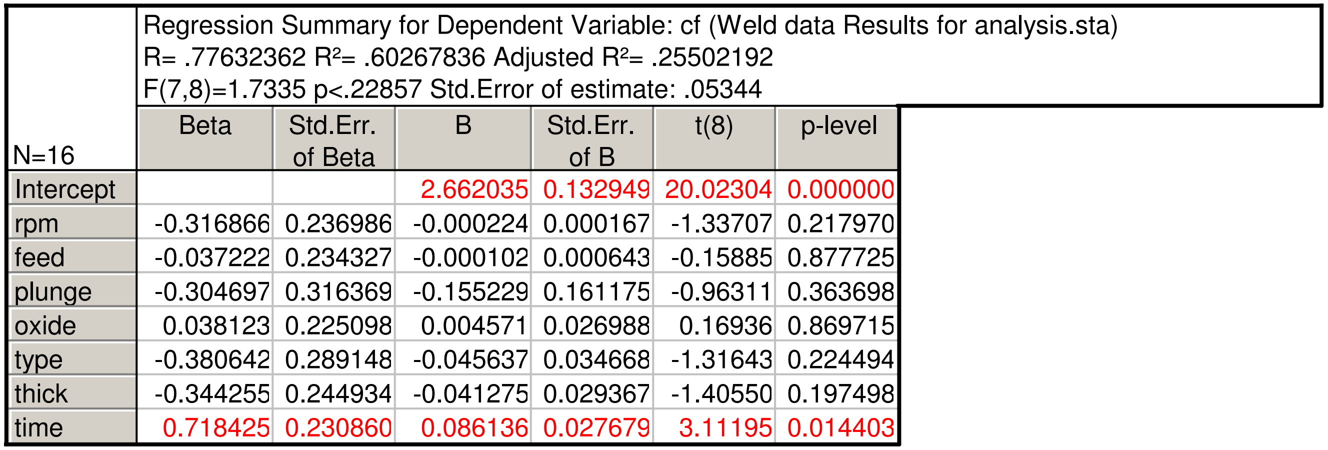
Regression Summary for Dependent Variable: tf (Weld data Results for analysis.sta)
R= .92915152 R^2= .86332254 Adjusted R^2= .74372977
F(7,8)=7.2189 p<.00614 Std.Error of estimate: .39655

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			-1.18050	0.986561	-1.19658	0.265725
rpm	0.243602	0.138995	0.00218	0.001242	1.75259	0.117767
feed	0.286521	0.137436	0.00994	0.004770	2.08476	0.070591
plunge	0.057189	0.185554	0.36862	1.196020	0.30820	0.765799
oxide	-0.056703	0.132023	-0.08602	0.200270	-0.42950	0.678895
type	0.683940	0.169589	1.03749	0.257255	4.03293	0.003773
thick	0.493608	0.143657	0.74877	0.217918	3.43602	0.008876
time	-0.395336	0.135402	-0.59970	0.205397	-2.91971	0.019300



Regression Summary for Dependent Variable: tf (Weld data Results for analysis.sta)
R= .92915152 R²= .86332254 Adjusted R²= .74372977
F(7,8)=7.2189 p<.00614 Std.Error of estimate: .39655

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			-1.18050	0.986561	-1.19658	0.265725
rpm	0.243602	0.138995	0.00218	0.001242	1.75259	0.117767
feed	0.286521	0.137436	0.00994	0.004770	2.08476	0.070591
plunge	0.057189	0.185554	0.36862	1.196020	0.30820	0.765799
oxide	-0.056703	0.132023	-0.08602	0.200270	-0.42950	0.678895
type	0.683940	0.169589	1.03749	0.257255	4.03293	0.003773
thick	0.493608	0.143657	0.74877	0.217918	3.43602	0.008876
time	-0.395336	0.135402	-0.59970	0.205397	-2.91971	0.019300



Regression Summary for Dependent Variable: cf (Weld data Results for analysis.sta)
R= .77632362 R²= .60267836 Adjusted R²= .25502192
F(7,8)=1.7335 p<.22857 Std.Error of estimate: .05344

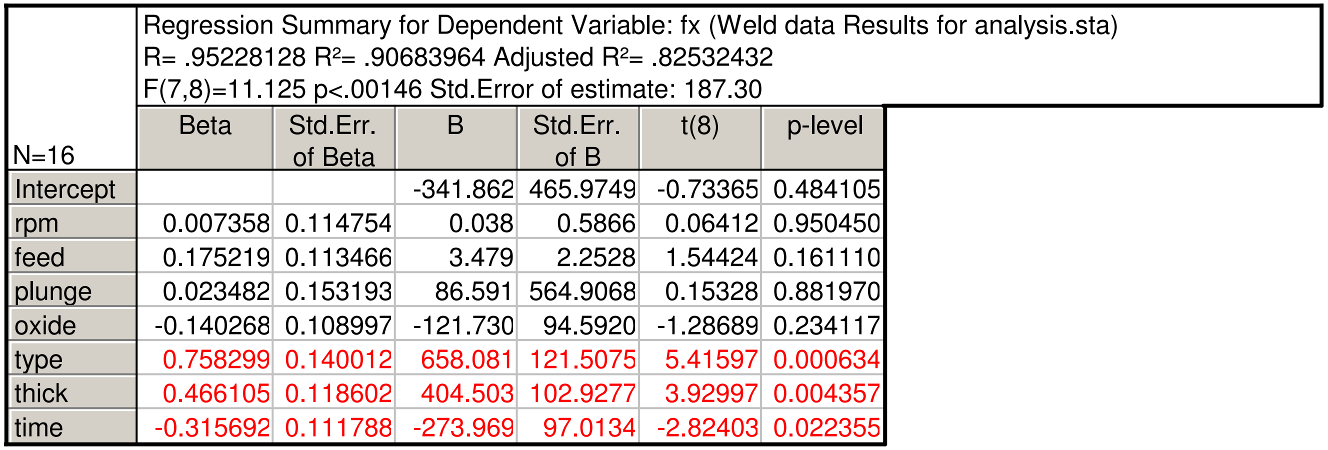
N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			2.662035	0.132949	20.02304	0.000000
rpm	-0.316866	0.236986	-0.000224	0.000167	-1.33707	0.217970
feed	-0.037222	0.234327	-0.000102	0.000643	-0.15885	0.877725
plunge	-0.304697	0.316369	-0.155229	0.161175	-0.96311	0.363698
oxide	0.038123	0.225098	0.004571	0.026988	0.16936	0.869715
type	-0.380642	0.289148	-0.045637	0.034668	-1.31643	0.224494
thick	-0.344255	0.244934	-0.041275	0.029367	-1.40550	0.197498
time	0.718425	0.230860	0.086136	0.027679	3.11195	0.014403

Regression Summary for Dependent Variable: cf (Weld data Results for analysis.sta)
R= .77632362 R²= .60267836 Adjusted R²= .25502192
F(7,8)=1.7335 p<.22857 Std.Error of estimate: .05344

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			2.662035	0.132949	20.02304	0.000000
rpm	-0.316866	0.236986	-0.000224	0.000167	-1.33707	0.217970
feed	-0.037222	0.234327	-0.000102	0.000643	-0.15885	0.877725
plunge	-0.304697	0.316369	-0.155229	0.161175	-0.96311	0.363698
oxide	0.038123	0.225098	0.004571	0.026988	0.16936	0.869715
type	-0.380642	0.289148	-0.045637	0.034668	-1.31643	0.224494
thick	-0.344255	0.244934	-0.041275	0.029367	-1.40550	0.197498
time	0.718425	0.230860	0.086136	0.027679	3.11195	0.014403

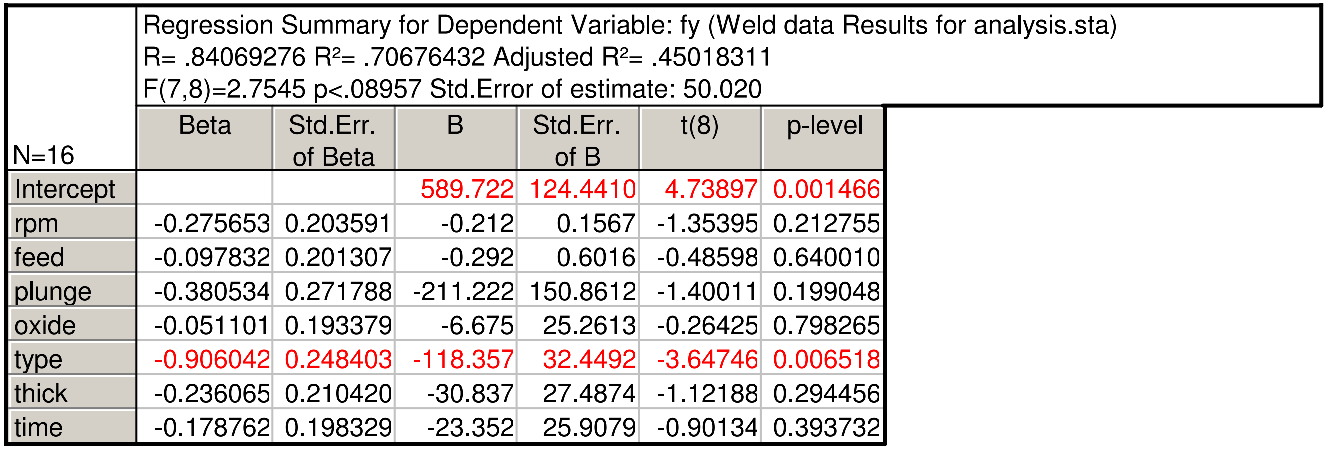
Regression Summary for Dependent Variable: t (Weld data Results for analysis.sta)
R= .79478862 R²= .63168895 Adjusted R²= .30941679
F(7,8)=1.9601 p<.18289 Std.Error of estimate: 27.752

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			406.7750	69.04195	5.89171	0.000365
rpm	0.360033	0.228170	0.1371	0.08692	1.57792	0.153237
feed	-0.449272	0.225610	-0.6647	0.33379	-1.99137	0.081600
plunge	0.391046	0.304600	107.4547	83.70036	1.28380	0.235144
oxide	0.158526	0.216724	10.2517	14.01538	0.73146	0.485367
type	-0.246808	0.278392	-15.9608	18.00336	-0.88655	0.401187
thick	0.431032	0.235823	27.8745	15.25045	1.82778	0.104994
time	-0.080133	0.222272	-5.1822	14.37416	-0.36052	0.727793



Regression Summary for Dependent Variable: fx (Weld data Results for analysis.sta)
R= .95228128 R²= .90683964 Adjusted R²= .82532432
F(7,8)=11.125 p<.00146 Std.Error of estimate: 187.30

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			-341.862	465.9749	-0.73365	0.484105
rpm	0.007358	0.114754	0.038	0.5866	0.06412	0.950450
feed	0.175219	0.113466	3.479	2.2528	1.54424	0.161110
plunge	0.023482	0.153193	86.591	564.9068	0.15328	0.881970
oxide	-0.140268	0.108997	-121.730	94.5920	-1.28689	0.234117
type	0.758299	0.140012	658.081	121.5075	5.41597	0.000634
thick	0.466105	0.118602	404.503	102.9277	3.92997	0.004357
time	-0.315692	0.111788	-273.969	97.0134	-2.82403	0.022355



Regression Summary for Dependent Variable: fy (Weld data Results for analysis.sta)
R= .84069276 R²= .70676432 Adjusted R²= .45018311
F(7,8)=2.7545 p<.08957 Std.Error of estimate: 50.020

N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			589.722	124.4410	4.73897	0.001466
rpm	-0.275653	0.203591	-0.212	0.1567	-1.35395	0.212755
feed	-0.097832	0.201307	-0.292	0.6016	-0.48598	0.640010
plunge	-0.380534	0.271788	-211.222	150.8612	-1.40011	0.199048
oxide	-0.051101	0.193379	-6.675	25.2613	-0.26425	0.798265
type	-0.906042	0.248403	-118.357	32.4492	-3.64746	0.006518
thick	-0.236065	0.210420	-30.837	27.4874	-1.12188	0.294456
time	-0.178762	0.198329	-23.352	25.9079	-0.90134	0.393732

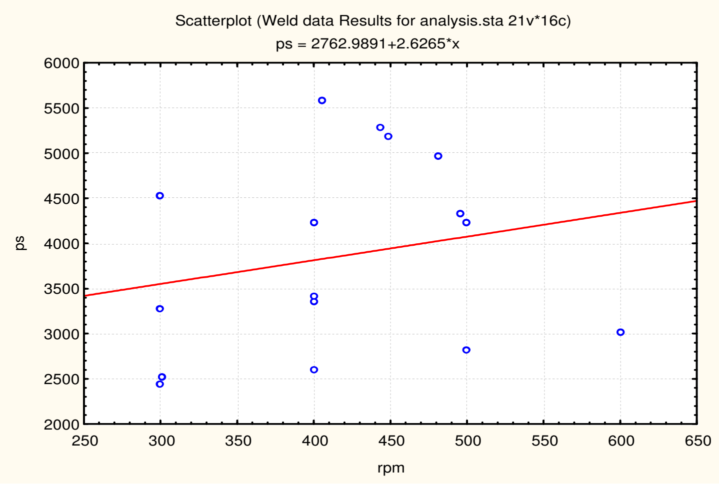
Regression Summary for Dependent Variable: r (Weld data Results for analysis.sta)
R= .84546614 R^2= .71481300 Adjusted R^2= .46527437
F(7,8)=2.8645 p<.08179 Std.Error of estimate: 4.2350

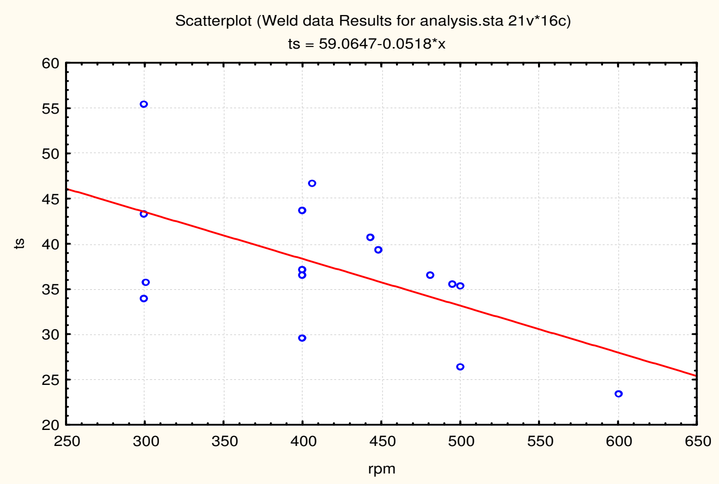
N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			-16.8060	10.53612	-1.59508	0.149361
rpm	0.206791	0.200778	0.0137	0.01326	1.02995	0.333164
feed	0.086083	0.198525	0.0221	0.05094	0.43361	0.676024
plunge	0.198191	0.268032	9.4448	12.77306	0.73943	0.480779
oxide	-0.090725	0.190706	-1.0175	2.13881	-0.47573	0.646984
type	0.833485	0.244971	9.3477	2.74740	3.40239	0.009327
thick	0.389339	0.207512	4.3665	2.32729	1.87623	0.097470
time	-0.254650	0.195588	-2.8560	2.19356	-1.30197	0.229161

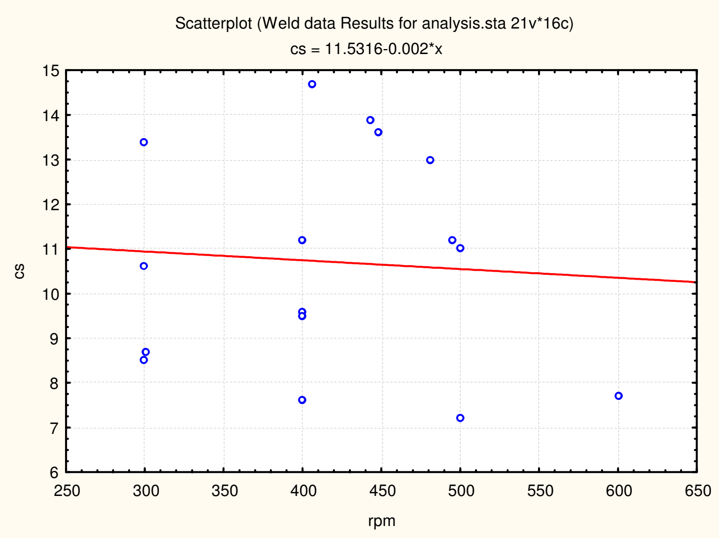
Regression Summary for Dependent Variable: fz (Weld data Results for analysis.sta)
R= .88689315 R^2= .78657946 Adjusted R^2= .59983649
F(7,8)=4.2121 p<.03058 Std.Error of estimate: 3.1524

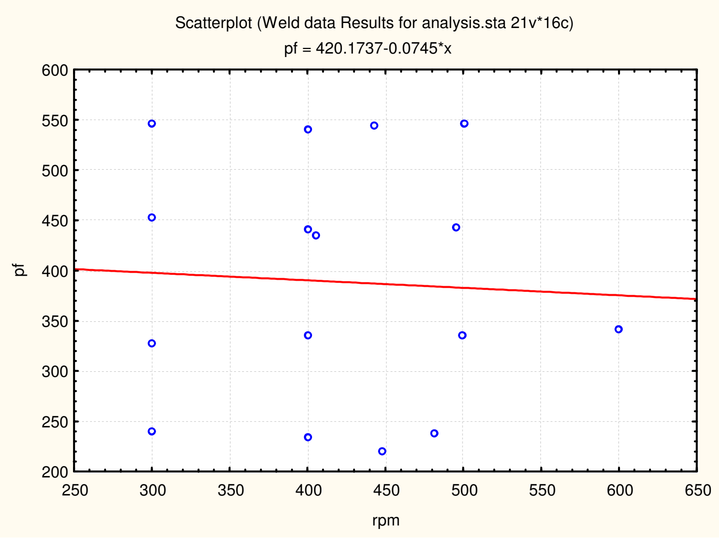
N=16	Beta	Std.Err. of Beta	B	Std.Err. of B	t(8)	p-level
Intercept			7.34152	7.842569	0.93611	0.376609
rpm	-0.214346	0.173688	-0.01218	0.009873	-1.23409	0.252192
feed	-0.571013	0.171739	-0.12607	0.037916	-3.32489	0.010464
plunge	-0.170515	0.231868	-6.99188	9.507637	-0.73540	0.483098
oxide	0.090810	0.164975	0.87633	1.592026	0.55045	0.597043
type	-0.473935	0.211918	-4.57352	2.045026	-2.23641	0.055737
thick	-0.545341	0.179513	-5.26259	1.732320	-3.03789	0.016115
time	0.240151	0.169198	2.31748	1.632780	1.41934	0.193566

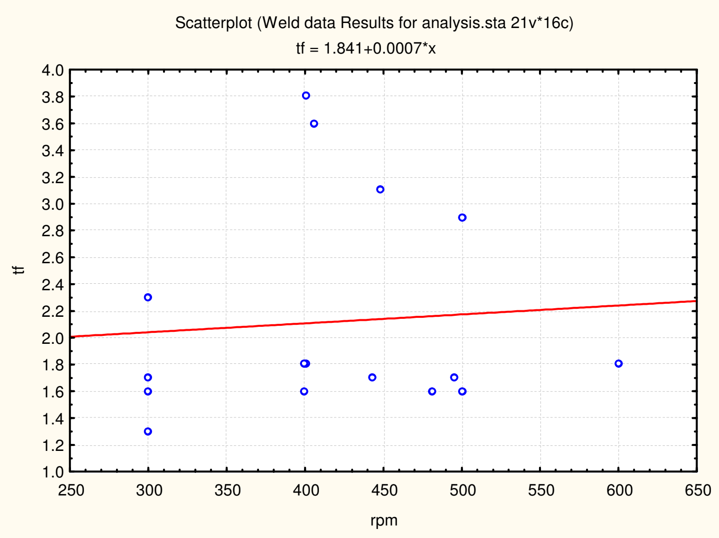
F.2: Scatter Plots for Independent Variables

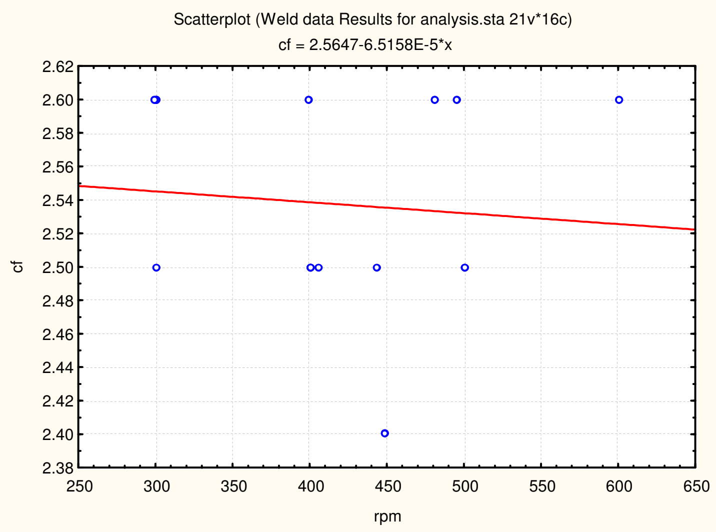


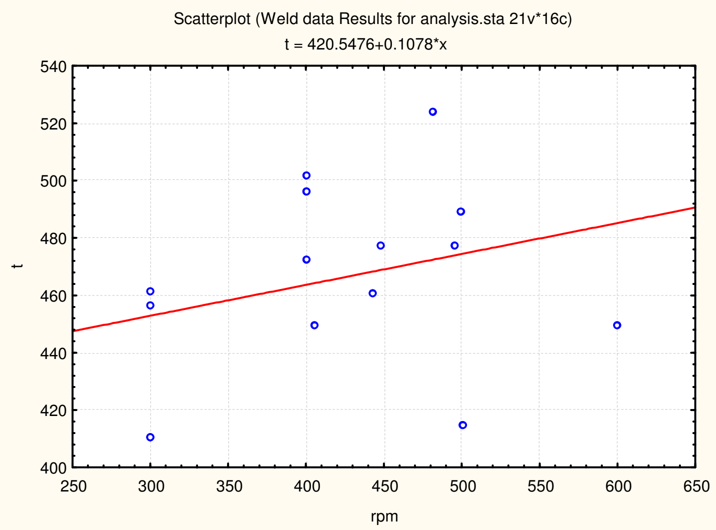

Scatterplot (Weld data Results for analysis.sta 21v*16c)
ts = 59.0647-0.0518*x
ts
rpm
20
25
30
35
40
45
50
55
60
250
300
350
400
450
500
550
600
650

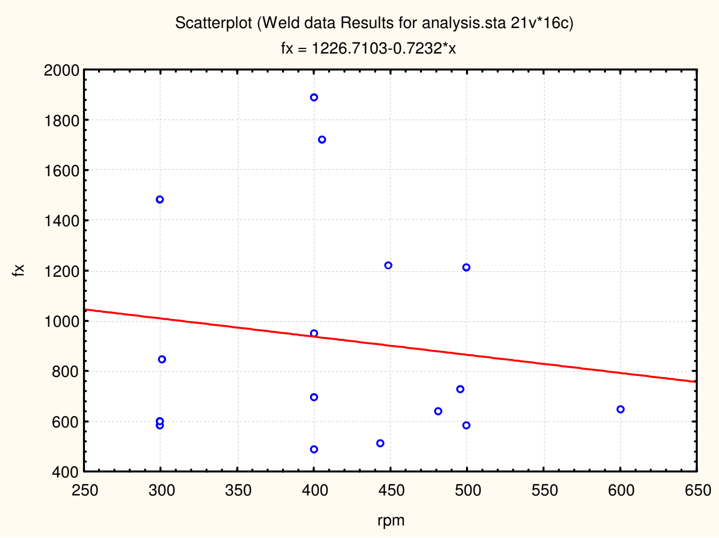

Scatterplot (Weld data Results for analysis.sta 21v*16c)
cs = 11.5316-0.002*x
cs
rpm
6
7
8
9
10
11
12
13
14
15
250
300
350
400
450
500
550
600
650

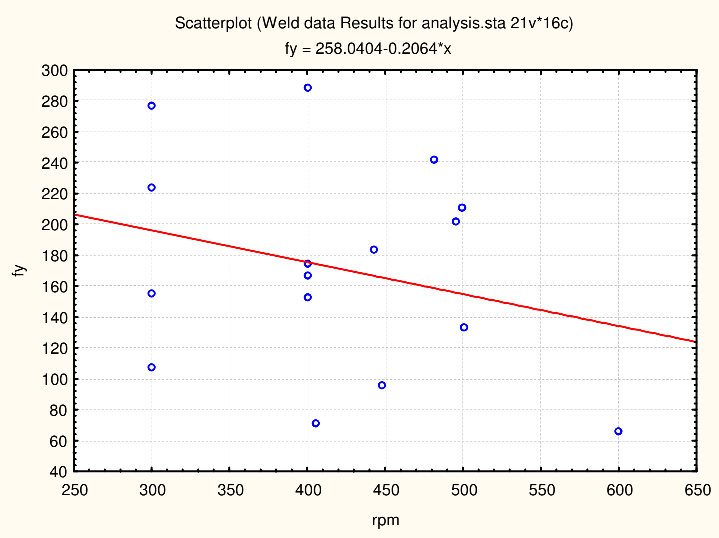

Scatterplot (Weld data Results for analysis.sta 21v*16c)
pf = 420.1737-0.0745*x
pf
rpm
600
550
500
450
400
350
300
250
200
250
300
350
400
450
500
550
600
650

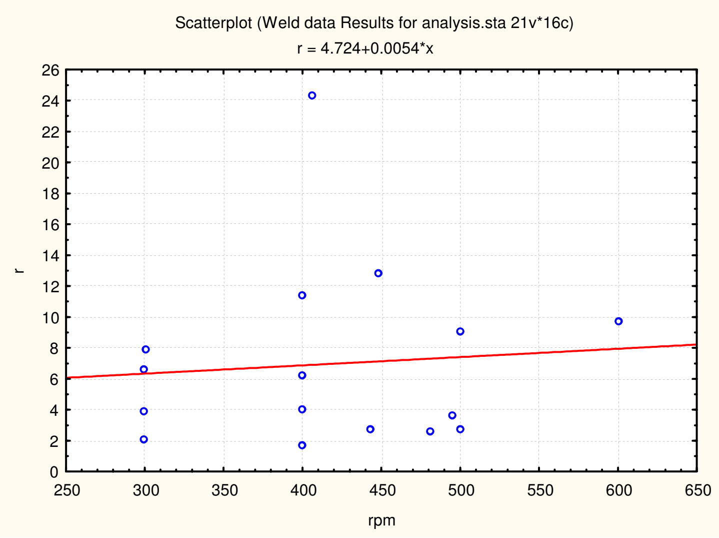

Scatterplot (Weld data Results for analysis.sta 21v*16c)
tf = 1.841+0.0007*x
tf
rpm
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
250
300
350
400
450
500
550
600
650

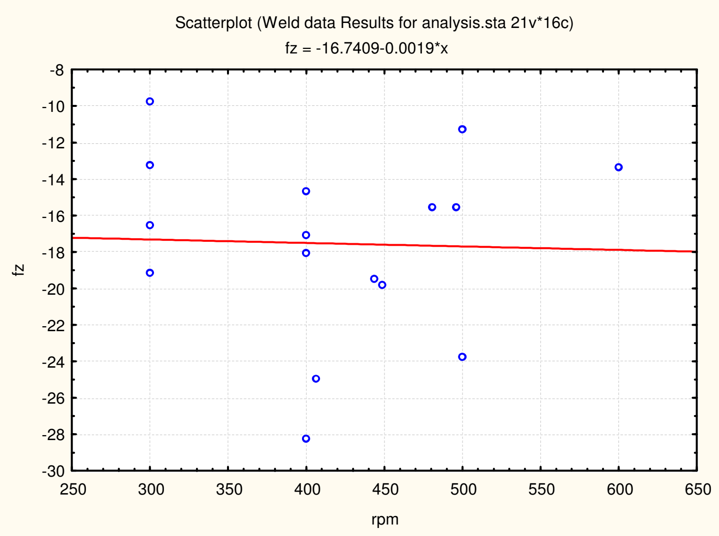

Scatterplot (Weld data Results for analysis.sta 21v*16c)
cf = 2.5647-6.5158E-5*x
cf
2.62
2.60
2.58
2.56
2.54
2.52
2.50
2.48
2.46
2.44
2.42
2.40
2.38
250
300
350
400
450
500
550
600
650
rpm

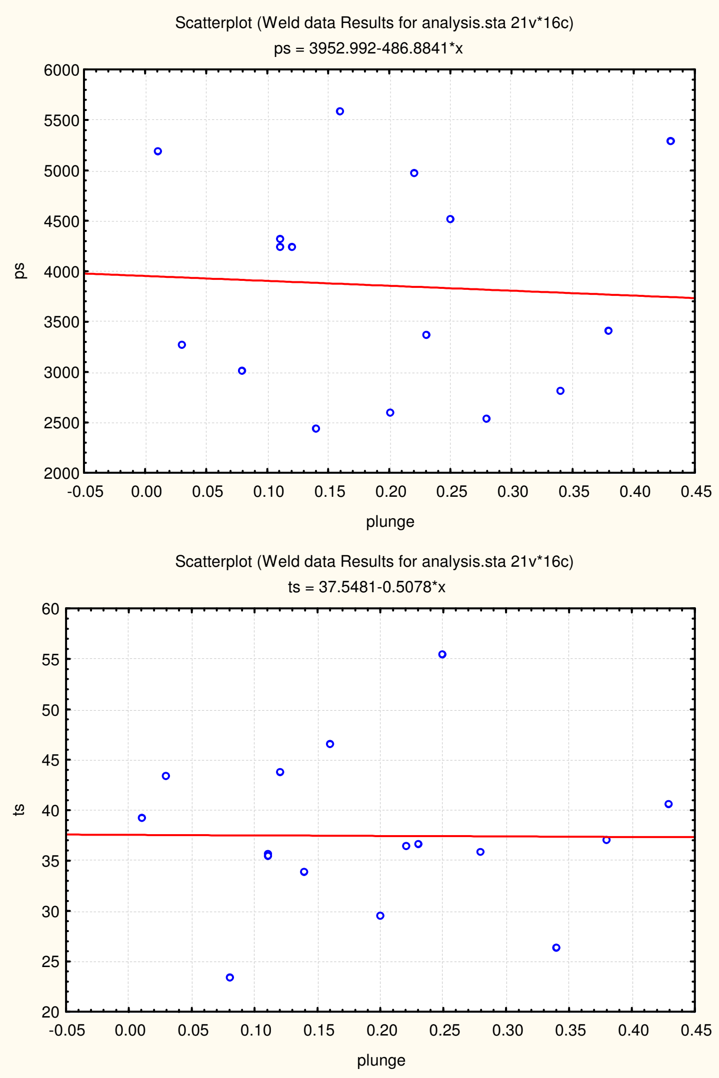

Scatterplot (Weld data Results for analysis.sta 21v*16c)
t = 420.5476+0.1078*x
t
540
520
500
480
460
440
420
400
250
300
350
400
450
500
550
600
650
rpm

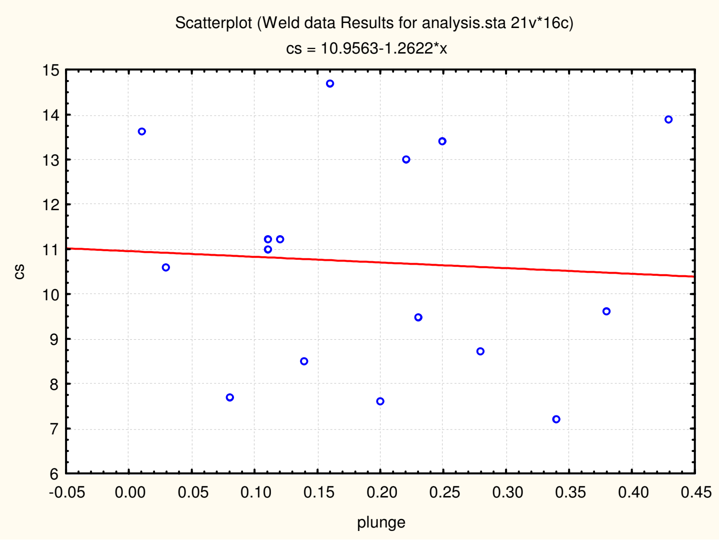

Scatterplot (Weld data Results for analysis.sta 21v*16c)
fx = 1226.7103-0.7232*x
fx
2000
1800
1600
1400
1200
1000
800
600
400
250
300
350
400
450
500
550
600
650
rpm

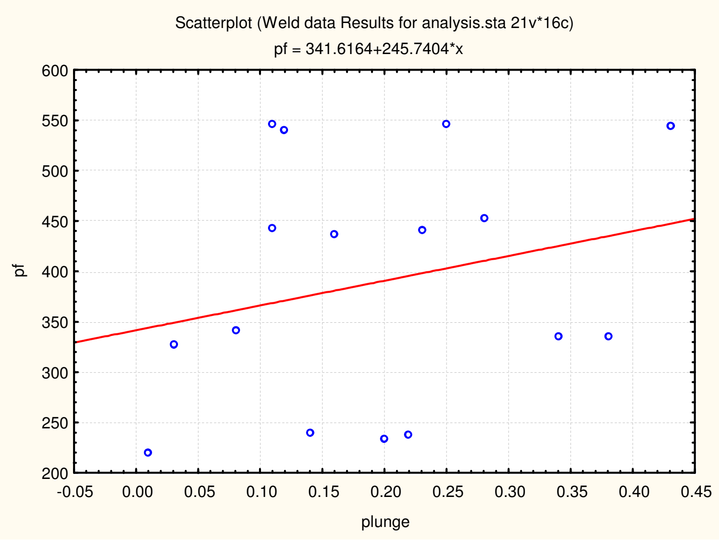

Scatterplot (Weld data Results for analysis.sta 21v*16c)
fy = 258.0404-0.2064*x
fy
300
280
260
240
220
200
180
160
140
120
100
80
60
40
250
300
350
400
450
500
550
600
650
rpm

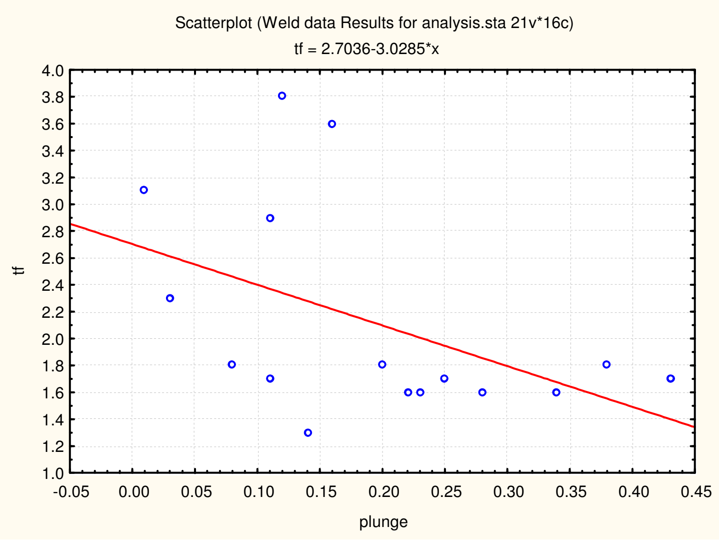

Scatterplot (Weld data Results for analysis.sta 21v*16c)
r = 4.724+0.0054*x
r
rpm
0
2
4
6
8
10
12
14
16
18
20
22
24
26
250
300
350
400
450
500
550
600
650

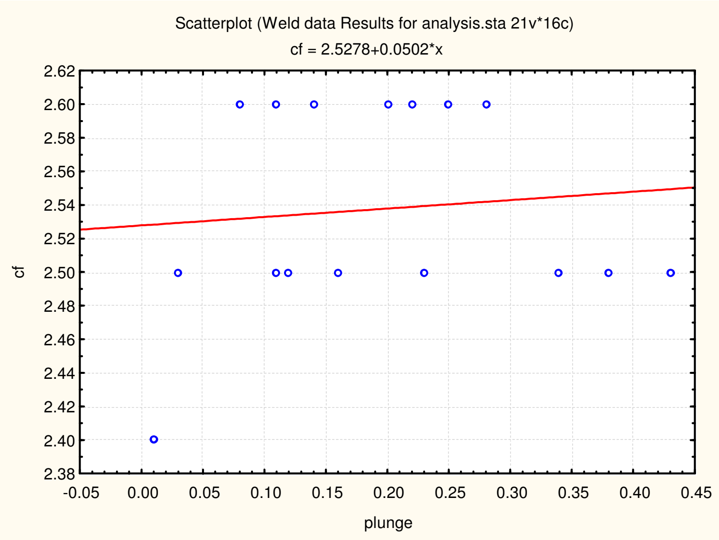

Scatterplot (Weld data Results for analysis.sta 21v*16c)
fz = -16.7409-0.0019*x
fz
rpm
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
250
300
350
400
450
500
550
600
650

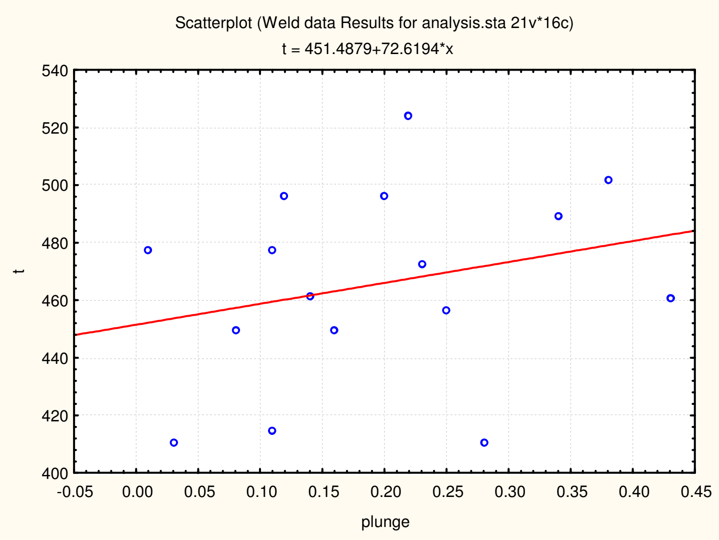

Scatterplot (Weld data Results for analysis.sta 21v*16c)
ps = 3952.992-486.8841*x
ps
6000
5500
5000
4500
4000
3500
3000
2500
2000
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge
Scatterplot (Weld data Results for analysis.sta 21v*16c)
ts = 37.5481-0.5078*x
ts
60
55
50
45
40
35
30
25
20
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge

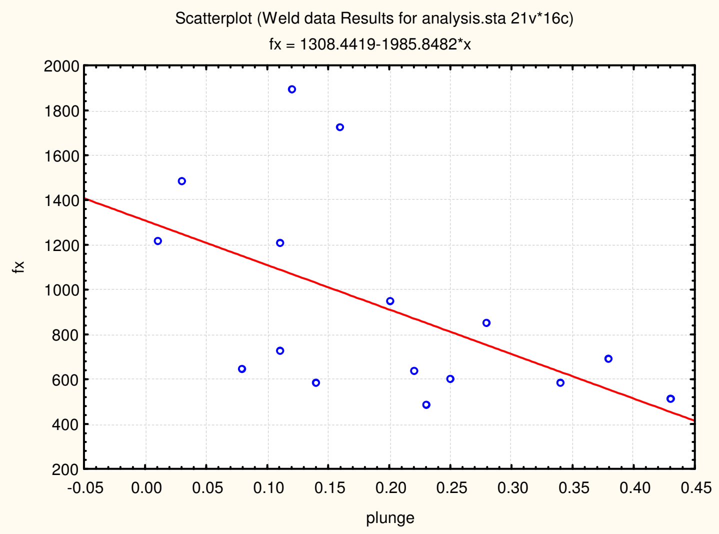

Scatterplot (Weld data Results for analysis.sta 21v*16c)
cs = 10.9563-1.2622*x
cs
15
14
13
12
11
10
9
8
7
6
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge

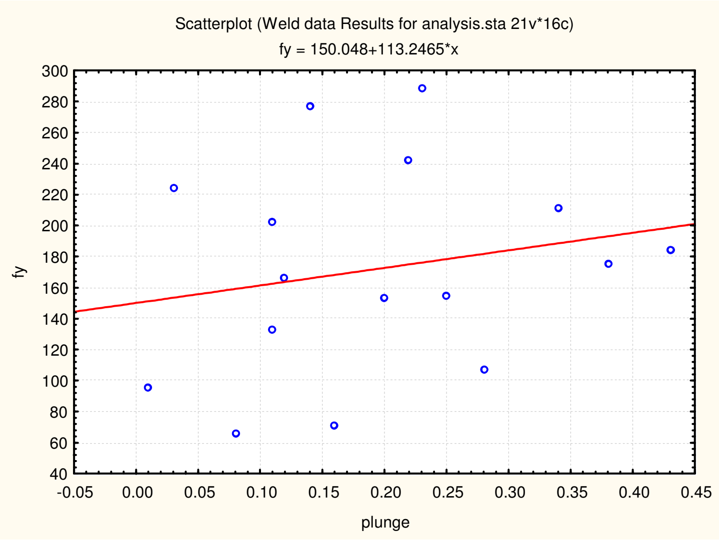

Scatterplot (Weld data Results for analysis.sta 21v*16c)
pf = 341.6164+245.7404*x
pf
600
550
500
450
400
350
300
250
200
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge

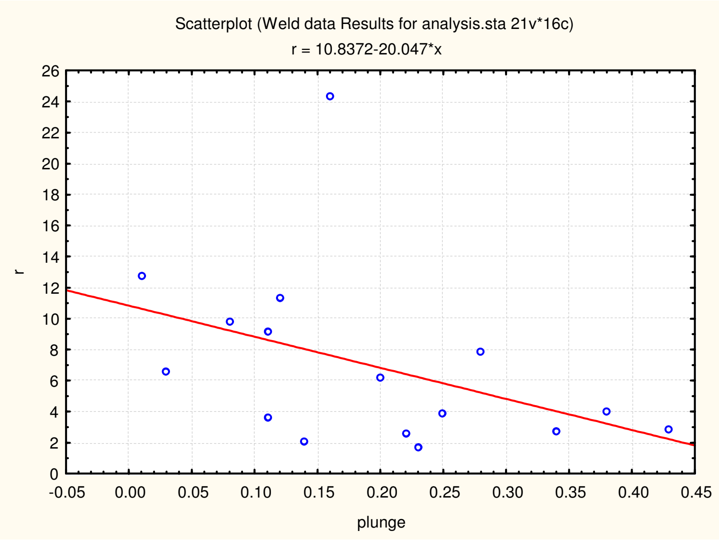

Scatterplot (Weld data Results for analysis.sta 21v*16c)
tf = 2.7036-3.0285*x
tf
plunge
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

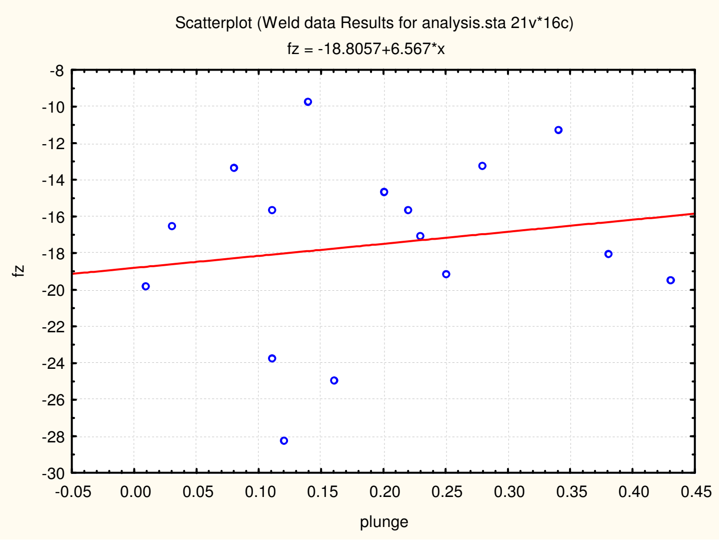

Scatterplot (Weld data Results for analysis.sta 21v*16c)
cf = 2.5278+0.0502*x
cf
plunge
2.62
2.60
2.58
2.56
2.54
2.52
2.50
2.48
2.46
2.44
2.42
2.40
2.38
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45


Scatterplot (Weld data Results for analysis.sta 21v*16c)
t = 451.4879+72.6194*x
t
540
520
500
480
460
440
420
400
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge


Scatterplot (Weld data Results for analysis.sta 21v*16c)
fx = 1308.4419-1985.8482*x
fx
2000
1800
1600
1400
1200
1000
800
600
400
200
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
plunge

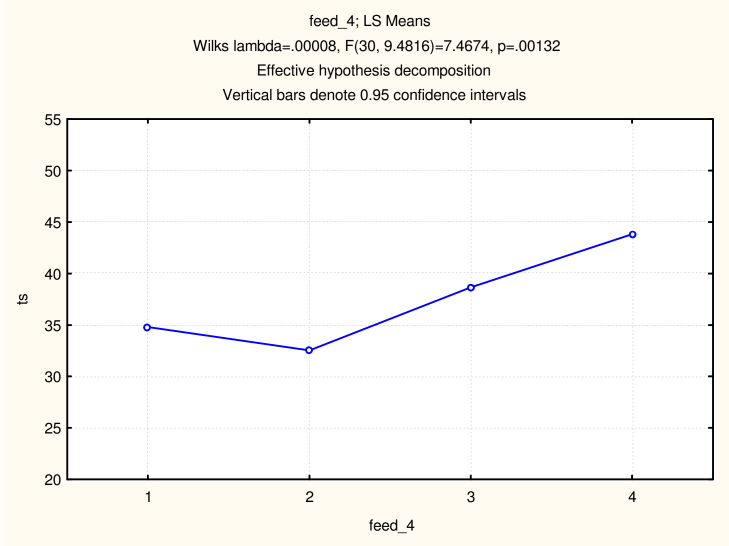

Scatterplot (Weld data Results for analysis.sta 21v*16c)
fy = 150.048+113.2465*x
fy
plunge
300
280
260
240
220
200
180
160
140
120
100
80
60
40
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

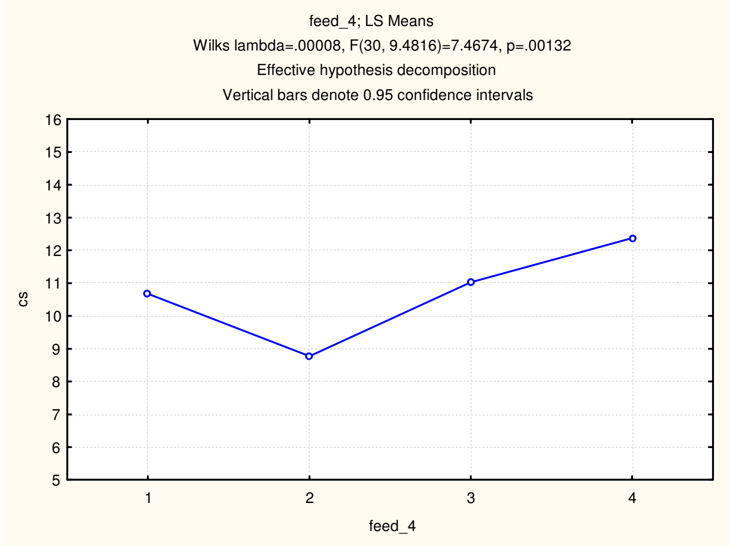

Scatterplot (Weld data Results for analysis.sta 21v*16c)
r = 10.8372-20.047*x
r
plunge
26
24
22
20
18
16
14
12
10
8
6
4
2
0
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

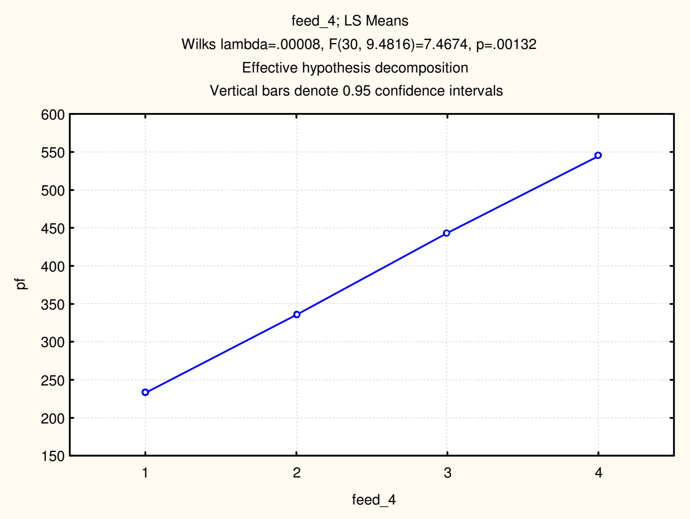


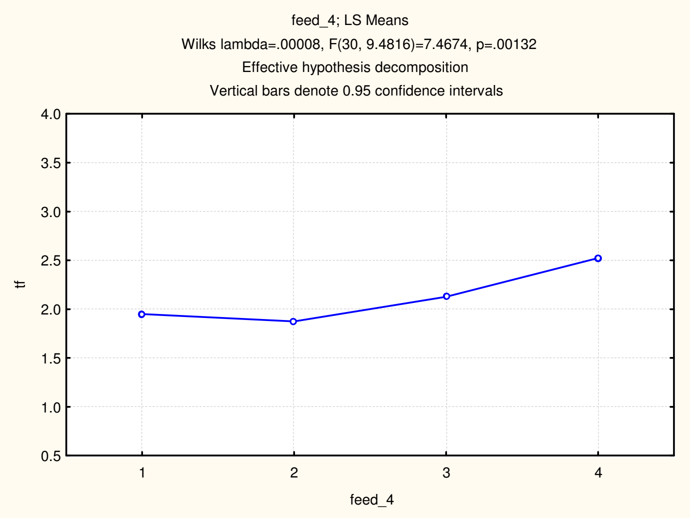
Scatterplot (Weld data Results for analysis.sta 21v*16c)
fz = -18.8057+6.567*x
fz
plunge
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45

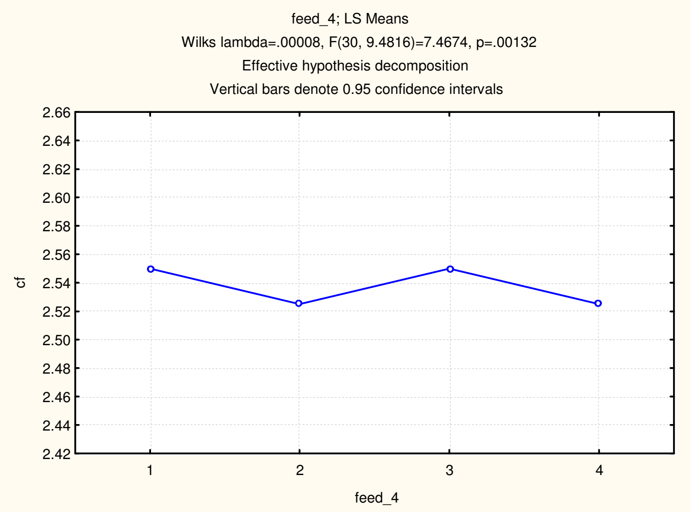
F.3: ANOVA Plots relating Dependent to Independent Variables

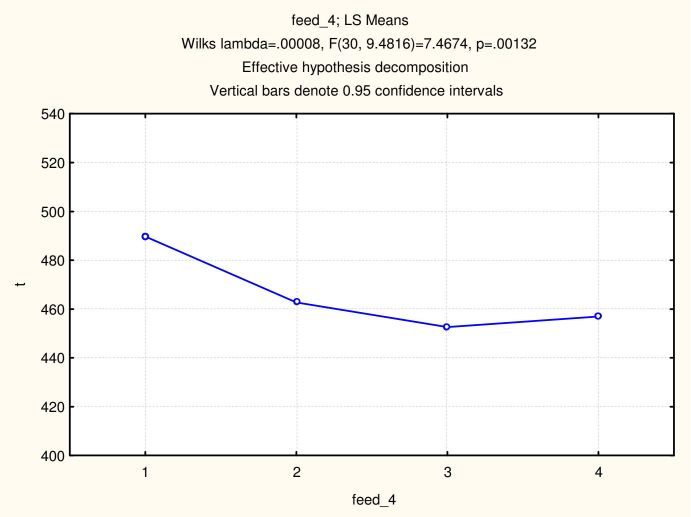

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
55
50
45
40
35
30
25
20
ts
1
2
3
4
feed_4

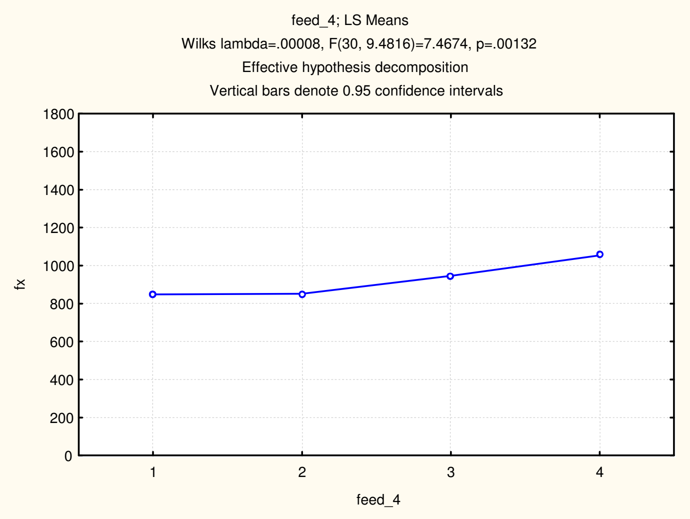

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
16
15
14
13
12
11
10
9
8
7
6
5
cs
1
2
3
4
feed_4

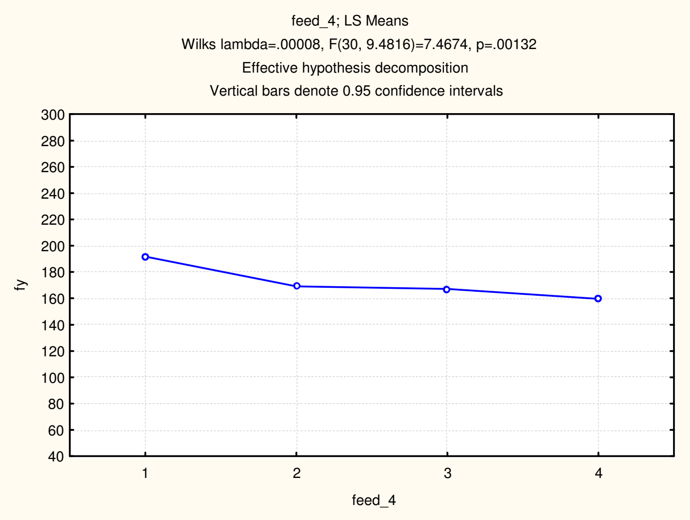

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
pf
600
550
500
450
400
350
300
250
200
150
1
2
3
4
feed_4

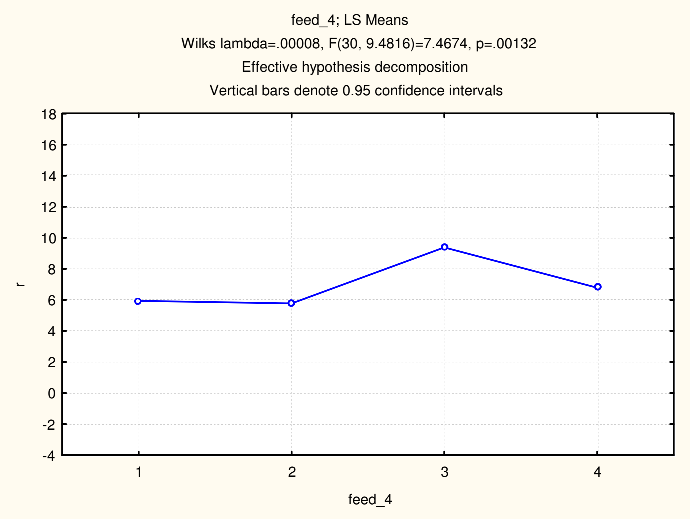

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
tf
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
1
2
3
4
feed_4

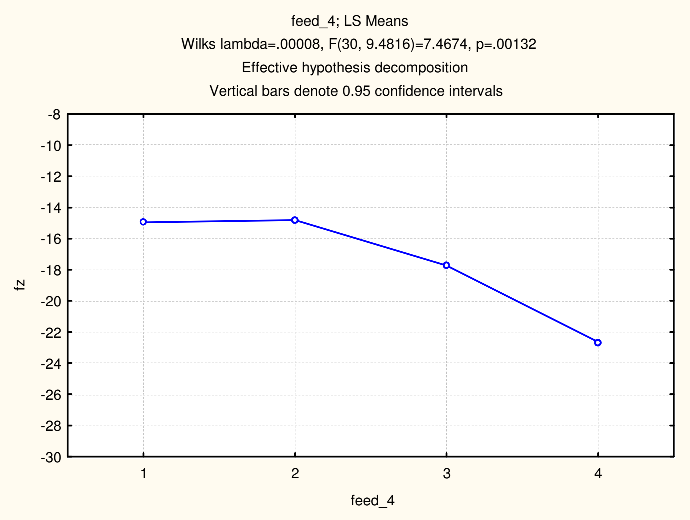

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
cf
2.66
2.64
2.62
2.60
2.58
2.56
2.54
2.52
2.50
2.48
2.46
2.44
2.42
1
2
3
4
feed_4

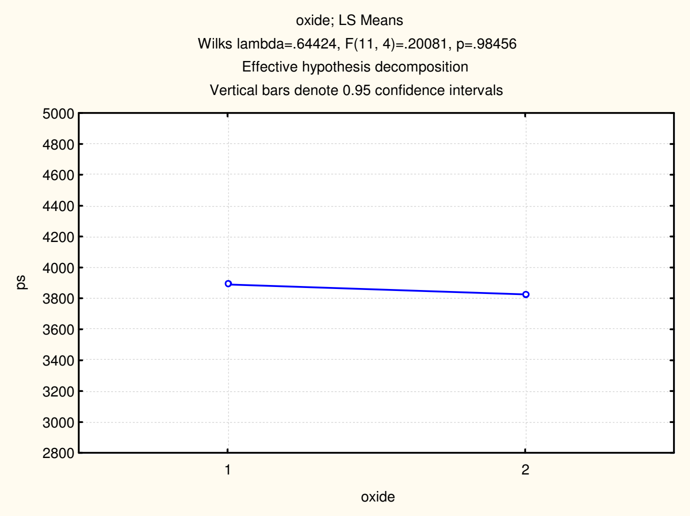

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
t
540
520
500
480
460
440
420
400
1
2
3
4
feed_4

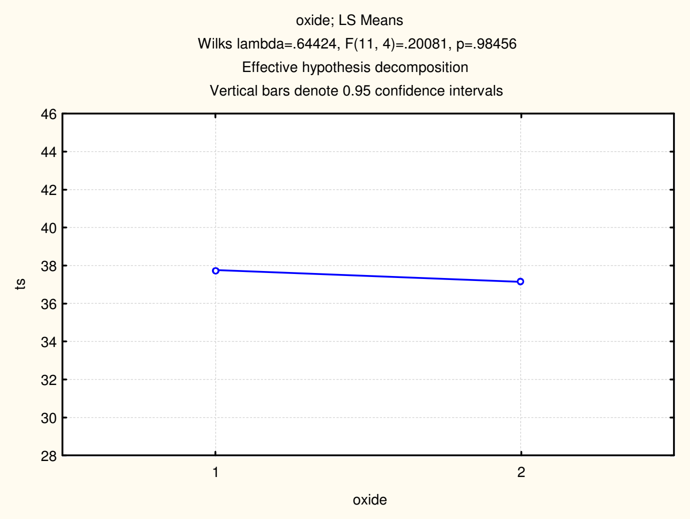

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
1800
1600
1400
1200
1000
800
600
400
200
0
fx
1
2
3
4
feed_4

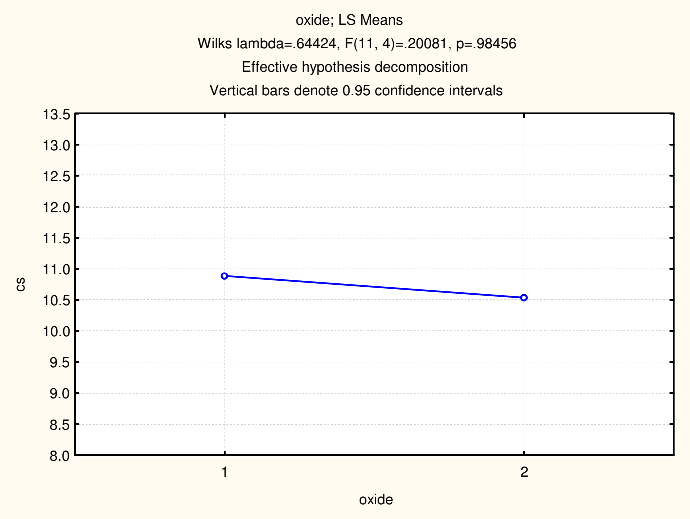

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
300
280
260
240
220
200
180
160
140
120
100
80
60
40
fy
1
2
3
4
feed_4

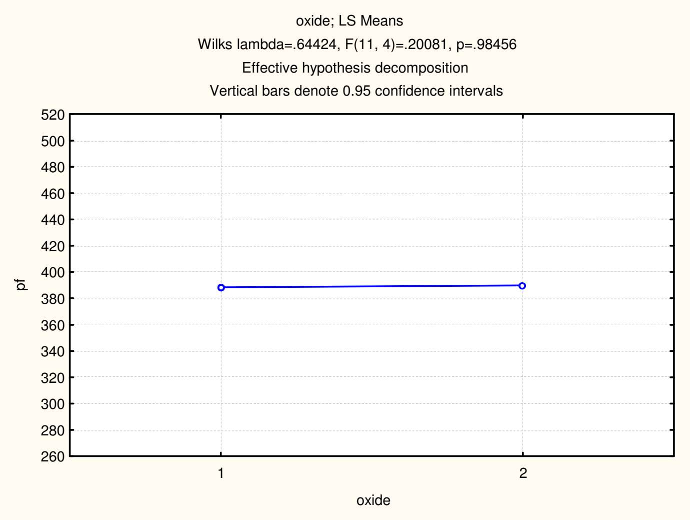

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
18
16
14
12
10
8
6
4
2
0
-2
-4
r
1
2
3
4
feed_4

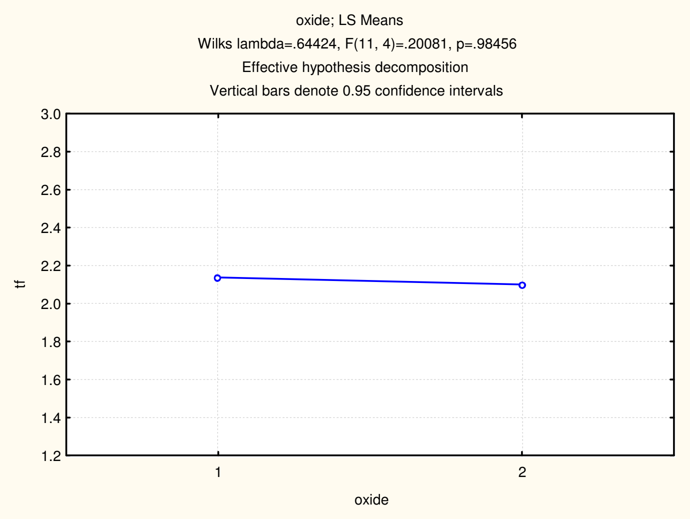

feed_4; LS Means
Wilks lambda=.00008, F(30, 9.4816)=7.4674, p=.00132
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
fz
1
2
3
4
feed_4

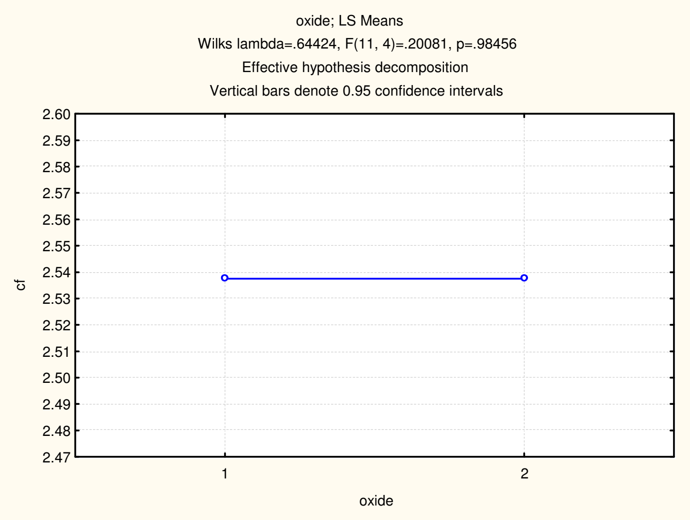

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
5000
4800
4600
4400
4200
4000
3800
3600
3400
3200
3000
2800
ps
1
2
oxide

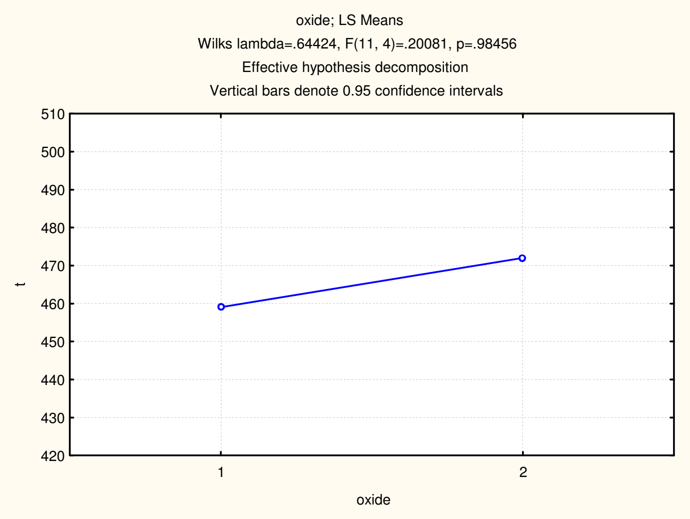

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
46
44
42
40
38
36
34
32
30
28
ts
1
2
oxide

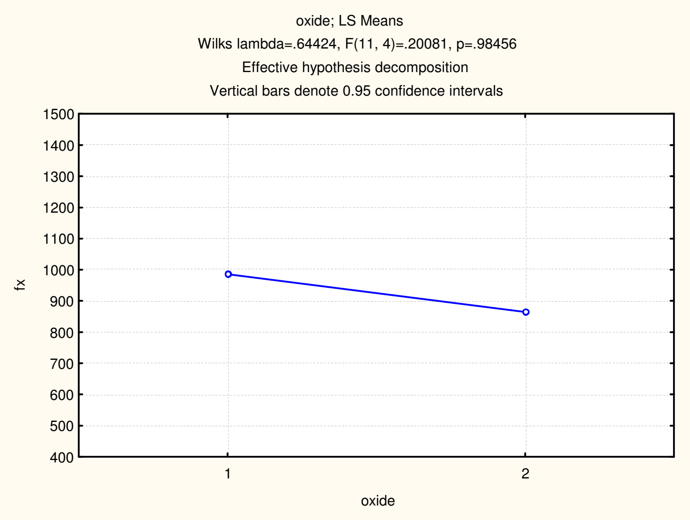

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
13.5
13.0
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
cs
1
2
oxide

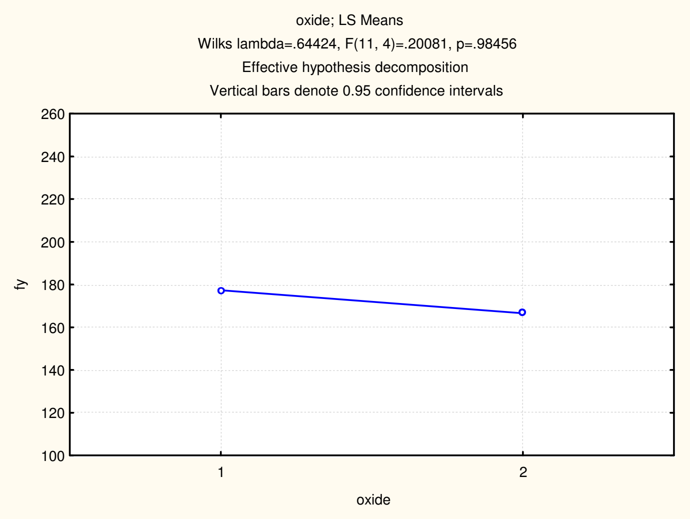

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
520
500
480
460
440
420
400
380
360
340
320
300
280
260
pf
1
2
oxide

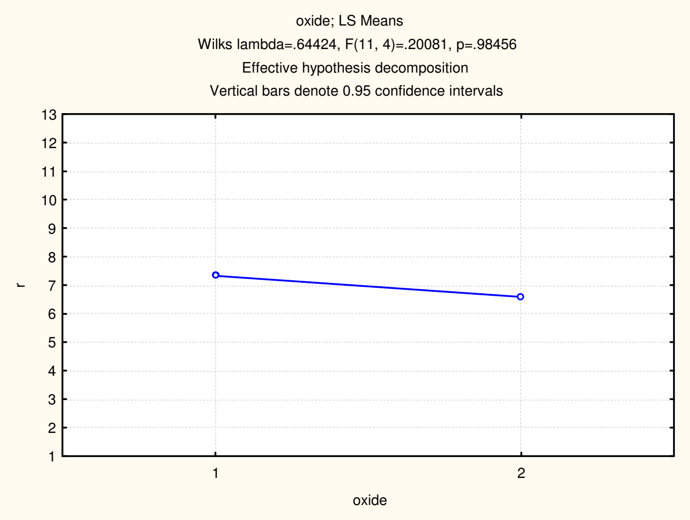

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
tf
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1
2
oxide

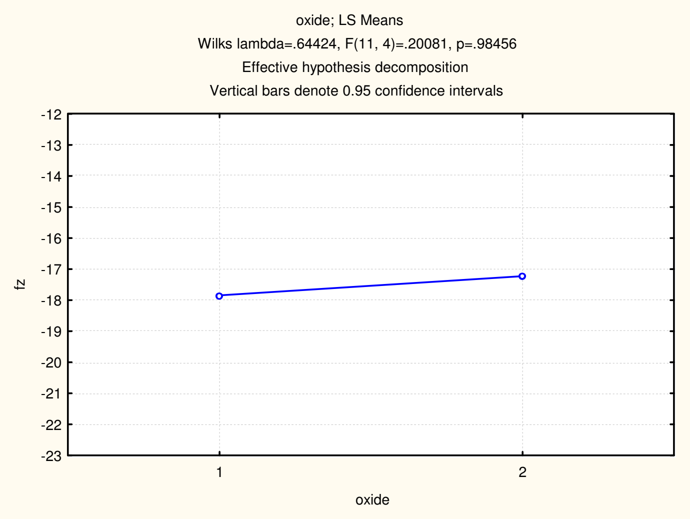

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
cf
2.60
2.59
2.58
2.57
2.56
2.55
2.54
2.53
2.52
2.51
2.50
2.49
2.48
2.47
1
2
oxide

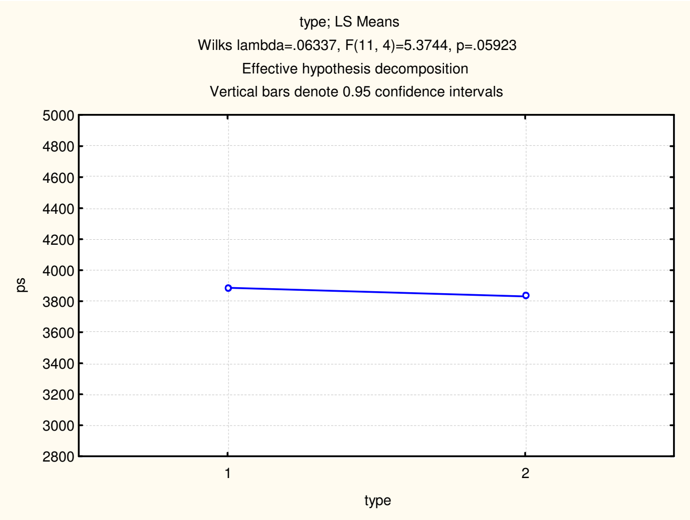

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals

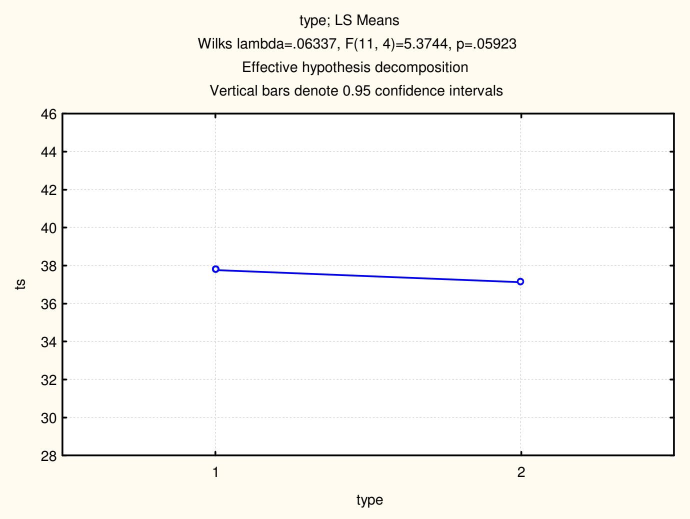

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals

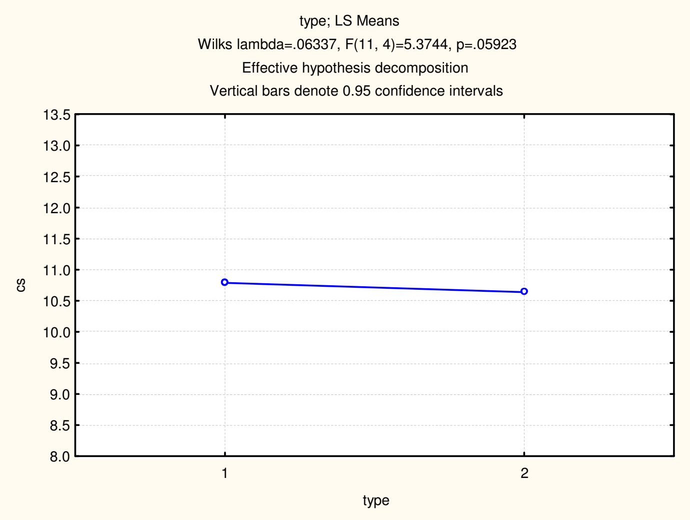

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
260
240
220
200
180
160
140
120
100
fy
1
2
oxide

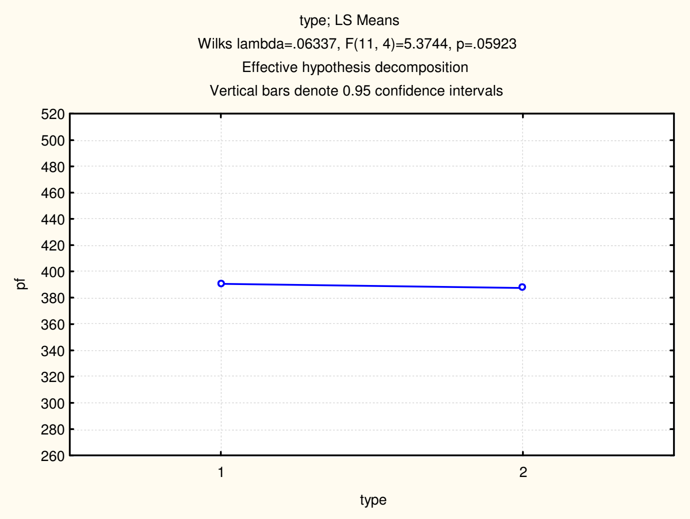

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
13
12
11
10
9
8
7
6
5
4
3
2
1
r
1
2
oxide

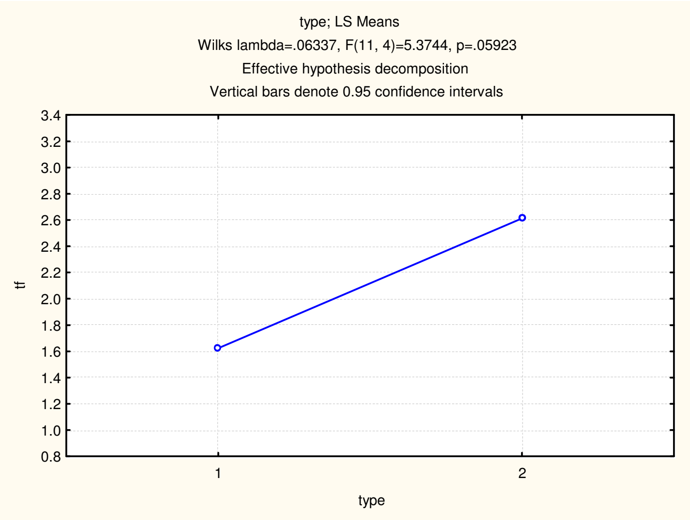

oxide; LS Means
Wilks lambda=.64424, F(11, 4)=.20081, p=.98456
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
fz
-12
-13
-14
-15
-16
-17
-18
-19
-20
-21
-22
-23
1
2
oxide

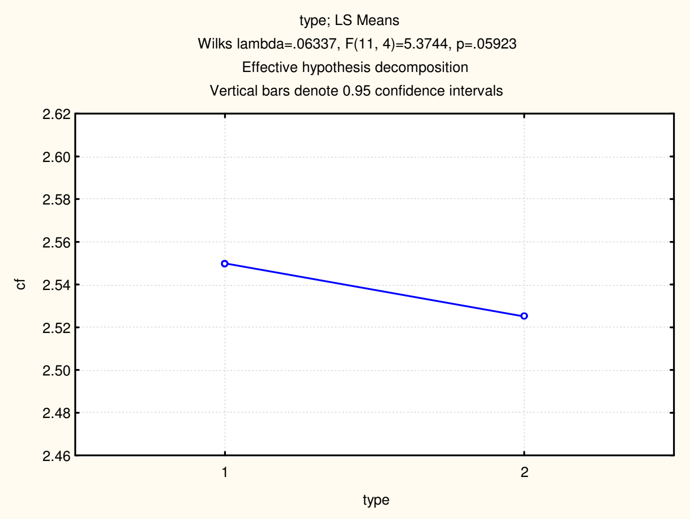

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
ps
5000
4800
4600
4400
4200
4000
3800
3600
3400
3200
3000
2800
1
2
type

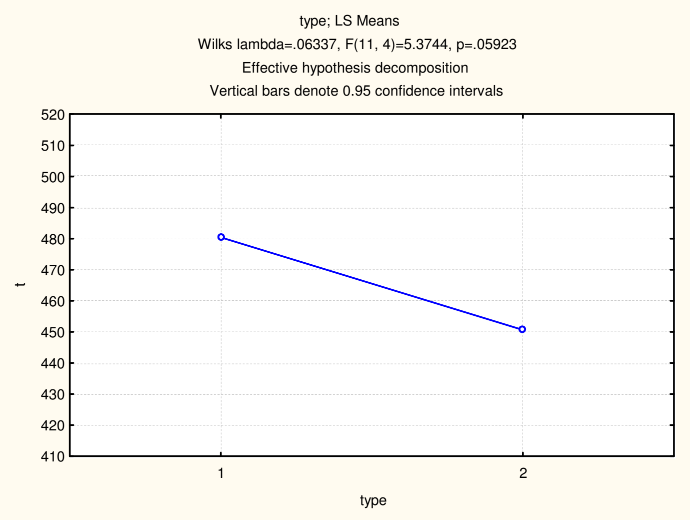

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
46
44
42
40
38
36
34
32
30
28
ts
1
2
type

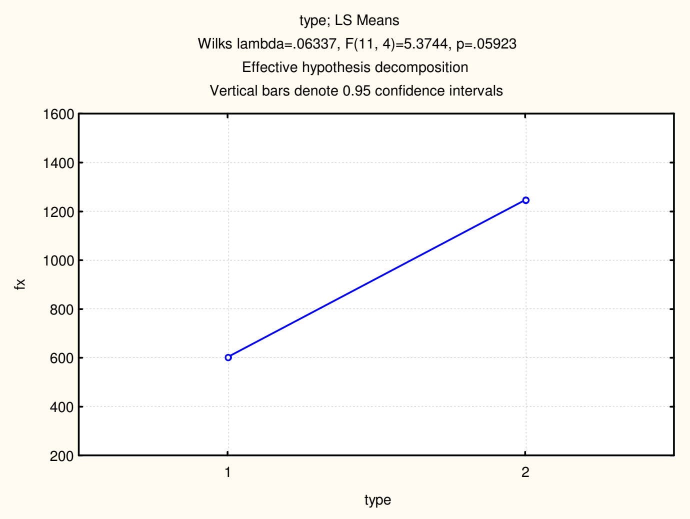

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
13.5
13.0
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
cs
1
2
type

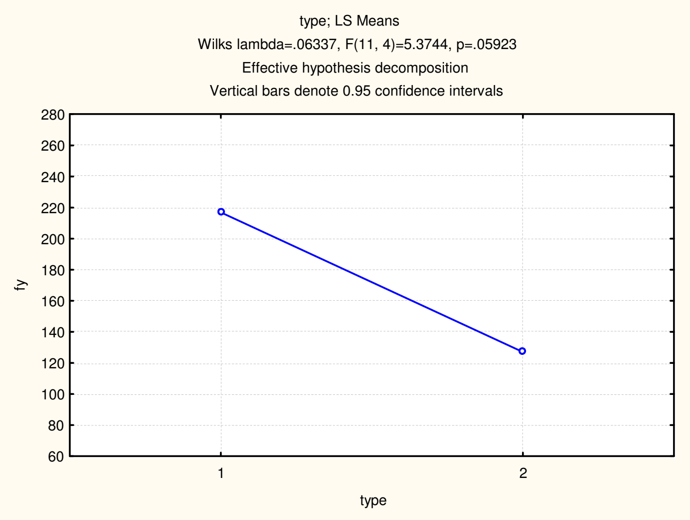

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
pf
520
500
480
460
440
420
400
380
360
340
320
300
280
260
1
2
type

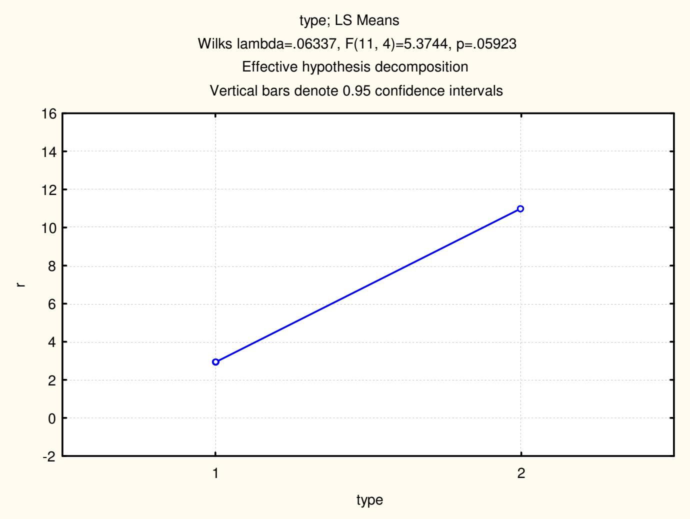

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
tf
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
1
2
type

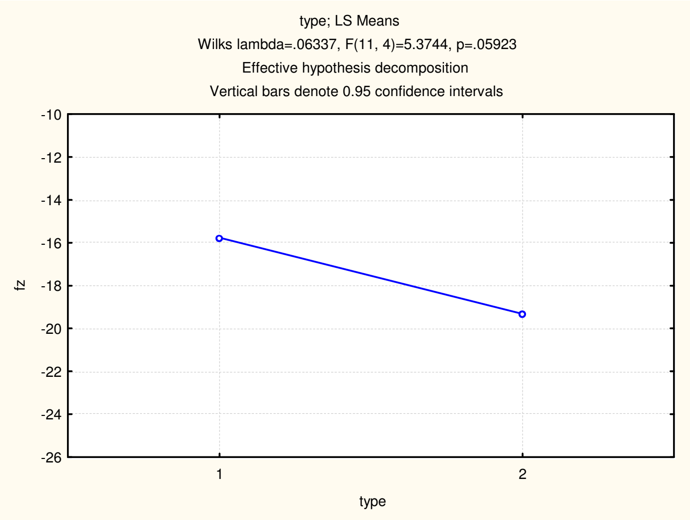

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
2.62
2.60
2.58
2.56
2.54
2.52
2.50
2.48
2.46
cf
1
2
type

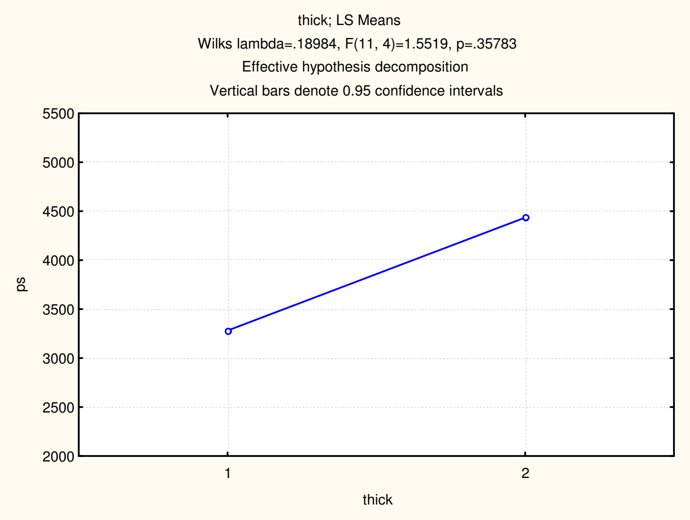

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
520
510
500
490
480
470
460
450
440
430
420
410
t
1
2
type

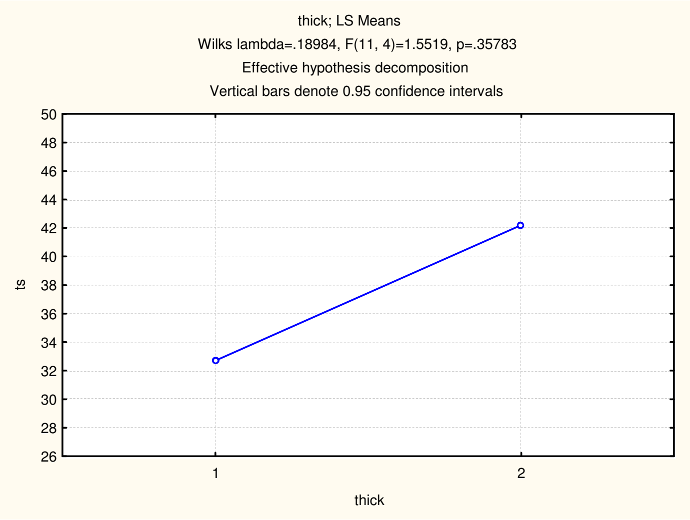

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
fx
1600
1400
1200
1000
800
600
400
200
1
2
type

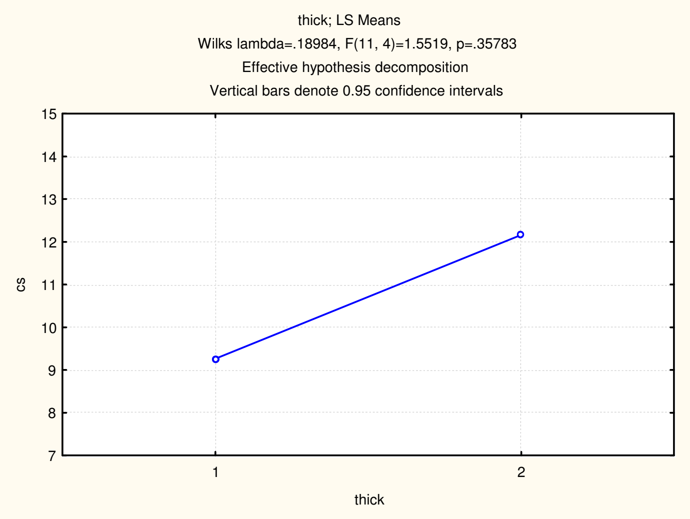

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
fy
280
260
240
220
200
180
160
140
120
100
80
60
1
2
type

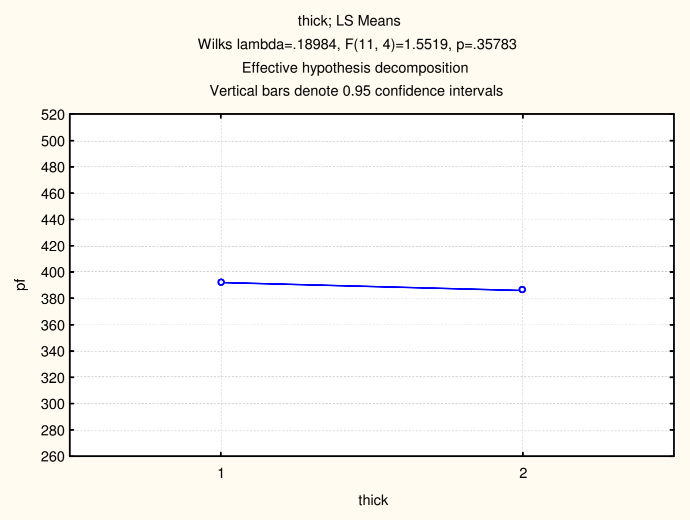

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
16
14
12
10
8
6
4
2
0
-2
r
1
2
type

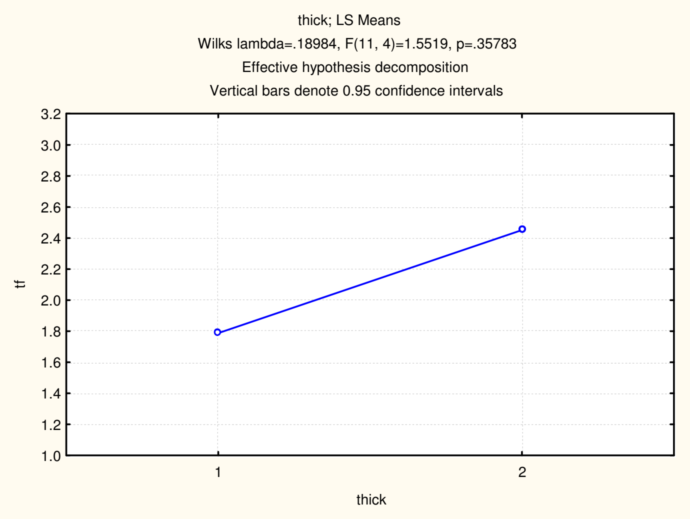

type; LS Means
Wilks lambda=.06337, F(11, 4)=5.3744, p=.05923
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
-10
-12
-14
-16
-18
-20
-22
-24
-26
fz
1
2
type


thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
5500
5000
4500
4000
3500
3000
2500
2000
ps
1
2
thick


thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
50
48
46
44
42
40
38
36
34
32
30
28
26
ts
1
2
thick


thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
15
14
13
12
11
10
9
8
7
cs
1
2
thick


thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
520
500
480
460
440
420
400
380
360
340
320
300
280
260
pf
1
2
thick

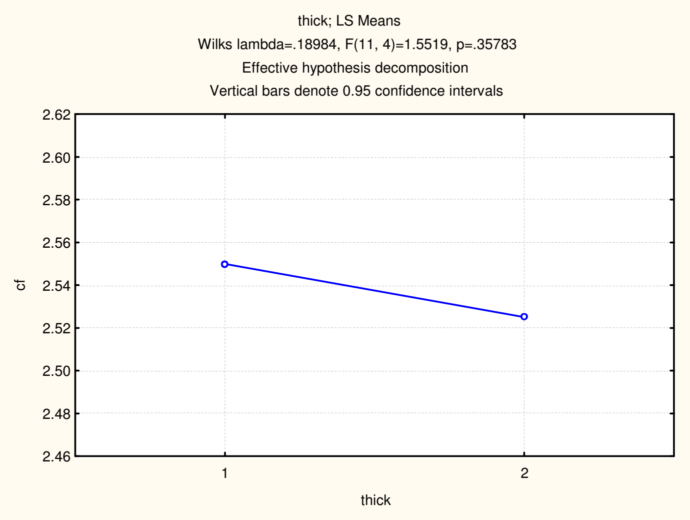


thick; LS Means

Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783

Effective hypothesis decomposition

Vertical bars denote 0.95 confidence intervals

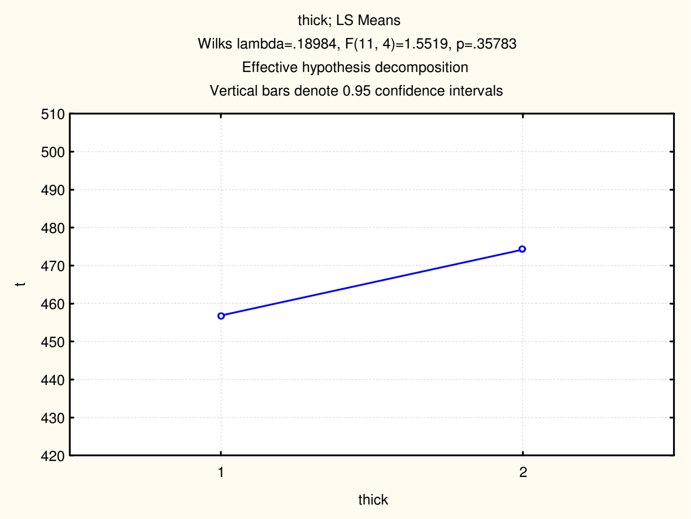


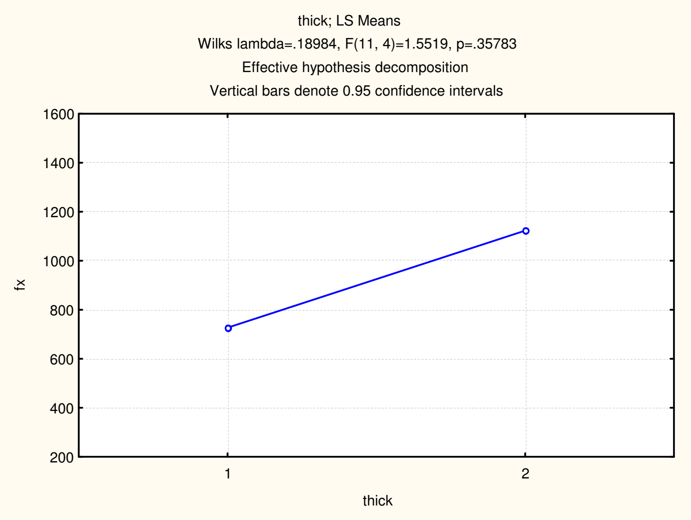
thick; LS Means

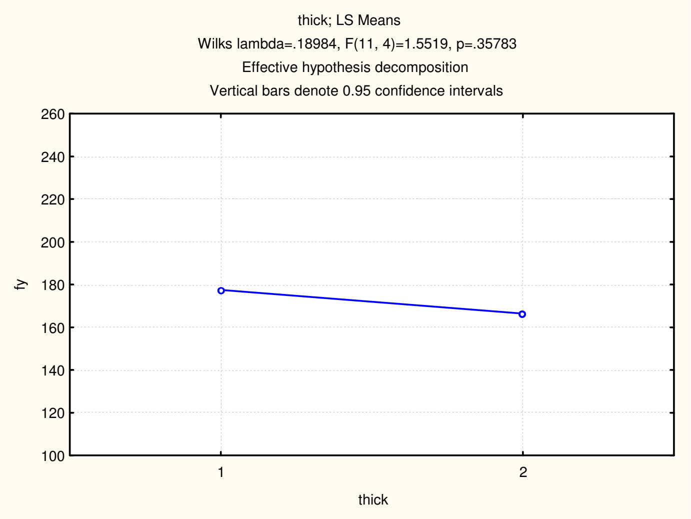
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783

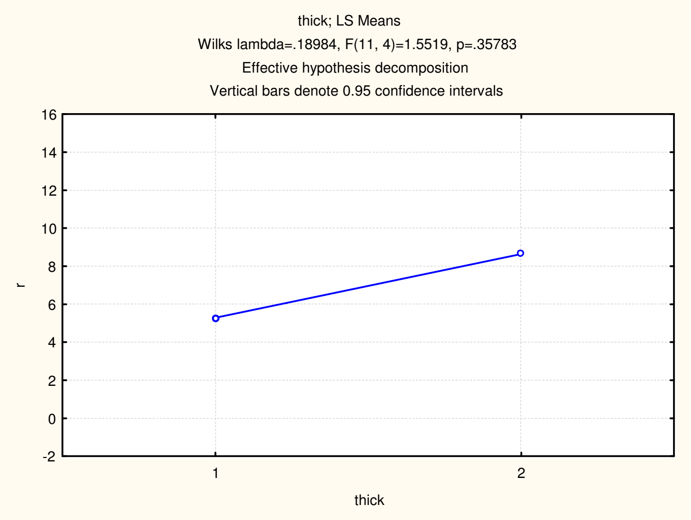
Effective hypothesis decomposition

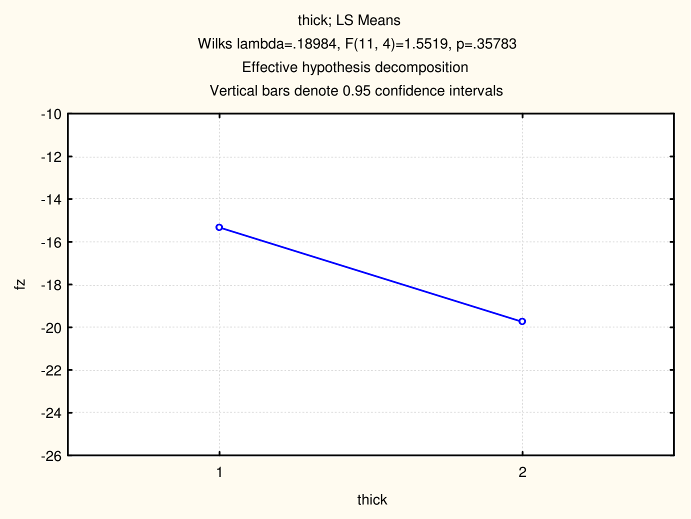
Vertical bars denote 0.95 confidence intervals

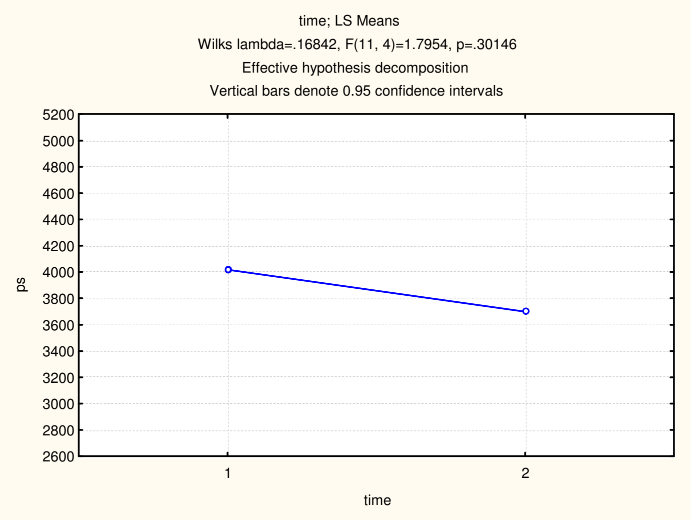

thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
510
500
490
480
470
460
450
440
430
420
t
1
2
thick

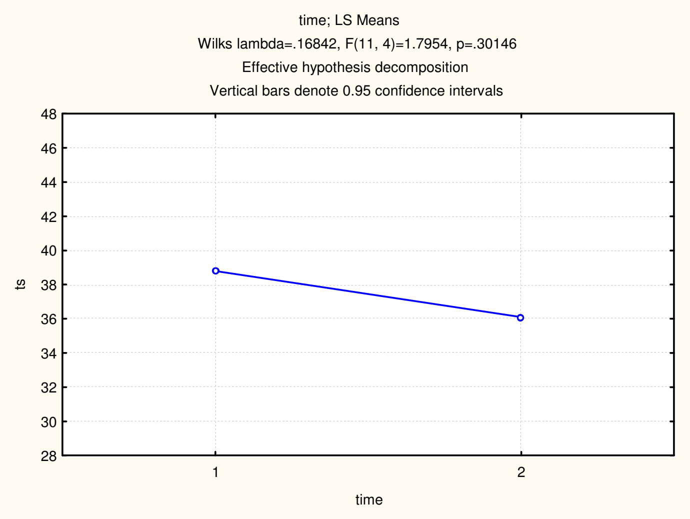

thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
1600
1400
1200
1000
800
600
400
200
fx
1
2
thick

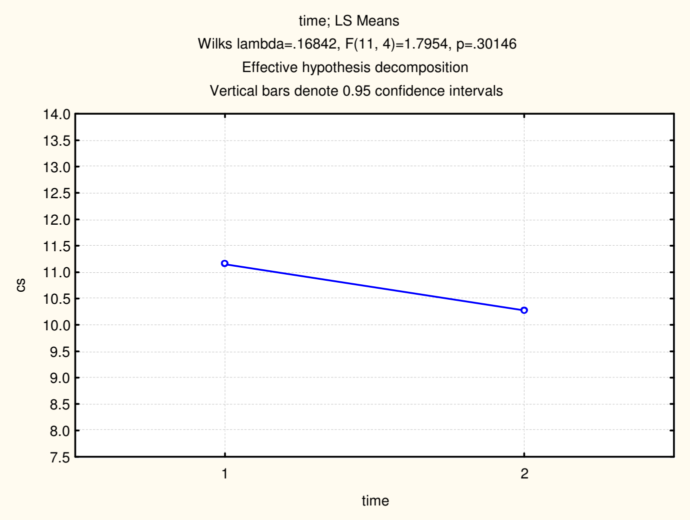

thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
fy
260
240
220
200
180
160
140
120
100
1
2
thick

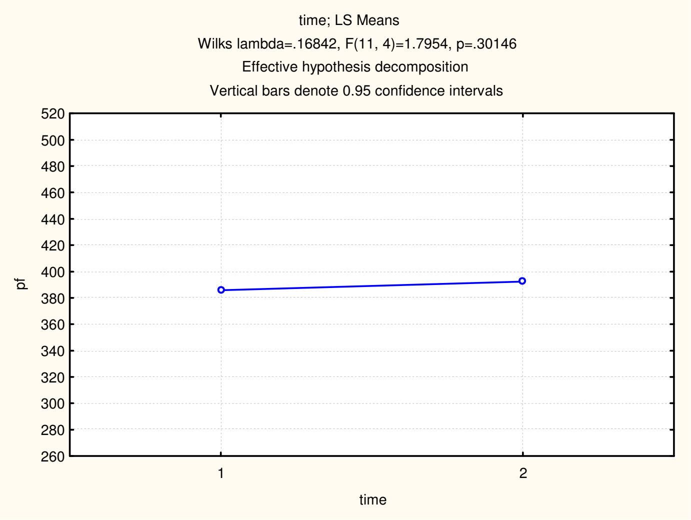

thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
r
16
14
12
10
8
6
4
2
0
-2
1
2
thick

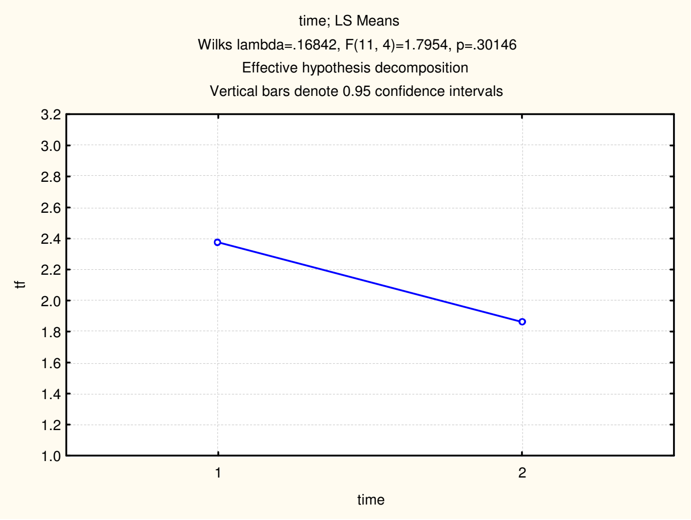

thick; LS Means
Wilks lambda=.18984, F(11, 4)=1.5519, p=.35783
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
-10
-12
-14
-16
-18
-20
-22
-24
-26
fz
1
2
thick

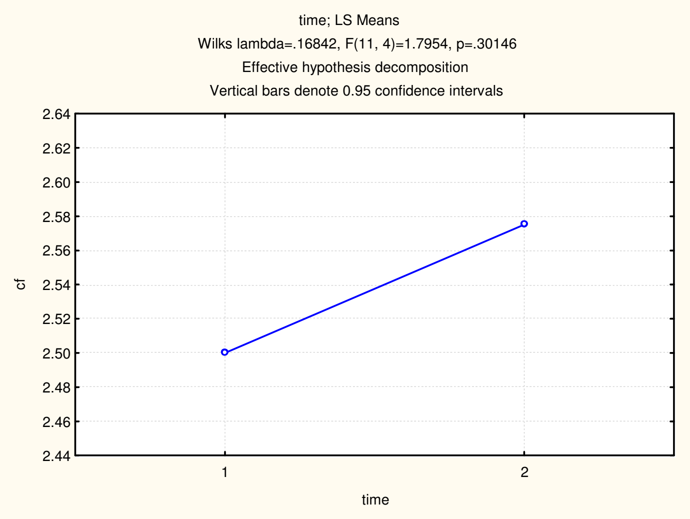

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
5200
5000
4800
4600
4400
4200
4000
3800
3600
3400
3200
3000
2800
2600
ps
1
2
time

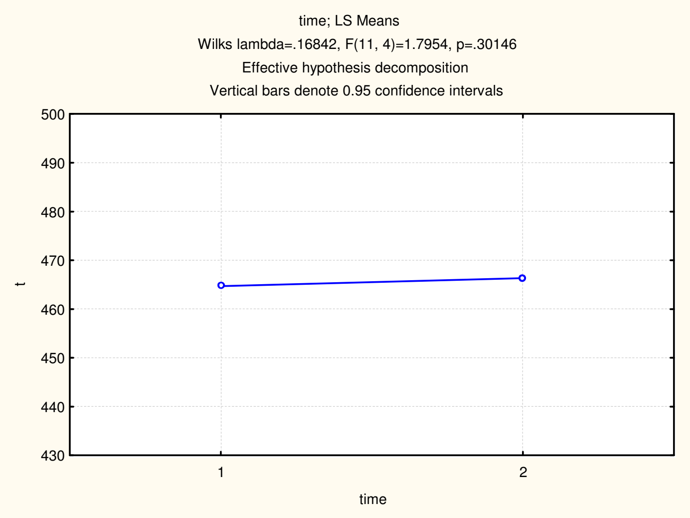

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
48
46
44
42
40
38
36
34
32
30
28
ts
1
2
time

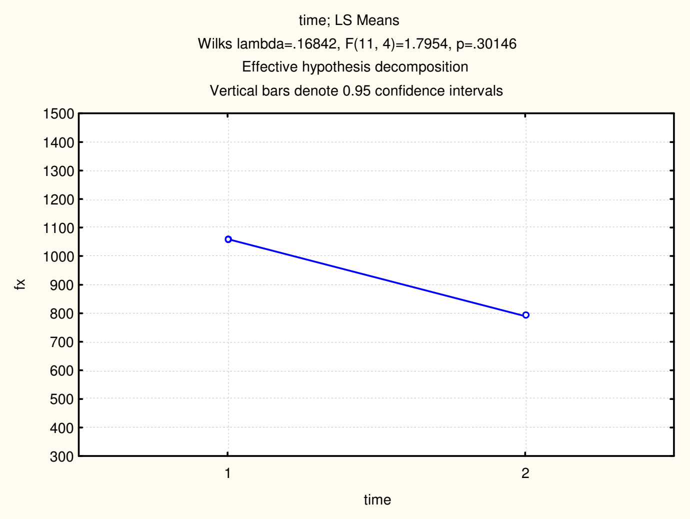

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
14.0
13.5
13.0
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
7.5
cs
1
2
time

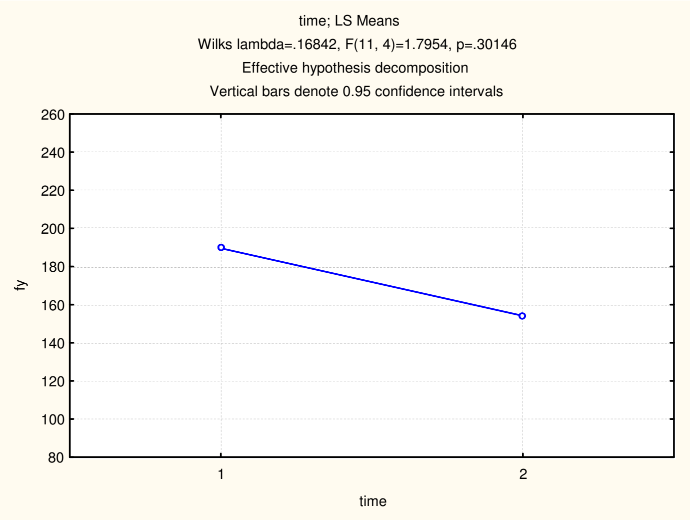

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
520
500
480
460
440
420
400
380
360
340
320
300
280
260
pf
1
2
time

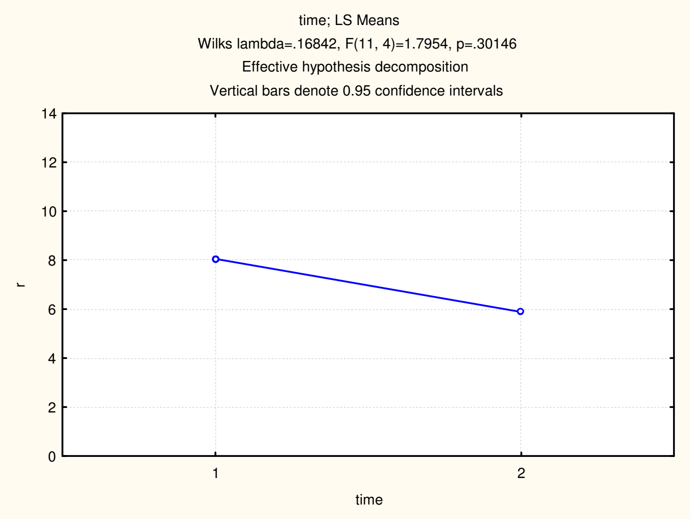

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
tf
1
2
time


time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
2.64
2.62
2.60
2.58
2.56
2.54
2.52
2.50
2.48
2.46
2.44
cf
1
2
time

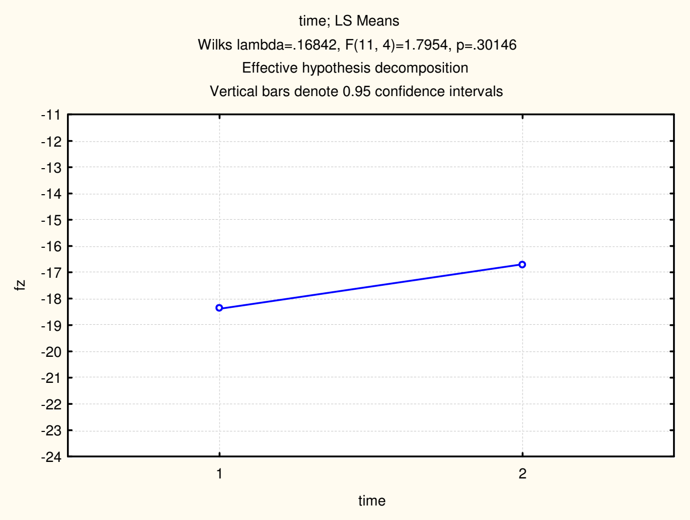

time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
500
490
480
470
460
450
440
430
t
1
2
time


time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
fx
1
2
time


time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
260
240
220
200
180
160
140
120
100
80
fy
1
2
time

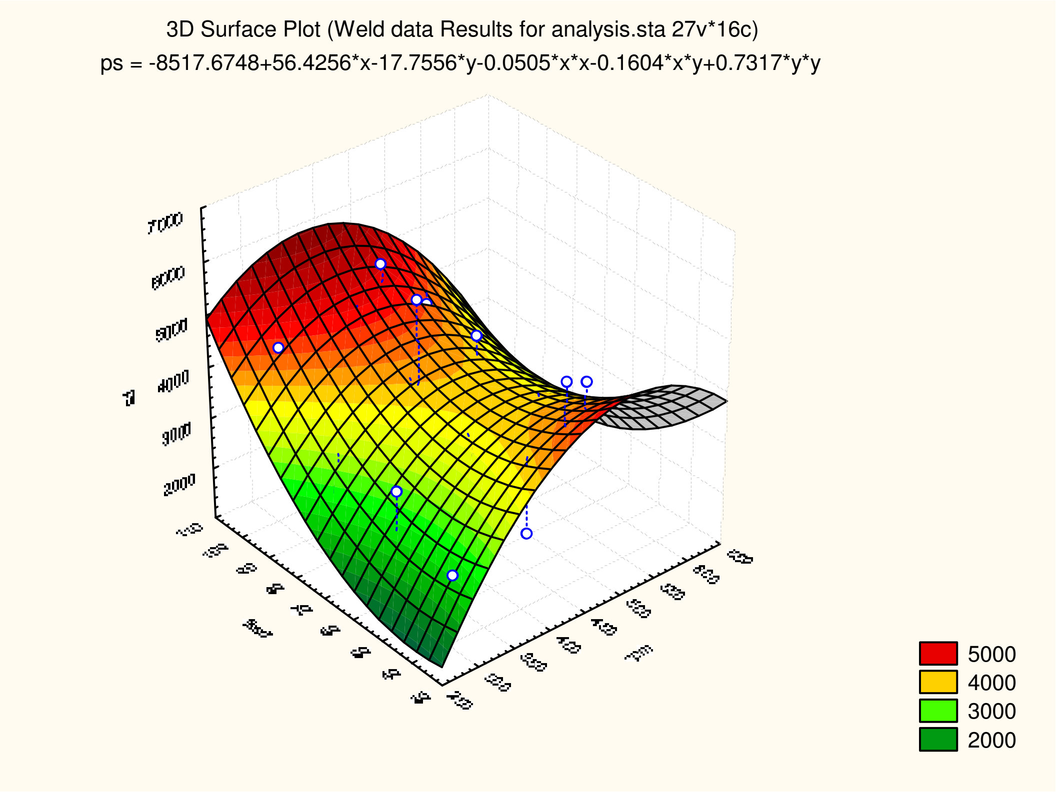


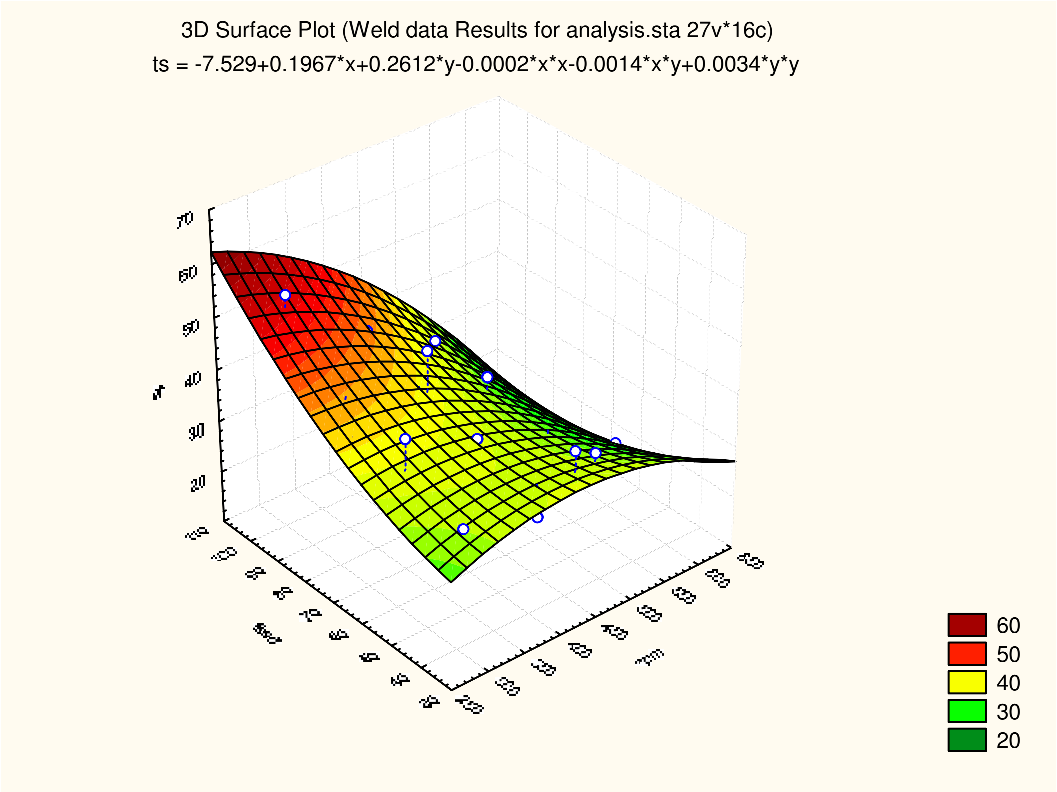
time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
14
12
10
8
6
4
2
0
r
1
2
time

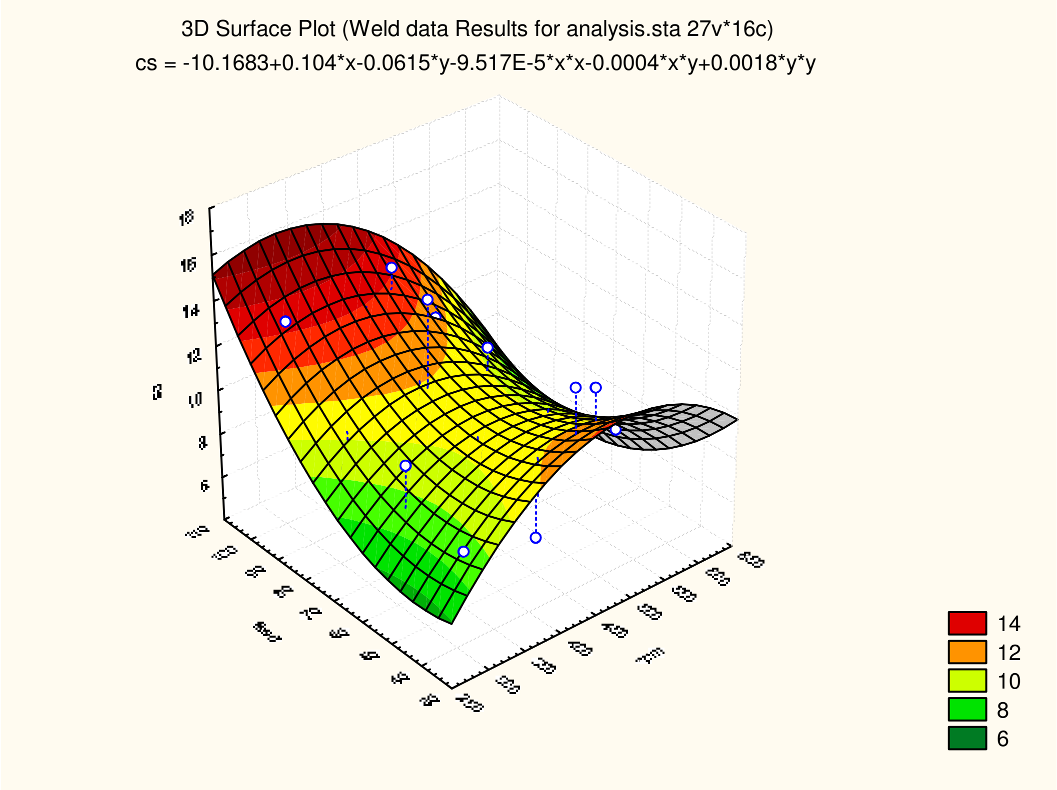


time; LS Means
Wilks lambda=.16842, F(11, 4)=1.7954, p=.30146
Effective hypothesis decomposition
Vertical bars denote 0.95 confidence intervals
-11
-12
-13
-14
-15
-16
-17
-18
-19
-20
-21
-22
-23
-24
fz
1
2
time

F.4: Surface Plots Relating Dependent to Independent Variables

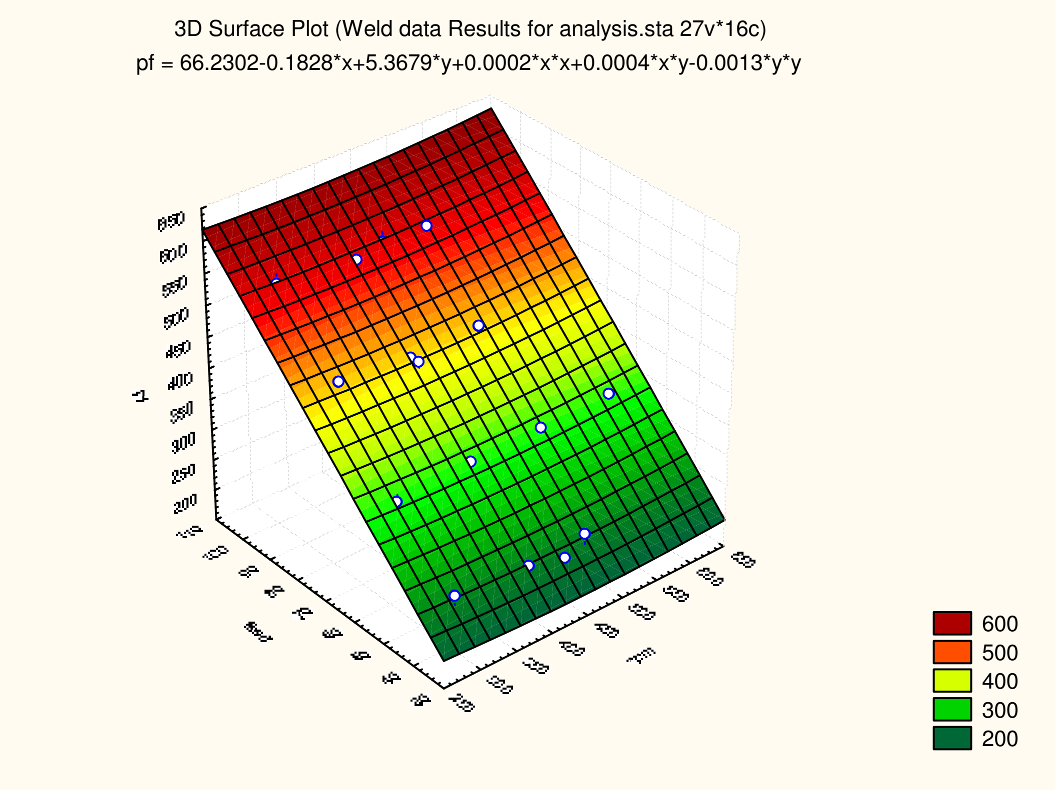


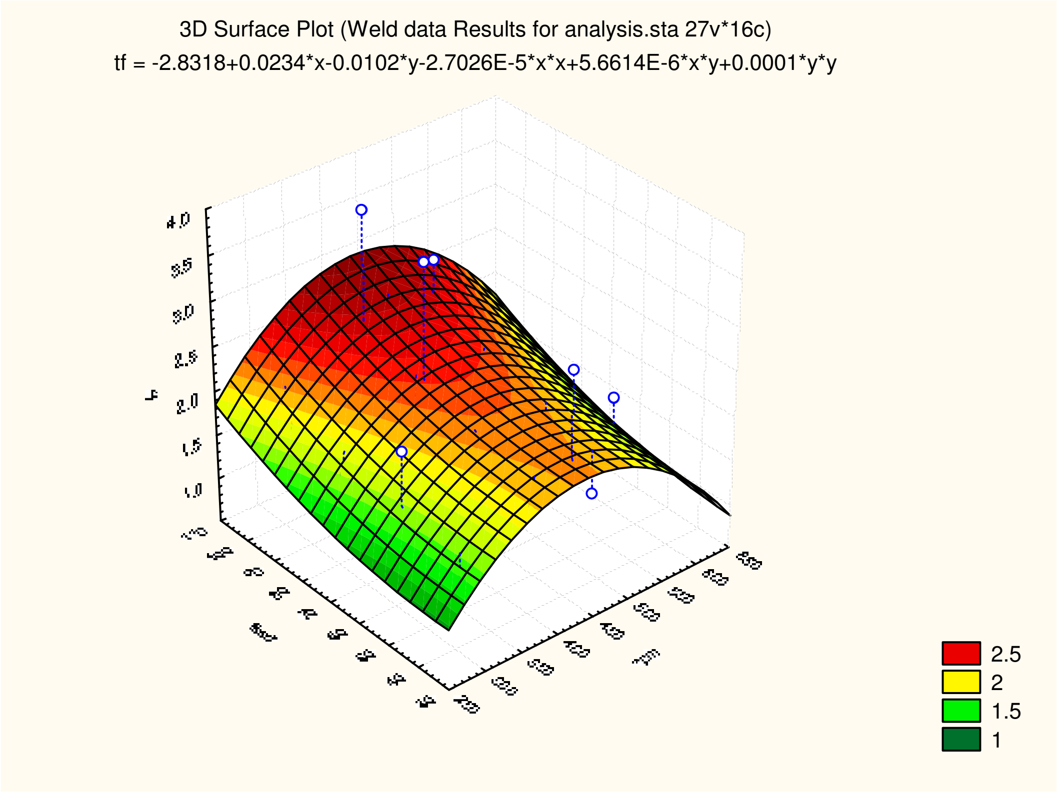

3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
ts = -7.529+0.1967*x+0.2612*y-0.0002*x*x-0.0014*x*y+0.0034*y*y
70
60
50
40
30
20
200
250
300
350
400
450
500
550
600
60
50
40
30
20

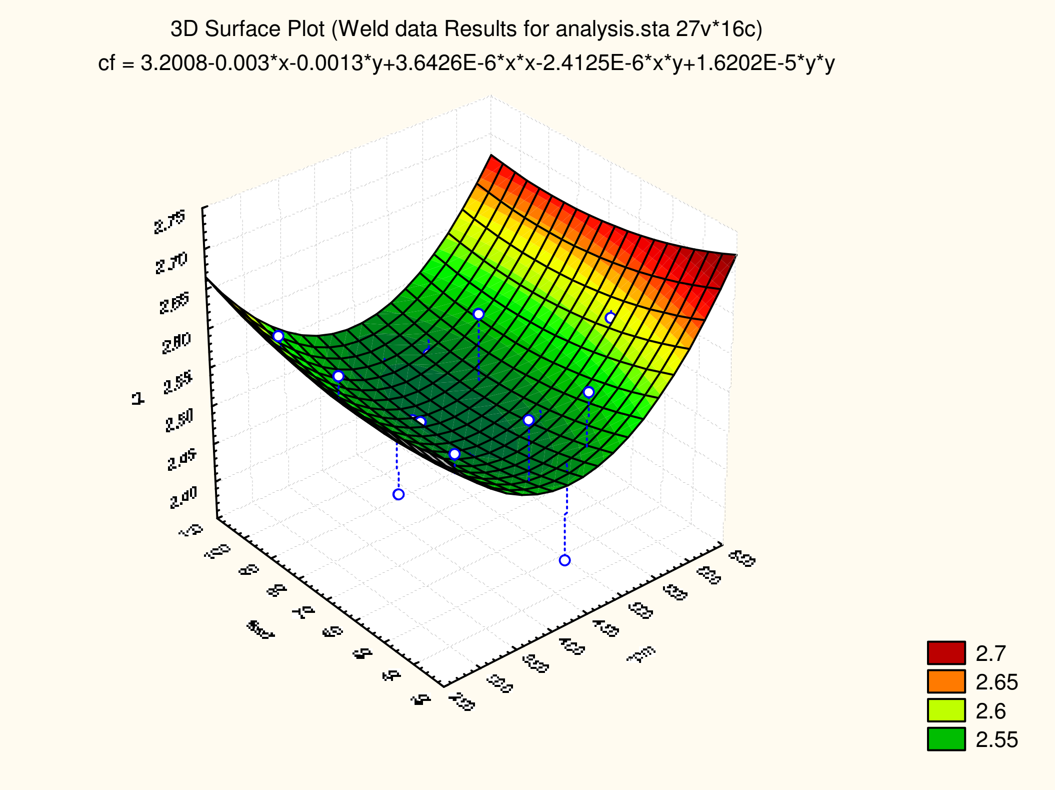


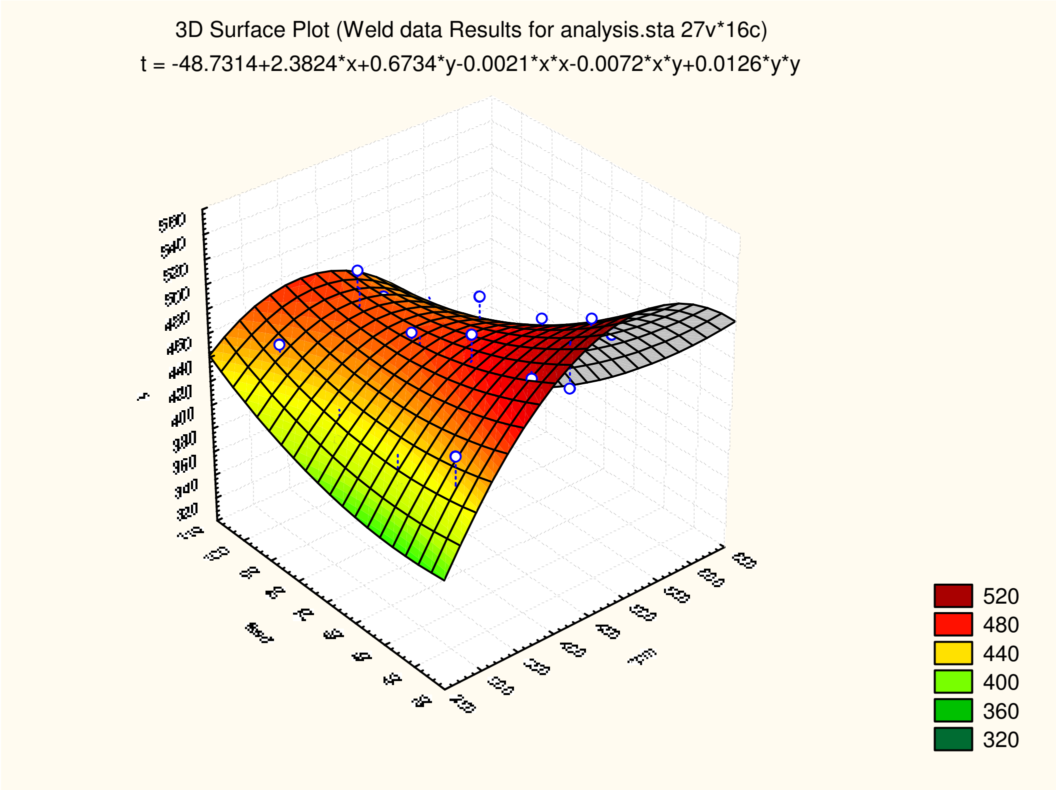
3D Surface Plot (Weld data Results for analysis.sta 27v*16c)

cs = -10.1683+0.104*x-0.0615*y-9.517E-5*x*x-0.0004*x*y+0.0018*y*y


3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
pf = 66.2302-0.1828*x+5.3679*y+0.0002*x*x+0.0004*x*y-0.0013*y*y
650
600
550
500
450
400
350
300
250
200
600
500
400
300
200

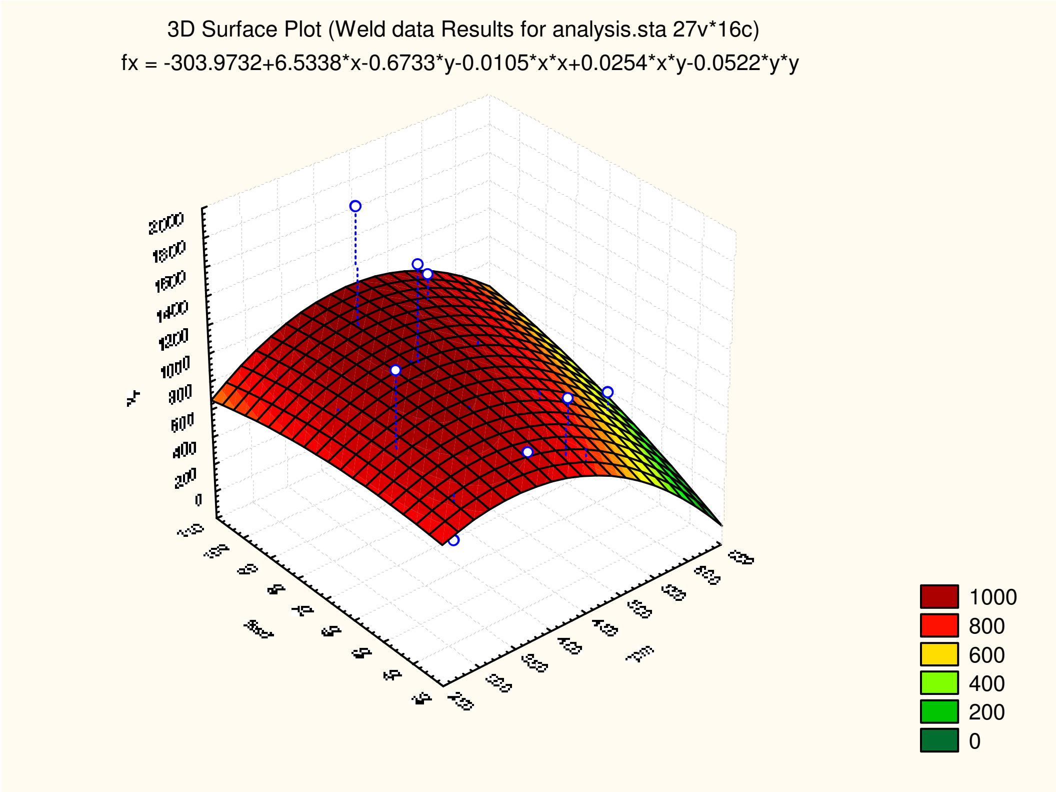

3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
tf = -2.8318+0.0234*x-0.0102*y-2.7026E-5*x*x+5.6614E-6*x*y+0.0001*y*y
4.0
3.5
3.0
2.5
2.0
1.5
1.0
300
350
400
450
500
550
600
650
2.5
2
1.5
1


3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
cf = 3.2008-0.003*x-0.0013*y+3.6426E-6*x*x-2.4125E-6*x*y+1.6202E-5*y*y
2.75
2.70
2.65
2.60
2.55
2.50
2.45
2.40
2.7
2.65
2.6
2.55



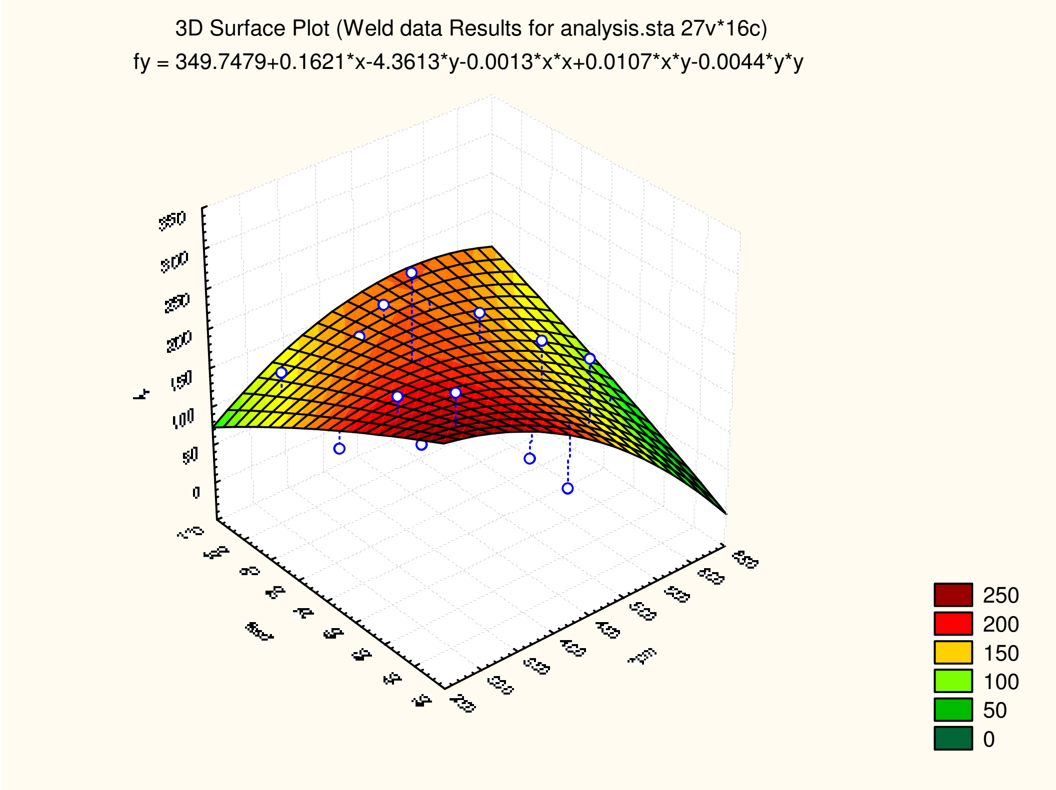
3D Surface Plot (Weld data Results for analysis.sta 27v*16c)

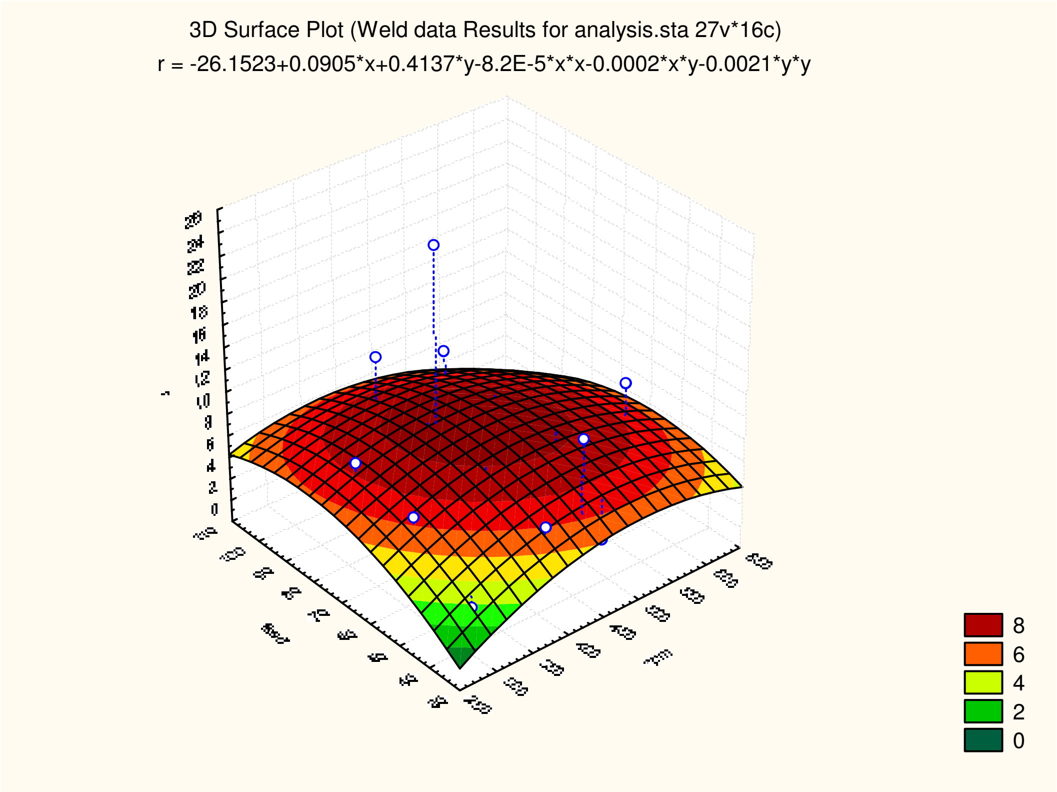
t = -48.7314+2.3824*x+0.6734*y-0.0021*x*x-0.0072*x*y+0.0126*y*y



3D Surface Plot (Weld data Results for analysis.sta 27v*16c)

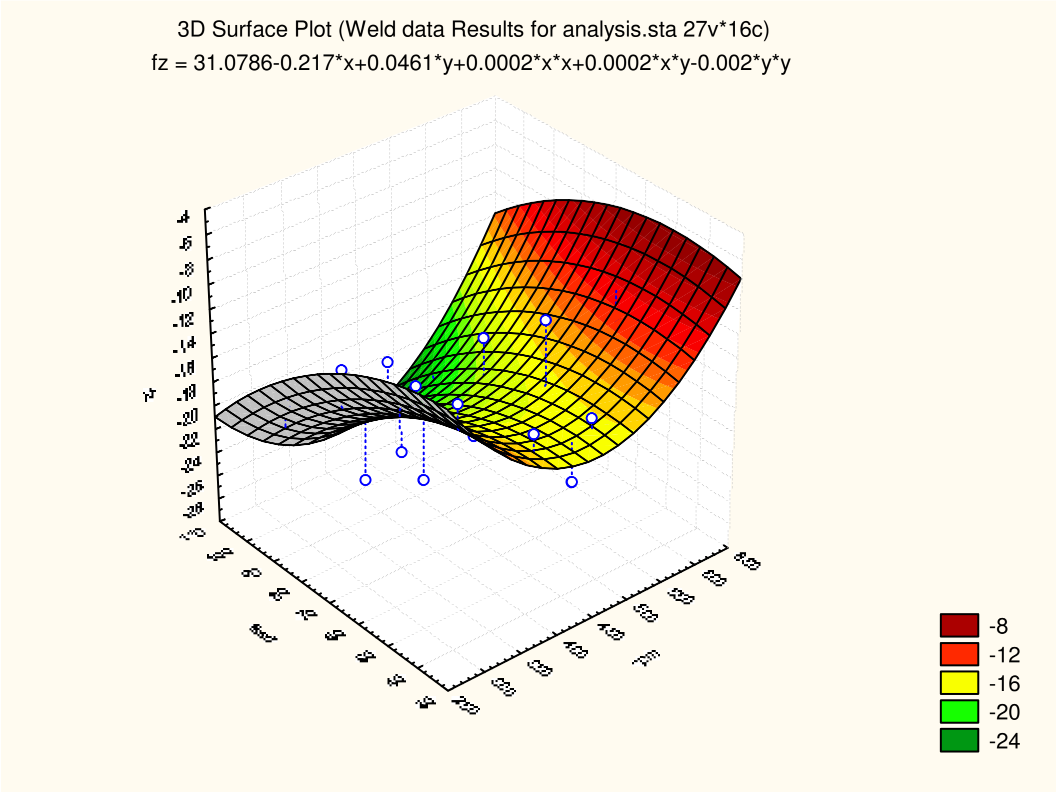
fx = -303.9732+6.5338*x-0.6733*y-0.0105*x*x+0.0254*x*y-0.0522*y*y


3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
fy = 349.7479+0.1621*x-4.3613*y-0.0013*x*x+0.0107*x*y-0.0044*y*y
350
300
250
200
150
100
50
0
fy
250
300
350
400
450
500
550
600
650
250
200
150
100
50
0



3D Surface Plot (Weld data Results for analysis.sta 27v*16c)

r = -26.1523+0.0905*x+0.4137*y-8.2E-5*x*x-0.0002*x*y-0.0021*y*y

3D Surface Plot (Weld data Results for analysis.sta 27v*16c)
fz = 31.0786-0.217*x+0.0461*y+0.0002*x*x+0.0002*x*y-0.002*y*y
-4
-6
-8
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
🟥 -8
🟥 -12
🟨 -16
🟩 -20
🟩 -24