DESIGN OF A MODERN LOW-COST, EXPANDABLE, OPEN-ARCHITECTURE GRINDING MACHINE CONTROL SYSTEM

John James Moruzzi

July 2016

DESIGN OF A MODERN LOW-COST, EXPANDABLE, OPEN-ARCHITECTURE GRINDING MACHINE CONTROL SYSTEM

JOHN MORUZZI

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Master of Philosophy

July 2016

Abstract

An expandable, extendable grinding machine controller has been researched and designed, in order to take advantage of recent improvements in the functionality and affordability of commercial electronic hardware. In additional it would provide for the incorporation of previous research studies and projects aimed at improving the efficiency and effectiveness of the grinding process.

Over the past 20 years there have been continuous improvements in the functional capabilities of grinding machines, their control systems and peripheral process monitoring equipment. Process enhancement technologies such as wheel balancing, touch detection (electrical power and acoustic emission) and in-process gauging may be incorporated into higher-end grinding machines according to specific customer requirements, however this requires significant customisation work by the manufacturer due to the differing features and functionality of equipment from different suppliers. Furthermore the implementation of long-proposed optimization strategies such as adaptive and intelligent control has not progressed significantly beyond specific research programs tied to a particular machine and controller, often using non-industrial (i.e. laboratory) equipment for the monitoring of key process data.

A need was identified to produce a modern, innovative control system architecture for the Jones & Shipman 1300X research grinder at LJMU AMTReL. The controller should be intuitive to configure and operate, and should have flexibility to allow the addition of new equipment and machining features. It would be a significant advance to produce a controller that can more easily integrate and adopt the latest process control and monitoring equipment, and later be expanded to incorporate enhanced production cycles as well as previously explored process optimization techniques and strategies.

The objective of the research was to unify the specification and implementation of key machine tool control features such as hardware configuration parameters, operational parameters, process variables and machining cycles into a rationalized, extendable, object-oriented framework suitable for implementation using current PC hardware, software and design methodologies. The design built on the outcomes of previous studies and developments in the area of optimised machine tool control. Several models of external monitoring equipment were evaluated in terms of functionality and interfacing, and a structured, integrated control system software design and application was produced. Its functionality was demonstrated on a subset of grinding operations and external hardware.

In this thesis the historical developments in controller architectures and technology are discussed, and previous studies into grinding process analysis and optimization are summarized. Various different types of grinding machines and their machining cycles are presented, and the features and functionality of several auxiliary monitoring devices are explained and quantified. The analysis and design of a hierarchical, modular, integrated system structure is then described, and finally the outcomes of the research are reviewed and further development recommendations suggested.

Acknowledgements

The author would like to thank his Director of Studies Professor Michael Morgan for his supervision, guidance and support in the undertaking of this research.

Thanks are also due to Second and Third supervisors Dr David Allanson and Professor Xun Chen for their technical input and encouragement.

It is also appropriate to acknowledge the advice and assistance received from AMTRel senior technician Peter Moran and Ing. Andrea Guidotti of Balance Systems srl.

Further appreciation is also due to the staff of the university's Graduate School and Engineering Research Support Office for their patience and understanding during the completion of this degree.

Finally, and not least, the author would like to express his gratitude to his parents for their support over the last years.

This work is dedicated to the memory of Professor Jim Moruzzi.

Nomenclature

AE	Acoustic Emission
AMTReL	Advanced Manufacturing Research Laboratory
CNC	Computer Numeric Control
Downfeed	Downward movement of the grinding wheel axis
DRO	Digital Readout
Feedrate	Speed of a machining axis in motion
HMI	Human-Machine Interface
Infeed	Inward movement of the grinding wheel axis
IO	Input/Output
LED	Light Emitting Diode
LJMU	Liverpool John Moores University
MPG	Manual Pulse Generator
MSDN	MicroSoft Developer Network
PC500	Original 1300X machine control unit
RPM	Revolutions Per Minute
SAMM	Servo-Assisted Manual Machine
UML	Universal Modelling Language
VB	Visual Basic
VDU	Visual Display Unit
Wheelspeed	Rotational speed of the grinding wheel
Workpiece	Component undergoing machining operations
Workspeed	Rotational speed of the workpiece

Contents

List of figures and tables

1 Introduction

This section summarises the functionality and operation of grinding machines, and the development of their control systems. The enhancement of the process performance using monitoring equipment and parameter optimization is discussed, and the opportunity and need for a new, expandable, low-cost control system is explained. The requirements for an updated control for the target machine are defined, and the objectives, methodology and scope of the investigation are detailed.

1.1 Background

Grinding machines are a type of machine tool that perform specific finishing operations on machined components, with the aim of producing a high quality surface finish and specific profile. Grinding is generally the final operation performed on a component, and can be performed under manual control for individual components, or automatic control for production batches of (identical) components.

Traditional manual machines required the operator to move the axes by rotating control handwheels, and set various switches and limits to control mechanically automated features such as axis reversal and incremental infeed or downfeed. Their effectiveness and performance is governed almost exclusively by the skill and experience of the operator, who will select appropriate increments and speeds depending on dimensions and materials, and assess grinding wheel condition and component quality. Once set these machines can run semi-autonomously, with the operator changing parts and restarting the cycle.

During the 1980s there was widespread introduction of more sophisticated and complex grinding machines using Computer Numeric Control (CNC) - these allowed full electronic control of axis movements via servo drives, control of machine safety, interfacing to other electronic equipment, and most importantly the ability to generate and run complex part-programs that would define the machining operations for a set of components.

A concept developed in the 1990s in conjunction with AMTReL at LJMU was the Servo-Assisted Manual Machine tool control system (SAMM). It was based on the simplicity of a traditional manual grinding machine, but with the ability to program and execute simple automatic grinding cycles due to the incorporation of servomotor control of the machine axes. It was therefore seen as a bridge between the two technology levels, suitable for quickly machining both one-off parts and also low production runs. In the hands of a skilled operator the SAMM machines continue to deliver high productivity and capability despite their relatively simple operator interface and features [1].

In the meantime, Grinding Machine control systems continued to advance and CNC became widely adopted, with various evolutions of proprietary architectures from different control system producers.

This coincided with the adoption and integration of PC and Microsoft Windows technologies in certain aspects of the complete control system, to make use of the power and flexibility this made available to operators. In addition various study and research programs were undertaken in order to explore and define the Open Systems or Open CNC concept, whereby a flexible and comprehensive definition of the aspects of CNC implementation would allow an easier, platform- and manufacturer independent solution with reduced production costs and improved usability. These projects were only partially successful and no final standards were achieved, although many concepts were adopted and advanced by individual participating control suppliers.

Simplified CNC controls have also been recently developed using touch-screen panels as the main HMI interface, rather than a conventional integrated CNC keyboard and VDU screen. These interfaces introduce a more intuitive and simplified "teaching" method of programming and operation, more suited to quick job set-ups and low production runs. They are however still based on a full CNC installation on the machine, despite the cheaper and simpler front-end philosophy.

Coinciding with these developments, auxiliary control and monitoring equipment for different aspects of the grinding process has been developed by various companies. This combination of sensors and electronic hardware could be mounted on the machines and interfaced to the operator panel and control system logic to improve grinding quality, efficiency and overall performance. This equipment is targeted at the issues of grinding wheel balance, wheel-workpiece contact (touch detection), and in-process size gauging, and has made similar improvements in capability and uptake as CNC control.

Finally, over the past 20 years there have been numerous studies and developments in the field of Intelligent and Adaptive Control of grinding, based on the analysis of grinding process parameters such as power and acoustic emission [2], [3]. This allows the dynamic or iterative optimization of the grinding cycle parameters based on real-time conditions on the machine. Many successful experimental implementations of adaptive control have been demonstrated as academic, commercial and official research, but a commercial implementation of a complete solution has so far not been achieved for various reasons.

1.2 Aims and Significance of the Investigation

A low-cost, simple control system has market potential for manufacturers of low-end grinding machines, and machine rebuilders who upgrade and modernise older machines, but do not want the expense or complexity of a full CNC installation. This investigation was therefore intended to implement a simple, affordable control system based on standard commercial hardware and running a software application with a simple, extendable user-friendly interface to enable easy machine-specific customization. This would give real advantages in the development of cost-effective but capable low-end grinding machines.

The PC500 control system was developed in the early 1990s for Jones & Shipman by Goodwin Electronic Controls, for applications on commercial external and internal cylindrical grinding machines. It was developed cooperatively with AMTReL on a prototype Jones & Shipman Universal Grinding Machine, which was then produced by Jones & Shipman as the very successful 1300X commercial grinder. The prototype 1300X has remained at AMTReL for use as a teaching and research machine. The implementation of a modern, affordable, expandable equivalent of this control system was provide the key motivation for this research.

Recent rapid developments in computing speed, memory and overall capability have provided an opportunity for an enhanced and modernised implementation of the SAMM concept, which would allow machine tool manufacturers to offer a low cost control with some of the features normally expected of current full CNC machines. The controls for these typically offer a friendly, intuitive Windows or touch-screen HMI, as well as the integration of process monitoring equipment such as wheel balancing, touch detection and in-process gauging. It was proposed that the use of a PC-based architecture would allow additional features and applications to be more easily incorporated, as well as offering a robust and standardised platform. A modernized, simplified operator interface would also improve usability and productivity.

There was therefore a clear research opportunity for the design and build of a modernized implementation of the SAMM controller concept, based on up-to-date industrial PC technology, commercial auxiliary hardware and current-standard software development and operating platforms. The novelty of the new control philosophy would be to design the system from the outset to give improved interfacing and integration with a variety of modern low-cost process monitoring and control equipment, in order to increase the ease of effectively enhancing and improving grinding operations through both conventional and adaptive control techniques. If the control system can readily configure, monitor and operate this additional equipment through a simple, consistent operator interface and internal software data and parameter structures, it would be significantly easier to offer these features as an affordable and effective enhancement to the machine and the grinding process.

Once these extra features are added to the control system, the various external signals and internal process data are easily available to the control software, and enhanced features such as adaptive control can be introduced to analyse and improve machine performance. In addition useful data values can be logged, displayed and exported, and optimized versions of the grinding cycles developed.

1.3 Research Objectives and Methodology

The work aimed to research and define an innovative and expandable control system design to enhance operator capability, allow for improvements to process performance and advance the machine control systems technology to facilitate future developments.

The main project objectives were:

- To research and document the key functions and features of the 1300X grinder and a selection of auxiliary process control devices.
- To replace or upgrade any faulty or obsolete hardware and re-commission the 1300X.
- To specify and demonstrate a simple, extendable grinding machine controller architecture and system using new and previously researched control system structures, with modern, hardware, software and HMI facilities.
- To expand the controller functionality to include and integrate auxiliary process monitoring and control equipment in a consistent and logical manner.
- To provide for the incorporation of simple Intelligent or Adaptive Grinding features for optimised grinding parameter selection and modification, including those previously investigated and developed in AMTReL

The specification, design and documentation of the software framework was done using the industry standard UML (Universal Modelling Language) approach, which allows clear definition and documentation of the system design. This allowed the formalized analysis and description of existing code available from previous projects, and their combination into an expanded design. The general methodology was to develop a test program comprising working sections of software with initial variations in style and structures, confirm its performance, and then identify and implement opportunities for rationalization.

The initial work comprised the analysis and understanding of the target grinding machine; firstly the operational characteristics and secondly the mechanical, electrical and control features. The technical specification of the original control system also needed to be understood and documented, so that a new design could be produced to replicate the standard functionality and then introduce enhanced capabilities at a later stage.

The machine has a large operator panel with a number of buttons, lamps, keys, displays, switches and handwheels. This was consistent with the manual programming and operation of simple grinding cycles, but was not suitable for the more demanding HMI requirements for grinding research and development work. The existing operator control panel was to be eventually substituted with simplified, modernised hardware upon which would be developed a more comprehensive user HMI using touch-screen technology.

The existing PC500 machine control featured inaccessible, proprietary MS-DOS software running on older generation industrial PC hardware. It was logical to implement the control hardware and software as up-to-date equivalents of those used in the original setup, therefore the new control unit design was based on a compact industrial PC motherboard with function cards, running an embedded Microsoft Windows operating system. Machine axis and hardware control was achieved using commercial servo motion control and signal interfacing cards. Background research was necessary to determine the requirements of the hardware, and the availability and capabilities of the appropriate commercial components.

As regards functional enhancements to the control system, the software was also designed to enable the integration and control of different external process monitoring equipment, such as the Balance Systems VM20 and VM9 ranges. This would allow for convenient and cost-effective extensions of the grinding machine's capabilities. The equipment's features could then be incorporated into the machine's conventional grinding cycles, and its process data used to further advance previous AMTReL research programs such as Adaptive Control and Intelligent Parameter Selection. This equipment would interface to the system using data communications protocols such as RS232, RS485 and Profibus, as well as conventional digital IO signals.

The use of a PC / Windows platform and commercial development tools allowed the easy use of database technology for maintaining process information such as wheel data, optimized cycles and grinding parameters: this process data could be then used to suggest initial grinding conditions to an operator and these can then be improved upon during actual machining. These intelligent techniques have been demonstrated in previous research programs using hybrid CNC-PC-Sensor experimental implementations, however a flexible, integrated research and development platform has not yet been realised. The features and outcomes of these projects were considered in this investigation.

The system solution made use of modern Object-Oriented software design and programming techniques to specify, design and implement the proposed system. Software development was in Microsoft Visual Basic .NET; a robust, modern, portable programming language that provides the control framework with a re-useable and extendable structure of basic and inherited software components with similar but customisable characteristics. Visual Basic was chosen for ease of use and compatibility with available software from previous projects.

1.4 Scope of the Investigation

This investigation was undertaken by the Author solely within the AMTReL facility of the University, and was not part of a wider research project with external participants. The basic functional objective of the new control system was to replicate the conventional Plunge, Traverse and Dress grinding cycles on a cylindrical grinding machine, and then to adapt them to allow more comprehensive parameter selection and optimization using process monitoring equipment. A modular architecture design would allow optimised machining cycles and extra features to be developed separately and accommodated at a later stage.

In addition the control system design needed to take into account the possibility of similar implementations on other types of grinding machines, such as surface and cylindrical grinders. The main differences are the machine architectures and grinding cycles, to be implemented as variations within a common framework. The main characteristics of different grinder types were therefore also investigated during the research.

The existing machine architecture and operations had to be studied, and the functionality implemented in the revised design. This comprised the various machining cycles, operator programming features, machine axis moves and grinding wheel and coolant control. An in-depth knowledge of the cycles, their various parameters and the corresponding effects on the grinding process was required.

Previous research and development projects in the field of Open Architecture / Open Control concepts were studied and reviewed, and any relevant standards, definitions and lessons learned incorporated in the finally implemented solution. This then supported a new control software framework that sought to fully define the operations, data and parameters of typical machines, and similarly provide a consistent control and interfacing strategy for typical grinding process control equipment.

A key initial task was to specify and design a viable and economic package of modern commercially available computer hardware suitable for controlling simple grinding machines (principally cylindrical).

A further requirement was to analyse and document the features and operational characteristics of a range of typical modern grinding process monitoring and control equipment, with a view to specifying and implementing a consistent design philosophy and framework for the enhanced incorporation of such equipment into the overall machine control solution.

The design of the proposed system included the study and understanding of various adaptive control cycles and applications, as previously defined and demonstrated by AMTReL and other researchers. It was desired to provide for the future inclusion of these techniques and features into the overall system structure. It was beyond the scope of the work to implement these enhanced operations and assess any cycle performance improvements, however the philosophy and design of typical adaptive techniques was to be evaluated and clearly defined if practicable.

2 Review of Previous Work

This section highlights historic developments in grinding control systems, and explains the significance of key machining parameters. It then introduces and the concept of the intelligent control to optimise process performance, and describes some relevant Adaptive Control strategies. Key process monitoring quantities, functions, sensors and equipment are then detailed. Significant previous projects in the field of optimised control are reviewed, and the requirements for further progress identified. Finally a selection of significant Open Control Systems research projects are discussed.

2.1 Introduction

The range of modern grinding operations is increasingly wide and accommodates the machining of complex forms, difficult-to-machine materials, ultra-high precision parts and other parts that generally cannot be finished to the required tolerance and integrity in any economically alternative manner.

The grinding process is affected by a large number of variables some of which can be controlled and others which are out of the immediate control of the operator and system. Variations in process inputs such as abrasive type, wheel condition, workpiece hardness and geometry, fluid type, speeds and feeds (control variables) all influence the efficiency of the process. Grinding temperature and grinding deflections also strongly affect grinding performance and geometric and surface texture achieved on the finished component, however these are outputs that are related to and dependent on the grinding cycle control parameters and other process inputs. This complex process can often be optimised by an experienced operator or production engineer, or by an intelligent control system. The aim is ensure consistent quality and speed of production in an environment subject to variations.

However, it has long been recognised that such reliance on operator skill and knowledge is costly and restrictive, and often leads to production shortcomings. This potential failing, together with demands on the process for higher productivity at lower cost, have driven research in the area of process control and optimisation. Development of automatic and adaptive control strategies requires the ability to interface the machine controller with a variety of auxiliary equipment, and to integrate these devices with the operating programs and cycles. This was a key consideration in this investigation.

2.2 Developments in Grinding Control Systems

2.2.1 Conventional Control

Machine tool control evolution started in the 1950s with Numerical Control (NC) of a simple machine tool with electromechanical and display features, and programming via paper tapes. CNC controls enable preset machining cycles (for grinding, milling and turning operations) to be designed, input and stored as a repeatable program that would be performed automatically without the operator explicitly controlling the movements and actions of the machine. It allowed a higher level of production control and optimisation to be attained, instead of the need to rely solely on the operator's experience and intuition with traditional manual machining. It also required the Production Engineer to becoming more involved in the manufacturing process, as he would be responsible for the part program generation based on formalised grinding data and component requirements.

The next stage was Direct Numerical Control (DNC) in the 1960s which introduced a communications link between an external computer and one or more machine tools. The master part programs were therefore downloaded and executed electronically.

Computer Numerical Control (CNC) started to be introduced in the 1970s as the final enhancement to the traditional manually controlled machine. Conventional Machine Tool controls are now based on variations of an industrial computer system with stored software cycles that manage the axis motions and component speeds in accordance to a programmed series of fixed instructions, operator adjustments and external events. Axis control is via Servomotors with encoder feedback, and external equipment interfacing is via Analogue and Digital signalling.

Grinding is a machining process that removes relatively small amounts of workpiece material (of the order of microns) from a semi-finished part, in order to produce an extremely accurate final size and fine surface finish. This is achieved by rotating an abrasive wheel and bringing it into controlled contact with a flat, angled or cylindrical workpiece. The workpiece will typically be also moving relative to the grinding wheel, with a rotational or traversing motion (or occasionally both combined).

The key operating parameters that define the operations and performance of the grinding cycles studied and implemented in this project are as follows:-

Quantity	Symbol	Description
• Infeed (or downfeed) speed	v_f	Wheel movement into part
• Infeed (or downfeed) amount	d_f	Material removal from part
• Wheel surface speed	v_s	Wheel movement across part
• Wheel RPM	n_s	Programmed wheel speed
• Work speed (or table speed)	v_w	Part rotation / translation
• Dwell (or Sparkout) time	t_{so}	Part relaxation / equalization

The grinding process can also be characterised by various derived/computed quantities that can be analysed to understand and improve the process; these quantities are chiefly concerned with the describing the physics of wheel and workpiece surface interaction:-

	Quantity	Symbol	Description
•	Normal Grinding Force	F_n	Force from wheel into part
•	Tangential Grinding Force	F_t	Force from wheel across part
•	Specific Energy	U	Energy transfer into part
•	Material Removal Rate	Q'_w	Specific material removal rate
•	Equivalent chip thickness	h_{eq}	A material removal rate parameter

In a grinding cycle the values of machining parameters such as infeed (or downfeed) speed and increment, traverse speed, wheel speed, work speed and sparkout are programmed to deliver a part of defined size and geometry to a target quality and production rate. On most commercially available grinding machine control systems it is generally not possible to modify the machining parameters during a cycle (or batch), except by overriding or adjusting spindle and axis speeds via an operator override switch (SSO and FRO). The sequence of machining cycles can be interrupted for a wheel dressing cycle at a determined interval, this however takes no account of the true condition of the wheel sharpness. It is therefore possible to damage a part or exceed target quality criteria between parts, and it became apparent to the industry that a more capable system of operation was desired to ensure production and quality targets were consistently achieved.

This situation led to the subsequent research in intelligent and adaptive control optimisation techniques to achieve increased efficiency and quality, rather than relying on operator skill and experience [2].

2.2.2 Intelligent control

"Intelligent control aims to achieve grinding conditions near optimum" [3].

It can be seen from the previously described 'conventional' system how the inability to easily modify parameter values affects the overall efficiency of the system. The process tends to be operated in a conservative manner with lower than optimal specific energy (grinding efficiency), inefficient dressing cycle intervals, and longer cycle times due to slow feedrates and long dwell periods. It follows that key areas of the process can be improved through intelligent control methods, i.e. application of monitoring systems, process modelling, strategy development and implementation. A system that facilitates interactive supervision and control (i.e. modifies machining parameters via a feedback mechanism either iteratively or in real-time) can be described as an Adaptive Control (AC) system.

In order to benefit from previous optimisation activities a fully implemented intelligent control system would have a database of known base cycles for a particular mode of grinding, generated from published data, existing part programs and accrued experience. This library could be used to provide an optimal set of parameters to begin the implementation of a new cycle thus avoiding the need to re-learn the optimal machining/cycle parameters each time the programme/component is changed. This has been described as an Intelligent Grinding Assistant (IGA) [4]. As production of a batch of workpieces proceeded these initial starting parameters could be further refined either as temporary dynamic adjustments, or saved as optimised versions of the part program. In principle the optimised parameters could be saved to a company database so as to allow all machining experience to be available to all IGA equipped machine tools.

A system that provides an optimal estimate of the correct grinding parameters to program for a particular job is often called an Offline Program Generator, where offline indicates separation from the active machine tool. They can generate an initial part program based on the part geometry and materials as well as the machine, grinding wheel and coolant types, and are a modern extension of the traditional grinding handbooks available to the industry and used by operators and process engineers. The derived part programs are loaded by the controller for execution as a grinding cycle. Examples include the Intelligent Grinding Parameter Selector (IGPS) [4] and the Generalized Intelligent Grinding Advisory System (GICAS) [5].

Figure 2.1 shows a combined IGPS / IGA system, which can generate an initial program (reference cycle) from a database, run the cycle and evaluate its performance, adjust parameters to produce an Optimized Cycle, and save this as a new program in the database. (Reproduced with permission: Inderscience Enterprises Limited).

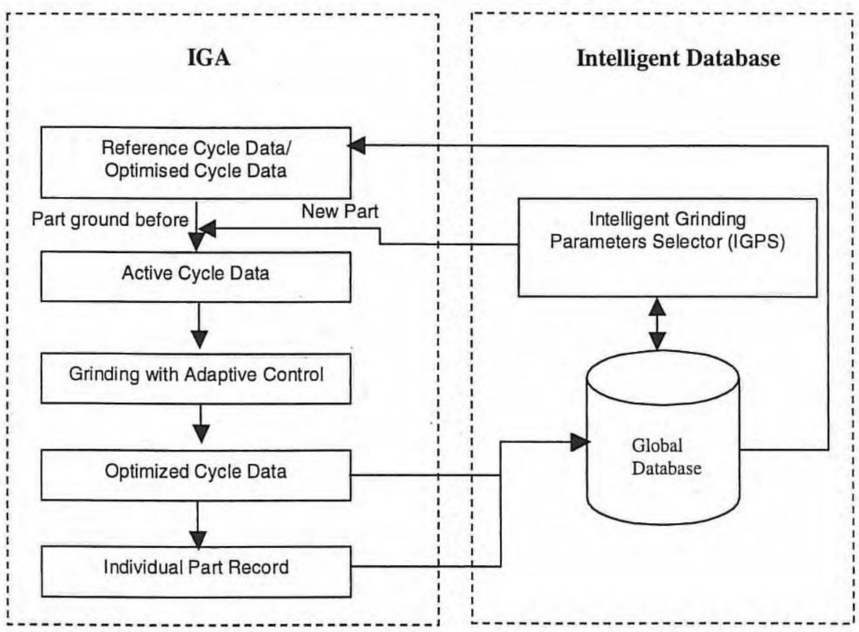


Figure 2.1 LJMU AMTReL IGA / IGPS intelligent grinding system

2.2.3 Adaptive Control

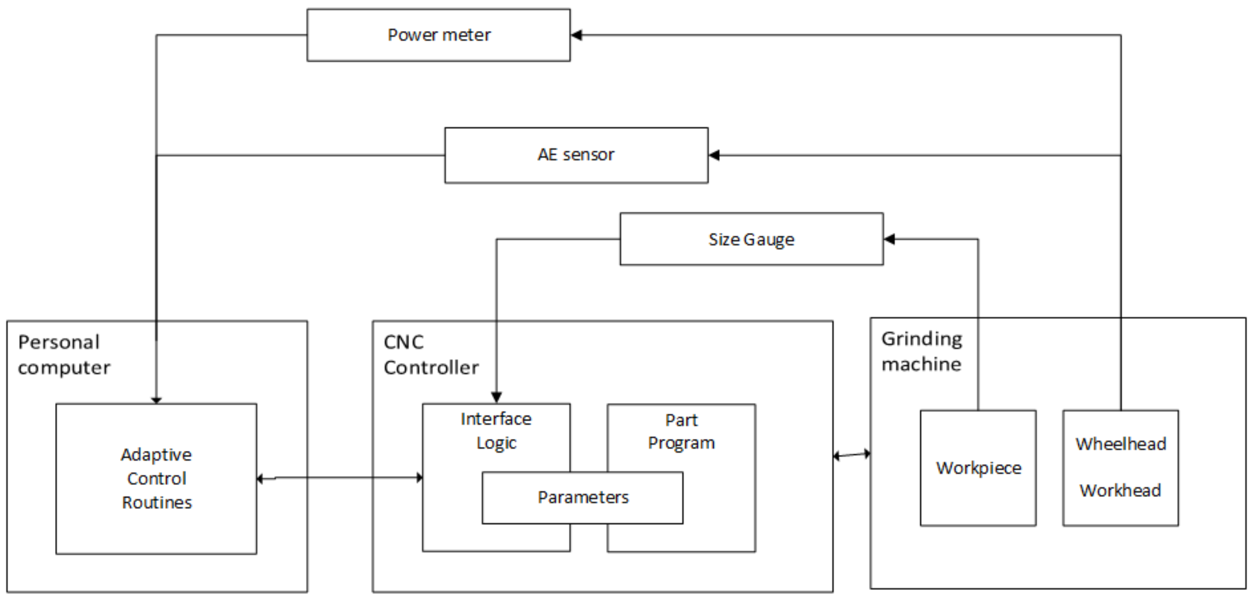


Figure 2.2 Structure of a typical Adaptive Control system

Adaptive Control in context of this work means modifying initial or baseline machining parameters in response to monitored signal values (feedback) in a manner defined by a suitable strategy. This form of Adaptive Control is most useful in batch production, where you may start with conservative grinding parameters and then systematically refine them. Adaptive Control allows compensation for process variations with time (e.g. wheel size and sharpness), and can increase the production rate as sparkout dwells, infeed rates, wheel speeds and work speeds are optimally adjusted up or down. For example, a high rate of wheel wear after dressing will often give an increase or decrease in grinding power [6]. This can be recognised and compensated for by the Adaptive Control system. Figure 2.2 illustrates the elements and functions of a control system featuring in-process monitoring and Adaptive Control [3]. Key physical quantities are measured and various algorithms implemented to quantify the process status, adjustments to the key input parameters are then calculated and fed back to the grinding controller for current or subsequent operations. (Adapted with permission: Elsevier Limited).

Several schemes have been previously devised to implement different forms of adaptive controlled grinding [6], and these were reviewed and considered for possible future incorporation into the new controller design. The strategies most relevant to the project machine's grinding cycles and parameters are presented below.

2.2.3.1 Controlled Force grinding

With Controlled Force grinding, Instead of using a constant infeed, one can vary the infeed rate dynamically as the Normal force is monitored and held at a target value. This compensates for elastic deflection (especially with internal grinding) as well as reducing the air-grinding period, the initial unproductive phase during which the grinding wheel is advancing towards the workpiece surface at a low feedrate [2].

2.2.3.2 Adaptive Feedrate Control

A target grinding power level is selected based on the grinding conditions and parameters. The Infeed rate is then regularly adjusted to optimise material removal rate, and also effects such as wheel self-sharpening.

The target power is not achieved immediately following wheel/workpiece contact, effects of compliance give rise to an approximately first order response. The resulting system behaviour is characterised by a Time Constant τ, which depends largely on the workpiece characteristics and geometry. One can evaluate the Time Constant during the (a) Infeed or (b) Dwell periods by monitoring spindle power and / or acoustic emission. The programmed infeed rate may be modified on the next cycle to a new value determined by the time constant and measured peak grinding power [6].

2.2.3.3 Adaptive Dwell Control

Analysis of the system time constant can also be used to optimise the post-machining dwell period [5]. If the Dwell is too short there is insufficient relaxation of system deflections, and roundness or size errors will result. If the Dwell is too long, the wheel becomes glazed (through polishing) and the cycle time is needlessly increased. Furthermore, the optimal Dwell period depends on wheel sharpness and is therefore variable and needs to be adapted throughout the machining of a batch of workpieces, or between operations on more complex components.

2.2.3.4 Adaptive Multiplunge grinding

Multiplunge grinding involves executing a series of plunge grinds along the length of a workpiece. With a long or flexible workpiece quality problems arise due to barrelling of the workpiece at the centre, where less material is removed due to increased system deflections in this region. Adaptive multiplunge grinding determines the time constant at each section of the workpiece, and adjusts the dwell period accordingly in order to reduce geometric errors and maintain the required surface parameters [3].

2.2.4 System implementation issues

Typical developmental system architectures have previously involved an external PC (or similar) to manage the optimised or adaptive control, connected to a specialised transducer (e.g. power meter, force sensor) with an Analog-to-Digital interface card to read the measured value. There would also be a Digital IO card to connect a number of data lines to an interface port in the CNC controller. Any other process monitoring equipment (such as gap elimination or in-process gauging) would be connected directly to the CNC and integrated with its grinding cycles. The analytical software on the PC would be a stand-alone custom application.

It can thus be seen that these arrangements have been applicable only for experimental and developmental work in a laboratory environment, but are not suitable for prolonged use in a production situation. For further useful progress to be made it can be seen that the sensing equipment, the calculations and the communications would need to be more formally and effectively combined into the machine control architecture as an integrated system. The optimised control logic would reside in a separate software partition, and would combine data from the machine tool and the sensing equipment to adjust the process parameters. This research aims to further the achievement of this goal.

2.3 Process Monitoring and Control in Grinding

Control of the process requires knowledge of the process performance with respect to target criteria and also the response of the system to a change in process input. The monitoring of the process achieves the objective of informing the operator/observer of the process performance. This information can be relayed to the operator in real-time whereby the monitored parameter is displayed by e.g. a screen or dial. Alternatively, the monitored data may be evaluated at the end of a cycle, e.g. where an Acoustic Emission (AE) profile is displayed after traversing a workpiece feature. In each case the operator is then able to assess the performance of the process in respect of the target criteria.

2.3.1 Process Monitoring: Sensors and data

A comprehensive review entitled 'Process Monitoring in Grinding' was presented by Tönshoff, Friemuth and Becker in the 2002 Annals of the CIRP [7]. The main motivation of this work was to review and critically discuss the state-of-the-art in grinding process control and monitoring. The key sensors for measuring process quantities were identified as: Force sensors, Power sensors, AE sensors and Temperature sensors. The principal output quantities measured were: work geometry, surface finish, residual stress level, workpiece temperature (burn), grinding power, force and AE. Surface finish and residual stress were generally measured using offline instruments. The authors reported on the dependencies of the process quantities on the machining parameters and how control of the machining parameters improved process efficiency. The quantities of most practical interest are as follows:

2.3.1.1 Force

Force sensors are fundamentally based on displacement. The displacement measurement can be realised by a variety of sensors, the most common being strain gauges and piezoelectric devices. Strain gauges are impractical in most machining operations and are constrained to the research laboratory, whereas Piezoelectric devices are commonly used in industry. In surface and cylindrical grinding operations a distinction is made between the two forces: Normal (F_n) and Tangential (F_t). The Tangential force is largely responsible for material removal however it is difficult to measure directly without physical intrusion and it is more practical to work with measured grinding power. The grinding power relationship is given as:

$P = F_t . v_s$ where v_s is wheel speed. (Eq. 1)

One implementation of adaptive controlled grinding is based on the Normal force using a constant force infeed method and optimised sparkout /retraction [2].

2.3.1.2 Power

Power monitoring and measurement is usually undertaken via voltage, current, or phase shift analysis. If the wheel spindle is driven by a Frequency Inverter drive it will increase electrical power during grinding in order to maintain constant spindle speed. There are various commercial power monitoring systems available for machine tools.

Power monitoring can be used for dynamic monitoring and Gap Elimination (reduction of air-grinding time) but there are limitations in its responsiveness due to time constants (e.g. wheel/spindle inertia) and switching delays. Furthermore the levels of grinding power can be relatively low compared to the overall spindle power (particularly for dressing), so the sensitivity will be compromised. Also the power consumed can be affected by variations in the moment of friction (i.e. lubricating conditions). Nevertheless spindle power is a useful and convenient monitoring quantity for process control and optimisation.

2.3.1.3 Acoustic Emission

Acoustic Emission (AE) monitoring is based on the detection and processing of the elastic energy waves that are released from the material structures during in the grinding process. The energy waves typically range in frequency up to 1 MHz, and propagate through the solid materials of the grinding wheel, workpiece, dresser, machine structure and also the coolant stream [3].

AE monitoring is suitable for very rapid and sensitive signal processing and is therefore highly applicable for contact and collision detection. It is also useful for profile monitoring, for example during the dressing of CBN grinding wheels where minimal abrasive material removal is required. The disadvantage of AE monitoring is that there are many sources of noise on the machine (motors, pumps, bearings, coolant, etc.) and these need to be carefully filtered or tuned-out so that only the appropriate frequencies are observed. The location and specification of the AE sensors is also critical to good performance [7].

2.3.1.4 Acceleration

Acceleration of machine components can be monitored as an indication of vibrations present on the machine during grinding. The vibration frequencies are in the kHz domain and extend to the start of AE ranges. Vibrations are often classified as forced vibrations and self-excited vibrations [3], and are typically due to two conditions – unbalance and chatter.

Unbalance is caused by rotating elements on the machine (typically the grinding wheel, spindle, pumps and pulleys) being out of balance: the unbalance increases with the rotation speed and mass of the rotating object. Unbalance will result in poor workpiece surface finish and increased spindle bearing and structure wear. The measured unbalance can be expressed as a velocity (mm/s) or more commonly a displacement (microns). Viewing unbalance as a displacement gives an indication of the effect on surface finish of the workpiece.

Unbalance of rotating machine components is normally corrected during machine assembly and commissioning, by adding or removing material. Unbalance of the grinding wheel assembly can be done statically (off the machine) by moving external compensation weights, or dynamically (on the machine) by equipment to move mechanical weights (or fluids) in an auxiliary balancing unit fitted to the spindle.

Chatter is a phenomenon caused by the interaction of the opposing movements of the workpiece and the grinding wheel, particularly in Cylindrical or Centreless grinding. It manifests itself as uneven workpiece surface finish, and is typically reduced by adjusting process parameters such as the relationship between workpiece and wheel speeds.

2.3.1.5 Temperature

Temperature measurement in grinding is rarely implemented in practice although it is frequently used in the laboratory to establish the veracity of analytical models [2]. In some instances however, temperature measurement can be undertaken using temperature probes (thermocouples) inserted axially in the face of the grinding wheel. Temperature monitoring can predict burn damage and identify wheel wear, but is impractical in a machining environment. It is more common to relate surface temperature to more readily measured process outputs, such a tangential force, grinding wheel speed and power consumed.

2.3.1.6 Size and roundness

Final size is the most critical factor in the quality of the finished workpiece. Size errors are generally due to differences between the programmed wheel infeed and the actual wheel infeed, and therefore the amount of material removal. The principal causes of these differences are deflections in the machine and workpiece due to mechanical forces when the wheel engages, and a reduction in wheelsize due to wear during production. These effects will both produce an oversized part, however wear of the wheel dressing tool will produce an undersized part if the wheel size compensation is incorrectly adjusted [3].

Gauging is used as an error compensation method to accurately determine when finished size has been reached. It will remove inaccuracies due to wheel wear and system deflection, and can optimise process times as well as quality. The gauge is programmed to indicate various size thresholds during machining of the part, and the control system or operator will modify the machine’s behaviour accordingly.

Roundness in cylindrical grinding is largely affected by the interaction of wheelspeed v_w and workpiece speed v_s , and surface effects on the finished part such as lobing (waviness) can be reduced by careful matching of these parameters. Roundness is normally measured offline on sample pieces in a laboratory using equipment from e.g. Taylor-Hobson, however some new gauging systems such as the Balance Systems VM20 can directly perform roundness analysis on the machine.

2.3.2 Process Monitoring: Applications and equipment

There are several commercial lines of grinding process improvement equipment available, with the following main functions and features:

- **Wheel Balancing** Semi- and Fully-automatic.

  Method: weights movement or fluid transfer.
- **Touch detection** Acoustic Emission and Power monitoring

  Also Force and Strain.
- **Gauging** Diameter (size control) or

  Shoulder / Flag (position measurement)
- **Probing** Part location / position measurement

The different manufacturers may offer solutions for more than one function, and also different levels of price/performance for a given function depending on the user's requirements. The main suppliers are Balance Systems, Dittel, Marposs, Movomatic, Renishaw and Schmitt.

Digital input and output connections are required to command resets of equipment, selection of cycles / channels, indication of reaching of thresholds/limits. There is a requirement for the logic in the machine tool to manage these signals and operations. In addition there may be more sophisticated communications using serial, parallel or bus communications to allow the reading and writing of operating parameters, system data etc.

The manufacturers of auxiliary process monitoring equipment each have their own control and data definitions and protocols, as well as operator interfaces, and the wide variety of these increases the cost and complexity of combining the various components into an effective control system. In addition there is the issue of running any custom process control algorithms on a PC subsystem and synchronising it with the machine control and auxiliary equipment.

2.4 Optimised Grinding Control at AMTReL

During the 1990s a series of complementary research programs was undertaken at the John Moores University AMTReL facility, in order to explore, expand and implement adaptive control applications for a production environment. These works built on each other successively until finally an adaptive control software solution was fully integrated with the cycle management of a Fanuc CNC control on a Jones & Shipman grinding machine [8].

These implementations all relied on monitoring and managing the grinding power, which is a convenient and effective measurement of the Normal grinding force and hence Specific energy. Grinding power can also give a clear indication of the wheel condition and hence the need for dressing operations to restore efficiency. Grinding power offered a simple way to monitor the process. Installation of a power monitor requires no modification to machine hardware and is relatively inexpensive, however the measurement is more distant from the cutting process than a typical force measurement.

2.4.1 System Time Constant

During grinding there is a positional lag between the programmed infeed and the actual material removal, due to deflections in the machine structure and workpiece bending. There will be compression in the system at the start of infeed, and relaxation at the end of infeed (or the start of Dwell). During Dwell (Sparkout) there will continue to be material removal (and hence consumption of grinding power), this will decay until the relaxation has completed and the part should then notionally be at size. This mechanical behaviour can be quantified by a System Time Constant (T).

The time constant can be determined by integrating the power signal during either the infeed or dwell phase of grinding. Allanson and Rowe [6] determined that the most accurate way to determine the time constant was to use a weighted least mean squares estimate over the duration of the dwell period. This reduced the effect of noise on the power signal and other disturbances. The optimised dwell time could then be set at a number x of time constants.

2.4.2 Thermal model

A Thermal Model was developed [4] to calculate the temperature effects of a grinding operation, and predict thermal damage to the workpiece (burn). The model focuses on the wheel, workpiece and fluid interaction, and the thermal and material properties of these components. From these quantities and a measurement of the spindle power used, a Burn algorithm calculates:

- Max allowed temperature
- Predicted Temperature
- Cycle Specific Energy (grinding power / material removal rate)

If the Specific energy is too high and a potentially workpiece damaging burn condition is approached, the wheel infeed rate can be reduced in order to lower grinding forces. At the end of the machining cycle a wheel dressing cycle can then be proposed, in order to restore wheel sharpness and improve grinding efficiency.

2.4.3 Control and monitoring synchronisation

The analysis and modification of the machining quantities involved in adaptive control (AC) must be synchronised with the grinding cycle phases as these are executed by the CNC. For example, the start of the infeed, fine feed, dwell and retraction phases needs to be signalled to the AC software so that signal measurement and analysis can be performed. Similarly the AC system must be able to read and update the programmed cycle parameters used by the CNC. This synchronisation requires the bidirectional communication of cycle states, parameter and system variable values, as well as control commands. This functionality can be best achieved with a standardised software design structure and ideally a fully integrated control system.

2.4.4 Previous Adaptive Control developments

Several implementations of Adaptive grinding control were undertaken at JMU AMTRel from the mid 1990's, each building on the advances of previous research programs.

Kelly and Rowe [9] initially developed a 'pecking' cycle on a Centreless grinding machine that determined the deflection response of a thin cylindrical cylinder liner. Later research was however primarily focussed on cylindrical plunge grinding,

Allanson [5] implemented an Adaptive Control system with an external PC interfaced to the Jones & Shipman Series 10 grinder and Allen Bradley 8200 Control. The PC featured power monitoring and the control incorporated in-process diameter gauging.

Thomas [11] further developed the system for adaptive plunge grinding. New cycles that modified the target infeed position and implemented adaptive dwell were implemented. In addition a new gauging cycle to simplify setup and reduce infeed position errors was added.

Chen [8] developed a Generic Intelligent Control System (GICS) hosted on the same PC hardware as the Jones & Shipman 1300x machine controller. The power monitoring and logging was performed by the CNC partition, which stored it to a common data area. The GICS analysed the grinding and power data post-process and provided the CNC with updated infeed and dwell parameters for the next cycle. The system was able to implement several adaptive strategies developed previously for external plunge grinding, and validate and assess them when applied to internal plunge grinding. This implementation used Object Orientated techniques to develop a modular and extendable design for the software.

Statham [8] designed and implemented an Open CNC Interface (OCI) for intelligent grinding control, based on previous studies for Open Architecture in the CNC field. A 4-layer Object-Oriented model was developed that allowed custom user applications (e.g. Adaptive Control) to interface to a variety of CNC systems with maximum portability and minimum integration effort. This is relevant to modern control systems which offer an embedded PC partition with standard operating systems (e.g. Windows 2000, Windows XP) as well as a proprietary API (Application Programmer's Interface). The legacy development environments and languages previously used (Borland Delphi, Microsoft C++) have been largely superseded by the .Net tools chosen for this research, but much of the software can be adapted and reused.

A software application and library was written to be hosted on a Jones & Shipman Supromat cylindrical grinder with a Fanuc 180 IB CNC featuring a Windows 95 PC partition. The OCI model was validated, and shown to be able to handle large scale high speed data transfer between the CNC and the user application. It was then ported to a newer Fanuc 210i CNC platform with only the introduction of a revised software dynamic link library (DLL) file. To port the application to a completely new control type (e.g. Siemens) would require the generation of another library file to interface to the Siemens API functions.

The Intelligent Grinding Assistant or IGA [4] was a European industry-funded collaborative program, comprising Liverpool JMU AMTReL, Tacchella SPA (machine tool manufacturer) and Balance Systems Srl (process control equipment supplier). It was based on an external PC that linked to the grinding machine CNC (via Ethernet) and the process control unit (via Profibus), and used Power, Acoustic Emission and Gauging signals to optimise and modify cycle parameters (i.e. infeed rate and dwell) iteratively during batch machining. It also updated the stored part programs with optimised values.

The IGPS (Intelligent Grinding Machine Selector) was a software program able to generate and modify machining part-programs using a combination of Rule-Based Reasoning (RBR) and Knowledge-Based Reasoning (KBR). It referred to a large database of part programs, grinding cycles, and machine, material and coolant properties and used various models to select machining and dressing strategies.

2.4.5 Future Adaptive Control developments

It can be seen that significant progress has been made historically towards intelligently optimising grinding operations, but to progress it further requires a tighter integration of auxiliary process control hardware and algorithms with the main machine tool and cycle control system. A revised and more flexible control architecture would therefore need to be analysed and implemented to provide a practical solution, and this was one of the key aims of this investigation.

2.5 Control System Developments

In order to propose a new control system structural design, a study of previous attempts to produce a standardised yet flexible specification and architecture was undertaken. Most of these were collaborative projects undertaken by various manufacturers, operators and academic institutions, and had varying degrees of success and commercial implementation. From the various project outcomes and publications the fundamental structures and actions identified and implemented were assessed and used to guide and validate the project work.

2.5.1 Open Control Systems developments

The Open Control Systems concept for machine tool controls originated in the 1990s in an attempt to define and harmonise the differing proprietary technologies that had evolved with CNC manufacturers. There was a clear need within industry to evolve traditional control designs into modern, well defined architectures using a more generic structure. This would allow easier definition and implementation of standardised components, and therefore system expansion and integration.

Various research programs were undertaken involving Industry and Academic Institutions, principally OSACA (Europe), OMAC (North America) and OSEC (Japan). Later follow-on programs in Europe included OSACA II and OCEAN. In most cases the basic outcomes and definitions were published but generally no recognised industry standards were adopted. Further developments were generally kept "in-house" by manufacturers (OSAI, NUM, Fagor, Siemens, Homag, Fidia), or offered by academic participants as Open-Source Software (OROCOS).

The fundamental features of an Open Systems Control were defined as

- Commercial or Industry standard hardware (able to be upgraded)
- Vendor-neutral architectures and application modules
- Modular software structures with well-defined software interfaces
- Layered approach to structure hides hardware-specific features
- Flexible and reconfigurable, adaptable to new technologies and processes

2.5.2 Open Control System projects

OSACA / OSACA II (**O**pen **S**ystems **A**rchitecture for **C**ontrols in **A**utomation Systems, 1992 / 1997)

The objective of OSACA (Project Ref: ESPRIT 3 - 6379) was the definition of a hardware-independent reference architecture for numerical machine tool equipment including the functionalities of numerical controls (NC), robot controls (RC), programmable logic controls (PLC), and cell controls (CC). It formed a common base for all kinds of automation systems and was made open for the adoption of new functionalities and computing equipment [12]. Object oriented principles were adopted for the specification of the reference architecture and definition of device interfaces. The OSACA communication system defined the internal communication within the control, as well as with superior (external) systems and with subordinate (internal) subsystems (such as sensors and actuators).

OMAC **(O**pen **M**odular **A**rchitecture **C**ontroller, 1994**)**

In the US automotive industry Chrysler, Ford Motor and General Motors recognized that they faced similar challenges to improve their manufacturing systems' efficiency. The proliferation of proprietary control systems from different vendors in a single facility made the maintenance and operation of these systems difficult to manage for plant personnel. The project specified the requirements for future open and modular controllers and systems in order to:

- Provide a common, standard look and feel user interface to all control systems
- Reduce control system development and integration time
- Allow incremental upgrades of control systems with technology improvements

OMAC did not define a fixed reference architecture, but specified a set of modules to allow the build-up of different types of controllers. In the current version OMAC, 1999] of the API 14 complex modules exist. The modules and their interfaces were specified in IDL (Interface Definition Language)

OSEC / OSEC II (**O**pen **S**ystem **E**nvironment for **C**ontrollers, 1994-1996)

In Japan eighteen major companies (including Mitsubishi, [13]) and a public research institute established the Open System Environment (OSE) consortium, to develop a Japanese open controller for factory automation (FA) machines. Later many similar efforts dealing with the factory automation were merged together under the umbrella of JOP (Japan FA Open Systems Promotion Group). OSEC was focused only on the PC / Windows platform and it did not allow distributed control.

OCI **(O**pen **C**NC **I**nterface, 1999)

In the UK Liverpool John Moores University and Jones & Shipman [8] devised a 4-layer Open CNC Interface (OCI) framework, with the aim of aiding the integration of custom PC application programs with different CNC interfaces such as Siemens or Fanuc. An object-oriented design enabled the structuring of software classes (Base and Derived) to provide a standardised interface to different (closed) controls using abstracted levels of hardware-specific functionality.

OROCOS (**O**pen **RO**bot **CO**ntrol **S**oftware, 2001)

This program was driven by the University of Leuven in Belgium, together with other European participants (Laas, KTH, LVD, Flanders Mechatronics). Originally intended to be a Free Software project for robot control using an "open" control framework, it was later extended into the general machine tool control field in conjunction with the complementary OCEAN program. A Real Time Toolkit (RTT) for Linux was produced, defining the components and interfaces necessary for system construction. A further evolution was a development toolkit called ROCK (Robot Construction Kit).

OCEAN (**O**pen **C**ontroller **E**nabled by an **A**dvanced real-time **N**etwork, 2002)

Funded by the European Commission Information Society Technologies (Project ref: FP5 IST-2001-37394), participants were manufacturers Homag, Fagor, OSAI, and Fidia, with the universities of Stuttgart and Aachen. It defined a new standard for the development of software modules based on a common communications protocol with and between modules, and the intellectual property within each module was protected. A Linux component library known as the DCRF (Distributed Control system Real-time Framework) was produced [14] and published as Open Source, as well as an extended component-based open control reference architecture with standardised interfaces for motion control components of machine tools.

2.5.3 Industry Adoption of Open Systems controllers

It was determined that although several projects were undertaken to achieve essentially the same objective, no recognised, definitive official standards were actually produced or published. Instead it can be concluded that the defined component structures and activities were relevant and applicable to most of the equipment and applications under consideration, and could therefore be refined and adopted for forthcoming products and projects of the contributors.

Research of the technical publications and advertising material of some of the European industrial project participants indicates that several have adopted the derived architectures and philosophies into their current (and therefore future) equipment ranges. Some key examples are given below:

OSAI (Italy)

This CNC supplier has developed the OPEN*control* range of equipment as a result of previous research programs: it is based on industrial PC hardware and Microsoft software, with functionality distributed among its component elements using a flexible suite of connections protocols, including EtherCAT, CANopen, Profibus, Modbus as well as the company's proprietary fieldbus "OS-Wire". The controls are configurable for metal, woodworking, plastics, glass and stone machining applications.

Fidia (Italy)

This machine tool OEM developed the "C Class" custom control line for their high-speed milling machines and for sale to external customers. The architecture manages the user interface with an Intel PC processor, and the real time control (axis and tool path) with a separate RISC processor.

NUM (France)

This CNC company's Flexium range is an Open Architecture system to provide a platform for a wide variety of applications. It offers the flexibility and scalability, with any of up to 32 connected devices to be configured as either an axis or a spindle.

2.5.4 New architecture requirements

This research project was intended to produce a revised grinding machine control architecture that would make use of the outcomes of earlier studies and projects mentioned in sections 2.1 and 2.5. These have sought to rationalise controller design philosophies and also use complementary technology and analysis strategies to optimise the grinding process. The new design was intended exploit the similarities highlighted between internal (OROCOS, OCEAN) and external (OCI, IGA) system elements in order to facilitate the enhanced expansion capabilities of the control system. The established Open Systems philosophy of layered functionality and hardware independence would facilitate this.

3 Grinding Machines and Control Systems

This section explains the principle features and applications of the most common grinding machine types, together with an introduction to some key grinding operations and machining cycles. The fundamentals of machine tool control and cycle programming are introduced, and their past and future evolution is discussed. Finally a description of the target machine for this research is presented, with a summary of its physical features and operational characteristics.

3.1 Grinding machine types

Grinding machines can generally be grouped into three main categories, characterised by the geometry of the machine or the parts they operate on. The greater part of this study will also be applicable directly across to these machines, with appropriate modifications to machine cycle and control parameters.

3.1.1 Cylindrical and Universal Grinders

External grinding machines are characterised by the provision of one or more external grinding wheels on a single spindle. They use relatively large (20cm - 60cm) and heavy grinding wheels running at speeds of 1000 – 2500 rpm, although speeds will generally be higher for grinding with CBN wheels (typically 3000 – 10,000 rpm).

The workpieces are cylindrical in form, and are held between centres and rotated or held stationary by a workhead unit. The grinding operations are performed on the outer surface of the part, and occasionally on the end surfaces (face grinding).

Internal grinding machines grind the internal surfaces of cylindrical parts, and therefore the grinding wheels must be small in diameter relative to the bore being machined. Wheels are generally mounted on an extended, removable shaft (or quill) to allow them to reach inside the part. To attain acceptable surface speeds with small diameter wheels, high speed spindles (often with air bearings) are used. Spindle speeds of 12,000 - 20,000 rpm are typical for Internal grinding operations.

A Universal grinding machine has the provision for External and Internal grinding, using an additional internal spindle mounted on the rear of the wheelhead assembly. To perform Internal grinding, the wheelhead is unclamped and rotated by 180°, and other appropriate machine adjustments are made (e.g. to spindle speed).

External and Internal grinding operations are characterised by wheel infeed movements (X axis, into the part) and crossfeed movements (Z axis, along the part). If the workpiece rotation has position control (i.e. it is not simply free running) it is known as the C axis. A typical universal grinder and its features are shown in Figure 3.1 below.

Z axis

Horizontal (traverse feed)
Positioning of workpiece for grinding

X axis

Tangential (wheelfeed / infeed)
Application of wheel for grinding

C axis

Workpiece rotation (workhead)
Part fixed between headstock and tailstock

Picture : Blue Diamond Machine Tools

Figure 3.1 Cylindrical (Universal) grinding machine

3.1.2 Surface Grinders

Surface grinders are used to grind flat surfaces by clamping them to a horizontal table. The grinding wheel moves vertically (downfeed axis) onto the part and the table moves the part across the wheel in a series of regular longitudinal and incremental traverse moves. The table axis moves the part from left to right with a crossfeed rate, and the saddle axis moves the part from the front of the machine to the back with a set increment. A typical surface grinder and its features are shown in Figure 3.2 below.

X axis

Horizontal (table)
Oscillation of workpiece for grinding

Y axis

Vertical ("downfeed")
Application of wheel for grinding

Z axis

Workpiece incremental translation (saddle)
Part clamped to table (e.g. magnetic chuck)

Picture : Blue Diamond Machine Tools

Figure 3.2 Surface grinding machine

3.1.3 Rotary Grinders

A rotary grinder features a rotating table to which the part or parts are attached. The table rotates while the vertical or horizontal surfaces of the part are ground by the wheel spindle as it moves vertically downwards. A Universal rotary grinder has a controlled, swivelling wheelhead that can grind horizontal or vertical surfaces of the part. An example of a rotary surface grinder is shown in Figure 3.3 below.

Y axis

Vertical ("downfeed")
Application of wheel for grinding

X axis

Horizontal (table)
Crossfeed of wheelhead for grinding

B axis

Rotary (table)
Rotation of workpieces for grinding

Picture : Sterling Machinery Exchange

Figure 3.3 Rotary surface grinding machine

3.1.4 Centreless Grinders

A Centreless grinding machine is typically used for the production of cylindrical parts that are not physically clamped in the machine, but are supported on a support blade and are rotated against the grinding wheel by a rubber bonded control wheel. The parts are often introduced on one side of the grinding wheel and traverse across the wheel during grinding, before being ejected from the machine and collected (through-feed grinding). Figure 3.4 below shows a typical centreless grinder.

Grinding wheel

Surface speed

X axis

Wheel Infeed against workpiece

Control wheel

Workpiece rotated against wheel

Picture : AJ Baker Grinding Ltd

Figure 3.4 Centreless grinding machine

The Centreless grinding machine can also be configured for plunge grinding operations whereby the part is ground in position on the work blade with no transverse motion. In this configuration a Centreless machine exhibits similar characteristics as a machine grinding between centres with regard to grinding cycle behaviour and control.

The grinding wheels for Centreless grinding are generally large and heavy, and of conventional materials. Spindle speeds are relatively low, and may be altered depending on wheel diameter. The process is defined by infeed of the wheel (X axis), and the relative speeds of the grinding wheel and control wheel.

In addition mechanical adjustments of the machine can be made, in order to vary the vertical position of the workpiece (on the workrest) and the tilt angle between the grinding wheel axis and the control wheel axis.

3.2 Grinding operations and cycle programming

A typical cylindrical grinding cycle will involve the rotation of the wheel at a set speed, rotation or translation of the workpiece, infeed of the wheel to the workpiece at various rates and to prescribed positions, and a final phase of zero infeed for either a dwell time or a number of passes in order to improve the evenness or roundness of the part.

In addition the wheel spindle, workpiece and coolant pump will be turned on and off at appropriate times, either manually by the operator pressing buttons, or automatically by the machine control executing instructions.

Following a number of grinding operations it will be necessary to dress the grinding wheel, this restores a sharp, even surface to the wheel and is done by passing the wheel across a hard cutting tool (typically a diamond point) a number of times with a particular depth of cut and rate of advance (feed) across the wheel face.

Most grinding operations will consist of three phases: Roughing, Cutting and Finishing. These correspond with varying the speed and force of wheel / workpiece contact in order to optimize the balance between speed of material removal and accuracy of finish and size.

A Sparkout or Dwell phase period is usually programmed at the end of the finishing phase when the part is notionally at size. The dwell will allow the system to relax and mechanical deflections to recover, as well as allowing the workpiece to rotate sufficiently to give satisfactory roundness. Note that an excessive dwell period without cutting can cause detrimental glazing of the grinding wheel. Reduced cutting effectiveness will increase the normal force generated and give rise to larger deflections, a need for longer dwell times, increased likelihood of thermal damage and reduced process efficiency.

Wheel and workpiece speeds should be chosen to ensure the elimination of chatter and burn during grinding. Burn occurs when too much grinding energy is input into a section of the part. Chatter can occur when cyclic lobes form on the part surface, often due to harmonic frequency generation due to the combining wheel and workpiece surface speeds.

Cycle efficiency can be improved by interfacing touch detection and gauging equipment, to optimize infeed speeds and reduce finishing times at the end of the cycle. Gap elimination is the process by which the wheel can approach the workpiece at rapid speed with the feedrate changing to coarse feed as soon as initial contact with the workpiece is detected, normally using acoustic emission or spindle power sensing. Size control uses in-process gauging to override the programmed feedrate transition points by monitoring actual part size.

3.2.1 Manual and Automatic operation

Manual grinding operations require the operator to explicitly execute any axis moves to the required positions during machining. This usually means turning one of the machine handwheels to move an axis either continuously or by a set amount. The operator will normally be observing the machining movements and positions, often using an axis position display unit known as a DRO (Digital Read Out). In manual grinding the operator is able to adjust some process parameters directly (e.g. workspeed and feedrates), and can also activate some automatic features (e.g. power traverse between points).

Automatic grinding operations require the cycle parameters to be entered and stored beforehand, usually in a special "Program" or MDI (Manual Data Entry) mode: once this is done and "Automatic" mode is selected then the Cycle Start button is pressed and the cycle executes and runs to completion. If the Cycle Stop button is pressed then the cycle is interrupted and will execute final instructions and retract moves before ending program execution.

3.2.2 Key Grinding Cycles for Cylindrical Grinding

3.2.2.1 Plunge grinding

Plunge grinding is a machining operation on one part of a circular workpiece: the main phases being Roughing, Stock removal, Finishing and Sparkout. Each phase (except sparkout) has a programmed axis feed rate and increment (or distance). Because the workpiece will have a finished circular diameter, the distance parameters (sizes) are also programmed as diameters. The fundamental concept is stock reduction from a nominal base to a final target diameter, relative to the centerline of the part on the X axis. Wheelspeeds, workspeeds and infeed rates are chosen to maximize stock removal rates and minimize thermal damage to the workpiece. The Sparkout phase performs final material removal without infeed as the machine and workpiece relax, this ensures a normalized part size and profile. Some research has been undertaken to investigate the application of a very fine feed rate instead of the normal dwell period. Feedrates of 0.1 micron per second have been found to be beneficial when combined with in-process gauging and in-process time constant estimation [6].

Table 3.1 below lists the phases and programming quantities of a basic plunge cycle.

Phase	Machine action and cycle parameters
0	Start wheelhead, start workhead, coolant on *Wheel rpm (Grinding wheel), Workpiece rpm (Workpiece / "C" axis)*
1	Initial infeed at axis rapid feed to the start position on workpiece. *Start position (Z axis), Rapid feedrate (Z axis)*
2	Initial infeed at axis rapid feed to the start diameter of workpiece. *Start Diameter (X axis), Rapid feedrate (X axis)*
3	Period of Coarse infeed to perform roughing operations and stock removal. *Medium / Fine Diameter (X axis), Coarse / Medium feedrate (X axis)*
4	Optional period of Medium infeed to perform main stock removal. *Fine Diameter (X axis), Medium feedrate (X axis)*
5	Period of Fine infeed to perform finishing. *Size diameter (X axis), Fine feedrate (X axis)*
6	Sparkout dwell period with no infeed. *Dwell time*
7	Retraction of wheel to safe distance. *Retract position / distance (X axis), Rapid feedrate (X axis)*

Table 3.1 Key phases of a typical Plunge grinding cycle

3.2.2.2 Enhanced Plunge grinding

It is often desirable to enhance or optimize the plunge grinding cycles on a production machine producing batches of parts. Established strategies include:

- Plunge cycle with Gap Elimination (initial touch monitoring)
- Plunge cycle with Burn elimination (wheel power monitoring)
- Plunge cycle with Size control (in-process gauging)
- Plunge cycle with overshoot
- Plunge cycle with adaptive dwell
- Plunge cycle with adaptive fine feed dwell period

3.2.2.3 Traverse grinding

As with plunge grinding, the objective of traverse grinding is stock reduction from a nominal base to a final target size. It is similar to plunge grinding in that material is generally removed in four phases (Roughing, Stock removal, Finishing and Sparkout), however it differs by making a series of passes along the workpiece, with each axis reversal accompanied by an infeed increment of a value that depends on the cycle phase. The roughing phase has a larger increment, this decreases for the cutting and finishing phases, and the sparkout phase has a number of finishing passes with no infeed. The relaxation of the workpiece during sparkout helps attain the correct diameter along the part's length, reducing the "barreling" effect. Table 3.2 below lists the phases and programming quantities of a basic traverse cycle.

Phase	Machine action and cycle parameters
0	Start wheelhead, start workhead, coolant on *Wheel rpm (Grinding wheel), Workpiece rpm (Workpiece / "C" axis)*
1	Traverse to left start end of workpiece at traverse feedrate. *Left Start position (Z axis), Traverse feedrate (Z axis)*
2	Initial infeed at axis rapid feed to the start diameter of workpiece. *Start Diameter (X axis), Rapid feedrate (X axis)*
3	Infeed by coarse increment, traverse to right end of workpiece. *Coarse Infeed Increment, (X axis)* *Right reverse position (Z axis), Traverse feedrate (Z axis)*
4	Infeed by coarse increment, traverse to left end of workpiece. *Left Start position (Z axis)*
5	Repeat until fine feed size reached, switch to finish grinding. *Fine Diameter (X axis)*
6	Infeed by fine increment, traverse to right end of workpiece. *Fine Infeed Increment (X axis)* *Right reverse position (Z axis), Traverse feedrate (Z axis)*
7	Infeed by fine increment, traverse to left end of workpiece. *Left Start position (Z axis)*
8	Repeat until final size reached, switch to sparkout grinding. *Size Diameter (X axis)*
9	Execute sparkout passes with no infeed. *Dwell passes*
10	Retraction of wheel to safe distance. *Retract position / distance (X axis), Rapid feedrate (X axis)*

Table 3.2 Key phases of a typical Traverse grinding cycle

3.2.2.4 Wheel Dressing

Dressing is the process of preparing or restoring the cutting surface of a grinding wheel. It is undertaken either initially before a new wheel is used, or during or after working when the wheel surface becomes dull or contaminated with machining residues. Dressing gives the wheel the correct roundness and also geometrical form (profile). It also gives the wheel the necessary roughness and cutting ability.

The simplest and most common dressing procedure is single-point diamond dressing. Here the wheel is traversed by the point of a diamond-tipped dressing tool that is mounted in a fixed location on the machine table. After each traverse the wheel is moved closer to the tool by a defined increment, and a further amount of the wheel material is removed by the next traverse pass. When the operator or the machine control determines that sufficient material has been removed, the wheel is retracted from the dresser and is ready for use. The wheel diameter will then have been reduced by a known amount.

The key parameters of the basic dressing procedure are given in Table 3.3: these are used directly to control the cycle execution [3] and are of fundamental importance to the process.

Parameter	Symbol	Notes
Total infeed	i_d	total amount of material to be removed
Depth of cut	a_d	infeed amount per pass
Crossfeed rate	v_f	traverse speed or dressing feedrate
Wheel speed	n_w	RPM

Table 3.3 Physical parameters of a dressing cycle

Several conceptual or calculated quantities are also given in Table 3.4 below: these are used to derive optimized settings for the parameters listed above.

Parameter	Symbol	Notes
Dressing lead	f_d	Crossfeed per revolution
Effective cutting width	b_c	Tip size of dressing tool
Overlap ratio	U_d	Number of passes over a point on the wheel

Table 3.4 Conceptual parameters of a dressing cycle

The overlap ratio determines the smoothness / coarseness of the wheel surface. The chosen overlap ratio depends on the grinding operation and desired surface finish:

- Roughing Ud = 2-3
- General grinding Ud = 3-4
- Finishing Ud = 6-8

From these the required Crossfeed rate can be determined:

Crossfeed rate = (wheel RPM x effective width) / overlap ratio

The following general rules apply:

- Small amounts of infeed should be used (5 – 30 micron).
- Higher crossfeed velocity gives higher surface roughness.
- The wheel should not be traversed without incrementing the infeed.
- Coolant should always be applied during dressing.

N.B. In many respects a wheel dressing cycle can be viewed as a variation of a traverse grinding cycle.

3.2.3 Part Programs and ISO programming

3.2.3.1 Part Programs

A part program is a sequence of instructions that will allow the machine tool to execute the machining operations required for a particular workpiece or part. The original implementation on early NC and CNC machines used punched tape to store the program, this then evolved into the now common method of having program files stored on computer disk.

A CNC system will require editing features such as a keyboard and display screens, in order for the operator or programmer to enter and later modify the part programs. Most controls allow the operator to step through the program one line (or action) at a time, for example in order to validate and optimise a program safely during development.

3.2.3.2 ISO programming (G and M code programming)

ISO (International Standards Organisation) programming is a traditional and widely recognised method of writing Part programs using a small and essentially standard instruction codes, together with an optional list of parameter variables and numeric values. These are known as G-codes (for machine moves) and M-codes (for machine functions). A part program is comprised of a (usually numbered) sequence of instruction lines or "blocks" of these codes, arranged to perform the required operations to machine a complete part.

3.2.3.3 G-codes

These are classed as Preparatory Commands [15], and they essentially control the motions of the machine. In addition they can change some operational features of the machine tool with respect to distances and positions, etc. G-codes are also called preparatory codes, and are a word in a CNC program that begins with the letter G followed by an ID number. Essentially it is a code telling the machine tool what type of action to perform, as well as defining the machine element affected and the required numeric value to apply. Some examples of G-codes and their variables for Fanuc controls are given in Tables 3.5 and 3.6 below:

G-code	Description
G00	Rapid positioning
G01	Linear interpolation
G04	Dwell
G20	Programming in inches
G21	Programming in mm
G28	Return to home position
G80	Cancel canned cycle
G90	Absolute programming
G91	Incremental programming

Table 3.5 Example G-codes (Fanuc)

G-code variable	Description
X	Absolute or incremental position of X axis
Y	Absolute or incremental position of Y axis
F	Defines feed rate for tool
S	Defines spindle speed
N	Line number in program
T	Tool selection

Table 3.6 Example G-code variables (Fanuc)

3.2.3.4 M-codes

These are known as Miscellaneous or Machine Functions, and they perform set actions or change some operational features of the machine tool. Some typical M-codes are shown in Table 3.7 below.

M-code	Description
M00	Program Stop (non-optional)
M01	Optional Stop, machine will only stop if operator selects this option
M02	End of Program
M03	Spindle on (CW rotation)
M04	Spindle on (CCW rotation)
M05	Spindle Stop
M06	Tool Change
M07	Coolant on (flood)
M08	Coolant on (mist)
M09	Coolant off

Table 3.7 Example M-codes (Fanuc)

3.3 Control System design

3.3.1 Introduction

Control systems for grinding machines vary from simple manual and mechanical interfaces to complete CNC implementations with features for complex machining and dressing operations with integrated process monitoring equipment. The system chosen by the manufacturer (and customer) will be determined by several factors:

- Ease or sophistication of operation
- Advantages of proprietary or commercial (i.e. standard) technology.
- Integration or addition of enhanced features
- Commonality with other machines and products
- Cost and complexity of solution

There are 4 main implementations of grinding machine control systems:

- Fully Manual
- Electronically Enhanced Manual (SAMM – Servo Assisted Manual Machine)
- Full CNC
- Full CNC with simplified HMI (e.g. Touch-screen, guided programming)

3.3.2 HMI requirements

A typical grinding machine HMI (Human Machine Interface) will feature most of the following in some form:

- Switches or buttons
- Indicator lamps or symbols
- Position displays
- Keyboard data entry / program editing features/auxiliary displays
- Manual axis control wheels
- Feedrate override
- Emergency Stop
- Cycle Start/Stop
- Progress messages / indications
- Error messages / indications

Modern control systems attempt to incorporate most control features into the main operator interface and remove the number of switches, buttons and lamps used. However there will remain a core of conventional operator controls that are needed for safe and convenient operation of machine features.

Touch-screen interfaces have been developed to allow compact, flexible and simple user interfaces to allow operators to quickly program and execute grinding cycles. These give the ease of use common to modern graphical computer interfaces, with simple graphics and numbers largely replacing text and input keys.

3.3.3 Control system elements

The main features of a machine tool control system are:

- Computer processing unit with parameter and program storage
- Operator HMI panel
- Operator input and control features
- Operator program and status display features
- Axis control and position measurement electronics
- Machine / Control signal interfacing
- Additional measurement and control equipment

A modern control system will typically comprise standard modules from a single manufacturer (e.g. Siemens, Fanuc, Heidenhain), configured according to the system needs. For lower-cost machines it can be cost-effective for a machine tool builder to develop or buy a dedicated, custom-made controller, although simplified product ranges are often available from commercial manufacturers.

3.3.4 Control solution development

A modern grinding machine will rarely be sold as a standard item, in practice it will be customised or adapted to the specific requirements of the customer. Extra equipment and operating features may be needed and it will normally be the job of an Applications Engineer to specify, design, implement and commission the solution. This can take up to 6 months depending on complexity and adds significantly to the project cost.

The engineer must tailor system software, logic and cycles to customer's need. This usually involves working with features such as ladder logic, M- and G-codes, macros, machine cycles and custom software programs. Enhancements such as automatic wheel balancing touch detection, power monitoring, probing and gauging will all require specific hardware and software interfacing with the machine control. The revised control system design reduces the workload of machine cycle customisation and auxiliary equipment installation.

System devices are interconnected to transmit and exchange:

- Control signals
- Status signals
- Process data
- Configuration data

3.3.5 Control system integration

The various control system devices described previously must be connected in order to exchange control and status signals, as well as the optional transmission of operational and configuration data. The relevant fundamental features, functions, quantities and interfaces in a typical grinding system are summarized in Figure 3.5 below [16].

The principal methods of machine tool equipment interconnection are shown in Table 3.8 below. These interface types must all be accommodated by any new, flexible system architecture design.

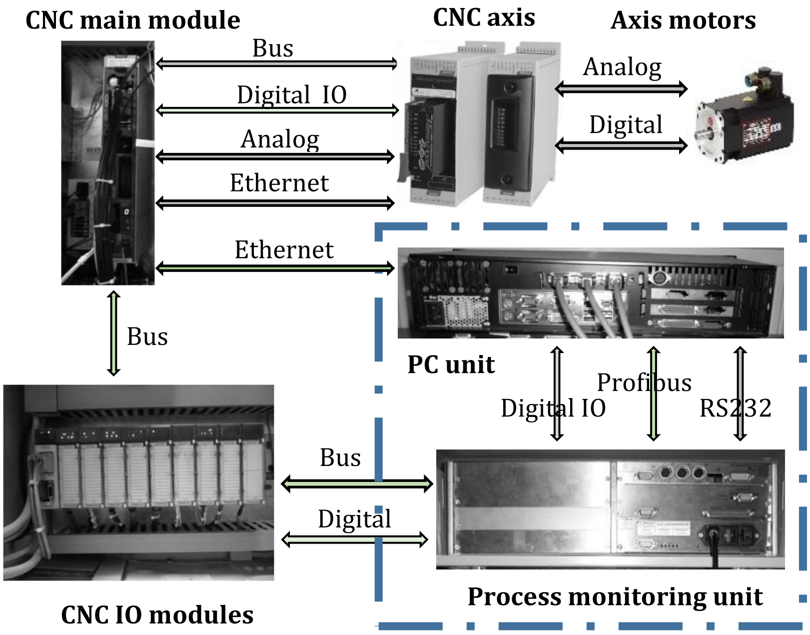


Figure 3.5 Typical control system components and connections

Interface	Features and notes
Analog	Conventional method: Variable voltage DC signals / commands. *Control of device, measurement signal from sensor*
Digital I/O	Conventional method: Low voltage DC digital inputs and outputs. *Each line set to 0 or 24V level : Low / High, On / Off, True / False*
Serial link	Conventional method: (RS232, RS422, RS485, USB). *Direct Port-Port connection, high speeds now possible*
Data bus	More sophisticated and capable: (Profibus, Modbus, Interbus, ...) *Devices daisy-chained together: 1 x Master, n x Slaves*
Network	Modern and flexible: (Profinet, DeviceNet, ...) *Ethernet / TCPIP based*

Table 3.8 Connection methods for control system components

3.3.6 Control system improvements

From Subsections 3.3.4 and 3.3.5 it can be seen that it would be beneficial within the overall control system architecture to handle the high-level transfer of data and signals between devices in a standardized way, independent of the device, connection and transport method used for the actual communications. Adoption of a harmonized transfer strategy would significantly reduce the effort associated with system configuration and bespoke machine development. It would also allow end users to more easily add further functionality to their existing equipment.

The outcomes of the previous research programs outlined in Subsection 2.5 have determined that it is logical to visualize the control architecture as a structure or network of interconnected objects or devices. These have distinct internal and external characteristics and functionality that can be described and defined in a formalized way for inclusion in an enhanced and expandable software design.

4 Auxiliary Process Control Equipment

This section details the issues and solutions relevant to the monitoring and improvement of the grinding process, and discusses how auxiliary equipment is installed and operated in conjunction with the machine tool control system. It was necessary to fully consider the integration and operation of these devices within the design framework. Two specific monitoring device examples were selected for inclusion in the project, and their characteristics and differences are explained.

4.1 Introduction

4.1.1 Control, monitoring and optimization of the Grinding process

A cylindrical grinding machine performs machining operations on rotating workpieces, and dressing operations on the grinding wheels. These actions are performed either manually under operator control, or automatically as a programmed cycle. The fundamental process parameters for grinding or dressing are wheelspeed and workpiece speed, and additionally the different cycles will define various infeed rates, infeed setpoints (Coarse, Medium, Fine, Final size), Reversal points (Left and Right) and Dwells. The machine control will execute a sequence of moves to the programmed axis positions at the appropriate speeds, and the expectation is that a satisfactory part will be produced.

In reality, the physical variability of the process, such as wheel wear, machine deflections, and temperature variations mean that adjustments to the grinding parameters need to be made in order to improve the quality of the finished part. Refinements to the process are often made in response to changes in the part dimensions, surface finish or roundness identified through post-process measurements, however any modifications to process parameters by the operator or process engineer will be based upon experience and knowledge of the process gained over many years.

By introducing process monitoring equipment (for example: gauging units, accelerometers, AE sensors, power sensors) to the machine it is possible to dynamically identify and subsequently correct quality problems in the process, and also to improve productivity and efficiency. The relevant production issues and monitoring solutions are summarized in Tables 4.1, 4.2 and 4.3 below:

Quality Issues	Causes	Monitoring Solution
Size tolerance	System deflections Wheel wear	In-Process Gauging Power / AE monitoring
Surface roughness	Wheel selection / condition Grinding parameters	Post-Process Gauging
Roundness	Grinding parameters Wheel unbalance	Post-Process Gauging Wheel balancing
Burning	Wheel wear / dulling Excessive infeed	Power monitoring
Spindle wear	Wheel unbalance	Wheel balancing

Table 4.1 Grinding quality issues

Efficiency Issues		**Monitoring Solution**
Reduction in:	Air grinding time before machining	Power / AE monitoring
	Dwell time after machining	Power / AE monitoring
	Excessive wheel dressing intervals Insufficient wheel dressing intervals	Power / AE monitoring
Optimisation of:	Infeed rates	Power monitoring
	Material removal during dressing	AE monitoring

Table 4.2 Grinding efficiency issues

Safety Issues		**Monitoring Solution**
Detection of:	Wheel collision	Power / AE monitoring
	Wheel failure	Unbalance monitoring
	Wheel overspeed	Speed monitoring

Table 4.3 Grinding safety issues

The inclusion of auxiliary equipment to improve the quality and efficiency of the grinding process was to be a key objective of this research.

4.1.2 Monitoring equipment characteristics

The external process monitoring equipment is generally a stand-alone unit connected to the machine control via a wiring or communications interface. The operator will set the device operating parameters and monitor its behaviour via a control panel. The devices can generally operate in Manual mode (operator control) or Automatic mode (machine control). They will generally execute one or more monitoring or activity cycles, which typically involves an initial start or reset operation that clears all status signals and commences the function. As various programmed conditions or threshold values are met during machining, appropriate signals and visual indications are set by the device. The operator or machine control will then respond to this information, for example by ending the machining cycle.

4.1.3 Wheel balancing

A wheel balancing unit will monitor vibration and speed of the grinding wheel, and calculate the unbalance magnitude and vector. It can display this to the operator, and set status signals if preset limits are exceeded. An automatic balance cycle can be run which moves mechanical weights or transfers fluid to minimise the unbalance. In addition the operator can manually control the unit. The balancer can also generate an alarm condition if excessive unbalance or wheel speed is detected.

4.1.4 Touch and crash detection

The touch detector unit (or “Gap and Crash”) typically monitors an acoustic (Ultrasonic) sensor channel and performs signal processing tasks upon it. If the sensor signal exceeds preset levels and logic conditions, a status signal is set to indicate reaching a particular limit or threshold. This typically allows the machine to approach a new or rough workpiece at high speed, and then reduce the feedrate immediately the first signs of wheel contact are detected. This strategy is known as Gap Elimination, and reduces unproductive "air grinding". Additionally a higher level of signal could indicate a wheel crash condition, triggering an emergency stop or shutdown (Alarm condition).

4.1.5 Power monitoring

Wheel spindle power monitoring is also very useful and can detect burn conditions as well as touch and crash. It has however the disadvantage that response times are slower due to the motor drive electronics and sensitivity is lower compared with systems using acoustic emission based detection. In addition the equipment must be calibrated to provide an absolute value for transmitted grinding power.

4.1.6 In-process gauging

The diameter gauge unit is programmed to indicate various size thresholds during machining of the part and these will trigger a change in the machine's behaviour e.g. infeed rate. These distances are set relative to a master (zero) part, and do not depend on variable machine conditions. When the part is at size the gauge will retract and the machine will end its cycle. Flag (or shoulder) gauging is used to locate a workpiece feature precisely in order to measure and transmit an exact position to be used by further grinding operations.

4.2 Operation with the machine tool control

4.2.1 Monitoring equipment cycles

The monitoring equipment is generally synchronized to the execution of a machining cycle by the CNC. When the grinding cycle starts the monitoring activity will be initialized with a "cycle start" or Reset command signal which will clear all threshold or failure signals and indications and start the specific monitoring or control logic.

A wheel balancer unit will perform a balancing cycle that achieves a minimum level of unbalance, it will then set an "In Tolerance" signal for the control. The cycle can be aborted by the CNC or operator if desired, alternatively it could "time out" if satisfactory balancing is not possible. It can be seen that there are certain similarities between a wheel balancer and a machine controller in that they both actively perform cycles of mechanical operations.

Touch Detector and Gauge unit monitoring cycles are similar in operation to each other in that they respond to the input sensors reaching a series of programmed limit values, and indicate these states to the machine control which adjusts the process accordingly. The gauge unit differs in that it will also extend and retract the gauge head towards and away from the workpiece (usually with pneumatics), and open and close the measuring fingers around the workpiece (electrically).

4.2.2 Grinding control system component structure

The fundamental features, functions, quantities and interfaces in a typical grinding system are summarised in Figure 4.1 below. From this it can be seen that from a system analysis perspective there are several key components that describe each control system element, and there are many physical and conceptual similarities between the various features and actions that the units provide.

Machine Control	
Cycles	Plunge Traverse Dress
Commands	Start / Reset cycle Select cycle Set parameter
Integration	Machine logic Custom software
Interfaces	Digital I/O Analog signals Serial Bus Network

In-Process Gauge	
Cycles	Diameter gauging Flag gauging
Sensors	Gauging head Rotation
Signals	Start / Reset cycle Limits Faults / Alarms
Data	Size Position

Wheel Balancer	
Cycles	Automatic balance Manual balance Neutral Balance
Sensors	Vibration Rotation
Signals	Start / Reset cycle Limits Faults / Alarms
Data	Unbalance Speed

Touch Detector	
Cycles	Touch monitoring Burn Monitoring Crash monitoring
Sensors	Acoustic Emission Power Special (Force, Strain, ...)
Signals	Start / Reset cycle Limits Faults / Alarms
Data	Acoustic levels Power levels

Figure 4.1 Summary of key grinding process control features

4.2.3 Interaction between the machine control and peripheral devices

The key features, operations and data of the grinding process control to be managed by the machine controller in an integrated system can be summarized as follows:

Main control actions:

- Device operation
- Device monitoring
- Process modification

Main process data:

- Control, status and alarm signals
- Process data values
- Device parameters

The issues and challenges relating to the development of a flexible, generic hardware and software interface for a range of controls and auxiliary equipment units are:

- Lack of standardisation between CNC and equipment manufacturers
- Different interfacing hardware and strategies with Process Control equipment.
- Different levels of functionality / complexity between equipment ranges

4.3 Grinding process control equipment study

4.3.1 Introduction

For the project it was necessary to specify and implement the interfacing for one or more items of commercial process control equipment. The principal units chosen were Touch Detection devices which have simpler installation requirements, typically comprising the control unit (or amplifier) and an acoustic emission sensor connected via a cable and mounted with a magnet or a screw fitting. Initially the Touch Detection unit of the Balance Systems VM20 modular system was studied, followed by the newer, simpler VM9TD device from the same manufacturer. Additionally the VM20 Wheel Balancing and Gauging units were also evaluated for comparison.

4.3.2 Balance Systems VM20

This is a modular system featuring a control panel, system rack with power supply, plus appropriate function cards for wheel balancing, touch detection, gauging and communications. The AMT laboratory VM20 has a Touch Detection card that can support 2 x Acoustic channels and a Power monitoring channel. It has digital and analog IO, and a Multilink communications card that allows control and data transfer via Serial, Parallel or Profibus channels. Wheel Balancing and Gauging functions can be added by slotting extra function cards into the rack and altering the system configuration settings. A typical installation in a machine control cabinet is shown in Figure 4.2 below.

The VM20 rack power supply unit is the master or “System” unit, it manages the overall function of the equipment package. The operator HMI panel is another system “device”, it is not actually needed to run the VM20 and can be disconnected or replaced with a PC front end. Each function card has several connectors for the input and output signals to the peripheral sensors, actuators and control units.

Figure 4.2 Balance Systems VM20 rack system with touch detector card

4.3.2.1 Operator HMI panel

This provides a screen and simple keypad to allow operator interaction with all installed function cards. The operator selects the required device page, and can enter the setup or operational screens for that device. The screens either display current process information as values, graph traces and symbols, or a selection of device parameters for viewing and editing.

Various password-controlled “Access Levels” are defined, comprising Observer, Operator, Programmer, Installer and Factory. These give different levels of authority to the user when accessing the system’s features. For example Observer level allows no interaction, Programmer allows modification of some parameters, and Factory enables low-level configuration of the system hardware and software.

4.3.2.2 Setup

The user enters a setup mode where the required device is selected and various levels of operating parameters are then accessed and edited. He navigates to the required quantity, adjusts its value and optionally saves the changes. The parameters are grouped in a hierarchy depending on overall function and effect: Work, Setup and Option. The Work parameters relate to the settings for the monitoring program (thresholds etc.), and the Setup parameters access the channel and sensor adjustments. The Option parameter level will enable or disable various special features and set configuration options.

4.3.2.3 Touch detector operation and monitoring

The main Touch detector operating screen allows the operator to select different “part programs”, which have sets of threshold conditions defined for a particular operation and based on input channel signal levels. The inputs can be from acoustic emission transducers or power monitoring units. In manual mode the operator can reset the monitoring cycle and limit signals: in automatic mode this is done by the external CNC. As well as numeric values and indicator signals the screen shows a graphic trace of the selected signal values plotted against time: this gives the operator an excellent overview of the grinding cycle as it progresses.

4.3.2.4 Communications

The unit's additional "Multilink" card provides several extra communication links for external control and monitoring of the VM20. In particular a Profibus interface is provided: this uses an RS485 command and data protocol and allows the VM20 to be configured as a Slave device on a network under the control of a Master. The master can issue commands to the VM20 device, and read and write data within the internal firmware. This is a significant enhancement to the overall system capabilities.

Profibus devices communicate both Cyclic and Acyclic information via a defined data structure. Cyclic data is transmitted at regular intervals (e.g status flags and monitoring values), and Acyclic data is transmitted in response to events or requests (e.g. read a set of parameters).

4.3.3 Balance Systems VM9 TD

This is a new lower-cost product for Touch Detection / Gap & Crash control. It is one of a range of 3 complementary units produced by this supplier, the others are the VM9BA (Wheel Balancer) and VM9GA (Gauge). All have a similar architecture, appearance and operational philosophy.

Figure 4.3 Balance Systems VM9TD with acoustic emission transducer

The Touch Detector unit features 2 Acoustic signal channels AE1 and AE2, each with 2 trigger thresholds and signals. Only one channel is active at a time and this is selected by the operator or CNC depending on which operation or area of the machine is to be monitored. The VM9TD has a simple pushbutton and LED operator interface, and permits CNC control via digital IO or RS232 serial communications. A partial interfacing description for this unit can be found in Section 4.4.

4.3.3.1 Setup

Each acoustic channel has a parameter setting for Trigger levels 1 and 2, which determine the signal amplitudes that determine a touch condition. The channel's gain and filter values determine the sensitivity and responsiveness of the channel's processed signal in amplitude and time. The operator configures the unit by selecting one feature at a time and adjusting the currently set value up or down via the pushbuttons. The selected parameter (channel selection, amplifier gain, smoothing filter setting, etc.) is indicated by an LED. A multifunction vertical bargraph display shows the currently edited parameter value.

4.3.3.2 Operation and monitoring

The unit can operate in manual mode (as an indicator unit for the operator) or automatic mode (synchronized with the CNC). Monitoring is started with a Reset operation. When the unit is in cycle the bargraph displays the live monitoring signal level for the selected channel. As an input signal threshold is reached the appropriate panel LED is lit and an output signal line is set "High" or "On". If the signals are configured as "latched" they will stay set after triggering until the monitoring cycle is reset, otherwise they will vary dynamically with the input signal level.

4.3.3.3 Communications

There is a serial RS232 communications protocol provided which will allow an external computer to read the monitoring signal settings and levels, as well as to control the VM9TD unit using commands and to configure its internal setup and working parameters. The unit is put into a mode where its front panel is overridden and it is then under external control.

4.4 Digital IO interfacing definitions

In order to clearly define the standard digital IO signal assignments required for various different monitoring units, a formalised IO Interface description table was developed. Using a standard tabular format and naming style, each unit's wiring connectors are identified by type and defined pin-by-pin so that each signal line's function is clearly described. As well as the functions it also identifies which pins are inputs and outputs, as well as power supply and ground / reference lines. The table also reports the signal logic levels (the sense of signal) and the signal wiring scheme (Source or Sink). Two examples of these definition tables for similar devices are given in figures 4.4 and 4.5 below.

4.5 Summary of selected equipment

The two process control devices selected for implementation were from the same supplier (Balance Systems) and performed the same function (Touch Detection), but differed in terms of complexity, functionality and connectivity. This allowed a comprehensive study of the different features available on typical units, and the methods and actions required to access their functions and data using different interfacing hardware and communication protocols.

Model	Description		Comments	Status	
VM20-GE	Modular 4-channel Touch Detector Unit		Gap / Crash	New	
Conn ID	**Name**	**Type**	**Standard**	**Style**	**Make**
T6	Digital IO port	15-W D Male	M	D-sub	Cannon
Pin	**Name**	**Type**	**Logic**	**Source**	**Sink**
1	GND	Supply ground			
2	Prog Select 1	Input 1	Positive	+	-
3	Alarm Reset B	Input 2	Positive	+	-
4	Alarm Reset A	Input 3	Positive	+	-
5	Touch B	Output 1	Negative	+	-
	Echo Prog Select 0		Positive	+	-
6	Alarm B	Output 2	Negative	+	-
	Echo Automatic		Positive	+	-
7	Burn A	Output 3	Negative	+	-
8	+24V	Supply +ve			
9	Reserved	Input 4		+	-
10	Prog Select 0	Input 5	Positive	+	-
11	Touch/Burn Reset B	Input 6	Positive	+	-
12	Touch/Burn Reset A	Input 7	Positive	+	-
13	Burn B	Output 4	Negative	+	-
	Echo Prog Select 1		Positive	+	-
14	Touch A	Output 5	Negative	+	-
15	Alarm A	Output 6	Negative	+	-

Figure 4.4 VM20GE digital interface definitions

Model	Description		Comments	Status	
VM9-TD	Compact 2-channel Touch Detector Unit		Gap / Crash	New	
Conn ID	**Name**	**Type**	**Standard**	**Style**	**Make**
T4	Digital IO port	25-W D Male	M	D-sub	Cannon
Pin	**Name**	**Type**	**Logic**	**Source**	**Sink**
1	Echo Automatic	Output 1	Positive	+	-
3	AE1 - Limit 1	Output 3	Negative (safety)	+	-
4	AE1 - Limit 2	Output 4	Negative (safety)	+	-
5	AE2 - Limit 1	Output 5	Negative (safety)	+	-
6	AE2 - Limit 2	Output 6	Negative (safety)	+	-
7	Echo Select AE1/AE2	Output 7	Positive	+	-
8	Select Automatic	Input 1	Positive	+	-
9	Outputs Common	Output 0		+	-
10	GND	Supply ground			
11	Select AE1/AE2	Input 2	Positive	+	-
12	Reserved	Input 3	Positive	+	-
13	Inputs Common	Input 0		+	-
15	Reserved	Output 8	TBD	+	-
21	24V DC	Supply +ve			
22	Reserved	Input 4	TBD	+	-
23	Reset AE1	Input 5	Positive	+	-
24	Reset AE2	Input 6	Positive	+	-
25	GND	Supply ground			

Figure 4.5 VM9TD digital interface definitions

5 System Analysis and Design

In this section the 1300X grinding machine and its operations are reviewed, together with the requirements and design of the new control hardware and any modifications needed. The key features and functions of a grinding machine control system are detailed, and a hierarchical structural design for the new controller is presented. A similar review of typical Process Control devices is undertaken, and their functional hierarchy from the operator interface to the system interconnection protocols is analysed. A layered component (or Device object) hierarchy is described, and a structural design for a generic Process Control device is presented.

5.1 Jones and Shipman 1300X Grinding Machine

Figure 5.1 AMTReL Jones & Shipman 1300X Grinder

The 1300X universal grinder at AMTReL is a well-used prototype machine had most recently been used for a program of research into high speed internal grinding and dressing. It had been non-operational for several years since being superseded in the laboratory by a newer Jones and Shipman Ultramat machine with Fanuc CNC control and a Balance Systems VM20 monitoring system.

The initial task was to fully investigate the machine, to identify all machine elements, control and ancillary equipment, and to catalogue relevant electrical and hydraulic features. This established a foundation upon which the revised system could be analysed and developed.

5.1.1 Machine features and operational status

The 1300X features a large fixed bed with a translating horizontal bed housing the workhead, tailstock and dresser unit - this is the Z axis. The grinding wheels and motors are mounted on a combined external and internal wheelhead unit: this moves perpendicularly to the workpiece as the X axis. The X axis and Z axis servomotors are each driven by a Control Techniques Midi-Maestro servo drive with a 415V 3-phase AC supply. Each axis has a rotary encoder mounted for position feedback to the motion controller. A hydraulic system provides lubrication to the axis slideways: the pump is activated when the machine

Control is on. There is an air supply to the machine for raising the wheelhead, operating the wheelguard and supplying air for cleaning the machine. An external coolant tank with integral pump circulates lubricating fluid to the grinding area, and machining debris from the wheel and workpiece are carried away to the tank for filtering.

As initially found the machine was non-operational as the X and Z axis servos had drifted so that the wheelhead and table had reached their end-stops and limit switches. It was necessary to bypass the limit switches and slowly move the two axes away from the overtravel limits by use of a potentiometer-controlled battery box connected directly to the servo drives. It was demonstrated that the external grinding wheel could be started but the workhead was not operational. The internal grinding wheel spindle had been removed. The external coolant tank pump was working, as were the machine internal hydraulics.

5.1.2 Original control system

The original PC500 control computer was a small, robust industrial PC of 1990s vintage, featuring a 486 DX microprocessor, ISA bus architecture, MS-DOS operating system and a flash hard disk. It featured a digital IO card with 48 signal lines (24 inputs and 24 outputs), and a Deva 004 4-axis motion control card that handled the control and feedback signals for the axis motors, axis handwheels and the workhead motor. It was a requirement to replace this PC unit and its components with modern, improved equivalents that would allow further development.

5.1.3 Operator control panel and HMI

There is a large control console on the front of the machine featuring numeric displays (for axis positions, etc.), indicator LEDs, selection buttons, a keypad for data entry and two axis control handwheels (MPGs) to allow manual movement of the axes. Programming in Inch or mm is selectable. The primary control buttons are an E-stop, Control On and Cycle Start / Stop. A Feedrate Override (FRO) switch can modify the programmed travel speed of the axes by fixed percentages, and the workhead speed can be set via the keypad. There are pushbuttons to activate the wheelhead motor, workhead motor and coolant pump, these are interlocked via control relays. The Operator panel communicates with the PC500 control via a 2-way RS232 protocol, and the lamps and switches are wired to terminals and relays in the electrical cabinet. It was a requirement to move much of the functionality of this panel to the new operator software panel.

5.2 Revised Control System

The new control hardware for the project was specified and sourced, and comprised an Industrial PC, touch-screen monitor, and a PCI bus axis control card. Some wiring modifications were needed to interface the control signals to the machine due to differences in the of axis, encoder and IO connections. The PC was configured with the Windows XP operating system and MSDN / Visual Basic .NET software, and development continued using a conventional PC monitor, mouse and keyboard.

5.2.1 New PC control unit

The new control system was based on a small industrialised PC motherboard housed in a 2U high 19" rack case. This was installed in the existing equipment console, and had the following key attributes:

Control PC	**IEI Kino 945 GSE** Industrial Mini ITX board with Intel Atom 1.6GHz CPU (Fanless) 1 x PCI expansion slot 1 x 200pin DDR2 SODIMM slot 2 x SATA connectors, 1 x IDE 2 x RS232 ports on rear IO 1 x RS232 port (internal) 1 x RS232/422/485 port (internal) 8-bit digital I/O (TTL)
Memory	512MB-DDR2-T 533 MHz
Hard Disk	SATA
Operating system	Windows XP embedded
Power supply	110V/240V AC

The large number of RS232 and other ports provides convenient interfacing to other equipment.

Motion card	**Deva 004 PCI**	
	4 servo axis control	Encoder Inputs, Servo outputs
	Digital IO	16 Inputs, 12 Outputs, 1 x WDog
	Driver software libraries	Win2000, XP, Vista
	Utility software programs	Jogger, Axistuner

The PC rack containing the motherboard and Deva motion control card fitted is shown in Figure 5.2, together with the signal interfacing rack containing circuitry to adapt the various connection lines.

Figure 5.2 New control PC rack (top) with signal interface unit (below)

5.2.2 Motion control and digital IO

The DEVA004 PCI card is the current version of the card used in the original 1300X control PC. It has a modified layout which means that many signal connectors were different compared to the older version and an extra signal interface unit for conversion of the drive and encoder wiring was needed. Extra features such as digital IO lines are now included with the motion card, however this removed the requirement for an additional IO card. Its key features are as follows:

5.2.3 Modifications to the workhead drive

The Workhead servomotor had a non-standard servo drive with a dedicated single-phase 240 Volt transformer, the drive had failed and therefore required replacement. A new servo drive compatible with the existing workhead motor had to be fitted and following enquiries an MTS-200 unit from the Italian company Axor was selected. A new 3-phase transformer was also required to provide the correct supply voltage to the drive, this required wiring changes in the cabinet including an extra fuse. A filter to reduce electromagnetic interference to the mains supply could have been added but was not deemed necessary.

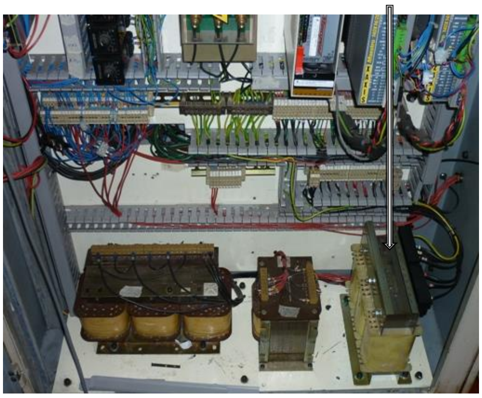


Figure 5.3 New wheel and workhead servo drive equipment

5.2.4 Further equipment modifications

The new design incorporated a new touchscreen monitor but kept the use of the original operator panel to provide various hardware functions such as the MPG handwheel. In addition the panel was treated in the design and development process as a discrete system device or component.

A Balance Systems power transducer box was added to the spindle drive electrical circuit, this monitors the voltage and current supplied to the motor and converts it to a power signal that connects to the VM20 Touch Detector unit via a fibre-optic cable.

5.3 Grinding controller analysis

Having studied and identified the main physical and operational features of the grinding machine, it was then necessary to identify the various functional groups or subsystems, and to identify discrete component devices and their interfaces. These were analyzed to determine similarities and differences in their characteristics, with a view to producing a hierarchical system design of “hard” and “soft” components. A hard component could be an actual device such as a coolant pump, and a soft component could be a more conceptual element such as a grinding cycle.

5.3.1 Control system analysis

The main features of a machine tool control system are:

- Computer processing unit with parameter and program storage
- Cycle and program management
- Safety strategy
- Operator HMI panel
- Operator input and control features
- Operator program and status display features
- Axis control and position measurement electronics
- Auxiliary process control interfaces

These elements can be considered as a set of packages or devices that compose the complete system. In Object-Oriented Design they are considered as Classes of Objects.

5.3.1.1 Machine operator HMI panel

A typical grinding machine HMI panel will feature most of the following in some form:

- Switches or buttons
- Indicator lamps or symbols
- Position displays
- Keyboard data entry / program editing features
- Manual axis control wheels
- Feedrate override
- Emergency Stop
- Cycle Start/Stop
- Progress messages / indications
- Error messages / indications

These are a mixture of hardware and software features, and are located on the control panel or screen but managed within the control application. The machine’s operator panel (hardware or software) is a Class of device made up of a selection of other lower level devices (or components) to implement its functions for display and operator input. The panel device comprises DRO displays, LEDs / lamps, switches, buttons and data-entry keypads.

Some hardware features can be simulated or replaced by software, particularly since the recent adoption of touch-screen interfaces. For example the physical operator keypad is used to enter and edit numeric cycle parameters, however this can be also done by a virtual keypad displayed on a screen. A "Keypad" device or object comprises a number of "Key" objects, each with their own defined attributes and actions. In addition a "Readout" object may display the currently entered value.

There is a data communication interface with the control program: the 1300X operator panel communicates with the PC500 control via a 2-way RS232 protocol sending formatted data strings. The traditional lamps and switches are wired to terminals and relays in the electrical cabinet.

5.3.1.2 Grinding cycle programming

The 1300X machine can run three Grinding operations: Plunge Grinding, Traverse Grinding and Wheel Dressing: these are selectable from the panel. These can be performed in Manual or Automatic mode. In Manual mode the operator controls the movement of the axes while referring to the position displays. In Automatic mode the steps of a programmed cycle are executed sequentially and the axis movements are controlled by the machine, visual feedback of the cycle status is given to the operator.

For each grinding operation various axis reference positions and speed parameters can be set. Most grinding cycle parameters are stored as a part program, typically in a named file that be selected, loaded and edited via a Program button. There may be a guided procedure to allow individual parameter modification, or a full editing screen may be provided. Other more general parameters such as axis datum positions may be saved separately during machine setup operations.

5.3.1.3 Grinding cycle execution

Automatic machining is initiated and controlled by the Cycle Start / Cycle Stop buttons. Once the program has been selected and entered the cycle operation can be started and then completed, paused or aborted. There may be other function buttons and switches available to manage cycle operation, for example axis hold buttons and Feed Rate Override switches. The controller software must monitor these and respond to them within the appropriate system modules.

5.3.2 Axis moves and operations

Axis control is fundamental to cycle execution, and the machine Axis object has key attributes position, speed, ID and channel. The principal Axis operations are Moves of various types.

In order perform a machining cycle, the X (wheelhead) and Z (workpiece) axes will have to be moved to various positions and at various speeds in a sequence determined by the cycle controller as it steps through the part program. The axis moves will often be relative to a preset Reference or Datum position, this can be a fixed position on the axis determined by a physical marker, or a variable position set by the operator and then stored. A series of axis moves must be prepared and activated, unless the move is being manually controlled by a handwheel or “jog” button.

Note that during programmed move execution there may be a conversion between any axis positions and part dimensions defined by the part program and the actual physical positions on the axes. This is the distinction between Incremental (relative) and Absolute programming. Also some machining dimensions may be defined as a final size or stock reduction as a *Diameter*, which must translate to a radial movement of the grinding wheel axis (X) to a specific position. Other parameters may define a move relative to a datum position on the workpiece or the axis itself.

A key subset of axis move types for use in automatic and manual operation is as follows:

5.3.2.1 Move at Speed

This simple move accelerates from rest and moves at a constant specified speed until the move is cancelled. This implements the axis Jog function. A negative speed moves the axis in reverse.

5.3.2.2 Move to Target

This is a fundamental axis move from the current position to a target position, with speed ramp-up and ramp-down. These moves can be queued consecutively to generate a move sequence.

5.3.2.3 Move with MPG

This causes the axis to move in response to position and speed demands from a Manual Pulse Generator wheel (MPG or HPG) on the operator console. The current Deva software does not provide this feature, so an interim solution was implemented by generating a queue of short axis moves at regular intervals depending on the data from the MPG encoder inputs.

5.3.2.4 Move to Reference

The axis will move slowly in a defined direction towards an encoder reference mark at one end of the axis, on detecting the mark it will save the axis position reading and optionally either add an offset value or set the position reading to zero.

5.3.3 Axis Motion Control

The Motion Control device manages the axis movements by adjusting the axis motor speed (via the servo drive) and monitoring the axis position (via the encoder): this is known as closed-loop feedback control. The actual motion control is implemented by the Deva 004 PC card and its associated driver libraries, with the motion profiles controlled according to predefined strategies and setup parameters. The control must detect error and alarm conditions such as when an axis stalls or hits a position limit marker. An axis must also be tuned or optimized to suit the machine’s mechanical characteristics and operational requirements, so that an acceptable motion response is achieved. A subset of the configuration parameters for the Deva 004 drivers are shown in Table 5.1 below:

Parameter	Description	Parameter	Description
CHANNEL	Axis Hardware port	MAXVOLT	Drive max. voltage
PITCH	Encoder pitch	MAXSPEED	Axis max. speed
COUNT	Encoder scale distance	MAXACCEL	Axis max. acceleration
KV	Proportional Gain	SRVERR	Axis move error limit
KI	Integral Gain	MINSRVERR	Axis standstill error limit

Table 5.1 Typical Deva 004 axis configuration parameters

5.3.4 Axis position measurement and display

Machine position readings (from the axis encoder) are displayed on a DRO or Digital Read Out device that converts the absolute axis position to a formatted and adjusted value for the operator to read. A DRO device will have a set number of digit / character elements available, plus display features such as leading zero suppression, adjustable decimal point, flashing, and extra non-numeric symbols. On the 1300X the DRO data is displayed to 3 decimal places and in the appropriate measuring units, the resolution is therefore 1 micron or 1 thousandth of an inch.

5.3.5 Machine auxiliary functions and control operations

There are a number of electromechanical devices and subsystems that are operated as part of the machine operations, either manually or when “In Cycle” and executing the various functions of the machine. These include the rotation of the Workhead and Spindle (both variations of an Axis), operation of coolant pumps and ventilation fans. They are generally actuated using digital IO signals, usually via relays. These elements are classed as system devices, their key feature is their state (On, Off, Enabled) and their key actions are their activation (Start, Stop).

5.3.6 Grinding controller structure summary

The diagram in Figure 5.4 below summarizes the hardware and software elements that comprise the Process Control Device concept. The Virtual Device is a software representation of the Actual physical device – it is the system interface that manages the relevant device data and manages the execution of the various device operations.

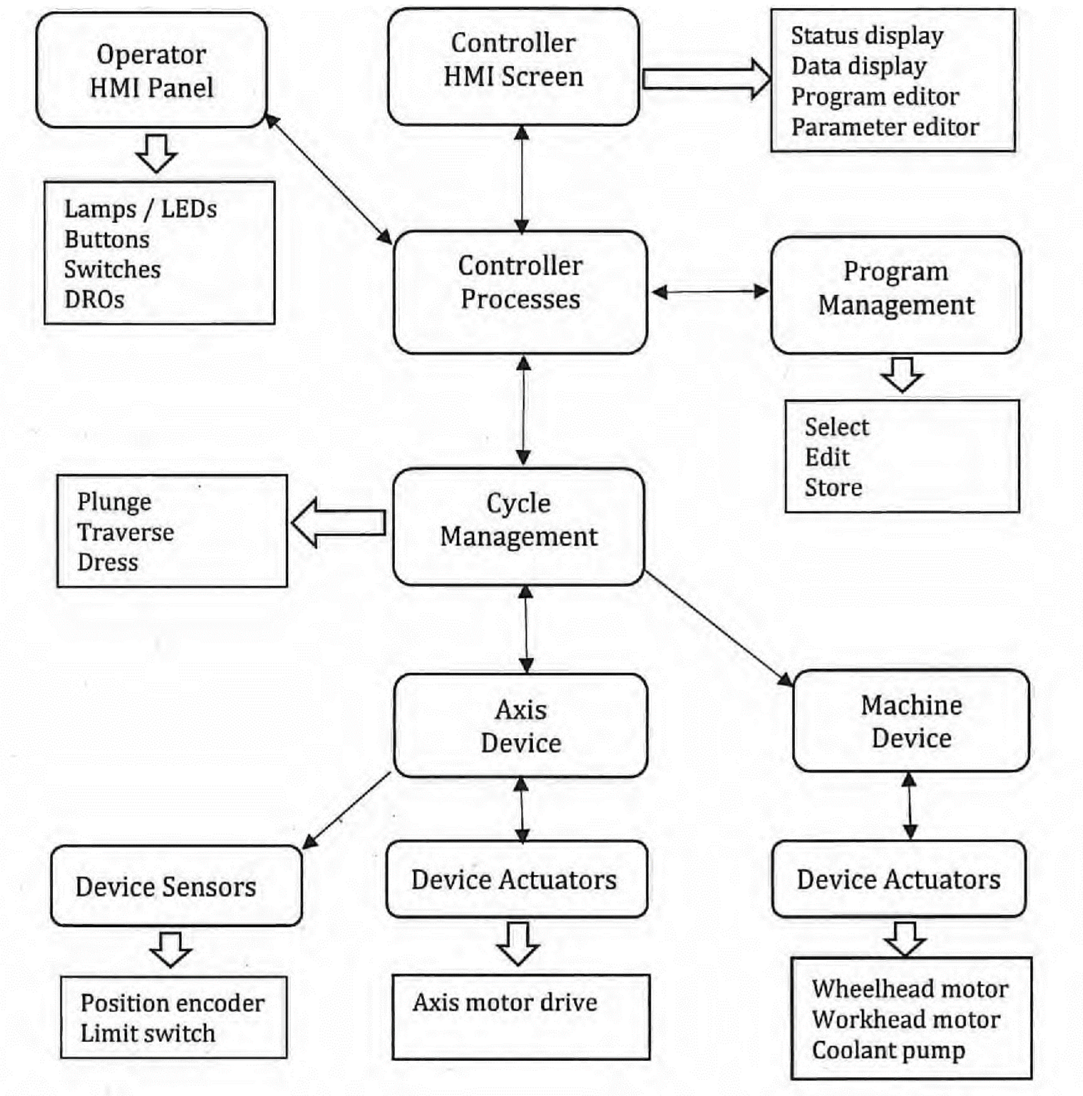


Figure 5.4 Functional structure of a grinding machine controller

5.4 Process control device analysis

5.4.1 Process control device characteristics

The various auxiliary grinding process monitoring and control devices described in Section 4 can be thought of as specialized variations of a generic device type or Class, specifically a Process Control (PC) Device. Implementing an Object-Oriented design, we can consider a simple Device base class, which can be refined into to a Machine Device class with its own specific features. From this we can define a Process Control Device class, and this can then give rise to more specific Touch Detector, Wheel Balancer and Gauge devices. Further refinement would then give even more specific derived classes, for example a Balance Systems Touch Detector Class and then a Balance Systems VM9TD Touch Detector Class.

The Balance Systems VM20 was studied in detail in order to quantify the key physical and operational features of a complete monitoring system offering all the relevant grinding control functions and sophisticated operating and communications features. The common or similar features between the differing device types were identified, and the handling of data within the system was formalised.

The key data types and operations of a typical fully-featured Process Control device are:

- Config data (Acyclic) Report device details
- Parameter data (Acyclic) Read / Write setup info
- Command data (ACyclic) Turn features On /Off, Request data, …
- Status data (Cyclic) Report device events
- Monitor data (Cyclic) Live device signal values
- Signal data (Cyclic) Digital Inputs / Outputs (limits, resets, …)

As mentioned previously, data operations are classed as *Cyclic* if performed regularly and *Acyclic* if performed intermittently or on-demand. The data is accessed via assorted communication channels and protocols, and will be in varying formats and from different locations within a device. For example, numeric data could be stored as two bytes at a particular memory address, and a signal could be represented as a bit setting within a byte. These diverse values should be collected and presented within the software in a standardized and logical format, and all access, decoding and formatting operations are hidden from the system programmer by layers of software within the device object. For example the live Monitor data from a VM9TD device could be read via an RS232 serial link and received as a single data package, whereas a VM20 device would provide a fixed block of Profibus data containing several items: both would be presented to the user program as a standard structure with a Value (formatted numeric) and a Valid (Boolean flag) element.

5.4.2 Device and machine control interfacing

Data and signal transmission between system devices and the control is generally achieved with a conventional digital IO interface, and optionally with a communications link such as RS232, data bus or network. These can be seen to be derived variants of a CommsLink object class, and they share common characteristics and operations while having specific differences that are handled internally within the component. They will all have common basic operations such as Initialise, Configure, Connect, Disconnect, Send data, Receive data, Decode data, etc., and these are provided to the user as standard library routines or Object Methods.

Once the machine control and a process control device are connected they can interact together with the control being the Master and the device being the Slave as summarized in Table 5.2 below:

Control actions	**Device actions**
Set Access level	Allow / deny operations
Select Automatic / Manual Select a program	Go to Automatic / Manual Change & report program
Change parameters Request parameters	Update parameters Return parameters
Set control commands Start / Stop a cycle	Respond to commands Execute a cycle
Monitor status signals Monitor process values Monitor errors / alarms	Return device status Return data values Report errors / alarms

Table 5.2 Control System and Monitoring Device interactions

5.4.3 Operator panel and controls

Process control devices generally have their own front control and display panels, but these can be replaced or duplicated by software screens provided by the machine control. Only one of the hardware or software panels should be allowed to be active at one time – we consider that the device is either under internal or external control. The control could send an "Override On" command or signal to disable the device panel, and "Override Off" to re-enable it.

The device software screen on the control should replicate the key features of the original device's HMI, but present them in a generally standard format and layout regardless of the specific device being controlled. Similarly any device parameter editing screens should be familiar to operators who have used equivalent screens for accessing the main control's setup and configuration features.

- Signal values display (numeric and / or bargraph)
- Threshold indicators display (lamp or symbol)
- Live signal trace (graph) display
- Data logging feature (to disk file)
- Editing features (including Keypad)

The "DataLogger" and "SignalGraph" components are very useful for in-process and post-process monitoring, giving a clear indication of the process behaviour as the grinding cycle executes.

5.4.4 Integrated control system construction

The generalised hierarchy of the Base and Derived Classes and the application objects comprising the process control device model is shown in Figure 5.3 below:

Level 4 **Application Object**	**Level 3** **Data Presentation**	**Level 2** **Session Management**	**Level 1** **Data Transport**
Object Instances	Derived Classes	Base classes	API routines
TouchDetectorUnit_1 TouchDetectorUnit_2 GaugeUnit BalancerUnit	TVM9_TD_Device TVM20_TD_Device TVM9_GA_Device TVM9_BA_Device	TDevice	
SerialLink PBLink	TSerial_Interface TProfibus_Interface		Win.Com1.Output = ..
TouchCycle1.Active	TSignalState	TProcessSignal	
AE.Channel(1).Value AE.Channel(1).Gain	TDataValue TParameterValue	TProcessData	
LogFile_12_27_11_2006 PlungeCycleLogfile	TData_Logfile TCycle_Logfile	TDataStore	Win.Stream1.Output = ..

Table 5.3 Control System and Monitoring Device interactions

The controller application creates an instance of the specific Device object (e.g. Marposs E20N Gap/Crash unit with 2 x Acoustic Emission channels), and the device interface accesses all the facilities, data and signals of the physical device. All specific communications and data conversion is handled within the device object. The controller application then provides standardized but flexible HMI access screens for the new device: their functionality and format will vary depending on the characteristics and capabilities of the equipment. For example some basic units only allow signal transmission, but others allow data communication and parameter setting.

5.4.5 Process control device structure

The diagram in Figure 5.5 below summarizes the hardware and software elements that comprise the Process Control Device concept. The Virtual Device is a software representation of the Actual physical device – it functions as the system interface that manages the relevant device data and manages the execution of the various operations performed by the Actual device.

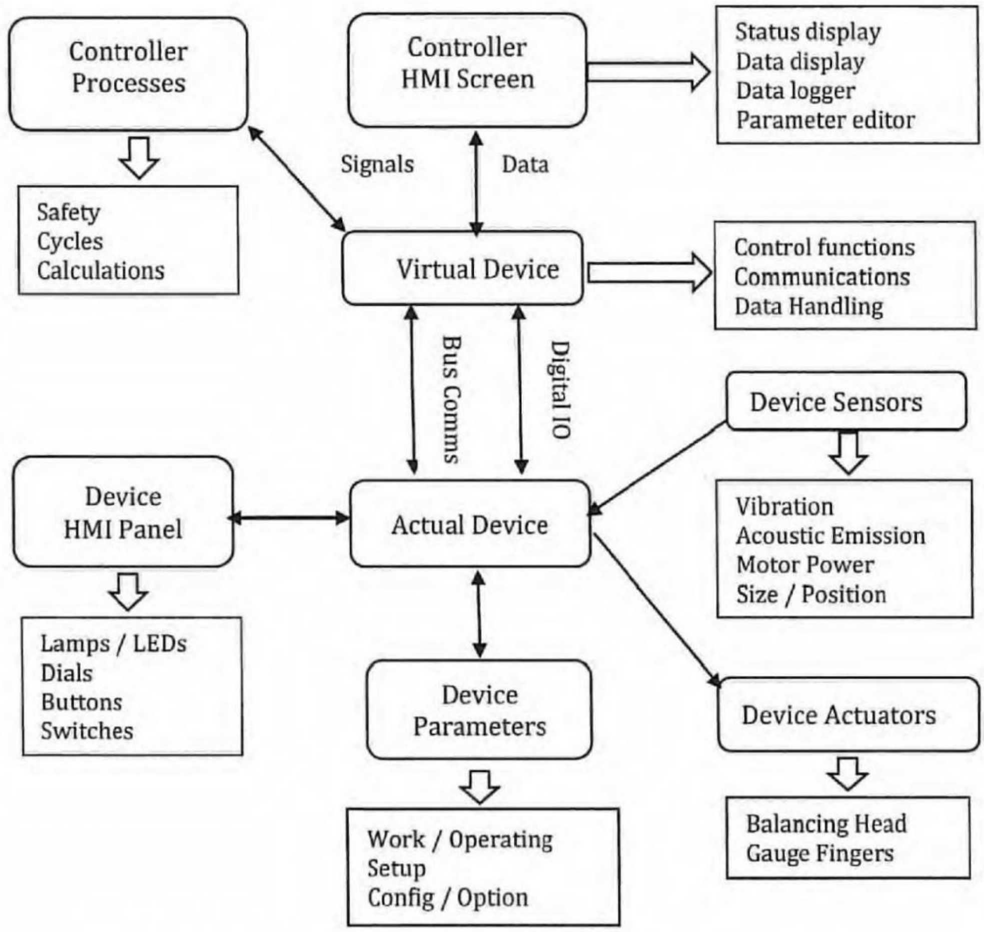


Figure 5.5 Machine Control and Process Control interaction

6 Implementation

This section presents the software concepts that provide the basis for the revised controller design, and describes the fundamental components, functions and structures that make up the design of the main controller. The design is expanded to accommodate the "Virtual Device" concept that defines and implements the additional equipment to be interfaced to the machine. The extension of the design framework to facilitate monitoring and control of other processes is discussed.

6.1 1300X controller software design

6.1.1 Software design methodology

6.1.1.1 Object hierarchy

As mentioned in section 5.4.4, the chosen software design structure comprises different levels of specific detail that creates functional abstraction and allows the top-level application to interact with a variety of devices without knowing the details of their low-level construction or operation. Each layer presents a generic "export" interface whereby its Public features are made available, and the object's Private or Protected features are kept hidden internally. This encapsulation allows internal modifications to be made without affecting existing users of the object library.

6.1.1.2 Naming and declaration of software objects.

Software items are named with a mixture of upper and lower case characters to indicate their function, the underscore character "_" is now generally avoided if possible. A Class of object is a definition of a Type of entity, and one common naming convention is to preface the descriptive part of its name with a "T" (for Type) or cls (for Class).

An object class has a series of properties or Attributes, and it allows these to be accessed by providing functions or Methods. A sub-class or Derived Class can be created from a Base class, and will inherit the Attributes and Methods of that class. It can add to these features but can also modify (Override) them to implement its own specific behavioral characteristics. For example this statement

TVM20TD Device Inherits From **TVM20Device**

creates a new variant of an existing class.

An Object is created as an Instance of a Class, so for example the declaration

BSTD1 = new TVM20TDDevice

creates and initializes a new device called BSTD1 within the software that implements and manages a VM20 Touch Detector.

6.1.2 Software modelling and structural description

6.1.2.1 Universal Modelling Language

Representing and communicating the architecture design clearly is critical for development reviews, as well as to ensure it is implemented correctly. Unified Modeling Language (UML) is a formal but flexible design approach that represents three views of a system model. The functional requirements view (functional requirements of the system from the point of view of the user, including use cases); the static structural view (objects, attributes, relationships, and operations including class diagrams); and the dynamic behavior view (collaboration among objects and changes to the internal state of objects, including sequence, activity, and state diagrams).

6.1.2.2 Class Diagrams

A UML class diagram will describe sets of related object Classes together with any relevant links and groupings. Each class is represented as a table with three sections: the top is the class name, below this is a list of key properties, and the third group is a list of the methods of the class. Not all elements of the class are shown, it is simply a diagrammatic summary of the interface. The data types of class members will generally be shown, together with symbols denoting their visibility during programming. Keeping the internal workings of a class hidden or restrictive allows internal changes to be easily made, while preserving the integrity of a simplified and clear external interface.

Visibility	Symbol	Description
Public	+	Visible to whole project
Private	-	Available within the class only
Package	~	Available within a class grouping
Protected	#	Available to derived classes

Table 6.1 OOP / OOD Visibility notations for Class members and methods

Figure 6.1 shows an example of a bank database class with no Public data or properties, but a Public method that will indicate whether a selected user ID is valid or not (True / False).

clsBankDatabase
+AuthenticateUser() : Boolean

Figure 6.1 Class description example

N.B. For clarity throughout the rest of this document the visibility symbols will generally not be shown. It should be understood that most features of a class remain private or protected, and the public methods and attributes are those with commonized naming but customized (or overloaded) functionality.

6.2 Revised Control System implementation

The new 1300X control software features a structure based on the fundamental classes of control system elements identified by earlier projects and enhanced by further studies. In particular the OCI layer model of Statham [8] was referenced as this had been developed for the same basic computing platform (PC / Microsoft) and was a previous AMTReL project. Many of the other control projects (OSACA, OCEAN, OROCOS) used Unix-based real-time operating systems and software [17]. The specific implementation of the Statham OCI was not directly modified or adapted, since it used different programming environments and languages (Borland Delphi and Microsoft C++) instead of the more modern Microsoft .Net / Visual Basic / C# development framework. Furthermore the overall purpose and concept of the earlier OCI was different from but complementary to the new control system, and so all new software was developed with some adaptation of the previous IGA routines.

Key controller features implemented as base and derived Object classes include:

- Operator HMI - panels, buttons, DROs, LEDs, LEDBars, Graphs, Data Entry (Keypad)
- Machine operations – motion control, axis control, scheduling, machine safety logic
- Cycle operations – programming and execution, process monitoring
- Equipment interfacing – Digital and Analog IO, Serial and Bus / Network communications
- File and disk operations –Program storage, Configuration storage, Process data logging

6.2.1 Control HMI Screens

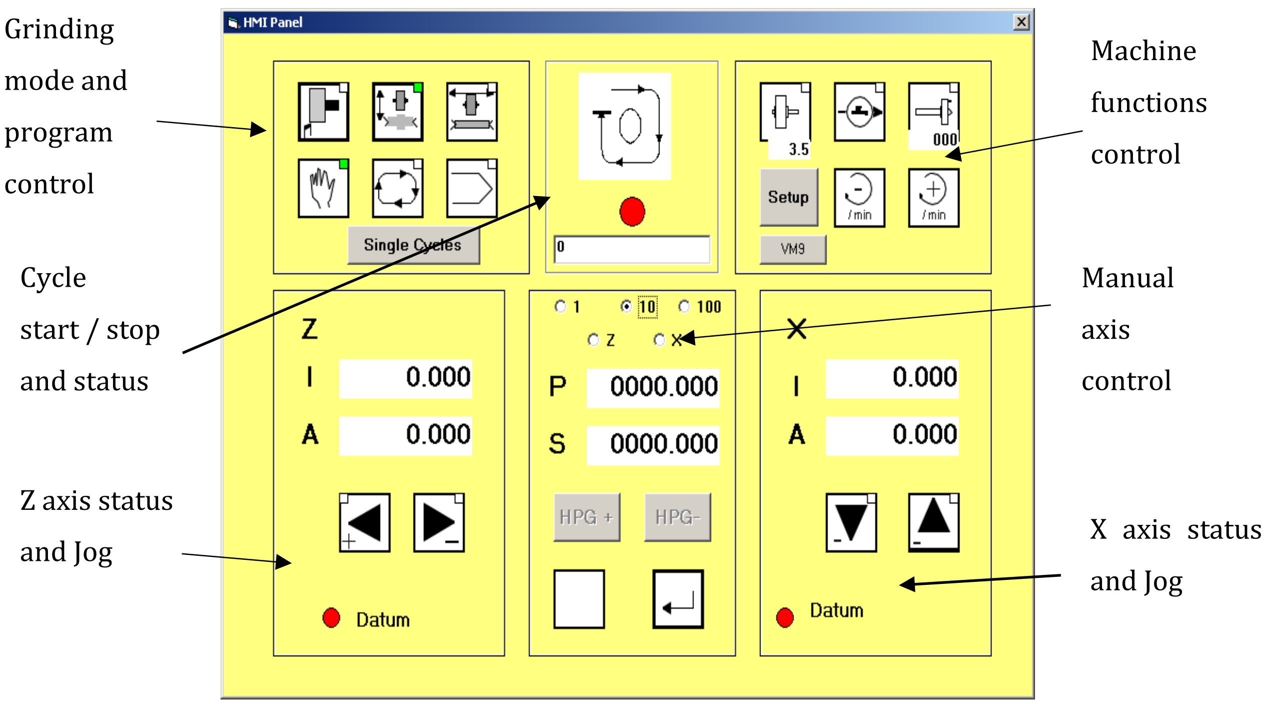


Figure 6.2 Main HMI screen and functions

These screens provide the main machine operator screen as well as extra screens for program editing, machine setup and auxiliary equipment access. The key machining operations are monitored and managed from here, via links to the internal modules of the control. In addition auxiliary screens to manage external devices (such as a VM9 unit) can be selected. Figure 6.2 shows the main screen of the 1300X controller application. The key display screen types and their component functions are summarized in Figure 6.3 below: these functions are in turn implemented by various object classes such as TDRODevice, and will use internal data structures provides as TMachineStatus, TCycleStatus, etc.

TMainScreen
X Position DRO
Z Position DRO
Screen selection
Machine Status

TCycleControlPanel
Cycle Start / Stop
Wheel On / Off
Workhead On / Off
Workpeed Up / Down
Coolant On / Off
Cycle Status

ProgramSelectScreen
Navigate / select
Grinding Type Select

TCycleEditScreen
Cycle Parameters
Editing Keypad

MachineSetupScreen
Axis Jog +ve / -ve
Datum Keys

VM9Screen
Device Status
Device Parameters

Figure 6.3 Display screen components

6.2.2 1300X operator panel

This has most of the features of the software panel implemented in traditional hardware, and it is treated as an external device connected via a serial (RS232) version of a TCommsLink object with its own transmission protocol. This feature implements functions such as InitLink, OpenLink, ReadRXData and DecodeRXData , and other routines format the panel's DisplayData. The panel retains some extra function keys and switches that enhance operator control of the grinding cycle, such as Feed Rate Override (a TFRODevice object).

6.2.3 Motion and Axis control

Low-level axis control is implemented by the Deva 004 card, and is managed as an instance of the class TAxisDevice. Above this are defined an X and Z axis of class TMachineAxis, and these objects provide the functions and data used by the higher level control functions to drive the axis hardware. An MPG device can be connected to an axis to allow an external demand signal to allow the operator to manually command axis movements.

The motion control object is responsible for the generation and management of the axis motion operations and profiles associated with a commanded axis move.

Figure 6.4 shows the key elements of three classes within the motion and axis control package. Note the use of overriding to enable each class to implement its own "GetStatus" operation.

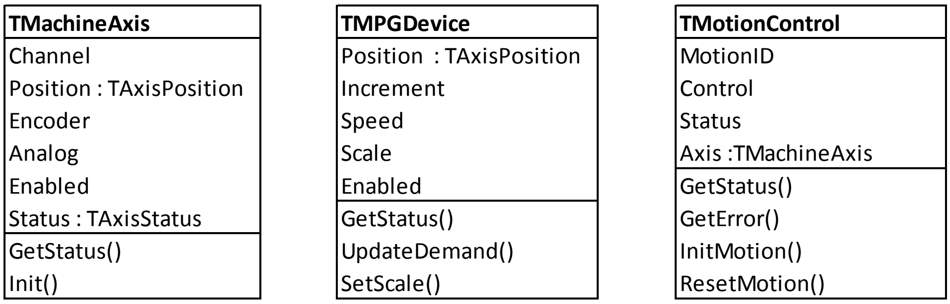


Figure 6.4 Class definitions for axis and motion control

6.2.4 Machine auxiliary functions

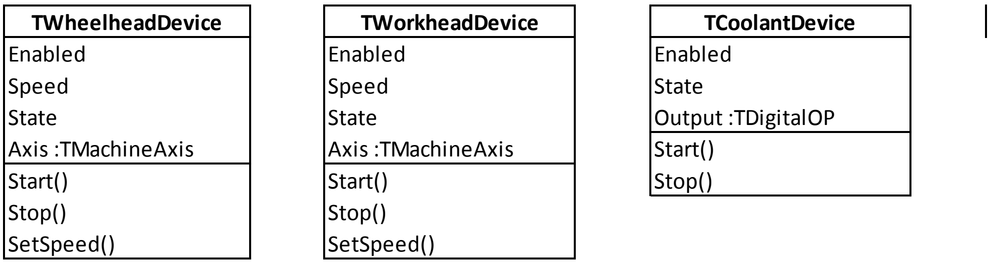


Figure 6.5 Class definitions for machine electromechanical devices

Figure 6.5 shows the key auxiliary functions of the machine implemented by the device classes. These items are able to be started or stopped, and to have their status monitored. The WheelSpindle and Workhead devices have a Speed property that can be read and may be set via the motor control. The CoolantDevice is switched on or off via an output signal and relay.

6.2.5 Grinding cycle execution

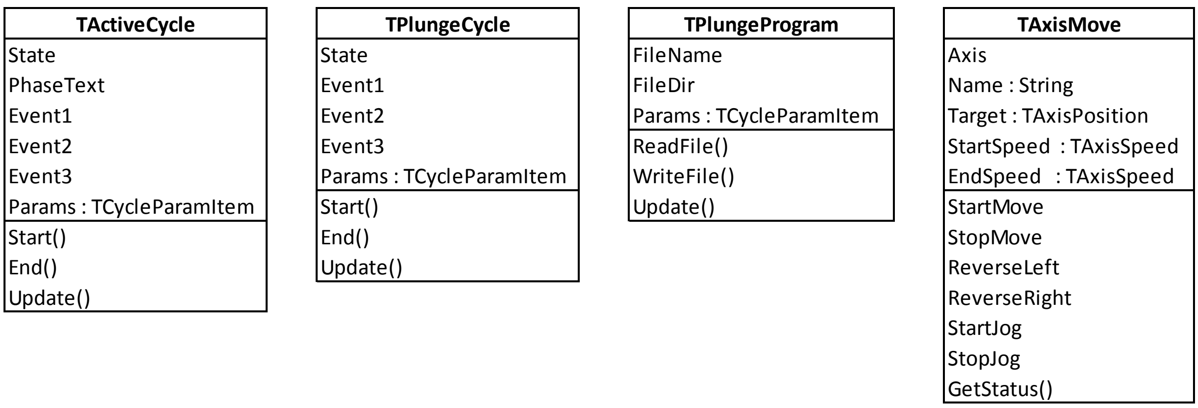


Figure 6.6 Class definitions for cycles, programs and moves

The TPlungeCycle, TTraverseCycle and TDressCycle classes derive from the base TGrindingCycle. The plunge cycle class can be developed into specific variations with extra stages or different behaviours, e.g. TAutoExtPlungeCycle, TAutoIntPlungeCycle, TManExtPlungeCycle.

The machine will execute plunge and traverse grinding cycles on cylindrical workpieces, and dressing cycles on the grinding wheel. These are performed either manually under operator control, or automatically as a programmed cycle. Most cycle parameters are stored as a part program which may be selected and edited, others may be stored as machine axis datum positions during setup operations.

As the cycle executes it will activate and monitor axis moves, and control other operations on machine hardware. It can respond to a set of machine events, such as process monitoring threshold signals or operator control inputs. A cycle is implemented as fixed set of phases or states, each state has actions performed on entry, and conditions which determine when it moves to the next appropriate state.

When a cycle is activated from the selected part program its parameters are converted from the external format to an internal format appropriate for actual machine operations. The Active cycle defines the current operation of the machine tool: its parameters can be altered from the original values set on cycle activation. It will contain more information than the basic cycle, for example the current cycle phase ID and description which can be communicated or displayed.

The active cycle is key to the implementation of Adaptive Control optimisations: it is here that the modified parameter values are applied as a result of the analysis of the current process data.

6.2.6 Grinding cycle programming and file operations

On entering Program mode a grinding cycle is selected from an existing program file or a default cycle, and presented to the operator for execution or editing. Three variants of the cycle entry screens are implemented, one for each grinding cycle type. Their key features are shown in Figure 6.7 below, note that common features are implemented by the parent class TCycleEditScreen:

TPlungeDataEntry
Plunge Parameters
Editing Keypad
File Load / Save

TTraverseDataEntry
Traverse Parameters
Editing Keypad
File Load /Save

TDressDataEntry
Dress Parameters
Editing Keypad
File Load /Save

Figure 6.7 Cycle editing components

The editing screen for a typical traverse grinding cycle is shown in Figure 6.8 below: the required parameter is selected and the appropriate value is entered via the numeric keypad. At the end of the edit the changes can be saved or abandoned, and another program file selected via the “Pick” button.

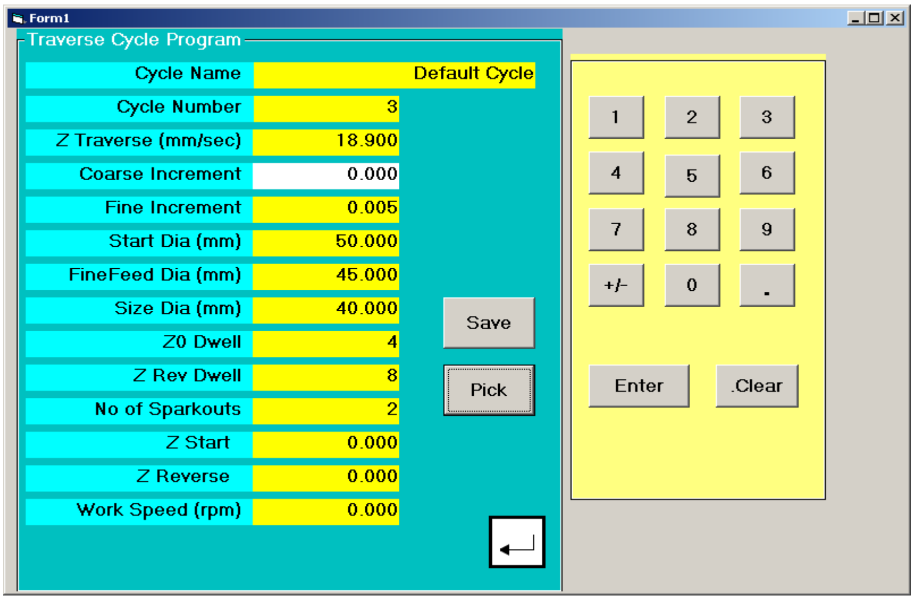


Figure 6.8 Traverse cycle parameter editing

6.2.7 External Device Interfacing

In order to connect to auxiliary process control devices it was necessary to implement a selection of established communications interfaces within the control. These comprise Digital and Analog IO, Serial and Bus / Network communications hardware and protocols. The key interface classes initially identified are shown in Figure 6.8 below: they have varying degrees of similarity and commonality within the main groupings but are broadly comparable in concept:

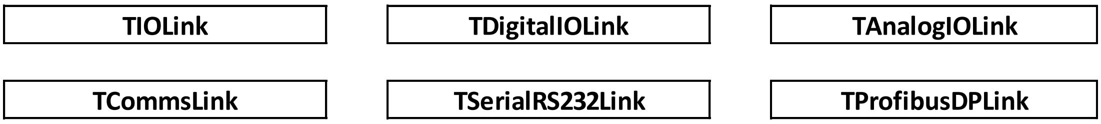


Figure 6.9 Key device communications classes

6.3 Process Control Device Implementation

6.3.1 Features and data of process monitoring devices

The Balance Systems VM9TD and VM20TD Touch Detector units were investigated and implemented: this allowed the study of both a simple system and also a more complex system with enhanced control and access features already built in. The two different philosophies and functionalities enabled the identification and rationalization of common features and attributes, and allowed the expansion of the software design to handle other device types.

As mentioned previously, some data is accessed by the main application Cyclically, in other words asynchronously and at regular intervals. Other data is accessed Acyclically, i.e. at required intervals and synchronised with specific application activities. Most device functions are performed Acyclically, in response to operator or control requests. Signals are read Cyclically.

The principal data types and functions of a typical Process Control device were summarized in Section 5.4.1, they are further detailed below.

6.3.1.1 Configuration Data

This contains low-level device setup information, and is accessed when the system is initially configured for its operation. A communications link might have the channel and protocol defined, or a device might have a sensor input defined as a particular type and model. These can be classed as "Option" or "Config" parameters, and require a certain authority to change them.

6.3.1.2 Parameter data

This is more dynamic setup information, and relates to the performance or behavior of specific features of the equipment, for example a gain setting or trigger level on an acoustic emission input channel. These are often known as "Setup" and "Work" parameters, and may usually be changed by the operator as required.

Note that Configuration data and Parameter data are variations of the same type (i.e. parameter); the main difference is when and how they are accessed.

6.3.1.3 Command data

Commands sent to the device can activate device features (like changing the operating mode) or initiate actions (such as a cycle start or reset). A command may result in a response from the device, such as the transmission of parameter data.

6.3.1.4 Signal data

This cyclic data is transmitted and analyzed constantly, and reflects the instantaneous state of the device's physical inputs and outputs (limits, selections, etc.). This is high priority data, and may require an immediate response by the machine tool control. For this reason it may come via the Digital IO wiring, and will need the software and operating system to respond rapidly.

6.3.1.5 Status data

This data will generally report device events and indicate any faults, errors or other useful information, such as confirmation of a requested function selection or activation.

It can be seen that Signal and Status data can both be considered as device status data.

6.3.1.6 Monitor data

Monitoring data provides processed device signal values and is read via a data transfer protocol. It is relatively low priority data, accessed cyclically for the purposes of display, logging and analysis. The ability to read and use this data significantly enhances the system's capabilities.

6.3.2 Levels of software functionality

6.3.2.1 HMI level functionality

The Human Machine Interface gives the top level access to the device, it provides the features the user needs to configure, monitor and operate the device. The HMI is organized into various screens and functional panels, and features components such as control buttons, lamp indicators, numeric and text displays, an editor and keypad, meters, graphs and data logging facilities. Any file and disk operations are managed by the HMI.

6.3.2.2 Virtual device level functionality

This software level maintains internal representations of all the device data (Monitor, Status, Parameter etc.) and implements the various device functions via command routines. The Virtual device provides everything needed by the HMI layer in an easily accessible, standardized format.

6.3.2.3 Hardware device level functionality

This is the physical implementation of the actual equipment, and it responds to the specifically designed software routines of the virtual layer. It is considered to include the lowest level driver software necessary to implement the communications between system elements.

6.3.3 VM9TD Software design

A stand-alone VM9TD unit is shown in Figure 4.3, Section 4.3.3 of this document. The hardware HMI panel comprising the front of the device shows the various operator features, including selector buttons, status LED lamps and “progress bars”. Their functionality is replicated or modified by the software implementation, and extra features are added.

6.3.3.1 HMI Features

The HMI level handles the functional components for the *Display* features of the VM9TD software device. Logical groups of display components are implemented on separate sub-panels. A Graph will display a live trace of the signal values from a defined DataSource, and these values can be saved to a named log file that can be analysed later to review the machining operation. Figure 6.10 below lists a selection of typical display panels and presents the graphing and logging features.

The HMI will have its own customised version of the device’s full Status data, in order to replicate the functionality and feel of the original hardware HMI panel. This data will include local versions of the actual device parameters which are modified and saved during setup and editing operations. Some of these features are illustrated in figure 6.11 below.

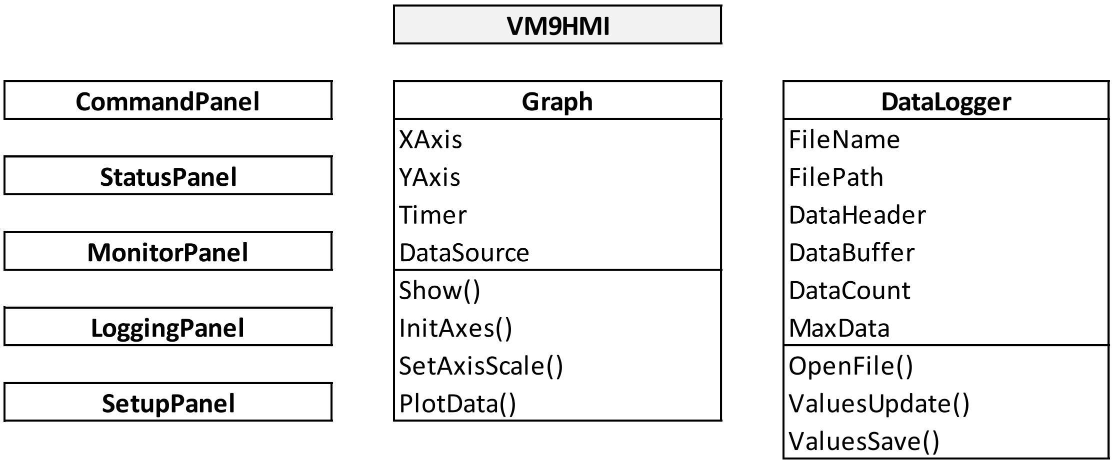


Figure 6.10 Process control device HMI data analysis features

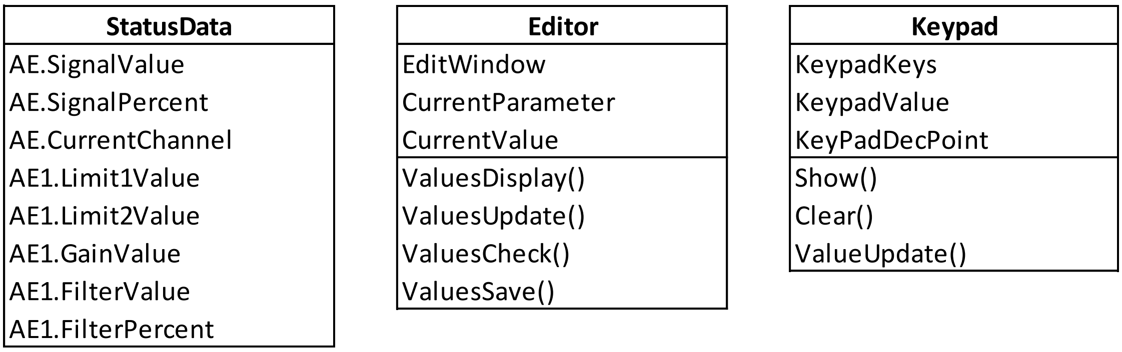


Figure 6.11 Process control device HMI parameters and editing features

6.3.3.2 VM9TD device features

The Device level contains the functional components for the *Control* features of the VM9TD software device, as well as local representations of the actual device's setup parameters, status data and signal / monitoring data. These are shown in figure 6.12 below.

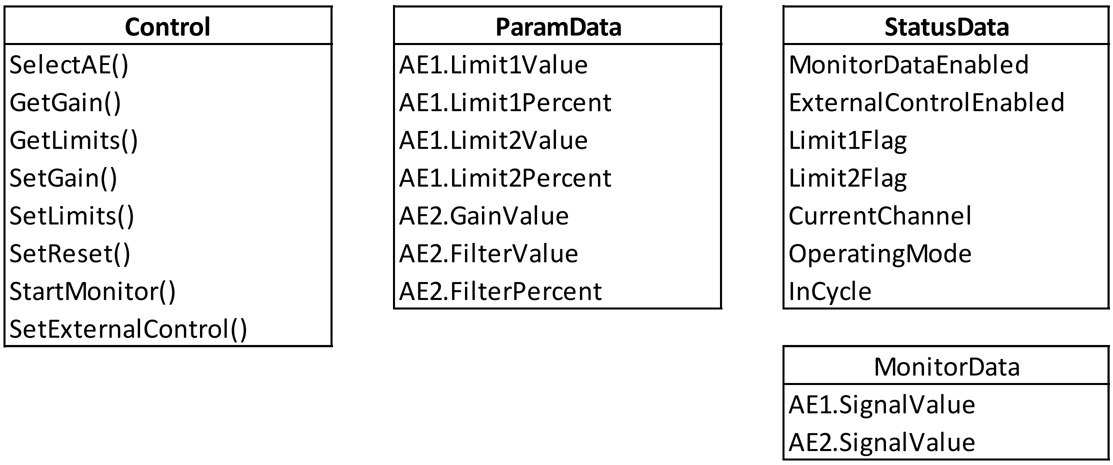


Figure 6.12 Process Control Device Control level features

6.3.3.3 VM9TD device Communications

The *Comms* features of the virtual device implement the conventional Digital IO link as well as a data communications link, which for this unit is an RS232 serial link. It can be seen from Figure 6.13 that these items have a similar structure to other "Devices", i.e. ParamData for Setup, etc.

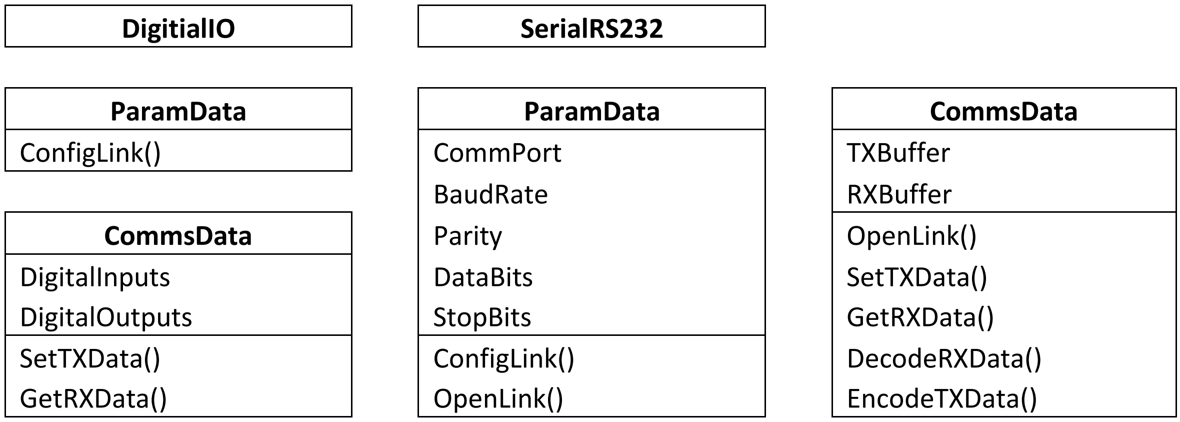


Figure 6.13 Process Control Device communications features

6.3.4 VM20TD Software design

The Balance System modular VM20 system features a main "System" rack housing dedicated units for the Balancing (BA), Touch detection (TD) and Gauging (GA) functions: Figure 4.2 in Section 4.3.2 shows a photograph of the physical unit. It has a more sophisticated and capable design than the VM9, consequently the software design was extended and enhanced to incorporate its extra features such as data monitoring and parameter modification via Profibus. The design was partially adapted from the earlier IGA application that had accessed certain VM20 features using relatively simple commands and data transfers.

6.3.4.1 VM20 system hierarchy

The features of the VM20 system were studied and categorised, and the Object / Class hierarchy in figure 6.14 below was devised to implement the different devices in a consistent and harmonised way.

The fundamental VM20 class is TVM20_Device, and it contains several data structures (e,g, StatusData, ParamData) derived from component VM20 classes (TVM20_Status, TVM20_Params). These are accessed by the device routines GetStatus, SetParams, etc.

From the generic class TVM20_Device, specific variants are derived to implement the BA, TD and GA device functions. In addition a VM20 "System" device class TVM20_SYS was developed by the author, which implements the overall management of the VM20 unit. The configuration settings for the VM20 unit will define which components are actually installed and operational. Figure 6.15 below shows the device classes that make up a VM20 system.

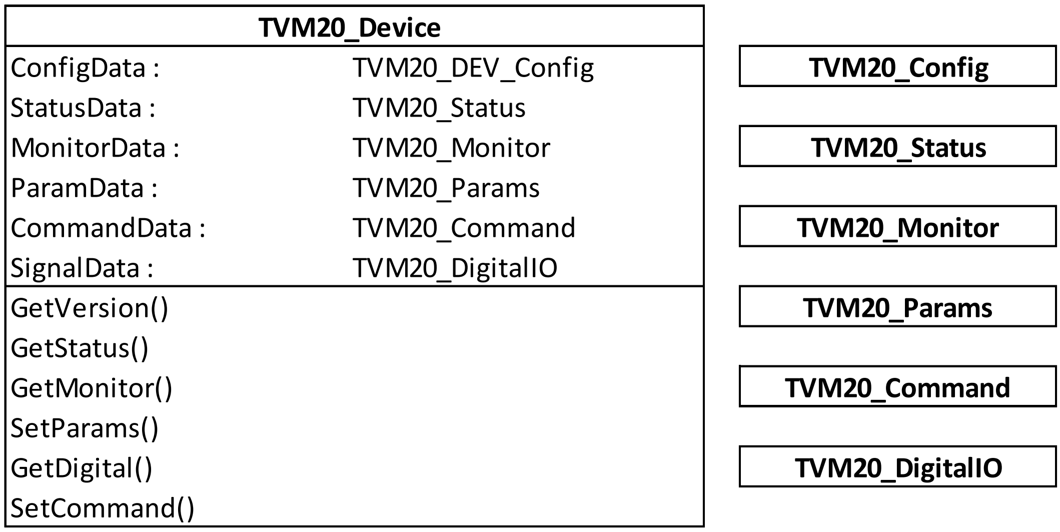


Figure 6.14 VM20 device component elements

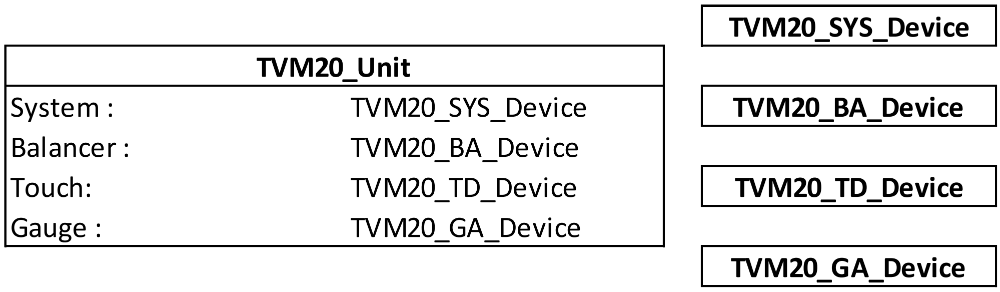


Figure 6.15 VM20 rack unit functional composition

6.3.4.2 HMI Features

Separate screens or pages are provided for each device type, the operator scrolls through to access the required function. The same basic HMI elements as found in the VM9 HMI are provided, but reformatted to reflect the operating characteristics of the original VM20 operator screens.

6.3.4.3 Connecting via Profibus

The method for transmitting and receiving commands and data via Profibus is much more complicated than via serial transmission, since communication occurs via a rigid buffer structure defined according to the particular equipment configuration. Furthermore, the reading and writing of commands and data to specific sections of the internal memory of the main VM20 system is done via a hierarchy of data pages and offsets. This means that a generalized and formalized access structure must be defined for interacting with the elements of the VM20. In other respects the basic handling of communications by Profibus is similar to e.g. Serial communications, with respect to configuration, initialization, buffer reading and writing, etc.

6.3.4.4 Status and monitoring data

To define the data structures needed to hold current VM20TD information it was necessary to develop specific Status and Monitor data classes for the Touch Detector, these are however comprised of generic VM20 classes representing a StatusItem, FlagItem, DataItem etc. as shown in Figure 6.16 below:

TVM20_TD_Status	
FlagByte1 :	TVM20_FlagItem
TCHLimit :	TVM20_StatusItem
BRNLimit :	TVM20_StatusItem
ALMLimit :	TVM20_StatusItem
Automatic :	TVM20_StatusItem
PP_Selected :	TVM20_DataItem
PP_SelectValid :	TVM20_StatusItem

TVM20_StatusItem	
BitMask :	Byte
Valid :	Boolean
Value :	Boolean
NameText :	String (12)

TVM20_FlagItem	
Value :	Byte
Address :	Integer

TVM20_TD_Monitor	
Enabled :	Boolean
Status :	Byte
Address :	Integer
Count :	Byte
Configuration :	Byte
AE1Signal :	TVM20_DataItem
AE2Signal :	TVM20_DataItem
PWR1Signal :	TVM20_DataItem
PWR2Signal :	TVM20_DataItem

TVM20_DataItem	
Valid :	Boolean
Value :	Double
NameText :	String (12)

Figure 6.16 VM20TD status and monitoring data

6.3.4.5 Configuration of parameters

A set of specific parameters are defined for the different sensor channels of the VM20TD device, these are of type TVM20_ParamItem which consists of a number of fields that fully define the parameters for the purposes of reading, writing, editing and displaying.

TVM20_TD_Config	
AE1_FullScale :	TVM20_ParamItem
AE1_Filter :	TVM20_ParamItem
AE1_Gain :	TVM20_ParamItem
AE2_FullScale :	TVM20_ParamItem
AE2_Filter :	TVM20_ParamItem
AE2_Gain :	TVM20_ParamItem
PWR1_FullScale :	TVM20_ParamItem
PWR1_Filter :	TVM20_ParamItem
PWR2_FullScale :	TVM20_ParamItem
PWR2_Filter :	TVM20_ParamItem

TVM20_ParamItem	
Value :	Double
MinValue :	Double
MaxValue :	Double
Section :	Integer
Offset :	Integer
Length :	Byte
Access :	Byte
NameText :	String
DataType :	Byte

Figure 6.17 VM20TD configuration and parameters

Figure 6.17 shows the Setup level parameters for the TD device. The Working parameters (trigger thresholds, logic formulae, etc.) are held in as "Part Programs" and are managed separately.

6.4 Intelligent Grinding Monitor (IGM)

The IGM specified as a stand-alone grinding process monitoring and analysis application, and is a derived from the IGA (Intelligent Grinding Assistant) concept. It will interface to different monitoring devices and read their data values and status signals, while optionally connecting to the grinding machine's CNC to synchronise the process data with the phases of the grinding cycle, such as rapid approach, coarse stock removal and sparkout.

Its chief function is process data acquisition and analysis, with live signal graphic display and data logging facilities. It is intended to be extended to include indications of grinding performance and behaviour by using the grinding process models previously developed at AMTReL and elsewhere.

6.5 Open Device Interface

The Open Device Interface concept is an extension of the Open Control Interface developed previously by Statham at JMU AMTReL. However instead of an Adaptive Control application seeking to interface with different CNC units and manufacturers, the perspective is reversed and a library of Device interfaces is defined which enables easier connection of an application (Control or otherwise) with a variety of comparable equipment types. It uses a similar layer model to provide the abstraction between the top-level software interface (or API) and the low-level hardware.

The existing OCI structure and components were reviewed and listed in a spreadsheet, and then a comparable list structure was used to summarize the elements and functions identified for the controller application. These were then formalized as a new UML design with a modern VB.Net implementation.

The basic ODI concept with example software classes and specific object instances is summarized in Figure 6.18 below; it is applicable to a machine control or a separate monitoring application. The devices may be internal machine or control elements, or external peripheral equipment.

Each new device added to the library would be derived from an existing device class, and comply with the ODI interfacing requirements. This would attempt to replicate the "Plug and Play" concept of PC peripheral equipment, whereby a variety of device types such as printers, scanners, cameras etc. of different makes and models can be easily connected and controlled by the host computer.

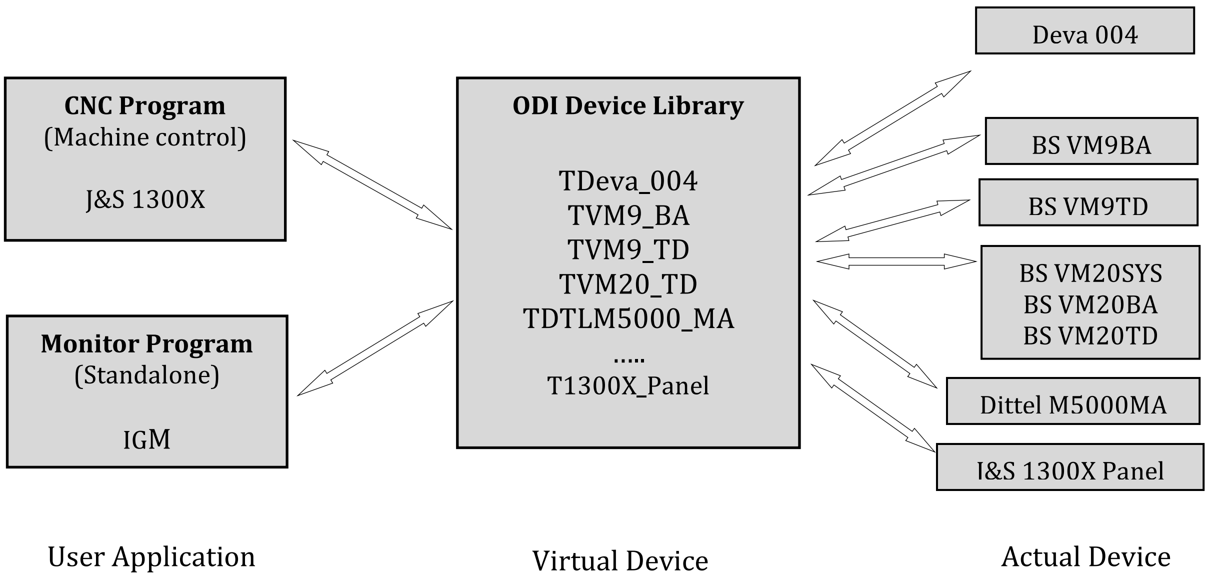


Figure 6.18 Open Device Interface application structure

7 Discussion and Conclusions

7.1 Critical evaluation

It was apparent from the studies of previous studies and developments in this field that no agreed or recognised frameworks, models or standards for a flexible, extendible Open-Architecture machine tool controller have been devised or published despite numerous previous industrial and academic collaborations. It does however appear that the various participants have incorporated the identified philosophies and structures into their latest controller designs, which do exhibit more distributed and interconnected architectures based on modern computer, software and communications technologies. Nevertheless there is still very little published information available regarding the formalised system descriptions and controller technology has remained largely proprietary.

During this research it was crucial to study and categorize the key features of examples of process control and monitoring equipment, and derive a framework whereby these can be treated as extension elements of a more capable, usable integrated control system. Significant progress has been made to produce a comprehensive design model using industry-standard UML to define the key elements, operations and interactions of such a system, and to exploit commonality with machine tool controls.

It was not feasible within the project time and scope constraints to fully implement a complete Object-Oriented application program. Instead, all of the relevant elements were separately analysed, implemented, tested and refined on the research hardware and then formalized iteratively into an Object Oriented UML design. This was then used to create a structured software framework in VB.Net to demonstrate an Open Device Interface library. Further effort and collaboration would allow the application software development to be completed and validated on the machine.

The PC / Windows platform has several advantages for a control system, chiefly the availability of affordable, capable hardware and comprehensive software libraries and tools. It does however have drawbacks in respect of real-time performance, but these can be managed with careful design.

As an alternative there are a number of Open-Source control applications and libraries available, based on Unix-derived operating systems. These can have benefits, chiefly in regard to performance.

7.2 Conclusions

A modular grinding controller with an Object Oriented framework has been developed to provide a flexible architecture with an improved capability to include enhanced and intelligent grinding cycles. This required the design of a common access strategy for a variety of device types both on the machine and also available externally for process monitoring and control. In addition the controller design was made extendable in order to implement different a range of grinding machine types with different operational and equipment features.

An Open Device Interface (ODI) to facilitate the integration of various grinding process control and monitoring devices has been designed and demonstrated. Equipment is treated as a generalized Device Object, implemented using layers of hardware and software abstraction, and a normalized user HMI philosophy allows developers and operators to work with different external devices more easily for parameter and data access. It was determined that many existing system components had similar characteristics to this auxiliary equipment, and they could be specified and managed in the same way.

Two types of Touch Detector unit have been interfaced to the new machine control software and a separate monitoring program to prove the concept. The ODI library can now be extended to the control of complementary device types such as the wheel balancing and size gauging modules of the Balance Systems VM20 and VM9 ranges, and also other manufacturer's products.

Finally the management of machining cycles and parameters has been implemented in such a way as to allow their alteration and optimization by Adaptive Control techniques.

8 Recommendations for future work

To prove the ease of adding new grinding capabilities to the current 1300X universal grinder, the facility to perform internal grinding cycles (Plunge and Traverse) should be implemented. Further to this the addition of linked (multiple sequential) cycle execution would be a significant enhancement.

The 1300X controller design has been made extendable in order to implement different a range of grinding machine types with different operational and equipment features. The next logical machine type to use this controller would be a surface grinder, followed by a centerless grinder. The philosophy of treating system features such as motors or operator panels as devices with similar attributes but different characteristics greatly simplifies this.

As regards the further development and proving of the Open Device Interface concept, a larger collaborative project involving a commercial control manufacturer and a monitoring equipment supplier would be required. Two obvious candidates would be the Italian companies OSAI and Balance Systems, these have a record of involvement in innovative research programs in this field and have worked with LJMU AMTReL in the past. In addition the participation of the University's Computing department would be beneficial, in order to ensure correct and efficient software development and identify improvements in the software structure and naming conventions.

The IGM Intelligent Monitoring Program should be further developed to access and modify the device parameters, and interact fully with the VM20 Wheel Balancer and Gauging modules, as well as the Fanuc CNC unit on the AMTReL Ultramat universal grinding machine. This application has great potential as a flexible, portable tool for grinding process analysis, and would be easier to complete than an actual machine tool control system. Data logging features should also be made compatible with National Instruments TDMS format for clarity and compatibility.

References

[1] Modern Machine Shop, 2001, 'Universal grinder takes the heat out of brake servo production', Magazine article, Case Study 7/15/2000

[2] MALKIN, S, 1989, *'Grinding Technology – Theory and applications of machining with abrasives'*, Society of Manufacturing Engineers, ISBN 0-87263-480-9, University of Massachusetts

[3] ROWE, W.B., 2009, *'Principles of Modern Grinding Technology'*, William Andrew, ISBN 978-0-81-552018-4, Burlington Massachusetts

[4] MORGAN M.N., CAI, R, GUIDOTTI, A, ALLANSON D.A., MORUZZI J.L., ROWE W.B., 2007, *'Design and Implementation of an Intelligent Grinding Assistant System'*, International Journal of Abrasive Technology, Vol. 1/1, pp. 106-135

[5] CHOI, T., SHIN, Y. C., 2005, '*Generalized Intelligent Grinding Advisory System*' International Journal of Production Research, Vol. 45/8, pp. 1899–1932

[6] ALLANSON, D.R., 1995, PhD Thesis, '*Coping with the Effects of Compliance in the Adaptive Control of Grinding Processes'*, PhD Thesis, Liverpool John Moores University, UK

[7] TÖNSHOFF H.K., FRIEMUTH T., BECKER J. C., 2002, *'Process Monitoring in Grinding'*, Annals of the CIRP, Vol. 51/2/2002, pp. 551-571

[8] STATHAM, C. G., 1999, *'An Open CNC Interface for Intelligent Control of Grinding'*, PhD thesis, Liverpool John Moores University, UK

[9] KELLY, S. K., 1994, '*Intelligent control for the centreless grinding of compliant, thin walled cylinders*', PhD Thesis, Liverpool John Moores University, UK

[10] THOMAS, A., 1996, *'Adaptive Control System for Precision Cylindrical Grinding'*, PhD Thesis, Liverpool John Moores University, UK

[11] CHEN, Y. 1998, '*A Generic Intelligent Control System for Grinding',* PhD thesis, Liverpool John Moores University, UK

[12] SPERLING, W., LUTZ, P., 1997, 'Designing Applications for an OSACA Control', Proceedings of the International Mechanical Engineering Congress and Exposition, Dallas USA, November 2007

[13] FUJITA, S., YOSHIDA, T., 1996, *'OSE: Open System Environment for Controller'*, OSE Consortium report, Mitsubishi Electric Corporation, Japan

[14] BOCK, H., KREMER, M., LAMB, I., 2007, *'Deliverable D4.10: Description of DCRF compliant Open Control System'*, OCEAN Consortium report, Stuttgart Germany

[15] SMID P., 2003, '*CNC Programming Handbook: A comprehensive guide to practical CNC programming'*, Industrial Press, ISBN 0831131586 0-831-131586, New York

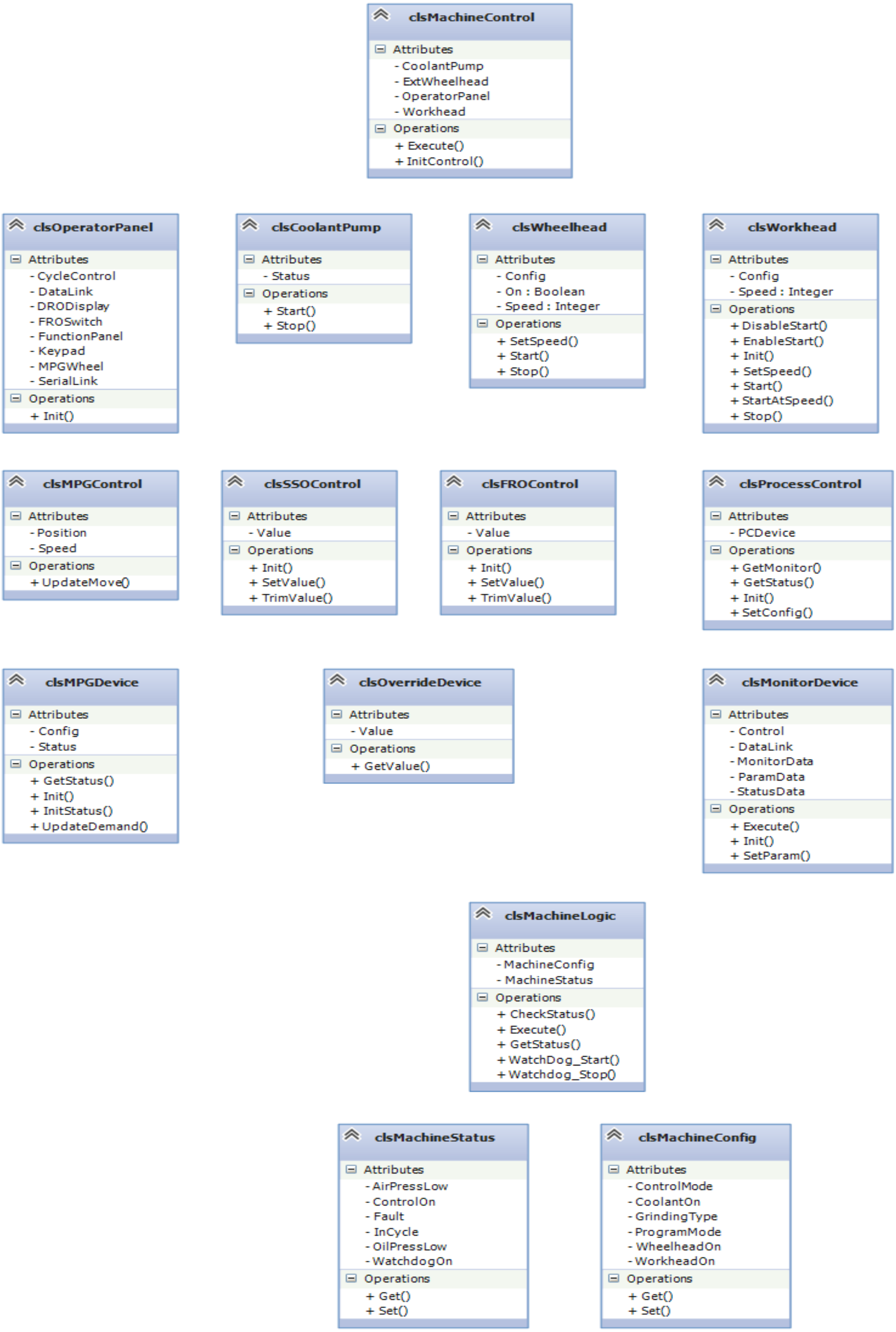
[16] MORGAN, M. N., MORUZZI, J.J., ALLANSON, D.R., CHEN, X, 2014, *'On the Development of Machine Controller Systems with Enhanced Facility for Process Monitoring Integration',* Key Engineering Materials Vol. 581, p533-538

[17] NACSA, J., 2001, *'Comparison of three different Open Architecture Controllers',* Research paper, Computer and Automation Research Institute, Hungarian Academy of Sciences

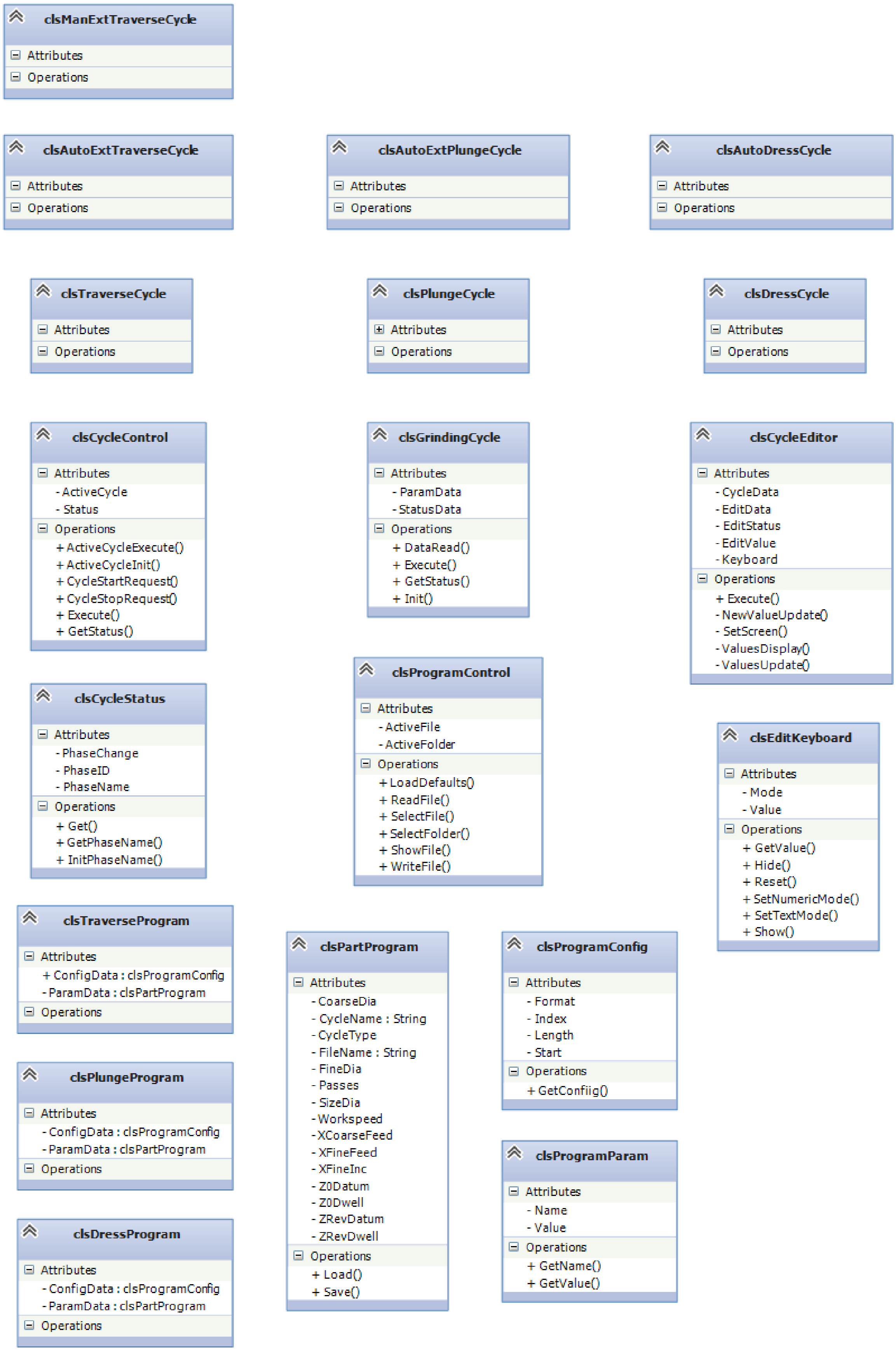
Appendices

Appendix 1 1300X System UML Diagrams

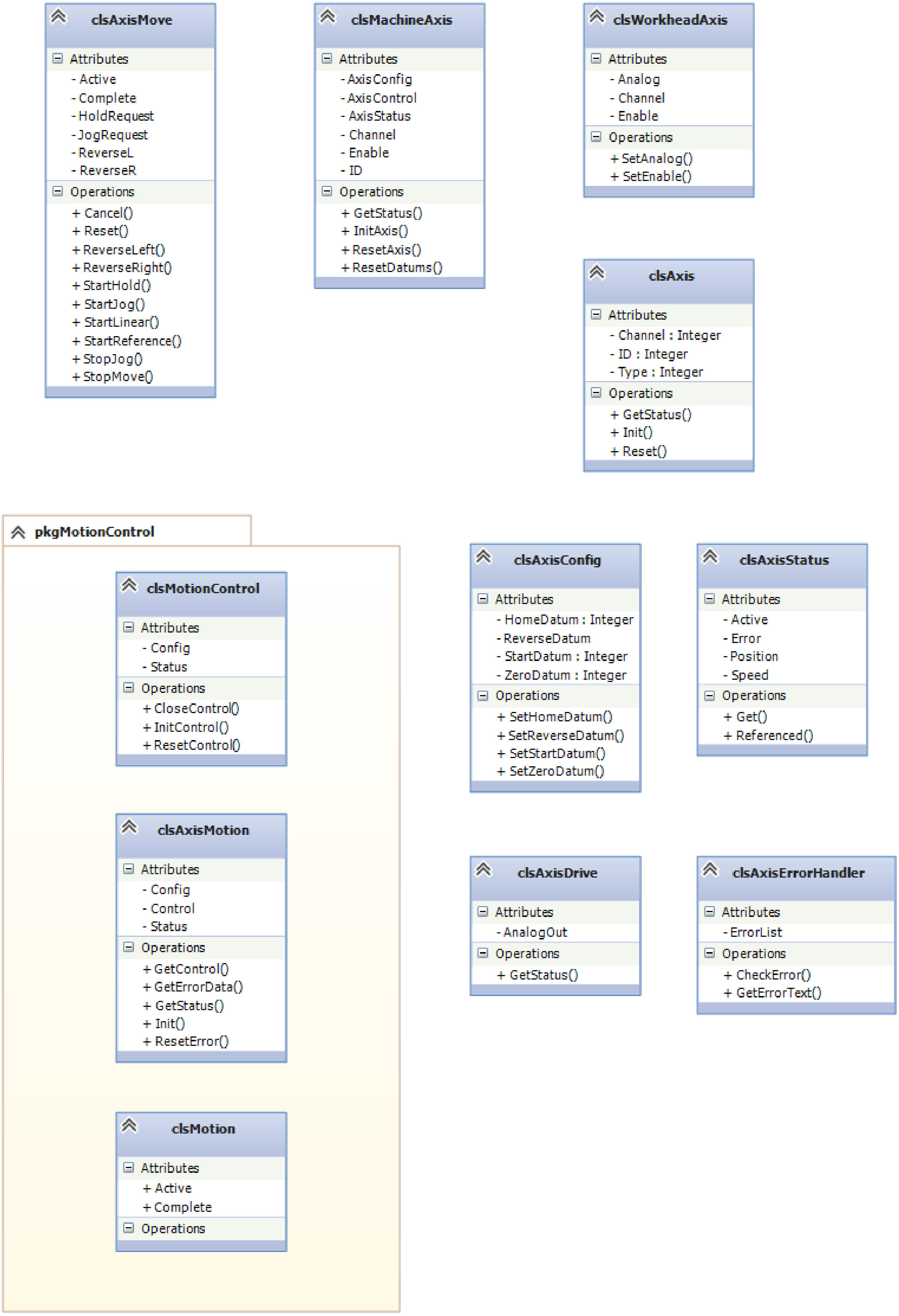
1 UML Class Diagram : Control1300X_MachineControl



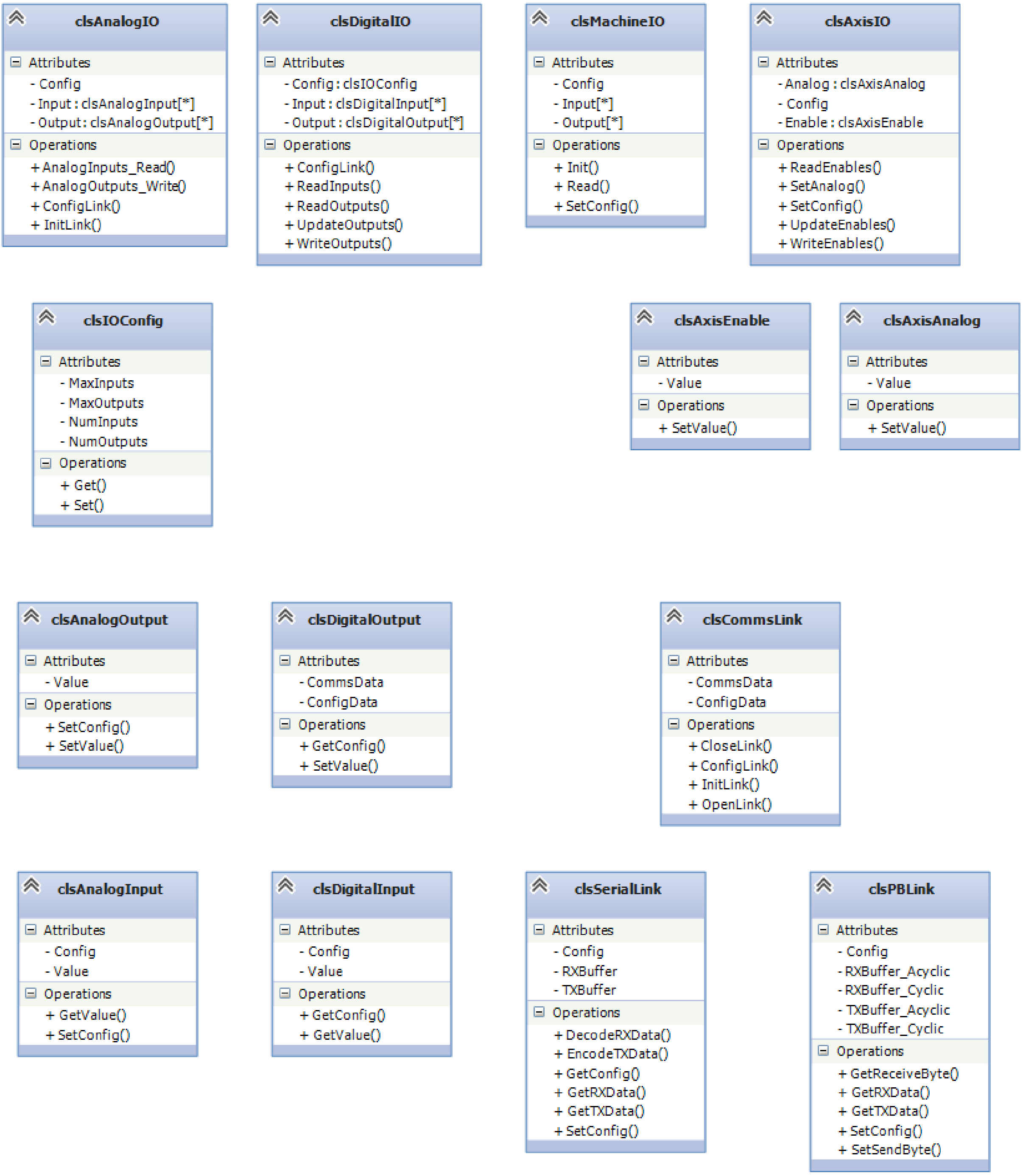
2 UML Class Diagram : Control1300X_MachineCycles



3 UML Class Diagram : Control1300X_MachineAxes



4 UML Class Diagram : Control1300X_MachineIF



Appendix 2 VM20 Profibus Communications Interface

1 Data structure of Profibus messages to and from VM20 Unit

BS VM20 **Profibus control and data** **1**

NCU => VM20

	TXData	*abSendData*			
Cyclic Commands	0	System			
	1	Balancer1			
	2	Balancer2			
	3	Touch1		e.g. Enable monitor	
	4	Gauge1	SecA		
	5		SecB		
	6	Gauge2	SecA		
	7		SecB		
Acyclic Data	8	Cmd code	255	&H32	&H33
	9	Card Addr			
	10	Param Offset			
	11	Num Bytes			
	12	Data0			
	13	Data1			
	14	Data2			
	15	Data3			
	16	Data4			
	17	Data5			
	18	Data6			
	19	Data7			
	12	Data8			
	13	Data9			
	14	Data10			
	15	Data11			
	16	Data12			
	17	Data13			
	18	Data14			
	19	Data15			

20	Data16
21	Data17
22	Data18
23	Data19
24	Data20
25	Data21
26	Data22
27	Data23
28	Data24
29	Data25
30	Data26
31	Data27
32	Data28
33	Data29
34	Data30
35	Data31

VM20 => NCU

	RXData	*abReceiveData*		
Cyclic Status	0	Balancer1		Status1
	1			Status2
	2	Balancer2		Status1
	3			Status2
	4	Touch1	SecA	Status1
	5		SecB	Status1
	6	Gauge1	SecA	Status1
	7			Status2
	8		SecB	Status1
	9			Status2
	10	Gauge2	SecA	Status1
	11			Status2
	12		SecB	Status1
	13			Status2
	14	Complex		
	15	System		

Acyclic command data (Output) **Configure block length**

MAX. N.PARAM. OF ACYCLIC OUTPUT (4/8/16/32)

Bytes	Nparam
4	2
8	4
16	8
32	16

BS VM20

Profibus control and data **2**

variable	21	P1	Data0
4 bytes	22		Data1
	23	P2	Data2
	24		Data3

VM20 => NCU

Acyclic Data	16		Echo Cmd
	17		Error
	18		Card Addr
	19		Param Addr
	20		Num Bytes
variable	21	P1	Data0
8 bytes	22		Data1
	23	P2	Data2
	24		Data3
	25	P3	Data4
	26		Data5
	27	P4	Data6
	28		Data7
Cyclic Monitor	29	Fixed	Flag (Data valid bits)
	30		ID 1 (Data formats)
	31		ID 2
	32	Unbalance	BA 1
	33		
	34	Wheelspeed	BA 2
	35		
	40	AE2	TD 1
	41		
	42	Power	TD 2
	43		
	44	Gauge A	GA 1
	45		
	46	Gauge B	GA 2
	47		

Acyclic receive data (Input) **Configure block length**

MAX. N.PARAM. OF ACYCLIC INPUT (4/8/16/32)

Bytes	Nparam
4	2
8	4
16	8
32	16

2 Data structure of “Monitor” message string from VM20 Devices

Monitor data format configuration

BALANCER 1 / 2 MONITOR	Config settings		BAL1 Byte	BAL1 Bits	BAL1 Code	BAL2 Byte	BAL2 Bits	BAL2 Code
OFF	no data	0	2	2,1,0	000	2	3,4,5	000
UNB+POS	unbalance and position	1			001			001
UNB+RPM	unbalance and rotation speed	2			010			010

TOUCH DET MONITOR	Config settings		TD1 Byte	Bits	Code	Byte	Bits	Code
OFF	no data	0	2	7,6	00	1	1,0	00
V1	vibration 1	1			01			00
V2	vibration 2	2			10			00
PA	section A power	3			11			00
PB	section B power	4			00			01
V1+V2	vibration 1 and 2	5			01			01
V1+PA	vibration 1 and section A power	6			10			01
V1+PB	vibration 1 and section B power	7			11			01
V2+PA	vibration 2 and section A power	8			00			10
V2+PB	vibration 2 and section B power	9			01			10
PA+PB	sections A and B power	10			10			10

GAUGE 1 / 2 MONITOR	Config settings	GA1 Byte	Bits	Code
OFF	no data	1	4,3,2	000
DIMA+DIMB	sections A and B dimension			001

3 Software configuration of Profibus interface to VM20 Device

```
// Module VM20code.bas

Private Sub VM20_Profibus_Connect()
    Call DP_OpenLink
End Sub

Private Sub VM20_Profibus_Disconnect()
  Call DP_CloseLink
End Sub

Public Sub VM20_OpenConnection()
' Connect and init device
  If VM20.system.simulation = False Then
    Call VM20_Profibus_Connect
  Else
    Call VM20_Profibus_SimConnect
  End If

  Call VM20_InitProfibus ' set addresses according to config
  Call VM20_InitMonitor ' set flags and monitor data
  Call VM20_InitStatus ' set flags and status data
End Sub

Private Sub VM20_InitMonitor()
'allocate addresses
' device data configuration
  VM20_PB_Monitor_Command = 0

' monitor data starts from 16 + 5 (fixed)
' plus 4 / 8 / 16 / 32 depending on Acyclic data
' we have 8 acyclic bytes

  If VM20.system.BA_Installed Then
  'VM20_Balancer_InitMonitor
  With VM20.Balancer
    .MonitorData.Configuration = 2 'UNB + RPM
    .MonitorData.status = 0
    .MonitorData.Address = VM20.Profibus.RX_Cyclic.Monitor_BA1_DataStart ' 32
    .MonitorData.count = 4 ' bytes = 2 data = DataLen
    .CommandData.FlagByte = 0
    .CommandData.Address = VM20.Profibus.TX_Cyclic.Command_BA1_DataStart ' 1
    'call Profibus_write_command(.CommandData.Address), .CommandData.FlagByte) 'pbSendData
  End With
  VM20_Balancer.MonitorData = VM20.Balancer.MonitorData 'local copy
  End If

  If VM20.system.TD_Installed Then
  'VM20_Touch_InitMonitor
  With VM20.Touch
    .MonitorData.Configuration = 8 'V2 + PWR
    .MonitorData.status = 0
    .MonitorData.count = 4 ' bytes = 2 data
    .MonitorData.Address = VM20.Profibus.RX_Cyclic.Monitor_TD1_DataStart '40 ' variable !!
```

```
      .CommandData.FlagByte = 0
      .CommandData.Address = VM20.Profibus.TX_Cyclic.Command_TD1_DataStart '3
      ' Profibus_write_command(.CommandData.Address, .CommandData.FlagByte) 'pbSendData

   End With
   VM20_Touch.MonitorData = VM20.Touch.MonitorData 'local copy
   End If

   If VM20.system.GA_Installed Then
   'VM20_Gauge_InitMonitor
   With VM20.Gauge
      .MonitorData.Configuration = 1 'Ga1 + Ga2
      .MonitorData.status = 0
      .MonitorData.Address = VM20.Profibus.RX_Cyclic.Monitor_GA1_DataStart '44 ' variable !!
      .MonitorData.count = 4 ' bytes = 2 data
      .CommandData.FlagByteA = 0
      .CommandData.AddressA = VM20.Profibus.TX_Cyclic.Command_GA1SECA_DataStart ' 4 ' and 5 =
secB
      'call Profibus_write_command(.CommandData.AddressA, .CommandData.FlagByteA) ) '
      .CommandData.FlagByteB = 0
      .CommandData.AddressB = 5 'secB VM20.Profibus.TX_Cyclic.Command_GA1SECB_DataStart
      'call Profibus_WriteCommand(.CommandData.AddressB) , .CommandData.FlagByteB)
   End With
   VM20_Gauge.MonitorData = VM20.Gauge.MonitorData 'local copy
   End If

End Sub 'VM20_InitMonitor

Public Sub VM20_InitStatus()
   If VM20.system.BA_Installed Then
      VM20_Balancer_InitStatus
   End If
   If VM20.system.TD_Installed Then
      VM20_Touch_InitStatus
   End If
   If VM20.system.GA_Installed Then
      VM20_Gauge_InitStatus
   End If
End Sub

Private Sub VM20_InitProfibus()
   Call Profibus_SetConfiguration(8)
End Sub
```

```
// Module Profibus.bas

Public Sub Profibus_SetConfiguration(DataCount As Byte)
  'VM20 TX data
  '  Cyclic Addresses
    With VM20.Profibus.TX_Cyclic
        .Command_System.DataStart = 0
        .Command_BA1.DataStart = 1
        .Command_BA2.DataStart = 2
        .Command_TD1.DataStart = 3
        .Command_GA1SECA.DataStart = 4
        .Command_GA1SECB.DataStart = 5
        .Command_GA2SECB.DataStart = 6
        .Command_GA2SECB.DataStart = 7
    End With
  '  ACyclic Data Addresses
    With VM20.Profibus.TX_Acyclic
        .Command.Code = 8
        .Command.DataStart = 9
        .Command.DataOffset = 10
        .Command.DataLen = 11
  ' Data OUT = 4 bytes / 8 bytes / 16 bytes / 32 bytes
        .Parameter.DataStart = 12
        .Parameter.DataLen = DataCount
    End With

  'VM20 RX data
  '  Acyclic data addresses
    With VM20.Profibus.RX_Acyclic
      .Command.Code = 16
      .Command.Error = 17
      .Command.DataStart = 18 '(StartAddress)
      .Command.DataOffset = 19 '(OffsetAddress)
      .Command.DataLen = 20 '(NumBytes)
      .Parameter.DataStart = 21 'StartAddress
      .Parameter.DataLen = datacount '8 ' => 28 ' NumBytes
    End With
  '  Cyclic data addresses
    With VM20.Profibus.RX_Cyclic
      .Monitor.Status.DataStart = VM20.Profibus.RX_Acyclic.Parameter.DataStart +
                                  VM20.Profibus.RX_Acyclic.Parameter.DataLen ' 29
      .Monitor.Config.DataStart = .Monitor.Status.DataStart + 1
      .Monitor.BA1.DataStart = .Monitor.Config_DataStart + 2 ' + DataLen
      .Monitor.BA2.DataStart = .Monitor.BA1.DataStart + 2 ' + DataLen
      .Monitor.TD1.DataStart = .Monitor.BA2.DataStart + 2 + 4 ' + DataLen
'      .Monitor.TD2.DataStart = .Monitor.TD.DataStart + 2 ' + DataLen
      .Monitor.GA1.DataStart = .Monitor.TD1.DataStart + 4 ' + DataLen
      .Monitor.GA2.DataStart = .Monitor.GA1.DataStart + 4 ' + DataLen
    End With
End Sub
```

Appendix 3 Conference Paper - Key Engineering Materials

Key Engineering Materials Vol. 581 (2014) pp 533-538
Online available since 2013/Oct/07 at www.scientific.net

doi:10.4028/www.scientific.net/KEM.581.533

On the Development of Machine Controller Systems with Enhanced Facility for Process Monitoring Integration

M N Morgan[1, a], J J Moruzzi[1, b], D R Allanson[2, c] and X Chen[1, d]

[1]General Engineering Research Institute,

[2]School of Engineering,
Liverpool John Moores University, Byrom Street, L3 3AF, UK

[a] m.n.morgan@ljmu.ac.uk, [b] j.j.moruzzi@2010.ljmu.ac.uk, [c] d.r.allanson@ljmu.ac.uk,
[d] x.chen@ljmu.ac.uk

Keywords: Grinding, Machine Controller, Process monitoring, Open-Architecture

Abstract. This paper reports on the progress achieved with the specification and development of a flexible, innovative open-architecture grinding machine control system. The aim is to unify the design and implementation of key machine tool features such as hardware configuration parameters, operational parameters, process variables and machining cycles into a rationalized, extendable, object-oriented framework. The application is PC/Windows based and developed in Visual Studio .NET software. The control design is extendable to surface as well as cylindrical grinding operations and cycles. It was intended from the outset to allow simpler integration of 3rd party sensor-based process monitoring equipment to facilitate optimised machining, including Adaptive or Intelligent control techniques. These peripheral devices from different suppliers must be analysed and classified depending on specific functionality, and a common programming and operating interface presented to the developer and user. Other device types on the machine such as operator panels and motion /axis control hardware have been identified and incorporated. An Open Device Interface (ODI) framework is proposed and defined: the development of this generic software structure and associated libraries masks specific underlying details of a device's implementation and allows simpler incorporation of additional features to the machine. The requirements for the transmission, handling, display and logging of process data is presented, and the successful demonstration of the new system on a universal grinding machine with sample peripheral devices is reported. The applicability of the Open Device Interface structure to other generalised control devices and monitoring applications is also discussed.

1. Requirements of the problem

A requirement arose to implement a modernised version of the basic control system applied to a SAMM (Servo Assisted Manual Machining) Cylindrical grinding machine, namely the Jones & Shipman 1300X from the 1990s. The original control was a simple PC-based unit (running MS-DOS) that controlled two axes via servomotors and drives, and allowed the operator to run the machine manually (via MPG handwheels on a front panel) or to program various simple automatic machining cycles and operations. This is an intermediate level of machine control, between a simple manual machine and a full CNC (Computerised Numeric Control).

It was decided to implement the new control with current industrial PC hardware and using modern features such as a touchscreen panel to improve usability and functionality. In addition it was desired to implement a modern, extendable, Object-Oriented software framework in order to allow the implementation of enhanced machining cycles and process monitoring, as well as the possible control of other grinding machine types, such as surface or centreless grinders. The software would run on Microsoft Windows XP or CE, and be implemented in Visual Basic.NET for portability.

In grinding it is often desirable to have a control system that can utilise external Process Control and Monitoring equipment, and perform optimised production cycles either using manual (operator) or automatic (CNC) intervention. Provided functions include wheel balancing, in-process gauging, and touch detection (power monitoring and acoustic emission). For optimised machining the equipment operation should be tightly linked to the machine control.

Process monitoring may be included in higher-end grinding machines in with specific requirements, however this involves significant development and customisation from the machine builder. In addition the equipment may have different operational features and characteristics, which can make it less usable to operators. It would therefore be highly beneficial to allow for the use of process monitoring equipment as part of the fundamental control design.

If the equipment could be treated as a Generic device configured as part of the machine and with a standardized interface, it would simplify its integration and operation considerably. It can further be seen that other elements of the control system could also be treated as a device, for example the operator control panel and the Axis or Motion control features.

A Generic Device interfacing strategy would address the following issues:

- Lack of standardisation between CNC and other equipment manufacturers.
- Different interfacing hardware and strategies for Process Control equipment.
- Different levels of functionality / complexity among equipment.
- Unexploited similarities between other equipment operations and specification.

2. Features of grinding machine control

2.1. Common grinding machine types

Three principal types of grinding machine are generally considered, depending on their layout and the type of machining operation performed; Cylindrical, Surface and Centreless. In all cases the basic operation consists of moving an abrasive wheel into contact with a workpiece, and moving the workpiece relative to the wheel. These actions may be continuous or performed in stages, and with different speeds [1].

This research relates principally to Cylindrical grinding, which can be implemented as External or Internal and has 3 main cycle types, namely Plunge Grinding, Traverse Grinding and Wheel Dressing. These operations can be implemented as enhanced variants if external monitoring is used, e.g. Plunge Grind with Touch Detection, Plunge Grind with Diameter Gauging, etc. [1]

2.2. Cycle programming and execution

A grinding cycle is defined as a sequence of axis movements together with wheel and workpiece actions. Additional machine features such as coolant, hydraulics and pneumatics must be managed manually or automatically. Grinding cycles are often predefined and executed by means of a defined G-code (e.g. G88 Plunge Grind), together with variable, programmable parameters that determine positions, speeds, etc. Alternatively a cycle may be “taught” by the operator, who moves to appropriate positions on the axes and saves the locations, along with any relevant numeric values.

Once a cycle has been programmed it is stored and can be executed automatically by pressing the “Cycle Start” button.

Most machines can store several of these part programs. It is desirable to define, store and export the part programs in a standardised data format so that other applications can access them for the purposes of analysis or optimization. Ideally the program filenames should identify the part ID, grinding type (P, T or D), and the cycle number if part of a sequence of linked cycles (a “canned cycle”).

2.3. Motion and axis control

The machine has motion and axis control features, implemented by a commercial PC card with associated software libraries. Axis management is done by activating servomotor drives, and monitoring encoder inputs for position and speed feedback, as well as operator handwheel demands. From this the motion control partition commands smoothed and synchronized axis movements in response to programmed requests from the cycle management routines.

The axis management has various setup and working parameters and data associated with it, as well as actions, and it can therefore be viewed as a device or object with associated methods and properties.

2.4. Interfacing with peripheral equipment

The principal methods of machine tool equipment interconnection are Digital I/O, Serial transmission (RS232/RS422/RS485), industrial Bus (Fanuc, Profibus, Modbus…) and Ethernet, as shown in Fig. 1:
System devices are interconnected to transmit and exchange:

- Control signals
- Status signals
- Process data
- Configuration data

It can be seen that it would be beneficial to handle the transfer of data and signals between devices in a standardised way, independent of the connection and transport method used for the actual communications.

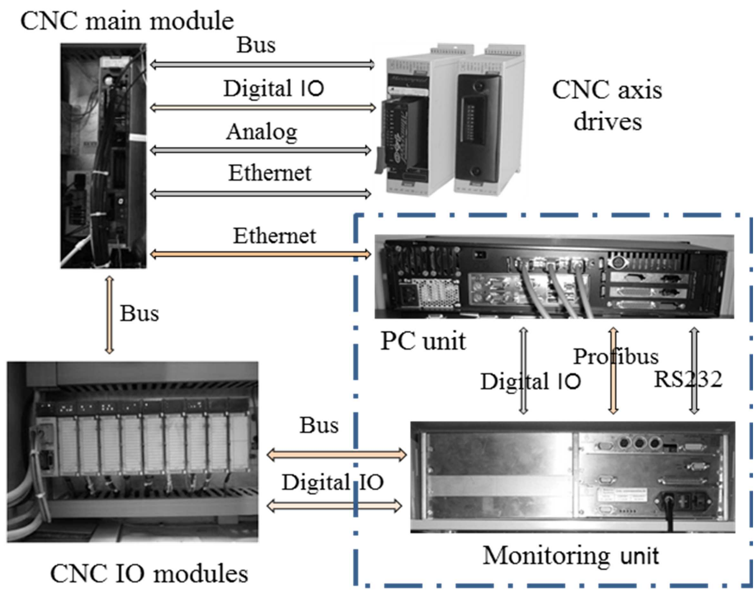


Figure 1: Communication interfaces between system equipment

3. Features of grinding process control

Grinding process monitoring equipment is typically a stand-alone unit connected to the machine control via a wiring or communications interface. The operator will set the device operating parameters and monitor its behaviour via a HMI panel. The unit will execute various monitoring cycles, and as certain programmed conditions are met during machining appropriate signals and visual indications are set by the device.

Initial studies had been done on the Balance Systems VM9-TD Touch Detector unit, with RS232 and Digital IO control. For a more comprehensive investigation and to further define the interfacing issues, the Balance Systems VM20 was also studied. This is a sophisticated modular system, which features Balancer (BA), Touch Detector (TD) and Gauge (GA) units. Full control of the VM20 via Profibus and Digital IO is provided [2].

3.1. Data and signal transmission between devices

The key data types and operations of a typical fully-featured device are:

- Config Data (Acyclic) Report device details
- Parameter Data (Acyclic) Read / Write setup info
- Command data (Cyclic) Turn features On /Off, Request data, …
- Status data (Cyclic) Report device events
- Monitor Data (Cyclic) Live device signal values
- Signal Data (Cyclic) Digital Inputs / Outputs (limits, resets, …)

Data operations are classed as *Cyclic* if performed regularly and *Acyclic* if performed intermittently. The data is accessed via assorted communication channels and protocols, and will be in varying formats and from different locations within a device. For example, numeric data could be stored as two bytes at a particular memory address, and a signal could be a bit setting within a byte. These should be collected and presented in the software using a standardised and transparent strategy.

3.2. Interaction between the machine control and peripheral devices

The tables below summarise the main features and operations to be managed:

Main control actions:

- Device Configuration
- Device Operation
- Device Monitoring
- Process Modification

Main process data:

- Control, status and alarm signals
- Process data values
- Device Parameters

Control actions	Device actions
Set Access level	Allow / deny operations
Select Automatic / Manual Select a program	Go to Automatic / Manual Change & report program
Change parameters Request parameters	Update parameters Return parameters
Set control commands Start / Stop a cycle	Respond to commands Execute a cycle
Monitor status signals Monitor process values Monitor errors / alarms	Return device status Return data values Report errors / alarms

Figure 2: System data and interactions

4. Object Oriented Design and Open Systems

4.1. Open Control Systems

The system philosophy builds on the Open Control Systems concept for machine tool controls, which originated in the 1990s in an attempt to define and harmonise the differing proprietary technologies that had evolved with different CNC manufacturers. Various research programs were undertaken involving Industry and Academic Institutions, principally OSACA (Europe), OMAC (North America) and OSEC (Japan). Later follow-on programs in Europe included OSACA II and OCEAN.

In most cases the basic outcomes were published but generally no recognised industry standards were adopted. Further developments were generally kept “in-house” by manufacturers (OSAI, NUM, Fagor, Siemens, Homag, Fidia), or offered by academic participants as Open-Source Software (OROCOS).

The fundamental features of an Open Systems Control were defined as

- Commercial or Industry standard hardware (able to be upgraded)
- Vendor-neutral architectures and application modules
- Modular software structures with well defined software interfaces
- Layered approach to structure hides hardware-specific features
- Flexible and reconfigurable, adaptable to new technologies and processes

4.2. New controller for the 1300X machine

The new machine control software features an Object.Oriented structure based on the fundamental classes of control system elements identified by earlier projects and enhanced by further studies.

Key features implemented as base and derived Object classes include:

- Machine operations – motion control, axis control, scheduling, machine safety logic.
- Cycle operations – programming and execution, process monitoring.
- Operator HMI - panels, buttons, DROs, LEDs, LEDBars, Graphs, Data Entry (Keypad).
- Equipment interfacing – Digital and Analog IO, Serial and Bus / Network communications.
- File and disk operations – Process data logging, Program storage, Configuration storage.

4.3. Open CNC and Open Device Interfaces

Statham [3] devised a 4-layer Open CNC Interface (OCI) framework, with the aim of aiding the integration of custom application programs with different CNC interfaces such as Siemens or Fanuc. An object-oriented design enabled the structuring of software classes (Base and Derived) to provide a standardised interface with abstracted levels of specific functionality.

An Open Device Interface (ODI) concept has been derived from the OCI model using an equivalent structure and philosophy. This has been designed to facilitate easier specification and integration of grinding process control and other equipment, using the common machine tool signal and data communication features.

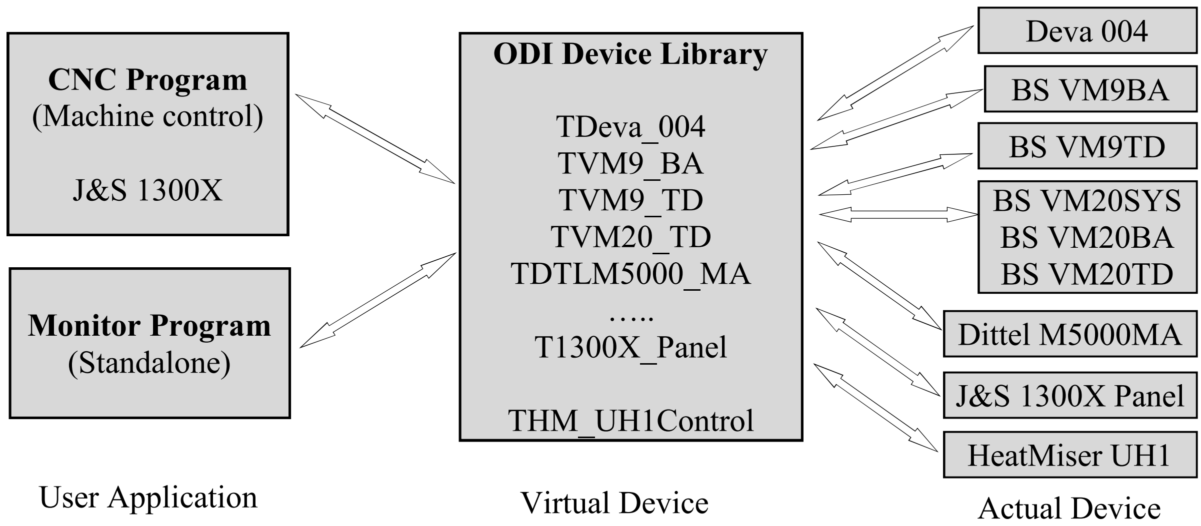


Figure 3: Overall ODI software and hardware structure

5. Control and Device classes and implementation

Some of the key object classes used to represent a VM20 Process Control unit are presented below in a UML Class Diagram format. It can be seen that a System may be configured with Balancer, Touch and Gauge devices, etc, if these are physically installed.

TVM20_Unit		
System	As	TVM20_SYS_Device
Balancer	As	TVM20_BA_Device
Touch	As	TVM20_TD_Device
Gauge	As	TVM20_GA_Device
Link	As	TVM20_ML_Device

TVM20_DEV_Config		
SoftwareVersion	As	String
Configuration	As	Byte
Simulation	As	Boolean
Installed	As	Boolean
Enabled	As	Boolean

TVM20_DataItem		
Valid	As	Boolean
Value	As	Double
NameText	As	String [12]

TVM20_TD_Device		
ConfigData	As	TVM20_DEV_Config
StatusData	As	TVM20_TD_Status
MonitorData	As	TVM20_TD_Monitor
ParamData	As	TVM20_TD_Params
CommandData	As	TVM20_TD_Command
SignalData	As	TVM20_TD_DigitalIO
StartPolling()		
EndPolling()		
GetVersion()		
GetStatus()		
GetMonitor()		
GetParams()		
SetParams()		
GetDigital()		
SetDigital()		
SetCommand()		

Figure 4: Sample classes implementing a VM20 Device

6. Conclusions and future work

A modular grinding controller with an Open Device Interface (ODI) to facilitate the integration of various Grinding process Control and Monitoring devices has been designed and demonstrated. The Object Oriented Framework allows a common access strategy for different makes and models of equipment (a generalised Device Object) by using layers of hardware and software abstraction.

Application software can now interact with different Devices much more easily, and a common user interface allows the operator to work with different devices easily. Two types of touch detector unit have been interfaced to the main machine control software to prove the functionality.

The concept is to be extended for the control of other Grinding machine types and further device types such as Motion / Axis Controllers and other peripheral monitoring equipment models.

References

[1] Thomas A 1996 *Adaptive Control System for Precision Cylindrical Grinding* PhD thesis Liverpool John Moores University, UK

[2] Balance Systems VM20 User and Installer Manuals: *VM20 Operation,* Profibus Interface, Remote Programming Version 9

[3] Statham C G 1999 *An Open CNC Interface for Intelligent Control of Grinding* PhD thesis Liverpool John Moores University UK