Design of a Real-Time Open Architecture Controller for Reconfigurable Machine Tool

I Masekamela[1*], N.S Tlale,[1] and C.M Kumile[2]

[1] Mechatronics and Micro Manufacturing
Council of Scientific and Industrial Research
Pretoria, Gauteng, 0001, South Africa

[2] Department of Mechanical Engineering
Tshwane University of technology
Pretoria, Gauteng, 0001, South Africa

ABSTRACT

Unprecedented and abrupt changes in market demands represent new conditions that manufactures of consumer goods need to operate within. Several factors are simultaneously contributing to these market changes, including globalisation of the world economy, saturated market and rapid advances in technology. The result has been sharp decrease in product cycles. Therefore higher quality products at lower cost and timely response to market changes become a necessary competitive advantage. This in turn requires appropriate business strategies and manufacturing technologies. More recently, the Reconfigurable Manufacturing System (RMS) concept was introduced to respond to this new market oriented manufacturing environment. In terms of design, RMS has a modular structure in form of modular machines and open architecture controllers that can quickly change the physical structure and appropriately adjust the control system to adapt to the new production requirements. The paper aims to present the design and the development of real-time open architecture controller for a reconfigurable machine tool.

1. INTRODUCTION

The current market trends which are largely influenced by globalisation are characterised by an unpredictable demand for highly customised and quality products. In turn manufactures are expected to adapt their machines and control systems to the production of these goods at reasonable time and cost. However, at present the machine control systems widely used do not provide the functionality, flexibility or cost effectiveness necessary to produce these goods. In trying to improve the control systems, manufactures often resort to the use of Commercially-Off-The-Shelf (COTS) controllers. Despite reducing the development cost and time, COTS controllers are proprietary and sometimes it is not easy to integrate them into available system. Also they largely depend on the control vendors for implementation and maintenance which increases the cost of manufacturing.

The new concept of RMS which is characterised by modular machine tools and open architecture controllers has been identified in recent years as a potential solution to those manufacturing problems. An ideal RMS would offer flexibility due to their ability to quickly rearrange the manufacturing system to adapt to different production conditions. They are composed of hardware and software modules which can be rearranged in a rapid and reliable way and they can be easily reused. Machines are built using modules that can be easily changed or integrated into the production line in short period of time, hence reducing down time. Open architecture controllers (OAC) are aimed at eliminating the problem of implementation by creating a flexible control system which can be attached to a wide variety of machine tools.

The advent of faster processors for personal computers (PC) and a general reduction in their prices have increased the use of PC-based controllers. PC-based controllers are generally flexible, open and can be easily integrated into other manufacturing functions. They also offer faster design cycles, lower down times due to the presence of diagnostic and simulation tools. These attributes help in enhancing productivity and reduces maintenance costs.

* Corresponding author: Tel.: (+27(0)12-841-3046; Fax: (+2712(0)12-841-4209; E-mail: IMasekamela@csir.co.za

The basic constituents of a PC-based controller are hardware platform which comprises of a computer hardware, communications network, peripheral devices and sensors. They are capable of generating, receiving, transmitting, processing and storing data or signals. Software layer which includes operating system, device drivers, network communication and application software enables inter-task communication, multitasking, interrupts handling, network, memory management, system level errors and controls device input and output handling.

The paper is arranged as follows; Section 2 describes the proposed experimental setup, Section 3 catalogues the proposed motion controller, Section for deals with modelling of open architecture controller while Sections 5 and 6 discuss and conclude the paper respectively.

2. EXPERIMENTAL SETUP

ng an open architecture controller for reconfigurable machine tool (RMT)
wever the project is still at its infant stage and is running concurrently with
which is also at its early stage. Most of analysis will be based on the some
ed or have been conceptualised by other people.

OL

xes is of major interest for this project as the resultant controller should be
lties. Figure 1 shows an example conceptual of a machine tool.

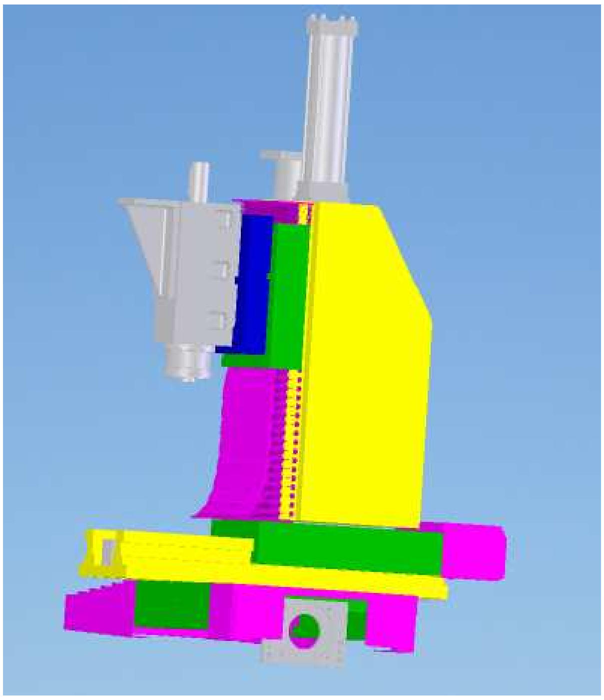

Figure 1:

The main physical components or mechanical modules of a machine tool include the table, which normally houses the workpiece holder, tool holder and other supporting links. As the machine boast different types of joints especially the translational and rotational joints, a user need to decide what type of a joint each axis will have depending on the type of task they want to carry out. So with the help of the RMT project, a host of possible configurations a machine tool can adopt are going to be availed through a library of mechanical modules. C++, an object-oriented programming language will be used to create a repository of mechanical modules. A suitable graphical user interface (GUI) will display the Computer Aided Design (CAD) drawings of the mechanical modules while the C++ programs will be running in the background. As cost effectiveness is one of the pillars of the project, SOLIDWORKS which is already in use in the department will be thoroughly explored with possibility of it being used as a GUI. However, it will be noted that in the event of the success of the project, a proper interface will be made for other CAD packages.

2.2. POSSIBLE CONFIGURATIONS

A provision will be made using a C++ program to allow the user to choose whatever configurations they want to adapt for their individual specific tasks. Hendrik Van Brussel et al in their analysis of three-axis machine tool defined ten configurational parameters that could be used to make different configurations of a machine tool [1]. If all the ten configurational parameters are viewed as independent entities about 6912 different machines configurations will be produced. Further subdivision of the ten parameters generated 56832 different configurations. However, the analysis of these different configurations together with their CAD drawings demanded huge processing powers and a powerful processor in HP 9000/782/C200+ still took 13-14 days to analyse the configurations. Bohez rather suggested a different perspective to the analysis of different configurations [2]. His approach consisted subdividing the machine into tool carrying and workpiece carrying axes. This approach generated 720 possible configurations which can possibly be trimmed down if an assumption that most machines have two rotary axes carrying the workpiece and three linear axes for the tool holder. Given that CAD drawings will be used to give the user the idea of the machine tool developed, it will be worthwhile to start generation of different configurations using Bohez approach and an expansion to that will be later explored.

2.3. ANALYSIS OF DIFFERENT CONFIGURATION

The main factors that influence the choice of a configuration are the task which will be performed by the machine tool and the workspace [2]. Bohez summarised some of the parameters that are used in the definition of the workspace so that the user could determine the task and the workspace that he wants before making the selection. Those parameters include workspace utilisation W_R, machinable volume, machine tool space efficiency MT_S, orientation space index of a five axis machine OS_I and the orientation angle index OA_I.

Workspace utilisation (W_R) is the ratio of the Boolean intersection of the tool workspace (WS_{Tool}) and the work piece workspace ($WS_{Workpiece}$) and the union of the two workspaces. Machinable volume is the total volume which can be removed from the work piece for particular instance of the tool and the work piece. The intersection of the machine tool workspace and that of the work piece gives the amount of material which can be removed. Machine tool space efficiency (MT_S) is the ratio of the machine tool workspace and the smallest convex volume which envelopes the machine. Orientation space index (OS_I) is the space orientation index is the ratio of the volume of the largest spherical dome which can be machined using the full range of the rotary axes divided by the machine tool workspace. The orientation angle is the ratio of the product of the maximum range of the two axes divided by the 360x180 multiplied by α_{12} /90 where α_{12} is the angle between two rotary axes

The application or the purpose of the machine tool is the ultimate decider in the choice of the configuration to be adopted. In most cases application of the machine tool is classified into two groups namely positioning and contouring [2]. Positioning is the orientation of the tool relative to the work piece in any direction while contouring is the controlling of the tool relative to the part while machining. In positioning, most of the axis positions are kept fixed during machining while in contouring the controller need to control all the axes during the material removal. The former is used for normal machining of geometrically defined parts like engines with the latter being suitable for complex shapes. The size and the weight of the part to be machined plays an important role in selecting the axes configuration as heavy materials require short work piece kinematics chain and also horizontal tables are preferred as they offer an easy platform for fixing and handling the work piece [2].

After the user has made a choice further analysis on the model will include kinematics and dynamics analyses of the resultant machine tool.

2.4. OPTIMIZATION OF DIFFERENT CONFIGURATIONS

Genetic algorithm seems to be a more appropriate approach in creating more robust mechanical configurations. In general the technique aims to maximise the static stiffness, maximise the lowest natural frequencies of the generated machine tool configurations and minimise the geometric interferences. Ultimately a suitable algorithm capable of executing this procedure will be created and tested on the generated tools.

2.5. KINEMATIC AND DYNAMIC MODEL OF A FIVE-AXIS MILLING MACHINE

Kinematics analysis of the machine will be carried out using the Denavit-Hartenberg representation. It is important to note that the key joints that are found in machine tools are either revolute or prismatic and C++ program which will be able to carry out the analysis will be made with emphasis on these two. The program should be able to carry out the dynamic analysis for all the different configurations possible.

3. DESIGN OF A MOTION CONTROLLER

The motion controller should be able to handle the different dynamical characteristics of the various generated machine tools. It should be able to interpret the data that show the configuration used and should adapt itself to that particular configuration. The main modules that would be built for the motion controller are the servo, spindle, interpolator and process controllers [5]. The servo controller will be responsible tracking the axis level while spindle controller will be responsible for the motion of the spindles. More the interpolator will coordinate the motion of the axes while the process controller will be responsible for optimisation of productivity and accuracy.

3.1. HARDWARE IMPLEMENTATION

The implementation of hardware will be one of the delicate matters to handle in ensuring a real time environment. Despite the use of the hardware that is real time compliant it is imperative to arrange the hardware in such a way that there are minimal communication errors and delays. This involves use of a hardware architecture that will be used in positioning the hardware modules of the system including sensors, microcontrollers and processors.

The proposed controller will use a hierarchical architecture where each axis will have its own servo controller and a supervisory controller which is likely to be a Wafer-Luke will be responsible for monitoring the entire system. The distributed nature and the multi-structure of this architecture will enhance reconfigurability. Even though the decentralised architecture seems to be more reconfigurable than the hierarchical architecture, its performance is highly likely to be affected by communication nodes and the decision was made to stick to a fail safe hierarchical architecture.

A communication network which will be responsible for providing reliable and temporally predictable message passing [3] between nodes will be adopted. The broadcast and bus topologies are currently the most common as they provide simultaneous arrival of signals at different tasks and are cost effective. At lower level Serial Real-time Communication system (SERCOS), a digital interface for communication between industrial controls and input/output (I/O) devices would be further investigated for possible implementation. Moreover the ability of SERCOS to coexist with other protocols like Ethernet makes it more suitable for implementation. Process Field Bus (PROFIBUS) is another communication interface which is normally used to operate sensors and actuators will also be investigated. Moreover Controller Area Network (CAN) bus, a computer network protocol would be explored.

4. DESCRIPTION AND MODELLING OF AN OPEN ARCHITECTURE CONTROLLER

It is very critical to design the overall controller in such a way that it is open for use in other manufacturing setups. The use of standards that are already available for a machine is important. Some of them include ISO 14649, an object-oriented programming interface that is used for part programming in Numerical control (NC) machines. Open Modular Architecture Controls (OMAC), an American organisation involved in standardising controls within America and the rest of the world has provided an application programming interface (API) that could be adopted for implementation as it comes with the notion of openness. Other useful standards include use of Universal Serial Bus (USB) as standard for connections with more investigations needed for its extensive use in a manufacturing setup.

4.1SOFTWARE IMPLEMENTATION

Careful analysis of software implementation for the controller will be made to ensure that the implementation adheres to the open and flexible environment required. So the problem solving will involve utilising some of the tried and tested approaches used in the design of software for an open architecture. Proper structuring of the software will be ensured through use of structures where a system can be decomposed into subsystems and consideration be made on the interaction of the resultant subsystems. So far there are nine structures already defined that will used. [3]. They range from conceptual structure which is used to understand functional requirements to the physical structure that is used for mapping software onto hardware and for reasoning about the performance. The other structures include class structure, data flow structure, control-flow structure, calls structure, uses structure and process structure, all handling different stages of the design process. Moreover the software implementation could be analysed from the architectural styles perspective, which is an extension to the structuring.

A good architectural style will ensure a well coordinated, synchronised and properly functioning system. A combination of styles normally creates a more robust system Key identifying characteristics include flow of data within a system, mode of transmission and good synchronicity. Each architectural style has its own advantages and disadvantages but for this particular project data abstraction and object oriented style will be the focus in software implementation. This style is characterised by encapsulation of data and its primitive operations into abstract data called objects [4]. The ability of an object to hide its representation from the clients makes it possible to change the implementation without affecting the client. The bundling of a set of accessing routines with the data they manipulate enables programmers to decompose problems into a collection of interacting agents [4]. C++, an object-oriented programming language will be used for this implementation.

In spite of favourable properties the object-oriented style posses some problems especially with interaction. The object must first establish the identity of another object it wants to interact with. In the event that identity of an object changes it is required that the updates are passed on to the objects that invoke it. However mixing this style with other will ensure more robustness. Other styles include pipes-and-filters, layered, time-triggered, reactive, process networks, publish and subscribe, client server, process control and finite state machine.

4.2 CONTOLLER IMPLEMENTATION

The proposed controller will be PC-based and the implementation would be affected by both software and hardware structuring. Motion control requires real-time abilities and an operating system with a real-time kernel will be used for implementation. Real-time (RT) Linux is capable of handling the hard real-time requirements of motion control but the initial stages which involves writing the algorithm for kinematics and dynamics analyses will be created using UBUNTU Linux. It has been found that most industries use windows for their front-end uses and therefore a provision for it will be availed. So far the OMAC API has been earmarked to act as interface between the operating system and the application programs.

5. DISCUSSION

The proposed controller will be developed with a clear understanding that it will be used to control different machine tools or at least different configuration of a machine tool. It will be equipped with the ability to make kinematics and dynamics analyses using algorithms written using C++, an object-oriented programming language known for its robustness. It should ultimately be able to adjust its functionality and performance to accommodate any possible configurations. Once the controller is running, new activities should be easily integrated to the existing system. The division of the machine tool into different mechanical modules requires a 1:1 mapping of the mechanical components to the controller components for effective implementation of openness. Furthermore each controller module should have a clearly defined communication interface so that it can easily communicate with other modules.

Self-diagnostic tools including sensors and microprocessors will be installed at each level to detect failures and for easy implementation of the recovery process. The hardware implementation would adopt the hierarchical architecture which promotes reconfigurability. Also proper positioning of sensors would lower the communications delays while at the same time enhancing real-time properties.

Architectural styles would be used to model the software for the controller with data abstraction and object-oriented style being the main. Finally it will be tested to determine if it meets the initial requirements which include openness, flexibility and modularity.

6. CONCLUSION

The need for a manufacturing system that can be quickly changed both at hardware and software level to adapt to rapidly changing markets intensified the need for modular machine tools and open architecture controllers.

The paper presented the steps that are going to be followed in developing an open architecture controller that will be used for controlling different machine tools or different configurations of machines. Although a number of similar project have been undertaken they have fallen short in terms of openness of the controller and in meeting the real-time requirements. It is expected that at the end, a controller that is capable of handling kinematics and dynamics configurations will be produced. It is expected that the architecture of the controller will be open to new activities within the manufacturing environment and it will be flexible enough to quickly adjust to varying production requirements.

The proposed controllers will be first modelled using the above mentioned tools and tested on virtual platform. Upon successful implementation on a virtual platform it would be tested on a real testbed which is still to be developed.

ACKNOWLEDGEMENTS

I would like to thank both my supervisors Dr Nkgatho Tlale and Chris Kumile for their guidance. I would like to acknowledge the both Council of Scientific and Industrial Research and Advanced Manufacturing Technology Strategy (AMTS) in providing financial support for undertaking the project.

REFERENCES

[1] Hendrik Van Brussel, Fellow, IEEE, Paul Sas, István Németh, Pierre De Fonseca and Pieter Van den Braembussche. "Towards a Mechatronic Compiler", IEEE/ASME Transactions on Mechatronics, vol 6, no 6, pp 90-105, 2001.

[2] E L J Bohez, " Five-axis milling machine tool kinematic chain design and analysis", International Journal of Machine Tools & Manufacture, vol 42, pp 505-520, 2000.

[3] D Chen, "Architecture for Systematic development of Mechatronic Software Systems", Royal Institute of Stockholm Masters Thesis, 2001.

[4] R Atta-Konadu, "Design and implementation of a modular controller for robotic machines", University of Saskatchenwan Doctorate Thesis, 2006.

[5] D.M Tilbury and S Kota, "Integrated Machine and Control Design for Reconfigurable Machine Tools", Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp 629-634, 1999.

[6] J Nasca, "Comparison of three different open architecture controllers", Proceedings of International Federation of Automatic Control (IFAC)-MIM, pp 134-138, Prague, Czech Republic, 2001.

[7] P Jaehun, S Birla, K G Shin, Z J Pasek, G Ulsoy,S Yansong and Y Koren, "An open architecture testbed for real-time monitoring and control of machining processes", Proceedings of the American Control Conference, pp 200-204, Seattle, USA, 1995.

[8] M Xiong-bo, H Zhen-yu, W Yong-zhang and F Hong-ya, "Development of a PC-Based Open Architecture Software-CNC System", Chinese Journal of Aeronautics, Science Direct, vol 20, pp 272-281, 2007.

[9] Y Altintas and N A Erol, "Open Architecture Modular Tool Kit for Motion and Machining Process", Annals of CIRP, Vol 47, issue 1,pp 295-300, 1998.

[10] L Willis, S Kannan, B Heck, G Vachtsevanos, C Restrepo, D Schrage and J V R Prasad, "An open software infrastructure for reconfigurable control system", Proceedings of the American Control Conference, pp 2799-2803, 2000.

[11] F Lian, J R Moyne and D M Tilbury, " Implementation of Networked Machine Tools in Reconfigurable Manufacturing Systems", 2000 Japan-USA Symposium on Flexible Automation, pp 349-361, Ann Arbor, 2000.