DESIGN AND DEVELOPMENT OF A 3-AXIS VERTICAL MILLING MACHINE CONTROL LOGIC ARCHITECTURE USING IEC 61499 FUNCTION BLOCK

M.A. Othman[1,2], Z. Jamaludin[1], M. Minhat[1] and M.A.U. Patwari[3]

[1]Faculty of Manufacturing Engineering,
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

[2]Centre for Diploma Studies,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia.

[3]Department of Mechanical and Chemical Engineering,
Islamic University of Technology, Gazipur 1704, Dhaka, Bangladesh.

Corresponding Author's Email: [1]zamberi@utem.edu.my

Article History: Received 6 December 2019; Revised 10 March 2020; Accepted 22 April 2020

ABSTRACT: The inflexibility of CNC control structure and the use of outdated CNC data interface are two major issues that limit the growth of CNC system. Emergence of STEP-NC standard provides comprehensive and rich information up to controller level that bring opportunity to overcome these problems. However, STEP-NC is only a passive data interface without intelligent functionality thus an IEC 61499 function block standard has been chosen as the research strategy to provide intelligent functionality to the proposed system. This then simultaneously resolves the inflexibility of the CNC control structure using distributed and layered architecture approach. This paper presents development of control logic motion system for a 3-axis vertical milling machines based on the IEC 61499 function block (FB). For control of the cutting processes, two types of motion control subsystems which are primary and feed motions are developed. Results showed that the designed FB is capable to provide the required functionality and represent actual machining working states. In future, this subsystem will be automatically forwarded with machining parameters and tool-paths data generation using native data generation module for performing actual cutting processes.

KEYWORDS: *Distributed Control System; IEC61499 Function Blocks; Milling; Motion Control; STEP-NC*

1.0 INTRODUCTION

Machining or metal cutting is the general term that refers to the removal process of undesirable part from a workpiece material in order to achieve a targeted shape. The process requires motions of the cutting tool relative to the workpiece (or vice versa) to accomplish every desired process. In term of capability to manufacture diverse and complex part features, machining process has widely been proven to be the most versatile and accurate among all the manufacturing processes [1]. Milling process is one of the most widely used metal removal methods. In milling process, rotary cutting tool is applied to cut away the unwanted material in the form of small chips by feeding the workpiece against the cutting edge of the tool. Normally, milling is selected to be used in manufacturing of parts that are not axially symmetric and have many features, such as holes, pockets, slots or even three-dimensional surface contours.

Development of machine tool for manufacturing is among the most significant paradigms shifts that has successfully revolute the metalworking industry. Employment of machine tool in shop floor has shifted the method product to be produce from man-made to machine assisted. However, at earlier stage, it was manually operated with some form of human assistance and required highly human skills to operate [2]. Establishment of Numerical Control (NC) technology and later the advanced version of NC technology called Computer Numerical Control (CNC) has brought great improvement and advancement to machine too1 applications. As a result, more and more automatic and flexible machine tools systems have been are created over the past decades. Now, CNC machines are embedded with multiple capabilities such, to name a few; multi-axis, multi-processes manufacturing and multi-tool to fulfill customer demands. CNC machines have become strategic manufacturing resources in many shop floors. As seen in Figure 1, in the near future, machine tools will undergo another important revolution, named the fourth revolution (MT4.0) [3]. New CNC machine tool with smarter, well connected, widely accessible, more adaptive and autonomous are expected to be developed and utilized in most manufacturing related activities.

However, limitations exist in CNC languages and inflexibility of the CNC control structure. These two major issues need to be addressed as pre-requisite to the development of the next generation CNC machine tool. To embed intelligent function on the machine tool, CNC controller must be equipped with comprehensive and high-level information in order to respond appropriately against any uncertainty or unplanned events. For this reason, a new data model called ISO14699 standard (also known as STEP-NC) has been progressively developed since the last decade [4]. STEP-NC provides standard and rich information and is set to be the future CNC interface. In addressing the inflexibility of CNC control structure , various efforts have been made towards open architecture control [5–8]. One of them is the development of software-based controllers to perform logical control in software rather than in the hardware as recognized by the industry [9]. Besides offering simplification of system architecture, the approach also improves system reusability, modularity, reconfigurability, and scalability. However, there are still many great deals of works yet to be done in this area.

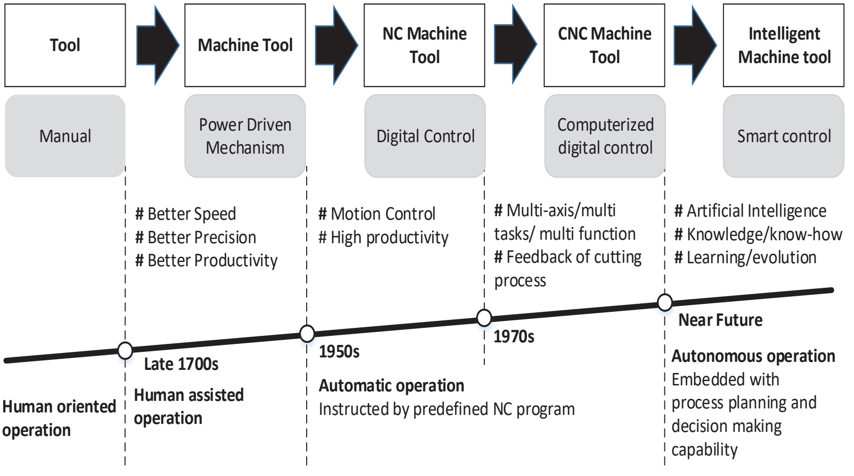


Figure 1: Evolution of machine tool technology

The main goal of this work as presented in this paper is to design and develop a 3-axis vertical milling machine control logic architecture as part of a proposed adaptive controller using IEC 61499 function blocks (FBs) [10]. This system intends to support the last-minute specific machine native information generation and dynamical changes of the events or/and parameters value. It is implemented in distributed control system environment using open source platform, Function Block Development Kit (FBDK) by Holobloc Inc [11].

2.0 METHODOLOGY

2.1 Control Architectures

Any CNC machine tool system essentially contains three main components as listed: (i) Program Input Devices, (ii) Machine Control Unit (MCU) and (iii) Machine Tool. Figure 2 represents the components of a CNC system. A part program is a series of coded instructions that are generated at offline stages beforehand. It contains an information related to the series of cutting tool movements and auxiliary functions in low level language. Through program input device, part program is forwarded into the CNC machine tool or more precisely into the MCU [12]. Basically, the MCU is divided into two main elements, which are data-processing unit (DPU) and control loops-unit (CLU). The functions of DPU are to read and to decode the input part program and implement interpolations to generate axes motion commands and auxiliary function control signals to the CLU. While, CLU is functional to feed the motion commands to the amplifier circuits for axis mechanism execution purpose and implement auxiliary control function such as coolant on/off and tool change. Besides that, the CLU is also responsible to receive feedback signals of position and velocity of each drive axis.

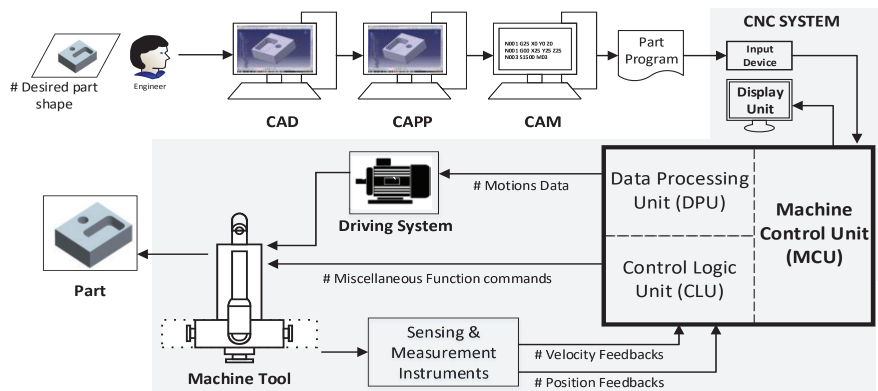


Figure 2: CNC machining systems

2.2 Requirements of Next-Generation CNC System

Developing a new breed of intelligent CNC machine tool is not an easy task to be implemented on existing CNC system especially when majority of the CNC machines available worldwide are principally still using outdated G&M codes as the machine control interface. Without having comprehensive and rich information on the machine level, intelligent decision making and control are nearly impossible to be realized. As guidelines, some of the commonly accepted characteristics reflecting the intelligent and versatility of the CNC system are as follows:

i. Seamless vertical integration between machine tools and CAx chain using high level programming language;
ii. Able to determine optimal machining parameters and generate tool path automatically at machine level;
iii. Able to respond adaptively and autonomously in real time to sustain a targeted performance;
iv. Open architecture with software-based modular structure as an implementation technology.

In order to create the new breed of CNC machines that are more flexible, interoperable and intelligent, the CNCs need high level data at machine level. The design of these CNCs must allow bi-directional flow system and will be based on open architecture.

2.3 IEC 61499 Overview

IEC 61499 standard was first introduced in 2005 with specially designed as reference architecture for distributed, modular and flexible control system development [4]. This standard extends from the subroutine like structure in IEC 61131-3, to the event driven execution semantics using the Function Block (FB) concept. Each of FBs consists of graphical event-data interface for communicate to other FBs in the network and a set of executable functional that encapsulate local data, state transitions and internal algorithms to provide an independent computational activity with its own set of variables for responds to a specific input [13]. The main concept of an individual FB execution in the network is triggered by the events it received, otherwise the FB will keep remaining in idle condition for the rest of the time. This well-defined event-data interface and the encapsulation of local data and control algorithms make each FB a reusable functional unit of software.

The main concept of IEC61499 execution is based on an explicit event-driven model. Every FB function as a software unit that consists of a graphical event-data interface. Each of them is encapsulated with local data, state transitions and internal algorithms to provide an independent computational activity with its own set of variables [13]. The execution of an individual FB in the network is triggered by the event it received, otherwise the FB will keep remaining in idle condition for the rest of the time. This standard aims to establish an open, component-oriented and platform independent development framework to improve re-usability, reconfigurability, interoperability, portability, and distributed of control software for complex distributed system [14].

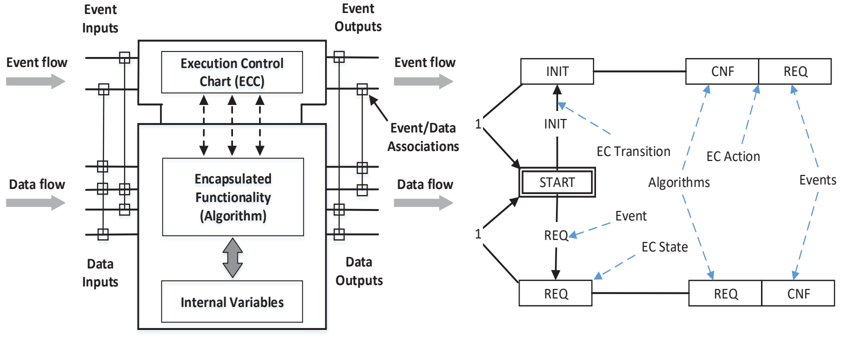
2.4 Function Block Structure and Types

Function block structure is divided into event flow and data flow as presented in Figure 4(a). Based on internal state designed inside the function block, numerous outputs can be produced even though the same inputs are applied. This dynamic of FB behavior is determined by its execution control chart (ECC) that is encapsulated inside. An ECC of FB consists of EC state, EC transition and EC actions as shown in Figure 4(b). The execution of ECC starts when EC transition was triggered and cause changes to the EC state. Based on execution of the associated algorithm, EC actions will take place and issued an output event.

There are three standard classes of function blocks defined in IEC 61499:

i. Basic function blocks: Elemental control function. Explicit event-triggered component containing algorithms and has an execution control chart (ECC), together with input and output data set variables.
ii. Composite function blocks: Conglomerate of multiple blocks (basic or/and composite FB) to encapsulate a higher level of control function. It is interconnected by external data sources and the functionality is determined by network of FBs inside.
iii. Service interface function blocks: Use to provide communication services between FB networks and its surrounding environment.

Numerical control procedure of machine tool is a typical event driving and interacting system, transitioning between different working states such as power up, machining state and emergency.



(a) (b)

Figure 4: (a) Basic function block structure and (b) ECC

3.0 RESULTS AND DISCUSSION

3.1 Data Input Model

The overall system receives generic STEP-NC as an input. Then, based on resource availability, the controller will generate native data for targeted machine tool. The machine specific data produce list of cutting tool, machining parameters and toolpath. Based on the generated data, the control logic commands will be feed to the physical machine tool for performing the desired operation.

3.2 Types of Machine Tool Motion

The main principle of all machining processes is providing suitable relative motions between the cutting tool and the workpiece to produce the characteristic geometry. Normally, there are two types of relative motion involve in every cutting process that is either using a machine tool or manual tool (as depicted in Figure 5):

i. Primary motion is also known as main motion. It represents the relative motion between the cutting tool and the workpiece. In vertical milling system, the primary motion represents the tool rotational by the spindle. The motion can be characterized with two components, which are velocity and rotation direction.
ii. Feed motion is also known as traverse motion. It is secondary motion that adds to the primary motion leading to a repeated or continuous chip removal and the creation of a machined surface with the desired geometric characteristics.

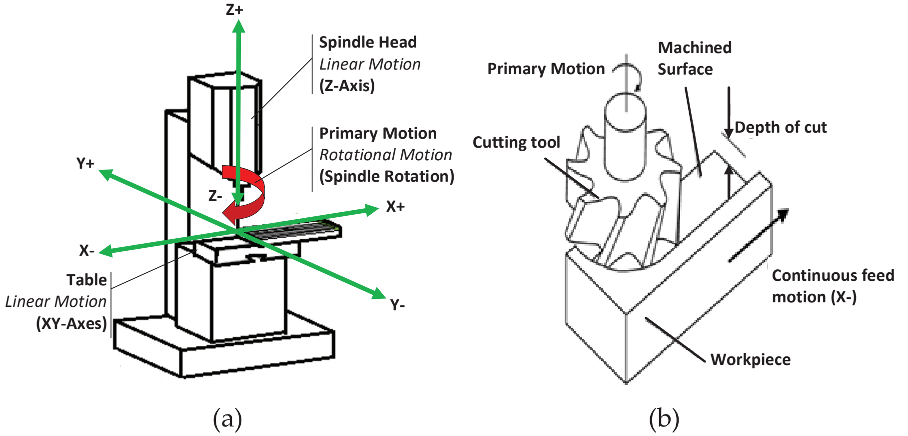


Figure 5: (a) Vertical milling machine architecture and (b) motion types milling [1]

Each configuration of machine tool produces the desired machining features together with operation required is mapped in Table 1. The type of FBs is needed to be developed in order to determine a machine control logic.

Table 1: Mapping motions toward cutting operation

Machining Features	Cutting Operations	Types of Motion	
		Primary Motion	Feed Motion
Milling Operations			
Face pocketing slotting.	Face milling bottom and side milling.	Spindle motion.	X-Axis positioning motion; Y-Axis cutting motion; X-Axis positioning motion; Y-Axis cutting motion; Z-Axis positioning motion; Z-Axis cutting motion.
Drilling Operations			
Holes.	Drilling boring reaming thread cut.	Spindle motion.	Z-Axis positioning motion; Z-Axis cutting motion.

3.3 Primary Motion Function Block Model

In a 3-axis vertical milling machine, the primary motion refers to the spindle motor movement condition. Spindle motor rotational direction and velocity are two parameters that need to be control based on desired machine state. The summary of events and the output data produced by the embedded algorithms are presented in Table 2.

Table 2: Primary motion control condition

Events	Output Data	Remark
INITO	Rotation Direction: Default Alarm: False Notification: "System Ready" Velocity: 0	Initial state condition when machine is power ON.
ACTIVE	Rotation Direction: CW/CCW Alarm: FALSE Notification: "Motor ON" Velocity: Input velocity	Condition when spindle motor is running at idle condition with desired velocity and direction.
BUSY	Rotation Direction: CW/CCW Alarm: False Notification: "Machining In progress" Velocity: Input velocity	Condition when machining process is being executed.
OFF	Rotation Direction: CW/CCW Alarm: False Notification: "Motor OFF" Velocity: 0	Condition when spindle motor is stop at desired velocity and direction.
ERROR	Rotation Direction: CW/CCW Alarm: True Notification: "Parameter invalid" or "Emergency stop" Velocity: 0	Condition when input parameter is invalid or when emergency stop button triggered.

The design and development of the primary motion control FB as depicted in Figure 6, consists of the basic machine states such power-up, motor run, machining execution and motor stop, the error state due to invalid parameter and emergency button triggered. Rules from the type such as IF <condition> Then <conclusion> are applied in the algorithms to perform the decision-making function. The event execution control chart for this FB is shown in Figure 7.

The primary motion control of FB with ECC and its algorithm are tested and the results in every event are presented in Figures 8(a)-(f). Results showed that the primary motion control FB was successfully developed and was able to produce desired action and output based on the input events.

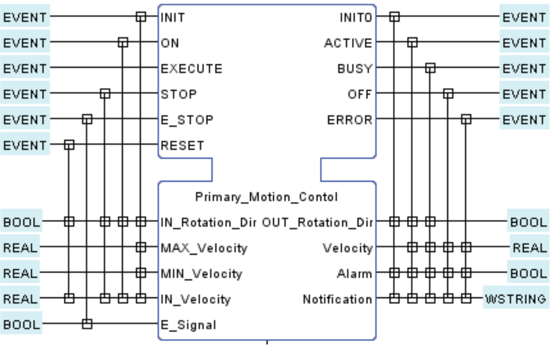


Figure 6: Primary motion control FB with ECC

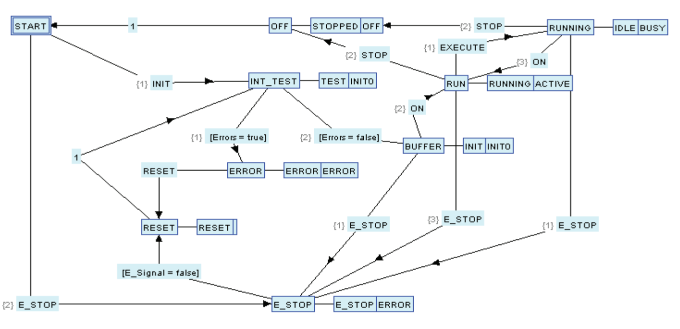


Figure 7: Execution control chart for primary motion control FB

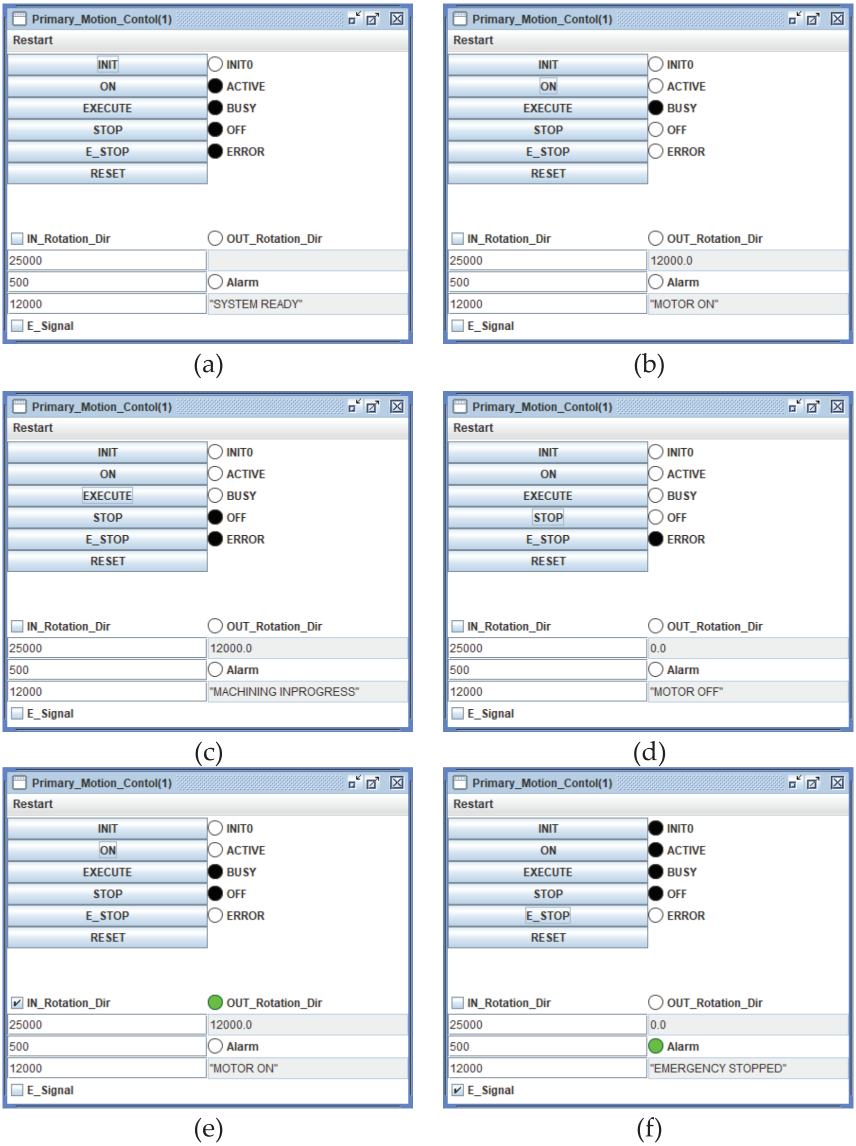


Figure 8: (a) System ready, (b) spindle rotate, (c) machine execution, (d) spindle stop, (e) spindle rotate counter motion and (f) emergency button active

3.4 Feed Motion Function Block Model

The z-axis, x-axis and y-axis are involved in modelling of feed motion FBs. The motion of z-axis represents the vertical motion of the cutting tool (spindle) and depth of cut parameter during machining whereas, the x and y axes represent the motion of workpiece/working table during machining process execution. Synchronization of these 3-axes motion together with primary motion is required during machining process to produce desired machining features. This kind of motion is called the traverse motion. Two types of motion are involved in feed motion, positioning (rapid) motion and cutting motion. Each has positive and negative direction along the axis. In order to increase the usability of the FBs, velocity control FB and axis directional control FB as shown in Figures 9 and 10, are designed to support the axis motion FB. The axis directional control FB considered three conditions of axis movement namely positive direction, static and negative direction.

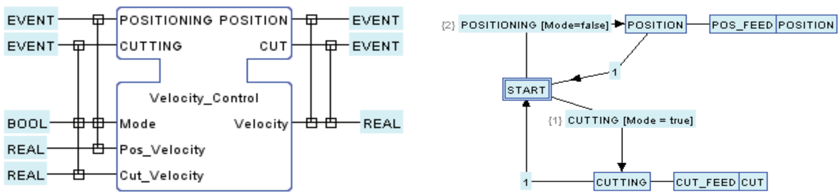


Figure 9: Velocity control FB with ECC

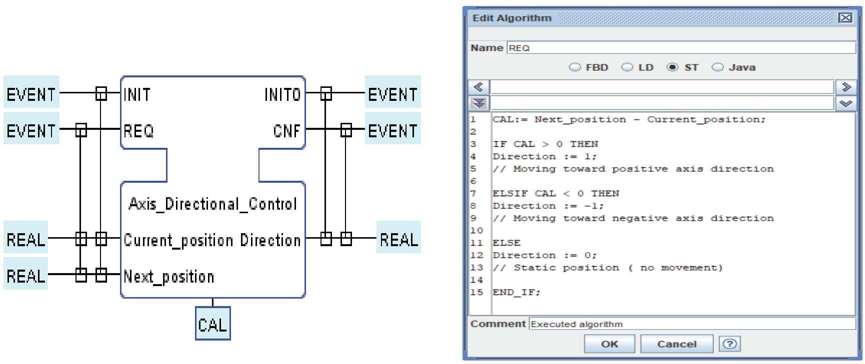


Figure 10: Axis directional control FB with REQ algorithm

4.0 CONCLUSION

The paper introduces design of motion control for a 3-axes vertical milling machine based on distributed control system using IEC 61499 standards. This approach offers more flexibility and less resource consumption in the development of logic control compared to hard coding using software-based modular structure. The main advantage of this approach is reusability of designed FBs that makes motion logic control architecture simpler. Currently, the traverse motion model only considers linear motion (X, Y, Z axes) and rotational motion (A, B, C axes) model need to be designed in the future to control 4-Axis or more CNC machine tool.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Universiti Teknikal Malaysia Melaka for the provision of the facility, equipment and materials used in the study.

REFERENCES

[1] A.Y.C. Nee, *Handbook of Manufacturing Engineering and Technology*, United Kingdom: Springer-Verlag London, 2015.

[2] T. Moriwaki and K. Shirase, "Intelligent machine tools: current status and evolutional architecture," *International Journal of Manufacturing Technology and Management*, vol. 9, no. 3-4, pp. 204–218, 2006.

[3] X. Xu, "Machine tool 4.0 for new era of manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 92, no. 5, pp. 1893–1900, 2017.

[4] ISO 14649-1. (2003). *Industrial automation systems and integration–physical device control–data model for computerized numerical controllers: part 1–overview and fundamental principles* [Online]. Available: https://www.iso.org/standard/34743.html

[5] H. Liu and Q. Liu, "Development of an open CNC system for multi-axis machine tools based on TwinCAT and .NET," in IEEE International Conference on Mechatronics and Automation, Beijing, 2015, pp. 2449-2453.

[6] Y. Yusof and K. Latif, "Development of new open soft-CNC system," in International Conference on Computer, Communications, and Control Technology, Kuching, 2015, pp. 82-86.

[7] C. Brecher, A. Verl, A. Lechler, and M. Servos, "Open control systems: State of the art," *Production Engineering*, vol. 4, no. 2, pp. 247–254, 2010.

[8] N.M. Seman, Z. Jamaludin, M. Minhat and S.C.K. Junoh, "Design of a CAD/CAM-simulink data exchange system For machine tool application," *Journal of Advanced Manufacturing Technology*, vol. 12, no. 1(2), pp. 1–14, 2018.

[9] X.W. Xu and S.T. Newman, "Making CNC machine tools more open, interoperable and intelligent—a review of the technologies," *Computers in Industry*, vol. 57, no. 2, pp. 141–152, 2006.

[10] M. Honegger, R. Brega and G. Schweiter, "Application of a nonlinear adaptive controller to a 6 dof parallel manipulator," in IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000, pp. 1930–1935.

[11] O. Rohat and D. Popescu, "Is FBDK suitable for developing and implementing process control optimization problems?" in International Conference on System Theory, Control and Computing, 2014, Sinaia, Romania, pp. 117-122.

[12] H. Sarhan, "A novel technique for controlling CNC systems," *Control Theory Informatics*, vol. 4, no. 5, pp. 82–92, 2014.

[13] V. Vyatkin, "IEC 61499 as enabler of distributed and intelligent automation: state-of-the-art review," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4. pp. 768–781, 2011.

[14] C. Pang, S. Patil, C.W. Yang, V. Vyatkin, and A. Shalyto, "A portability study of IEC 61499: semantics and tools," in IEEE International Conference on Industrial Informatics, Brazil, 2014, pp. 440–445.