

Design of Joint Controller Circuit for PA10 Robot Arm

Sereiratha Phal and Manop Wongsaisuwan

Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, 10330, Thailand.
phal.sereiratha@gmail.com

Abstract— This paper presents the design of circuit and algorithms for controlling the joint of Mitsubishi PA10 Robot Arm. To control the entire Mitsubishi PA10 arm robot, which all the joints equip with a permanent magnet synchronous motor (PMSM) for rotation and two resolvers as feedback devices, one for joint angle position and another one for rotor angular position, the joint controller is really important. The aim of the joint controller design in this paper is to be able to drive the whole PA10 robot arm when combining seven of this controller to meet the seven joints of the robot.

Index Terms— Permanent magnet synchronous motor; Pulse-width modulation; Sine pulse-width modulation; Micro controller unit.

I. INTRODUCTION

Mitsubishi PA10-7C robot Arm is a 7 degrees-of-freedom (DoFs) articulate robot arm made by Mitsubishi Heavy Industries. The robot consists 7 revolute joints, 3 joints for rotation (R), 4 joints for pivot (P) and configured as R-P-R-P-R-P-R from robot-mounting side. Each joint of the robot consists of one Permanent Magnet Synchronous Motor (PMSM), two resolvers as position and velocity feedback, one is attached to the rotor of the motor and the other one is attached to the motor gear, and one electrical brake [1, 2]. The joint controller design is an important step in design the whole robot controller as the whole robot cannot work without sub controller of each joint. There are studies on driving PMSM motor (type of motor of the robot joint) with resolver sensor [3, 4] and sensorless [5] which try to drive the PMSM as the command order. According to the default configuration of the robot joints, the controller in this paper will be designed using the resolvers at the joint as feedback device.

The main contribution of this paper are the demonstration of the real circuit design (hardware) for driving the joint of the robot, including inverter driver circuit and the resolver interface circuit, overall technique to overcome the limitation of necessary calculation on the MCU level and also the algorithm for the robot joint controller.

The rest of this paper is organized as follows. The next section introduce the theory, hardware needed and circuit design for the controller. In section 3 deal with algorithm and software design for MCU follow by the current output of the project in section 4. Finally section 5 provides our conclusion and future work.

II. THEORY AND HARDWARE DESIGN

There are three parts in this section. Firstly, the inverter of the motor drive is presented. The second part is followed by the needed function of MCU. The last part is about the resolver feedback and its interface circuit.

A. Inverter driver

The nature of the PMSM is that it needs to be fed by the sinusoidal current according to the position of the rotor. That is why it needs the three-phase voltage source PWM inverter as shown in Figure 1, to switch DC sources on and off to make PWM signal with the form of sinusoidal, called Sine Pulse-Width Modulation (SPWM). The circuit model of a typical three-phase voltage source PWM inverter as shown in Figure 2 consists of 6 power switches that shape the output, which are controlled by the switch variable a, a', b, b', c, and c'. When an upper transistor is switched on (a, b, c, is 1) the corresponding lower switches are turned off (a', b', c' is 0).

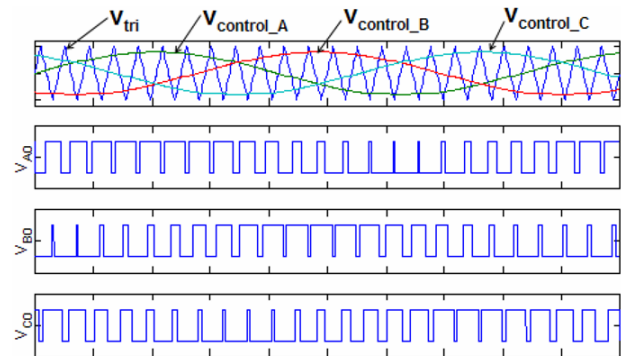


Figure 1: Sine Pulse Width Modulation (SPWM)

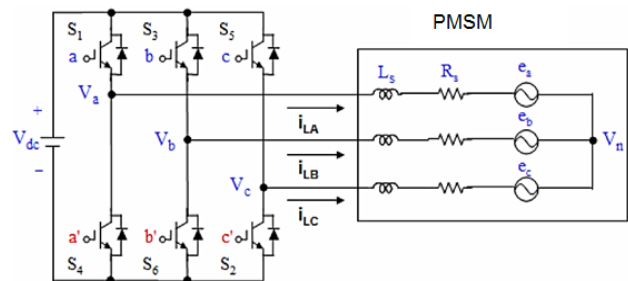


Figure 2: Three-phase voltage source PWM Inverter

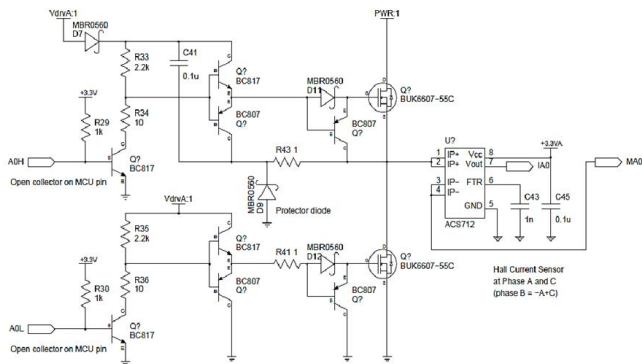


Figure 3: Switching circuit design

The feedback from the rotor position is important for making the flux of the stator to push the rotor with the specific sinusoidal waveform [6]. After the rotor changed position, the flux vector has to change as well. So the interrupt of the controller design has to be used in order to generate the flux vector from the stator according to the update of rotor position. The angle of flux vector which is generated by the stator to the flux vector of the rotor is really important as it limits the torque that is used to push the rotor. In order to get maximum torque, the stator flux vector must be maintained and keep orthogonal to the rotor flux vector as shown in Figure 4. The inverter driver board is created as the design from Figure 2 and Figure 3 is shown in Figure 5.

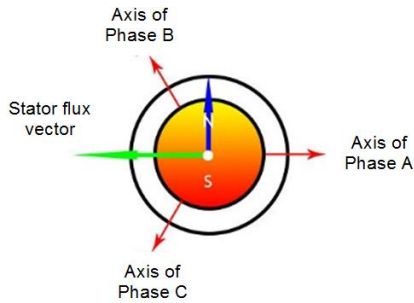


Figure 4: Three-phase voltage source PWM Inverter

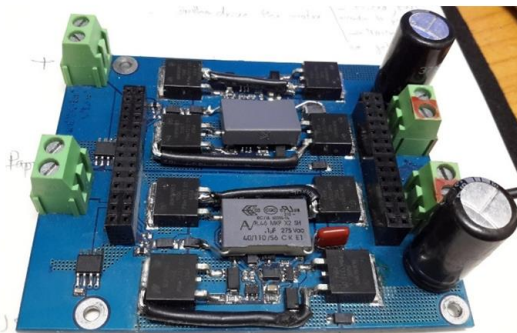


Figure 5: Inverter driver board

B. Microcontroller and Hardware Structure

The joint robot controller structure in Figure 6 is deployed in the prototype experiment. The PC will command the

position of the robot to the MCU and MCU will read the joint position from the resolver and calculation for the duty cycle of PWM needed to produce SPWM signal. The SPWM signal will drive the PWM inverter to drive the motor at the joint robot.

A micro controller unit (MCU) SAM4S is used in this project as the requirements such as PWM (for generating SPWM according to the position of the rotor flux vector), USART (for communication the PC), SPI (for communicating between MCU and AD2S1205), and Counter Interrupt.

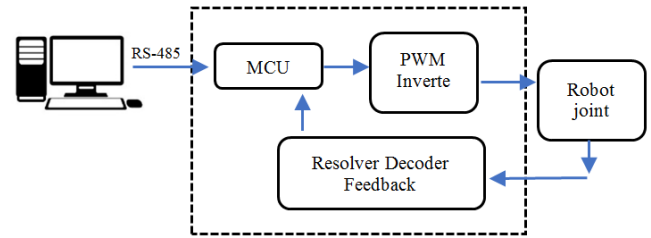


Figure 6: Joint robot controller design structure

C. Resolver and Interface Circuit

As the configuration of the robot joint which resolver is used as the feedback device due to its robustness, measurement accuracy, non-contacting working principle and resolution over a wide ambient temperature [6]. Resolver can give absolute position of rotation with high resolution. AD2S1205 is deployed in this prototype for decoding the rotor position from resolver.

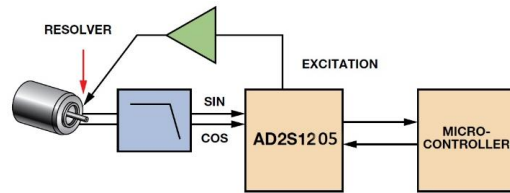


Figure 7: Typical resolver system block diagram

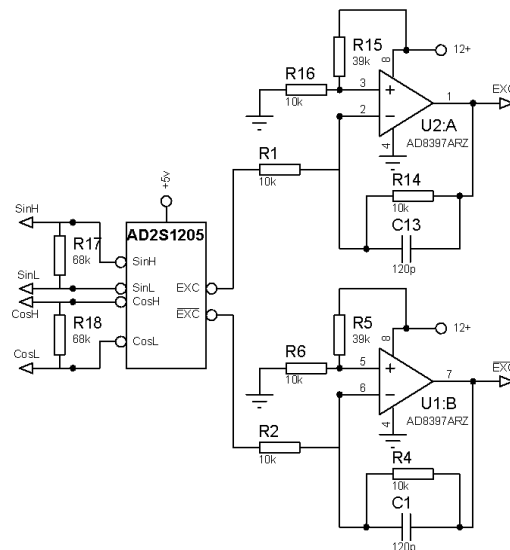


Figure 8: AD2S1205 Interface and Buffer Circuit

The typical resolver system block diagram is represented in Figure 7. The working principle of resolver is the sinusoidal reference signal (excitation signal) is amplified and injected to the resolver via EXC and EXCnot then the resolver generate Sin and Cos wave form as the output position. The Sin and Cos signal is computed by the analog to digital decoder AD2S1205 and generate digital output via SPI protocol. The excitation signal is generated by AD2S1205 and needed to be amplified before injected to the resolver.

Figure 8 demonstrates the interface of analog to digital converter for resolver (AD2S1205) and the buffer circuit of the Excitation pins (EXC and EXCnot).

As the outline of the IC in the loss of signal detection section, 68k Ω resistor is put between the CosH and CosL input pins and the SinH and SinL input pins to ensure the loss of signal detection when the four signal inputs from resolver are disconnected [7].

Also High-current dual op amp with rail-to-rail output to amplify and level shift the reference oscillator (excitation signal output from AD2S1205 for injecting to the resolver) AD8397 is used to optimize the interface with the resolver. AD8397 can achieve wide dynamic range, low-distortion, and high output current, make it suitable for the use with resolver [8]. The resolver interface board in Figure 9 was constructed using the design in Figure 8 and worked well after the testing with the real robot joint.

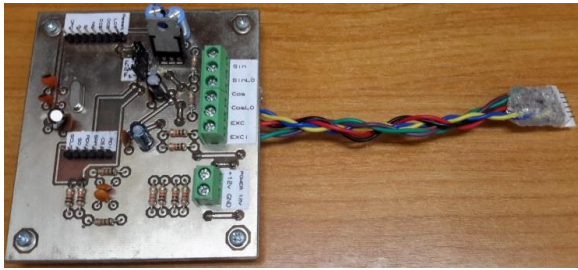


Figure 9: Resolver interface circuit board

III. ALGORITHM FOR SOFTWARE DESIGN OF THE CONTROLLER

A. System Level

Figure 10 shows the overview abstract of the controller algorithms and motor algorithms. Figure 10(a) presents the high level controller algorithms (PC). Using the trajectory information of the robot, the controller compute the distance and the target for the motor joint. This will compare the current position to the target in order to generate speed and command the motor to move. The low level controller algorithms (motor level) is demonstrated in Figure 10(b). The motor gets the command from the controller. The MCU read the current position of the rotor then calculate for the flux vector of the stator for moving the motor in maximum torque. After that send the PWM output for making SPWM to the stator accordingly.

B. Calculation Problems for MCU

MCU ability of computation is much less than computer nowadays which can perform with high speed up to GHzs. More than this the computation time for floating point variable operation do concern with this low speed and especially non-floating point micro controller unit. It is not only a few clock cycles to do arithmetic operation for floating point calculation, especially multiplication operation which is used very often for controlling process. Some compilers can help making optimization for fixing value variable (Constant) but not the variable which its value is changing from time to time. And there are many variables of this type are frequently used in the programming. So for reducing the calculation time, the optimization of using variables as integer data type must be used and also making the floating point variable to be constant value.

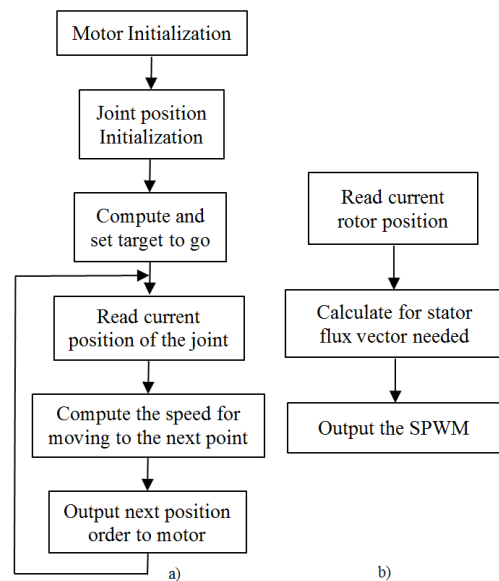


Figure 10: High level specification; left: Controller algorithms, right: Motors algorithms

In this project SAM4S16C is used. This MCU can perform up to 120MHz in speed but it is not floating point MCU. The using counter interrupt is necessary for updating the rotor and joint position in a specific period in order to driving the motor. Thus the time for computation in the MCU is limited. The maximum frequency of the PWM of the MCU module is 30KHz [9] so the update SPWM must be at most equal to half of PWM frequency. So the counter interrupt was set to 12.8 KHz or equal to 78 μ s. In this 78 μ s period there are many processes to do such as retrieve position from feedback sensors, controlling process and also calculation for SPWM.

One of the most spending time is making SPWM signal which require the value from $\sin(x)$ and multiplied by the PWM duty cycle. MCU does not have the operation for $\sin(x)$ and it normally can be computed by using Taylor series:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} \dots\dots$$

but it contains many multiplication operation and it can exceed the limited computation time just for this calculation. As shown in Figure 11, it spent almost 70% for calculation of sin(x) using Taylor series.

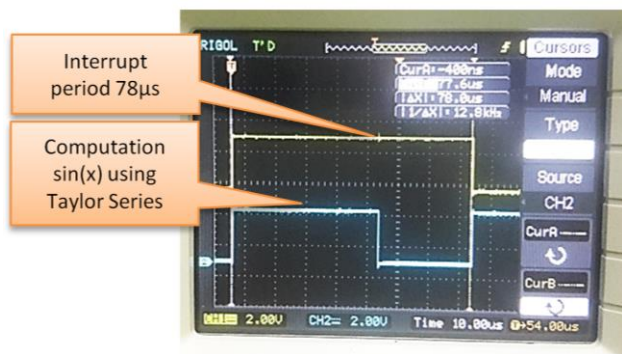


Figure 11: Computation time for Sin(x) using Taylor Series

To solve this problem the use of the constant array for the sin(x) variable is deployed. In the coding, the value of sin(x) from 0° to 360° was divided to 127 resolution and stored as the array elements where the index of the array are the angle x with the conversion 127 resolution as well. Doing this it means that the multiplication of PWM duty cycle and sin(x) just the operation of integer value and constant floating point value only. After the experiment doing these, the computing time is enough for the limited period. Figure 12 presents the total time using in driving the robot joint buy using calculation technique above. It also include the time for reading the feedback from resolver, calculation for SPWM, and setting the PWM to inverter board.

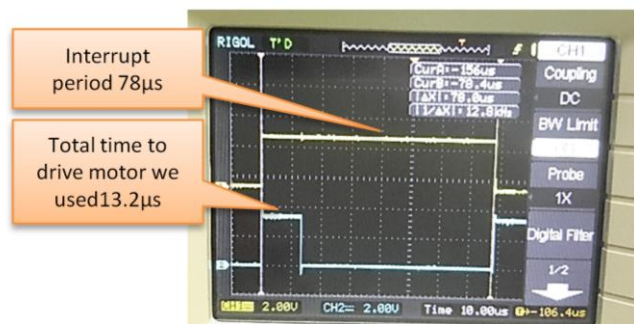


Figure 12: Total time using in driving the joint

IV. CURRENT OUTPUT

In the current output of this project, the circuit boards required for controlling the joint of Mitsubishi PA 10 Robot Arm were already constructed, inverter driver board in Figure 5 and resolver interface circuit board in Figure 9, and the

algorithms and computing technique for MCU were implemented with the successfully driving PMSM motor at the joint of the robot arm.

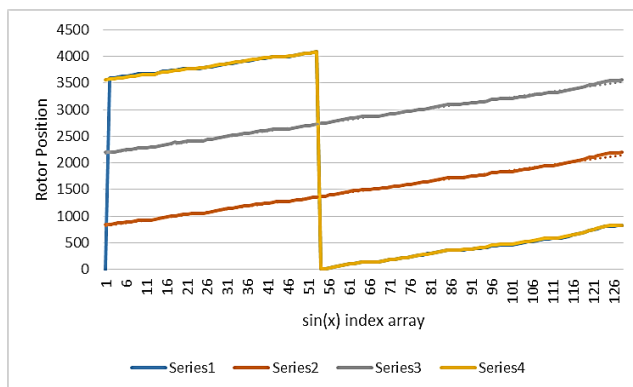


Figure 13: Rotor Position According to SPWM Input

Figure 13 is the experiment result of driving PMSM motor with the SPWM signal. The vertical axis is the position of the rotor and the horizontal axis is the angle of sin(x) which is mapped to 0 to 127 resolution instead of 0° to 360°.

V. CONCLUSION AND FUTURE WORK

This work presents the circuit design, calculation technique for MCU, and algorithms that make it possible for controlling the robot joint. The joint controller is one of the really important parts as it gives the sight of controlling the whole robot when making combination of the controller to meet the number of the robot joints with the proper control design from high level system (PC). The current work is not sufficient yet. Next, defining the suitable protocol to communicate between the controller and PC via RS485 standard is needed which lets the PC to function as the high level controller to control all sub controller to drive the whole Mitsubishi PA10 Robot Arm in the future.

ACKNOWLEDGMENT

I would like to express my sincere thank to Dr. Sirichai Pornsarayouth for his valueable time, immeasurable help, encouragement, and effecent guidance for this project.

REFERENCES

- [1] *Portable General Purpose Intelligent Arm Operating Manual*, L. Mitsubishi Heavy Industries, Editor.
- [2] A. Mosuwan, *Design of Simplified and Low Cost Controller for Industrial Robot Arm*, in *Electrical Engineering*, 2014, Chulalongkorn University.
- [3] D.W. Brown, *Real-time fault detection and accommodation for COTS resolver position sensors*. in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, 2008.
- [4] W. Kaewjindam, and M. Konghirun. *A DSP - Based Vector Control of PMSM Servo Drive Using Resolver Sensor*. in *TENCON2006. IEEE Region 10 Conference*. 2006.
- [5] P. Kumar, N. Kaushik, and S. Dahiya. *Simulation of sensor less speed control of PMSM based on FOC method with MRAS adaptive speed estimator*. in *India Conference (INDICON), 2013 Annual IEEE*. 2013.

- [6] P. Pillay, and R. Krishnan, *Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives*. Industry Applications, IEEE Transactions on, 1991. 27(5): p. 986-996.
- [7] *AD2S1205*, I. Analog Devices, Editor. 2007.
- [8] J. Symczak, S.O.M., J. S. Gealon, C. Nelson *Precision Resolver-to-Digital Angular Position and Velocity*. 2014.
- [9] *SAM4S Series*, Atmel, Editor. 2015.