Design and Construction and Motion Control of 6-Axis Robot Manipulator for Industrial Applications

Supachock Tuntivivat*, Pradya Prempraneerach

Department of Mechanical Engineering, Rajamangala University of Technology Thanyaburi, Pathumthani, Thailand 12110
*Corresponding Author: supachock_chock@hotmail.com, 085-0471123, 02-549-3430

Abstract

In this research, a 6-axis robot manipulator arm is designed and constructed for industrial applications. Then, the robot arm motion is controlled in a position mode by users' specified angular motion of each joint so that the robot arm can move to desired locations with high accuracy even with load variation and repeatability less than 10 micrometer. A motion control has been developed within Visual C++.NET to control individual or combined joints' position, velocity, acceleration. The purposes of this robot-arm motion-control implementation are to accelerate users' learning process and to interact with the industrial robot in a user-friendly environment.

Keywords: Industrial Manipulator Arm, Motion Control Software, Accuracy and Repeatability.

1. Introduction

In past decades, industrial robot manipulator arms have revolutionized various industrial applications, especially in high-technology factories, because of their high-yield production capability, high-accuracy repeatability, flexibility and small contamination in clean environment. However, a lacking of specialized personals in the robot control field and a high investment cost and maintenance cost are major obstacles for a wide-spread use of the industrial manipulator arms in Thailand.

Chandrakasem Rajabhat University [5] has proposed to develop 5-axis robot arms for the educational purpose so that students might be able to transfer knowledge to their future industrial workplaces. Institute of FIeld roBOtics [6] has designed and constructed the industrial 6-axis robot arm that can perform a welding process for automotive industries.

Therefore, this research focuses on the design and construction of the industrial robot arm as well as the motion-control software development using Visual C++.NET, which has user friendly interface. As a result, any person without a prior knowledge can comprehend and control the robot arm control with a minimal learning time.

This paper can be subdivided into four sections. Section 2 reviews the design and construction of robot mechanisms and motion controller units. Section 3 emphasizes on the implementation of the motion-control software. Lastly, the performance tests of this robot arm are summarized in Section 4.

2. Robot Arm Construction

The construction of the 6-axis robot manipulator arm is divided into two main parts: 1) robot mechanism construction and 2) controller and drive construction. Both constructions can be carried out at the same time and then assembled together afterward. Fig. 1 shows an overall schematic diagram of this 6-axis manipulator arm, where circles represent each joint of the robot arm. A computer passes user commands through the SPiiPlus Motion Controller [2] to control motor drives, connecting with 6 DC and AC motors and 6 optical encoders. All DC and AC motors coupling with harmonic gears provide torque to each axis of the robot arm and encoder signal is used as feedback joint angular position.

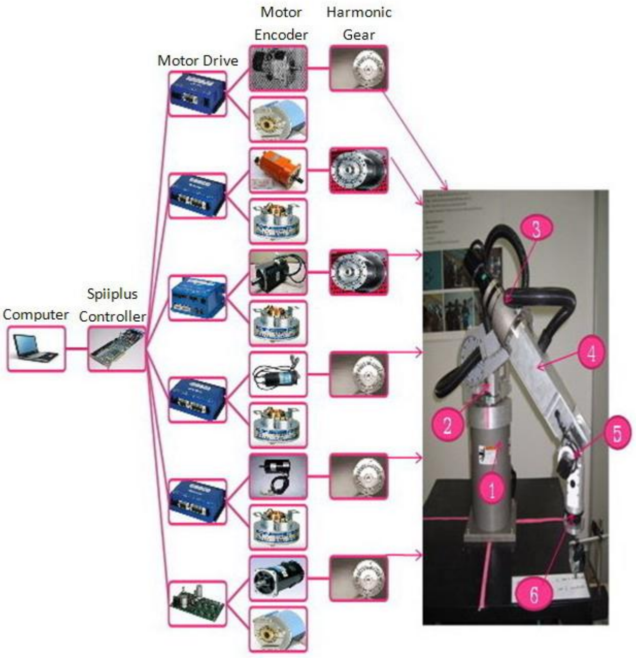


Fig. 1 Overall schematic diagram of the 6-axis manipulator arm.

2.1 Robot Mechanism Construction

Originally, this manipulator arm is designed and assembled in Solidwork, shown in Fig. 2, to test feasibilities of individual joint movement, robot material strength, motor sizing, and robot workspace. Each joint angle is limited by two limit switches such that a wire entangling problem can be avoided during the robot operation. For the joint construction, motors are attached to flanges before assembling to machined parts and then the encoders are coupled to the motor shafts. The 6-axis joint space of the robot arm is exhibited in Table.1.

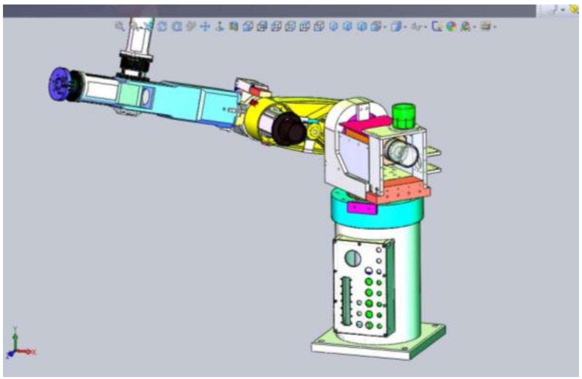


Fig. 2 Design of the 6-axis manipulator arm

To achieve high accuracy at the robot end effector within micrometer range, high resolution encoders as well as precision harmonic gears, as shown in Table. 2, must be employed in each joint of the robot arm.

Table. 1 The 6-axis joint space of the robot arm

Axis	Joint Angle (degree)	
	Clockwise	Counter-clockwise
1	145	150
2	45	115
3	110	110
4	175	85
5	85	80
6	160	175

Table. 2 The encoder resolution and gear ratio for each axis of the robot arm

Axis	Encoder Resolution	Gear Ratio
1	2048	1:78
2	2500	1:80
3	2500	1:50
4	2500	1:50
5	2500	1:50
6	2048	1:50

2.2. Controller and Drive Construction

Two main components of the motion control unit are 1) SPiiPlus Motion Controller compartment and 2) Motor Drive compartment. First, Inside the SPiiPlus Motion Controller unit in Fig. 3, there are 24V and 5V DC power supplies and SPiiPlus PCI 8 from ACS Motion Control, which acts as the robot-arm main supervisory and connects to the user's computer through ethernet. Moreover, amplify circuits using op-amp, included in this unit, help amplifying the signal from all limit switches before passing through the SPiiPlus board.

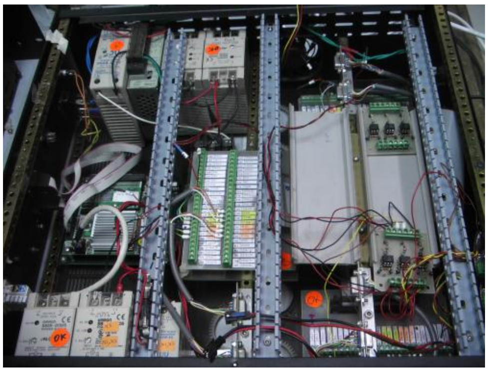


Fig. 3 SPiiPlus Motion Controller compartment

Second, the Motor Drive compartments for 6 axes in Fig. 4 and 5 are composed of power supply circuits, noise reduction circuits, 6 motor drives and connection terminals. Model for motor drives for axis 1 to 6 are Junus, Accelus, Xenus, 2 Accelus, and Accelus card [1], respectively.

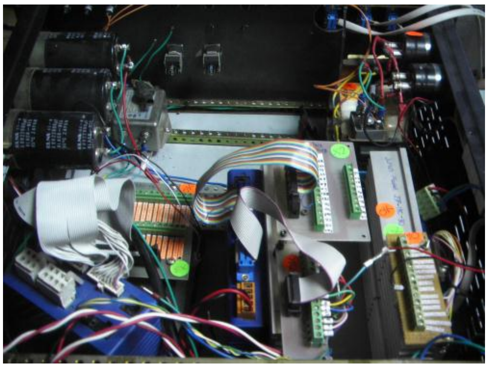


Fig. 4 Motor drive compartment for joint 1, 2, 3.

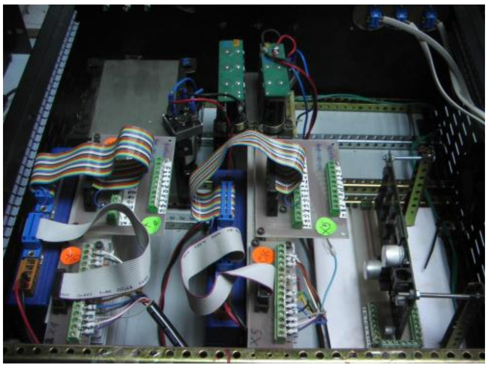
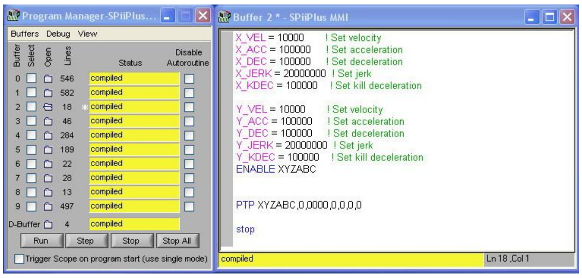


Fig. 5 Motor drive compartment for joint 4, 5, 6.

3. Motion Control Software Development

To be able to control the robot-arm motion, users must learn new syntax and commands of the original SPiiPlus Controller software, which is similar to C commands. And all programming codes must be written in a command line window shown in Fig. 6. As a result, the user must go through a steep learning curve of SPiiPlus programming before they can operate the robot arm safely and properly.



Buffer	Select	Open	Lines	Status	Disable Autoroutine
0	☐		546	compiled	☐
1	☐		582	compiled	☐
2	☐		18	* compiled	☐
3	☐		46	compiled	☐
4	☐		284	compiled	☐
5	☐		189	compiled	☐
6	☐		22	compiled	☐
7	☐		28	compiled	☐
8	☐		13	compiled	☐
9	☐		497	compiled	☐
D-Buffer			4	compiled	

☐ Trigger Scope on program start (use single mode)

```
X_VEL = 10000       ! Set velocity
X_ACC = 100000      ! Set acceleration
X_DEC = 100000      ! Set deceleration
X_JERK = 20000000  ! Set jerk
X_KDEC = 100000     ! Set kill deceleration

Y_VEL = 10000       ! Set velocity
Y_ACC = 100000      ! Set acceleration
Y_DEC = 100000      ! Set deceleration
Y_JERK = 20000000  ! Set jerk
Y_KDEC = 100000     ! Set kill deceleration
ENABLE XYZABC


PTP XYZABC,0,0000,0,0,0,0

stop
```

compiled
Ln 18 ,Col 1

Fig. 6 Original Interface of the SPiiPlus Motion Controller Softeware.

To shorten this period, the motion control software has been developed using Visual C++.NET [4] from Microsoft Visual Studio 2008 based on SPiiPlus Motion Control library that provides a user-friendly interface, as shown in Fig. 7. The motion control software for individual or combined joints employs the SPiiPlus MMI commands [2] of the SPiiPlus Motion Controller. The user can select the joint axes and specify the joint acceleration, velocity, position without learning SPiiPlus programming commands. Moreover, users can record sequences of the joint angular positions after the robot end-effector is moved to desired locations. And then, users can playback different motion sequences from recorded lists. During the robot-arm movement, the feedback angular positions of each joint are also shown on the screen in real time. Fig. 8 demonstrates the robot-arm playback movement, drawing a segment of circle, from the recorded sequence.

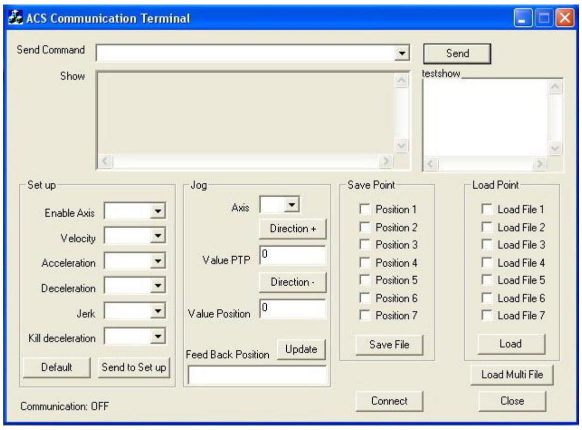

Value PTP: 0
Value Position: 0
Communication: OFF

Fig. 7 Developed motion control softeware using Visual C++.NET for this robot arm

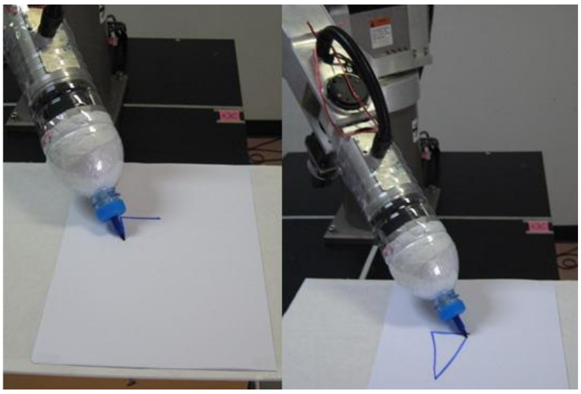


Fig. 8 Playback motion of the robot end effector

4. Experimental Results

The robot arm performances are tested on the motion repeatability, accuracy, and different load handling. In the first test, the joint velocity errors are measured

4.1 Repeatability Test

To measure a motion accuracy of the robot-arm end effector in performing a repetitive task, the robot arm is programmed to touch a dial gauge, attached at the end of table as shown in Fig. 9, from a pointing-upward position. This motion is repeated for 15 times and the readings from the dial gauge are recorded and shown in Fig. 10. The result reveals that the motion error of the end effector is bounded within 0.002 mm or 2 µm and a variation from its mean position are also small. This implies that the accumulative error from all joints must be very small as well.

Fig. 9 Motion repeatability test of the robot arm touching the dial gauge.

4.2 Load Handling Test

To measure the motion accuracy of the robot arm subjected to various loads, three different weights: 1.2, 1.6 and 2.0 kg, are attached to the end effector and both joint commanded velocity from the SPiiPlus controller and feedback velocities from encoders are recorded during the upward motion. Then the joint velocity error is computed from the difference between the commanded and feedback velocities for 5 axes. Results in Table. 3 show that the larger the load attached to the end effector is, the larger the velocity error becomes.

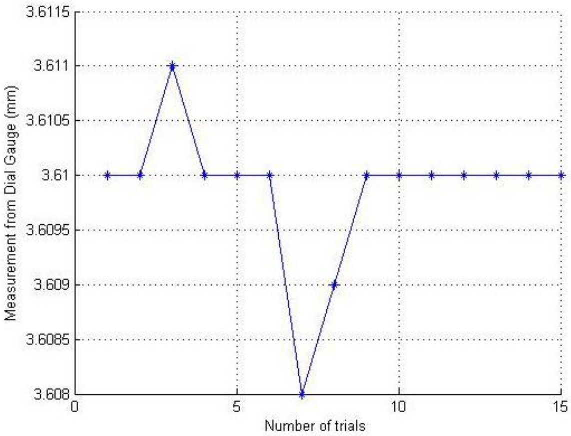


Fig. 10 Dial gauge readings from 15 trails in repeatability test

Table. 3 Joint angular velocity errors of the 5-axis robot arm tests with three different loads (1.2,1.6, and 2.0 kg) attached to the end effector

Joint	Velocity Error (rad/sec)			
	No Mass	Mass 1.2 kg	Mass 1.6 kg	Mass 2.0 kg
1	0.00079	0.00085	0.00098	0.00089
2	0.00019	0.00063	0.00021	0.00024
3	0.00002	0.00007	0.00092	0.00003
4	0.00232	0.00302	0.00258	0.00244
5	0.00004	0.00005	0.00005	0.00007

4.3 Motion Accuracy of Iterative Inverse Kinematics Computation

An inverse kinematics, based on the Resolved Motion Rate Controller (RMRC) technique [7], of this robot arm has been implemented using the Jacobian singularity-robust inverse. The RMRC technique is one of the iterative inverse kinematic methods to compute the position of end effecter given the desired joint angular position and velocity. The The joints' angular velocities in this experiment are set equal to [0.098, 0.078, 0.295, 0.295, 0.295, 0.295] rad/s. The motion discrepancy of the end effector between the commanded joint positions calculated from the RMRC technique and the measured position are tested for 8 different positions, as shown in Table. 4. The error is defined using a root sum square, as in Eq (1).

$$error = \sqrt{(X_d - X_m)^2 + (Y_d - Y_m)^2 + (Z_d - Z_m)^2} \quad (1)$$

From Table. 4, the difference between the measured and command position are very small, the main source of error in this test is from the measurement error and from link parameter error.

Table. 4 Desired and measured end effector positions using the RMRC technique.

Command Position (cm)			Measured Position (cm)			Error (cm)
X_d	Y_d	Z_d	X_m	Y_m	Z_m	
55	40	70	55.3	38.4	70.3	1.602
-55	40	70	-54.2	39.6	71.0	1.337
55	-40	70	54.3	-41.1	70.7	1.472
-55	-40	70	-54.6	-38.1	70.1	1.904
50	50	50	50.0	48.0	50.7	2.119
-50	50	50	-49.4	50.0	51.9	1.971
50	-50	50	49.3	-51.0	51.8	2.213
-50	-50	50	-49.9	-46.9	51.4	3.431

6. Summary

The 6-axis robot manipulator arm has been designed, constructed, and tested on high motion accuracy, precision repeatability, and load variation handling capability. Moreover, the motion control software, emphasized on the user-friendly interface, has been developed using Visual C++.NET such that new users can learn how to control this robot arm quickly, record the motion sequences, as well as playback those different sequences as desired.

6. Acknowledgement

The authors would like to thank the National Research Council of Thailand (NRCT) for a research grant 2553 to support this research.

7. References

[1] Copley Controls, Accelus & Junus, URL:http://www.copleycontrols.com/motion/Products/Drives/Digital/accelus-junus.html, access on 12/07/2011.

[2] ACS Motion Control, Downloads, URL: http://www.acsmotioncontrol.com/index.php?option=com_phocadownload&view=category&id=5:user-guides&start=20&Itemid=34, access on 12/07/2011.

[3] Craig, J.J. (1989). Introduction to robotics: mechanics and control, 2nd edition, ISBN: 0-201-09528-9, Addison-Wesley Publishing Company, Inc.

[4] นิรุธ อำนวยศิลป์. เขียนโปรแกรมบนวินโดส์ด้วย Visual C++ และ MFC, กรุงเทพฯ:THAIDEV.COM.

[5]Reocities,research,URL:http://reocities.com/SiliconValley/Screen/8319/research/robot/5_axis_robot_proposal.html, access on 14/07/2011.

[6] FIBO , แขนกลอุตสาหกรรม, URL: http://fibo.kmutt.ac.th/thai/index.php?option=co_content&task=view&id=302&Itemid=0, access on 12/07/2011.

[7] Prempraneerach, P. and Kulvanit, P. (2010) Implementation of Resolved Motion Rate Controller with 5-Axis Robot Manipulator Arm, *The first TSME international conference on mechanical engineering 2010*, Ubon Ratchathani, Thailand.