Retrofitting of an IBM 7540 SCARA Robot

Sara Sobh[1], Ali Harake[2], Hadi Jomaa[3], Maher Dankar[4]

Department of Mechanical Engineering

American University of Beirut

Riad El Solh, Beirut 1107 2020, Lebanon

[1]Shs13@aub.edu.lb, [2]ash18@aub.edu.lb, [3]hsj04@aub.edu.lb, [4]mmd27@aub.edu.lb

Abstract- This paper presents how the control system of an IBM7540 SCARA robot was replaced by a custom- built control algorithm. The robot can now be controlled via LabVIEW. The control program can take coordinates or manipulator joint angles as inputs. This platform can handle high level control algorithms. By implementing cross-coupling control, the motion of the joint axes was synchronized.

I. INTRODUCTION

A. Project Description

The aim of this project was to retrofit an IBM 7540 SCARA robot .The 1983 robot has been sitting in AUB's Control Lab unattended for many years. This robot is accurate, fast and extremely powerful. It is capable of performing intricate tasks with precision. Since 1983, the swift advancements in robotics and programming left this old-fashioned robot neglected and of no interest to modern day users.

The robot was originally programmed using AML (A Manufacturing Language) and only small, simple programs could be executed. The aim of this project is to replace the control system of the IBM 7540 with a custom -built control algorithm. A user-friendly LabVIEW interface was designed to control the robotic arm. The IBM 7540 can now be controlled through LabVIEW and programmed to perform desired tasks. This enables users to control the robot via a LabVIEW program. LabVIEW's simple yet remarkable programming alongside its visual user-interface makes it a great tool for teachers and students to use for programming the robot. This, coupled with the precision and strength of the robot, makes the IBM 7540 a powerful teaching tool for courses such as Automatic Control, Instrumentations, and Robotics. Retrofitting the robot will give rise to several educational opportunities for the Mechanical and Electrical Engineering departments

II. THE IBM 7540 MANUFACTURING SYSTEM

B. Technical Description

The IBM 7540 manufacturing system shown in Figure 1 consists of a four joint DC-servo actuated selective complaint assembly robot arm. We use the following nomenclature throughout the paper:

- Theta 1 (θ_1) : the angle of the first arm
- Theta 2 (θ_2) : the angle of the second arm
- Roll (R_0): the angle of the roll motor

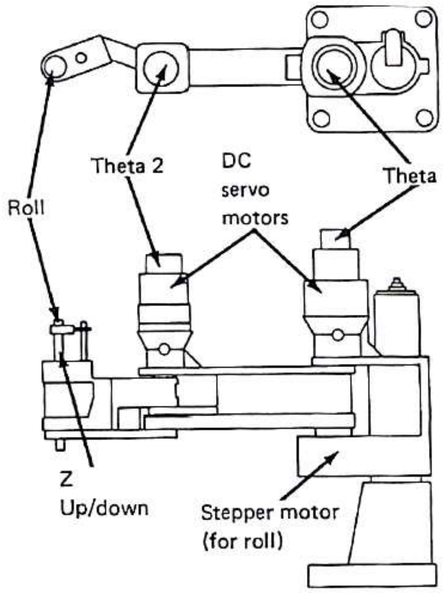

Figure 1 IBM 7540 SCARA Components
Source: IBM Manufacturing systems hardware library [1]

The control unit contains a power supply, drivers for the servo motor (Yaskawa Servopack CPCR-MR-05C Y20), a power distribution board (Relay based), and a motor control board (referred to in this report as the MTCB). The MTCB contains a microprocessor, memory, communication interface and trajectory planning circuits. The block diagram of the whole system is show below in Figure 2.

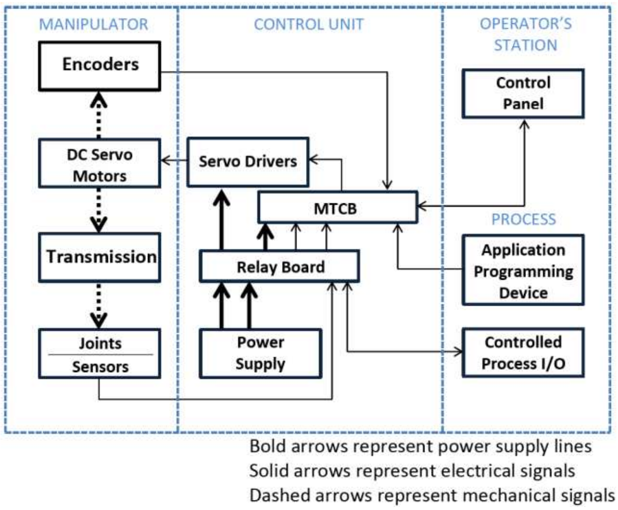

Figure 2 Block Diagram of System [2]

C. *Manipulator Kinematics*

The manipulator is a four degree-of-freedom device with rigid links. The shoulder (θ_1) and elbow (θ_2) joints are revolute joints that give the freedom of motion in the x-y plane. θ_1 can go from 0-200^0 and is measured with respect to the x-axis and $\theta 2$ from 0-135^0 and is measured with respect to the radial axis.

The Z-joint is a pneumatic prismatic joint that can go from 0 to 250mm downwards along the z-axis. The final degree of freedom is achieved with the roll joint (revolute of the z-shaft) that can rotate from 180^0 to -180^0. These ranges can be seen in Figure 3.

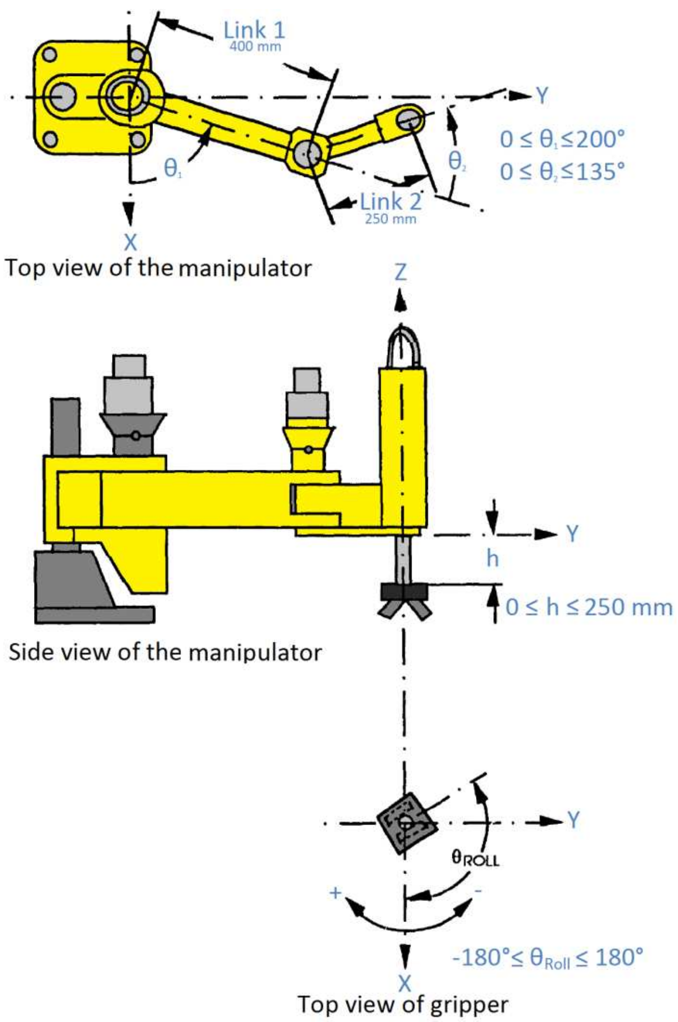

Figure 3 Limits Of Manipulator Motor Angles
Source: IBM Manufacturing systems hardware library [1]

The Forward Kinematics of the Manipulator:
(L1=400mm and L2=250mm)

$$X=L_2 \cos(\theta_1+\theta_2)+L_1 \cos\theta_1$$

$$Y=L_2 \sin(\theta_1+\theta_2)+L_1 \sin\theta_1$$

$$Z= - h$$

The Inverse Kinematics of the Manipulator:

$$\theta_1= \tan^{-1} 2(Y,X)+ \tan^{-1} (\pm\sqrt{X^2\text{-}Y^2 \text{-}\omega^2} ,\omega)$$
$$\theta_2=\tan^{-1}(\text{-}X \sin\theta_1 +Y \cos\theta_1 , X \cos\theta_1 + Y \sin\theta_1 \text{-}L_1)$$

It should be noted that the manipulator has no positional redundancies and each point in space has a unique set of joints values. The work area of the robot is shown below in Figure 4.

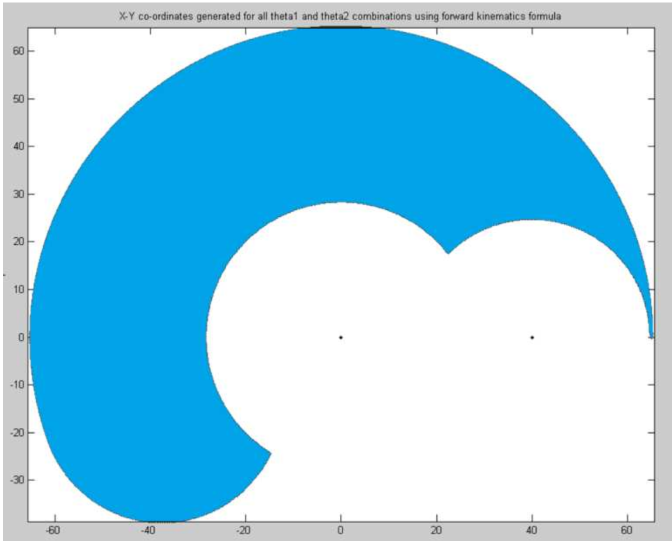

Figure 4 Work Area of Manipulator

The inverse and forward kinematics of the manipulator are those of the standard 2-link serial manipulator. The work envelop was developed using MATLAB by plotting the end effector position for all values of theta 1 and theta 2.

III. Serial Communication

The IBM controller is controlled via an RS-232 serial interface with the AML/E interface. The asynchronous communications protocol is used with full duplex transmission with a half-duplex end-to-end user protocol, 4800-baud rate, even parity, 7 data bits and 2 stop bits. At first, a 25-pin to 9-pin RS232 serial cable was made according to the specified connections from the IBM Manufacturing system manual. The connections are shown below in Figure 5:

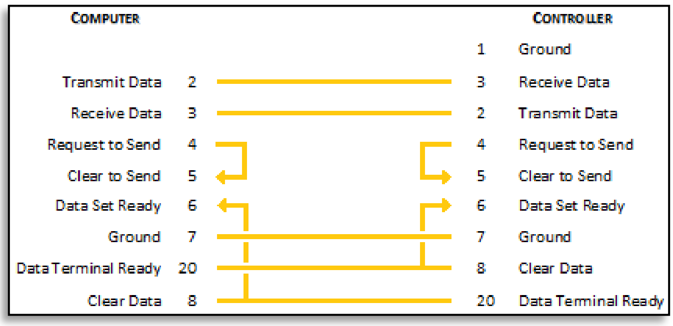

Figure 5 IBM Specified Connections for RS-232 Serial Communication

A special power up sequence was required to start the manipulator. AML/E was run on several computers with different operating systems (Windows 8/7/Vista/XP) and different connection methods (USB to serial, Direct RS232 connection). After several attempts following IBM specifications and protocols, we could not establish an online connection. The assumption made was that the serial port is malfunctioning.

IV. SYSTEM CONTROL

This sections deals with the steps taken to control the manipulator. In order to control the manipulator, we had to bypass the existing motor control board MTCB, for as mentioned in a previous section, the serial attempt failed.

The MTCB was not shut off completely since it is also used to power the optical encoders from which we were reading the position of the manipulator. Figure 6 illustrates the theta axis movement.

The objective is to control all the degrees of freedom of the IBM 7450, namely $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, and roll angle R_o .Controlling $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ underwent similar procedures.

First off, we used the optical encoders above the corresponding motors to read the angle of each. A present gear ratio in the manual provided us with an accurate equation for the angle.

- $\boldsymbol{\theta}_1$ reduction ratio is 242:1
- $\boldsymbol{\theta}_2$ reduction ratio is 157:1

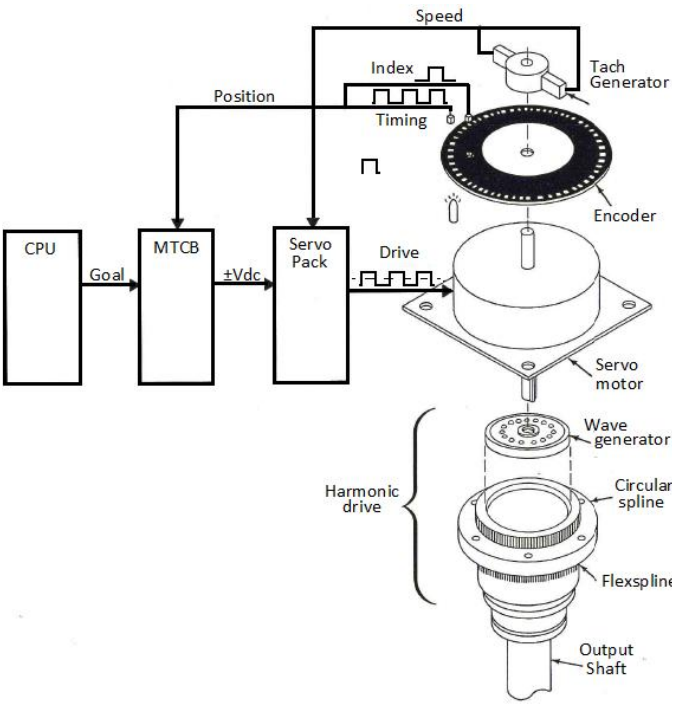

Figure 6 Theta Axis Movement
Source: IBM Manufacturing systems hardware library [1]

A. Electronics

I. Motor Drives

The motor drives were accessed directly at the level of the voltage speed reference. The value of the speed reference ranges from -10 to 10 volts but was limited to ±3 volts by our program for safety measures. The motor drives include analogue control circuitry for speed control. It employs tachometer feedback and uses PWM control to control location of the DC servo motors at each joint.

II. Encoders

The encoder signals were extracted directly from the encoders and prevented from entering the MTCB for two reasons:

- If the error measured by the MTCB between the current angle saved in the memory and the actual angle provided by the encoder signal becomes too large, the MTCB cuts off power to the manipulator.
- The MTCB counters interfere with the encoder signals giving false angle readings in LabVIEW.

The encoder provides signals at 10 volts (not TTL compatible). A signal conditioning circuit made of 4 potentiometers was built (one for each channel) to keep the encoder voltage at 5 volts.

<u>Encoder specifications:</u>

- 500 pulses/revolution
- 3 Channels
    - A&B: timing
    - Z: Index
- Signal is TTL Compatible
- Extraction point: CN12 M

III. Limit switches

Since the encoders are incremental encoders, at start-up, the manipulator must return to its home position. For this reason, the built-in home limit switch was used. The limit switch was connected to relays that are normally closed. When the robot hits the limit switch, the relay will open resulting in a voltage drop from 24 V to 0 V. This range was reduced to 5-0 V by a signal conditioning circuit (potentiometers). The same procedure was applied to the overrun limit switches which cut off power to the manipulator once the $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ angles exceed their limits.

IV. Manipulator power

The MTCB was disconnected from the CPU and is currently being used to supply power to the relays and the encoders. The manipulator power-up sequence was also altered and replaced by a push button for quick shutdown in case of emergencies.

B. Control Algorithm

First off, we used the optical encoders above the corresponding motors to read the angle of each. A present gear ratio in the manual provided us with an accurate equation for the angle. To be able to control the joint position PI control logic was utilized. First, after start-up the manipulator is sent to its home position, returns 20 degrees back and re-adjusts to re-calibrate the home position.

X and Y coordinates are inputted into a custom-built LabVIEW program by the user and equations for the inverse kinematics are used to obtain the angles θ_1 and θ_2. If the angles lie outside of the manipulator span shown in Figure 4 the program is not executed. The reference angle is compared with the current angle position read from the encoders. The error is continuously fed into the PI transfer function that relates the voltage reference with the error signal. A block diagram of the control system is shown in Figure 7. PI control is used since the system is already overdamped (evident from step response) thus it is only required to reduce steady state error which is done by adding an integrator to the system through the PI algorithm.

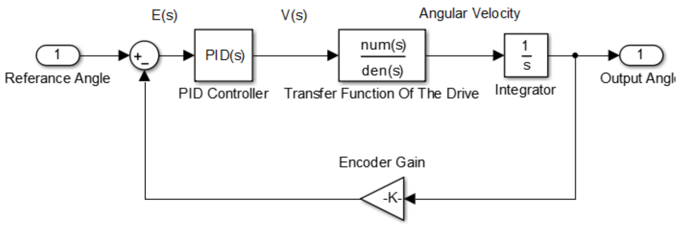

Figure 7 Block diagram of control system

C. Roll Motor

The roll motor rotates clockwise and counter clockwise by the providing a reference voltage at different ports independently. Figure 8 shows the roll axis movement. We tried to control the roll motor by removing the integrated circuit drive present in the controller, however since this is a five phase stepper motor, it would have required extensive amount of time and unnecessary rebuilding of a new circuitry. Thus, we decided to install an encoder to be able to read and control the position of the motor. The stepper motor is voltage controlled. A square signal is sent at different frequencies that vary the speed. The degree of rotation is given by the encoder.

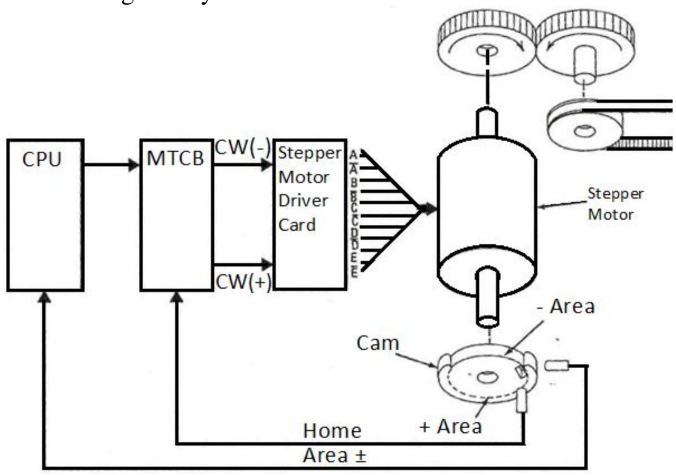

Figure 8 Roll Axis Movement
Source: IBM Manufacturing systems hardware library [1]

D. Z-axis

The existing Z-axis in Figure 9 provides limited applications. It was pneumatically controlled and the level was limited to only up and down positions with no intermediates.

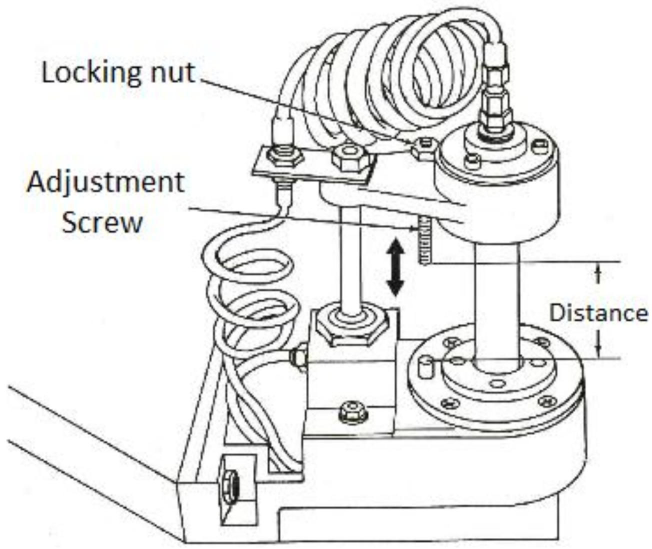

Figure 9 Z-axis Schematic Diagram [1]

To design a lifting mechanism that adds one degree of freedom to the IBM robotic arm along the Z-axis we chose a ball screw mechanism.

<u>The specifications of the mechanism:</u>

Screw Orientation: Vertical

Load Supported vertically: Coefficient of friction equals 1

Sliding Load is 56.84 lbs. Max (weight of machine parts to be lifted)

Stroke Length: 9 cm

Travel rate: 0.02 cm per minute ⇔ 1 cm per second (Max.)

Duty Cycle: 20 cycles per hour, 15 hours per day, 250 days per year (3 terms)

<u>Required life L_d (cm):</u>

L_d = 9 cm/stroke * 2 strokes/cycle * 20 cycles/hour * 15 hours/day * 250 days/year = 1,350,000 cm

The weight of the sliding load is nearly 5.8 Kg * 9.81 m/s^2 = 56.84 N

The F_p force component can be taken as the result of a load of 2 kg carried by the gripper = 19.62 N

We get P = 76.46 N total applied dynamic load

Expected Service life (single nut)

Life under actual load (L_{rev}) = $\left(\frac{C}{P}\right)^3 * 10^6$

Where :

C is the Rated load used to choose a screw at 10^6 rev

P is the actual load

Conversion from Life in travel distance to revolutions:

$$L_{rev} = \frac{L_{distance}}{lead}$$

If we multiply both ends of the main equation by lead we can directly use Life in travel distance independent of lead we get

$$C = P * \left(\frac{L_{distance}}{10^6}\right)^{\frac{1}{3}} = 76.46 * \left(\frac{13{,}500{,}000}{1000000}\right)^{\frac{1}{3}} = 182.06\ N$$

Ball Screw Selection:

Load Rating: Requires Ball Screw Operating Load Capacity of 182.06 N Minimum

For our application we need high precision positioning so we will based our calculations on a lead of 5 mm

Remark:

The required life estimation and the following calculations are based on an efficiency factor of 90% with which most screws are rated.

The driving torque needed to obtain the thrusting force

$$T = \frac{P * l}{2 * \pi * \eta} = \frac{76.46 * 0.005}{5.65} = 0.0677\ Nm$$

V. RESULTS

The control system of the IBM 7540 SCARA robot has been replaced with a custom-built control algorithm. The robot can now be controlled using a LabVIEW program. Users can now input coordinates or angles of motors along with a desired speed and the manipulator will smoothly move to the specified location. There exists a minor error in the manipulator's position of magnitude 0.0003 degrees.

VI. CONCLUSIONS AND FUTURE WORK

The IBM 7540 manipulator was provided enough processing power and flexibility to be used as a platform for advanced robot control strategies. From the software point of view, we have provided the basic building blocks for most low level tasks used in the implementation of our control algorithms. We showed that this platform could handle high level control algorithms by implementing cross-coupling control to synchronize the motion of the axes. In terms of hardware, modifications can be made to the Z-axis to add a degree of freedom to the manipulator. An optical encoder could be added to the roll axis to provide closed loop control over the roll stepper motor. A new gripper for the arm can also be designed. In its current state, the platform can be incorporated into the robotics and control courses at AUB. The retrofitted IBM 7540 can be used for research to develop and test new control strategies. New labs such as 3D motion control, visual servoing, and others can now be taught at AUB.

ACKNOWLEDGMENT

Our group would like to thank our advisors Dr. Daniel Asmar and Dr. Elie Shammas for their continuous support and guidance. We would also like to thank Mr. Ghassan Deeb for his help.

REFERENCES

[1] I. B. M. Corporation, IBM 7535/7540 Manufacturing System - Hardware Library, 1983.

[2] B. Jonathan Neil, "The Design and Implementation of an Advanced Robot Controller," Department of Electrical and Computer Engineering- Concordia University, Montreal, 1990.