Implementation and control of three degrees of freedom robot inspired by Viper 650

Mostafa Hamidifard, Mahshad Aliyan, Fateme Karrar, Sina Rashidizade, Mohammadsadegh Najafi, Maryam Heydari
Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran

***Abstract*—In this paper, a three-degree-of-freedom robot is designed and implemented. In the following, the steps of making and assembling the robot are explained, and the control of the robot is done using Arduino and MATLAB. finally, the appropriate solutions for better control and implementation challenges are described.**

Keywords—robot, control, Arduino, MATLAB, implementation

I. Introduction

The Viper 650 robot is an articulated robot with 6 degrees of freedom used in machining, assembly and material handling. High resolution, using of absolute encoder and high precision, driving with low inertia and light weight arm to provide maximum acceleration are among the advantages of this robot.

In this paper, we implemented a three-degree-of-freedom robot inspired by Viper, In which in the place of each of the joints of the designed robot, the motor plays the role of rotating the links.

Next, we will discuss how to control the motors using software and other elements so that the robot tracks our desired path.

Finally, it is observed that the robot has the ability to follow the sinusoidal path with low error.

II. Robot Structure

We first designed a three-degree-of-freedom robot in the SolidWorks software.

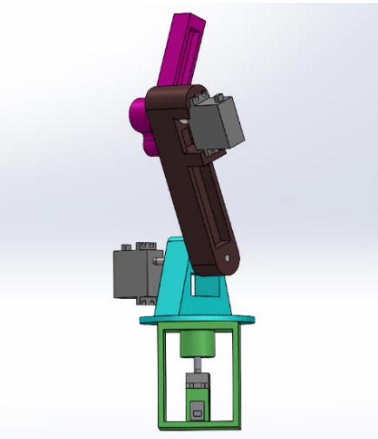

Fig.1-The final designed robot in SolidWorks environment

Then we printed the links and the location of the motors used in the joints using STL file in Cora software.

Three servo motors have been used in the robot.

Also we used three 626ZZ Deep Grove Ball bearings, 6mm x 6mm Rigid coupling and shafts to connect the robot components.

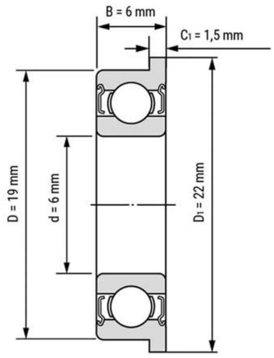

Fig.2-Schematic of groove ball bearing

The shafts used in the robot are 6 mm thick and 8 cm long.

The final shape of the assembled robot is as follows:

Fig.3-Assembled robot in zero configuration

In this case, we used MG996R model of servo motor.

For better performance, before installing the motors, we opened them and lubricated them with servo grease, which reduces the noise of the gears.

The actual sizes of the motors have been measured using calipers and the design has been made based on these sizes.

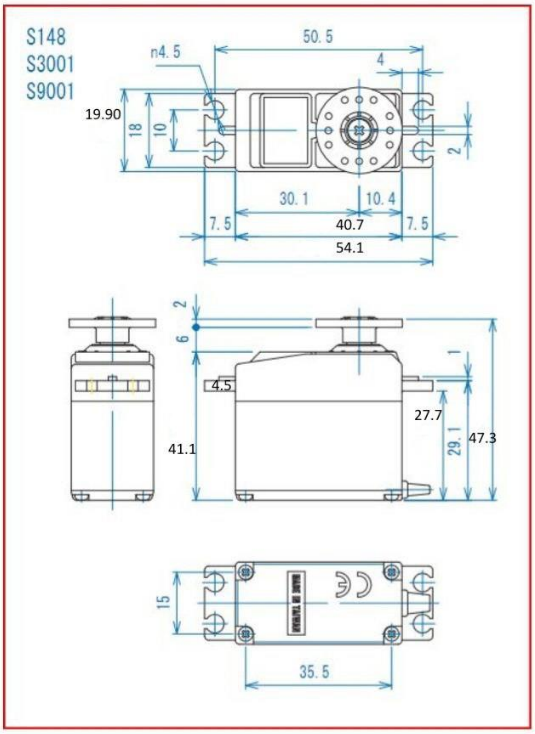

Fig.4-Servo motor dimension

Inside each motor there is a board connected to a potentiometer. To calculate the error signal in MATLAB, feedback was taken from the potentiometer inside the motor using wire.

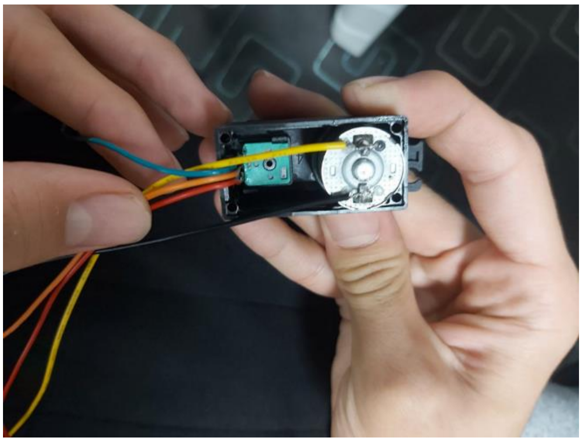

Fig.5- A view of the potentiometer and the interior of the motor

III. CONNECTIONS AND STARTING WITH MOTORS

Five wires are connected to each motor.

The feedback signal taken from the potentiometer which is placed in the motor is connected respectively to the pins A0, A1 and A2 of Arduino UNO for each motor.

Also the potentiometer needs 5v which is provided by connecting to the 5v pin of Arduino.

The motor's and other elements' ground are connected to the ground.

Since Arduino could not provide the required power and current of the motors, we used L298.

Both ends of the rotor are connected to L298, which is used to increase the power of the motor. The source required for L298 is 12 volts with a current of 2 amps.

We used a 12V voltage adapter to start the L298.

The other three pins of L298 are named EN, IN1 and IN2, where the EN pin is connected to the Arduino PWM wave generator pin. The other two pins are connected to digital pins.

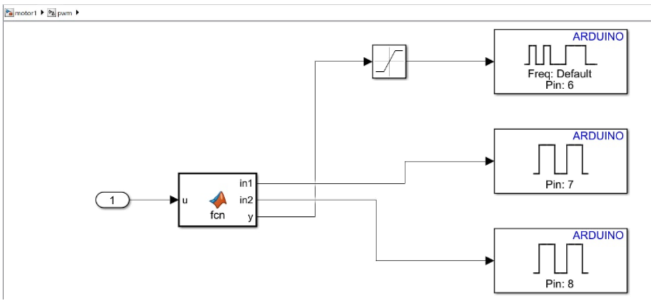

Fig.6-PWM subsystem used in Simulink MATLAB

IV. MOTOR CONTROLING USING MATLAB AND ARDUINO

At first we need to install Simulink Support Package for Arduino Hardware in MATLAB due to the connection between MATLAB and Arduino.

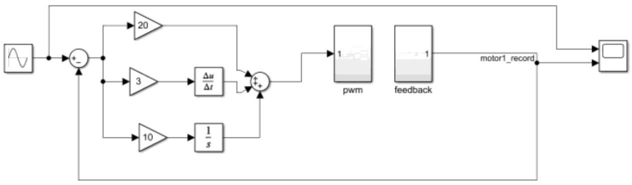

Fig.7-Overview of Simulink

There are two PWM and Feedback subsystems inside Simulink.

The feedback taken from the potentiometer connected to pin A, which is the analog pin of Arduino, is transferred to MATLAB through Arduino.

Considering that Arduino is ten-bit, the target number is a number between 0 and 1023, which is mapped to 0 to 360 degrees in feedback subsystem.

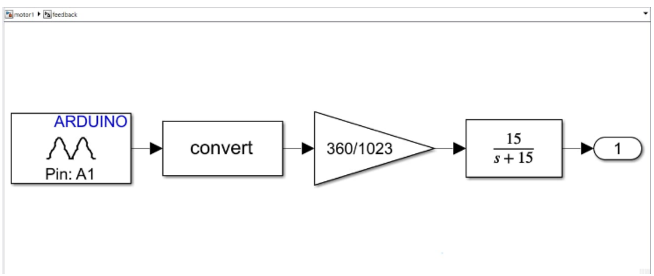

Fig.8- Feedback subsystem used in Simulink

As a result, the motor angle is obtained from the feedback signal.

At first, we give the desired signal that we want to be followed by the robot, such as a sinusoidal signal, as an input to the control loop.

By using the feedback signal, the angle error is obtained, which goes to the PWM subsystem after the PID controller.

The difference between the desired angle and the motor angle determines the direction of rotation.

Finally, it was observed that the desired 1 rad/s sinusoidal signal is well tracked by motors.

V. Robot Controlling Using Inverse Kinematics

By using inverse kinematics of the robot, we can make the End Effector to follow a desired path.

In MATLAB, the inverse kinematics code calculates the angle of each joint in real-time.

The calculated angles are taken as Simulink input in the previous section.

VI. Simulation and Controlling by Inverse Dynamics

At first, the axes of links are specified separately in the SolidWorks software.

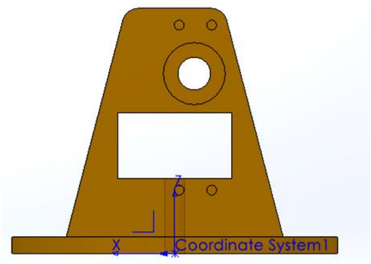

Fig.9-The axes of Robot's Link - Part 1

After specifying the axes of each part in SolidWorks, the inertia matrix is obtained.

$$I_{Link_1} = \begin{bmatrix} 384957.06 & -2691.73 & 43626.55 \\ -2691.73 & 180695.89 & -19024.35 \\ 43626.55 & -19024.35 & 309313.62 \end{bmatrix}$$

$$I_{Link_2} = \begin{bmatrix} 5886595.04 & 330234.57 & 368561.17 \\ 330234.57 & 4128626.67 & 2824947.69 \\ 368561.17 & 2824947.69 & 2826241.18 \end{bmatrix}$$

$$I_{Link_3} = \begin{bmatrix} 179937.28 & 34003.18 & -94339.01 \\ 34003.18 & 207856.24 & -34453.66 \\ -94339.01 & -34453.66 & 157860.07 \end{bmatrix}$$

Link	Mass (g)	Center of Mass
1	168.13	$\begin{bmatrix} -2.47 \\ 32.05 \\ -9.31 \end{bmatrix}$
2	167.87	$\begin{bmatrix} 16.87 \\ 116.62 \\ 144.30 \end{bmatrix}$
3	45.54	$\begin{bmatrix} 43.73 \\ 21.06 \\ -45.00 \end{bmatrix}$

Table I- Data used in calculation of the robot's dynamics

By having the inertia matrix, the dynamics of the robot is obtained by calculating the matrices C, G, and M using the Lagrange method in MATLAB.

These matrices are considered as the dynamics of the built robot, in MATLAB function.

According to the figure 10, the desired angles are entered as input in MATLAB Simulink. After passing through the inverse dynamics block, the torque of the robot is obtained through the Eequation 1:

$$\tau = M\, a_q + C\dot{q} + G$$

Equation 1

In which the a_q is according to the Equation 2:

$$a_q = \ddot{q}_d - K_0(q - q_d) - K_1(\dot{q} - \dot{q}_d)$$

Equation 2

q represents the angel and q_d is the desired angel. By changing the coefficients K_0 and K_1 the output can be controlled.

Finally, using the equation 3, the angle of each joint is displayed in the output.

$$\ddot{q} = M^{-1}(\tau - C\dot{q} - G)$$

Equation 3

The output is as follows:

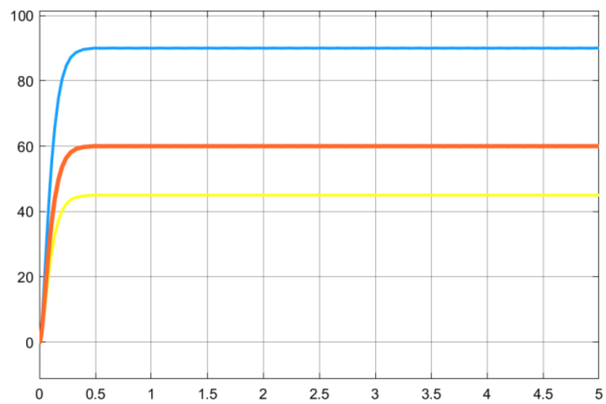

Fig.11-The result of the simulation

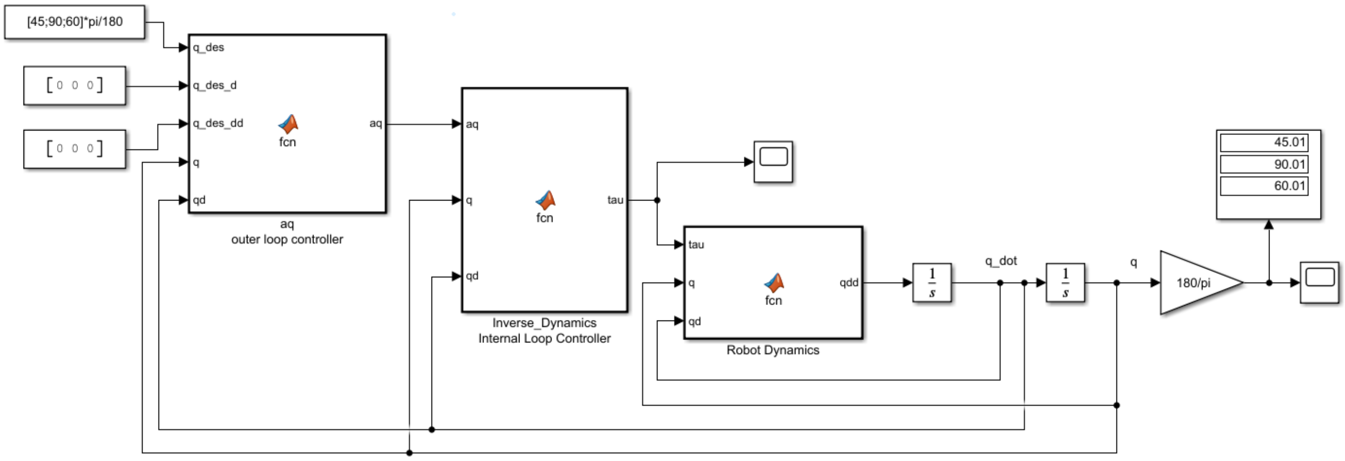

Fig.10-Simulink of the robot's simulation

VII. Conclusion

In this project, we implemented a robot and controlled the position of the motors used in the robot in order to track the desired path with high acceleration and minimum delay.

We printed the robot using ABS filament. The advantages of 3D printer ABS filament are that this material has the ability to melt and cool without changing much of its chemical properties. Therefore, it is considered an ideal filament for 3D printers. This filament is famous for its strength, hardness and high durability. It also has acceptable resistance to surface scratches, heat and common chemical corrosives. Therefore, ABS filament is suitable for printing objects subject to depreciation or pressure. When you press a strand of ABS filament, it bends and deforms before breaking, while PLA filament breaks quickly.

Also in MATLAB Simulink the feedback signal is passed through a filter Before entering to the control loop. Although the presence of the filter causes delay, we use the filter unnecessarily because of the wires and the noise of the potentiometer.

References

[1] Mark W. Spong, Seth Hutchinson and M. Vidyasagar, Robot Modeling and Control, 1st ed, Dec 12, 2022.

[2] John J. Craig, Introduction to Robotics Mechanics and Control, Third Edition, Pearson Education International.