Robot Axis Control Using a Differential Learning Algorithm

Brook Abegaz*

*Loyola University of Chicago
1125 W Loyola Avenue, Chicago, IL, USA, babegaz@luc.edu

ABSTRACT

The motion control of robotic arms using brushless DC motors has been implemented using various mechanisms, including a model predictive controller (MPC), a fuzzy controller and supervised machine learning methods. In order to provide a more efficient means of navigation, a new type of unsupervised learning mechanism based on a novel differential clustering algorithm has been designed. The differential clustering control method is useful for real time motion planning of robots and autonomous systems, where standard geometric and topological models may not work as desired due to the large operational complexity of such autonomous systems. The control mechanism has been implemented for a six degrees of freedom robotic arm and the axis control results were compared with the results of implementing MPC, fuzzy, neural network and a supervised classification based machine learning control approach. It was observed that the clustering based robotic arm control method provides comparably superior control of the torque (Nm), the speed (rpm), the angle (rad) and the angular speed (rad/s) of the robotic arm.

Keywords: robots, control, unsupervised-learning, machine-learning, motors

1 INTRODUCTION

This paper deals with the design of a control system for a robotic axis using brushless DC motor drive in order to find a decent system to guide the motion of a robotic arm. The six degrees of freedom robot (6-DOF) has three DC motors that control the movement of the arm with rotation from the motors around the axes. The motors manipulate the arm in such a way that it could move into a variety of positions in a plane. Two motors directly manipulate and rotate the axes in the arm, and the third motor rotates the base of the arm, turning it so it can work in other planes. The motors turn the axes via speed reducers by using belts to transfer the torque of the moving motors to the joints.

A general purpose 6-DOF robot shown in Figure 1 has been studied in literature [1], [2], [3] consisting of six axes of motion in the θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6 directions. The three axes are related to the arm positioning and the other three axes are related to the orientation of the robot end.

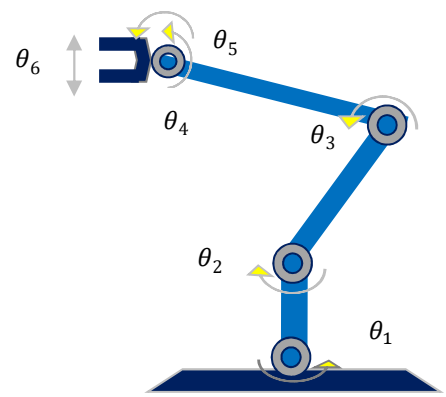

Figure 1: A general purpose 6-DOF robot

In order to effectively manipulate the robotic arm, control systems must be in place to allow the arm to be managed. This paper deals with this research problem and presents competing control mechanism for the efficiency and performance of the robot arm.

2 THEORIES AND METHODS OF CONTROL

The 6-DOF robot shown in Figure 1 consists of brushless permanent magnet synchronous motors and pulse width modulator (PWM) based converters. The three arm motors and the end motor are rated at 0.5kW.

Three control loops are used in series in the model. The first control loop is a position control loop, the second one is a speed control loop and the third one is a current control loop. Sensors that measure the position and the speed of the rotor in the synchronous motor of the robot are used to provide feedback to the control loops.

A pulse-width modulated inverter is used to control the three phase voltage and current of the brushless permanent magnet synchronous motors as shown in Figure 2. The motors then run the robot structure in coordination to get the desired final position of the robotic arm.

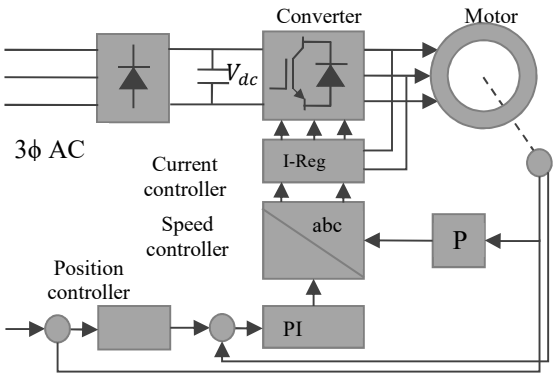

Figure 2: PWM control the three phase voltage and current of the brushless permanent magnet synchronous motors

2.1 Positional PID Control

The position system relies on two different sections, the trajectory generator and the actual positional proportional integral derivative (PID) controllers. The trajectory generator is a system that take the approximate value of what the next theta should be over each discrete instance of time as given in equation 1. It generates reference angles for each direction$\theta_{1,ref}$ $\theta_{2,ref}$ $\theta_{3,ref}$ $\theta_{4,ref}$, $\theta_{5,ref}$ and $\theta_{6,ref}$ directions.

Then, the position control section of the system takes the reference directional information and puts it into six position controllers, one for each axis. The position controller takes the projected trajectory of angles and compares it to the current angle and obtains their difference to see how far the angle needs to change. The current angular velocity of that motor is also factored into the control system to account for the fact that it is not stopped at any random point. These values are then multiplied by predetermined gain values to better control the output, and are then summed together and get filtered to get the output speed reference, $N_{1,ref}$ as given in equation 2. The output is fed back into the system of the robotic arm which then changes the current angle. With testing, it was found that amplifying differences in theta by a proportional gain K_p of 32000 and amplifying the angular velocity by a derivative gain K_d of 800 at the same time results in an effective gain for the positional PID system to keep up with the generated angles.

The effective torque for each motor is calculated using the value of the angle θ_i, the inertia of the motor J_i, the Coriolis force coefficient C_i, and the gravitational coefficient G_i as given in equation 3.

$$\theta_{i,ref} = t_{clk}\left(\frac{3(\theta_{i,final} - \theta_{i,initial})}{t_{final}}\right) \quad t_{clk} < t_{final} \tag{1}$$

$$\theta_{1,ref} = \theta_{1,final} \qquad t_{clk} \geq t_{final}$$

$$N_{i,ref} = (\theta_{i,ref} - \theta_i)K_p + (\omega_i)K_d \tag{2}$$

$$T_i = J_i\frac{d\theta_i^2}{dt^2} + C_i\frac{d\theta_i}{dt} + G_i\theta_i \tag{3}$$

By setting up trajectories of each angle to follow, and by effectively tracking the current position, the actual angle and the angular velocity, the control mechanism guides where the robotic arm needs to be steadily.

The inputs of the robot axis control include the initial position in angles, where θ_1 equals -30^0 and θ_2 = -45^0. The desired final position were set as θ_1 equals +30^0 and θ_2 = +45^0. The desired final time is set as 1.5 seconds. The system variables included the speed and the torque of the rotor, where as the output was recorded in terms of angles. The trajectory generator, the position controllers and the speed controllers have formulas that relate the input variables, the system variables and the output variables.

3 LEARNING ALGORITHMS

3.1 Fuzzy Logic Control

In attempting to create a control system to manipulate the arm, a fuzzy control system was created to see how simplifying the processes in the arm would affect the movement of the motors. The fuzzy logic controller was created with two inputs and two outputs as shown in Figure 3. The inputs consisted of two inputs for the thetas and the outputs were the resulting speed references of the respective axes. The outputs relied only on one of the inputs each and they functioned in the same way logically, but with different ranges of values of the inputs and outputs.

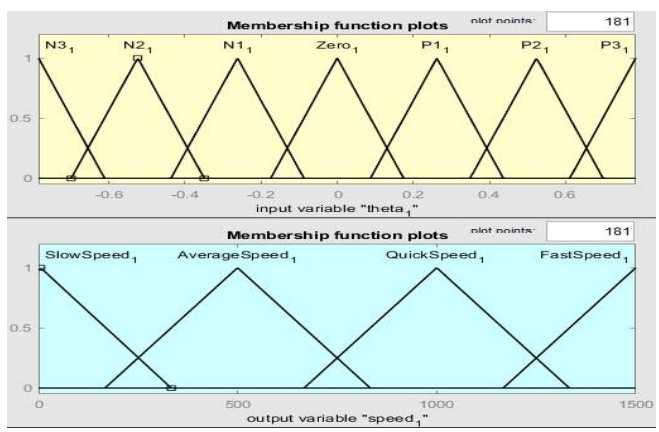

Figure 3: Fuzzy membership function plots

The fuzzy inputs are fuzzified with the angles having several membership functions separated as sections. These functions are taken in their relation to how close they are to zero and put into the output membership functions, with the values closer to zero going to membership functions that result in higher total values to generate the speed references as shown in Figure 4. This approach provided a controlled start and slow down for the motor as opposed to impulsive starting and ending, which should not occur in actual operation.

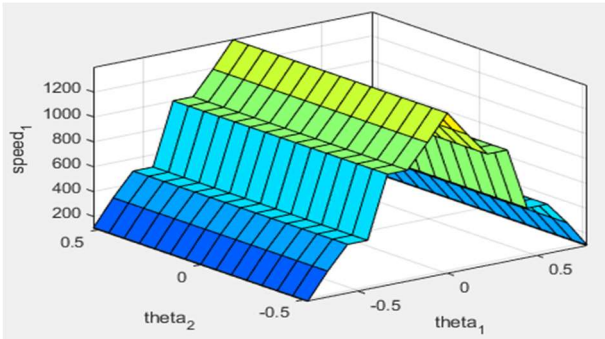

Figure 4: Input-output relations of the fuzzy controller

3.2 Neural Network Control

The neural network based control was implemented by training the robot motor controller using cost horizon of 2 seconds, a control horizon of 2 seconds, and control weighting factor of 0.05 or 5%. The sample time was 1 second and the training samples were 100. The Levenberg-Marquardt backpropagation was used for training the controller, and various minimization routines were used for backtracking.

3.3 Differential Learning Control

Although other learning algorithms and machine learning controllers have been used in conventional robot controllers in the past, the use of differential learning algorithm is novel method of robot control that is proposed in this paper. The differential learning algorithm groups the output voltage of the converter connected to the motors of the robot arm into at least three clusters based on their correlation. Then, based on historical information on the implementation of the other learning algorithms, the clustering coefficients and indices, the switching frequencies of the PWM are adjusted. The detailed description of the learning algorithm is shown in Algorithm 1, where the output is first converted to a time-series and then, it could be grouped into switching clusters.

Algorithm 1: Differential Learning Algorithm

```
Input:Voltage (vout)
Output:Clustering Index (idx);Coefficient (y)
  voutData = vout.data;
  VoutMatrix =
  zeros(length(voutData),length(voutData));
  VoutTimeseries =
  zeros(size(vout,1),size(vout,1));
  for m = 1:length(voutData)
      VoutMatrix(m,:) = voutData;
  end
  VoutTimeseries = timeseries(VoutMatrix)
  function y = fcn(VoutTimeseries, vout)
  u2 = zeros(12,1)
  u2(1:11,1) = VoutTimeseries;
  u2(12,1) = vout;
  idx3 = kmeans(u2,3);
  idx = cluster(u2,3);
  y = idx3(12);
End
```

4 IMPLEMENTATION AND RESULTS

The simulation of the robot axis control mechanisms of the fuzzy control, the neural network control and the differential unsupervised learning based control are shown in Figure 5.

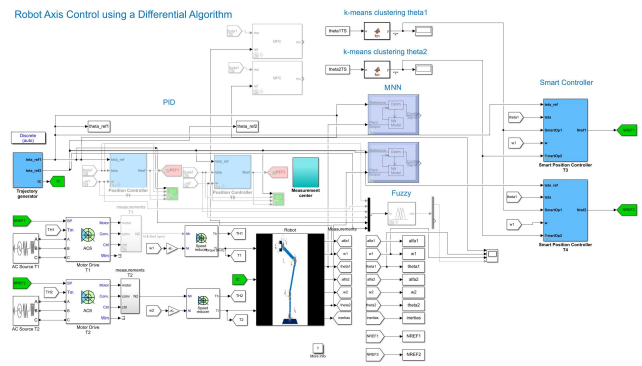

Figure 5: Simulation of the robot axis control mechanism

A comparison of the robot motor three phase voltages, currents, torque and speed values for the control approaches is shown in Figure 6, for torque, speed and angle values generated as shown in Figure 7.

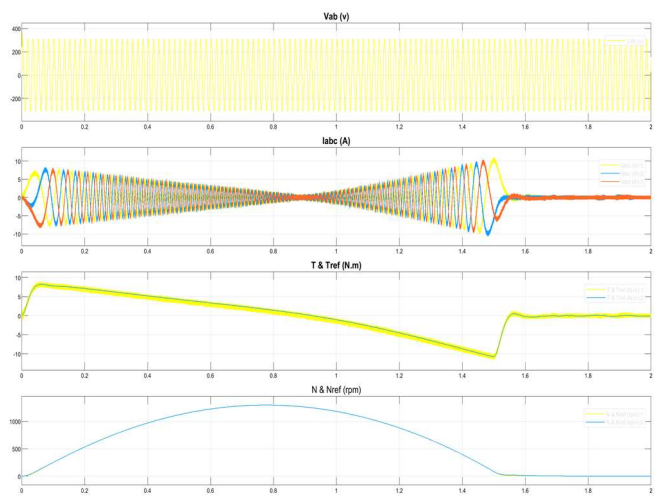

Figure 6: Comparison of the robot motor three phase voltages, currents, torque and speed values for the control approaches

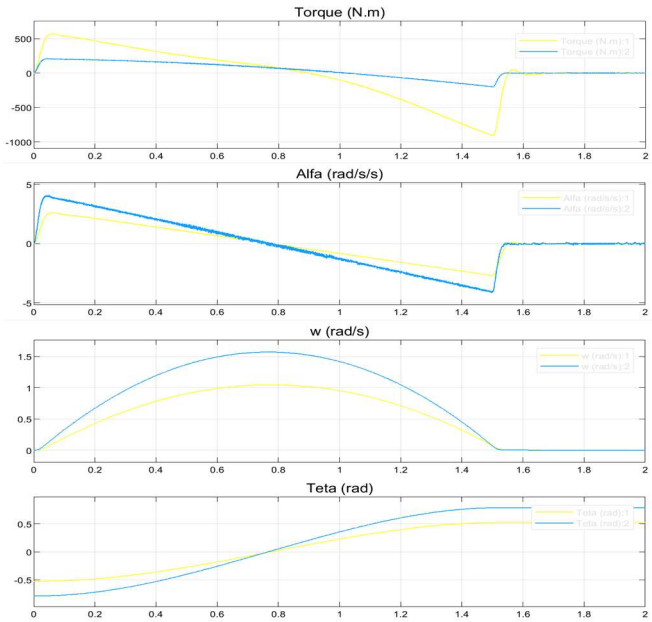

Figure 7: Generated torque, speed and angle values

5 DISCUSSION

The differential learning algorithm results in torque (Nm), the speed (rpm), the angle (rad) and the angular speed (rad/s) of the robotic arm that closely follows the generated trajectory values. When testing the fuzzy controller method against the differential learning control method, fluctuations on the system could be seen, but not too efficiently. The slopes of the output reference speed increase as theta approaches zero, and after it crosses past zero, the slope begins to decrease. This shows that the membership functions of the fuzzy controller work as intended. However, it does take a few microseconds of time for this to take place as the speed starts relatively slow and the slope does not increase too much overall even when theta equals zero before it begins to slow down as shown in Figure 8.

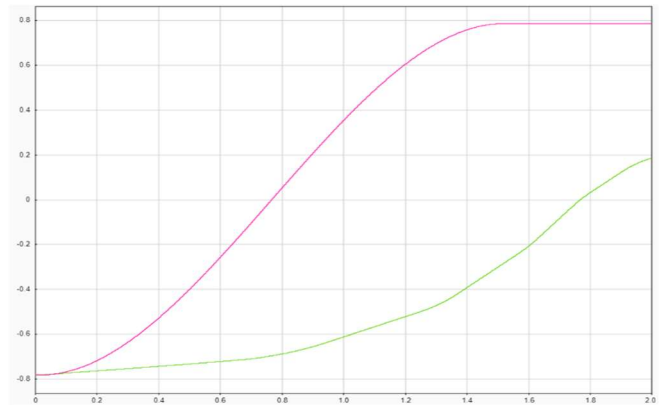

Figure 8: Plot of change in angle between the fuzzy control in green and the differential learning control in pink

6 CONCLUSION

Through observation of the results, it is clear that the differential learning system is more efficient in terms of the torque (Nm), the speed (rpm), the angle (rad) and the angular speed (rad/s) values generated closely to the desired values than the competing approaches such as the fuzzy system, the neural network system or the model predictive system. These methods would result a faster change in position and angle of the robotic arm system that enables to efficiently control the brushless DC motors in coordination. The control approach could have applications in industrial processes where robots are used to perform well-defined movements and automated tasks.

REFERENCES

[1] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Dynamics and Control, New York: John Wiley & Sons, Jan 2004.

[2] T. J. E. Miller, Brushless Permanent-Magnet and Reluctance Motor Drives, Oxford, UK: Clarendon Press, 1989.

[3] Mathworks, "Robot Axis Control Using Brushless DC Motor Drive," The MathWorks, Inc., 2018.