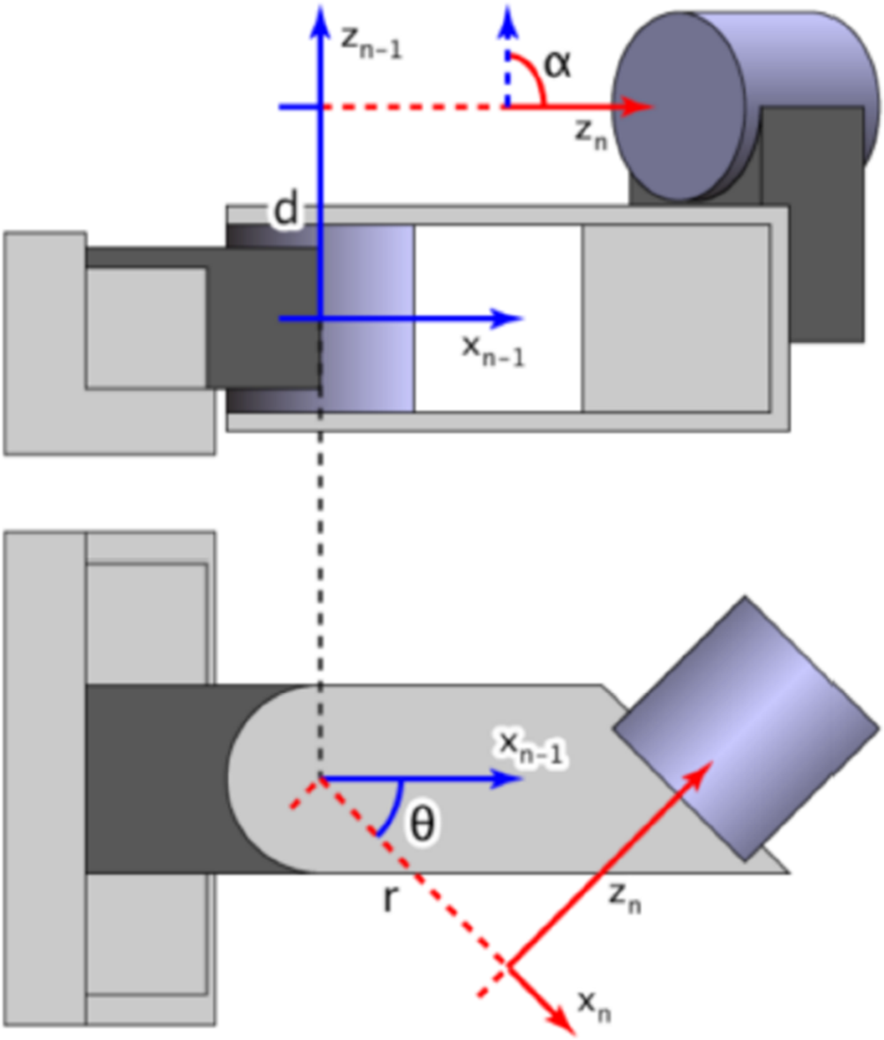
Solution to Provide the Inverse Kinematics Calculations (D-H Parameters)

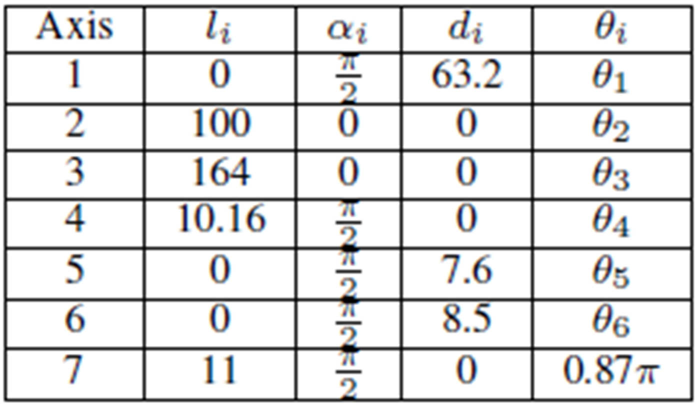
For the purpose of provide the solution to the Inverse Kinematics calculations we use a Denavit-Hartenburg parameter table.

A commonly used convention for selecting frames of reference in robotics applications is the Denavit and Hartenberg (D-H) convention which was introduced by Jaques Denavit and Richard S. Hartenberg. In this convention, each homogeneous transformation is represented as a product of four basic transformations. The common normal between two lines was the main geometric concept that allowed Denavit and Hartenberg to find a minimal representation.



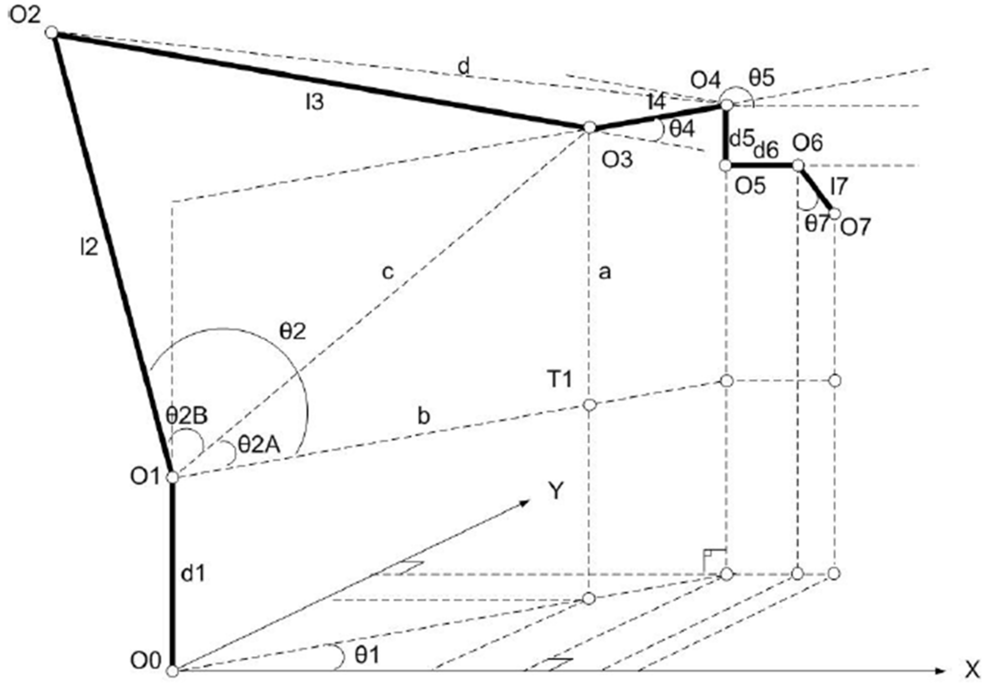
Denavit-Hartenburg Parameter Representation

The table used in building the bot is shown below:



Axis	l_i	α_i	d_i	θ_i
1	0	$\frac{\pi}{2}$	63.2	θ_1
2	100	0	0	θ_2
3	164	0	0	θ_3
4	10.16	$\frac{\pi}{2}$	0	θ_4
5	0	$\frac{\pi}{2}$	7.6	θ_5
6	0	$\frac{\pi}{2}$	8.5	θ_6
7	11	$\frac{\pi}{2}$	0	0.87π

With some constraints, if the end-point is given then we will obtain the appropriate angle 1 to 6. For example byconsidering that the end-point orientation is kept constant (7 and 6 are fixed), if O7 is given then O6, O5 and O4 are known easily. Using O4 and trigonometry rule, 1 and 5 then can be computed. Finally if 1 is known then O3, 2, 3 and 4 can be computed as well by trigonometry rule.



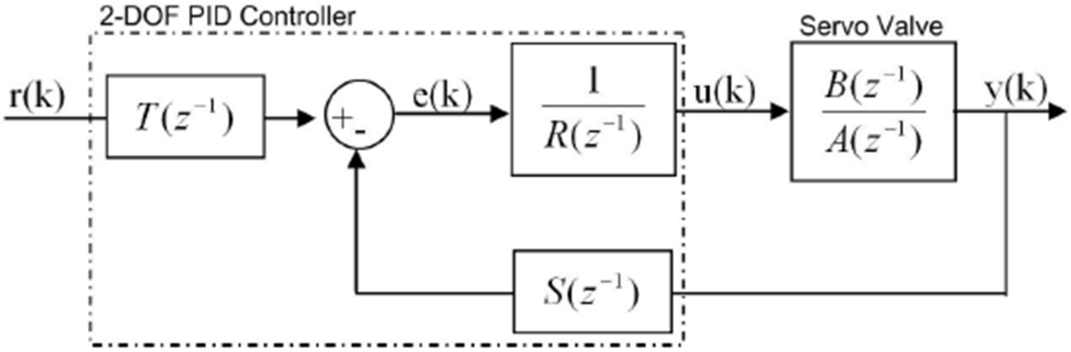
Inverse Kinematics Closed form solution

Identification and Controller Design

The servo valve in every axis is identified using Recursive Least Square (RLS) Algorithm. Pseudo-Random Binary Sequence (PRBS) signals are given as the inputs and the outputs are recorded. This recorded data will be processed by RLS to obtain $A(z^{-1})$ and $B(z^{-1})$ as shown in (1). For the reason of simpliciy, 2nd order model is chosen. However it yields good enough estimation for the controller design in section V-C.

$$H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \qquad (1)$$

Conventional 2-DOF PID controller is designed with the estimated model which its roots have been determined previously. The PID controller structure is described in the below figure. By solving (2) and (3), the controller coeficients of $R(z^{-1})$, $S(z^{-1})$ and $T(z^{-1})$ are obtained as shown in Table where $D(z^{-1})$ is a reference polinomial with defined roots.



Structure of 2-DOF PID Controller

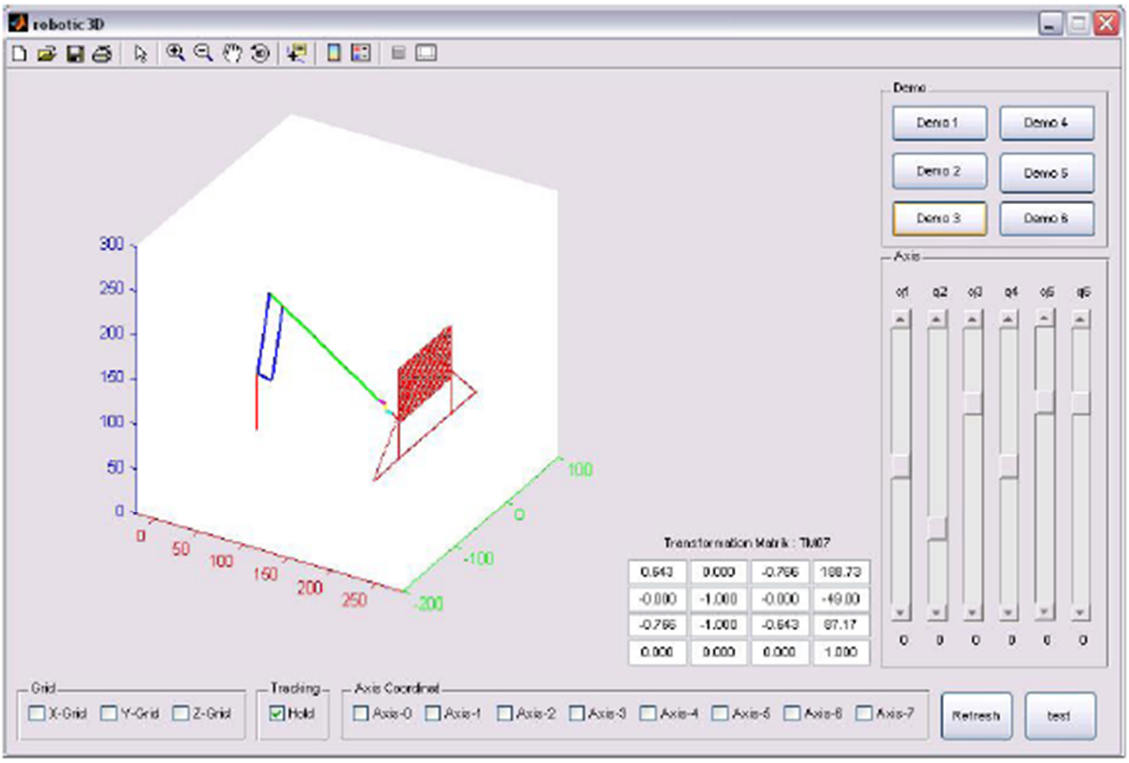
$$A(z^{-1})R(z^{-1}) + B(z^{-1})S(z^{-1}) = D(z^{-1}) \qquad (2)$$

$$T(z^{-1}) = S(z^{-1})|_{z=1} \qquad (3)$$

Results of the Experiment, Simulation and Discussion

The inverse kinematics simulation is done before loading the trajectory file to the microcontroller memory. This simulation is performed by Matlab. In this experiment, 3 trajectory patterns are generated and simulated i.e. square, triangle and zig-zag. To generate this trajectory files, some points must be defined in cartesian coordinat. For example, points P1(160,- 50,180), P2(160,-50,130), P3(160,50,130) and P4(160,50,180) are determined as a via points to generate square trajectory pattern in cm unit length. Every 5 cm movement in any direction, the cartesian coordiante will be converted into polar coordinate using closed-form inverse kinematics calculation.

Coressponding polar coordinate data which has been converted to digital value will be collected and stored in an excel format as trajectory file. This Trajectory file should be simulated to ensure that its trajectory follow the desired pattern as specified previously. The figure below shows the visualization of this simulation.



Transformation Matrik : T0/07

0.643	0.000	-0.766	188.73
-0.000	-1.000	-0.000	-49.00
-0.766	-1.000	-0.643	87.17
0.000	0.000	0.000	1.000

Grid: □ X-Grid □ Y-Grid □ Z-Grid

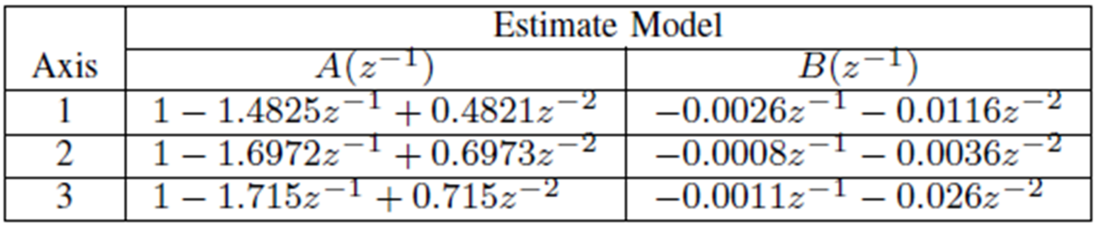
Tracking: ☑ Hold

Axis Coordinat: □ Axis-0 □ Axis-1 □ Axis-2 □ Axis-3 □ Axis-4 □ Axis-5 □ Axis-6 □ Axis-7

Screenshot of the simulation process in MATLAB

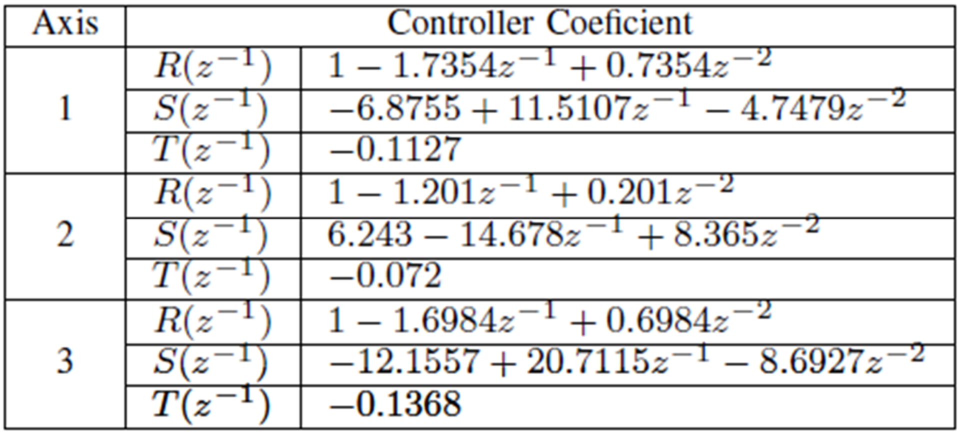
Estimate model and the Controller Parameters

The following two table show the estimate model of servo valves and the controller parameters particularly at joint 1, 2 and 3. The other joints are obtained using the same procedure the D-H Parameter Model and by applying RLS algorithm.



Axis	Estimate Model	
	$A(z^{-1})$	$B(z^{-1})$
1	$1 - 1.4825z^{-1} + 0.4821z^{-2}$	$-0.0026z^{-1} - 0.0116z^{-2}$
2	$1 - 1.6972z^{-1} + 0.6973z^{-2}$	$-0.0008z^{-1} - 0.0036z^{-2}$
3	$1 - 1.715z^{-1} + 0.715z^{-2}$	$-0.0011z^{-1} - 0.026z^{-2}$

Estimate Model of the Servo Values



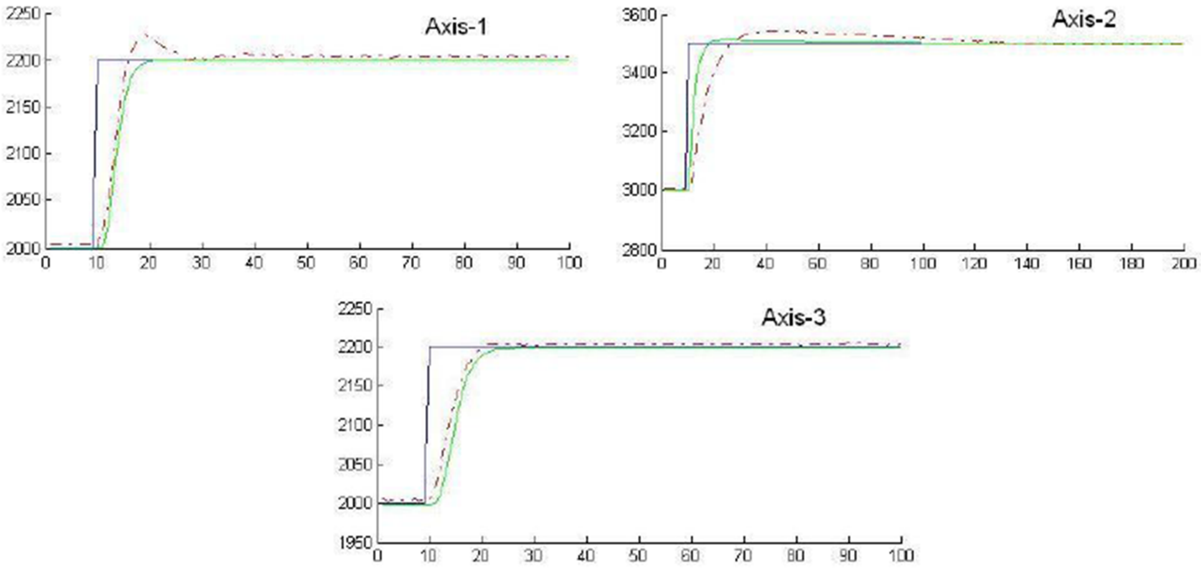
Axis	Controller Coeficient	
1	$R(z^{-1})$	$1 - 1.7354z^{-1} + 0.7354z^{-2}$
	$S(z^{-1})$	$-6.8755 + 11.5107z^{-1} - 4.7479z^{-2}$
	$T(z^{-1})$	-0.1127
2	$R(z^{-1})$	$1 - 1.201z^{-1} + 0.201z^{-2}$
	$S(z^{-1})$	$6.243 - 14.678z^{-1} + 8.365z^{-2}$
	$T(z^{-1})$	-0.072
3	$R(z^{-1})$	$1 - 1.6984z^{-1} + 0.6984z^{-2}$
	$S(z^{-1})$	$-12.1557 + 20.7115z^{-1} - 8.6927z^{-2}$
	$\boldsymbol{T(z^{-1})}$	$\mathbf{-0.1368}$

The controller coefficient values

Performance of the PIC Controller

Having generated and simulated, trajectory file is loaded into microcontroller memory. Now, it is the PID controller responsibility to drive the robot so that the angle in every axis following the value inside the trajectory file.

The performance of PID controller for every axis is shown in the figure below. Step input is employed to observe the PID controller performance individually. Solid line curve is the response of PID controller from simulation based on the estimate model obtained by identification whereas dotted line curve is the measured response of PID controller from the experiment. The figure below shows that the PID controller performance from experiment is almost the same as the one from simulation. It means that the estimate model of servo valves are closed enough to the real model. Errors that is the difference between measured angles and the reference inputs, are less than 0x0006 (digital value) or less than 0.5degrees.



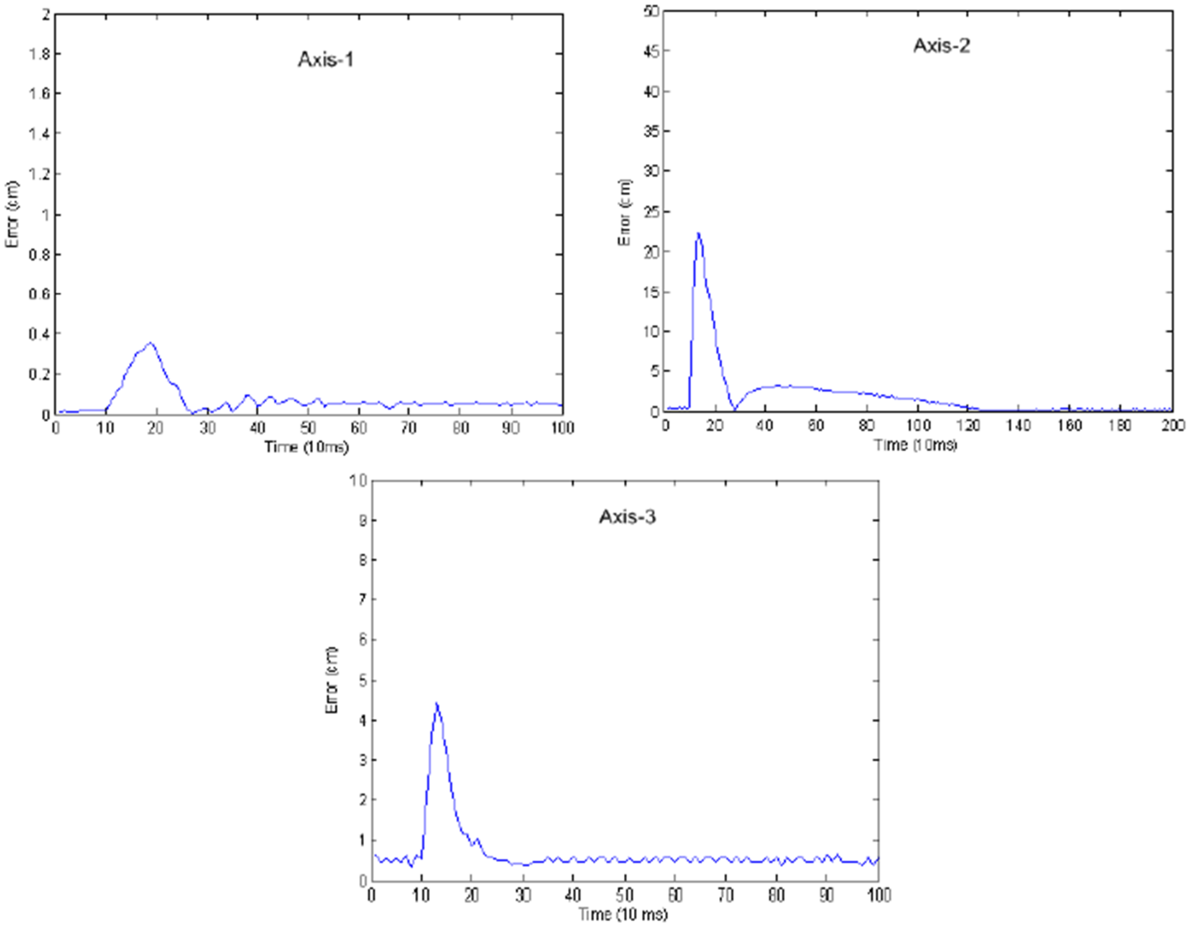
PID Controller Performance in different axes

The End Point Error

The following figure shows the experiment result of the end-point error (the difference between the real position and the simulated one) when each axis is driven independently.

The first figure changes the positions from (191,6,149) to (189,23,149), 2nd figure from (154,0,206) to (124,0,233) and 3rd Figure from (180,0,135) to (187,0,121) respectively.

This result shows that the end-point error is very small. This error (in cm unit) is closed to zero at steady state. Only the 2nd-axis motion which has so big error at transient time. However its steady state error is very small enough.



The End-Point Errors

Discussion

As shown in PID Controller performance Diagram, the axis-2 requires much more time to reach steady state. Most axes require 10 to 30 sampling time to be stable. With 10 ms sampling rate it is equal to 100 ms to 300 ms. Only axis-2 that requires more than 100 sampling time or more than 1 second to achieve the steady state.

Based on experiment, axis-2 can not be forced to reach the reference input too fast. If this is done, it yields oscillation and it never be stable. This problem is happened because the center of mass of the whole machine, i.e. weight of the machine is on this axis, hence the velocity of this axis is not directly proportional to the DC component of the servo valve. PID controller will think that the weigh is the disturbance of the system i.e that have not formulated in the model.

Moreover when the angle of the axis changes, the center of mass moves and the force that counter weight changes too. It means that the model may change dependent on the angle at a particular time. Therefore, we have to choose slower rise-time for this axis. During the transient, the slowest axis-2 may result end-point error. This error is small enough and could be ignored. Most of the time the error is acceptable such that the painting target do not need rework.

Conclusions and Future Work

An embedded system was constructed to read angles from potentiometers and command servo valves according to PID controller. The servo valves were identified. A set of formulae were developed to solve inverse kinematics problem.

Experiments show that all above said has been performed very well. Adaptive controllers such as MRAC or repetitive-path optimization are other points that can be further considered.

Bibliography

While creating the report I have used information from the following sources:

- IEEE Xplore
- www.robots.com
- Society of Robots – www.societyofrobots.com
- NASA Robotics - http://prime.jsc.nasa.gov/
- Wikipedia – http://www.wikipedia.org
- A book on Robot System Design