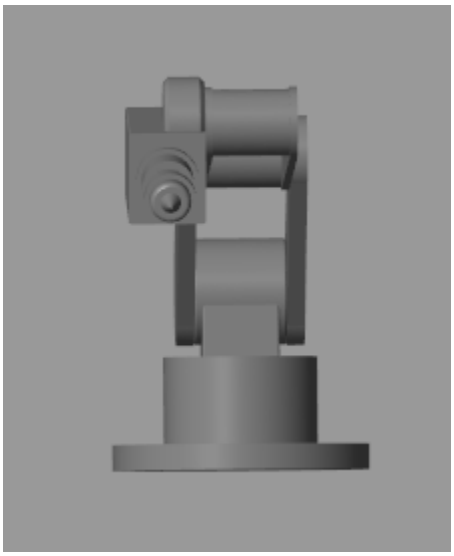
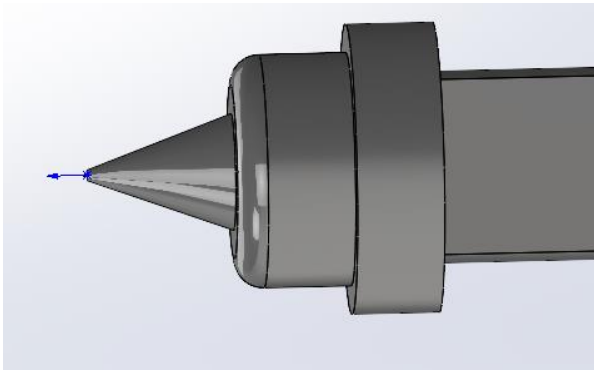
Omar Hisham Ahmed Zaitoun 7444

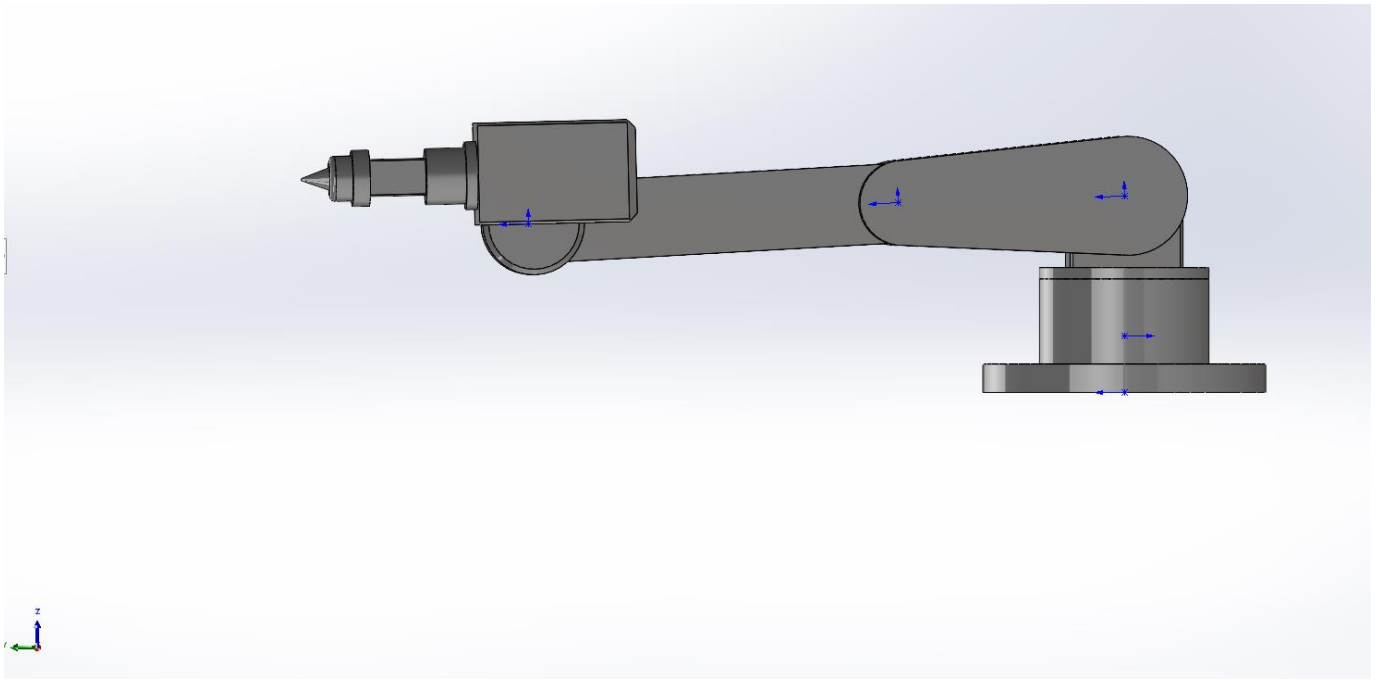
Robotics Project



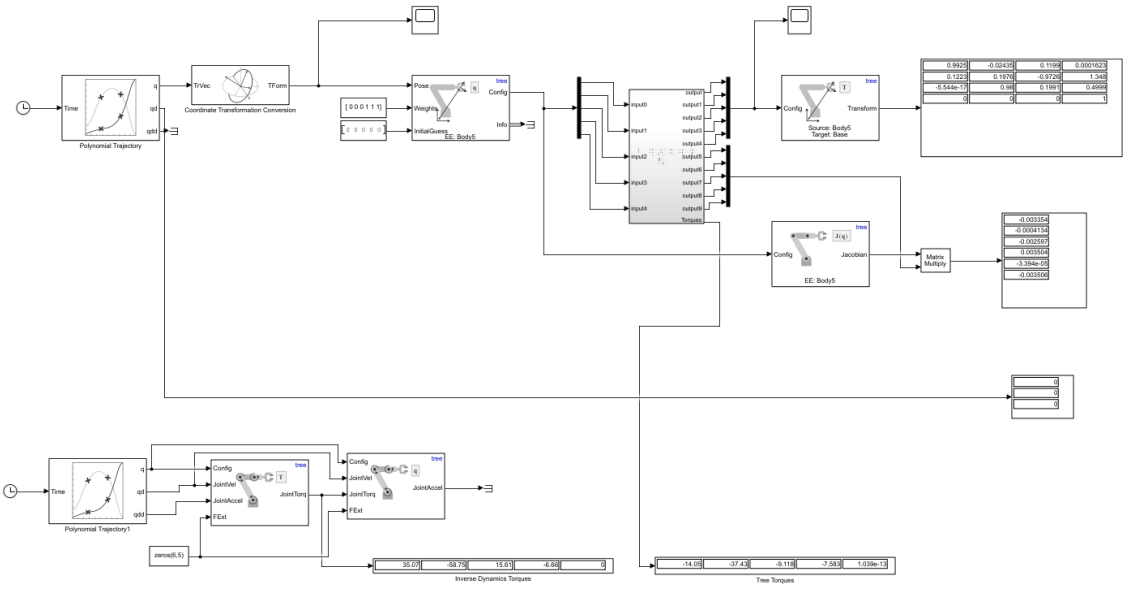
Drawing:

Coordinate system: I adjusted the origins of the parts to act as co ordinate systems on matlab, without inserting a new coordinate system. The * represents the Z axis of each part



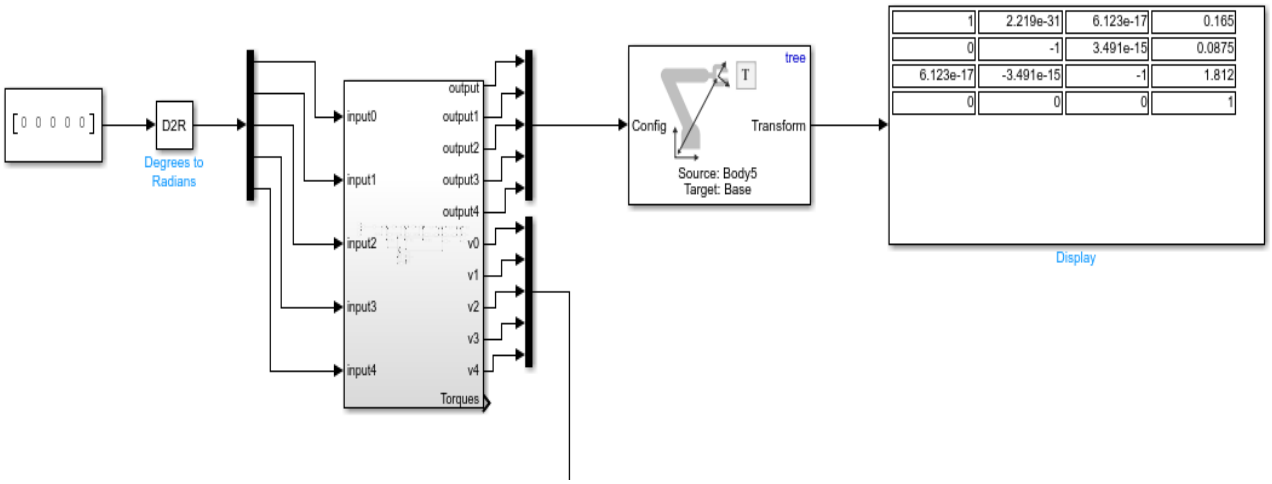


Simulink:

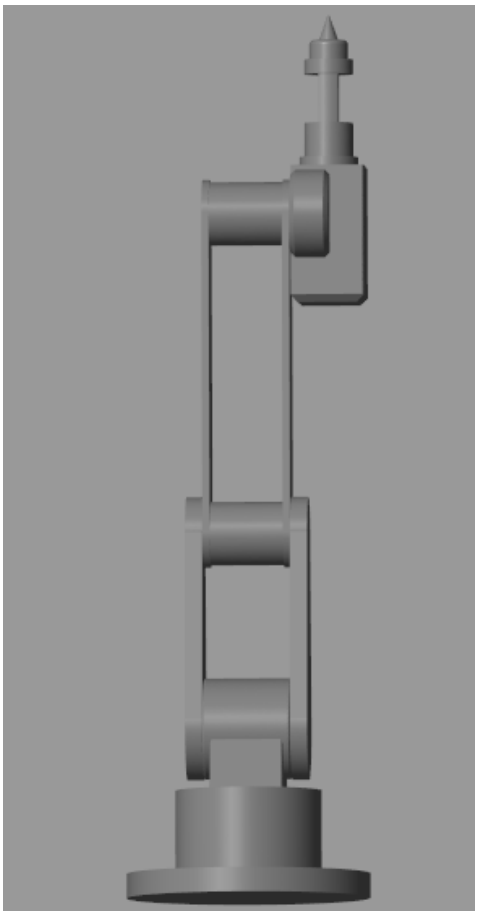

Time
q
qd
qdd
Polynomial Trajectory
TrVec
TForm
Coordinate Transformation Conversion
[0 0 0 1 1 1]
[0 0 0 0 0 0]
Pose
Weights
InitialGuess
Config
Info
EE: Body5
tree
input0
input1
input2
input3
input4
output
output1
output2
output3
output4
output5
output6
output7
output8
Torques
Config
T
Transform
Source: Body5
Target: Base
0.9923
-0.02435
0.1199
0.0001623
0.1223
0.1978
-0.9726
1.348
-5.544e-17
0.98
0.1991
0.4999
0
0
0
1
J(q)
Jacobian
EE: Body5
Matrix
Multiply
-0.003354
-0.0004134
-0.002597
0.003506
-3.384e-05
-0.003506
0
0
0
Time
q
qd
qdd
Polynomial Trajectory1
Config
JointVel
JointAccel
FExt
JointTorq
zeros(6,5)
JointTorq
JointAccel
35.07
-58.75
15.61
-6.86
0
Inverse Dynamics Torques
-14.05
-37.43
-9.118
-7.583
1.039e-13
Tree Torques

Forward Kinematics:

Input of the robot is initial position that makes the robot stand vertically as shown below:

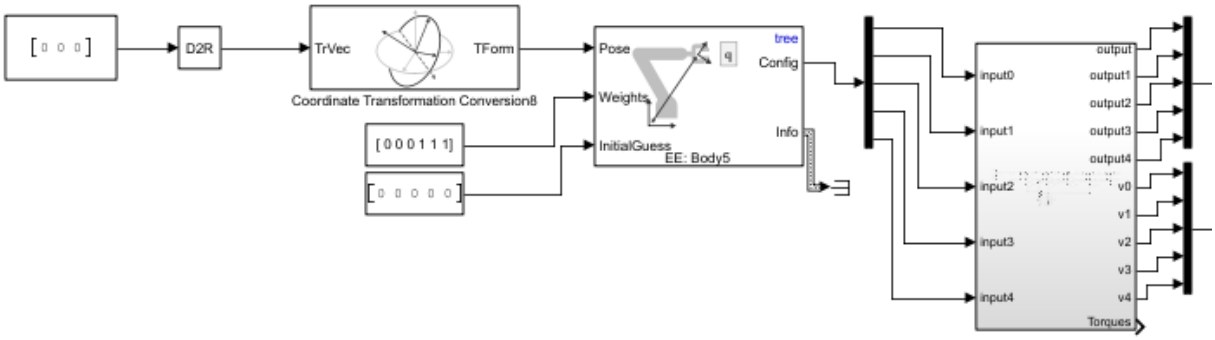


Output:

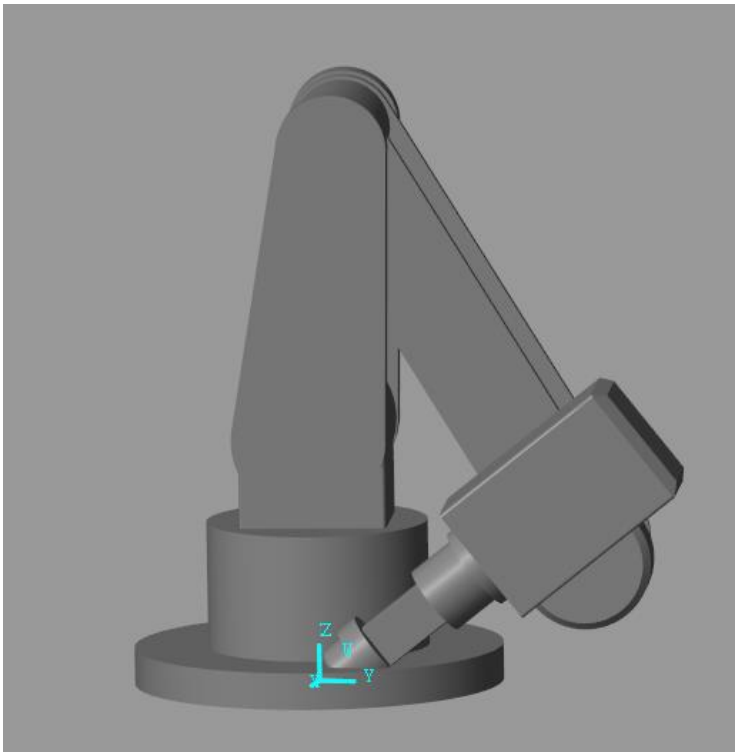


1	2.219e-31	6.123e-17	0.165
0	-1	3.491e-15	0.0875
6.123e-17	-3.491e-15	-1	1.812
0	0	0	1

Inverse Kinematics:



The end effector position reaches the nearest point to the origin, however it doesn't reach it , due to the construction of the robot , the end effector is not on the same plane and the origin is outside the workspace of the robot.



Trajectory:

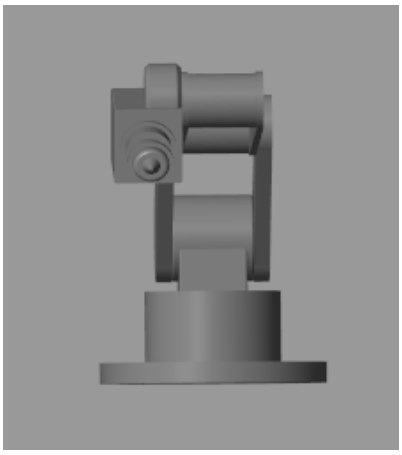
Required Shape: X

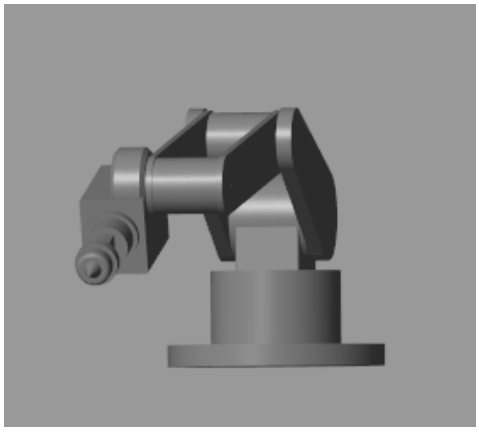
Way Points:

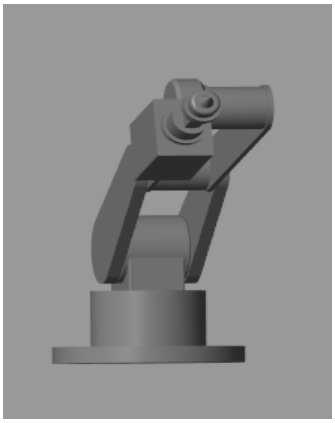
[0, -0.3, 0.3, 0, 0.3, -0.3, 0; 1.35, 1.35, 1.35, 1.35, 1.35, 1.35, 1.35; 0.5, 0.2, 0.8, 0.5, 0.2, 0.8, 0.5]

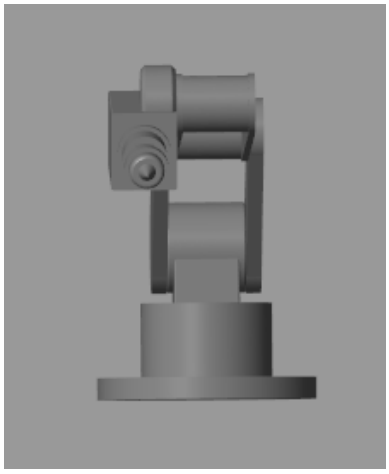
7 Points, to make the X shape and return to the original starting point.

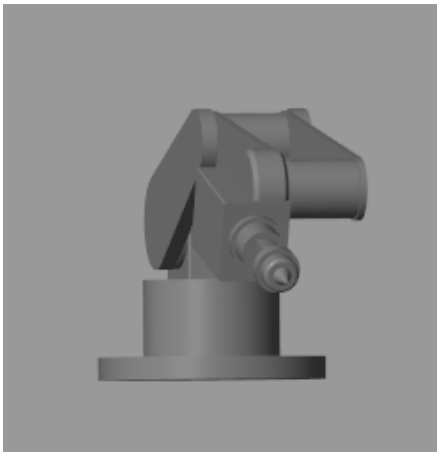
Cubic Velocity conditions: [0.1, 0.1, 0.2, 0.1, 0.1, 0.2, 0.1; 0, 0, 0, 0, 0, 0, 0; 0.1, 0.1, 0.2, 0.1, 0.1, 0.2, 0.1]

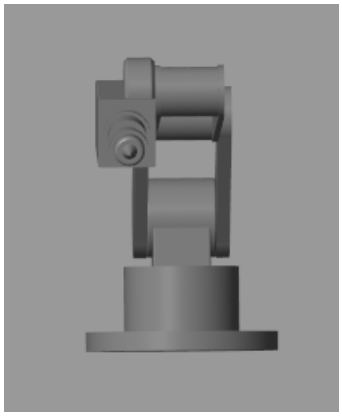


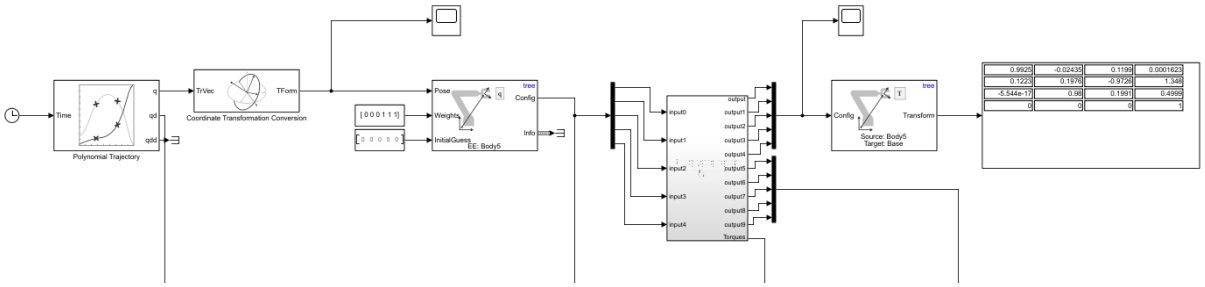


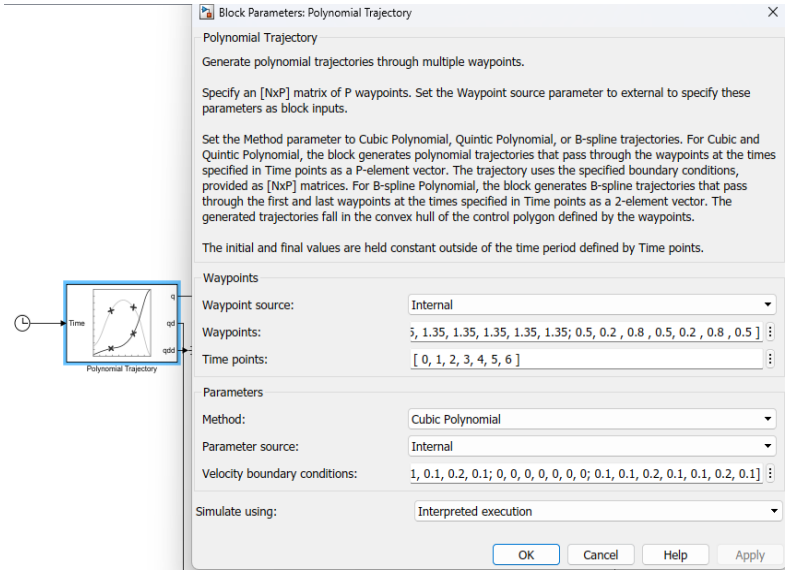


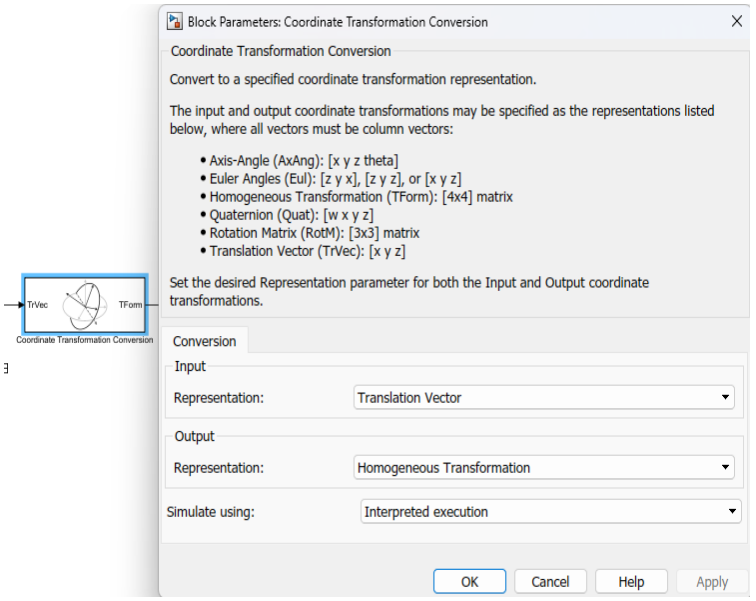


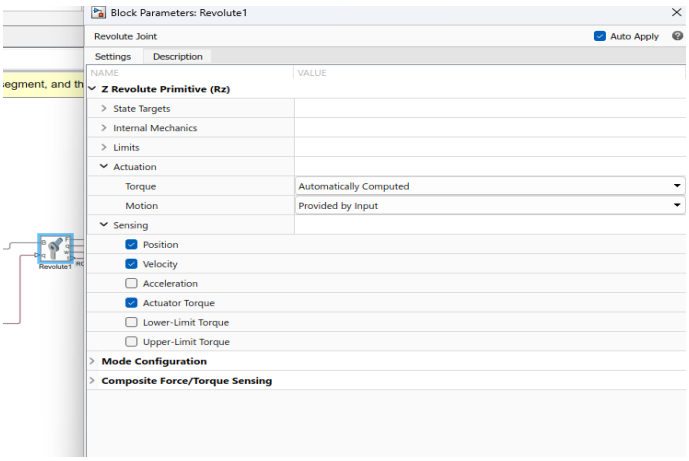


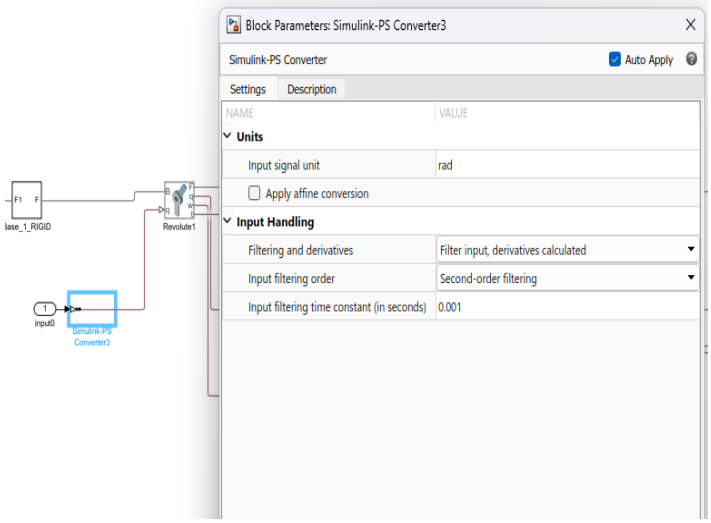



Time
q
qd
qdd
Polynomial Trajectory
TrVec
TForm
Coordinate Transformation Conversion
[0 0 0 1 1]
[0 0 0 0 0]
Pose
Weights
InitialGuess
Config
Info
EE: Body5
tree
input0
input1
input2
input3
input4
output0
output1
output2
output3
output4
output5
output6
output7
output8
output9
Torques
Config
Transform
Source: Body5
Target: Base
0.9928 -0.02438 0.1199 0.0001623
0.1223 0.1976 -0.9726 1.348
-5.544e-17 0.98 0.1991 0.4999
0 0 0 1


Block Parameters: Polynomial Trajectory
Polynomial Trajectory
Generate polynomial trajectories through multiple waypoints.
Specify an [NxP] matrix of P waypoints. Set the Waypoint source parameter to external to specify these parameters as block inputs.
Set the Method parameter to Cubic Polynomial, Quintic Polynomial, or B-spline trajectories. For Cubic and Quintic Polynomial, the block generates polynomial trajectories that pass through the waypoints at the times specified in Time points as a P-element vector. The trajectory uses the specified boundary conditions, provided as [NxP] matrices. For B-spline Polynomial, the block generates B-spline trajectories that pass through the first and last waypoints at the times specified in Time points as a 2-element vector. The generated trajectories fall in the convex hull of the control polygon defined by the waypoints.
The initial and final values are held constant outside of the time period defined by Time points.
Waypoints
Waypoint source:
Internal
Waypoints:
5, 1.35, 1.35, 1.35, 1.35, 1.35; 0.5, 0.2 , 0.8 , 0.5, 0.2 , 0.8 , 0.5]
Time points:
[0, 1, 2, 3, 4, 5, 6]
Parameters
Method:
Cubic Polynomial
Parameter source:
Internal
Velocity boundary conditions:
1, 0.1, 0.2, 0.1; 0, 0, 0, 0, 0, 0, 0; 0.1, 0.1, 0.2, 0.1, 0.1, 0.2, 0.1]
Simulate using:
Interpreted execution
OK
Cancel
Help
Apply

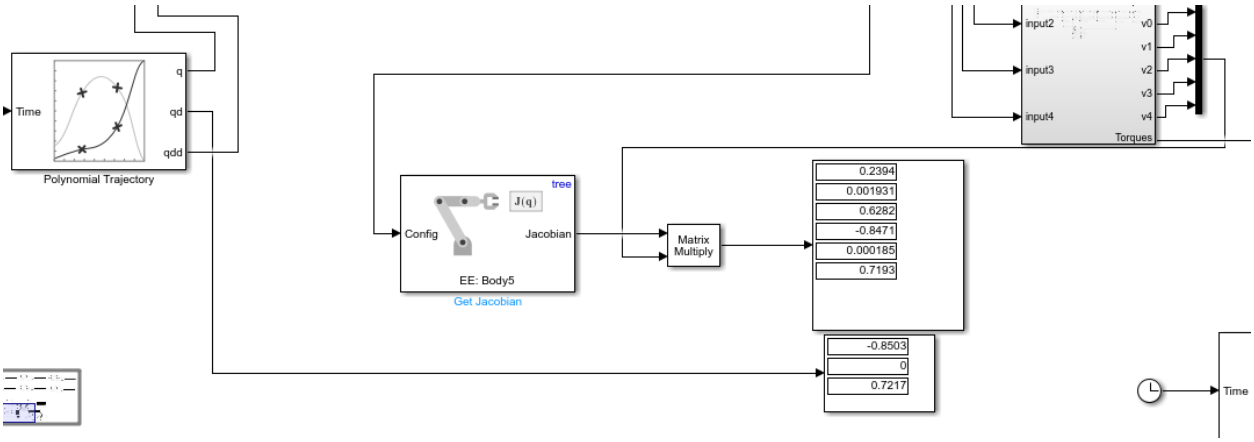

Block Parameters: Coordinate Transformation Conversion
Coordinate Transformation Conversion
Convert to a specified coordinate transformation representation.
The input and output coordinate transformations may be specified as the representations listed below, where all vectors must be column vectors:
• Axis-Angle (AxAng): [x y z theta]
• Euler Angles (Eul): [z y x], [z y z], or [x y z]
• Homogeneous Transformation (TForm): [4x4] matrix
• Quaternion (Quat): [w x y z]
• Rotation Matrix (RotM): [3x3] matrix
• Translation Vector (TrVec): [x y z]
Set the desired Representation parameter for both the Input and Output coordinate transformations.
Conversion
Input
Representation:
Translation Vector
Output
Representation:
Homogeneous Transformation
Simulate using:
Interpreted execution
OK
Cancel
Help
Apply
TrVec
TForm
Coordinate Transformation Conversion


Block Parameters: Revolute1
Revolute Joint
Auto Apply
Settings
Description
NAME
VALUE
Z Revolute Primitive (Rz)
State Targets
Internal Mechanics
Limits
Actuation
Torque
Automatically Computed
Motion
Provided by Input
Sensing
Position
Velocity
Acceleration
Actuator Torque
Lower-Limit Torque
Upper-Limit Torque
Mode Configuration
Composite Force/Torque Sensing


Block Parameters: Simulink-PS Converter3
Simulink-PS Converter
Auto Apply
Settings
Description
NAME
VALUE
Units
Input signal unit
rad
Apply affine conversion
Input Handling
Filtering and derivatives
Filter input, derivatives calculated
Input filtering order
Second-order filtering
Input filtering time constant (in seconds)
0.001
F1
input0
Simulink-PS Converter3
Revolute1

Jacobian:

In this part Velocities were taken as outputs from the subsystem, 1 from each joint and compared with the Jacobian Block on displays.

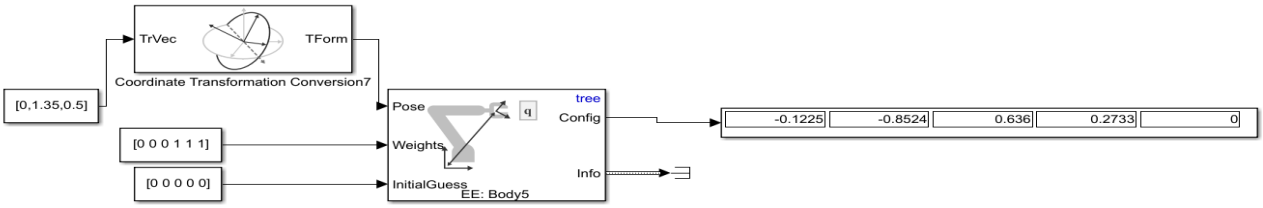


System Dynamics:

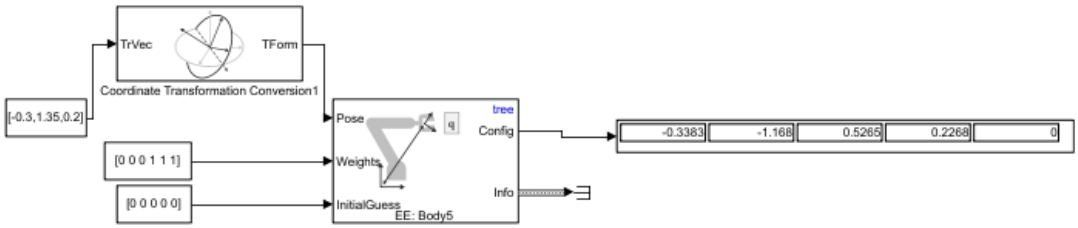
In this part the waypoints from the polynomial trajectory block were used point by point to calculate the angles required using the inverse dynamics block, then added to another polynomial trajectory block to calculate the torques generated, then compared with the torques from the subsystem on a scope.

Waypoints: [0, -0.3, 0.3, 0, 0.3, -0.3, 0; 1.35, 1.35, 1.35, 1.35, 1.35, 1.35, 1.35; 0.5, 0.2, 0.8, 0.5, 0.2, 0.8, 0.5]

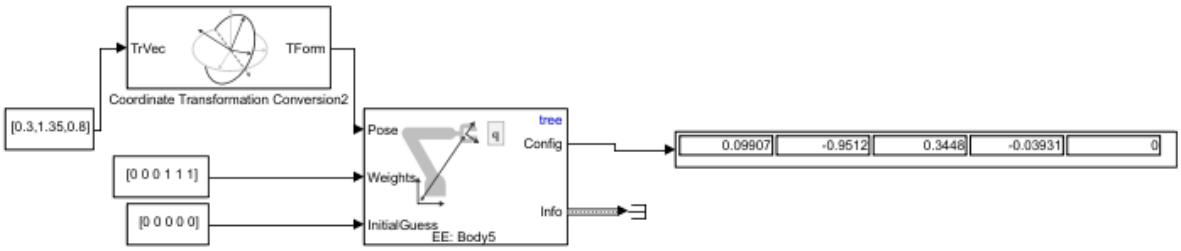
1st, 4th ,7th points:



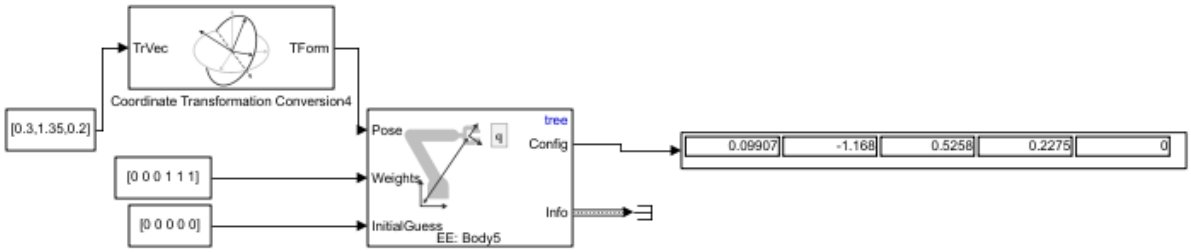
2nd point:



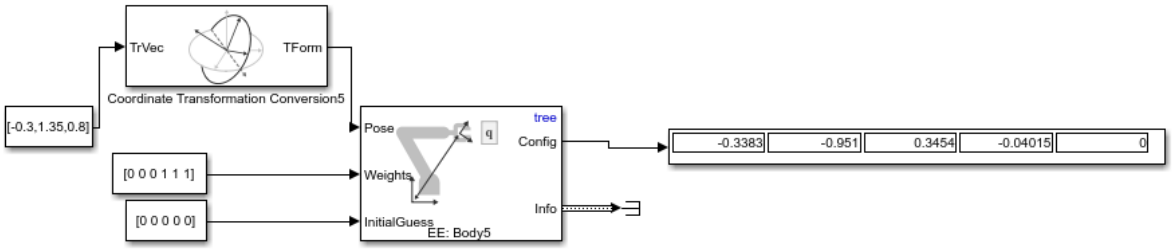
3rd point:



5th point:

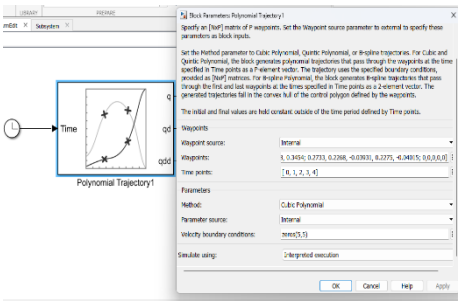


6th point:

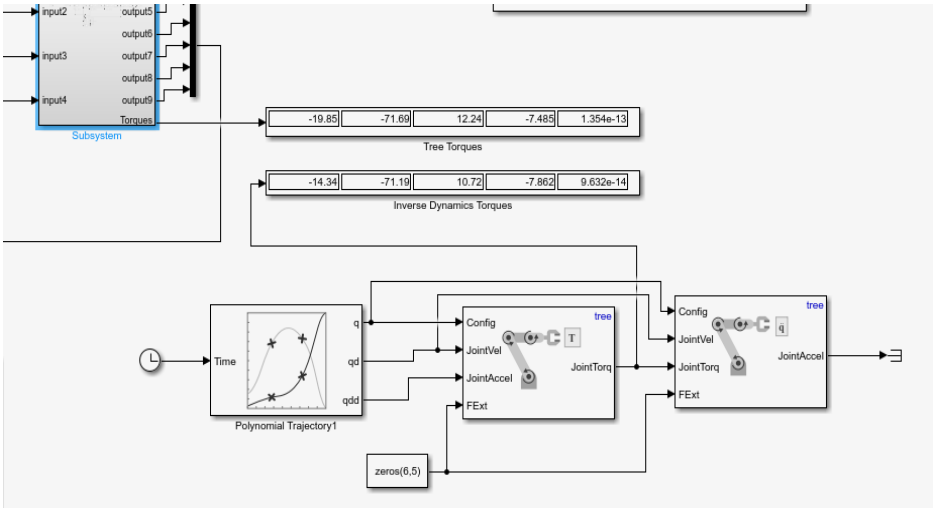


New Trajectory waypoints:

[-0.1225, -0.3383, 0.09907, 0.09907, -0.3383; -0.8524, -1.168, -0.9512, -1.168, -0.951; 0.636, 0.5265, 0.3448, 0.05258, 0.3454; 0.2733, 0.2268, -0.03931, 0.2275, -0.04015; 0,0,0,0,0]



Comparison:



Forward Dynamics:

Compared the Positions generated from the block to the positions from the subsystem: