MEEN 612: Final Project

Texas A&M University

MEEN 612-600

December 10th, 2021

Roman Yoder

Roman Yoder
Mech 612: Project 1 Report

Abstract:

This project explores the dynamics and control of a 3-link robotic manipulator that could be used for a multitude of applications. The settled application being an arm to transfer selected products from one assembly line to another. The overall goal of this project was to simulate this application with realistic trajectories, control, and mass change. First, the DH table is developed, then the dynamic equations are computed from the Lagrange equations and Christoff symbols. Second, control laws are developed and simulated via the dynamics of the system. The outputs of these are mapped in three-dimensional space. Finally, an object is created which interferes with the original trajectory for which a new trajectory is developed.

Roman Yoder
Mech 612: Project 1 Report

Introduction:

The selected robot is presented in the drawing below with two revolute joints and one prismatic.

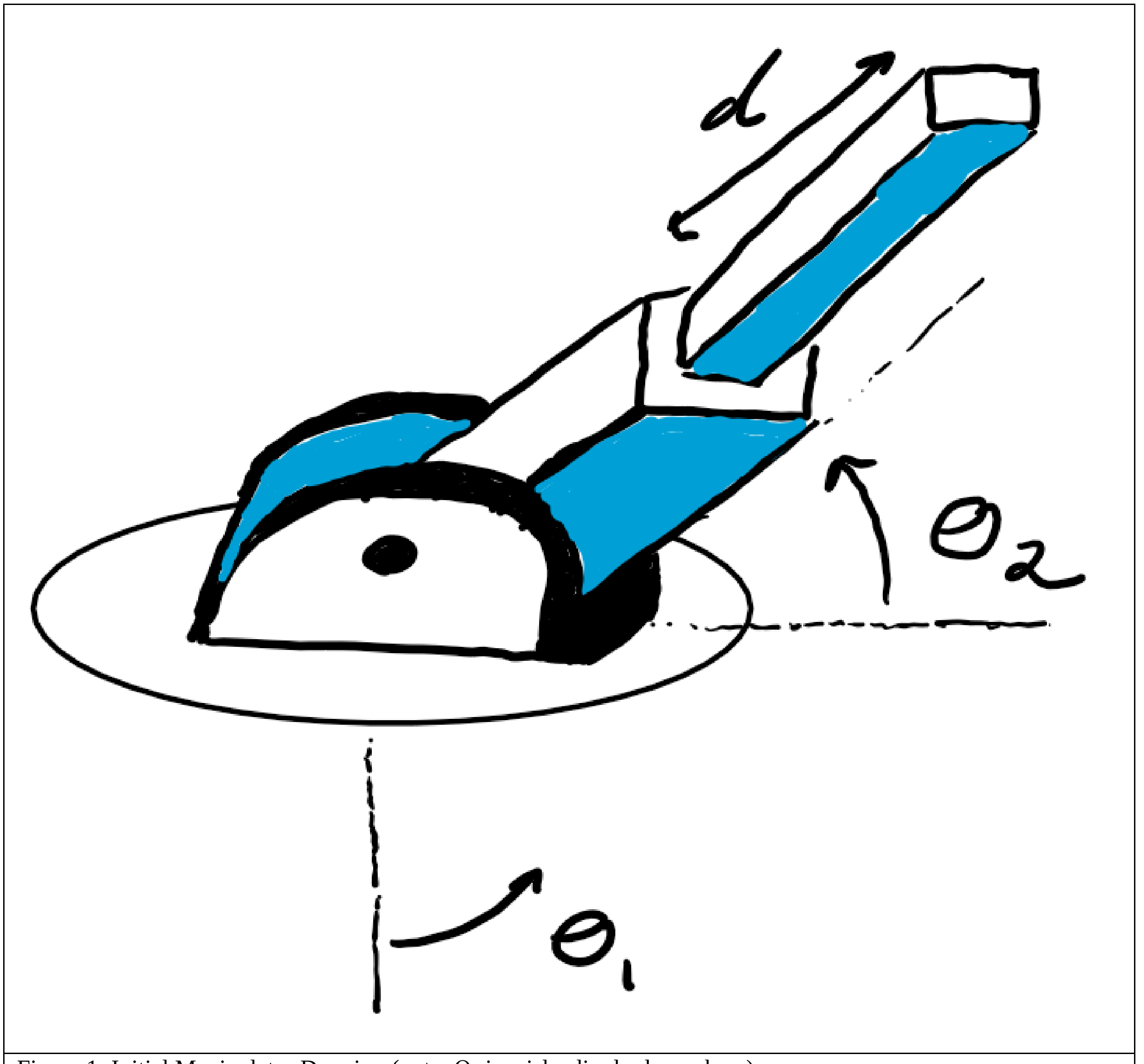


Figure 1: Initial Manipulator Drawing (note: Θ_2 is misleadingly drawn here)

Clearly this robot could move manufacturing products from assembly line to assembly line as simulated. Another possible application would be that of an autonomous fire ladder with a hose at the end effector. The goal of this project was to pick desired values for each joint variables that would capture its application. This was done by rotating 90 degrees, elevating 45 degrees and extending the arm 1 m.

This would position the arm to another assembly line or possibly, elevate the arm to put out a fire. Applications of proportional control with no gravity compensation, proportional control with gravity compensation, computed torque, and adaptive control are explored.

Body:

Once the manipulator joints were selected the DH table was developed from the joint axes drawn below.

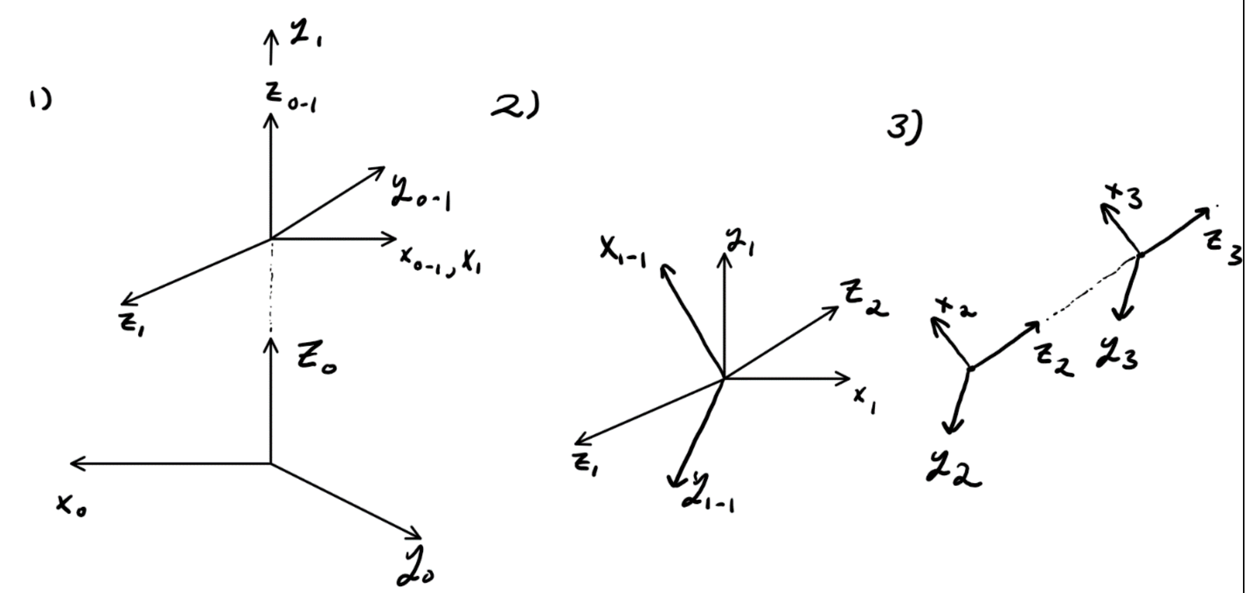


Figure 2: Joint Axes

From these joint axes a DH table is developed.

$$\begin{pmatrix} \theta_1 & d_1 & 0 & \frac{\pi}{2} \\ \theta_2 & 0 & 0 & \frac{\pi}{2} \\ 0 & d_3 & 0 & 0 \end{pmatrix}$$

Figure 3: DH table (MATLAB output)

This DH table can be used with the A matrix (shown below) to develop the end effector position by multiplying out the A matrix for each link.

$$\begin{pmatrix} \cos(\theta) & -\cos(\alpha)\sin(\theta) & \sin(\alpha)\sin(\theta) & a\cos(\theta) \\ \sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha)\cos(\theta) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 4: Homogenous Transformation Matrix A

$$\begin{pmatrix} \cos(\theta_1)\cos(\theta_2) & \sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) & d_3\cos(\theta_1)\sin(\theta_2) \\ \cos(\theta_2)\sin(\theta_1) & -\cos(\theta_1) & \sin(\theta_1)\sin(\theta_2) & d_3\sin(\theta_1)\sin(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & d_1 - d_3\cos(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 5: End Effector Homogenous Transformation Matrix

With the known joint variables, the homogenous transformation matrix can be evaluated for the end effector position and orientation (as in the code) providing the forward kinematics. The more relevant problem however is solving the inverse kinematics problem; where, given a final position and orientation the necessary joint angles are computed. This problem is typically more difficult to solve than the forward kinematics problem. But as this is a three joint manipulator that nearly reflects a spherical coordinate system this problem can be solved by simply applying trigonometry to derive the following identities.

$$d_3 = \sqrt{\left(z_f - d_1\right)^2 + \left(x_f\right)^2 + \left(y_f\right)^2}$$

$$\theta_2 = \sin^{-1}\left({(Z_f)}/{d_3} \right) + \pi$$

$$\theta_1 = \tan^{-1}\left({(Y_1)}/{X_1} \right)$$

Equations of motion:

The system is put into the following standard form:

$$M(q)\ddot{q} + C_{(q,\dot{q})}\dot{q} + G(q) = \tau$$

Where q represents the joint variables, τ is the torque or force input, and M is the mass at the center of gravity for each joint.

First M is derived with the following summation over each joint (i):

$$M = \sum_{i=1}^{n} [m_i * J_{vi}\ (q)^T * J_{vi}(q) + \ J_{\omega i}\ (q)^T * R_i(q) * I_i * R_i(q)^T * \ J_{\omega i}(q)]$$

With the following definitions:

$$J_{\omega i}(q) = \ [\rho_1 z_0 \cdots \rho_n z_{n-1}]$$

Where Jω is the angular velocity Jacobian and, ρ is 0 if joint (i) is prismatic and 1 if revolute.

$$J_{vi}(q) = \frac{\partial O_n^0}{\partial q_i}$$

Where Jv is the linear velocity jacobian and O represents the position in the global frame, and is differentiated with respect to each joint variable for each link.

$$I_i = \begin{bmatrix} I_{ixx} & 0 & 0 \\ 0 & I_{iyy} & 0 \\ 0 & 0 & I_{izz} \end{bmatrix}$$

Here I represents the Inertia tensor, assuming that each link is symmetric about the x, y and z axes this becomes the presented diagonal matrix. Each diagonal element of I is calculated with all joint variables (q) set to zero, this allows the Rotation matrix (R) to account for these joint variables and the change in moments of inertia in the global frame.

The Rotation matrix is defined, in simplest terms, as the upper 3x3 matrix of the composite homogenous transformation matrix created by multiplying matrix A out to the current link (i). For the selected manipulator all matrices in this standard form will be 3x3 and arrays will by 3x1 as there are 3 joint variables.

Then the C matrix is calculated from the following Christof relation:

$$C_{kj} = \sum_{i=1}^{n} \frac{1}{2}\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\}\dot{q}_i$$

Where d represents individual elements of the mass matrix called at indexes k, j, and i; $\dot{q}$ represents the first derivative of joint variable i with respect to time.

With M and C derived the validity of the system can be confirmed by checking if,

$$\dot{M} - 2C$$

is skew symmetric.

Given M and C, G is derived from the Lagrange equations:

$$L = Ke - Pe$$

Where Ke and Pe represent the sum of kinetic and potential energies over the system and can be expressed as:

$$K_e = \frac{1}{2}\dot{q}^T[m_i * J_{vi}\,(q)^T * J_{vi}(q) + \,J_{\omega i}\,(q)^T * R_i(q) * I_i * R_i(q)^T * \,J_{\omega i}(q)]\dot{q}$$

$$Pe = mgh$$

Pe can be solved for each link by taking m as the links mass and h as the position of the links center of gravity.

The equations of motion can be represented as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau$$

Which can then be set equal to the previous equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = M(q)\ddot{q} + C_{(q,\dot{q})}\dot{q} + G(q)$$

Solving for G.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} - (M(q)\ddot{q} + C_{(q,\dot{q})}\dot{q}) = G(q)$$

Moments of Inertia:

The moments of Inertia for each link can be calculated with the following formulas:

$$I_{zz} = \frac{1}{2}(m_i)(r)^2$$

$$I_{xx} = \frac{1}{12}(m_i)(r)^2$$

$$I_{yy} = \frac{1}{12}(m_i)(r)^2$$

Assuming each link is orientated vertically- these relations may change otherwise.

Trajectory Development:

Trajectories can be developed by using polynomial interpolation, given n constraints for the joint variable (in position, velocity, and acceleration etc.)

$$q(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1}$$

$$\dot{q}(t) = a_1 + \cdots + (n-1) a_{n-1} t^{n-2}$$

Given constraints for,

$$q(t)$$
$$\dot{q}(t)$$

at any finite time, the solution takes the form of:

$$\begin{bmatrix} 1 & t_0 & t_0^2 \cdots \\ 0 & 1 & 2t_0 \cdots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ \vdots \end{bmatrix}$$

Which can be expanded for additional constraints in either time or higher order derivatives, and then solved by using matrix left division.

Control Laws:

Proportional:

The control laws applied include a proportional control with no gravity compensation, a proportional control with gravity compensation, a computed torque law and an adaptive torque law.

The proportional control law takes the form of:

$$\tau = -\mathrm{Kp} * (\mathrm{q}_e) - \mathrm{Kd} * (\dot{\mathrm{q}}_e)$$

$$M(q)\ddot{q} + C_{(q,\dot{q})}\dot{q} + G(q) = -\mathrm{Kp} * (\mathrm{q}_e) - \mathrm{Kd} * (\dot{\mathrm{q}}_e)$$

Where q_e the error function,

$$\mathrm{q}_e = q_t - q_d$$

is the true value minus the desired.

This form of control fails to compensate for the gravity array G and will not execute the desired trajectory if the potential at the desired configuration is not a local minimum of the Pe function. Introducing gravitational compensation results in the form:

$$\tau = G(q) - \mathrm{Kp} * (\mathrm{q}_e) - \mathrm{Kd} * (\dot{\mathrm{q}}_e)$$

$$M(q)\ddot{q} + C_{(q,\dot{q})}\dot{q} = -\mathrm{Kp} * (\mathrm{q}_e) - \mathrm{Kd} * (\dot{\mathrm{q}}_e)$$

This will execute correctly regardless of if the potential energy at the desired location is a local minima. However, this control method typically requires high gains (or control outputs) and does not predict the coming change in the desired trajectory, for this, the computed torque law is derived which includes a feedforward term.

Computed Torque:

First, an energy like function is developed for the system:

$$E = \frac{1}{2}\dot{q}_e{}^T * M * \dot{q}_e + \frac{1}{2}\dot{q}_e{}^T * Kp * \dot{q}_e$$

With M and Kp as positive definite matrices the only way for E to be zero is for the error and change in error to be equal to zero for some finite value of time.

Differentiating with respect to time yields:

$$\frac{dE}{dt} = \ddot{q}_e{}^T * M * \dot{q}_e + \frac{1}{2}\dot{q}_e{}^T * \dot{M} * \dot{q}_e + q_e{}^T * Kp * \dot{q}_e$$

Utilizing the fact that $\dot{M}$-2C is skew symmetric results in:

$$\frac{dE}{dt} = \ddot{q}_e{}^T * M * \dot{q}_e + \dot{q}_e{}^T * C * \dot{q}_e + q_e{}^T * Kp * \dot{q}_e$$

Which can be simplified to:

$$\frac{dE}{dt} = \dot{q}_e{}^T(M * \ddot{q}_e + C * \dot{q}_e + Kp * q_e)$$

Expanding q_e and substituting the original form:

$$\frac{dE}{dt} = \dot{q}_e{}^T(\tau - G(q_t) - M * \ddot{q}_d - C * \dot{q}_d + Kp * q_e)$$

Then the right-hand side is set equal to:

$$\frac{dE}{dt} = \dot{q}_e{}^T(-Kd)\dot{q}_e$$

If Kd is positive definite, for any value of $\dot{q}_e$ the energy of the system will be (and remain) zero only when $\dot{q}_e$ is zero for a finite time. This guarantees that our system will always approach a zero-energy value which is defined by the desired configuration, via the error. Then the toque law can be solved by setting the last two equations equal.

$$\tau = -Kd * \dot{q}_e - Kp * q_e + M * \ddot{q}_d + C * \dot{q}_d + G(q_t)$$

Clearly, this torque law not only compensates for the error between the true and desired value, but also feeds forward the change in desired value.

<u>Adaptive Control:</u>

Adaptive control predicts unknown values, like mass, and compensates for them in the torque law.

First, A composite error is defined as:

$$S = \dot{q}_e + \lambda * q_e$$

Where λ is the designers' choice and scales the effect of $\dot{q}_e$ versus q_e.

Then $\dot{q}_r$ is defined as:

$$\dot{q}_r = \dot{q}_d - \lambda q_e$$

The torque law defined as:

$$\tau = M * \ddot{q}_r + C * \dot{q}_r + G(q_t) - Kd * S$$

Then,

$$M * \ddot{q}_r + C * \dot{q}_r + G(q_t)$$

Can be written as:

$$Y_0 + \widehat{M}\, Y_1$$

Where Y_0 represents the known terms of the system and $\widehat{M}$ represents the unknown mass, with Y_1 being the related terms. The toque law is then written as:

$$\tau = Y_0 + \widehat{M}\, Y_1 - Kd * S$$

The unknown mass is then adjusted for with:

$$(\dot{\widehat{M}}) = -\frac{1}{\gamma} * S^T * Y_1$$

And integrated in real time.

This does not guarantee that the error in the predicted mass will go to zero, but rather that the torque law will provide the correct input via S and $\widehat{M}$ that will drive the system to the correct trajectory.

<u>Results:</u>

Link values:

Values were selected based on expected weights and lengths for a typical assembly line application.

Table 1: Selected Mass Values			
M1	M2	M3	M4 (end effector mass)
50 Kg	25 Kg	12.5 Kg	10 Kg (subject to change)

Table 2: Link parameters	
	Link Dimensions (Length, Radius) (Meters)
Link 1	(1,0.2)
Link 2	(1,0.15)
Link 3	(1,0.1)

Table 3: Moments of Inertia			
Link:	1	2	3
I_{xx} (Kg*M^2)	N/A	2.083	1.0417
I_{yy} (Kg*M^2)	N/A	2.083	1.0417
I_{zz} (Kg*M^2)	1	0.2813	0.0625

Equations of Motion:

<u>Link Dynamics:</u>

Table 4: Link 1 Dynamics	
Rotation Matrix	$\begin{pmatrix} \cos(\theta_1(t)) & -\sin(\theta_1(t)) & 0 \\ \sin(\theta_1(t)) & \cos(\theta_1(t)) & 0 \\ 0 & 0 & 1 \end{pmatrix}$
Linear Jacobian	
Angular Jacobian	Jw1 = 3×3 0 0 0 0 0 0 1 0 0
Kinetic Energy	$\dfrac{I_{1ZZ} \left(\frac{\partial}{\partial t}\, \theta_1(t)\right)^2}{2}$
Potential Energy	N/A

Table 5: Link 2 Dynamics (Cg represents its center of mass)	
Rotation Matrix	$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$
Linear Jacobian	$\begin{pmatrix} -\mathrm{cg}_{12}\sin(\theta_1(t))\sin(\theta_2(t)) & \mathrm{cg}_{12}\cos(\theta_1(t))\cos(\theta_2(t)) & 0 \\ \mathrm{cg}_{12}\cos(\theta_1(t))\sin(\theta_2(t)) & \mathrm{cg}_{12}\cos(\theta_2(t))\sin(\theta_1(t)) & 0 \\ 0 & \mathrm{cg}_{12}\sin(\theta_2(t)) & 0 \end{pmatrix}$
Angular Jacobian	$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$
Kinetic Energy Translational	$\frac{M_2\,\mathrm{cg}_{12}^2\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}+\frac{M_2\,\mathrm{cg}_{12}^2\left(\frac{\partial}{\partial t}\theta_2(t)\right)^2}{2}-\frac{M_2\,\mathrm{cg}_{12}^2\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}$
Kinetic Energy Rotational	$\frac{I_{2XX}\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}+\frac{I_{2XX}\left(\frac{\partial}{\partial t}\theta_2(t)\right)^2}{2}+\frac{I_{2ZZ}\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}-\frac{I_{2XX}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}+\frac{I_{2ZZ}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}$
Potential Energy	$-M_2\,\mathrm{cg}_{12}\,g\cos(\theta_2(t))$

Table 6: Link 3 Dynamics	
Rotation Matrix	$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$
Linear Jacobian	$\begin{pmatrix} -\frac{\sin(\theta_1(t))\sin(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_1(t))\cos(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_1(t))\sin(\theta_2(t))}{2} \\ \frac{\cos(\theta_1(t))\sin(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_2(t))\sin(\theta_1(t))\,d_3(t)}{2} & \frac{\sin(\theta_1(t))\sin(\theta_2(t))}{2} \\ 0 & \frac{\sin(\theta_2(t))\,d_3(t)}{2} & -\frac{\cos(\theta_2(t))}{2} \end{pmatrix}$
Angular Jacobian	$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$
Kinetic Energy Translational	$\frac{M_3\sin(\theta_2(t))^2\,d_3(t)^2\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{8}+\frac{M_3\,d_3(t)^2\left(\frac{\partial}{\partial t}\theta_2(t)\right)^2}{8}+\frac{M_3\left(\frac{\partial}{\partial t}d_3(t)\right)^2}{8}$
Kinetic Energy Rotational	$\frac{I_{3XX}\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}+\frac{I_{3XX}\left(\frac{\partial}{\partial t}\theta_2(t)\right)^2}{2}+\frac{I_{3ZZ}\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}-\frac{I_{3XX}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}+\frac{I_{3ZZ}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{4}$
Potential Energy	$-\frac{M_3\,g\cos(\theta_2(t))\,d_3(t)}{2}$

Table 7: End Effector Dynamics	
Rotation Matrix	$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$
Linear Jacobian	$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$
Angular Jacobian	$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$
Kinetic Energy Translational	$\frac{M_4 \sin(\theta_2(t))^2 d_3(t)^2 \left(\frac{\partial}{\partial t}\theta_1(t)\right)^2}{2} + \frac{M_4 d_3(t)^2 \left(\frac{\partial}{\partial t}\theta_2(t)\right)^2}{2} + \frac{M_4 \left(\frac{\partial}{\partial t} d_3(t)\right)^2}{2}$
Kinetic Energy Rotational	N/A
Potential Energy	$-M_4\, g \cos(\theta_2(t))\, d_3(t)$

The Lagrange equations and the standard form are then developed from these values.

<u>Lagrange Equations:</u>

$$\frac{M_3\,\sigma_4}{8}+\frac{M_4\,\sigma_4}{2}+\frac{I_{2XX}\,\sigma_1}{2}+\frac{I_{2XX}\,\sigma_3}{2}+\frac{I_{3XX}\,\sigma_1}{2}+\frac{I_{3XX}\,\sigma_3}{2}+\frac{I_{1ZZ}\,\sigma_1}{2}+\frac{M_3\,d_3(t)^2\,\sigma_1}{8}+\frac{M_3\,d_3(t)^2\,\sigma_3}{8}+\frac{M_4\,d_3(t)^2\,\sigma_1}{2}+\frac{M_4\,d_3(t)^2\,\sigma_3}{2}-\frac{I_{2XX}\,\sigma_2\,\sigma_1}{2}-\frac{I_{3XX}\,\sigma_2\,\sigma_1}{2}+\frac{I_{2ZZ}\,\sigma_2\,\sigma_1}{2}+\frac{I_{3ZZ}\,\sigma_2\,\sigma_1}{2}+\frac{M_2\,\mathrm{cg}_{12}{}^2\,\sigma_1}{2}+\frac{M_2\,\mathrm{cg}_{12}{}^2\,\sigma_3}{2}+\frac{M_3\,g\cos(\theta_2(t))\,d_3(t)}{2}+M_4\,g\cos(\theta_2(t))\,d_3(t)-\frac{M_2\,\mathrm{cg}_{12}{}^2\,\sigma_2\,\sigma_1}{2}-\frac{M_3\,\sigma_2\,d_3(t)^2\,\sigma_1}{8}-\frac{M_4\,\sigma_2\,d_3(t)^2\,\sigma_1}{2}+M_2\,\mathrm{cg}_{12}\,g\cos(\theta_2(t))$$

where

$$\sigma_1=\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2$$

$$\sigma_2=\cos(\theta_2(t))^2$$

$$\sigma_3=\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2$$

$$\sigma_4=\left(\frac{\partial}{\partial t}\,d_3(t)\right)^2$$

Figure 6: Lagrange Equation

$$\begin{pmatrix} I_{1ZZ}+I_{2ZZ}+I_{3ZZ}+I_{2XX}\,\sigma_1+I_{3XX}\,\sigma_1-I_{2ZZ}\,\sigma_1-I_{3ZZ}\,\sigma_1+M_2\,\mathrm{cg}_{12}{}^2\,\sigma_1+\frac{M_3\,\sigma_1\,d_3(t)^2}{4}+M_4\,\sigma_1\,d_3(t)^2 & 0 & 0 \\ 0 & I_{2XX}+I_{3XX}+M_2\,\mathrm{cg}_{12}{}^2+\frac{M_3\,d_3(t)^2}{4}+M_4\,d_3(t)^2 & 0 \\ 0 & 0 & \frac{M_3}{4}+M_4 \end{pmatrix}$$

where

$$\sigma_1=\sin(\theta_2(t))^2$$

Figure 7: Mass Matrix

$$\begin{pmatrix} \frac{d_3(t)\ (M_3+4\,M_4)\sin(\theta_2(t))^2 \frac{\partial}{\partial t}\ d_3(t)}{4} + \sin(2\,\theta_2(t))\,\sigma_4 \frac{\partial}{\partial t}\ \theta_2(t) & \sigma_3 & \sigma_1 \\ -\sigma_3 & \frac{d_3(t)\ (M_3+4\,M_4) \frac{\partial}{\partial t}\ d_3(t)}{4} & \sigma_2 \\ -\sigma_1 & -\sigma_2 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \frac{\sin(\theta_2(t))^2\, d_3(t)\ (M_3+4\,M_4) \frac{\partial}{\partial t}\ \theta_1(t)}{4}$$

$$\sigma_2 = \frac{d_3(t)\ (M_3+4\,M_4) \frac{\partial}{\partial t}\ \theta_2(t)}{4}$$

$$\sigma_3 = \sin(2\,\theta_2(t)) \frac{\partial}{\partial t}\ \theta_1(t)\,\sigma_4$$

$$\sigma_4 = \frac{I_{2XX}}{2} + \frac{I_{3XX}}{2} - \frac{I_{2ZZ}}{2} - \frac{I_{3ZZ}}{2} + \frac{M_2\,{cg_{12}}^2}{2} + \frac{M_3\,d_3(t)^2}{8} + \frac{M_4\,d_3(t)^2}{2}$$

Figure 8: Coriolis Matrix

$$\begin{pmatrix} 0 & -\sigma_1 & -\sigma_2 \\ \sigma_1 & 0 & -\frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_2(t)}{2} \\ \sigma_2 & \frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_2(t)}{2} & 0 \end{pmatrix}$$

where

$$\sigma_1 = 2\sin(2\,\theta_2(t))\,\frac{\partial}{\partial t}\ \theta_1(t)\ \left(\frac{I_{2XX}}{2}+\frac{I_{3XX}}{2}-\frac{I_{2ZZ}}{2}-\frac{I_{3ZZ}}{2}+\frac{M_2\,\mathrm{cg}_{12}{}^2}{2}+\frac{M_3\,d_3(t)^2}{8}+\frac{M_4\,d_3(t)^2}{2}\right)$$

$$\sigma_2 = \frac{\sin(\theta_2(t))^2\,d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_1(t)}{2}$$

Figure 9: $\dot{M}$-2C

$$\begin{pmatrix} 0 \\ \frac{g\sin(\theta_2(t))\ (2\,M_2\,\mathrm{cg}_{12}+M_3\,d_3(t)+2\,M_4\,d_3(t))}{2} \\ -\frac{g\cos(\theta_2(t))\ (M_3+2\,M_4)}{2} \end{pmatrix}$$

Figure 10: Gravity Array

Roman Yoder
Mech 612: Project 1 Report

<u>Derive Adaptive Controller:</u>

$$\begin{pmatrix} \sin(\theta_2(t))\, d_3(t)\, \left(\mathrm{dtheta1r}\, \sin(\theta_2(t))\, \frac{\partial}{\partial t}\, d_3(t) + \mathrm{dd3r}\, \sin(\theta_2(t))\, \frac{\partial}{\partial t}\, \theta_1(t) + \mathrm{ddtheta1r}\, \sin(\theta_2(t))\, d_3(t) + \mathrm{dtheta1r}\, \cos(\theta_2(t))\, d_3(t)\, \frac{\partial}{\partial t}\, \theta_2(t) + \mathrm{dtheta2r}\, \cos(\theta_2(t))\, d_3(t)\, \frac{\partial}{\partial t}\, \theta_1(t)\right) \\ \dfrac{d_3(t)\, \left(2\, g\, \sin(\theta_2(t)) + 2\, \mathrm{dtheta2r}\, \frac{\partial}{\partial t}\, d_3(t) + 2\, \mathrm{dd3r}\, \frac{\partial}{\partial t}\, \theta_2(t) + 2\, \mathrm{ddtheta2r}\, d_3(t) - \mathrm{dtheta1r}\, \sin(2\, \theta_2(t))\, d_3(t)\, \frac{\partial}{\partial t}\, \theta_1(t)\right)}{2} \\ -\mathrm{dtheta1r}\, d_3(t)\, \sin(\theta_2(t))^2\, \frac{\partial}{\partial t}\, \theta_1(t) - \mathrm{dtheta2r}\, d_3(t)\, \frac{\partial}{\partial t}\, \theta_2(t) + \mathrm{ddd3r} - g\, \cos(\theta_2(t)) \end{pmatrix}$$

Figure 15: Y_1

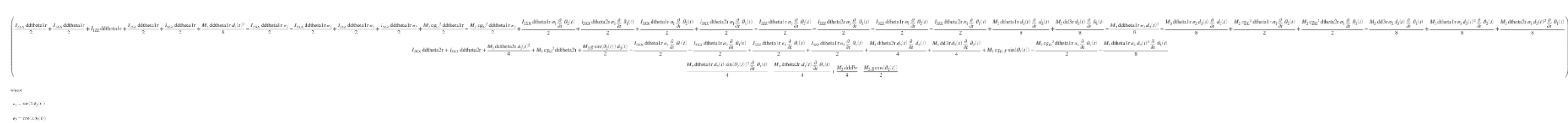


Figure 16: Y_0 (no good way to present this)

Desired Trajectory:

Trajectories were selected based on moving the arm from an assembly line to a higher assembly line perpendicular to the first.

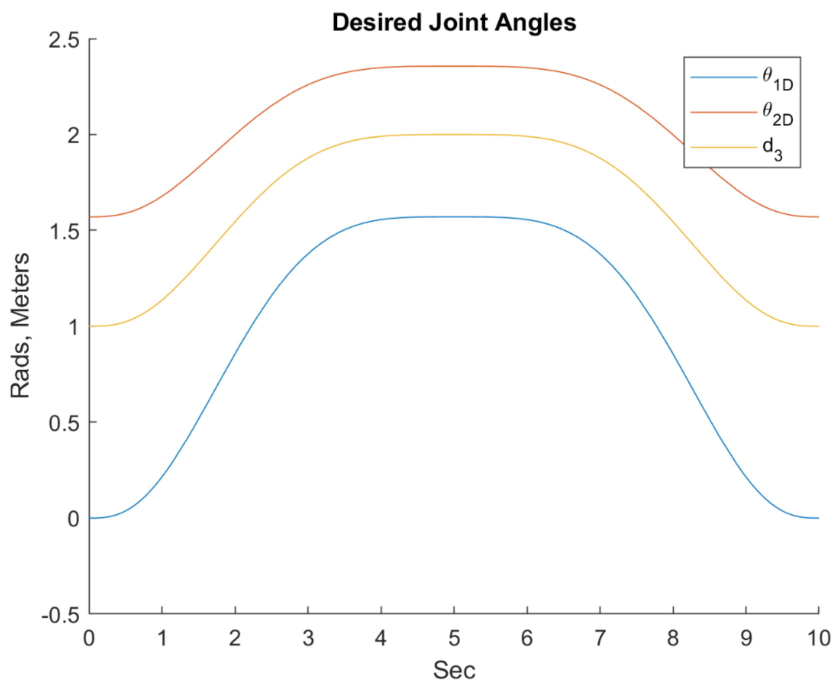


Figure 11

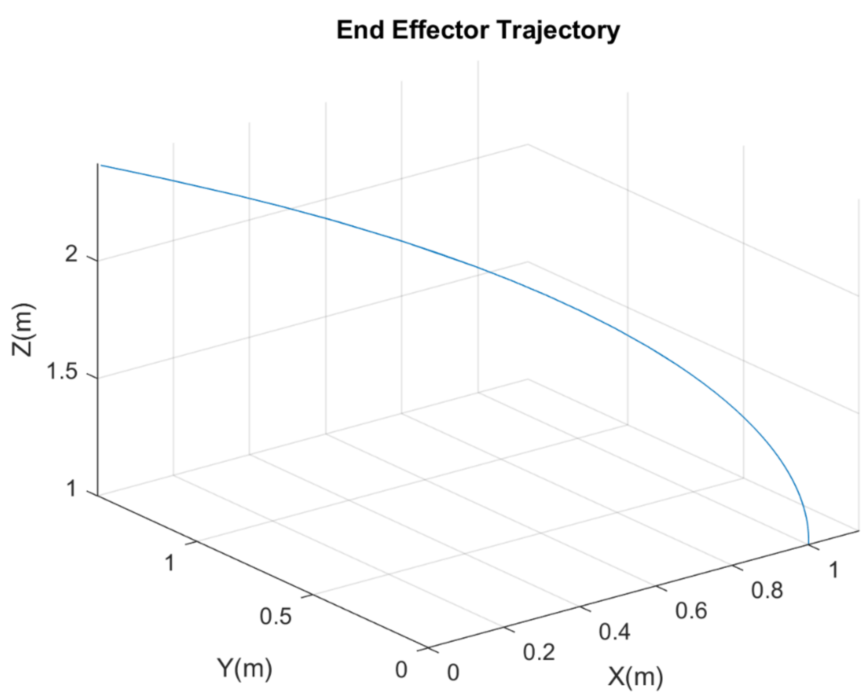


Figure 12

Proportional Control No Gravity Adjustment:

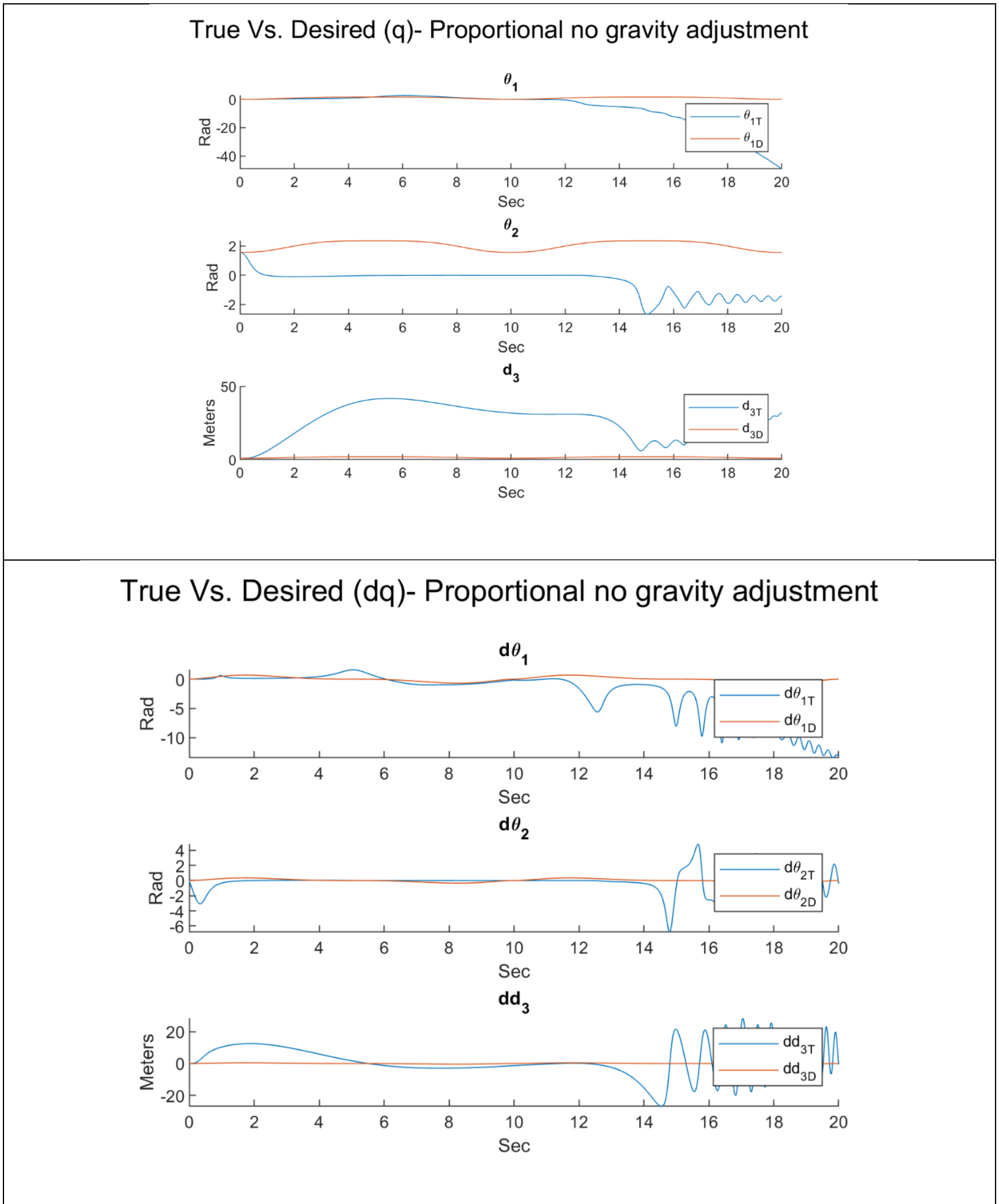


Figure 13

Proportional Control Gravity Adjustment:

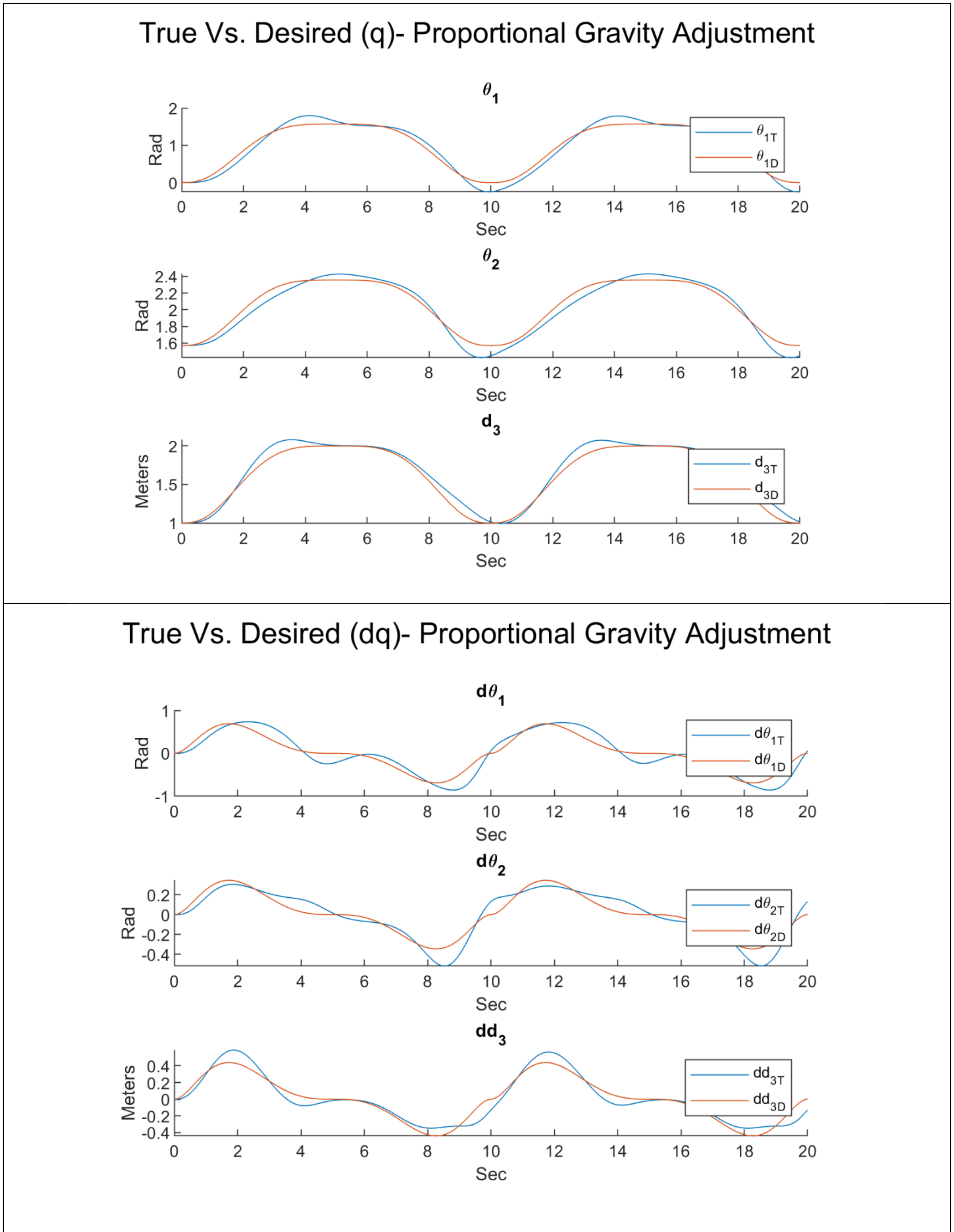


Figure 14

Computed Torque:

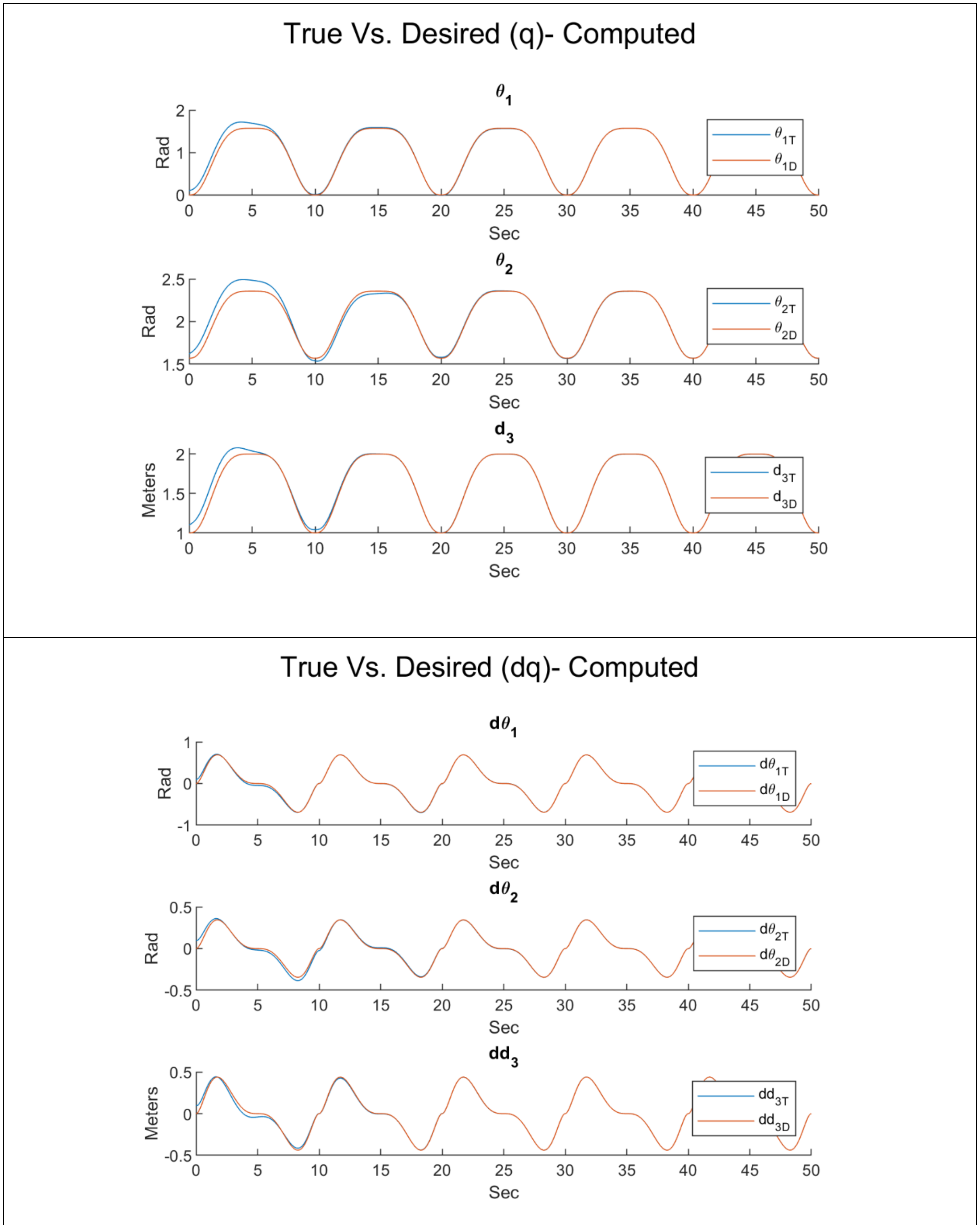
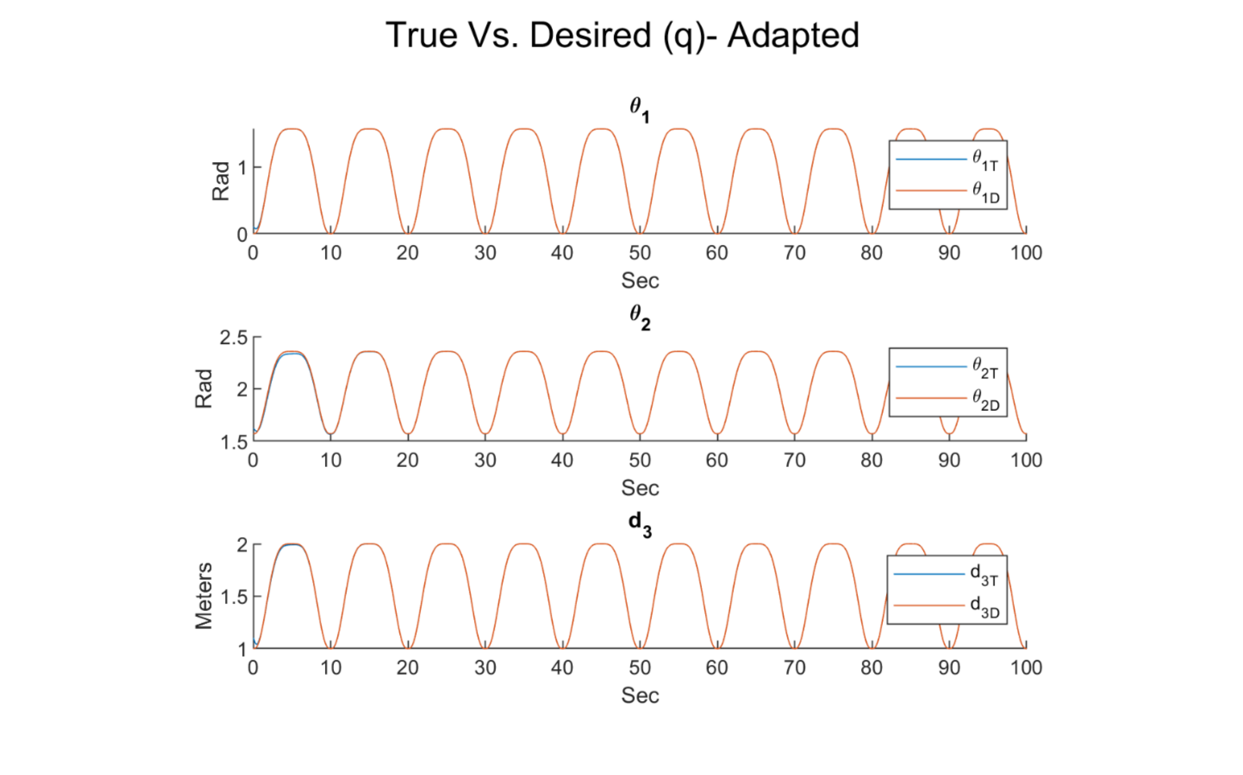


Figure 15

Adapted Torque:



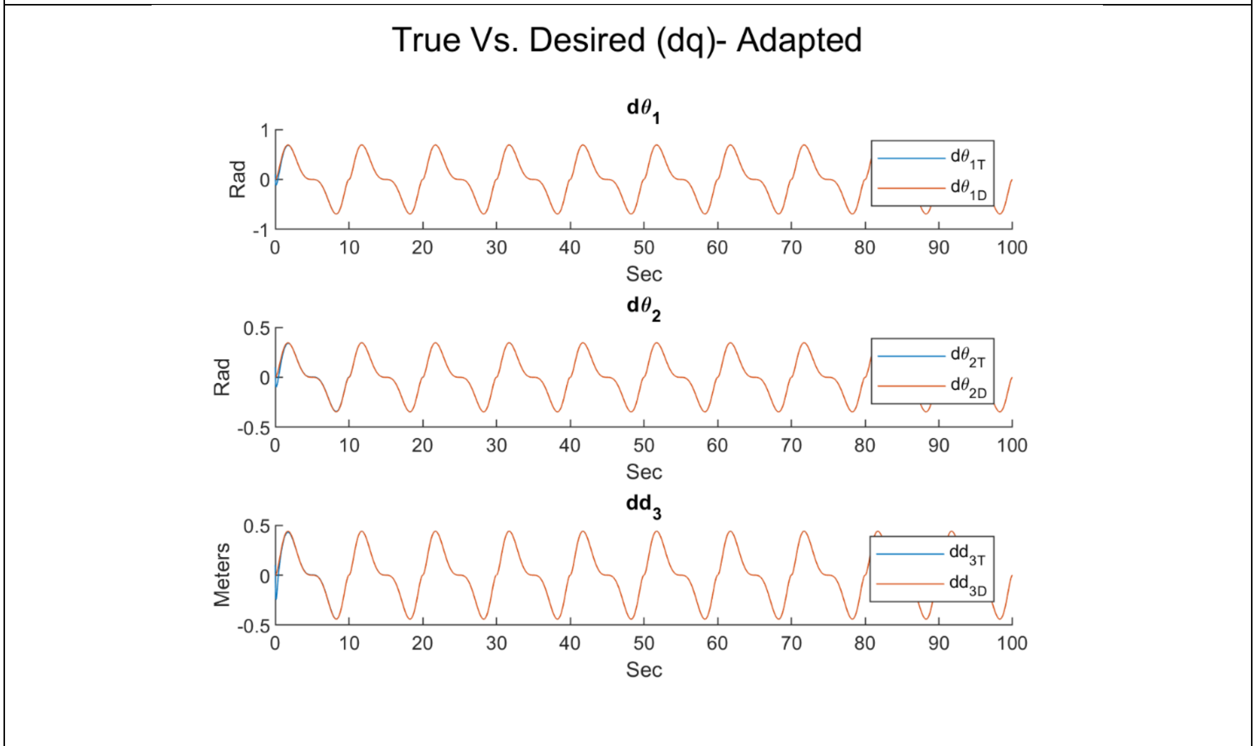


Figure 16

3D plot of Adapted Controller:

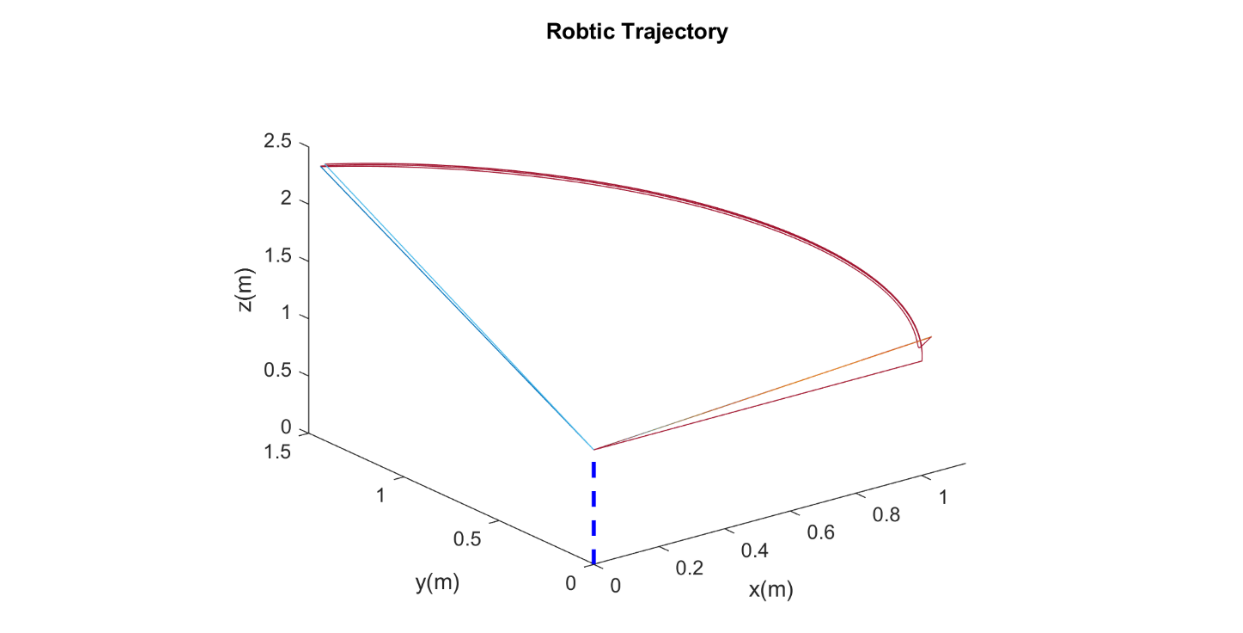


Figure 17

Obstacle Collision:

The current trajectory was then made to collide with an object in the workspace.

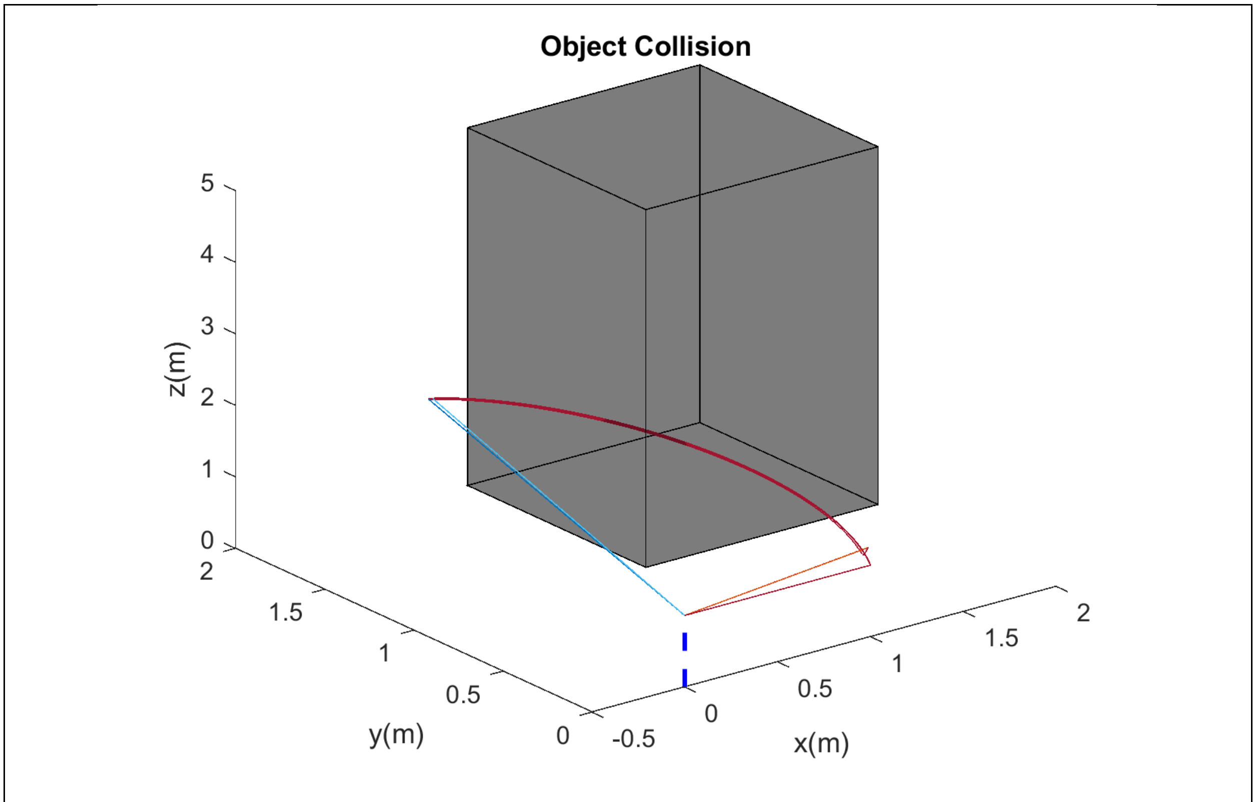


Figure 18

Redesigned Trajectory:

A trajectory was then redesigned as follows.

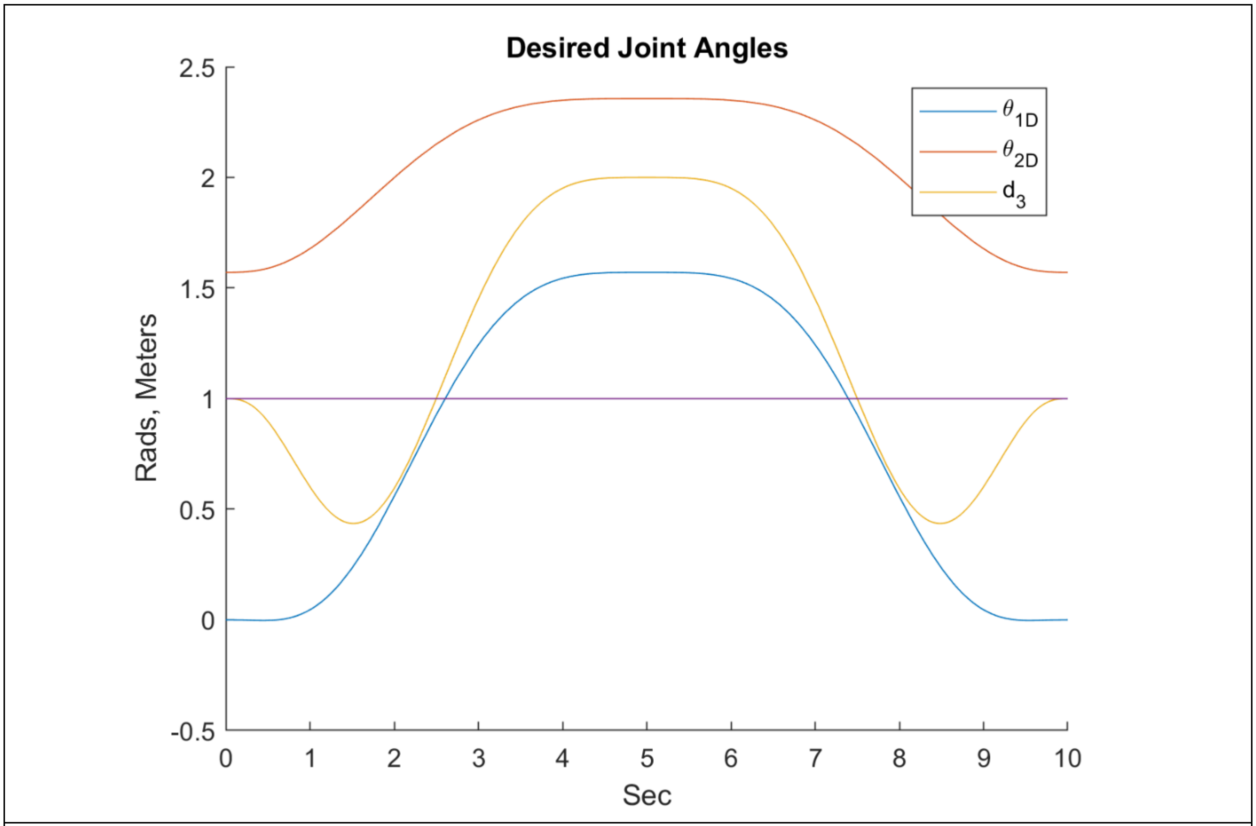


Figure 19

Here a piecewise trajectory was used for d_3 in which it is zero (blue line) up until it passed the collision point and then follows the yellow line

Controlled Object Avoidance:

This new trajectory was then implemented with if statements for the piecewise function.

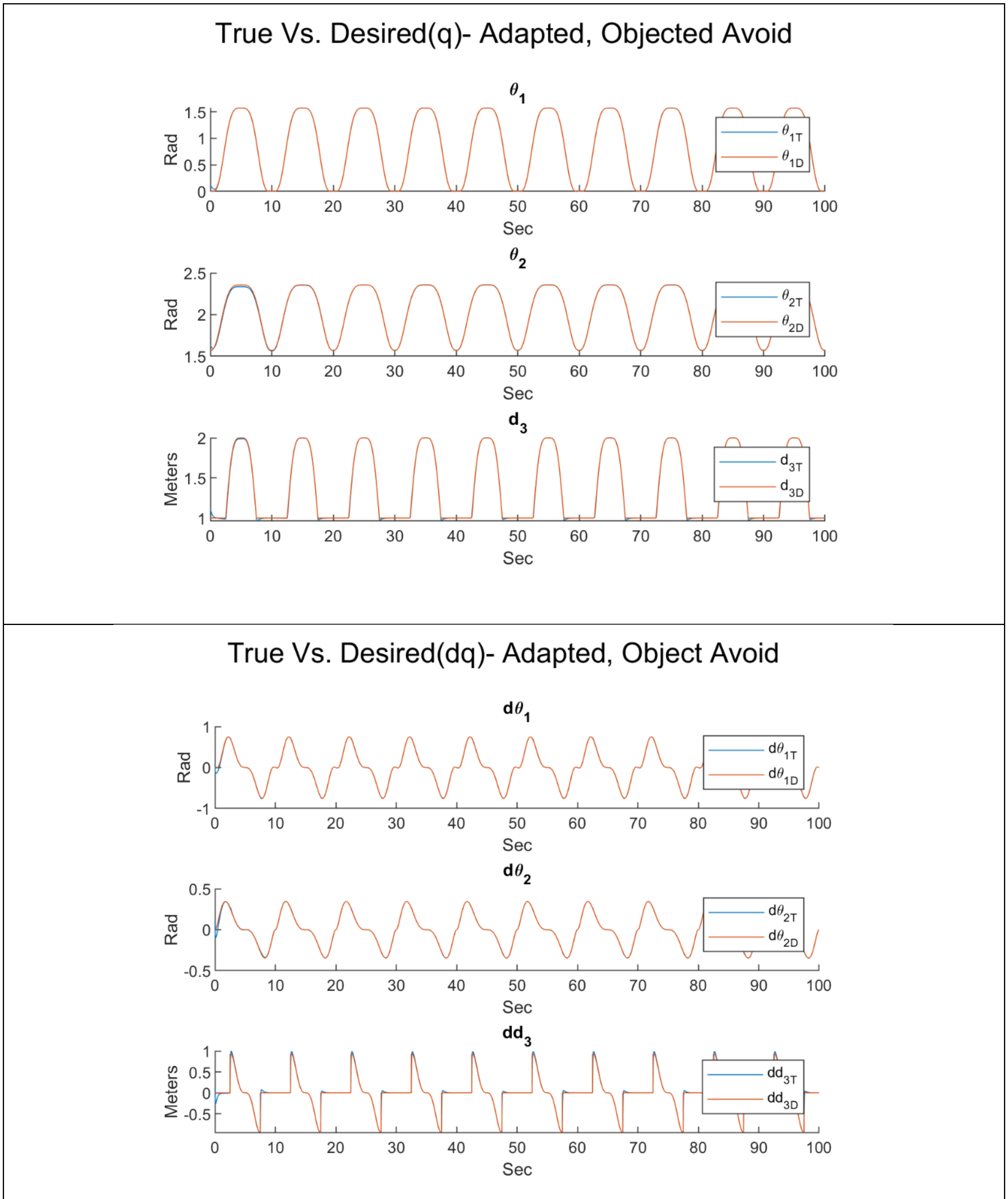


Figure 20

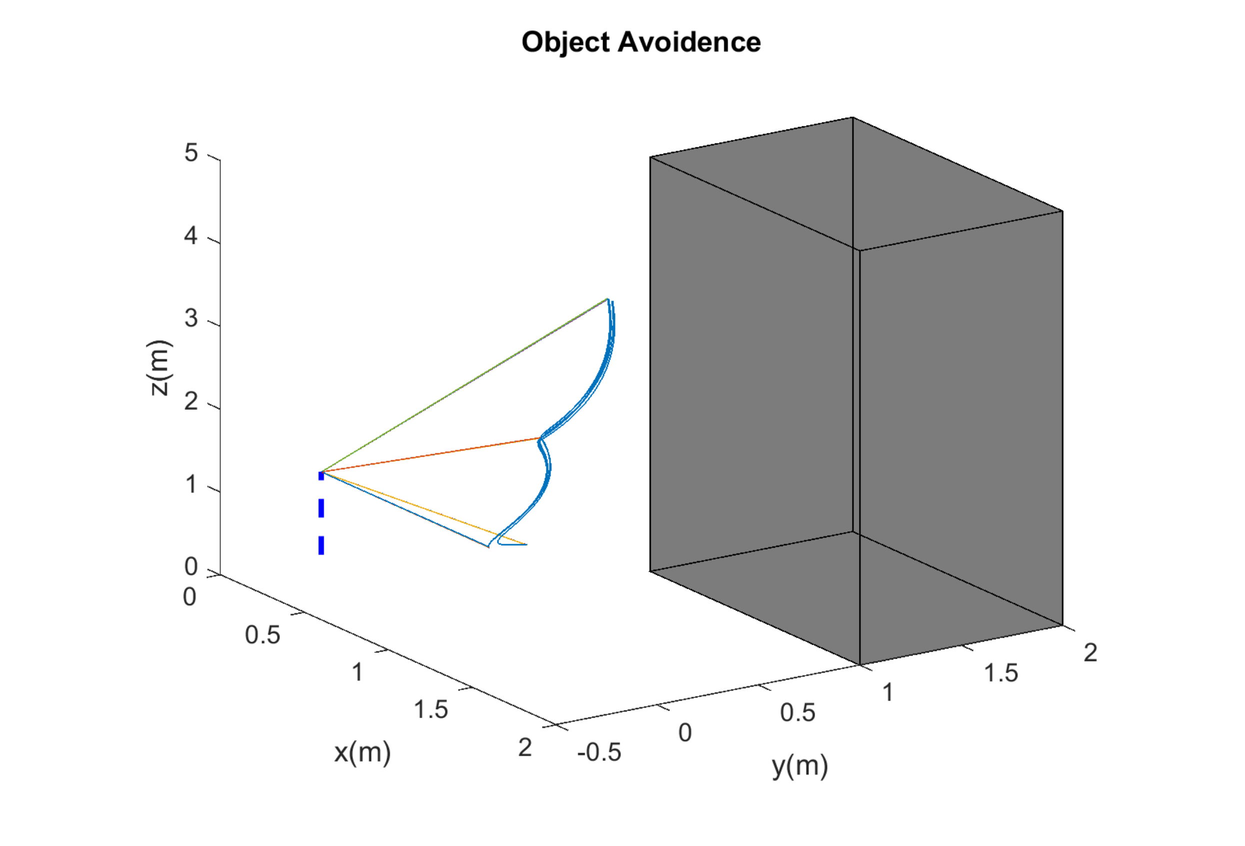
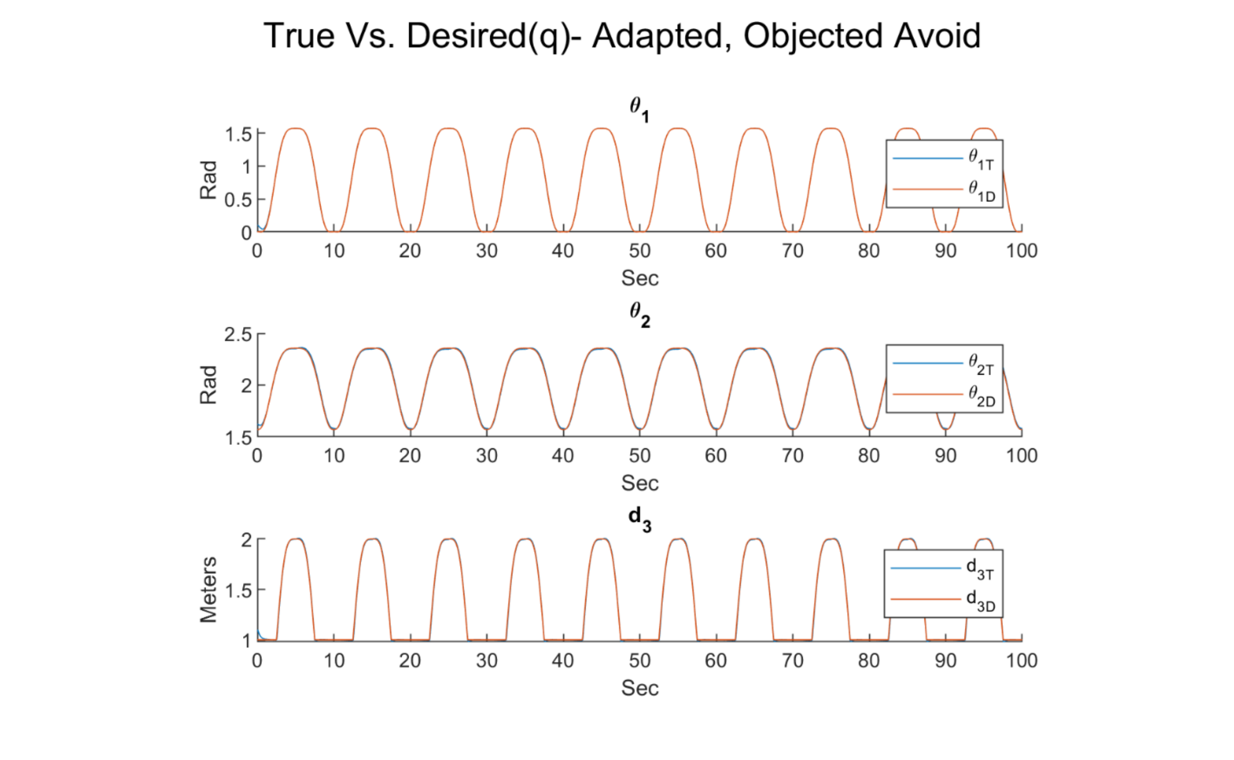


Figure 21

Accurate Simulation:

To increase the accuracy of the simulation and test the robustness of the adaptive control mass must be modulated. This was done to reflect use in an industrial application where the end effector would at first carry a pay load and have a total mass of 15 kg, once it reaches the second assembly line halfway through its period its mass decreases to 12 kg as it drops its pay load.


True Vs. Desired(q)- Adapted, Objected Avoid
θ_1
Rad
Sec
θ_{1T}
θ_{1D}
θ_2
Rad
Sec
θ_{2T}
θ_{2D}
d_3
Meters
Sec
d_{3T}
d_{3D}

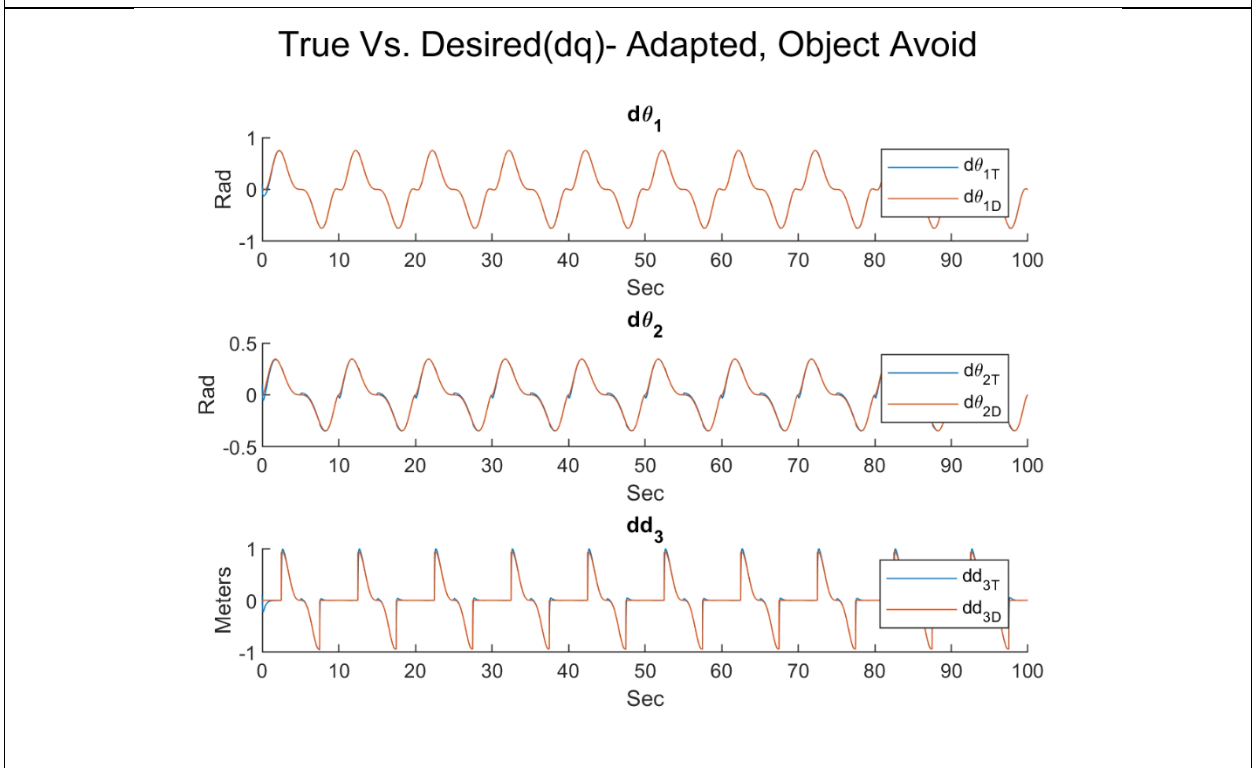

True Vs. Desired(dq)- Adapted, Object Avoid
$d\theta_1$
Rad
Sec
$d\theta_{1T}$
$d\theta_{1D}$
$d\theta_2$
Rad
Sec
$d\theta_{2T}$
$d\theta_{2D}$
dd_3
Meters
Sec
dd_{3T}
dd_{3D}

Figure 22

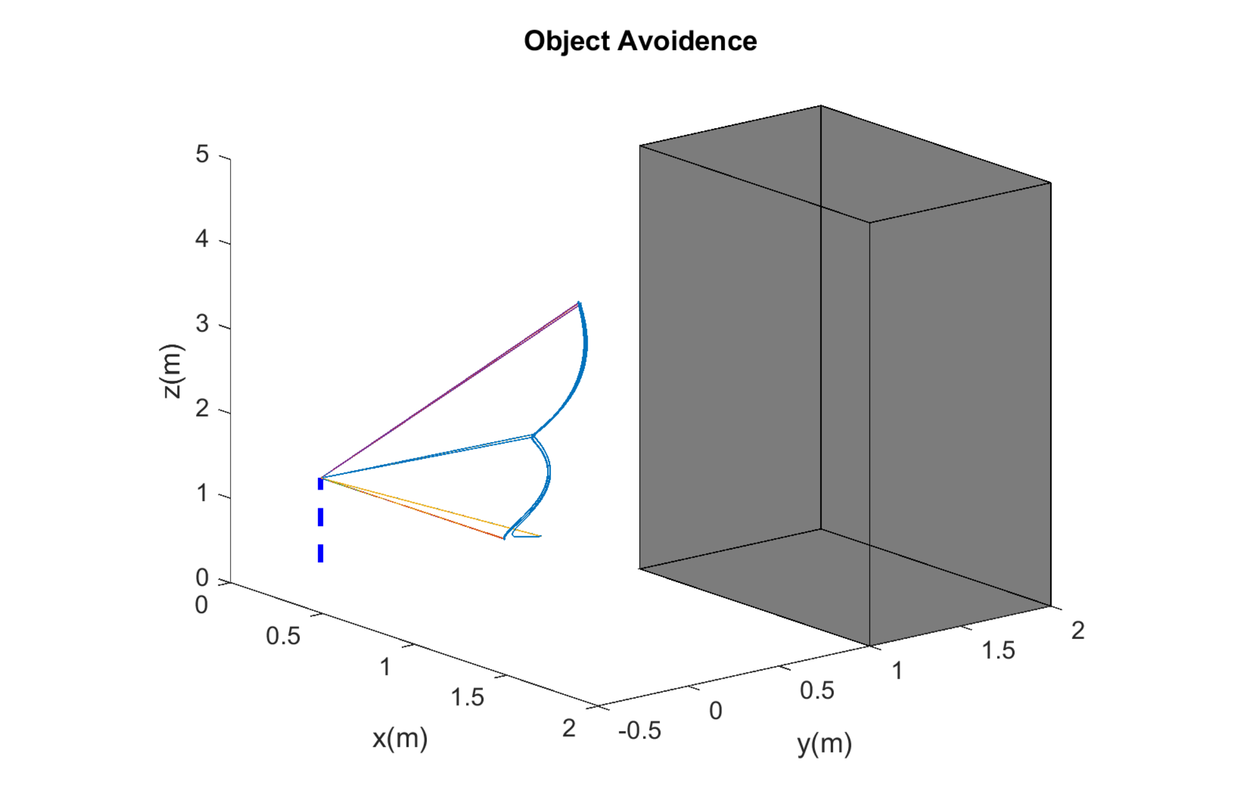


Figure 23

Each simulation was also tested with initial errors to prove their robustness.

Simulation Flow Chart:

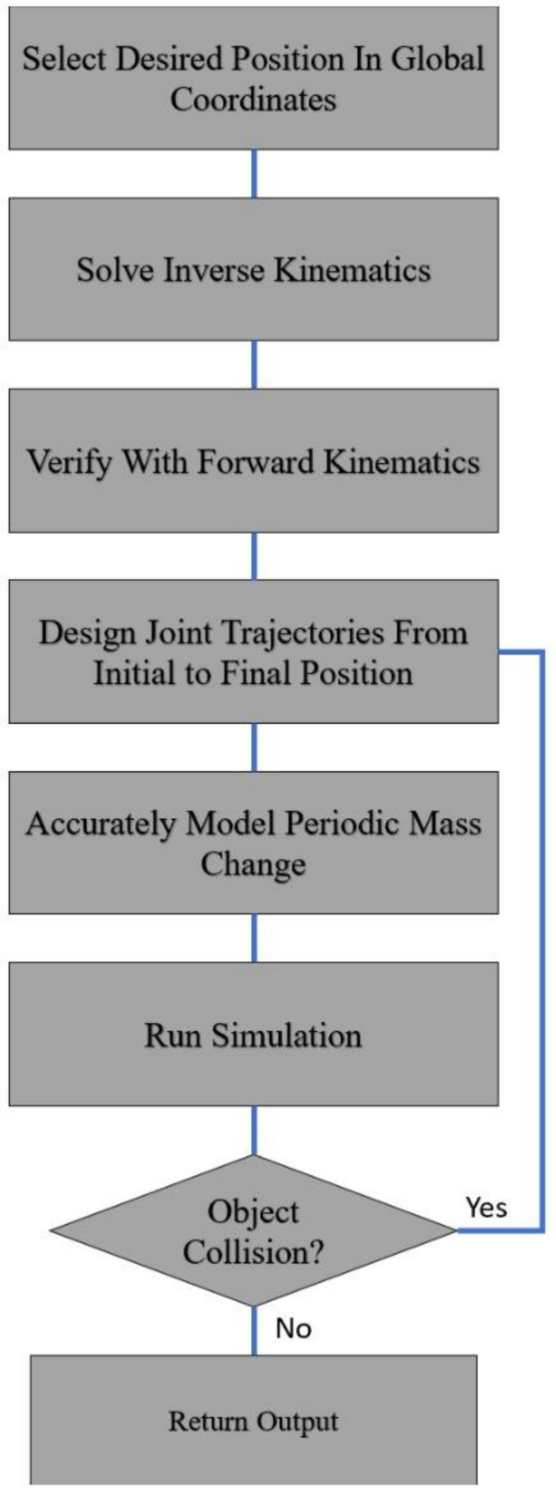


Figure 24: Simplified Simulation Flowchart

Discussion/Conclusion:

Overall, all control algorithms worked for the desired trajectory, except proportional control that lacked gravity compensation. This control would have only worked for a local minimum which is not occurring in these trajectories.

Of the control algorithms that did work, necessary gain was vastly different from one method to the next. Being that proportional control required the highest gains when compared to computed or adaptive control. The true values of this proportional control could be improved by further raising (or tuning) the gain values, but this would put more demand on a real-world controller. The beauty of the computed torque method is the feed forward term represented by the first and second derivate of the desired trajectory. This estimates the coming change that will be necessary and accounts for them in the torque law, minimizing the error that can develop at every turn. For this reason, this requires lower gains. However, the only mass this controller can account for is known mass. Yes, if on an assembly line with an extremely repetitive process where the periodic change in mass is known, one could program an update feature for the end effector mass. However, this is not nearly as robust as the adaptive control for several reasons.

First, if the adaptive control becomes slightly out of sink or maintains a mass longer then the predicted time it will still account for this unknown mass in its torque law. Second, if there are variations between payloads of individual products the adaptive control will account for this. Third in industrial tooling application, if bits wear out and more applied force is needed over time the adaptive control will self-adjust for this.

It should be noted that many of the presented simulations are not a perfect reflection of physical implementation nor are all the solutions optimum. For example, the first trajectory was calculated simply from polynomial interpolation, it is likely that this solution is close to the optimum path, but not proven. The trajectory for the obstacle avoidance was selected simply because it avoided the added object in the workspace and is not optimized. However, the piecewise function was implemented to reduce the number of unnecessary movements in the prismatic joint as there is no need to initially retract.

In the final simulation presented, an as close to possible real-world simulation was created. With the mass of the end effector repeatedly changing over each period as a product is lifted and then released. Increasing this change in mass resulted in poorer tracking of the desired trajectory; but this could be compensated for by retuning the gains of the system. This is equivalent to providing more controller output and instantaneous torque from the motors. It was shown that the current tuning could not only account for an initial error, but then adjust for a repeated change of 3 kg and still track the desired path.

Overall this project was a great opportunity to gain experience in the trajectory design and control implementation of using robotics to solve real world applications.

Appendix (Code):

FinalPR1Live.pdf

AccurateSim.pdf

ACOASS.pdf

ACOA.pdf

PGA.pdf

AC.pdf

CT.pdf

PNG.pdf

```
function dydt = PNG(t,y)

M1=50;
M2=25;
M3=12.5;
M4=10;
cg_l2=0.5;
g=9.807;

X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625;
(12*pi)/78125; -(3*pi)/781250];
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625;
(6*pi)/78125; -(3*pi)/1562500];
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125);
-(3/390625)];

I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set states
theta1=y(1);
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);

%Desired Trajectories
% desq1=subs(destheta1,(t-floor(t/10)));
% desq2=subs(destheta2,(t-floor(t/10)));
% desq3=subs(desd3,(t-floor(t/10)));
% desdq1=subs(desdtheta1,(t-floor(t/10)));
% desdq2=subs(desdtheta2,(t-floor(t/10)));
% desdq3=subs(desdd3,(t-floor(t/10)));

desq1=X1(9)*(t-floor(t/10))^8+X1(8)*(t-floor(t/10))^7+X1(7)*(t-floor
(t/10))^6+X1(6)*(t-floor(t/10))^5+X1(5)*(t-floor(t/10))^4+X1(4)*(t-floor
(t/10))^3+X1(3)*(t-floor(t/10))^2+X1(2)*(t-floor(t/10))+X1(1);
```

```
desq2=X2(9)*(t-floor(t/10))^8+X2(8)*(t-floor(t/10))^7+X2(7)*(t-floor↙
(t/10))^6+X2(6)*(t-floor(t/10))^5+X2(5)*(t-floor(t/10))^4+X2(4)*(t-floor↙
(t/10))^3+X2(3)*(t-floor(t/10))^2+X2(2)*(t-floor(t/10))+X2(1);
desq3=X3(9)*(t-floor(t/10))^8+X3(8)*(t-floor(t/10))^7+X3(7)*(t-floor↙
(t/10))^6+X3(6)*(t-floor(t/10))^5+X3(5)*(t-floor(t/10))^4+X3(4)*(t-floor↙
(t/10))^3+X3(3)*(t-floor(t/10))^2+X3(2)*(t-floor(t/10))+X3(1);
desdq1=8*X1(9)*(t-floor(t/10))^7+7*X1(8)*(t-floor(t/10))^6+6*X1(7)*(t-↙
floor(t/10))^5+5*X1(6)*(t-floor(t/10))^4+4*X1(5)*(t-floor(t/10))^3+3*X1↙
(4)*(t-floor(t/10))^2+2*X1(3)*(t-floor(t/10))+X1(2);
desdq2=8*X2(9)*(t-floor(t/10))^7+7*X2(8)*(t-floor(t/10))^6+6*X2(7)*(t-↙
floor(t/10))^5+5*X2(6)*(t-floor(t/10))^4+4*X2(5)*(t-floor(t/10))^3+3*X2↙
(4)*(t-floor(t/10))^2+2*X2(3)*(t-floor(t/10))+X2(2);
desdq3=8*X3(9)*(t-floor(t/10))^7+7*X3(8)*(t-floor(t/10))^6+6*X3(7)*(t-↙
floor(t/10))^5+5*X3(6)*(t-floor(t/10))^4+4*X3(5)*(t-floor(t/10))^3+3*X3↙
(4)*(t-floor(t/10))^2+2*X3(3)*(t-floor(t/10))+X3(2);

%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];
qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];
qe=q-qd;
dqe=dq-dqd;
%Set gains
Kp=5;
Kd=7;

Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin↙
(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(theta2)↙
^2*d3^2)/4+M4*sin(theta2)^2*d3^2,0,0;0,I_2XX+I_3XX+M2*cg_l2^2+(M3*d3^2)↙
/4+M4*d3^2,0;0,0,M3/4+M4];
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 +↙
I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)↙
*dtheta2,sin(2*theta2)*dtheta1*(I_2XX/2 + I_3XX/2 - I_2ZZ/2 - I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(sin(theta2)^2*d3*(M3 + 4*M4)↙
*dtheta1)/4;-sin(2*theta2)*dtheta1*(I_2XX/2+I_3XX/2-I_2ZZ/2-I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(d3*(M3+4*M4)*dd3)/4,(d3*↙
(M3+4*M4)*dtheta2)/4;-(sin(theta2)^2*d3*(M3+4*M4)*dtheta1)/4,-(d3*↙
(M3+4*M4)*dtheta2)/4,0];
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*(M3↙
+ 2*M4))/2];
```

```
Tau=-Kp*(qe)-Kd*(dqe);
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3)];

end
```

```
function dydt = CT(t,y)

M1=50;
M2=25;
M3=12.5;
M4=10;
cg_l2=0.5;
g=9.807;

X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625;↙
(12*pi)/78125; -(3*pi)/781250];
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625;↙
(6*pi)/78125; -(3*pi)/1562500];
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125);↙
-(3/390625)];

I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set states
theta1=y(1);
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);

%Desired Trajectories
% desq1=subs(destheta1,t);
% desq2=subs(destheta2,t);
% desq3=subs(desd3,t);
% desdq1=subs(desdtheta1,t);
% desdq2=subs(desdtheta2,t);
% desdq3=subs(desdd3,t);

desq1=X1(9)*(t-10*floor(t/10))^8+X1(8)*(t-10*floor(t/10))^7+X1(7)*(t-↙
10*floor(t/10))^6+X1(6)*(t-10*floor(t/10))^5+X1(5)*(t-10*floor(t/10))↙
^4+X1(4)*(t-10*floor(t/10))^3+X1(3)*(t-10*floor(t/10))^2+X1(2)*(t-↙
```

```
10*floor(t/10))+X1(1);
desq2=X2(9)*(t-10*floor(t/10))^8+X2(8)*(t-10*floor(t/10))^7+X2(7)*(t-↙
10*floor(t/10))^6+X2(6)*(t-10*floor(t/10))^5+X2(5)*(t-10*floor(t/10))↙
^4+X2(4)*(t-10*floor(t/10))^3+X2(3)*(t-10*floor(t/10))^2+X2(2)*(t-↙
10*floor(t/10))+X2(1);
desq3=X3(9)*(t-10*floor(t/10))^8+X3(8)*(t-10*floor(t/10))^7+X3(7)*(t-↙
10*floor(t/10))^6+X3(6)*(t-10*floor(t/10))^5+X3(5)*(t-10*floor(t/10))↙
^4+X3(4)*(t-10*floor(t/10))^3+X3(3)*(t-10*floor(t/10))^2+X3(2)*(t-↙
10*floor(t/10))+X3(1);
desdq1=8*X1(9)*(t-10*floor(t/10))^7+7*X1(8)*(t-10*floor(t/10))^6+6*X1(7)↙
*(t-10*floor(t/10))^5+5*X1(6)*(t-10*floor(t/10))^4+4*X1(5)*(t-10*floor↙
(t/10))^3+3*X1(4)*(t-10*floor(t/10))^2+2*X1(3)*(t-10*floor(t/10))+X1(2);
desdq2=8*X2(9)*(t-10*floor(t/10))^7+7*X2(8)*(t-10*floor(t/10))^6+6*X2(7)↙
*(t-10*floor(t/10))^5+5*X2(6)*(t-10*floor(t/10))^4+4*X2(5)*(t-10*floor↙
(t/10))^3+3*X2(4)*(t-10*floor(t/10))^2+2*X2(3)*(t-10*floor(t/10))+X2(2);
desdq3=8*X3(9)*(t-10*floor(t/10))^7+7*X3(8)*(t-10*floor(t/10))^6+6*X3(7)↙
*(t-10*floor(t/10))^5+5*X3(6)*(t-10*floor(t/10))^4+4*X3(5)*(t-10*floor↙
(t/10))^3+3*X3(4)*(t-10*floor(t/10))^2+2*X3(3)*(t-10*floor(t/10))+X3(2);
desddq1=56*X1(9)*(t-10*floor(t/10))^6+42*X1(8)*(t-10*floor(t/10))↙
^5+30*X1(7)*(t-10*floor(t/10))^4+20*X1(6)*(t-10*floor(t/10))^3+12*X1(5)*↙
(t-10*floor(t/10))^2+6*X1(4)*(t-10*floor(t/10))+2*X1(3);
desddq2=56*X2(9)*(t-10*floor(t/10))^6+42*X2(8)*(t-10*floor(t/10))↙
^5+30*X2(7)*(t-10*floor(t/10))^4+20*X2(6)*(t-10*floor(t/10))^3+12*X2(5)*↙
(t-10*floor(t/10))^2+6*X2(4)*(t-10*floor(t/10))+2*X2(3);
desddq3=56*X3(9)*(t-10*floor(t/10))^6+42*X3(8)*(t-10*floor(t/10))↙
^5+30*X3(7)*(t-10*floor(t/10))^4+20*X3(6)*(t-10*floor(t/10))^3+12*X3(5)*↙
(t-10*floor(t/10))^2+6*X3(4)*(t-10*floor(t/10))+2*X3(3);



%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];

qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];

ddqd=[desddq1;desddq2;desddq3];

qe=q-qd;
dqe=dq-dqd;
```

```
%Set gains
Kp=5;
Kd=10;

Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin↙
(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(theta2)↙
^2*d3^2)/4+M4*sin(theta2)^2*d3^2,0,0;0,I_2XX+I_3XX+M2*cg_l2^2+(M3*d3^2)↙
/4+M4*d3^2,0;0,0,M3/4+M4];
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 +↙
I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)↙
*dtheta2,sin(2*theta2)*dtheta1*(I_2XX/2 + I_3XX/2 - I_2ZZ/2 - I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(sin(theta2)^2*d3*(M3 + 4*M4)↙
*dtheta1)/4;-sin(2*theta2)*dtheta1*(I_2XX/2+I_3XX/2-I_2ZZ/2-I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(d3*(M3+4*M4)*dd3)/4,(d3*↙
(M3+4*M4)*dtheta2)/4;-(sin(theta2)^2*d3*(M3+4*M4)*dtheta1)/4,-(d3*↙
(M3+4*M4)*dtheta2)/4,0];
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*↙
(M3+2*M4))/2];


Tau=Mre*ddqd+C*dqd+G-Kp*qe-Kd*dqe;
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3)];

end
```

```
function dydt = AC(t,y)

M1=50;
M2=25;
M3=12.5;
M4=15;

% if t>20
%     M4=200;
% end

cg_l2=0.5;
g=9.807;

X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625;↙
(12*pi)/78125; -(3*pi)/781250];
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625;↙
(6*pi)/78125; -(3*pi)/1562500];
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125);↙
-(3/390625)];

I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set gains
%Kp=5;
Kd=50;
Lambda=25;
gamma=20;

%Set states
theta1=y(1);
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);
mh=y(7);
```

```
%Desired Trajectories
% desq1=subs(destheta1,t);
% desq2=subs(destheta2,t);
% desq3=subs(desd3,t);
% desdq1=subs(desdtheta1,t);
% desdq2=subs(desdtheta2,t);
% desdq3=subs(desdd3,t);

desq1=X1(9)*(t-10*floor(t/10))^8+X1(8)*(t-10*floor(t/10))^7+X1(7)*(t-↙
10*floor(t/10))^6+X1(6)*(t-10*floor(t/10))^5+X1(5)*(t-10*floor(t/10))↙
^4+X1(4)*(t-10*floor(t/10))^3+X1(3)*(t-10*floor(t/10))^2+X1(2)*(t-↙
10*floor(t/10))+X1(1);
desq2=X2(9)*(t-10*floor(t/10))^8+X2(8)*(t-10*floor(t/10))^7+X2(7)*(t-↙
10*floor(t/10))^6+X2(6)*(t-10*floor(t/10))^5+X2(5)*(t-10*floor(t/10))↙
^4+X2(4)*(t-10*floor(t/10))^3+X2(3)*(t-10*floor(t/10))^2+X2(2)*(t-↙
10*floor(t/10))+X2(1);
desq3=X3(9)*(t-10*floor(t/10))^8+X3(8)*(t-10*floor(t/10))^7+X3(7)*(t-↙
10*floor(t/10))^6+X3(6)*(t-10*floor(t/10))^5+X3(5)*(t-10*floor(t/10))↙
^4+X3(4)*(t-10*floor(t/10))^3+X3(3)*(t-10*floor(t/10))^2+X3(2)*(t-↙
10*floor(t/10))+X3(1);
desdq1=8*X1(9)*(t-10*floor(t/10))^7+7*X1(8)*(t-10*floor(t/10))^6+6*X1(7)↙
*(t-10*floor(t/10))^5+5*X1(6)*(t-10*floor(t/10))^4+4*X1(5)*(t-10*floor↙
(t/10))^3+3*X1(4)*(t-10*floor(t/10))^2+2*X1(3)*(t-10*floor(t/10))+X1(2);
desdq2=8*X2(9)*(t-10*floor(t/10))^7+7*X2(8)*(t-10*floor(t/10))^6+6*X2(7)↙
*(t-10*floor(t/10))^5+5*X2(6)*(t-10*floor(t/10))^4+4*X2(5)*(t-10*floor↙
(t/10))^3+3*X2(4)*(t-10*floor(t/10))^2+2*X2(3)*(t-10*floor(t/10))+X2(2);
desdq3=8*X3(9)*(t-10*floor(t/10))^7+7*X3(8)*(t-10*floor(t/10))^6+6*X3(7)↙
*(t-10*floor(t/10))^5+5*X3(6)*(t-10*floor(t/10))^4+4*X3(5)*(t-10*floor↙
(t/10))^3+3*X3(4)*(t-10*floor(t/10))^2+2*X3(3)*(t-10*floor(t/10))+X3(2);
desddq1=56*X1(9)*(t-10*floor(t/10))^6+42*X1(8)*(t-10*floor(t/10))↙
^5+30*X1(7)*(t-10*floor(t/10))^4+20*X1(6)*(t-10*floor(t/10))^3+12*X1(5)*↙
(t-10*floor(t/10))^2+6*X1(4)*(t-10*floor(t/10))+2*X1(3);
desddq2=56*X2(9)*(t-10*floor(t/10))^6+42*X2(8)*(t-10*floor(t/10))↙
^5+30*X2(7)*(t-10*floor(t/10))^4+20*X2(6)*(t-10*floor(t/10))^3+12*X2(5)*↙
(t-10*floor(t/10))^2+6*X2(4)*(t-10*floor(t/10))+2*X2(3);
desddq3=56*X3(9)*(t-10*floor(t/10))^6+42*X3(8)*(t-10*floor(t/10))↙
^5+30*X3(7)*(t-10*floor(t/10))^4+20*X3(6)*(t-10*floor(t/10))^3+12*X3(5)*↙
(t-10*floor(t/10))^2+6*X3(4)*(t-10*floor(t/10))+2*X3(3);
```

```
%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];

qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];

ddqd=[desddq1;desddq2;desddq3];

qe=q-qd;
dqe=dq-dqd;

%Composite Error
S=dqe+Lambda*qe;

dqr=[desdq1;desdq2;desdq3]-(Lambda)*qe;
ddqr=[desddq1;desddq2;desddq3]-(Lambda)*dqe;

dtheta1r=dqr(1);
dtheta2r=dqr(2);
dd3r=dqr(3);

ddtheta1r=ddqr(1);
ddtheta2r=ddqr(2);
ddd3r=ddqr(3);

%dynamics
Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin↙
(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(theta2)↙
^2*d3^2)/4+M4*sin(theta2)^2*d3^2,0,0;0,I_2XX+I_3XX+M2*cg_l2^2+(M3*d3^2)↙
/4+M4*d3^2,0;0,0,M3/4+M4];
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 +↙
I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)↙
*dtheta2,sin(2*theta2)*dtheta1*(I_2XX/2 + I_3XX/2 - I_2ZZ/2 - I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(sin(theta2)^2*d3*(M3 + 4*M4)↙
*dtheta1)/4;-sin(2*theta2)*dtheta1*(I_2XX/2+I_3XX/2-I_2ZZ/2-I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(d3*(M3+4*M4)*dd3)/4,(d3*↙
(M3+4*M4)*dtheta2)/4;-(sin(theta2)^2*d3*(M3+4*M4)*dtheta1)/4,-(d3*↙
(M3+4*M4)*dtheta2)/4,0];
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*↙
```

```
(M3+2*M4))/2];

%Regressors

Y0= [(I_2XX*ddtheta1r)/2 + (I_3XX*ddtheta1r)/2 + I_1ZZ*ddtheta1r +↙
(I_2ZZ*ddtheta1r)/2 + (I_3ZZ*ddtheta1r)/2 + (M3*ddtheta1r*d3^2)/8 -↙
(I_2XX*ddtheta1r*cos(2*theta2))/2 - (I_3XX*ddtheta1r*cos(2*theta2))/2 +↙
(I_2ZZ*ddtheta1r*cos(2*theta2))/2 + (I_3ZZ*ddtheta1r*cos(2*theta2))/2 +↙
(M2*cg_l2^2*ddtheta1r)/2 - (M2*cg_l2^2*ddtheta1r*cos(2*theta2))/2 +↙
(I_2XX*dtheta1r*sin(2*theta2)*dtheta2)/2 + (I_2XX*dtheta2r*sin(2*theta2)↙
*dtheta1)/2 + (I_3XX*dtheta1r*sin(2*theta2)*dtheta2)/2 +↙
(I_3XX*dtheta2r*sin(2*theta2)*dtheta1)/2 - (I_2ZZ*dtheta1r*sin(2*theta2)↙
*dtheta2)/2 - (I_2ZZ*dtheta2r*sin(2*theta2)*dtheta1)/2 -↙
(I_3ZZ*dtheta1r*sin(2*theta2)*dtheta2)/2 - (I_3ZZ*dtheta2r*sin(2*theta2)↙
*dtheta1)/2 + (M3*dtheta1r*d3*dd3)/8 + (M3*dd3r*d3*dtheta1)/8 -↙
(M3*ddtheta1r*cos(2*theta2)*d3^2)/8 - (M3*dtheta1r*cos(2*theta2)*d3*dd3)↙
/8 + (M2*cg_l2^2*dtheta1r*sin(2*theta2)*dtheta2)/2 +↙
(M2*cg_l2^2*dtheta2r*sin(2*theta2)*dtheta1)/2 - (M3*dd3r*cos(2*theta2)↙
*d3*dtheta1)/8 + (M3*dtheta1r*sin(2*theta2)*d3^2*dtheta2)/8 +↙
(M3*dtheta2r*sin(2*theta2)*d3^2*dtheta1)/8;
     I_2XX*ddtheta2r + I_3XX*ddtheta2r + (M3*ddtheta2r*d3^2)/4 +↙
M2*cg_l2^2*ddtheta2r + (M3*g*sin(theta2)*d3)/2 - (I_2XX*dtheta1r*sin↙
(2*theta2)*dtheta1)/2 - (I_3XX*dtheta1r*sin(2*theta2)*dtheta1)/2 +↙
(I_2ZZ*dtheta1r*sin(2*theta2)*dtheta1)/2 + (I_3ZZ*dtheta1r*sin(2*theta2)↙
*dtheta1)/2 + (M3*dtheta2r*d3*dd3)/4 + (M3*dd3r*d3*dtheta2)/4 +↙
M2*cg_l2*g*sin(theta2) - (M2*cg_l2^2*dtheta1r*sin(2*theta2)*dtheta1)/2 -↙
(M3*dtheta1r*sin(2*theta2)*d3^2*dtheta1)/8;
     - (M3*dtheta1r*d3*sin(theta2)^2*dtheta1)/4 -↙
(M3*dtheta2r*d3*dtheta2)/4 + (M3*ddd3r)/4 - (M3*g*cos(theta2))/2];


Y1=[sin(theta2)*d3*(dtheta1r*sin(theta2)*dd3 + dd3r*sin(theta2)*dtheta1↙
+ ddtheta1r*sin(theta2)*d3 + dtheta1r*cos(theta2)*d3*dtheta2 +↙
dtheta2r*cos(theta2)*d3*dtheta1);
    (d3*(2*g*sin(theta2) + 2*dtheta2r*dd3 + 2*dd3r*dtheta2 +↙
2*ddtheta2r*d3 - dtheta1r*sin(2*theta2)*d3*dtheta1))/2;
    - dtheta1r*d3*sin(theta2)^2*dtheta1 - dtheta2r*d3*dtheta2 + ddd3r -↙
g*cos(theta2)];


Tau=Y0+mh*Y1-Kd*S;
```

```
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3);-(1/gamma)
*transpose(S)*Y1];

end
```

```
function dydt = PGA(t,y)

M1=50;
M2=25;
M3=12.5;
M4=10;
cg_l2=0.5;
g=9.807;

X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625;↙
(12*pi)/78125; -(3*pi)/781250];
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625;↙
(6*pi)/78125; -(3*pi)/1562500];
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125);↙
-(3/390625)];

I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set states
theta1=y(1);
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);
%Desired Trajectories
% desq1=subs(destheta1,t);
% desq2=subs(destheta2,t);
% desq3=subs(desd3,t);
% desdq1=subs(desdtheta1,t);
% desdq2=subs(desdtheta2,t);
% desdq3=subs(desdd3,t);

desq1=X1(9)*(t-10*floor(t/10))^8+X1(8)*(t-10*floor(t/10))^7+X1(7)*(t-↙
10*floor(t/10))^6+X1(6)*(t-10*floor(t/10))^5+X1(5)*(t-10*floor(t/10))↙
^4+X1(4)*(t-10*floor(t/10))^3+X1(3)*(t-10*floor(t/10))^2+X1(2)*(t-↙
10*floor(t/10))+X1(1);
```

```
desq2=X2(9)*(t-10*floor(t/10))^8+X2(8)*(t-10*floor(t/10))^7+X2(7)*(t-↙
10*floor(t/10))^6+X2(6)*(t-10*floor(t/10))^5+X2(5)*(t-10*floor(t/10))↙
^4+X2(4)*(t-10*floor(t/10))^3+X2(3)*(t-10*floor(t/10))^2+X2(2)*(t-↙
10*floor(t/10))+X2(1);
desq3=X3(9)*(t-10*floor(t/10))^8+X3(8)*(t-10*floor(t/10))^7+X3(7)*(t-↙
10*floor(t/10))^6+X3(6)*(t-10*floor(t/10))^5+X3(5)*(t-10*floor(t/10))↙
^4+X3(4)*(t-10*floor(t/10))^3+X3(3)*(t-10*floor(t/10))^2+X3(2)*(t-↙
10*floor(t/10))+X3(1);
desdq1=8*X1(9)*(t-10*floor(t/10))^7+7*X1(8)*(t-10*floor(t/10))^6+6*X1(7)↙
*(t-10*floor(t/10))^5+5*X1(6)*(t-10*floor(t/10))^4+4*X1(5)*(t-10*floor↙
(t/10))^3+3*X1(4)*(t-10*floor(t/10))^2+2*X1(3)*(t-10*floor(t/10))+X1(2);
desdq2=8*X2(9)*(t-10*floor(t/10))^7+7*X2(8)*(t-10*floor(t/10))^6+6*X2(7)↙
*(t-10*floor(t/10))^5+5*X2(6)*(t-10*floor(t/10))^4+4*X2(5)*(t-10*floor↙
(t/10))^3+3*X2(4)*(t-10*floor(t/10))^2+2*X2(3)*(t-10*floor(t/10))+X2(2);
desdq3=8*X3(9)*(t-10*floor(t/10))^7+7*X3(8)*(t-10*floor(t/10))^6+6*X3(7)↙
*(t-10*floor(t/10))^5+5*X3(6)*(t-10*floor(t/10))^4+4*X3(5)*(t-10*floor↙
(t/10))^3+3*X3(4)*(t-10*floor(t/10))^2+2*X3(3)*(t-10*floor(t/10))+X3(2);

% desq1=3;
% desq2=3;
% desq3=3;
% desdq1=0;
% desdq2=0;
% desdq3=0;



%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];
qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];
qe=q-qd;
dqe=dq-dqd;
%Set gains
Kp=100;
Kd=50;

Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin↙
(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(theta2)↙
^2*d3^2)/4+M4*sin(theta2)^2*d3^2,0,0;0,I_2XX+I_3XX+M2*cg_l2^2+(M3*d3^2)↙
```

```
/4+M4*d3^2,0;0,0,M3/4+M4];
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 +↙
I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)↙
*dtheta2,sin(2*theta2)*dtheta1*(I_2XX/2 + I_3XX/2 - I_2ZZ/2 - I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(sin(theta2)^2*d3*(M3 + 4*M4)↙
*dtheta1)/4;-sin(2*theta2)*dtheta1*(I_2XX/2+I_3XX/2-I_2ZZ/2-I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(d3*(M3+4*M4)*dd3)/4,(d3*↙
(M3+4*M4)*dtheta2)/4;-(sin(theta2)^2*d3*(M3+4*M4)*dtheta1)/4,-(d3*↙
(M3+4*M4)*dtheta2)/4,0];
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*(M3↙
+ 2*M4))/2];


Tau=G-Kp*(qe)-Kd*(dqe);
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3)];

end
```

```
function dydt = ACOA(t,y)

M1=50;
M2=25;
M3=12.5;
M4=15;

% if t>20
%     M4=200;
% end

cg_l2=0.5;
g=9.807;


X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/1562500];
X1=[(0);(0);(0);(16704665593025/7421703487488)-(96*(pi))/125;(568*(pi))/625-(36750264304655/14843406974976);(11359172603257/9895604649984)
X3=[(1);0;0;-(192/125);(1136/625);-(2736/3125);(3536/15625);-(2664/78125);(1173/390625);-(56/390625);(28/9765625)];


I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set gains
%Kp=5;
Kd=50;
Lambda=25;
gamma=20;

%Set states
theta1=y(1);
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);
mh=y(7);

if theta1>0.927295218002
desq3=X3(11)*(t-10*floor(t/10))^10+X3(10)*(t-10*floor(t/10))^9+X3(9)*(t-10*floor(t/10))^8+X3(8)*(t-10*floor(t/10))^7+X3(7)*(t-10*floor(t/1
desdq3=10*X3(11)*(t-10*floor(t/10))^9+9*X3(10)*(t-10*floor(t/10))^8+8*X3(9)*(t-10*floor(t/10))^7+7*X3(8)*(t-10*floor(t/10))^6+6*X3(7)*(t-1
desddq3=90*X3(11)*(t-10*floor(t/10))^8+(72)*X3(10)*(t-10*floor(t/10))^7+56*X3(9)*(t-10*floor(t/10))^6+42*X3(8)*(t-10*floor(t/10))^5+30*X3(
else
    desq3=1;
    desdq3=0;
    desddq3=0;
end

%Desired Trajectories
% desq1=subs(destheta1,t);
% desq2=subs(destheta2,t);
% desq3=subs(desd3,t);
% desdq1=subs(desdtheta1,t);
% desdq2=subs(desdtheta2,t);
% desdq3=subs(desdd3,t);

desq2=X2(9)*(t-10*floor(t/10))^8+X2(8)*(t-10*floor(t/10))^7+X2(7)*(t-10*floor(t/10))^6+X2(6)*(t-10*floor(t/10))^5+X2(5)*(t-10*floor(t/10))
desdq2=8*X2(9)*(t-10*floor(t/10))^7+7*X2(8)*(t-10*floor(t/10))^6+6*X2(7)*(t-10*floor(t/10))^5+5*X2(6)*(t-10*floor(t/10))^4+4*X2(5)*(t-10*f
desddq2=56*X2(9)*(t-10*floor(t/10))^6+42*X2(8)*(t-10*floor(t/10))^5+30*X2(7)*(t-10*floor(t/10))^4+20*X2(6)*(t-10*floor(t/10))^3+12*X2(5)*(

desq1=X1(11)*(t-10*floor(t/10))^10+X1(10)*(t-10*floor(t/10))^9+X1(9)*(t-10*floor(t/10))^8+X1(8)*(t-10*floor(t/10))^7+X1(7)*(t-10*floor(t/1
desdq1=10*X1(11)*(t-10*floor(t/10))^9+9*X1(10)*(t-10*floor(t/10))^8+8*X1(9)*(t-10*floor(t/10))^7+7*X1(8)*(t-10*floor(t/10))^6+6*X1(7)*(t-1
desddq1=90*X1(11)*(t-10*floor(t/10))^8+(72)*X1(10)*(t-10*floor(t/10))^7+56*X1(9)*(t-10*floor(t/10))^6+42*X1(8)*(t-10*floor(t/10))^5+30*X1(


%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];

qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];

ddqd=[desddq1;desddq2;desddq3];

qe=q-qd;
```

```
dqe=dq-dqd;

%Composite Error
S=dqe+Lambda*qe;

dqr=[desdq1;desdq2;desdq3]-(Lambda)*qe;
ddqr=[desddq1;desddq2;desddq3]-(Lambda)*dqe;

dtheta1r=dqr(1);
dtheta2r=dqr(2);
dd3r=dqr(3);

ddtheta1r=ddqr(1);
ddtheta2r=ddqr(2);
ddd3r=ddqr(3);

%dynamics
Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(th
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 + I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)*dtheta2,
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*(M3+2*M4))/2];

%Regressors

Y0= [(I_2XX*ddtheta1r)/2 + (I_3XX*ddtheta1r)/2 + I_1ZZ*ddtheta1r + (I_2ZZ*ddtheta1r)/2 + (I_3ZZ*ddtheta1r)/2 + (M3*ddtheta1r*d3^2)/8 - (I_
     I_2XX*ddtheta2r + I_3XX*ddtheta2r + (M3*ddtheta2r*d3^2)/4 + M2*cg_l2^2*ddtheta2r + (M3*g*sin(theta2)*d3)/2 - (I_2XX*dtheta1r*sin(2*th
     - (M3*dtheta1r*d3*sin(theta2)^2*dtheta1)/4 - (M3*dtheta2r*d3*dtheta2)/4 + (M3*ddd3r)/4 - (M3*g*cos(theta2))/2];


Y1=[sin(theta2)*d3*(dtheta1r*sin(theta2)*dd3 + dd3r*sin(theta2)*dtheta1 + ddtheta1r*sin(theta2)*d3 + dtheta1r*cos(theta2)*d3*dtheta2 + dth
    (d3*(2*g*sin(theta2) + 2*dtheta2r*dd3 + 2*dd3r*dtheta2 + 2*ddtheta2r*d3 - dtheta1r*sin(2*theta2)*d3*dtheta1))/2;
    - dtheta1r*d3*sin(theta2)^2*dtheta1 - dtheta2r*d3*dtheta2 + ddd3r - g*cos(theta2)];


Tau=Y0+mh*Y1-Kd*S;
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3);-(1/gamma)*transpose(S)*Y1];

end
```

```
function dydt = ACOAAS(t,y)

M1=50;
M2=25;
M3=12.5;
M4=15;

% if t>20
%     M4=200;
% end

cg_l2=0.5;
g=9.807;


X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625;↙
(6*pi)/78125; -(3*pi)/1562500];
X1=[(0);(0);(0);(16704665593025/7421703487488)-(96*(pi))/125;(568*(pi))↙
/625-(36750264304655/14843406974976);(11359172603257/9895604649984)↙
-(1368*(pi))/3125;(1768*(pi))/15625-(28732024820003/98956046499840);↙
(668186623721/15461882265600)-(1332*(pi))/78125;(1173*(pi))/781250-↙
(4677306366047/1236950581248000);(668186623721/3710851743744000)-(28*↙
(pi))/390625;(14*(pi))/9765625-(668186623721/1855425871872000000)];
X3=[(1);0;0;-(192/125);(1136/625);-(2736/3125);(3536/15625);-(2664↙
/78125);(1173/390625);-(56/390625);(28/9765625)];


I_1ZZ=(0.5)*(M1)*(0.2)^2;
I_2ZZ=(0.5)*(M2)*(0.15)^2;
I_2XX=(1/12)*(M2)*(2*cg_l2)^2;
I_3ZZ=(0.5)*(12.5)*(.1)^2;
I_3XX=(1/12)*(12.5)*(1)^2;

%Set gains
%Kp=5;
Kd=50;
Lambda=25;
gamma=20;

%Set states
theta1=y(1);
```

```
theta2=y(2);
d3=y(3);
dtheta1=y(4);
dtheta2=y(5);
dd3=y(6);
mh=y(7);

if theta1>0.927295218002
desq3=X3(11)*(t-10*floor(t/10))^10+X3(10)*(t-10*floor(t/10))^9+X3(9)*(t-↙
10*floor(t/10))^8+X3(8)*(t-10*floor(t/10))^7+X3(7)*(t-10*floor(t/10))↙
^6+X3(6)*(t-10*floor(t/10))^5+X3(5)*(t-10*floor(t/10))^4+X3(4)*(t-↙
10*floor(t/10))^3+X3(3)*(t-10*floor(t/10))^2+X3(2)*(t-10*floor(t/10))+X3↙
(1);
desdq3=10*X3(11)*(t-10*floor(t/10))^9+9*X3(10)*(t-10*floor(t/10))^8+8*X3↙
(9)*(t-10*floor(t/10))^7+7*X3(8)*(t-10*floor(t/10))^6+6*X3(7)*(t-↙
10*floor(t/10))^5+5*X3(6)*(t-10*floor(t/10))^4+4*X3(5)*(t-10*floor↙
(t/10))^3+3*X3(4)*(t-10*floor(t/10))^2+2*X3(3)*(t-10*floor(t/10))+X3(2);
desddq3=90*X3(11)*(t-10*floor(t/10))^8+(72)*X3(10)*(t-10*floor(t/10))↙
^7+56*X3(9)*(t-10*floor(t/10))^6+42*X3(8)*(t-10*floor(t/10))^5+30*X3(7)*↙
(t-10*floor(t/10))^4+20*X3(6)*(t-10*floor(t/10))^3+12*X3(5)*(t-10*floor↙
(t/10))^2+6*X3(4)*(t-10*floor(t/10))+2*X3(3);
else
    desq3=1;
    desdq3=0;
    desddq3=0;
end

if (t-10*floor(t/10))<5
    M4=15;
else
    M4=12;
end




%Desired Trajectories
% desq1=subs(destheta1,t);
% desq2=subs(destheta2,t);
% desq3=subs(desd3,t);
% desdq1=subs(desdtheta1,t);
% desdq2=subs(desdtheta2,t);
```

```
% desdq3=subs(desdd3,t);

desq2=X2(9)*(t-10*floor(t/10))^8+X2(8)*(t-10*floor(t/10))^7+X2(7)*(t-↙
10*floor(t/10))^6+X2(6)*(t-10*floor(t/10))^5+X2(5)*(t-10*floor(t/10))↙
^4+X2(4)*(t-10*floor(t/10))^3+X2(3)*(t-10*floor(t/10))^2+X2(2)*(t-↙
10*floor(t/10))+X2(1);
desdq2=8*X2(9)*(t-10*floor(t/10))^7+7*X2(8)*(t-10*floor(t/10))^6+6*X2(7)↙
*(t-10*floor(t/10))^5+5*X2(6)*(t-10*floor(t/10))^4+4*X2(5)*(t-10*floor↙
(t/10))^3+3*X2(4)*(t-10*floor(t/10))^2+2*X2(3)*(t-10*floor(t/10))+X2(2);
desddq2=56*X2(9)*(t-10*floor(t/10))^6+42*X2(8)*(t-10*floor(t/10))↙
^5+30*X2(7)*(t-10*floor(t/10))^4+20*X2(6)*(t-10*floor(t/10))^3+12*X2(5)*↙
(t-10*floor(t/10))^2+6*X2(4)*(t-10*floor(t/10))+2*X2(3);

desq1=X1(11)*(t-10*floor(t/10))^10+X1(10)*(t-10*floor(t/10))^9+X1(9)*(t-↙
10*floor(t/10))^8+X1(8)*(t-10*floor(t/10))^7+X1(7)*(t-10*floor(t/10))↙
^6+X1(6)*(t-10*floor(t/10))^5+X1(5)*(t-10*floor(t/10))^4+X1(4)*(t-↙
10*floor(t/10))^3+X1(3)*(t-10*floor(t/10))^2+X1(2)*(t-10*floor(t/10))+X1↙
(1);
desdq1=10*X1(11)*(t-10*floor(t/10))^9+9*X1(10)*(t-10*floor(t/10))^8+8*X1↙
(9)*(t-10*floor(t/10))^7+7*X1(8)*(t-10*floor(t/10))^6+6*X1(7)*(t-↙
10*floor(t/10))^5+5*X1(6)*(t-10*floor(t/10))^4+4*X1(5)*(t-10*floor↙
(t/10))^3+3*X1(4)*(t-10*floor(t/10))^2+2*X1(3)*(t-10*floor(t/10))+X1(2);
desddq1=90*X1(11)*(t-10*floor(t/10))^8+(72)*X1(10)*(t-10*floor(t/10))↙
^7+56*X1(9)*(t-10*floor(t/10))^6+42*X1(8)*(t-10*floor(t/10))^5+30*X1(7)*↙
(t-10*floor(t/10))^4+20*X1(6)*(t-10*floor(t/10))^3+12*X1(5)*(t-10*floor↙
(t/10))^2+6*X1(4)*(t-10*floor(t/10))+2*X1(3);


%Create Error Functions etc.
q=[theta1;theta2;d3];
dq=[dtheta1;dtheta2;dd3];

qd=[desq1;desq2;desq3];
dqd=[desdq1;desdq2;desdq3];

ddqd=[desddq1;desddq2;desddq3];

qe=q-qd;
dqe=dq-dqd;

%Composite Error
```

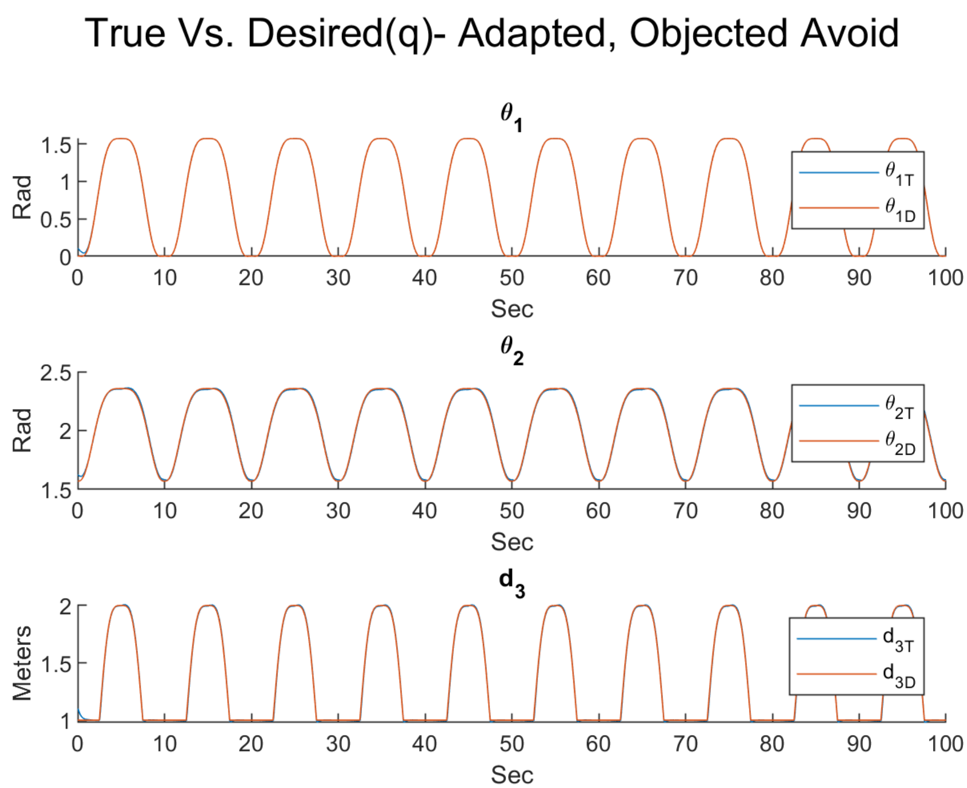
```
S=dqe+Lambda*qe;

dqr=[desdq1;desdq2;desdq3]-(Lambda)*qe;
ddqr=[desddq1;desddq2;desddq3]-(Lambda)*dqe;

dtheta1r=dqr(1);
dtheta2r=dqr(2);
dd3r=dqr(3);

ddtheta1r=ddqr(1);
ddtheta2r=ddqr(2);
ddd3r=ddqr(3);

%dynamics
Mre=[I_1ZZ+I_2ZZ+I_3ZZ+I_2XX*sin(theta2)^2+I_3XX*sin(theta2)^2-I_2ZZ*sin↙
(theta2)^2-I_3ZZ*sin(theta2)^2+M2*cg_l2^2*sin(theta2)^2+(M3*sin(theta2)↙
^2*d3^2)/4+M4*sin(theta2)^2*d3^2,0,0;0,I_2XX+I_3XX+M2*cg_l2^2+(M3*d3^2)↙
/4+M4*d3^2,0;0,0,M3/4+M4];
C=[(d3*(M3+4*M4)*sin(theta2)^2*dd3)/4 + sin(2*theta2)*(I_2XX/2 +↙
I_3XX/2-I_2ZZ/2 -I_3ZZ/2+(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2)↙
*dtheta2,sin(2*theta2)*dtheta1*(I_2XX/2 + I_3XX/2 - I_2ZZ/2 - I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(sin(theta2)^2*d3*(M3 + 4*M4)↙
*dtheta1)/4;-sin(2*theta2)*dtheta1*(I_2XX/2+I_3XX/2-I_2ZZ/2-I_3ZZ/2+↙
(M2*cg_l2^2)/2+(M3*d3^2)/8+(M4*d3^2)/2),(d3*(M3+4*M4)*dd3)/4,(d3*↙
(M3+4*M4)*dtheta2)/4;-(sin(theta2)^2*d3*(M3+4*M4)*dtheta1)/4,-(d3*↙
(M3+4*M4)*dtheta2)/4,0];
G=[0;(g*sin(theta2)*(2*M2*cg_l2 + M3*d3+2*M4*d3))/2; -(g*cos(theta2)*↙
(M3+2*M4))/2];

%Regressors

Y0= [(I_2XX*ddtheta1r)/2 + (I_3XX*ddtheta1r)/2 + I_1ZZ*ddtheta1r +↙
(I_2ZZ*ddtheta1r)/2 + (I_3ZZ*ddtheta1r)/2 + (M3*ddtheta1r*d3^2)/8 -↙
(I_2XX*ddtheta1r*cos(2*theta2))/2 - (I_3XX*ddtheta1r*cos(2*theta2))/2 +↙
(I_2ZZ*ddtheta1r*cos(2*theta2))/2 + (I_3ZZ*ddtheta1r*cos(2*theta2))/2 +↙
(M2*cg_l2^2*ddtheta1r)/2 - (M2*cg_l2^2*ddtheta1r*cos(2*theta2))/2 +↙
(I_2XX*dtheta1r*sin(2*theta2)*dtheta2)/2 + (I_2XX*dtheta2r*sin(2*theta2)↙
*dtheta1)/2 + (I_3XX*dtheta1r*sin(2*theta2)*dtheta2)/2 +↙
(I_3XX*dtheta2r*sin(2*theta2)*dtheta1)/2 - (I_2ZZ*dtheta1r*sin(2*theta2)↙
*dtheta2)/2 - (I_2ZZ*dtheta2r*sin(2*theta2)*dtheta1)/2 -↙
(I_3ZZ*dtheta1r*sin(2*theta2)*dtheta2)/2 - (I_3ZZ*dtheta2r*sin(2*theta2)↙
```

```
*dtheta1)/2 + (M3*dtheta1r*d3*dd3)/8 + (M3*dd3r*d3*dtheta1)/8 -↙
(M3*ddtheta1r*cos(2*theta2)*d3^2)/8 - (M3*dtheta1r*cos(2*theta2)*d3*dd3)↙
/8 + (M2*cg_l2^2*dtheta1r*sin(2*theta2)*dtheta2)/2 +↙
(M2*cg_l2^2*dtheta2r*sin(2*theta2)*dtheta1)/2 - (M3*dd3r*cos(2*theta2)↙
*d3*dtheta1)/8 + (M3*dtheta1r*sin(2*theta2)*d3^2*dtheta2)/8 +↙
(M3*dtheta2r*sin(2*theta2)*d3^2*dtheta1)/8;
     I_2XX*ddtheta2r + I_3XX*ddtheta2r + (M3*ddtheta2r*d3^2)/4 +↙
M2*cg_l2^2*ddtheta2r + (M3*g*sin(theta2)*d3)/2 - (I_2XX*dtheta1r*sin↙
(2*theta2)*dtheta1)/2 - (I_3XX*dtheta1r*sin(2*theta2)*dtheta1)/2 +↙
(I_2ZZ*dtheta1r*sin(2*theta2)*dtheta1)/2 + (I_3ZZ*dtheta1r*sin(2*theta2)↙
*dtheta1)/2 + (M3*dtheta2r*d3*dd3)/4 + (M3*dd3r*d3*dtheta2)/4 +↙
M2*cg_l2*g*sin(theta2) - (M2*cg_l2^2*dtheta1r*sin(2*theta2)*dtheta1)/2 -↙
(M3*dtheta1r*sin(2*theta2)*d3^2*dtheta1)/8;
     - (M3*dtheta1r*d3*sin(theta2)^2*dtheta1)/4 -↙
(M3*dtheta2r*d3*dtheta2)/4 + (M3*ddd3r)/4 - (M3*g*cos(theta2))/2];


Y1=[sin(theta2)*d3*(dtheta1r*sin(theta2)*dd3 + dd3r*sin(theta2)*dtheta1↙
+ ddtheta1r*sin(theta2)*d3 + dtheta1r*cos(theta2)*d3*dtheta2 +↙
dtheta2r*cos(theta2)*d3*dtheta1);
    (d3*(2*g*sin(theta2) + 2*dtheta2r*dd3 + 2*dd3r*dtheta2 +↙
2*ddtheta2r*d3 - dtheta1r*sin(2*theta2)*d3*dtheta1))/2;
    - dtheta1r*d3*sin(theta2)^2*dtheta1 - dtheta2r*d3*dtheta2 + ddd3r -↙
g*cos(theta2)];


Tau=Y0+mh*Y1-Kd*S;
%Returns
accel=Mre\(Tau-(C*dq)-G);

%dydt=[q;accel];
dydt=[dtheta1;dtheta2;dd3;accel(1);accel(2);accel(3);-(1/gamma)↙
*transpose(S)*Y1];

end
```

Plot Object Avoidence

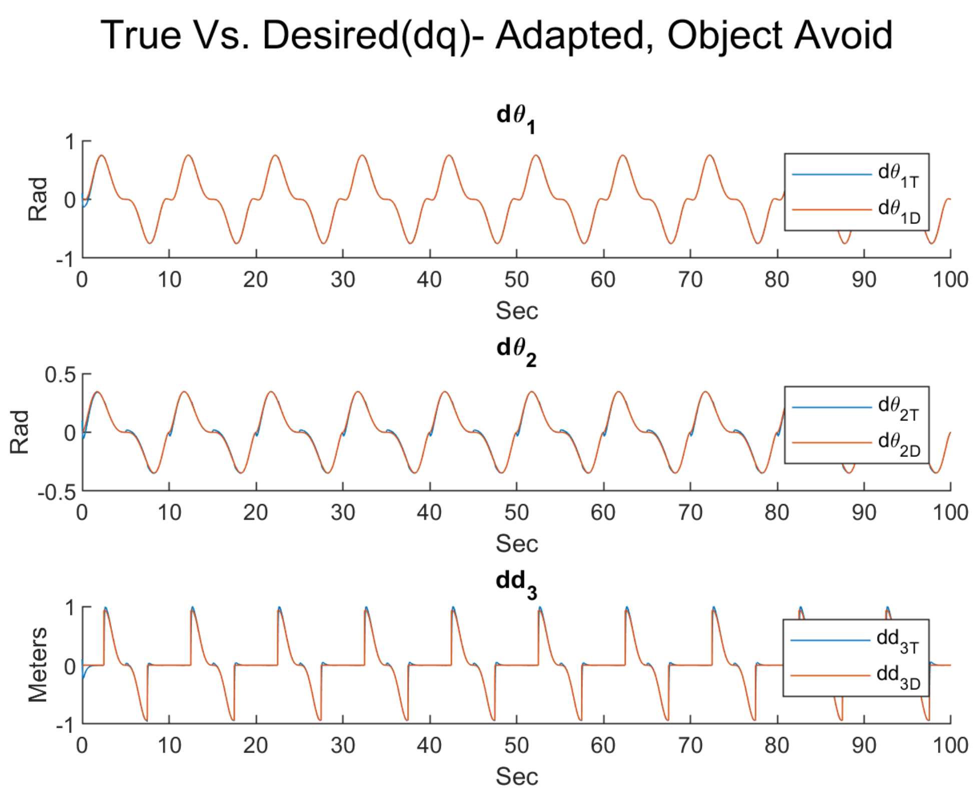
```
tspan=[0 100];
intal=[0.1; (pi+0.1)/2; 1.1; 0.1; 0.1; 0.1; 15];
[t,y] = ode45(@ACOAAS,tspan,intal);

X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X1=[(0);(0);(0);(16704665593025/7421703487488)-(96*(pi))/125;(568*(pi))/625-(36750264304655/148
X3=[(1);0;0;-(192/125);(1136/625);-(2736/3125);(3536/15625);-(2664/78125);(1173/390625);-(56/39

des=zeros(length(t),6);
x=t;

for k=1:length(t)
    des((k),1)=X1(11)*(x(k)-10*floor(x(k)/10))^10+X1(10)*(x(k)-10*floor(x(k)/10))^9+X1(9)*(x(k)
    des((k),4)=10*X1(11)*(x(k)-10*floor(x(k)/10))^9+9*X1(10)*(x(k)-10*floor(x(k)/10))^8+8*X1(9)
    des((k),2)=X2(9)*(x(k)-10*floor(x(k)/10))^8+X2(8)*(x(k)-10*floor(x(k)/10))^7+X2(7)*(x(k)-10
    des((k),5)=8*X2(9)*(x(k)-10*floor(x(k)/10))^7+7*X2(8)*(x(k)-10*floor(x(k)/10))^6+6*X2(7)*(x
    if des(k,1)>0.927295218002 && des(k,1)>0.927295218002
        des((k),3)=X3(11)*(x(k)-10*floor(x(k)/10))^10+X3(10)*(x(k)-10*floor(x(k)/10))^9+X3(9)*(
        des((k),6)=10*X3(11)*(x(k)-10*floor(x(k)/10))^9+9*X3(10)*(x(k)-10*floor(x(k)/10))^8+8*X

    else
        des((k),3)=1;
        des((k),6)=0;
    end
end



figure(14)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,des(:,1))
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,des(:,2))
legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')
```

```
subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,des(:,3))
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired(q)- Adapted, Objected Avoid')
```




```
figure(15)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,des(:,4))
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
```

```
plot(t,des(:,5))
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,des(:,6))
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired(dq)- Adapted, Object Avoid')
```



Final Plotting

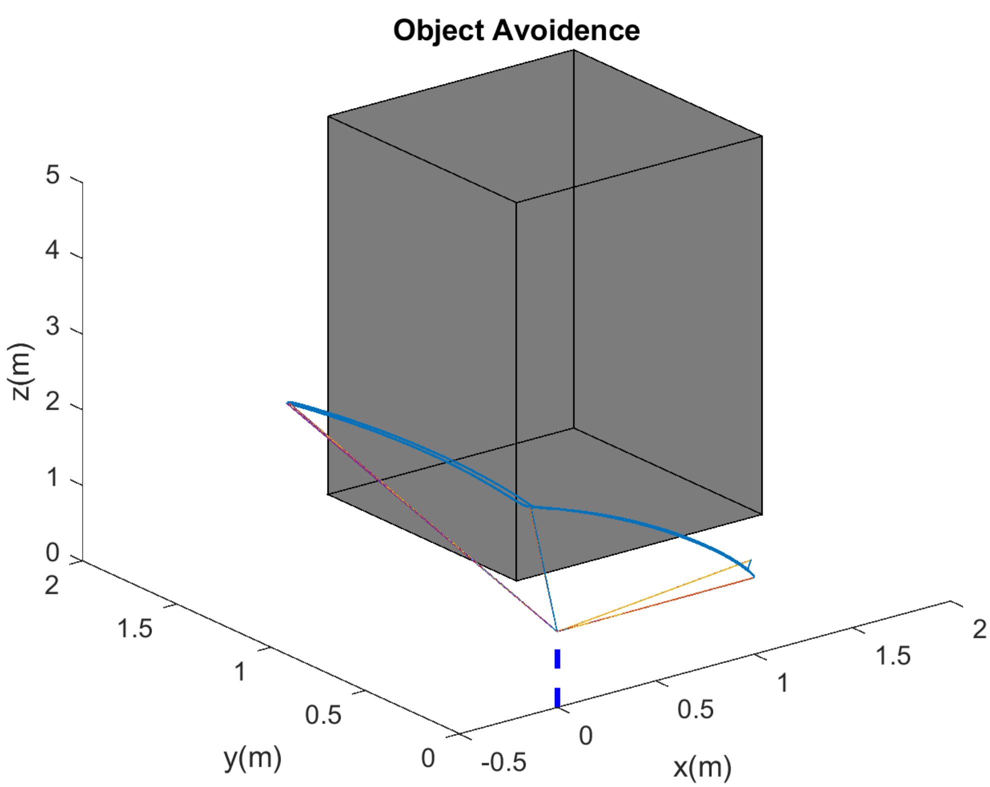
```
%End effector

d1=1;
x3e=(y(:,3).*cos(y(:,1)).*sin(y(:,2)));
y3e=(y(:,3).*sin(y(:,1)).*sin(y(:,2)));
z3e=(d1-y(:,3).*cos(y(:,2)));

%Origin and link 1
```

```
x0=[0 0];
y0=[0 0];
z0=[0 1];

%link 2
i=0;
p=[0,0,0];
for x=1:length(t)
    if rem(t(x),5)<0.0051 | rem(t(x),2.5)<0.000325 | rem(t(x),7.5)<0.000325
        p=[p;y(x,1:3)];
        i=i+1;
    end
end
i
```

```
i = 155
```

```
p=p(2:length(p),:);

x2e=[zeros(length(p),1), (p(:,3).*cos(p(:,1)).*sin(p(:,2)))];
y2e=[zeros(length(p),1), (p(:,3).*sin(p(:,1)).*sin(p(:,2)))];
z2e=[ones(length(p),1), (d1-p(:,3).*cos(p(:,2)))];




x1 = [2 2 0.75 0.75];
y1 = [1 2 2 1];
z1 = [0 0 0 0];

x2 = [0.75 0.75 0.75 0.75];
y2 = [1 2 2 1];
z2 = [0 0 5 5];

x3 = [0.75 2 2 0.75];
y3 = [1 1 1 1];
z3 = [0 0 5 5];

x4 = [2 2 0.75 0.75];
y4 = [1 2 2 1];
z4 = [5 5 5 5];

x5 = [0.75 2 2 0.75];
y5 = [2 2 2 2];
z5 = [0 0 5 5];

x6 = [2 2 2 2];
y6 = [1 2 2 1];
z6 = [0 0 5 5];

clear alpha

figure(16)
clf
```

```
hold on

patch(x1,y1,z1,'black')
%alpha(0.3)
patch(x2,y2,z2,'black')
%alpha(0.3)
patch(x3,y3,z3,'black')
%alpha(0.3)
patch(x4,y4,z4,'black')
%alpha(0.3)
patch(x5,y5,z5,'black')
%alpha(0.3)
patch(x6,y6,z6,'black')
alpha(0.3)

title('Object Avoidence')
plot3(x3e,y3e,z3e)
plot3(x0,y0,z0,'--','LineWidth',2,'Color','Blue')
plot3(transpose(x2e),transpose(y2e),transpose(z2e))
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
view(3)
```



Project 1

Forward Kinematics

First We require the DH table for the robot.

```
syms theta alpha d a theta1 theta2 d1 d3
dh=[theta1,d1,0,pi/2;theta2,0,0,pi/2;0,d3,0,0]
```

dh =

$$\begin{pmatrix} \theta_1 & d_1 & 0 & \frac{\pi}{2} \\ \theta_2 & 0 & 0 & \frac{\pi}{2} \\ 0 & d_3 & 0 & 0 \end{pmatrix}$$

Given this DH table we can evaluate the forward kinematics using the A matrix.

```
A=[cos(theta),-sin(theta)*cos(alpha),sin(theta)*sin(alpha),a*cos(theta);
    sin(theta),cos(theta)*cos(alpha),-cos(theta)*sin(alpha),a*sin(theta);
    0,sin(alpha),cos(alpha),d;
    0, 0, 0, 1]
```

A =

$$\begin{pmatrix} \cos(\theta) & -\cos(\alpha)\sin(\theta) & \sin(\alpha)\sin(\theta) & a\cos(\theta) \\ \sin(\theta) & \cos(\alpha)\cos(\theta) & -\sin(\alpha)\cos(\theta) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Tfsym=Tdes(dh)
```

Tfsym =

$$\begin{pmatrix} \cos(\theta_1)\cos(\theta_2) & \sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) & d_3\cos(\theta_1)\sin(\theta_2) \\ \cos(\theta_2)\sin(\theta_1) & -\cos(\theta_1) & \sin(\theta_1)\sin(\theta_2) & d_3\sin(\theta_1)\sin(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & d_1 - d_3\cos(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
TFN=double(subs(Tfsym, {theta1, theta2, d1, d3}, {pi/3, 5*pi/4,1,7}))
```

```
TFN = 4×4
   -0.3536    0.8660   -0.3536   -2.4749
   -0.6124   -0.5000   -0.6124   -4.2866
   -0.7071         0    0.7071    5.9497
         0         0         0    1.0000
```

This can be evaluated for any input parameters as shown.

Inverse Kinematics

The inverse kinematics tend to be more complecated but are a more relevant problem.

Given a final position, we must be able to back compute the joint variables.

For this 3 link robot this problem is greately simplified simbly by aligning the vector d3 from the point d1 to the final point.

as follows.

Given a final point:

```
EE=TFN(:,4)
```

```
EE = 4×1
   -2.4749
   -4.2866
    5.9497
    1.0000
```

We can first compute d3, as a magnitude between the diffrence of two points.

```
PB1=double(subs(EE-[0;0;d1;0], {d1}, {1}))
```

```
PB1 = 4×1
   -2.4749
   -4.2866
    4.9497
    1.0000
```

```
d3=sqrt(PB1(1)^2+PB1(2)^2+PB1(3)^2)
```

```
d3 = 7
```

This output comes out correct so we can now solve for theta 1 and 2 with simple triangles

```
theta2=asin(PB1(3)/d3)+(pi)
```

```
theta2 = 3.9270
```

```
theta1=atan(PB1(2)/PB1(1))
```

```
theta1 = 1.0472
```

EOM

```
Tdes(dh(1,:))
```

ans =

$$\begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
%Tdes(dh(1:3,:))
```

```
syms I_1ZZ I_2ZZ I_2YY I_2XX M2 M3 I_3ZZ I_3YY I_3XX theta1(t) theta2(t) d3(t) cg_l2 g M4
```

Link 1

```
R1=[cos(theta1),-sin(theta1),0;sin(theta1),cos(theta1),0;0,0,1]
```

R1(t) =

$$\begin{pmatrix} \cos(\theta_1(t)) & -\sin(\theta_1(t)) & 0 \\ \sin(\theta_1(t)) & \cos(\theta_1(t)) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
Jw1=[0,0,0;0,0,0;1,0,0]
```

```
Jw1 = 3×3
     0     0     0
     0     0     0
     1     0     0
```

```
syms I_1YY I_1XX
I1=[I_1XX,0,0;0,I_1YY,0;0,0,I_1ZZ]
```

I1 =

$$\begin{pmatrix} I_{1XX} & 0 & 0 \\ 0 & I_{1YY} & 0 \\ 0 & 0 & I_{1ZZ} \end{pmatrix}$$

```
transpose(Jw1)*R1*I1*transpose(R1)*Jw1
```

ans(t) =

$$\begin{pmatrix} I_{1ZZ} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
ke1=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*transpose(Jw1)*(R1*I1*tr
```

ke1(t) =

$$\frac{I_{1ZZ}\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{2}$$

Link 2

```
I2=[I_2XX,0,0;0,I_2XX,0;0,0,I_2ZZ]
```

I2 =

$$\begin{pmatrix} I_{2XX} & 0 & 0 \\ 0 & I_{2XX} & 0 \\ 0 & 0 & I_{2ZZ} \end{pmatrix}$$

```
RO2=[cos(theta1)*cos(theta2),sin(theta1),cos(theta1)*sin(theta2);sin(theta1)*cos(theta2),-cos(t
```

```
RO2(t) =
```

$$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$$

```
Jv2cm=[-cg_l2*sin(theta1)*sin(theta2),cg_l2*cos(theta1)*cos(theta2),0;
        cg_l2*cos(theta1)*sin(theta2),cg_l2*sin(theta1)*cos(theta2),0;
        0,cg_l2*sin(theta2),0]
```

```
Jv2cm(t) =
```

$$\begin{pmatrix} -\mathrm{cg}_{l2}\sin(\theta_1(t))\sin(\theta_2(t)) & \mathrm{cg}_{l2}\cos(\theta_1(t))\cos(\theta_2(t)) & 0 \\ \mathrm{cg}_{l2}\cos(\theta_1(t))\sin(\theta_2(t)) & \mathrm{cg}_{l2}\cos(\theta_2(t))\sin(\theta_1(t)) & 0 \\ 0 & \mathrm{cg}_{l2}\sin(\theta_2(t)) & 0 \end{pmatrix}$$

```
Jw2cm=[0,sin(theta1),0;
       0,-cos(theta1),0;
       1,0,0]
```

```
Jw2cm(t) =
```

$$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
ke2T=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*M2*transpose(Jv2cm)*Jv2
```

```
ke2T(t) =
```

$$\frac{M_2\,\mathrm{cg}_{l2}^2\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}+\frac{M_2\,\mathrm{cg}_{l2}^2\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2}{2}-\frac{M_2\,\mathrm{cg}_{l2}^2\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}$$

```
ke2R=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*transpose(Jw2cm)*(RO2*I
```

```
ke2R(t) =
```

$$\frac{I_{2XX}\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}+\frac{I_{2XX}\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2}{2}+\frac{I_{2ZZ}\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}-\frac{I_{2XX}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}+\frac{I_{2ZZ}\cos(2\,\theta_2(t))}{4}$$

```
pe2=M2*g*(-cos(theta2)*cg_l2)
```

pe2(t) = $-M_2\,\mathrm{cg}_{l2}\,g\cos(\theta_2(t))$

Link 3

```
I3=[I_3XX,0,0;0,I_3XX,0;0,0,I_3ZZ]
```

```
I3 =
```

$$\begin{pmatrix} I_{3XX} & 0 & 0 \\ 0 & I_{3XX} & 0 \\ 0 & 0 & I_{3ZZ} \end{pmatrix}$$

```
RO3=[cos(theta1)*cos(theta2),sin(theta1),cos(theta1)*sin(theta2);sin(theta1)*cos(theta2),-cos(t
```

RO3(t) =

$$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$$

```
Jv3cm=[-(d3/2)*sin(theta1)*sin(theta2),(d3/2)*cos(theta1)*cos(theta2),(cos(theta1)*sin(theta2)/
        (d3/2)*cos(theta1)*sin(theta2),(d3/2)*sin(theta1)*cos(theta2),(sin(theta1)*sin(theta2)/
        0,(d3/2)*sin(theta2),-cos(theta2)/2]
```

Jv3cm(t) =

$$\begin{pmatrix} -\frac{\sin(\theta_1(t))\sin(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_1(t))\cos(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_1(t))\sin(\theta_2(t))}{2} \\ \frac{\cos(\theta_1(t))\sin(\theta_2(t))\,d_3(t)}{2} & \frac{\cos(\theta_2(t))\sin(\theta_1(t))\,d_3(t)}{2} & \frac{\sin(\theta_1(t))\sin(\theta_2(t))}{2} \\ 0 & \frac{\sin(\theta_2(t))\,d_3(t)}{2} & -\frac{\cos(\theta_2(t))}{2} \end{pmatrix}$$

```
Jw3cm=[0,sin(theta1),0;
       0,-cos(theta1),0;
       1,0,0]
```

Jw3cm(t) =

$$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
ke3T=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*M3*transpose(Jv3cm)*Jv3
```

ke3T(t) =

$$\frac{M_3\sin(\theta_2(t))^2\,d_3(t)^2\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{8}+\frac{M_3\,d_3(t)^2\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2}{8}+\frac{M_3\left(\frac{\partial}{\partial t}\,d_3(t)\right)^2}{8}$$

```
ke3R=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*transpose(Jw3cm)*(RO3*I
```

ke3R(t) =

$$\frac{I_{3XX}\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}+\frac{I_{3XX}\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2}{2}+\frac{I_{3ZZ}\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}-\frac{I_{3XX}\cos(2\,\theta_2(t))\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{4}+\frac{I_{3ZZ}\cos(2\,\theta_2(t))}{4}$$

```
pe3=M3*g*(-cos(theta2)*d3/2)
```

pe3(t) =

$$-\frac{M_3\,g\cos(\theta_2(t))\,d_3(t)}{2}$$

Unkown End effector mass

```
RO3=[cos(theta1)*cos(theta2),sin(theta1),cos(theta1)*sin(theta2);sin(theta1)*cos(theta2),-cos(t
```

RO3(t) =

$$\begin{pmatrix} \cos(\theta_1(t))\cos(\theta_2(t)) & \sin(\theta_1(t)) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_2(t))\sin(\theta_1(t)) & -\cos(\theta_1(t)) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ \sin(\theta_2(t)) & 0 & -\cos(\theta_2(t)) \end{pmatrix}$$

```
Jv4cm=[-(d3)*sin(theta1)*sin(theta2),(d3)*cos(theta1)*cos(theta2),(cos(theta1)*sin(theta2));
        (d3)*cos(theta1)*sin(theta2),(d3)*sin(theta1)*cos(theta2),(sin(theta1)*sin(theta2));
        0,(d3)*sin(theta2),-cos(theta2)]
```

Jv4cm(t) =

$$\begin{pmatrix} -\sin(\theta_1(t))\sin(\theta_2(t))\,d_3(t) & \cos(\theta_1(t))\cos(\theta_2(t))\,d_3(t) & \cos(\theta_1(t))\sin(\theta_2(t)) \\ \cos(\theta_1(t))\sin(\theta_2(t))\,d_3(t) & \cos(\theta_2(t))\sin(\theta_1(t))\,d_3(t) & \sin(\theta_1(t))\sin(\theta_2(t)) \\ 0 & \sin(\theta_2(t))\,d_3(t) & -\cos(\theta_2(t)) \end{pmatrix}$$

```
Jw4cm=[0,sin(theta1),0;
       0,-cos(theta1),0;
       1,0,0]
```

Jw4cm(t) =

$$\begin{pmatrix} 0 & \sin(\theta_1(t)) & 0 \\ 0 & -\cos(\theta_1(t)) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

```
ke4T=simplify(0.5*transpose([diff(theta1,t);diff(theta2,t);diff(d3,t)])*M4*transpose(Jv4cm)*Jv4
```

ke4T(t) =

$$\frac{M_4\sin(\theta_2(t))^2\,d_3(t)^2\,\left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2}{2}+\frac{M_4\,d_3(t)^2\,\left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2}{2}+\frac{M_4\,\left(\frac{\partial}{\partial t}\,d_3(t)\right)^2}{2}$$

```
pe4=M4*g*(-cos(theta2)*d3)
```

pe4(t) = $-M_4\,g\cos(\theta_2(t))\,d_3(t)$

Lagrange

```
Lag=simplify((ke1+ke2T+ke2R+ke3T+ke3R+ke4T)-(pe2+pe3+pe4))
```

Lag(t) =

$$\frac{M_3\,\sigma_4}{8}+\frac{M_4\,\sigma_4}{2}+\frac{I_{2XX}\,\sigma_1}{2}+\frac{I_{2XX}\,\sigma_3}{2}+\frac{I_{3XX}\,\sigma_1}{2}+\frac{I_{3XX}\,\sigma_3}{2}+\frac{I_{1ZZ}\,\sigma_1}{2}+\frac{M_3\,d_3(t)^2\,\sigma_1}{8}+\frac{M_3\,d_3(t)^2\,\sigma_3}{8}+\frac{M_4\,d_3(t)^2\,\sigma_1}{2}+$$

where

$$\sigma_1 = \left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2$$

$$\sigma_2 = \cos(\theta_2(t))^2$$

$$\sigma_3 = \left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2$$

$$\sigma_4 = \left(\frac{\partial}{\partial t}\,d_3(t)\right)^2$$

```
eqm=simplify([diff(diff(Lag,diff(theta1,t)),t)-diff(Lag,theta1);
     diff(diff(Lag,diff(theta2,t)),t)-diff(Lag,theta2);
     diff(diff(Lag,diff(d3,t)),t)-diff(Lag,d3)])
```

```
eqm(t) =
```

$$\left(\frac{I_{2XX}\,\sigma_3}{2}+\frac{I_{3XX}\,\sigma_3}{2}+I_{1ZZ}\,\sigma_3+\frac{I_{2ZZ}\,\sigma_3}{2}+\frac{I_{3ZZ}\,\sigma_3}{2}+\frac{M_3\,d_3(t)^2\,\sigma_3}{8}+\frac{M_4\,d_3(t)^2\,\sigma_3}{2}-\frac{I_{2XX}\,\sigma_4\,\sigma_3}{2}-\frac{I_{3XX}\,\sigma_4\,\sigma_3}{2}+\frac{I_{2ZZ}\,c}{2}\right.$$

where

$$\sigma_1 = \sin(2\,\theta_2(t))$$

$$\sigma_2 = \left(\frac{\partial}{\partial t}\,\theta_1(t)\right)^2$$

$$\sigma_3 = \frac{\partial^2}{\partial t^2}\,\theta_1(t)$$

$$\sigma_4 = \cos(2\,\theta_2(t))$$

$$\sigma_5 = \frac{\partial^2}{\partial t^2}\,\theta_2(t)$$

$$\sigma_6 = \frac{\partial^2}{\partial t^2}\,d_3(t)$$

$$\sigma_7 = \left(\frac{\partial}{\partial t}\,\theta_2(t)\right)^2$$

$$\sigma_8 = \cos(\theta_2(t))^2$$

Try new strategy

```
Mre=simplify(M4*transpose(Jv4cm)*Jv4cm+transpose(Jw3cm)*(RO3*I3*transpose(RO3))*Jw3cm+M3*transp
```

```
Mre(t) =
```

$$\left(\begin{array}{ccc} I_{1ZZ} + I_{2ZZ} + I_{3ZZ} + I_{2XX}\,\sigma_1 + I_{3XX}\,\sigma_1 - I_{2ZZ}\,\sigma_1 - I_{3ZZ}\,\sigma_1 + M_2\,\mathrm{cg}_{l2}{}^2\,\sigma_1 + \frac{M_3\,\sigma_1\,d_3(t)^2}{4} + M_4\,\sigma_1\,d_3(t)^2 & & \\ 0 & & I_{2XX} + \\ 0 & & \end{array}\right.$$

where

$$\sigma_1 = \sin(\theta_2(t))^2$$

Calculate C

```
C=sym('A',[3 3]);
dtheta1=diff(theta1,t);
dtheta2=diff(theta2,t);
dd3=diff(d3,t);
q=[theta1;theta2;d3]
```

q(t) =

$$\begin{pmatrix} \theta_1(t) \\ \theta_2(t) \\ d_3(t) \end{pmatrix}$$

```
qd=[dtheta1;dtheta2;dd3]
```

qd(t) =

$$\begin{pmatrix} \frac{\partial}{\partial t}\,\theta_1(t) \\ \frac{\partial}{\partial t}\,\theta_2(t) \\ \frac{\partial}{\partial t}\,d_3(t) \end{pmatrix}$$

```
qddot=[diff(dtheta1,t);diff(dtheta2,t);diff(dd3,t)]
```

qddot(t) =

$$\begin{pmatrix} \frac{\partial^2}{\partial t^2}\,\theta_1(t) \\ \frac{\partial^2}{\partial t^2}\,\theta_2(t) \\ \frac{\partial^2}{\partial t^2}\,d_3(t) \end{pmatrix}$$

```
v=children(Mre);
vq=children(q);
vqd=children(qd);
vqd{1}{1}
```

ans = $\theta_1(t)$

```
for k=1:3
    for j=1:3
        l=0;
        for i=1:3
%              [vqd{i}{1},vqd{j}{1},vqd{k}{1},diff(vqd{i}{1},t)]
            l=(0.5)*(diff(sum([v{k,j}{:}]),vqd{i}{1})+diff(sum([v{k,i}{:}]),vqd{j}{1})-diff(sur
        end

        C(k,j)=simplify(l);
    end

end
C
```

C =

$$\begin{pmatrix} \frac{d_3(t)\ (M_3+4\,M_4)\sin(\theta_2(t))^2\,\frac{\partial}{\partial t}\,d_3(t)}{4}+\sin(2\,\theta_2(t))\,\sigma_4\,\frac{\partial}{\partial t}\,\theta_2(t) & \sigma_3 & \sigma_1 \\ -\sigma_3 & \frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\,d_3(t)}{4} & \sigma_2 \\ -\sigma_1 & -\sigma_2 & 0 \end{pmatrix}$$

where

$$\sigma_1=\frac{\sin(\theta_2(t))^2\,d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\,\theta_1(t)}{4}$$

$$\sigma_2=\frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\,\theta_2(t)}{4}$$

$$\sigma_3=\sin(2\,\theta_2(t))\,\frac{\partial}{\partial t}\,\theta_1(t)\,\sigma_4$$

$$\sigma_4=\frac{I_{2XX}}{2}+\frac{I_{3XX}}{2}-\frac{I_{2ZZ}}{2}-\frac{I_{3ZZ}}{2}+\frac{M_2\,\mathrm{cg}_{l2}^2}{2}+\frac{M_3\,d_3(t)^2}{8}+\frac{M_4\,d_3(t)^2}{2}$$

```
simplify(diff(Mre,t)-2*C)
```

ans(t) =

$$\begin{pmatrix} 0 & -\sigma_1 & -\sigma_2 \\ \sigma_1 & 0 & -\frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_2(t)}{2} \\ \sigma_2 & \frac{d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_2(t)}{2} & 0 \end{pmatrix}$$

where

$$\sigma_1 = 2\sin(2\,\theta_2(t))\,\frac{\partial}{\partial t}\ \theta_1(t)\ \left(\frac{I_{2XX}}{2}+\frac{I_{3XX}}{2}-\frac{I_{2ZZ}}{2}-\frac{I_{3ZZ}}{2}+\frac{M_2\,\mathrm{cg}_{12}{}^2}{2}+\frac{M_3\,d_3(t)^2}{8}+\frac{M_4\,d_3(t)^2}{2}\right)$$

$$\sigma_2 = \frac{\sin(\theta_2(t))^2\,d_3(t)\ (M_3+4\,M_4)\,\frac{\partial}{\partial t}\ \theta_1(t)}{2}$$

```
G=simplify(eqm-(Mre*qddot)-(C*qd))
```

G(t) =

$$\begin{pmatrix} 0 \\ \frac{g\sin(\theta_2(t))\ (2\,M_2\,\mathrm{cg}_{12}+M_3\,d_3(t)+2\,M_4\,d_3(t))}{2} \\ -\frac{g\cos(\theta_2(t))\ (M_3+2\,M_4)}{2} \end{pmatrix}$$

Develop trajectory

```
syms t a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 t_0 t_1 t_2
q1=a_0+(a_1*t)+(a_2*t^2)+(a_3*t^3)+(a_4*t^4)+(a_5*t^5)+(a_6*t^6)+(a_7*t^7)+(a_8*t^8)
```

q1 = $a_8\,t^8+a_7\,t^7+a_6\,t^6+a_5\,t^5+a_4\,t^4+a_3\,t^3+a_2\,t^2+a_1\,t+a_0$

```
dq1=diff(q1,t)
```

dq1 = $8\,a_8\,t^7+7\,a_7\,t^6+6\,a_6\,t^5+5\,a_5\,t^4+4\,a_4\,t^3+3\,a_3\,t^2+2\,a_2\,t+a_1$

```
ddq1=diff(dq1,t)
```

ddq1 = $56\,a_8\,t^6+42\,a_7\,t^5+30\,a_6\,t^4+20\,a_5\,t^3+12\,a_4\,t^2+6\,a_3\,t+2\,a_2$

```
D=[subs(q1,t_0);subs(dq1,t_0);subs(ddq1,t_0);subs(q1,t_1);subs(dq1,t_1);subs(ddq1,t_1);subs(q1,
```

D =

$$\begin{pmatrix} a_8\,t_0^8+a_7\,t_0^7+a_6\,t_0^6+a_5\,t_0^5+a_4\,t_0^4+a_3\,t_0^3+a_2\,t_0^2+a_1\,t_0+a_0 \\ 8\,a_8\,t_0^7+7\,a_7\,t_0^6+6\,a_6\,t_0^5+5\,a_5\,t_0^4+4\,a_4\,t_0^3+3\,a_3\,t_0^2+2\,a_2\,t_0+a_1 \\ 56\,a_8\,t_0^6+42\,a_7\,t_0^5+30\,a_6\,t_0^4+20\,a_5\,t_0^3+12\,a_4\,t_0^2+6\,a_3\,t_0+2\,a_2 \\ a_8\,t_1^8+a_7\,t_1^7+a_6\,t_1^6+a_5\,t_1^5+a_4\,t_1^4+a_3\,t_1^3+a_2\,t_1^2+a_1\,t_1+a_0 \\ 8\,a_8\,t_1^7+7\,a_7\,t_1^6+6\,a_6\,t_1^5+5\,a_5\,t_1^4+4\,a_4\,t_1^3+3\,a_3\,t_1^2+2\,a_2\,t_1+a_1 \\ 56\,a_8\,t_1^6+42\,a_7\,t_1^5+30\,a_6\,t_1^4+20\,a_5\,t_1^3+12\,a_4\,t_1^2+6\,a_3\,t_1+2\,a_2 \\ a_8\,t_2^8+a_7\,t_2^7+a_6\,t_2^6+a_5\,t_2^5+a_4\,t_2^4+a_3\,t_2^3+a_2\,t_2^2+a_1\,t_2+a_0 \\ 8\,a_8\,t_2^7+7\,a_7\,t_2^6+6\,a_6\,t_2^5+5\,a_5\,t_2^4+4\,a_4\,t_2^3+3\,a_3\,t_2^2+2\,a_2\,t_2+a_1 \\ 56\,a_8\,t_2^6+42\,a_7\,t_2^5+30\,a_6\,t_2^4+20\,a_5\,t_2^3+12\,a_4\,t_2^2+6\,a_3\,t_2+2\,a_2 \end{pmatrix}$$

```
D=subs(D,{t_0,t_1,t_2},{0,5,10})
```

D =

$$\begin{pmatrix} a_0 \\ a_1 \\ 2\,a_2 \\ a_0+5\,a_1+25\,a_2+125\,a_3+625\,a_4+3125\,a_5+15625\,a_6+78125\,a_7+390625\,a_8 \\ a_1+10\,a_2+75\,a_3+500\,a_4+3125\,a_5+18750\,a_6+109375\,a_7+625000\,a_8 \\ 2\,a_2+30\,a_3+300\,a_4+2500\,a_5+18750\,a_6+131250\,a_7+875000\,a_8 \\ a_0+10\,a_1+100\,a_2+1000\,a_3+10000\,a_4+100000\,a_5+1000000\,a_6+10000000\,a_7+100000000\,a_8 \\ a_1+20\,a_2+300\,a_3+4000\,a_4+50000\,a_5+600000\,a_6+7000000\,a_7+80000000\,a_8 \\ 2\,a_2+60\,a_3+1200\,a_4+20000\,a_5+300000\,a_6+4200000\,a_7+56000000\,a_8 \end{pmatrix}$$

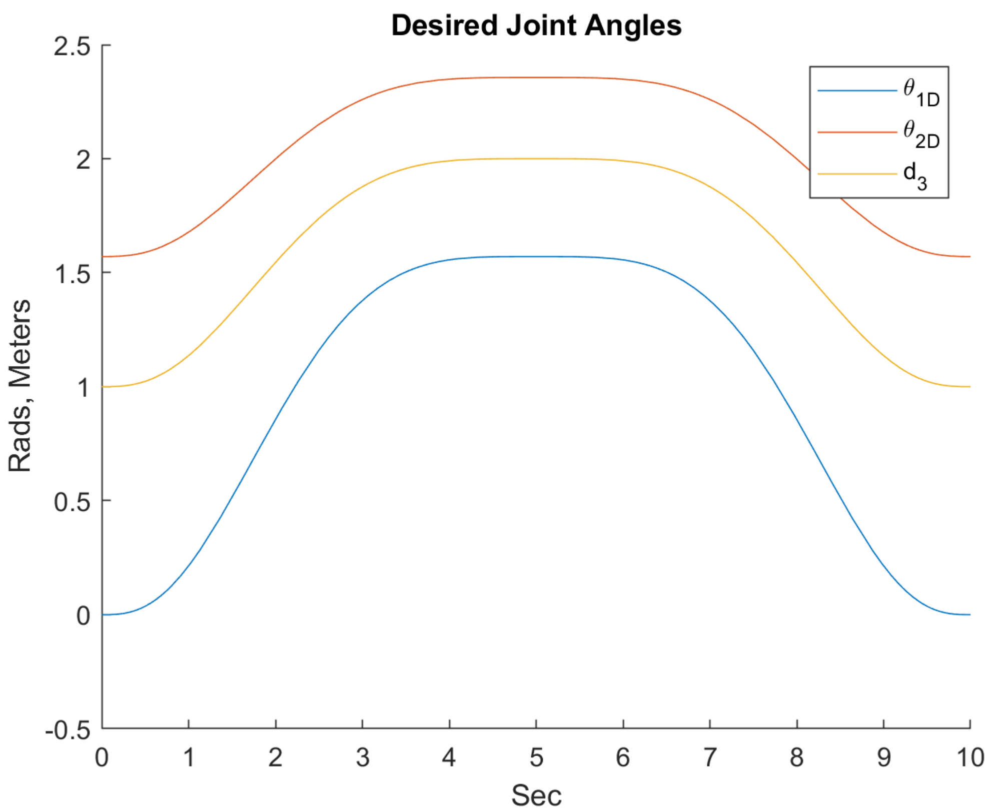
```
V1=[0;0;0;pi/2;0;0;0;0;0];V2=[pi/2;0;0;3*pi/4;0;0;pi/2;0;0];V3=[1;0;0;2;0;0;1;0;0];
[A1] = equationsToMatrix(D, [a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8])
```

A1 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 25 & 125 & 625 & 3125 & 15625 & 78125 & 390625 \\ 0 & 1 & 10 & 75 & 500 & 3125 & 18750 & 109375 & 625000 \\ 0 & 0 & 2 & 30 & 300 & 2500 & 18750 & 131250 & 875000 \\ 1 & 10 & 100 & 1000 & 10000 & 100000 & 1000000 & 10000000 & 100000000 \\ 0 & 1 & 20 & 300 & 4000 & 50000 & 600000 & 7000000 & 80000000 \\ 0 & 0 & 2 & 60 & 1200 & 20000 & 300000 & 4200000 & 56000000 \end{pmatrix}$$

```
X1 = linsolve(A1,V1);X2 = linsolve(A1,V2);X3 = linsolve(A1,V3);
destheta1=subs(q1,[a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8],transpose(X1));destheta2=subs(q1,[a_0,a

figure(1)
clf
hold on
fplot(destheta1,[0 10]);
fplot(destheta2,[0 10]);
```

```
fplot(desd3,[0 10]);
legend('\theta_1_D','\theta_2_D','d_3')
xlabel('Sec')
ylabel('Rads, Meters')
title('Desired Joint Angles')
hold off
```




```
%Tdes(dh(1:2,:))

d1=1;
x=(desd3*cos(destheta1)*sin(destheta2))
```

x =

$$\sin\left(-\frac{\sigma_1}{1562500}+\frac{6\,\pi\,t^7}{78125}-\frac{19\,\pi\,t^6}{15625}+\frac{6\,\pi\,t^5}{625}-\frac{24\,\pi\,t^4}{625}+\frac{8\,\pi\,t^3}{125}+\frac{\pi}{2}\right)\cos\left(-\frac{\sigma_1}{781250}+\frac{12\,\pi\,t^7}{78125}-\frac{38\,\pi\,t^6}{15625}+\frac{12}{62}\right.$$

where

$$\sigma_1 = 3\,\pi\,t^8$$

```
y=(desd3*sin(destheta1)*sin(destheta2))
```

y =

$$\sin\left(-\frac{\sigma_1}{1562500}+\frac{6\,\pi\,t^7}{78125}-\frac{19\,\pi\,t^6}{15625}+\frac{6\,\pi\,t^5}{625}-\frac{24\,\pi\,t^4}{625}+\frac{8\,\pi\,t^3}{125}+\frac{\pi}{2}\right)\sin\left(-\frac{\sigma_1}{781250}+\frac{12\,\pi\,t^7}{78125}-\frac{38\,\pi\,t^6}{15625}+\frac{12\,\pi}{62}\right.$$

where

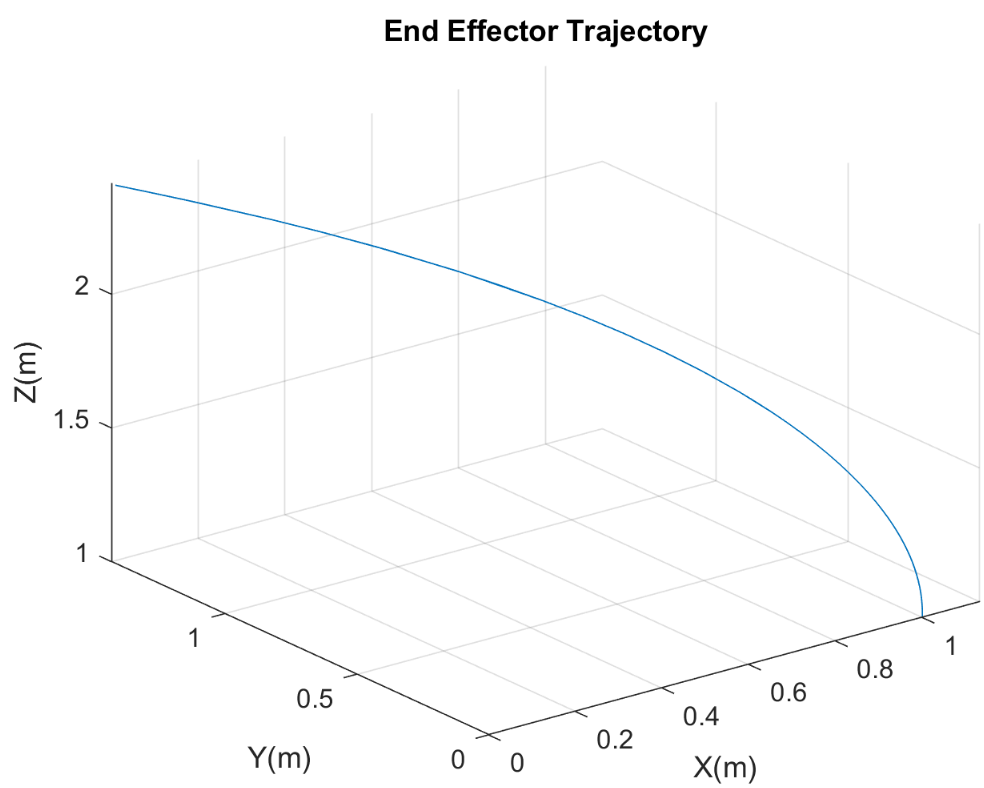
$$\sigma_1 = 3\,\pi\,t^8$$

```
z=(d1-desd3*cos(destheta2))
```

z =

$$1-\cos\left(-\frac{3\,\pi\,t^8}{1562500}+\frac{6\,\pi\,t^7}{78125}-\frac{19\,\pi\,t^6}{15625}+\frac{6\,\pi\,t^5}{625}-\frac{24\,\pi\,t^4}{625}+\frac{8\,\pi\,t^3}{125}+\frac{\pi}{2}\right)\left(-\frac{3\,t^8}{390625}+\frac{24\,t^7}{78125}-\frac{76\,t^6}{15625}+\frac{24}{62}\right.$$

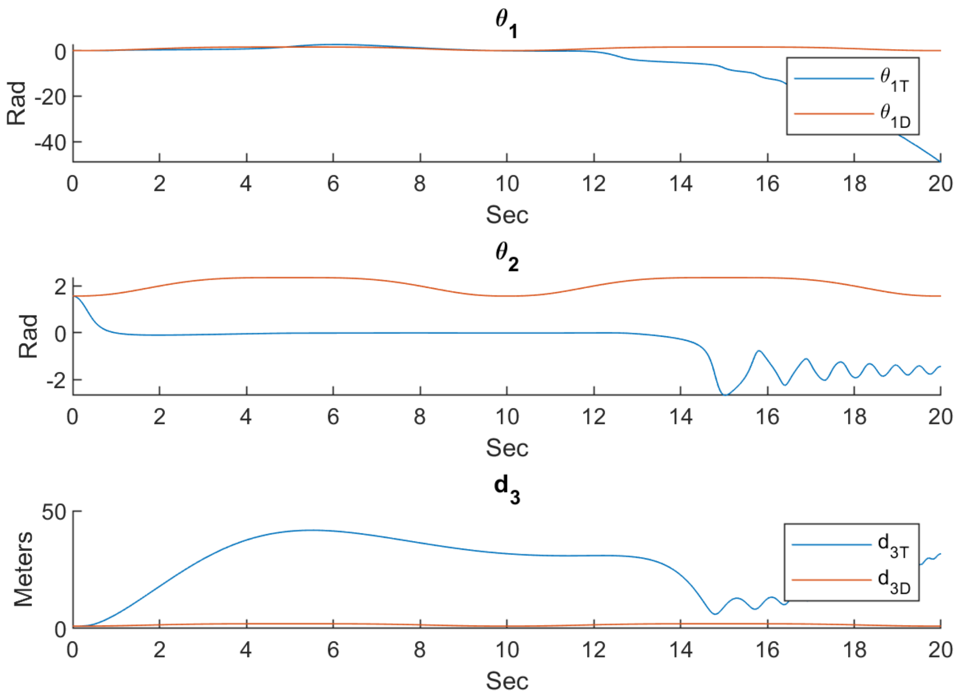
```
figure(2)
fplot3(x,y,z,[0 10])
title('End Effector Trajectory')
xlabel('X(m)')
ylabel('Y(m)')
zlabel('Z(m)')
```



Proportional Control-no gravity

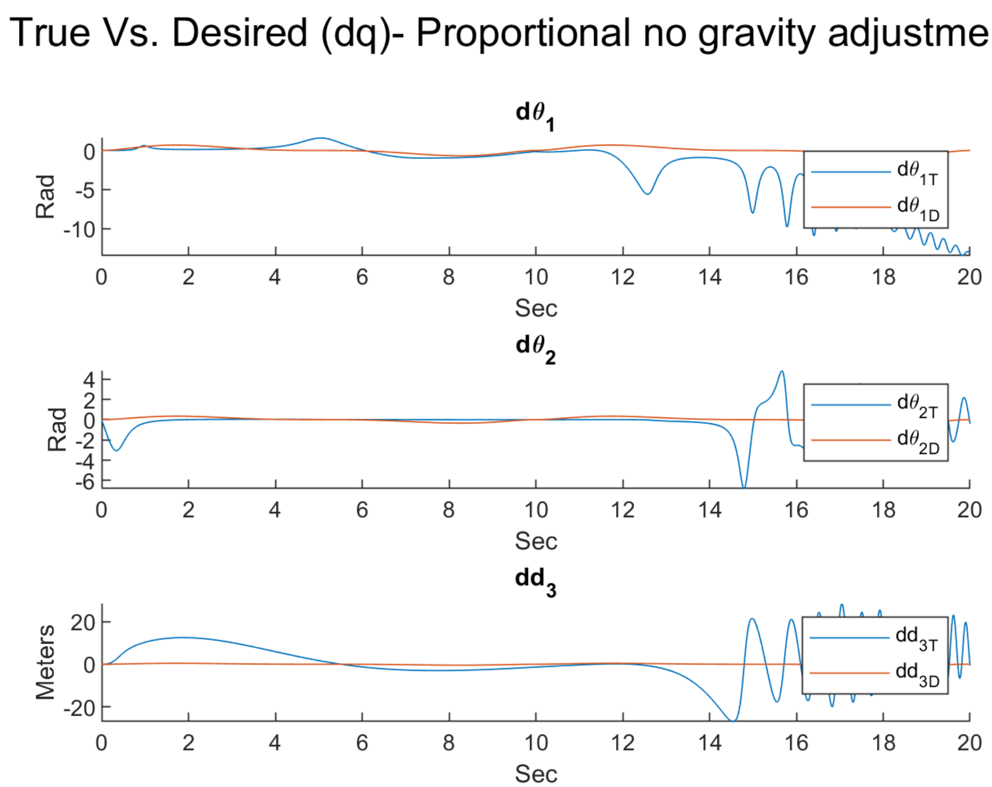
```
tspan=[0 20];
intal=[0; pi/2; 1; 0; 0; 0];
[t,y] = ode45(@PNG,tspan,intal);
X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625; (12*pi)/78125; -(3*pi)/781
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125); -(3/390625)];

x=t;



desq1=X1(9).*(x-floor(x./10)*10).^8+X1(8).*(x-floor(x./10)*10).^7+X1(7).*(x-floor(x./10)*10).^6
desq2=X2(9).*(x-floor(x./10)*10).^8+X2(8).*(x-floor(x./10)*10).^7+X2(7).*(x-floor(x./10)*10).^6
desq3=X3(9).*(x-floor(x./10)*10).^8+X3(8).*(x-floor(x./10)*10).^7+X3(7).*(x-floor(x./10)*10).^6
desdq1=8*X1(9).*(x-floor(x./10)*10).^7+7*X1(8).*(x-floor(x./10)*10).^6+6*X1(7).*(x-floor(x./10)
desdq2=8*X2(9).*(x-floor(x./10)*10).^7+7*X2(8).*(x-floor(x./10)*10).^6+6*X2(7).*(x-floor(x./10)
desdq3=8*X3(9).*(x-floor(x./10)*10).^7+7*X3(8).*(x-floor(x./10)*10).^6+6*X3(7).*(x-floor(x./10)



figure(3)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,desq1)
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,desq2)
%legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,desq3)
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (q)- Proportional no gravity adjustment')
```

True Vs. Desired (q)- Proportional no gravity adjustment




```
figure(4)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,desdq1)
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
plot(t,desdq2)
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,desdq3)
```

```
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (dq)- Proportional no gravity adjustment')
```

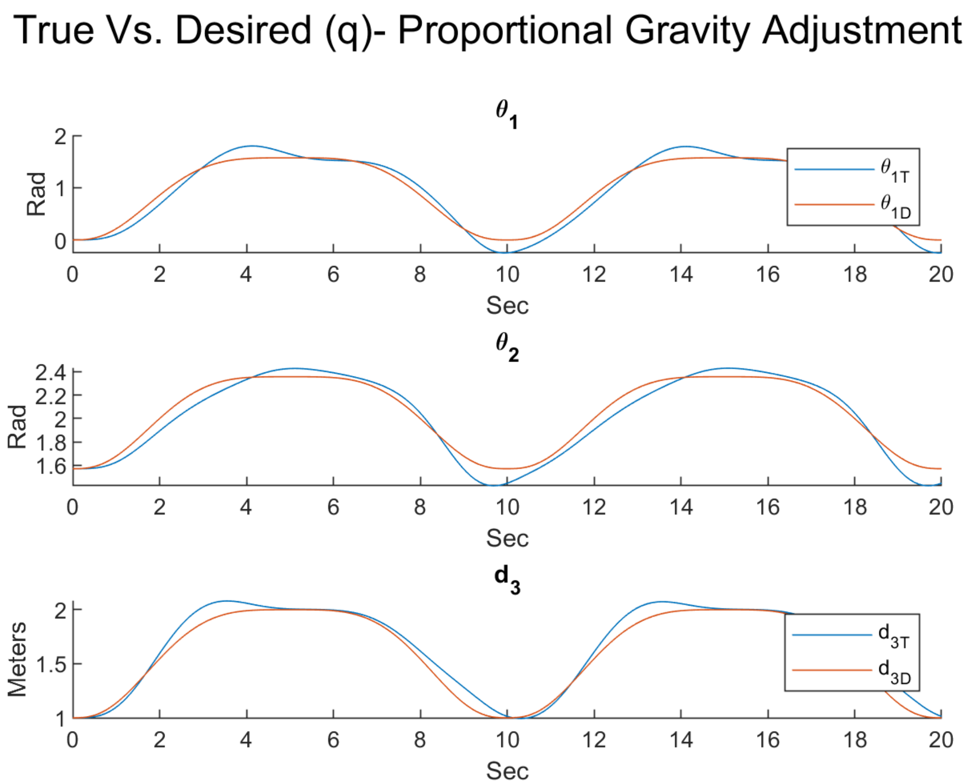


As expected it fails

Proportional Control Gravity Adjustment

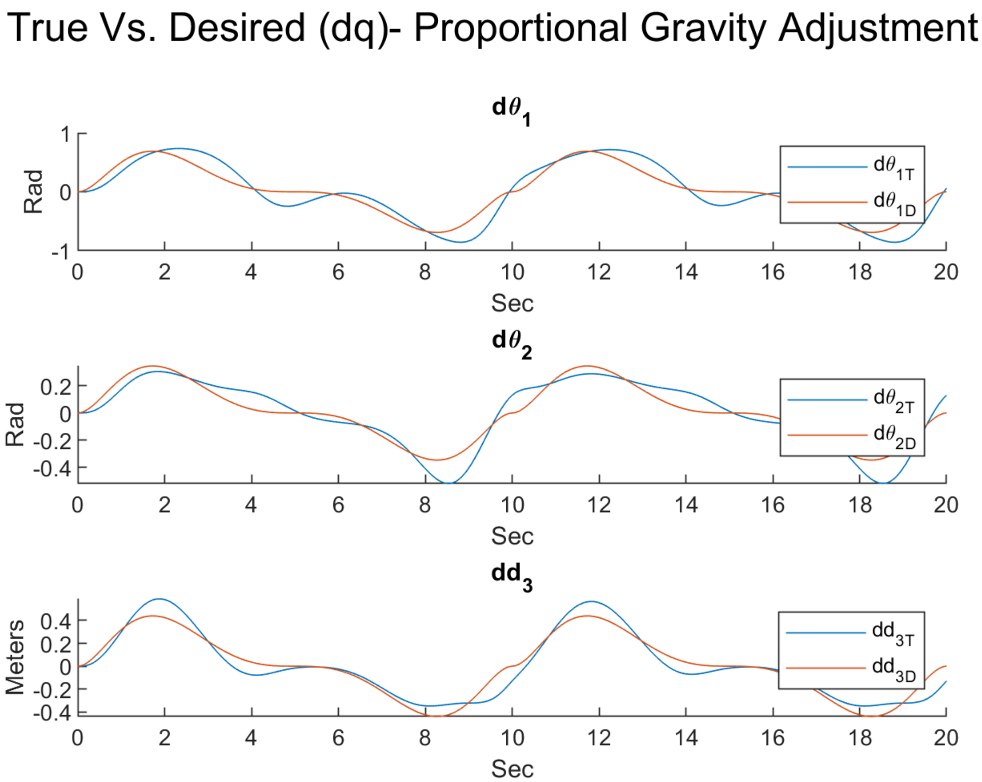
```
tspan=[0 20];
intal=[0; pi/2; 1; 0; 0; 0];
[t,y] = ode45(@PGA,tspan,intal);
X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625; (12*pi)/78125; -(3*pi)/781
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125); -(3/390625)];

x=t;



desq1=X1(9).*(x-floor(x./10)*10).^8+X1(8).*(x-floor(x./10)*10).^7+X1(7).*(x-floor(x./10)*10).^6
desq2=X2(9).*(x-floor(x./10)*10).^8+X2(8).*(x-floor(x./10)*10).^7+X2(7).*(x-floor(x./10)*10).^6
desq3=X3(9).*(x-floor(x./10)*10).^8+X3(8).*(x-floor(x./10)*10).^7+X3(7).*(x-floor(x./10)*10).^6
desdq1=8*X1(9).*(x-floor(x./10)*10).^7+7*X1(8).*(x-floor(x./10)*10).^6+6*X1(7).*(x-floor(x./10
desdq2=8*X2(9).*(x-floor(x./10)*10).^7+7*X2(8).*(x-floor(x./10)*10).^6+6*X2(7).*(x-floor(x./10
```

```
desdq3=8*X3(9).*(x-floor(x./10)*10).^7+7*X3(8).*(x-floor(x./10)*10).^6+6*X3(7).*(x-floor(x./10)



figure(5)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,desq1)
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,desq2)
%legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,desq3)
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (q)- Proportional Gravity Adjustment')
```




```
figure(6)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,desdq1)
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
plot(t,desdq2)
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,desdq3)
```

```
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (dq)- Proportional Gravity Adjustment')
```

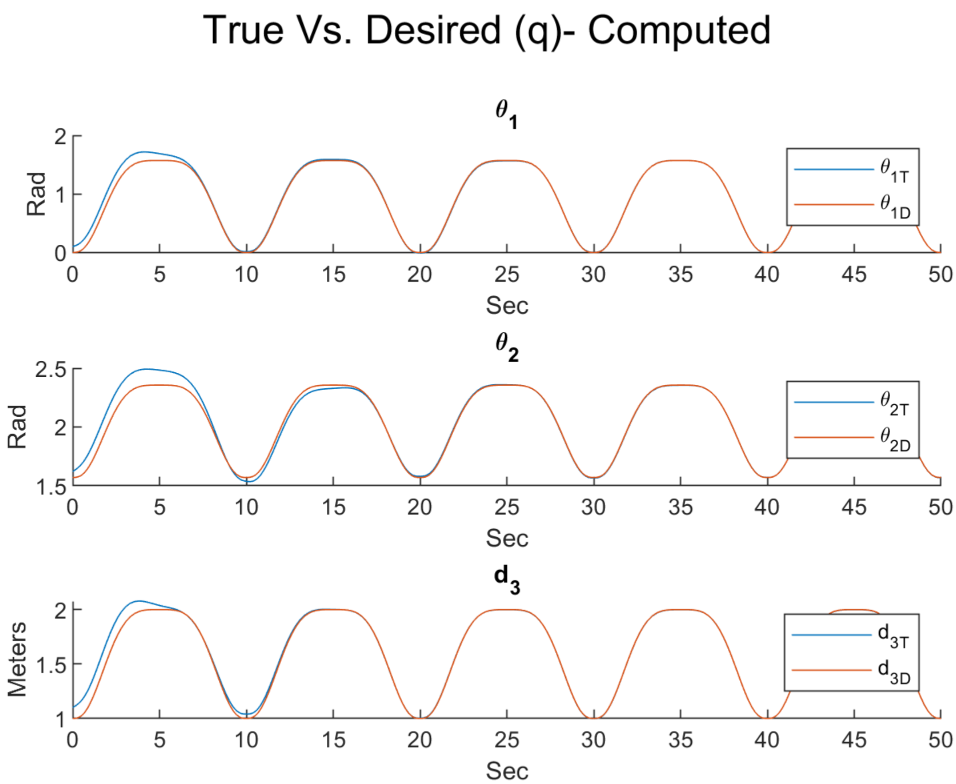



```

```

Computed Torque

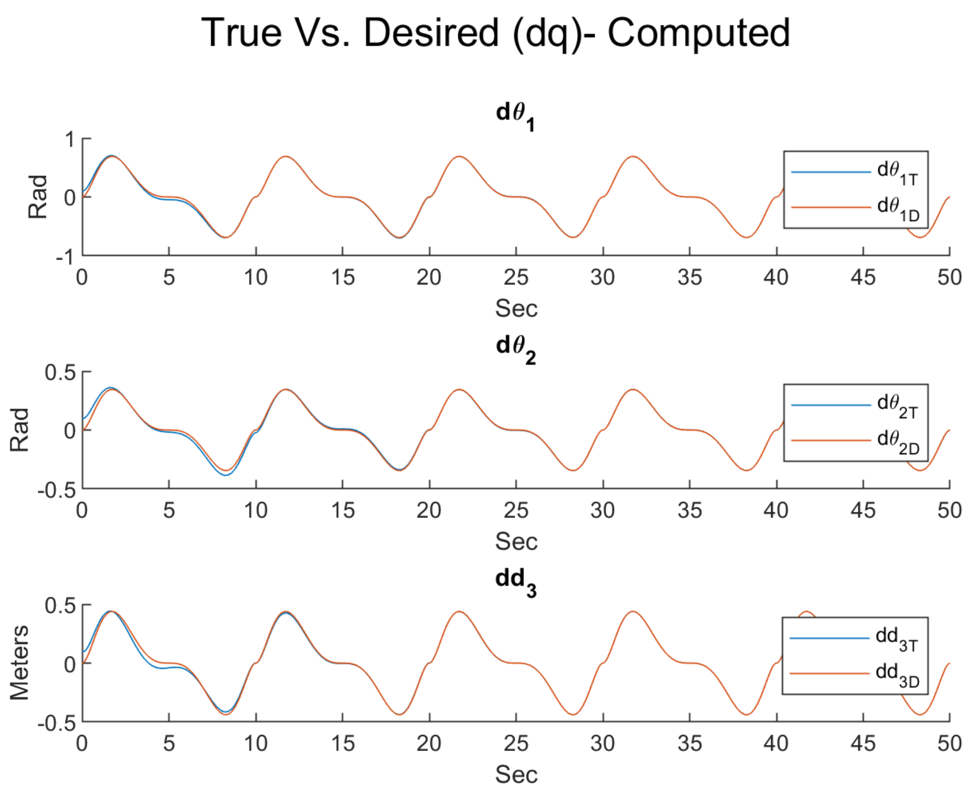
```
tspan=[0 50];
intal=[0.1; (pi+.1)/2; 1.1; 0.1; 0.1; 0.1];
[t,y] = ode45(@CT,tspan,intal);
X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625; (12*pi)/78125; -(3*pi)/781
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125); -(3/390625)];

x=t;



desq1=X1(9).*(x-floor(x./10)*10).^8+X1(8).*(x-floor(x./10)*10).^7+X1(7).*(x-floor(x./10)*10).^6
desq2=X2(9).*(x-floor(x./10)*10).^8+X2(8).*(x-floor(x./10)*10).^7+X2(7).*(x-floor(x./10)*10).^6
desq3=X3(9).*(x-floor(x./10)*10).^8+X3(8).*(x-floor(x./10)*10).^7+X3(7).*(x-floor(x./10)*10).^6
desdq1=8*X1(9).*(x-floor(x./10)*10).^7+7*X1(8).*(x-floor(x./10)*10).^6+6*X1(7).*(x-floor(x./10
desdq2=8*X2(9).*(x-floor(x./10)*10).^7+7*X2(8).*(x-floor(x./10)*10).^6+6*X2(7).*(x-floor(x./10
```

```
desdq3=8*X3(9).*(x-floor(x./10)*10).^7+7*X3(8).*(x-floor(x./10)*10).^6+6*X3(7).*(x-floor(x./10



figure(7)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,desq1)
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,desq2)
legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,desq3)
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (q)- Computed')
```




```
figure(8)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,desdq1)
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
plot(t,desdq2)
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,desdq3)
```

```
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (dq)- Computed')
```



Derive Adaptive Controller

```
%Mre
syms ddtheta1r ddtheta1r ddd3r dtheta1r dtheta2r dd3r
V=children(expand(Mre*[ddtheta1r;ddtheta2r;ddd3r]));
Y1_1=[sum(has([V{1}{:}],M4).*[V{1}{:}]);sum(has([V{2}{:}],M4).*[V{2}{:}]);sum(has([V{3}{:}],M4
Y0_1=[sum(not(has([V{1}{:}],M4)).*[V{1}{:}]);sum(not(has([V{2}{:}],M4)).*[V{2}{:}]);sum(not(has

%C
V=children(expand(C*[dtheta1r;dtheta2r;dd3r]));
Y1_2=[sum(has([V{1}{:}],M4).*[V{1}{:}]);sum(has([V{2}{:}],M4).*[V{2}{:}]);sum(has([V{3}{:}],M4
Y0_2=[sum(not(has([V{1}{:}],M4)).*[V{1}{:}]);sum(not(has([V{2}{:}],M4)).*[V{2}{:}]);sum(not(has


V=children(expand(G));
Y1_3=[sum(has([V{1}{:}],M4).*[V{1}{:}]);sum(has([V{2}{:}],M4).*[V{2}{:}]);sum(has([V{3}{:}],M4
Y0_3=[sum(not(has([V{1}{:}],M4)).*[V{1}{:}]);sum(not(has([V{2}{:}],M4)).*[V{2}{:}]);sum(not(has


Y1=simplify((Y1_1+Y1_2+Y1_3)/M4)
```

```
Y1 =
```

$$\left(\sin(\theta_2(t))\,d_3(t)\ \left(\text{dtheta1r}\,\sin(\theta_2(t))\,\frac{\partial}{\partial t}\,d_3(t)+\text{dd3r}\,\sin(\theta_2(t))\,\frac{\partial}{\partial t}\,\theta_1(t)+\text{ddtheta1r}\,\sin(\theta_2(t))\,d_3(t)+\text{dthe}\right.\right.$$

$$\frac{d_3(t)\ \left(2\,g\,\sin(\theta_2(t))+2\,\text{dtheta2r}\,\frac{\partial}{\partial t}\,d_3(t)+2\,\text{dd3r}\,\frac{\partial}{\partial t}\,\theta_2(t)+2\,\text{ddtheta2r}\,d_3(t\right.}{2}$$

$$-\text{dtheta1r}\,d_3(t)\,\sin(\theta_2(t))^2\,\frac{\partial}{\partial t}\,\theta_1(t)-\text{dtheta2r}\,d_3(t)\,\frac{\partial}{\partial t}\,\theta_2(t)+$$

```
Y0=simplify((Y0_1+Y0_2+Y0_3))
```

```
Y0 =
```

$$\left(\frac{I_{2XX}\,\text{ddtheta1r}}{2}+\frac{I_{3XX}\,\text{ddtheta1r}}{2}+I_{1ZZ}\,\text{ddtheta1r}+\frac{I_{2ZZ}\,\text{ddtheta1r}}{2}+\frac{I_{3ZZ}\,\text{ddtheta1r}}{2}+\frac{M_3\,\text{ddtheta1r}\,d_3(t)^2}{8}-\right.$$

where

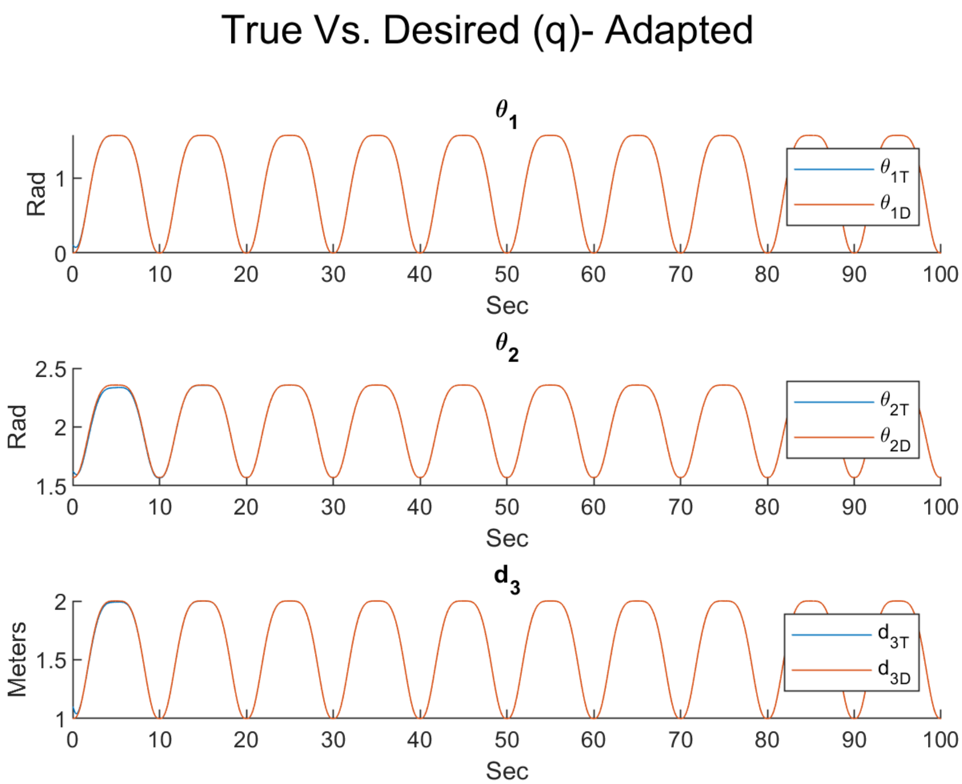
$$\sigma_1=\sin(2\,\theta_2(t))$$

$$\sigma_2=\cos(2\,\theta_2(t))$$

Adaptive Controller

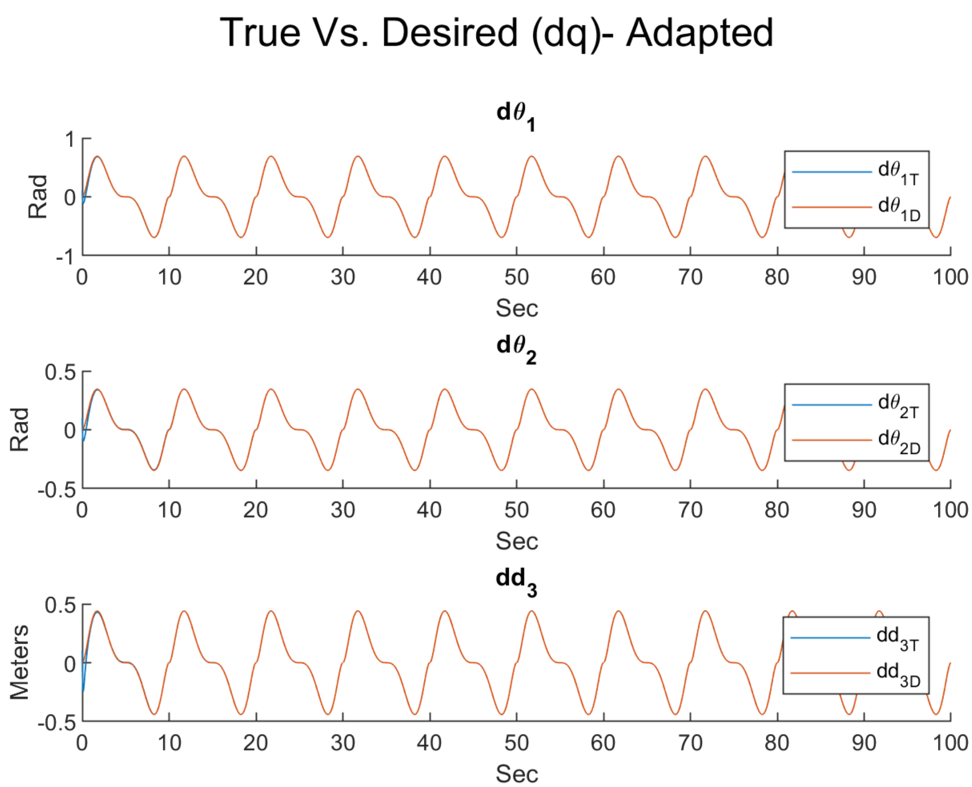
```
tspan=[0 100];
intal=[0.1; (pi+.1)/2; 1.1; 0.1; 0.1; 0.1; 12];
[t,y] = ode45(@AC,tspan,intal);
X1=[0; 0; 0; (16*pi)/125; -(48*pi)/625; (12*pi)/625; -(38*pi)/15625; (12*pi)/78125; -(3*pi)/781
X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X3=[(1); 0; 0; (32/125); -(96/625); (24/625); -(76/15625); (24/78125); -(3/390625)];

x=t;



desq1=X1(9).*(x-floor(x./10)*10).^8+X1(8).*(x-floor(x./10)*10).^7+X1(7).*(x-floor(x./10)*10).^6
desq2=X2(9).*(x-floor(x./10)*10).^8+X2(8).*(x-floor(x./10)*10).^7+X2(7).*(x-floor(x./10)*10).^6
desq3=X3(9).*(x-floor(x./10)*10).^8+X3(8).*(x-floor(x./10)*10).^7+X3(7).*(x-floor(x./10)*10).^6
desdq1=8*X1(9).*(x-floor(x./10)*10).^7+7*X1(8).*(x-floor(x./10)*10).^6+6*X1(7).*(x-floor(x./10)
desdq2=8*X2(9).*(x-floor(x./10)*10).^7+7*X2(8).*(x-floor(x./10)*10).^6+6*X2(7).*(x-floor(x./10)
desdq3=8*X3(9).*(x-floor(x./10)*10).^7+7*X3(8).*(x-floor(x./10)*10).^6+6*X3(7).*(x-floor(x./10)
```

```
figure(9)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,desq1)
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,desq2)
legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,desq3)
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (q)- Adapted')
```




```
figure(10)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,desdq1)
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
plot(t,desdq2)
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,desdq3)
```

```
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired (dq)- Adapted')
```



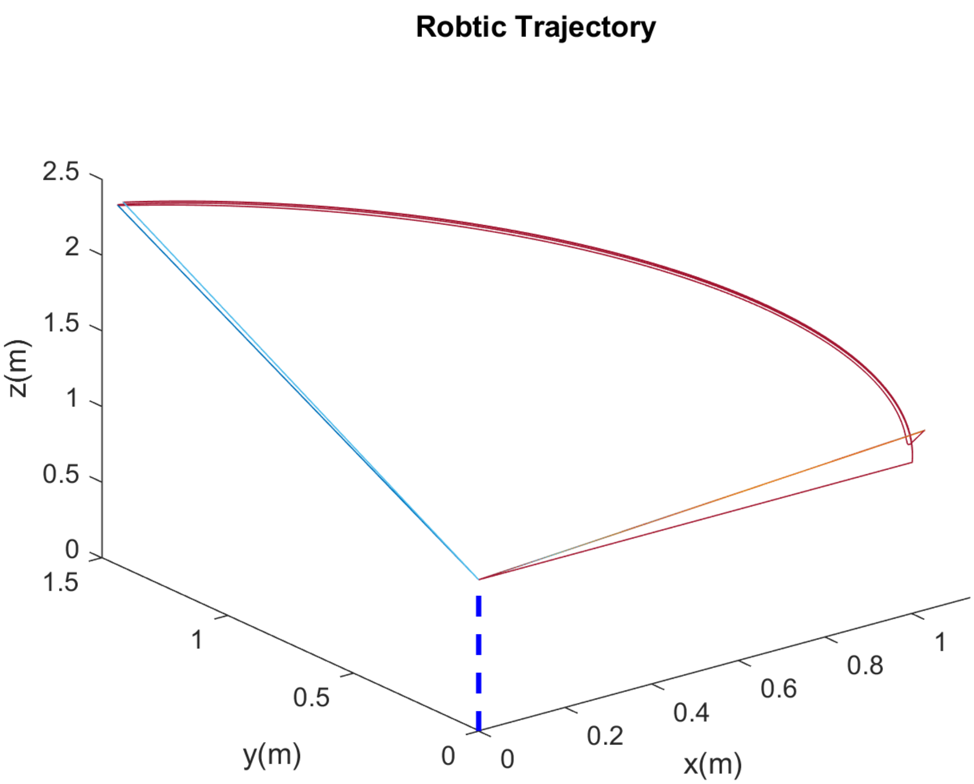
Plot Results

```
%End effector

d1=1;
x3=(y(:,3).*cos(y(:,1)).*sin(y(:,2)));
y3=(y(:,3).*sin(y(:,1)).*sin(y(:,2)));
z3=(d1-y(:,3).*cos(y(:,2)));

%Origin and link 1
x0=[0 0];
y0=[0 0];
z0=[0 1];

%link 2
i=0;
p=[0,0,0];
for x=1:length(t)
    if rem(t(x),5)<0.0051
        p=[p;y(x,1:3)];
        i=i+1;
```

```
    end
end
i
```

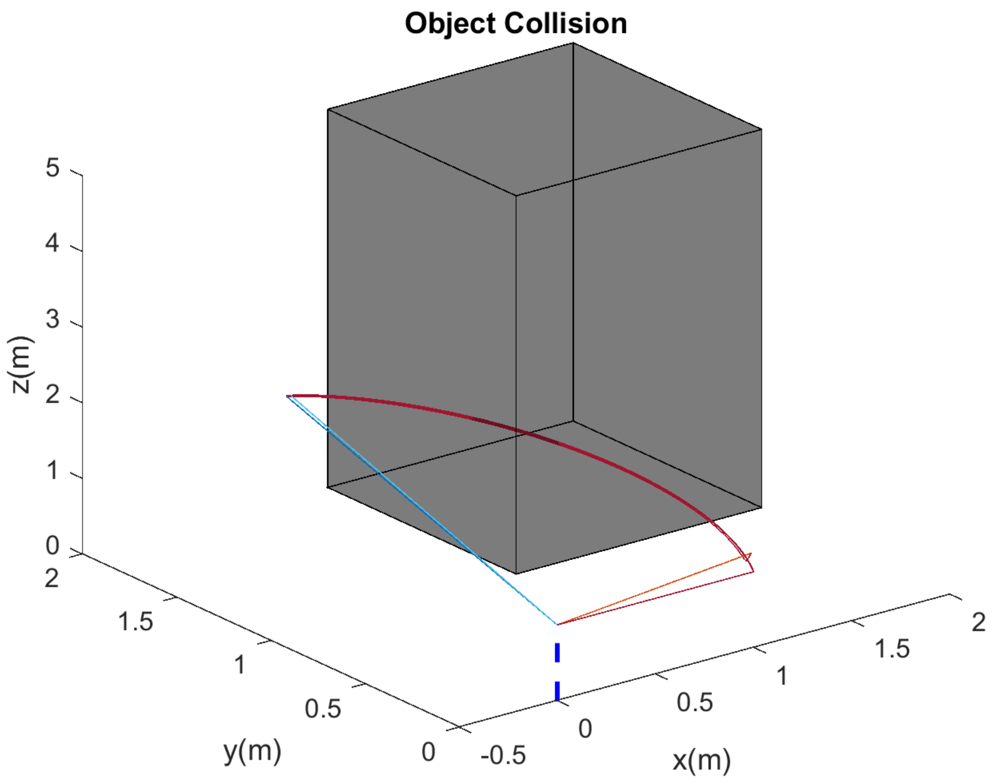
```
i = 20
```

```
p=p(2:length(p),:);

x2=[zeros(length(p),1), (p(:,3).*cos(p(:,1)).*sin(p(:,2)))];
y2=[zeros(length(p),1), (p(:,3).*sin(p(:,1)).*sin(p(:,2)))];
z2=[ones(length(p),1), (d1-p(:,3).*cos(p(:,2)))];




figure(11)
hold on

plot3(x3,y3,z3)
plot3(x0,y0,z0,'--','LineWidth',2,'Color','Blue')
plot3(transpose(x2),transpose(y2),transpose(z2))
title('Robtic Trajectory')
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
```



Obstacale Avoidence (Optional) pt.1

```
figure(11)

x1 = [2 2 0.75 0.75];
y1 = [1 2 2 1];
z1 = [0 0 0 0];

x2 = [0.75 0.75 0.75 0.75];
y2 = [1 2 2 1];
z2 = [0 0 5 5];

x3 = [0.75 2 2 0.75];
y3 = [1 1 1 1];
z3 = [0 0 5 5];

x4 = [2 2 0.75 0.75];
y4 = [1 2 2 1];
z4 = [5 5 5 5];

x5 = [0.75 2 2 0.75];
y5 = [2 2 2 2];
z5 = [0 0 5 5];

x6 = [2 2 2 2];
y6 = [1 2 2 1];
z6 = [0 0 5 5];

clear alpha

patch(x1,y1,z1,'black')
%alpha(0.3)
patch(x2,y2,z2,'black')
%alpha(0.3)
patch(x3,y3,z3,'black')
%alpha(0.3)
patch(x4,y4,z4,'black')
%alpha(0.3)
patch(x5,y5,z5,'black')
%alpha(0.3)
patch(x6,y6,z6,'black')
alpha(0.3)

title('Object Collision')
```



Design new trajectory

```
syms t a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_10 t_0 t_1 t_2 t_3 t_4

q1=a_0+(a_1*t)+(a_2*t^2)+(a_3*t^3)+(a_4*t^4)+(a_5*t^5)+(a_6*t^6)+(a_7*t^7)+(a_8*t^8)+(a_9*t^9)
```

q1 = $a_{10}\,t^{10} + a_9\,t^9 + a_8\,t^8 + a_7\,t^7 + a_6\,t^6 + a_5\,t^5 + a_4\,t^4 + a_3\,t^3 + a_2\,t^2 + a_1\,t + a_0$

```
dq1=diff(q1,t)
```

dq1 = $10\,a_{10}\,t^9 + 9\,a_9\,t^8 + 8\,a_8\,t^7 + 7\,a_7\,t^6 + 6\,a_6\,t^5 + 5\,a_5\,t^4 + 4\,a_4\,t^3 + 3\,a_3\,t^2 + 2\,a_2\,t + a_1$

```
ddq1=diff(dq1,t)
```

ddq1 = $90\,a_{10}\,t^8 + 72\,a_9\,t^7 + 56\,a_8\,t^6 + 42\,a_7\,t^5 + 30\,a_6\,t^4 + 20\,a_5\,t^3 + 12\,a_4\,t^2 + 6\,a_3\,t + 2\,a_2$

```
D=[subs(q1,t_0);subs(dq1,t_0);subs(ddq1,t_0);subs(q1,t_1);
    subs(q1,t_2);subs(dq1,t_2);subs(ddq1,t_2);subs(q1,t_3);
    subs(q1,t_4);subs(dq1,t_4);subs(ddq1,t_4)]
```

D =

$$\begin{pmatrix} a_{10}\,t_0^{10}+a_9\,t_0^9+a_8\,t_0^8+a_7\,t_0^7+a_6\,t_0^6+a_5\,t_0^5+a_4\,t_0^4+a_3\,t_0^3+a_2\,t_0^2+a_1\,t_0+a_0 \\ 10\,a_{10}\,t_0^9+9\,a_9\,t_0^8+8\,a_8\,t_0^7+7\,a_7\,t_0^6+6\,a_6\,t_0^5+5\,a_5\,t_0^4+4\,a_4\,t_0^3+3\,a_3\,t_0^2+2\,a_2\,t_0+a_1 \\ 90\,a_{10}\,t_0^8+72\,a_9\,t_0^7+56\,a_8\,t_0^6+42\,a_7\,t_0^5+30\,a_6\,t_0^4+20\,a_5\,t_0^3+12\,a_4\,t_0^2+6\,a_3\,t_0+2\,a_2 \\ a_{10}\,t_1^{10}+a_9\,t_1^9+a_8\,t_1^8+a_7\,t_1^7+a_6\,t_1^6+a_5\,t_1^5+a_4\,t_1^4+a_3\,t_1^3+a_2\,t_1^2+a_1\,t_1+a_0 \\ a_{10}\,t_2^{10}+a_9\,t_2^9+a_8\,t_2^8+a_7\,t_2^7+a_6\,t_2^6+a_5\,t_2^5+a_4\,t_2^4+a_3\,t_2^3+a_2\,t_2^2+a_1\,t_2+a_0 \\ 10\,a_{10}\,t_2^9+9\,a_9\,t_2^8+8\,a_8\,t_2^7+7\,a_7\,t_2^6+6\,a_6\,t_2^5+5\,a_5\,t_2^4+4\,a_4\,t_2^3+3\,a_3\,t_2^2+2\,a_2\,t_2+a_1 \\ 90\,a_{10}\,t_2^8+72\,a_9\,t_2^7+56\,a_8\,t_2^6+42\,a_7\,t_2^5+30\,a_6\,t_2^4+20\,a_5\,t_2^3+12\,a_4\,t_2^2+6\,a_3\,t_2+2\,a_2 \\ a_{10}\,t_3^{10}+a_9\,t_3^9+a_8\,t_3^8+a_7\,t_3^7+a_6\,t_3^6+a_5\,t_3^5+a_4\,t_3^4+a_3\,t_3^3+a_2\,t_3^2+a_1\,t_3+a_0 \\ a_{10}\,t_4^{10}+a_9\,t_4^9+a_8\,t_4^8+a_7\,t_4^7+a_6\,t_4^6+a_5\,t_4^5+a_4\,t_4^4+a_3\,t_4^3+a_2\,t_4^2+a_1\,t_4+a_0 \\ 10\,a_{10}\,t_4^9+9\,a_9\,t_4^8+8\,a_8\,t_4^7+7\,a_7\,t_4^6+6\,a_6\,t_4^5+5\,a_5\,t_4^4+4\,a_4\,t_4^3+3\,a_3\,t_4^2+2\,a_2\,t_4+a_1 \\ 90\,a_{10}\,t_4^8+72\,a_9\,t_4^7+56\,a_8\,t_4^6+42\,a_7\,t_4^5+30\,a_6\,t_4^4+20\,a_5\,t_4^3+12\,a_4\,t_4^2+6\,a_3\,t_4+2\,a_2 \end{pmatrix}$$

```
D=subs(D,{t_0,t_1,t_2,t_3,t_4},{0,2.5,5,7.5,10})
```

D =

$$\begin{pmatrix} a_0 \\ a_1 \\ 2\,a_2 \\ a_0+\frac{5\,a_1}{2}+\frac{25\,a_2}{4}+\frac{125\,a_3}{8}+\frac{625\,a_4}{16}+\frac{3125\,a_5}{32}+\frac{15625\,a_6}{64}+\frac{78125\,a_7}{128}+\frac{390625\,a_8}{256}+\frac{19\cdots}{} \\ a_0+5\,a_1+25\,a_2+125\,a_3+625\,a_4+3125\,a_5+15625\,a_6+78125\,a_7+390625\,a_8+19\cdots \\ a_1+10\,a_2+75\,a_3+500\,a_4+3125\,a_5+18750\,a_6+109375\,a_7+625000\,a_8+351562\cdots \\ 2\,a_2+30\,a_3+300\,a_4+2500\,a_5+18750\,a_6+131250\,a_7+875000\,a_8+5625000\cdots \\ a_0+\frac{15\,a_1}{2}+\frac{225\,a_2}{4}+\frac{3375\,a_3}{8}+\frac{50625\,a_4}{16}+\frac{759375\,a_5}{32}+\frac{11390625\,a_6}{64}+\frac{170859375\,a_7}{128}+\frac{2562890625\,a_8}{256}\cdots \\ a_0+10\,a_1+100\,a_2+1000\,a_3+10000\,a_4+100000\,a_5+1000000\,a_6+10000000\,a_7+100000000\,a_8\cdots \\ a_1+20\,a_2+300\,a_3+4000\,a_4+50000\,a_5+600000\,a_6+7000000\,a_7+80000000\,a_8+9000\cdots \\ 2\,a_2+60\,a_3+1200\,a_4+20000\,a_5+300000\,a_6+4200000\,a_7+56000000\,a_8+7200000\cdots \end{pmatrix}$$

```
V1=[0;0;0;0.927295218002;
    pi/2;0;0;0.927295218002;
    0;0;0];
V3=[1;0;0;1;
    2;0;0;1;
    1;0;0];
[A1] = equationsToMatrix(D, [a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_10])
```

A1 =

$$
\left(\begin{array}{ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \frac{5}{2} & \frac{25}{4} & \frac{125}{8} & \frac{625}{16} & \frac{3125}{32} & \frac{15625}{64} & \frac{78125}{128} & \frac{390625}{256} & \frac{1953125}{512} & \frac{976562}{1024} \\
1 & 5 & 25 & 125 & 625 & 3125 & 15625 & 78125 & 390625 & 1953125 & 976562 \\
0 & 1 & 10 & 75 & 500 & 3125 & 18750 & 109375 & 625000 & 3515625 & 1953125 \\
0 & 0 & 2 & 30 & 300 & 2500 & 18750 & 131250 & 875000 & 5625000 & 3515625 \\
1 & \frac{15}{2} & \frac{225}{4} & \frac{3375}{8} & \frac{50625}{16} & \frac{759375}{32} & \frac{11390625}{64} & \frac{170859375}{128} & \frac{2562890625}{256} & \frac{38443359375}{512} & \frac{576650390}{1024} \\
1 & 10 & 100 & 1000 & 10000 & 100000 & 1000000 & 10000000 & 100000000 & 1000000000 & 10000000 \\
0 & 1 & 20 & 300 & 4000 & 50000 & 600000 & 7000000 & 80000000 & 900000000 & 10000000 \\
0 & 0 & 2 & 60 & 1200 & 20000 & 300000 & 4200000 & 56000000 & 720000000 & 9000000
\end{array}\right.
$$

```
X1 = linsolve(A1,V1),X3 = linsolve(A1,V3)
```

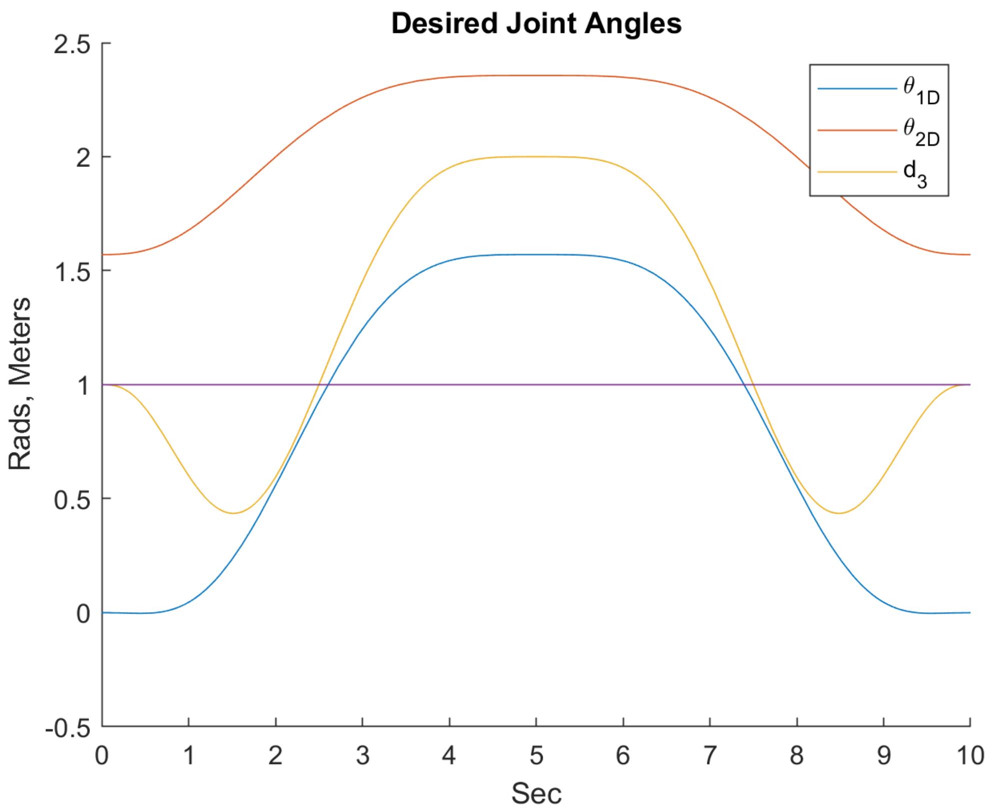
X1 =

$$
\begin{pmatrix}
0 \\
0 \\
0 \\
\frac{16704665593025}{7421703487488} - \frac{96\,\pi}{125} \\
\frac{568\,\pi}{625} - \frac{36750264304655}{14843406974976} \\
\frac{11359172603257}{9895604649984} - \frac{1368\,\pi}{3125} \\
\frac{1768\,\pi}{15625} - \frac{28732024820003}{98956046499840} \\
\frac{668186623721}{15461882265600} - \frac{1332\,\pi}{78125} \\
\frac{1173\,\pi}{781250} - \frac{4677306366047}{1236950581248000} \\
\frac{668186623721}{3710851743744000} - \frac{28\,\pi}{390625} \\
\frac{14\,\pi}{9765625} - \frac{668186623721}{185542587187200000}
\end{pmatrix}
$$

X3 =

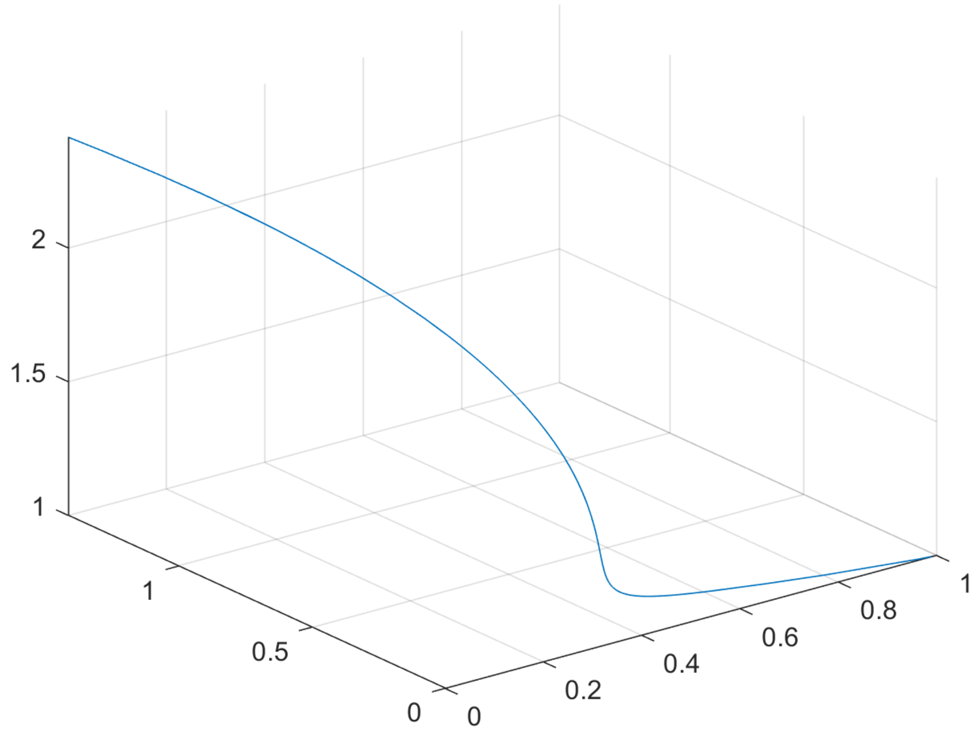
$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ -\frac{192}{125} \\ \frac{1136}{625} \\ -\frac{2736}{3125} \\ \frac{3536}{15625} \\ -\frac{2664}{78125} \\ \frac{1173}{390625} \\ -\frac{56}{390625} \\ \frac{28}{9765625} \end{pmatrix}$$

```
destheta1=subs(q1,[a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_10],transpose(X1));desd3=subs(q1,[

figure(12)
clf
hold on
fplot(destheta1,[0 10]);
fplot(destheta2,[0 10]);
fplot(desd3,[0 10]);
fplot(1,[0 10]);
legend('\theta_1_D','\theta_2_D','d_3')
xlabel('Sec')
ylabel('Rads, Meters')
title('Desired Joint Angles')
hold off
```



Will use a piecse wise trajectory

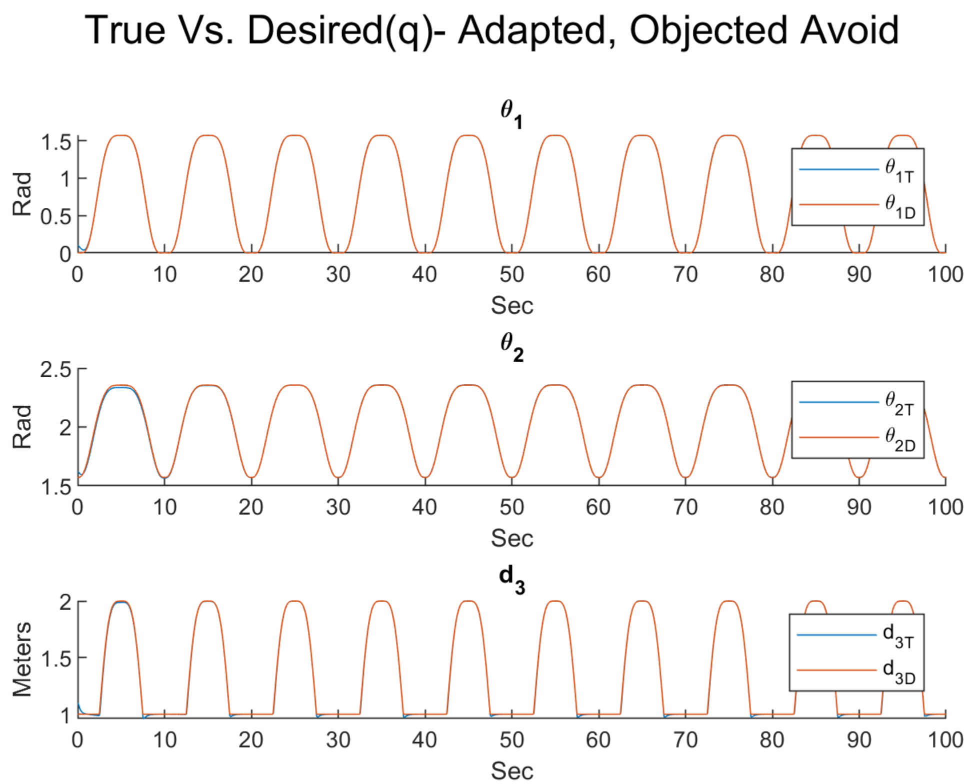
```
d1=1;
x=(desd3*cos(destheta1)*sin(destheta2));
y=(desd3*sin(destheta1)*sin(destheta2));
z=(d1-desd3*cos(destheta2));

figure(13)
fplot3(x,y,z,[0 10])
```



Plot Object Avoidence

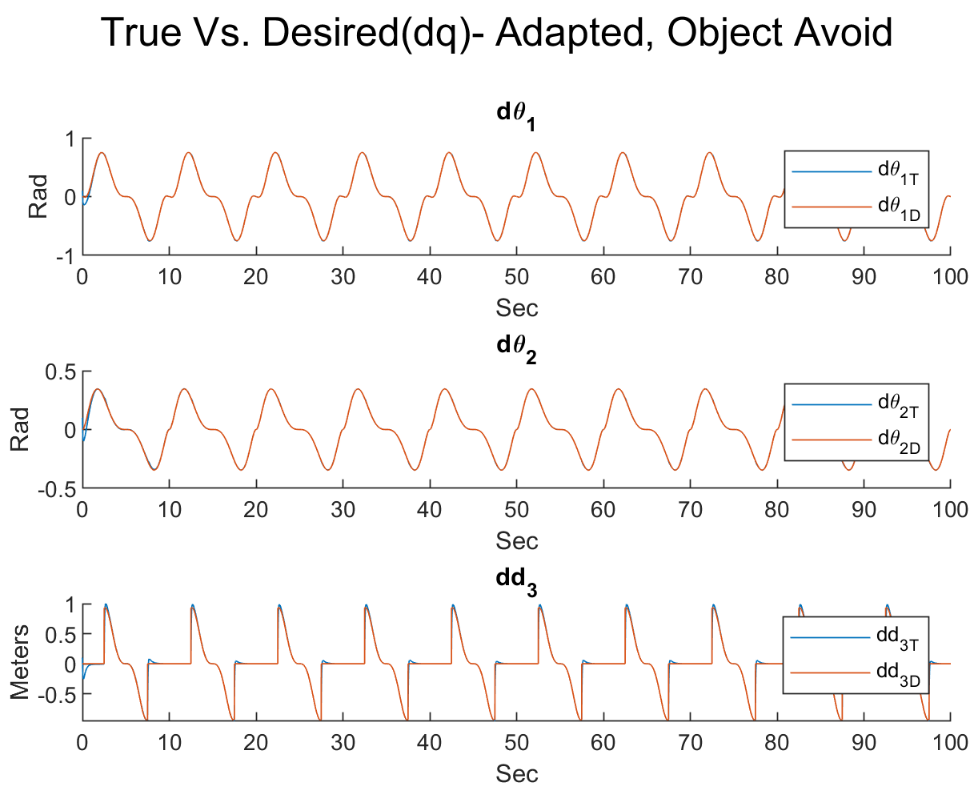
```
tspan=[0 100];
intal=[0.1; (pi+0.1)/2; 1.1; 0.1; 0.1; 0.1; 12];
[t,y] = ode45(@ACOA,tspan,intal);

X2=[pi/2; 0; 0; (8*pi)/125; -(24*pi)/625; (6*pi)/625; -(19*pi)/15625; (6*pi)/78125; -(3*pi)/156
X1=[(0);(0);(0);(16704665593025/7421703487488)-(96*(pi))/125;(568*(pi))/625-(36750264304655/148
X3=[(1);0;0;-(192/125);(1136/625);-(2736/3125);(3536/15625);-(2664/78125);(1173/390625);-(56/39

des=zeros(length(t),6);
x=t;

for k=1:length(t)
    des((k),1)=X1(11)*(x(k)-10*floor(x(k)/10))^10+X1(10)*(x(k)-10*floor(x(k)/10))^9+X1(9)*(x(k
    des((k),4)=10*X1(11)*(x(k)-10*floor(x(k)/10))^9+9*X1(10)*(x(k)-10*floor(x(k)/10))^8+8*X1(9
    des((k),2)=X2(9)*(x(k)-10*floor(x(k)/10))^8+X2(8)*(x(k)-10*floor(x(k)/10))^7+X2(7)*(x(k)-10
    des((k),5)=8*X2(9)*(x(k)-10*floor(x(k)/10))^7+7*X2(8)*(x(k)-10*floor(x(k)/10))^6+6*X2(7)*(x
    if des(k,1)>0.927295218002 && des(k,1)>0.927295218002
        des((k),3)=X3(11)*(x(k)-10*floor(x(k)/10))^10+X3(10)*(x(k)-10*floor(x(k)/10))^9+X3(9)*(
        des((k),6)=10*X3(11)*(x(k)-10*floor(x(k)/10))^9+9*X3(10)*(x(k)-10*floor(x(k)/10))^8+8*X

    else
        des((k),3)=1;
        des((k),6)=0;
    end
```

```
end



figure(14)
clf

subplot(3,1,1);
hold on
plot(t,y(:,1))
%fplot(destheta1,tspan)
plot(t,des(:,1))
legend('\theta_1_T','\theta_1_D')
title('\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,2))
%fplot(destheta2,tspan)
plot(t,des(:,2))
legend('\theta_2_T','\theta_2_D')
title('\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,3))
%fplot(desd3,tspan)
plot(t,des(:,3))
legend('d_3_T','d_3_D')
title('d_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired(q)- Adapted, Objected Avoid')
```




```
figure(15)
clf

subplot(3,1,1);
hold on
plot(t,y(:,4))
%fplot(destheta1,tspan)
plot(t,des(:,4))
legend('d\theta_1_T','d\theta_1_D')
title('d\theta_1')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,2);
hold on
plot(t,y(:,5))
%fplot(destheta2,tspan)
plot(t,des(:,5))
legend('d\theta_2_T','d\theta_2_D')
title('d\theta_2')
ylabel('Rad')
xlabel('Sec')

subplot(3,1,3);
hold on
plot(t,y(:,6))
%fplot(desd3,tspan)
plot(t,des(:,6))
```

```
legend('dd_3_T','dd_3_D')
title('dd_3')
ylabel('Meters')
xlabel('Sec')

sgtitle('True Vs. Desired(dq)- Adapted, Object Avoid')
```



Final Plotting

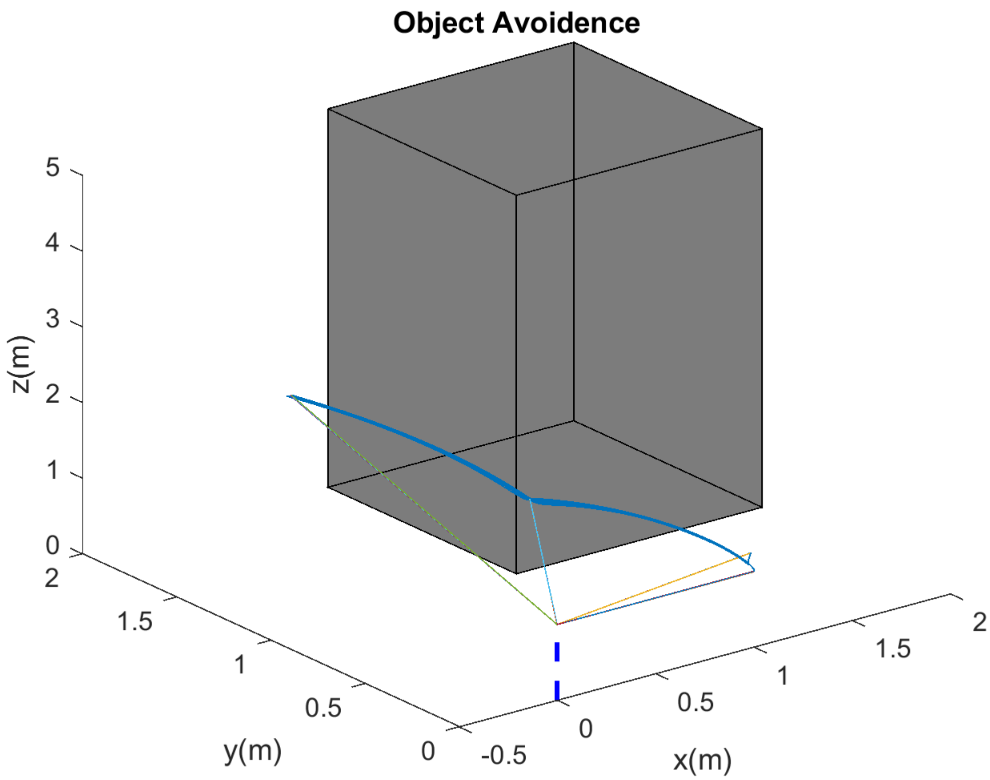
```
%End effector

d1=1;
x3e=(y(:,3).*cos(y(:,1)).*sin(y(:,2)));
y3e=(y(:,3).*sin(y(:,1)).*sin(y(:,2)));
z3e=(d1-y(:,3).*cos(y(:,2)));

%Origin and link 1
x0=[0 0];
y0=[0 0];
z0=[0 1];

%link 2
i=0;
p=[0,0,0];
for x=1:length(t)
    if rem(t(x),5)<0.0051 | rem(t(x),2.5)<0.000325 | rem(t(x),7.5)<0.000325
        p=[p;y(x,1:3)];
        i=i+1;
```

```
    end
end
i
```

```
i = 60
```

```
p=p(2:length(p),:);

x2e=[zeros(length(p),1), (p(:,3).*cos(p(:,1)).*sin(p(:,2)))];
y2e=[zeros(length(p),1), (p(:,3).*sin(p(:,1)).*sin(p(:,2)))];
z2e=[ones(length(p),1), (d1-p(:,3).*cos(p(:,2)))];




x1 = [2 2 0.75 0.75];
y1 = [1 2 2 1];
z1 = [0 0 0 0];

x2 = [0.75 0.75 0.75 0.75];
y2 = [1 2 2 1];
z2 = [0 0 5 5];

x3 = [0.75 2 2 0.75];
y3 = [1 1 1 1];
z3 = [0 0 5 5];

x4 = [2 2 0.75 0.75];
y4 = [1 2 2 1];
z4 = [5 5 5 5];

x5 = [0.75 2 2 0.75];
y5 = [2 2 2 2];
z5 = [0 0 5 5];

x6 = [2 2 2 2];
y6 = [1 2 2 1];
z6 = [0 0 5 5];

clear alpha

figure(16)
clf
hold on

patch(x1,y1,z1,'black')
%alpha(0.3)
patch(x2,y2,z2,'black')
%alpha(0.3)
patch(x3,y3,z3,'black')
%alpha(0.3)
patch(x4,y4,z4,'black')
%alpha(0.3)
patch(x5,y5,z5,'black')
```

```
%alpha(0.3)
patch(x6,y6,z6,'black')
alpha(0.3)

title('Object Avoidence')
plot3(x3e,y3e,z3e)
plot3(x0,y0,z0,'--','LineWidth',2,'Color','Blue')
plot3(transpose(x2e),transpose(y2e),transpose(z2e))
xlabel('x(m)')
ylabel('y(m)')
zlabel('z(m)')
view(3)
```



Breaking apart wanted terms

```
%
% eqm1=simplify(diff(diff(Lag,diff(theta1,t)),t)-diff(Lag,theta1))
% eqm2=simplify(diff(diff(Lag,diff(theta2,t)),t)-diff(Lag,theta2))
% eqm3=simplify(diff(diff(Lag,diff(d3,t)),t)-diff(Lag,d3))
%
%
% eqmh=0;
% MM=sym('A',[3 3]);
% MM1=sym('A',[3 3]);
% for i1=1:1:3
%     if i1==1
```

```
%           eqmh=eqm1;
%       elseif i1==2
%           eqmh=eqm2;
%       else
%           eqmh=eqm3;
%       end
%       for i2=1:1:3
%           if i2==1
%               x=diff(diff(theta1,t),t);
%           elseif i2==2
%               x=diff(diff(theta2,t),t);
%           else
%               x=diff(diff(d3,t),t);
%           end
%           z=children(eqmh);
%           f=0;
%           for i3=1:1:length(z)
%               S=z{i3};
%               if has(S,x)
%                   f=f+S;
%               end
%           end
%           MM(i1,i2)=simplify((f)/x);
%           MM1(i1,i2)=((f));
%       end
% end
% %MM
% MM1
```

C matrix

```
% trunc=children(MM1)
%
% eqm1=eqm1-sum([trunc{1,1}{:}])
% eqm2=eqm2-sum([trunc{2,2}{:}])
% eqm3=eqm3-sum([trunc{3,3}{:}])
% eqmh=0;
% CC=sym('A',[3 3]);
% CC1=sym('A',[3 3]);
% for i1=1:1:3
%       if i1==1
%           eqmh=eqm1;
%       elseif i1==2
%           eqmh=eqm2;
%       else
%           eqmh=eqm3;
%       end
%       for i2=1:1:3
%           if i2==1
%               x=diff(theta1,t);
%           elseif i2==2
%               x=diff(theta2,t);
%           else
%               x=diff(d3,t);
```

```
%           end
%           z=children(eqmh);
%           f=0;
%           for i3=1:1:length(z)
%               S=z{i3};
%               if has(S,x)
%                   if i2 == 2
%                       b=children(CC1);
%                       if has(sum([b{i1,1}{:}]),S)
%                           S=S/2; %adjusts current term
%                           %now adjust previous term
%                           e=b{i1,1};
%                           for i4=1:length(e)
%                               if has(e{i4},S*2)
%                                   l=[[e{1:i4}],S,[e{(i4+2):length(e)}]];
%                                   CC1(i1,1)=sum(l);
%                               end
%                           end
%
%                       end
%                   end
%                   if i2==3
%                       b=children(CC1);
%                       if has(sum([b{i1,1}{:}]),S/2) && has(sum([b{i1,2}{:}]),S/2)
%                           %% Adjust previous two terms and one current
%                           S=S/3
%
%                           e1=b{i1,1};
%                           e2=b{i1,2};
%                           for i4=1:length(e1)
%                               if has(e{i4},S*3)
%                                   l=[[e1{1:i4}],S,[e1{i4+2:length(e1)}]];
%                                   CC1(i1,1)=sum(l);
%                               end
%                           end
%
%                           for i4=1:length(e2)
%                               if has(e{i4},S*3)
%                                   l=[[e{1:i4}],S,[e{i4+2:length(e)}]];
%                                   CC1(i1,1)=sum(l);
%                               end
%                           end
%
%
%                       elseif has(sum([b{i1,2}{:}]),S) && not(has(sum([b{i1,1}{:}]),S))
%                           %% Adjust previous term and one current
%                           S=S/2; %adjusts current term
%                           %now adjust previous term
%                           e=b{i1,2};
%                           for i4=1:length(e)
%                               if has(e{i4},S*2)
%                                   l=[[e{1:i4}],S/2,[e{i4+2:length(e)}]];
%                                   CC1(i1,1)=sum(l);
%                               end
```

```
%                          end
%
%
%
%                      end
%                  end
%                  f=f+S;
%              end
%
%          end
%
%          %CC(i1,i2)=simplify((f)/x);
%          CC1(i1,i2)=((f));
%      end
% end
% CC1=simplify(CC1)
% % CC
%
% trunc=children(CC1)
% %[trunc{1,1}{:}]
%
% eqm1=eqm1-sum([trunc{1,1}{:}])-sum([trunc{1,2}{:}])-sum([trunc{1,3}{:}])
% eqm2=eqm2-sum([trunc{2,1}{:}])-sum([trunc{2,2}{:}])-sum([trunc{2,3}{:}])
% eqm3=eqm3-sum([trunc{3,1}{:}])-sum([trunc{3,2}{:}])-sum([trunc{3,3}{:}])
```