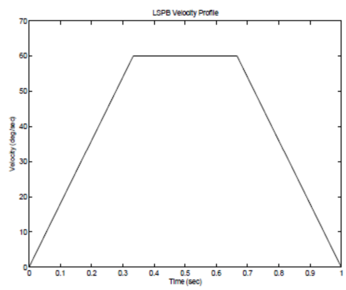
Introduction to Robotics Final Exam

Ian Sinclair
ENGR 4730-1

April 12, 2022
Dr. Kimon Valavanis

Problem 1:

Suppose we desire a joint space trajectory $\frac{dq_i^d(t)}{dt}$ for the i'th revolution joint hat begins at rest at position q_0 at time t_0 and reaches position q_1 in 2 seconds with a final velocity of $1 rad/sec$. Compute a cubic polynomial satisfying these constraints and sketch the trajectory as a function of time. Continuing, compute a LSPB (Linear Segments with Parabolic Blends) trajectory to satisfy the same requirements. Sketch the resulting position, velocity, and acceleration profiles. HINT: The LSPB trajectory has a trapezoidal velocity profile, and it is suitable when a constant velocity is desired along a path. See a general velocity profile below.]



Consider a joint system with joint space trajectory, $\frac{dq_i^d(t)}{dt}$, that begins at rest at point q_0 at time t_0.
Then, after 2 seconds reaches point q_1 with a final velocity of $1 rad/sec$.
Desired is a smooth cubic polynomial capable of representing the trajectory of the joint and satisfying the above constraints.
As a result, let $q(t)$ be such a polynomial in the form,

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Then, at the start and end points $q(t)$ takes the form,

$$\begin{aligned} q_0 &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \\ \dot{q}_0 &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 \\ q_1 &= a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 \\ \dot{q}_1 &= a_1 + 2a_2 t_1 + 3a_3 t_1^2 \end{aligned}$$

Where t_1 is the time the joint reaches position p_1.
And has equivalent matrix form,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_1 \\ \dot{q}_1 \end{bmatrix}$$

From the trajectory constraints, because the joint must move from q_0 to q_1 in 2 seconds, it must be that $q_1 = q_0 + 2$.
Additionally, the joint starts at rest; and so, $\dot{q_0} = 0rad/sec$.
Lastly, because the joint must reach q_1 at $1rad/sec$, it follows, $\dot{q}_1 = 1ad/sec$.
Therefore,

$$\begin{aligned}
q_0 &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \\
0 &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 \\
q_1 &= a_0 + a_1(t_0+2) + a_2(t_0+2)^2 + a_3(t_0+2)^3 \\
1 &= a_1 + 2a_2(t_0+2) + 3a_3(t_0+2)^2
\end{aligned}$$

And in matrix form,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & (t_0+2) & (t_0+2)^2 & (t_0+2)^3 \\ 0 & 1 & 2(t_0+2) & 3(t_0+2)^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ 0 \\ q_1 \\ 1 \end{bmatrix}$$

Next, assume that motion is counted from the start time, and is shifted; such that, $t_0 = 0seconds$. Then the matrix expression becomes,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 4 & 8 \\ 0 & 1 & 4 & 12 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ 0 \\ q_1 \\ 1 \end{bmatrix}$$

And so taking the adjoined matrix and row reducing,

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 1 & 2 & 4 & 8 & \mathbf{q_1} \\ 0 & 1 & 4 & 12 & \mathbf{1} \end{array}\right) \sim \left(\begin{array}{cccc|r} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 0 & 2 & 4 & 8 & \mathbf{q_1 - q_0} \\ 0 & 1 & 4 & 12 & \mathbf{1} \end{array}\right) \sim \left(\begin{array}{cccc|r} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 4 & 8 & \mathbf{q_1 - q_0} \\ 0 & 0 & 4 & 12 & \mathbf{1} \end{array}\right)$$

$$\sim \left(\begin{array}{cccc|r} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 4 & 8 & \mathbf{q_1 - q_0} \\ 0 & 0 & 0 & -4 & \mathbf{q_1 - q_0 - 1} \end{array}\right) \sim \left(\begin{array}{cccc|r} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 4 & 8 & \mathbf{q_1 - q_0} \\ 0 & 0 & 0 & 1 & \mathbf{\frac{1}{4}(q_0 - q_1 + 1)} \end{array}\right)$$

$$\sim \left(\begin{array}{cccc|r} 1 & 0 & 0 & 0 & \mathbf{q_0} \\ 0 & 1 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 1 & 0 & \mathbf{\frac{3}{4}q_1 - \frac{3}{4}q_0 - \frac{1}{2}} \\ 0 & 0 & 0 & 1 & \mathbf{\frac{1}{4}(q_0 - q_1 + 1)} \end{array}\right)$$

And so it follows that,

$$a_0 = q_0 \qquad a_1 = 0 \qquad a_2 = \frac{3}{4}q_1 - \frac{3}{4}q_0 - \frac{1}{2} \qquad a_3 = \frac{1}{4}(q_0 - q_1 + 1)$$

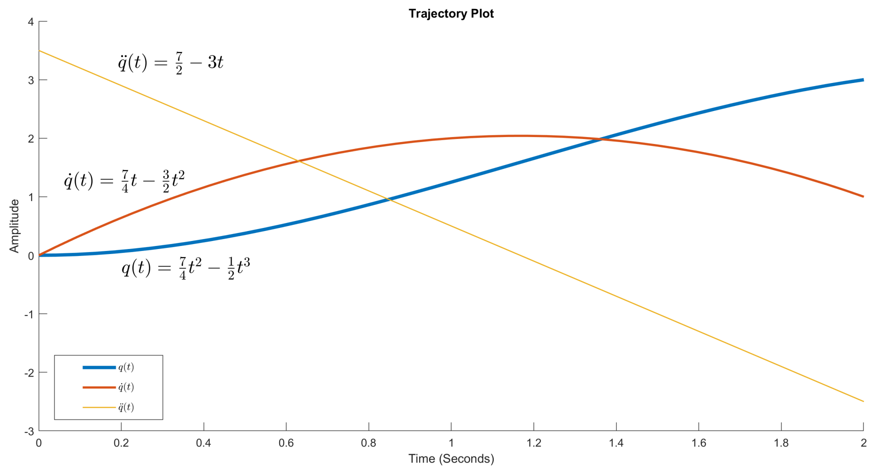
And so the trajectory polynomial $q(t)$ becomes,

$$q(t) = q_0 + \left(\frac{3}{4}q_1 - \frac{3}{4}q_0 - \frac{1}{2}\right)t^2 + \left(\frac{1}{4}(q_0 - q_1 + 1)\right)t^3$$

And now it is possible to model the trajectory, velocity and acceleration of the system by selecting an initial start and end position.
And so as an example select,

$$q_0 = 0, \qquad q_1 = 5$$

Then the resulting trajectory is modeled by,



0.1 Compute a LSPB profile to satisfy the same requirements

A different method to find a trajectory profile satisfying the above constraints is using LSPB (Linear Segments with Parabolic Blends) where the trajectory is divided into three stages, acceleration, constant velocity, and deceleration phase. Where ideally, the majority of the path follows the constant velocity (linear motion) stage. Then consider each stage separately,

i) Acceleration Phase:

Let t_a be the duration of the acceleration phase where $0 \leq t_a \leq t_1$, then within the phase, $t \in [0, t_a]$.
Here the trajectory of the joint can be modeled by a quadratic curve.

$$q_a(t) = a_{a_0} + a_{a_1} t + a_{a_2} t^2$$

Then the profile equations are,

$$\begin{aligned}
q_a(t_0) &= a_{a_0} + a_{a_1} t_0 + a_{a_2} t_0^2 \\
\dot{q}_a(t_0) &= a_{a_1} + 2a_{a_2} t_0 \\
\ddot{q}_a(t_0) &= 2a_{a_2} \\
q_a(t_a) &= a_{a_0} + a_{a_1} t_a + a_{a_2} t_a^2 \\
\dot{q}_a(t_a) &= a_{a_1} + 2a_{a_2} t_a \\
\ddot{q}_a(t_a) &= 2a_{a_2}
\end{aligned}$$

Then, it becomes critical to define the desired velocity $\dot{q}_v$ at the end of the phase; such that, $\dot{q}_a(t_a) = \dot{q}_v$.
Then, assuming null initial velocity and $t_0 = 0$, the profile for the phase becomes,

$$a_0 = q_0, \qquad a_1 = 0, \qquad a_2 = \frac{\dot{q}_v}{2t_a}$$

Which results in the curve,

$$q_a(t) = q_0 + \left(\frac{\dot{q}_v}{2t_a}\right)t^2$$

ii) Constant Velocity Phase:

After the acceleration phase, it is desired to reduce the acceleration to 0 and move for some time at constant velocity.
Correspondingly, let t_d be the duration of the final phase (deceleration phase) then the constant

velocity lasts on the range, $t \in [t_a, t_1 - t_d]$. By convention, let $t_d = t_a$. Then, the phase is on the interval, $t \in [t_a, t_1 - t_a]$.
Now let the trajectory be defined by,

$$q_c v(t) = b_0 + b_1 t$$
$$\dot{q}_c v(t) = b_1$$
$$\ddot{q}_c v(t) = 0$$

Then, to ensure smoothness, $b_1 = \dot{q}_{cv}(t) = \dot{q}_v$.
Additionally, for continuity,

$$b_0 + b_1 t_a = q_0 + \Big(\frac{\dot{q}_v}{2}\Big) t_a$$

So by inspection,

$$b_0 = q_0 - \Big(\frac{\dot{q}_v}{2}\Big) t_a$$

And so,

$$q_c v(t) = q_0 - \Big(\frac{\dot{q}_v}{2}\Big) t_a + \dot{q}_v t$$

iii) Deceleration Phase:

Again, let t_d be the duration of the deceleration phase, where $t_d = t_a$, then the phase is defined on, $t \in [t_1 - t_a, t_1]$.
Now, similarly to the acceleration phase, define the trajectory by the quadratic,

$$q_d(t) = c_0 + c_1 t + c_2 t^2$$
$$\dot{q}_d(t) = c_1 + 2c_2 t$$
$$\ddot{q}_d(t) = 2c_2$$

Then, because $q(t)$ must be smooth continuous, it follows that for starting position, $(t_1 - t_a)$ then

$$c_1 + 2c_2(t_1 - t_a) = \dot{q}_v$$

And because the end velocity must be $\dot{q}(t_1) = 1 rad/sec$, where $t_1 = 2 seconds$ it follows,

$$c_1 + 4c_2 = 1 \longrightarrow c_1 = 1 - 4c_2$$

And so,

$$1 - 4c_2 + 2c_2(2 - t_a) = \dot{q}_v \longrightarrow c_2 = -\frac{\dot{q}_v - 1}{2t_a}$$

Then,

$$c_1 = 1 + 4\frac{\dot{q}_v - 1}{2t_a}$$

And finally, by the continuity of q,

$$c_0 + (1 + 4\frac{\dot{q}_v - 1}{2t_a})(t_1 - t_a) - (\frac{\dot{q}_v - 1}{2t_a})(t_1 - t_a)^2 = q_0 - \Big(\frac{\dot{q}_v}{2}\Big) t_a + \dot{q}_v(t_1 - t_a)$$

For $t_1 = 2$ implies,

$$c_0 = q_0 + 4\dot{q}_v - t_a\dot{q}_v + t_a + \frac{-t_a^2 + 4t_a - 4t_a\dot{q}_v - 4\dot{q}_v + 4}{2t_a} - 2$$

And so the polynomial curve for the deceleration phase is,

$$q_d(t) = \Big(q_0 + 2\dot{q}_v - t_a\dot{q}_v + t_a + \frac{-t_a^2 + 3t_a - 4\dot{q}_v - 2}{2t_a} - 2\Big) + \Big(1 + \frac{4\dot{q}_v - 1}{2t_a}\Big)t - \Big(\frac{\dot{q}_v - 1}{2t_a}\Big)t^2$$

Consequently the peicewise combination of which phases results in the smooth continuous equation,

$$\begin{cases} q_0 + \left(\frac{\dot{q}_v}{2t_a}\right)t^2 & 0 \le t \le t_a, \\ q_0 - \left(\frac{\dot{q}_v}{2}\right)t_a + \dot{q}_v t & t_a \le t \le t_1 - t_a, \\ \left(q_0 + 2\dot{q}_v - t_a\dot{q}_v + t_a + \frac{-t_a^2+3t_a-4\dot{q}_v-2}{2t_a} - 2\right) + \left(1 + \frac{4\dot{q}_v-1}{2t_a}\right)t - \left(\frac{\dot{q}_v-1}{2t_a}\right)t^2 & t_1 - t_a \le t \le t_1. \end{cases}$$

Now the profile reduces the a system with 4 input parameters, the start and end locations, and the adjustable velocity $\dot{q}_v$ and phase duration, t_a.
However, by symmetry it must be that, $t_a < \frac{t_1+t_0}{2}$, and that,

$$\ddot{q}t_a = \frac{\frac{t_1+t_0}{2} - q(t_a)}{\frac{t_1}{2} - t_a}$$

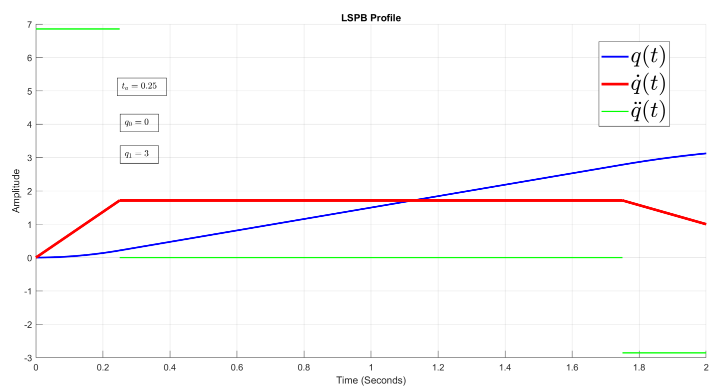
$$\ddot{q}t_a^2 - \ddot{q}t_1 t_a + (q_1 - q_0) = 0$$

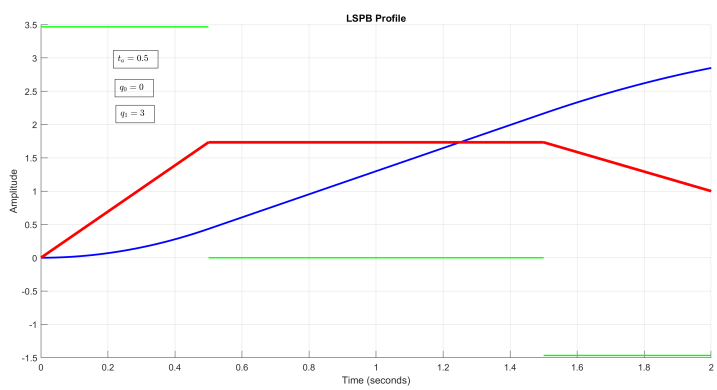
Which implies,

$$\dot{q}_v = \frac{q_1 - q_0}{t_1 - t_a}$$

And so selecting parameters values as an example,

$$q_0 = 0, \qquad q_1 = 3, \qquad t_a = 0.25 second$$





Solution

And so the cubic polynomial trajectory follows the curve,

$$q(t) = q_0 + \left(\frac{3}{4}q_1 - \frac{3}{4}q_0 - \frac{1}{2}\right)t^2 + \left(\frac{1}{4}(q_0 - q_1 + 1)\right)t^3$$

For parameters q_0, q_1.

And the LSPB profile follows the peicewise smooth curve,

$$\begin{cases} q_0 + \left(\frac{\dot{q}_v}{2t_a}\right)t^2 & 0 \leq t \leq t_a, \\ q_0 - \left(\frac{\dot{q}_v}{2}\right)t_a + \dot{q}_v t & t_a \leq t \leq t_1 - t_a, \\ \left(q_0 + 2\dot{q}_v - t_a\dot{q}_v + t_a + \frac{-t_a^2 + 3t_a - 4\dot{q}_v - 2}{2t_a} - 2\right) + \left(1 + \frac{4\dot{q}_v - 1}{2t_a}\right)t - \left(\frac{\dot{q}_v - 1}{2t_a}\right)t^2 & t_1 - t_a \leq t \leq t_1. \end{cases}$$

For parameters q_0, q_1, t_a and where $\dot{q}_v = \frac{q_1 - q_0}{t_1 - t_a}$.

Problem 2:

Trajectory Planning: Find a suitable trajectory (determine the minimum order of the polynomial) based on the following specifications: The initial/starting time is t_0, the final time is t_f with $t_f - t_0 = T = 10 seconds$; $q_i(t_0) = 10°$, $\frac{dq(t)}{dt}|_{t=t_0} = 0$, and $\frac{d^2q(t)}{dt^2}|_{t=t_0} = 0$; the corresponding conditions at t_f are, $100°, 0, 0$. Plot the position, velocity and acceleration trajectory when $t_0 = 0$ and $T = 10 sec$. (You may use MATLAB.)

Consider a trajectory with starting time t_0 and ending time t_f, where $t_f - t_0 = T = 10 seconds$.
With initial condition, $q_0 = q_i(t_0) = 10°$, and the joint starts at rest, $\frac{dq(t)}{dt}|_{t=t_0} = 0$. With null acceleration, $\frac{d^2q(t)}{dt^2}|_{t=t_0} = 0$.
Then, for the final position take,

$$q_f = q_i(t_f) = 100°, \qquad \dot{q}_f = \frac{dq(t)}{dt}|_{t=t_f} = 0, \qquad \ddot{q}_f = \frac{d^2q(t)}{dt^2}|_{t=t_f} = 0.$$

Lastly, take the starting time $t_0 = 0$ seconds.
Then there are 6 independent constraints, and so two methods to find a trajectory and optimize jerk include solve a single quintic polynomial, or adjoining two cubic polynomials, $3 - 3$. Here, the quintic option is analyzed.

Quintic Polynomial Path:

Assume that the trajectory of the joint under the constrains can be modeled by a quintic polynomial in the form,

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

Has derivatives,

$$\begin{aligned}
q(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\
\dot{q}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\
\ddot{q}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \\
\dddot{q}(t) &= 6a_3 + 24a_4 t + 60a_5 t^2
\end{aligned}$$

Then evaluating at t_0 and t_f, it follows,

$$\begin{aligned}
q(t_0) &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \\
\dot{q}(t_0) &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \\
\ddot{q}(t_0) &= 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \\
q(t_f) &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
\dot{q}(t_f) &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
\ddot{q}(t_f) &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3
\end{aligned}$$

Or in matrix form,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix}$$

Now substituting $t_0 = 0$, and $t_f = 10$, along with the position, velocity and acceleration constraints,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 10 & 100 & 1000 & 10,000 & 100,000 \\ 0 & 1 & 20 & 300 & 4,000 & 50,000 \\ 0 & 0 & 2 & 60 & 1,200 & 20,000 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 100 \\ 0 \\ 0 \end{bmatrix}$$

Has adjoined form,

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & \mathbf{10} \\ 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 2 & 0 & 0 & 0 & \mathbf{0} \\ 1 & 10 & 100 & 1000 & 10,000 & 100,000 & \mathbf{100} \\ 0 & 1 & 20 & 300 & 4,000 & 50,000 & \mathbf{0} \\ 0 & 0 & 2 & 60 & 1,200 & 20,000 & \mathbf{0} \end{array}\right)$$

Now solving numerically, using MATLAB consider the following code,

```
A = [1  0  0  0  0  0;
      0  1  0  0  0  0;
      0  0  2  0  0  0;
      1  10  100  1000  10000  100000;
      0  1  20  300  4000  50000;
      0  0  2  60  1200  20000];
B = [10;  0;  0;  100;  0;  0];
Matrix = [A B];
M = rref(Matrix);
Y = mldivide(A,B);
```

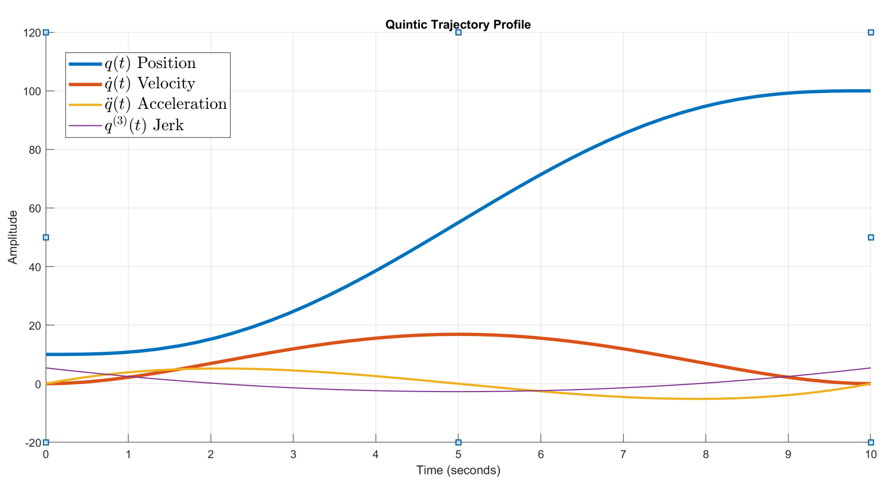
Results in coefficient values,

$$a_0 = 10, \qquad a_1 = 0, \qquad a_2 = 0, \qquad a_3 = 0.9, \qquad a_4 = -0.135, \qquad a_5 = 0.0054$$

And so the corresponding quintic trajectory polynomial is,

$$q(t) = 10 + 0.9t^3 - 0.135t^4 + 0.0054t^5$$

And so has trajectory profile,



Problem 3:

Trajectory Planning: a trajectory is assigned by specifying a sequence of desired points (via-points) without indication on the velocity in these points. For example, consider the following via-points and corresponding $q_i's$,

$t_0 = 0$	$t_1 = 2$	$t_2 = 4$	$t_3 = 8$	$t_4 = 10$
$q_0 = 10°$	$q_1 = 20°$	$q_2 = 0°$	$q_3 = 30°$	$q_4 = 40°$

In these cases, the "most suitable" values for the velocities must be automatically computed.This assignment is quite simple with heuristic rules such as zero initial and final value velocity, i.e., $\frac{dq}{dt} = 0$ at t_0 and t_f. While the other (intermediate points) the assignment is,

$$\frac{dq_k}{dt} = \begin{cases} 0 & sign(v_k) \neq sign(v_{k+1}), \\ 0.5(v_k + v_{k+1}), \qquad v_k = \frac{q_k - q_{k-1}}{t_k - t_{k-1}}, & sign(v_k) = sign(v_{k+1}) \end{cases}$$

Consider a third (3^{rd}) order polynomial trajectory and given the above restrictions/information, plot the position, velocity and acceleration trajectories using MATLAB.

Consider a trajectory following the via points,

$t_0 = 0$	$t_1 = 2$	$t_2 = 4$	$t_3 = 8$	$t_4 = 10$
$q_0 = 10°$	$q_1 = 20°$	$q_2 = 0°$	$q_3 = 30°$	$q_4 = 40°$

Under the velocity constraints,

$$\frac{dq_k}{dt} = \begin{cases} 0 & sign(v_k) \neq sign(v_{k+1}), \\ 0.5(v_k + v_{k+1}), \qquad v_k = \frac{q_k - q_{k-1}}{t_k - t_{k-1}}, & sign(v_k) = sign(v_{k+1}) \end{cases}$$

First consider a trajectory created by adjoining cubic splines where the result is differentiable smooth. In this case, each section can be represented in the form,

$$q_i(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Now because there are 3 intermediate via points, 4 cubic splines will be needed to represent the system. And so consider each independently,

i) Cubic $q_0 \to q_1$: Consider a trajectory and its derivatives defined by

$$q(t) = a_0 + a_1 t + a_2 t^2 + a3t^3$$
$$\dot{q}(t) = a_1 + 2a_2 t + 3a3t^2$$

Then, approaching the phase constrains, note it is known that $v_0 = \frac{dq_0}{dt}|_{t=t_0} = 0$, $t_0 = 0$, and $q_0 = 10°$. Additionally, $t_1 = 2$ seconds, and $q_1 = 20°$.
Lastly, note the expression for the velocity at q_1 it in terms of v_k, such that,

$$v_k = \frac{q_k - q_{k-1}}{t_k - t_{k-1}}$$

And so,

$$v_1 = \frac{q_1 - q_0}{t_1 - t_0} = \frac{20 - 10}{2 - 0} = 5 > 0$$
$$v_2 = \frac{q_2 - q_1}{t_2 - t_1} = \frac{0 - 20}{4 - 2} = -10 < 0$$

Therefore,

$$sign(v_1) \neq sign(v_2) \longrightarrow \frac{dq_1}{dt} = 0$$

And so the matrix form system of polynomials is as follows,

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_1 \\ \dot{q}_1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 4 & 8 \\ 0 & 1 & 4 & 12 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 20 \\ 0 \end{bmatrix}$$

By row reduction implies,

$$a_0 = 10, \qquad a_1 = 0, \qquad a_2 = \frac{15}{2}, \qquad a_3 = -\frac{5}{2}$$

And so the first cubic becomes,

$$q(t) = 10 + \frac{15}{2}t^2 - \frac{5}{2}t^3$$

ii) Cubic $q_1 \to q_2$: Again define a curve by trajectories,

$$q(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3$$
$$\dot{q}(t) = b_1 + 2b_2 t + 3b_3 t^2$$

Then desired is a smooth differential curve connecting the cubics, and so take $q_1 = 20°$, $t_1 = 2$ seconds and $\dot{q}_1 = 0 rad/sec$,
Additionally, let $q_2 = 0°$ and $t_2 = 4 seconds$.
Then define v_2, v_3 by a symmetric argument,

$$v_2 = -10 < 0$$
$$v_3 = \frac{q_3 - q_2}{t_3 - t_2} = \frac{30 - 0}{8 - 4} = \frac{15}{2} > 0$$

And so,

$$sign(v_2) \neq sign(v_3) \longrightarrow \frac{dq_2}{dt} = 0 rad/sec$$

Then for matrix form cubic,

$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 0 & 1 & 2t_2 & 3t_2^2 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} q_1 \\ \dot{q}_1 \\ q_2 \\ \dot{q}_2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 & 4 & 8 \\ 0 & 1 & 4 & 12 \\ 1 & 4 & 16 & 64 \\ 0 & 1 & 8 & 48 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

By row reduction implies,

$$b_0 = -80, \qquad b_1 = 120, \qquad b_2 = -45, \qquad b_3 = 5$$

And so has corresponding polynomial,

$$q(t) = -80 + 120t - 45t^2 + 5t^3$$

iii) Cubic $q_2 \to q_3$ Again take the trajectory polynomial,

$$q(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$$
$$\dot{q}(t) = c_1 + 2c_2 t + 3c_3 t^2$$

Then note that, $q_2 = 0°$, $t_2 = 4seconds$, $t_3 = 8seconds$, $q_3 = 30°$, and $\dot{q}_2 = 0rad/sec$.
And take v_3 by,

$$v_3 = \frac{15}{2} > 0$$
$$v_4 = \frac{q_4 - q_3}{t_4 - t_3} = \frac{40 - 30}{10 - 8} = 5 > 0$$

Now because,

$$sign(v_3) = sign(v_4)$$

It follows that,

$$\frac{dq_3}{dt} = 0.5(v_3 + v_4) = 0.5(\frac{15}{2} + 5) = 6.25$$

And so the cubic matrix becomes,

$$\begin{bmatrix} 1 & t_2 & t_2^2 & t_2^3 \\ 0 & 1 & 2t_2 & 3t_2^2 \\ 1 & t_3 & t_3^2 & t_3^3 \\ 0 & 1 & 2t_3 & 3t_3^2 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} q_2 \\ \dot{q}_2 \\ q_3 \\ \dot{q}_3 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 4 & 16 & 64 \\ 0 & 1 & 8 & 48 \\ 1 & 8 & 64 & 512 \\ 0 & 1 & 16 & 192 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 30 \\ 6.25 \end{bmatrix}$$

By row reduction has solution,

$$c_0 = 100, \qquad c_1 = -58.75, \qquad c_2 = 10.625, \qquad c_3 = -0.546875.$$

Which results in the polynomial,

$$q(t) = 100 - 58.75t + 10.625t^2 - 0.546875t^3$$

iv) Cubic $q_3 \to q_4$: For the last spline polynomial take the form,

$$q(t) = d_0 + d_1 t + d_2 t^2 + d_3 t^3$$

Additionally, all constraints are known, $t_3 = 8seconds$, $q_3 = 30°$, $\dot{q}_3 = 6.25$, $t_4 = 10$, $q_4 = 40°$, and $\dot{q}_4 = 0$.
Therefore, the cubic form matrix becomes,

$$\begin{bmatrix} 1 & t_3 & t_3^2 & t_3^3 \\ 0 & 1 & 2t_3 & 3t_3^2 \\ 1 & t_4 & t_4^2 & t_4^3 \\ 0 & 1 & 2t_4 & 3t_4^2 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} q_3 \\ \dot{q}_3 \\ q_4 \\ \dot{q}_4 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 8 & 64 & 512 \\ 0 & 1 & 16 & 192 \\ 1 & 10 & 100 & 1000 \\ 0 & 1 & 20 & 300 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 30 \\ 6.25 \\ 40 \\ 0 \end{bmatrix}$$

By row reduction has a solution,

$$d_0 = 540, \qquad d_1 = -193.75, \qquad d_2 = 23.75, \qquad d_3 = -0.9375$$

And has correspond polynomial,

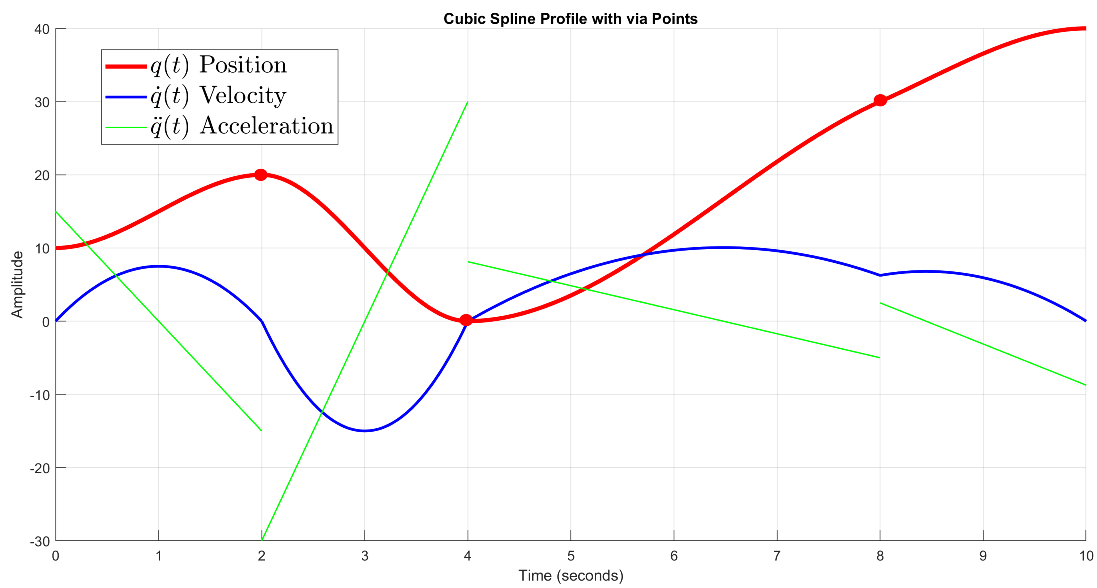
$$q(t) = 540 - 193.75t + 23.75t^2 - 0.9375t^3$$

Solution

All spline cubics are now computed and it is possible to define $q(t)$ intersecting all via points,

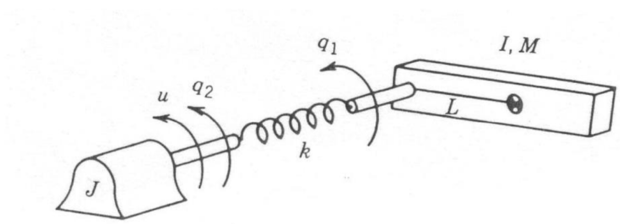
$$q(t) = \begin{cases} 10 + \frac{15}{2}t^2 - \frac{5}{2}t^3, & 0 \le t \le 2, \\ -80 + 120t - 45t^2 + 5t^3, & 2 \le t \le 4 \\ 100 - 58.75t + 10.625t^2 - 0.546875t^3, & 4 \le t \le 8, \\ 540 - 193.75t + 23.75t^2 - 0.9375t^3, & 8 \le t \le 10. \end{cases}$$

Which has a trajectory curve defined by the following figure.
Note, with cubic splines, it is difficult to create a smooth or even continuous acceleration profile, and so the trajectory is likely to be jurky.



Problem 4:

The idealized model to represent robot joint flexibility is shown below. An actuator is connected to a load through a torsional spring representing the joint flexibility. The input, for simplicity, is the motor torque u The stiffness constant k represents the effective torsional stiffness of the harmonic drive. Note that $q_1 = \theta_l$ and $q_2 = \theta_m$, $J_1 = I$, and $J_m = J$ are the load and motor inertias, B_l and B_m are the load and motor damping constants.



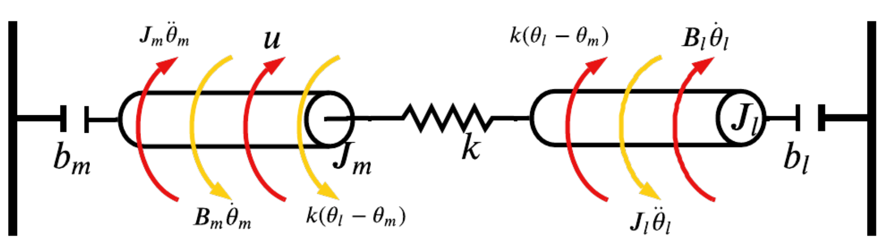
Derive the equations of motion. Find the transfer function with input u and output $q_1 = \theta_l$, $\frac{\Theta_l(s)}{U(s)}$ and give a block diagram of the system. Consider $J_1 = I = 10$, $J_m = J = 2$, $B_1 = 1$, $B_m = 0.5$, and $k = 100$. Sketch the open loop poles of the transfer function. Then apply a PD compensator to the system described by the obtained transfer function. You may also use motor angle feedback $q_2 = \theta_m$ for your design. Sketch the root locus for your chosen values of the PD controller and comment on the stability of the system. Justify any assumption you make.

Derive the Equations of Motion

Consider the system shown above where $q_1 = \theta_l$ is the position of the load, $q_2 = \theta_m$ it the rotation of the motor axle, and $J_1 = I$, $J_m = J$ are the load and motor inertias. And lastly, B_l, B_m are the load and motor damping constants.
Note, clockwise rotation on both q_1, q_2 is taken be positive. Additionally, note the torsional spring between the motor and load with spring constant k. Then the harmonic drive system can be modeled by the well-known spring-mass-damped network equations of motion.

$$\begin{cases} [inertia]f_1(\ddot{\theta_1}, \ddot{\theta_2}) + [friction]f_2(\dot{\theta_1}, \dot{\theta_2}) + [stiffness]f_3(\theta_1, \theta_2) = f_{ext} \\ [inertia]g_1(\ddot{\theta_2}, \ddot{\theta_1}) + [friction]g_2(\dot{\theta_2}, \dot{\theta_1}) + [stiffness]g_3(\theta_2, \theta_1) = 0 \end{cases}$$

Correspondingly, take the friction and torsional stiffness to oppose the direction of motion. Additionally, note that torsional stiffness follows Hook's law, $f_i = k(\theta_i - \theta_j)$. For spring force f_i.
Lastly, take u to be the motor torque and is the input of the system. This is in the place of armature voltage, and so it is assumed that the electrical equations taking armature voltage to u is known.
And so consider the following force diagram.



now, equations of motion can be generated using Newton's second law,

$$\sum t_i = J\ddot{\theta}$$

And so it follows from inspection and the construction of the rotational axis,

$$-B_m\dot{\theta}_m + k(\theta_l - \theta_m) + u = J_m\ddot{\theta}_m$$
$$-B_l\dot{\theta}_l - k(\theta_l - \theta_m) = J_l\ddot{\theta}_l$$

And simplifying to standard form the equations of motion for the system are,

$$\begin{cases} J_m\ddot{\theta}_m + B_m\dot{\theta}_m - k(\theta_l - \theta_m) = u \\ J_l\ddot{\theta}_l + B_l\dot{\theta}_l + k(\theta_l - \theta_m) = 0 \end{cases}$$

Find the transfer function $\frac{\Theta_l(s)}{U(s)}$ with u as input and $q_1 = \theta_l$ as output.

Consider the dynamic equations of motion for the system,

$$\begin{cases} J_m\ddot{\theta}_m + B_m\dot{\theta}_m - k(\theta_l - \theta_m) = u \\ J_l\ddot{\theta}_l + B_l\dot{\theta}_l + k(\theta_l - \theta_m) = 0 \end{cases}$$

Then assuming 0 initial conditions, $q_1(0) = q_2(0) = 0$ and taking the Laplace transform,

$$\mathscr{L}\{J_m\ddot{\theta}_m + B_m\dot{\theta}_m - k(\theta_l - \theta_m)\} = \mathscr{L}\{u\}$$
$$\mathscr{L}\{J_l\ddot{\theta}_l + B_l\dot{\theta}_l + k(\theta_l - \theta_m)\} = 0$$

$$J_m s^2\Theta_m + B_m s\Theta_m - k(\Theta_l - \Theta_m) = U(s)$$
$$J_l s^2\Theta_l + B_l s\Theta_l + k(\Theta_l - \Theta_m) = 0$$

Simplifying,

$$\big(J_m s^2 + B_m s + k\big)\Theta_m(s) - k\Theta_l(s) = U(s)$$
$$\big(J_l s^2 + B_l s + k\big)\Theta_l = k\Theta_m(s)$$

And so decoupling the equations take,

$$\Theta_m(s) = \big(\frac{J_l}{k}s^2 + \frac{B_l}{k}s + 1\big)\Theta_l$$

And substituting,

$$\big(J_m s^2 + B_m s + k\big)\big(\frac{J_l}{k}s^2 + \frac{B_l}{k}s + 1\big)\Theta_l - k\Theta_l(s) = U(s)$$

Results in a transfer function,

$$\begin{aligned} \frac{\Theta_l(s)}{U(s)} &= \frac{1}{\big(J_m s^2 + B_m s + k\big)\big(\frac{J_l}{k}s^2 + \frac{B_l}{k}s + 1\big) - k} \\ &= \frac{1}{\big(\frac{J_m J_l}{k}\big)s^4 + \big(\frac{J_m B_l + J_l B_m}{k}\big)s^3 + \big(J_m + J_l + \frac{B_m B_l}{k}\big)s^2 + \big(B_m + B_l\big)s} \end{aligned}$$

And so there is a final fourth order transfer function,

$$\frac{\Theta_l(s)}{U(s)} = \frac{1}{s\Big(\big(\frac{J_m J_l}{k}\big)s^3 + \big(\frac{J_m B_l + J_l B_m}{k}\big)s^2 + \big(J_m + J_l + \frac{B_m B_l}{k}\big)s + \big(B_m + B_l\big)\Big)}$$

Give a Block Diagram of the System

Here consider motor angle state, $q_2 = \theta_m$ and refer to the original dynamic equations in the Laplace domain,

$$\big(J_m s^2 + B_m s + k\big)\Theta_m(s) = k\Theta_l(s) + U(s)$$
$$\big(J_l s^2 + B_l s + k\big)\Theta_l = k\Theta_m(s)$$

Implies,

$$\Theta_l = \frac{k}{\big(J_l s^2 + B_l s + k\big)}\Theta_m$$

And,

$$\Theta_m = \frac{1}{\big(J_m s^2 + B_m s + k\big)}\Big(k\Theta_l + U(s)\Big)$$

Which results in the block diagram,

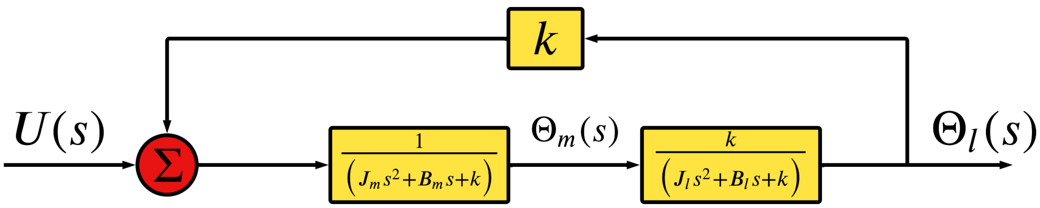


Figure 1: System block diagram

Sketch the open loop poles of the transfer function.

Again note the open loop transfer function,

$$\frac{\Theta_l(s)}{U(s)} = \frac{1}{\big(J_m s^2 + B_m s + k\big)\big(\frac{J_l}{k}s^2 + \frac{B_l}{k}s + 1\big) - k}$$

for constraints,

$$J_l = 10, \qquad J_m = 2, \qquad B_l = 1, \qquad B_m = 0.5, \qquad k = 100.$$

Then the physical system transfer function becomes,

$$\frac{\Theta_l(s)}{U(s)} = \frac{1}{s^4 + 0.35s^3 + 60s^2 + 7.5s}$$

Solving numerically with MATLAB consider the following pzmap.

```
%% Problem 4 Part 1 {Open Loop Poles}
clear

s = tf('s');

Jl = 10;
Jm = 2;
Bl = 1;
Bm = 0.5;
k = 100;

G = 1/( (Jm*s^2 + Bm*s + k)*((Jl/k)*s^2 + (Bl/k)*s + 1   ) - k);

pzmap(G)
```

Note, that the poles of the open loop transfer function under this configuration are in the negative right half $s-$plane. And so the system is initially stable, however, may still highly variant to disturbance.

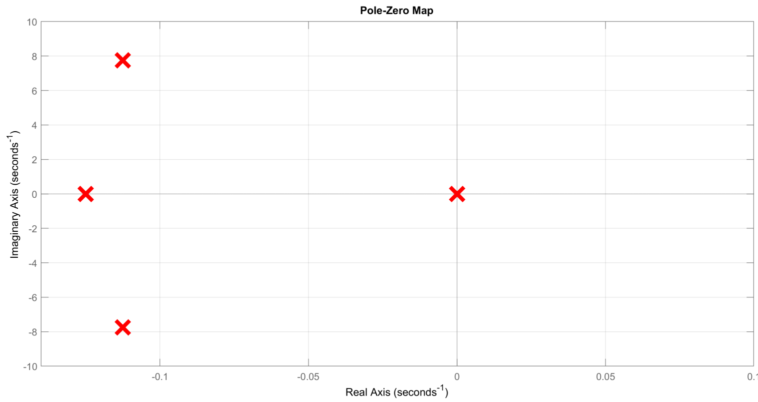


Figure 2: Pole Zero Map of open loop transfer function

Apply a PD compensator to the System Use Motor Angle Feedback And/Or Load Angle Feedback

Here a PD controller can be added to the system in the form,

$$C(s) = K_p + K_D s$$

for parameters, K_p, K_D, and error feedback input, $U(s) = C(s)\Big(r - \Theta(s)\Big)$.

Load Angle Feedback

Here, sensors are placed at the load, θ_l, and so the PD controller is feed the load output of the system. Note the following block diagram.

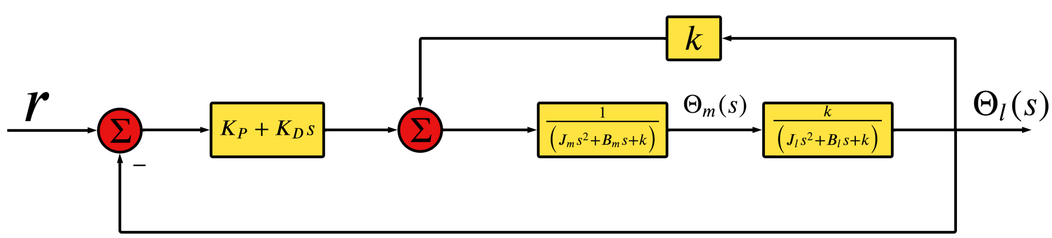


Figure 3: PD controller block diagram of Load Angle Feedback Configuration

Has open loop transfer function,

$$\frac{\Theta_l(s)}{U(s)} = \Big(K_P + K_D s\Big)\frac{1}{\big(J_m s^2 + B_m s + k\big)\big(\frac{J_l}{k}s^2 + \frac{B_l}{k}s + 1\big) - k}$$

And now tuning K_P, K_D to optimize the stability and convergence rate of the closed loop transfer function consider the following.

$$K_P = 1.374, \qquad K_D = 0.6601$$

Which results in step response,

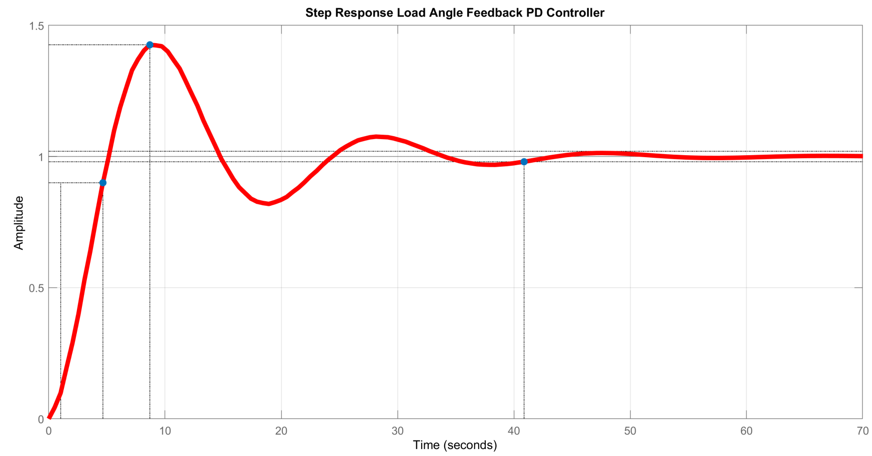


Figure 4: Step Response of closed loop transfer function with Load Angle Feedback PD controller.

Note, here the response time is very slow and has significant overshoot.
Also, approaching the stability of the system, consider the root locus plot using K_D as a gain parameter.

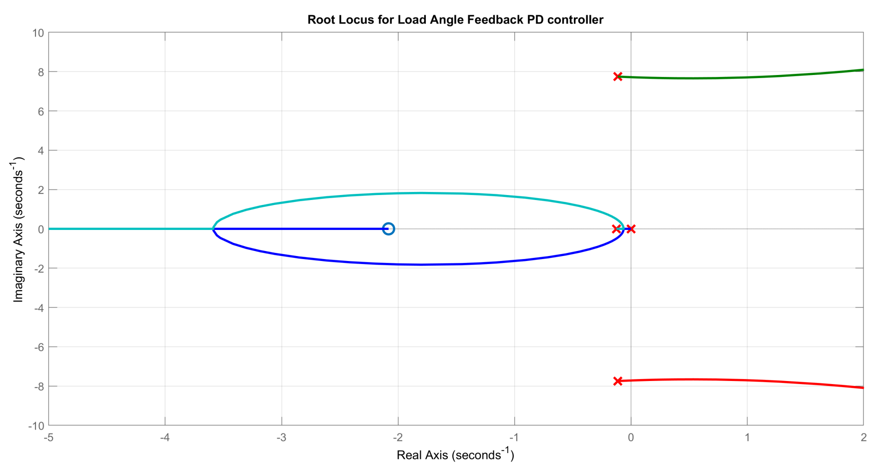


Figure 5: Root Locus for closed loop load angle feedback PD controller

And so note for large K_D gain the system because unstable, and so would need to be bounded and is less robust to disturbance.

```
G = 1/( (Jm*s^2 + Bm*s + k)*((Jl/k)*s^2 + (Bl/k)*s + 1  ) - k);

Kp = 1.374;
Ki = 0;
Kd = 0.6601;

C = pid(Kp,Ki,Kd,0);

step(feedback(C*G,1));
rlocus(C*G);
```

Motor Angle Feedback Controller

Because the above system configuration is unstable at high gain values (and thereby could be highly variant to un-modeled disturbance), a motor angle feedback system is considered.
Here, note the modified block diagram, where $q_1 = \Theta_m$ is fed back into the input.

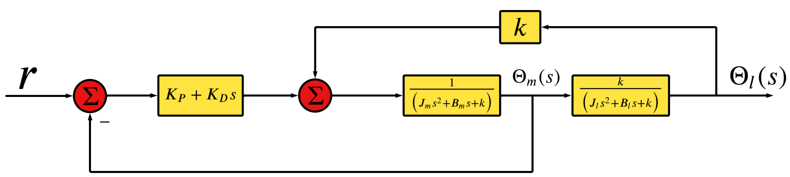


Figure 6: Block diagram: Motor Angle Feedback with PD controller.

Which admits the following SIMULINK configuration,

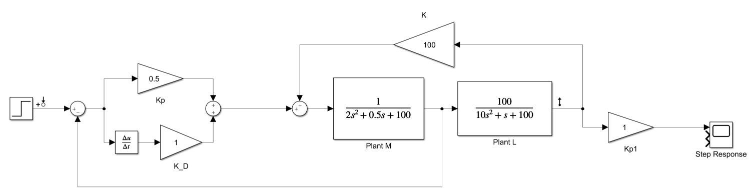


Figure 7: Motor Angle Feedback SIMULINK model with PD controller.

Now, under the parameters,

$$K_p = 5, \qquad K_D = 10$$

Has step response,

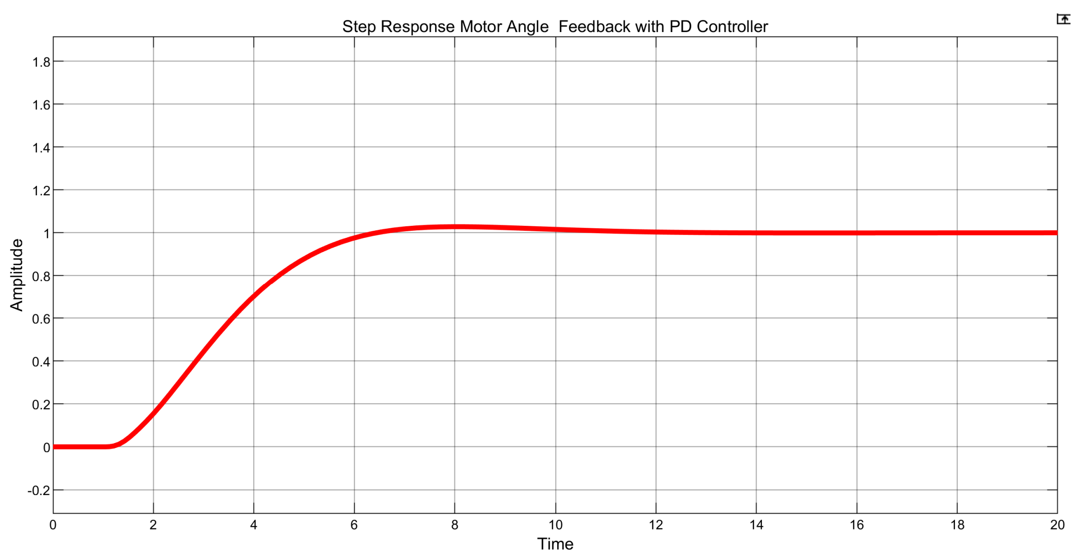


Figure 8: Step Response Motor Angle Feedback with PD Controller

```
%% Problem 4 Part 2 {PD Controller}
clear

mdl = 'RoboticsFinalExamProblem4MotorAngleFeedbackController';
```

```
open_system(mdl)

io(1) = linio('RoboticsFinalExamProblem4MotorAngleFeedbackController/Out 1',1,'input');

io(2) = linio('RoboticsFinalExamProblem4MotorAngleFeedbackController/Plant L',1,'openoutput');

linsys1 = linearize(mdl,io);

Kp = 0.5;
Ki = 0;
Kd = 1;

C = pid(Kp,Ki,Kd,0);

step(feedback(C*G*plant,(1/plant)));
rlocus(C*G);
```

And has root locus plot,

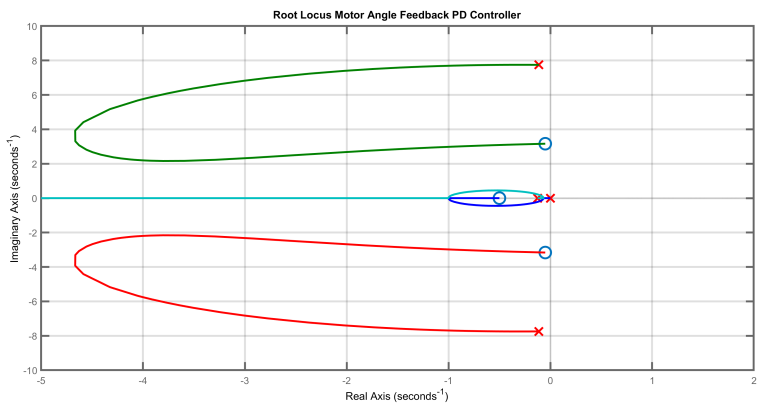


Figure 9: Root Locus of Motor Angle Feedback with PD controller.

Note, here the system is stable for all gain values, K_D. However, the poles are close to 0, and so the system may have undesired oscillations or become unstable to small disturbances not modeled by the plant.

Linear Quadratic Regulator (LQR)

Note, both PD feedback configurations had significant limitations for stability under disturbance, and so a state feedback system with a linear quadratic regulator (LQR) is consider.

State Space Form

First consider the state space form of the system,

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

for state vector x.
Then by inspection of the equations of motion,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_l \\ \dot{\theta}_l \\ \theta_m \\ \dot{\theta}_m \end{bmatrix}, \qquad \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_l \\ \ddot{\theta}_l \\ \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_l} & -\frac{B_l}{J_l} & \frac{k}{J_l} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & -\frac{k}{J_m} & \frac{B_m}{J_m} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_m} \end{bmatrix}, \qquad C = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad D = 0$$

State Feedback

Here the state variables are fed back into the input, and so rather than using either θ_l or θ_m are feedback parameters, this configuration combines both.

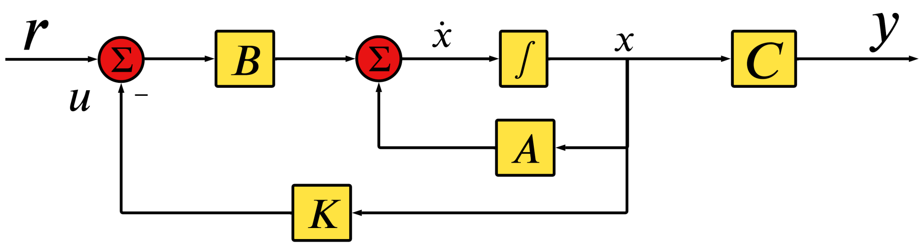


Figure 10: State feedback block diagram

Where K is a feedback transfer function.
Now approaching the validity of the state feedback model for the given system consider the theorem,

Controllability and State Feedback

> Let $\alpha(x) = s^n + \alpha_{n-1}s^{n-1} + ... + \alpha_1 s + \alpha_0$ be an arbitrary polynomial of degree n with real coefficients. THen there exists a state feedback control law of the form $-k^T x + u_r$ such that,
>
> $$\det(sI - A + BK^T) = \alpha(s)$$
>
> if and only if the corresponding system is controllable.

Or that, if the system is fully controllable then the reachable space spans and it is possible to achieve any desired pole location placement.
Correspondingly consider the controllablity matrix,

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}$$

For an $n \times n$ A matrix.
Then for this system,

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & 5.0000 \\ 0 & 0 & 5.0000 & 0.7500 \\ 0 & 0.5000 & 0.1250 & -24.9688 \\ 0.5000 & 0.1250 & -24.9688 & -12.4922 \end{bmatrix}$$

And so,

$$Rank(\mathcal{C}) = 4$$

Which is full rank, and so the system is fully controllable. Which as a result, means that a state feedback controller can be used to control the system to whatever behavior is desired.
As a result, it is useful to define a cost function to dictate optimal behavior.

Linear Quadratic Regulator Cost (LQR)

Define the cost as the performance criteria,

$$J = \int_0^\infty \{x^T Q x + R u^2\} dt$$

With parameters Q is symmetric and positive definite, and $R > 0$. For desired input u.
Next, the optimal linear control law that minimizes this cost function is,

$$u = -K_*^T x$$

Where,

$$K_* = \frac{1}{R} B^T P$$

Where P is the unique positive definite matrix satisfying the matrix algebraic Riccati equations,

$$A^T P + PA - \frac{1}{R} PBB^T P + Q = 0$$

The solution of which depends on weighting matrix Q, R which are system specific.
Within the scope of this system, weighting matrix are initially taken to be trivial; such that, take

$$Q = C^T C, \qquad R = 1$$

Then the Riccati equation becomes,

$$A^T P + PA - PBB^T P + C^T C = 0$$

Which has a solution, P,

$$P = \begin{bmatrix} 95.1876 & 8.1152 & -91.2627 & 1.6196 \\ 8.1152 & 38.4314 & 1.9038 & 5.8461 \\ -91.2627 & 1.9038 & 92.2681 & 0.3804 \\ 1.6196 & 5.8461 & 0.3804 & 3.0108 \end{bmatrix}$$

And so it follows that the optimal feedback gain vector is,

$$K_*^T = \begin{bmatrix} 0.8098 \\ 2.9230 \\ 0.1902 \\ 1.5054 \end{bmatrix}$$

Now the system can be redefined by,

$$A_c = A - BK_* \qquad B_C = B \qquad\qquad C_C = C \qquad D_C = D$$

Which is implemented by,

```
clear
Jl = 10;
Jm = 2;
Bl = 1;
Bm = 0.5;
k = 100;

A = [0 1 0 0
     -k/Jl -Bl/Jl k/Jl 0
     0 0 0 1
     k/Jm 0 -k/Jm Bm/Jm];
B = [0 0 0 1/Jm]';

C = [1 0 0 0];

D = [0];

sys = ss(A,B,C,D);
```

```
CC = ctrb(sys);
r = rank(CC);

Q = C'*C;
R = 1;
N = 1;

[K, S, CLP] = lqr(sys,Q,R);

Ac = (A-B*K); Bc = B; Cc = C; Dc = D;

sys_lqr = ss(Ac, Bc, Cc, Dc);
```

Which has a step response of,

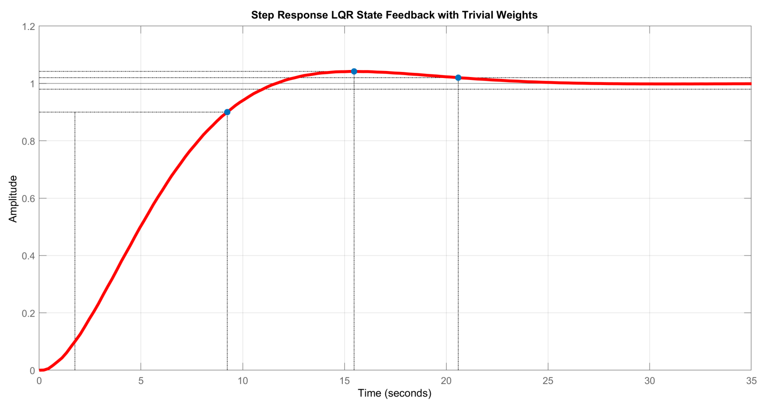


Figure 11: Step Response LQR State Feedback with Trivial Weights

Which has significantly faster settling time than the load feedback PD controller.
And has pole/ zero diagram,

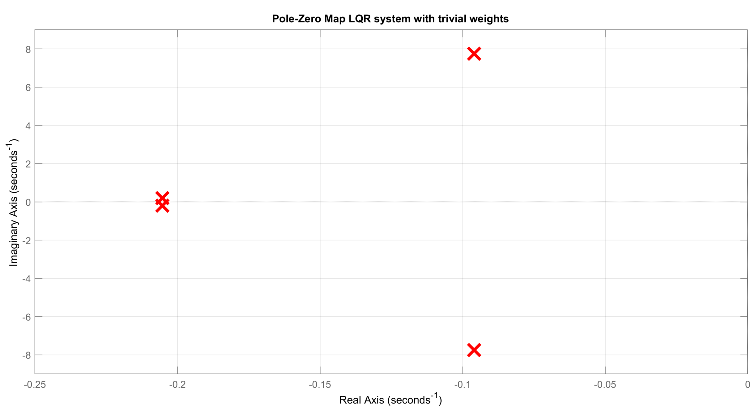


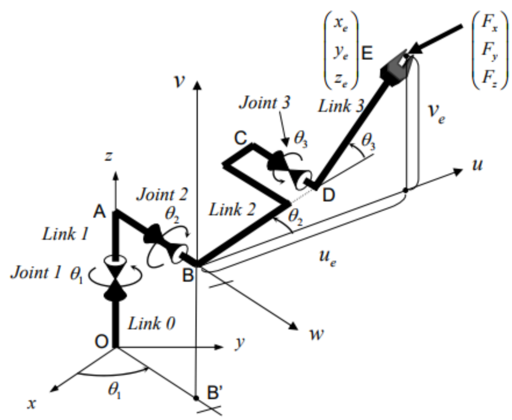
Figure 12: Pole Zero Map LQR State Feedback with Trivial Weights

Problem 5:

Shown below is a robot arm with 3 revolute joints. Coordinate system $O - xyz$, fixed to the base $link\ 0$, represents the Cartesian coordinates of the endpoints x_e, y_e, z_e. Joint angle θ_1 is measured about the joint axis OA, ($z$axis,) from the $x - axis$ to line OB', where B' is the projection of point B onto the xy plane. Another coordinate system, $B - uvw$ is placed at point B in such a way that the u and w axes are parallel to the xy plane, and that v is parallel to z. The second joint axis AB is horizontal and joint angle θ_2 is measured from axis u to line BD. Joint angle θ_3 is measured about joint axis CD, from line BD to $link3$. $i.e.$ Line DE. Link dimensions are,

$$OA = \ell_0, \qquad AB = \ell_1, \qquad BD = \ell_2, \qquad DE = \ell_3$$

Note also, the angle $OAB = ABD = 90°$.

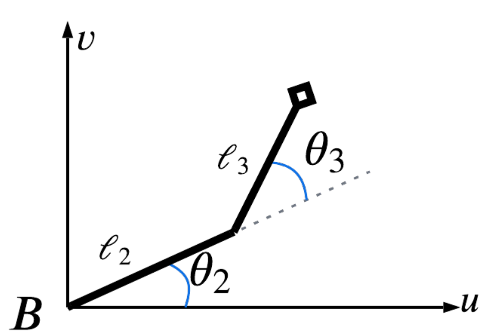


Obtain the coordinates of the end points E viewed from the frame $B - uvw$, that is, u_e, v_e in the figure.

Consider the system shown above, where the frame $B - uvw$ is originated at point B. And so the projection B' of frame B onto the xy plane is defined by θ_1 by,

$$B' = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \ell_1 \cos(\theta_1) \\ \ell_1 \sin(\theta_1) \\ 0 \end{bmatrix}$$

Next, consistent with DH parameters, let θ_3 rotate around an axis parallel to w, then, the movement of the end effector as scene from frame B is planar.
And so note that line $BD = \ell_2$ and $DE = \ell_3$, and so the coordinates of the end effector as scene from B can be found from the following,



And so geometrically is follows that,

$$\begin{bmatrix} u_e \\ v_e \\ w_e \end{bmatrix} = \begin{bmatrix} \ell_2 \cos(\theta_2) + \ell_3 \cos(\theta_2 + \theta_3) \\ \ell_2 \sin(\theta_2) + \ell_3 \sin(\theta_2 + \theta_3) \\ 0 \end{bmatrix}$$

Obtain the endpoint coordinates x_e, y_e, z_e as viewed from the base frame $O - xyz$.

First consider the transformation matrix from frame O to frame B, by inspection can defined in global reference coordinates,

$$A_B^O = Rot_x(90°)Rot_z(90° + \theta_1)trans(\ell_1 \cos(\theta_1), \ell_1 \sin(\theta_1), \ell_0)$$

$$A_B^O = \begin{bmatrix} \cos(\theta_1 + 90) & -\sin(\theta_1 + 90) & 0 & \ell_1 \cos(\theta_1 + 90)\cos(\theta_1) - \ell_1 \sin(\theta_1 + 90)\sin(\theta_1) \\ 0 & 0 & -1 & -\ell_0 \\ \sin(\theta_1 + 90) & \cos(\theta_1 + 90) & 0 & \ell_1 \cos(\theta_1 + 90)\sin(\theta_1) + \ell_1 * \sin(\theta_1 + 90)\cos(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then the position of the end effector in frame O can be found by,

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = (A_B^O)^{-1} \begin{bmatrix} u_e \\ v_e \\ w_e \\ 1 \end{bmatrix}$$

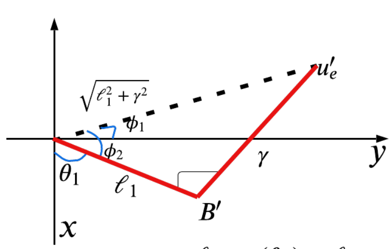
And so note the inverse of A_B^O,

$$(A_B^O)^{-1} = \begin{bmatrix} \frac{\cos(\theta_1+90)}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & 0 & \frac{\sin(\theta_1+90)}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\ell_1 \cos(\theta_1) \\ \frac{-\sin(\theta_1+90)}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & 0 & \frac{\cos(\theta_1+90)}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\ell_1 \sin(\theta_1) \\ 0 & -1 & 0 & -\ell_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And so now it follows,

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = (A_B^O)^{-1} \begin{bmatrix} u_e \\ v_e \\ w_e \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{(\cos(\theta_1+90)(\ell_3 \cos(\theta_2+\theta_3)+\ell_2 \cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)-\ell_1 \cos(\theta_1)} \\ \frac{\ell_1 \sin(\theta_1)+(\sin(\theta_1+90)(\ell_3 \cos(\theta_2+\theta_3)+\ell_2 \cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ \ell_0 + \ell_2 \sin(\theta_2) + \ell_3 \sin(\theta_2 + \theta_3) \\ 1 \end{bmatrix}$$

Which can also be shown geometrically by taking the projection of u_e onto the xy plane,



$$\gamma = \ell_2 \cos(\theta_2) + \ell_3 \cos(\theta_2 + \theta_3)$$

Here the x, y coordinate of the end effector can be found by also projecting u'_e onto the x and y axis respectively.
And so consider consider an expression of ϕ_1; by,

$$\phi_1 = atan2(\frac{\gamma}{\ell_1})$$

Then it follows that,

$$\phi_2 = \theta_1 + \phi_2 - 90^\circ = \theta_1 - 90 + atan2(\frac{\gamma}{\ell_1})$$

And so taking the x, y projection,

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\left(\theta_1 - 90 + atan2(\frac{\gamma}{\ell_1})\right)\sqrt{\ell_1^2 + \ell_2\cos(\theta_2) + \ell_3\cos(\theta_2+\theta_3)} \\ \sin\left(\theta_1 - 90 + atan2(\frac{\gamma}{\ell_1})\right)\sqrt{\ell_1^2 + \ell_2\cos(\theta_2) + \ell_3\cos(\theta_2+\theta_3)} \\ \ell_0 + \ell_2\sin(\theta_2) + \ell_3\sin(\theta_2+\theta_3) \\ 1 \end{bmatrix}$$

For,

$$\gamma = \ell_2\cos(\theta_2) + \ell_3\cos(\theta_2+\theta_3)$$

Which is an equivalent form to the end effector position equation found through coordinate transformations. However, the equation from coordinate transformation method may be simpler to compute in real time, and so it is taken and the forward kinematic of the position of the end effector.

For a given endpoint position how many solutions exist to the inverse kinematics problem? Sketch all different configurations leading to the same endpoint configuration.

Pick an endpoint position,

$$E = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$$

and assume E is reachable to all primary configurations of the robot.
Correspondingly, note that there are 4 primary robot configurations, defined here by,

i) Positive Angle Elbow Up

ii) Positive Angle Elbow Down

iii) Negative Angle Elbow Up

iv) Negative Angle Elbow Down

Or rather, any particular value θ_1 is sufficient to uniquely define the position of the B frame.
Then, joints corresponding to θ_2 and θ_3 are planar to frame B.
As a result, the intersection of this plane and the end effector location E, can be defined by a single value θ_1.
However, note that the B frame plane generated by a particular value θ_1 is co-planar to a B^* frame plane generated from $\theta_1 + 90\circ$.
And so assuming the robot's θ_1 axis can rotate 360°, for every position E, there are two, $(\theta_1, \theta_1 + 90\circ$, that generate a (90°) plane incident with both B and E for which the end effector can reach the desired endpoint position.

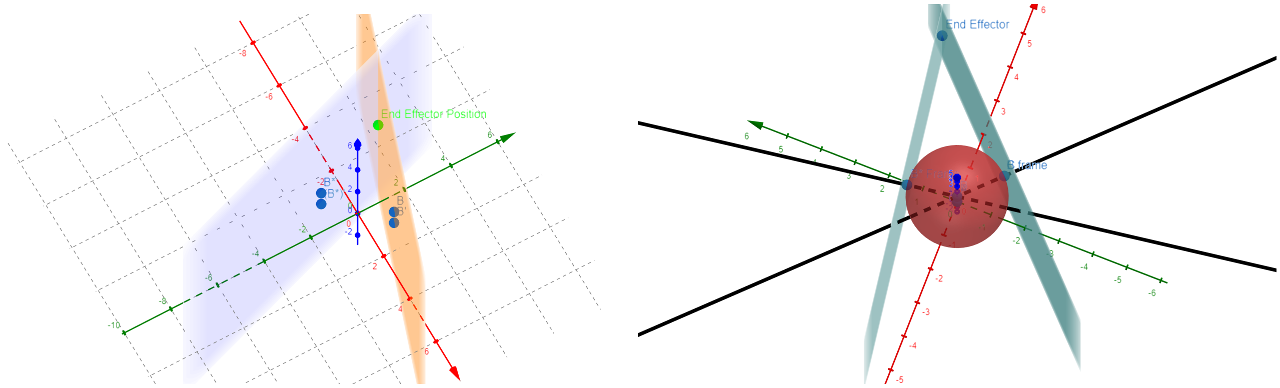


Figure 13: Image displaying two possible B frames, (B, B^*) defined by $(\theta_1, \theta_1 + 90^\circ$ that along with their projections $(B', (B^*)')$ generate a valid (90^*) plane that is incident to the desired end effector position.

Note again that the fixed angle made by $(ABD = 90^\circ)$ is all that is required to define the relationship between the two θ_1 values that generate a plane incident with the desired end effector position. Here, because $ABD = 90^\circ$ the two θ_1 values are $\theta_1, \theta_1 + 90^\circ$.

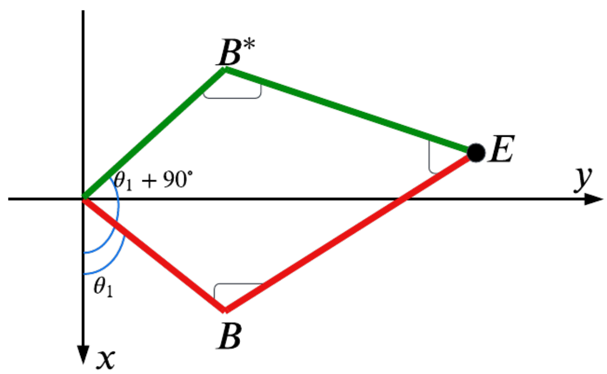
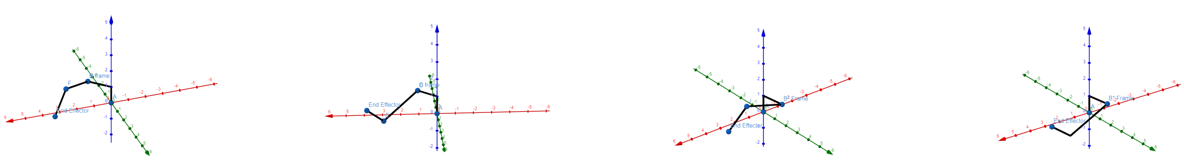
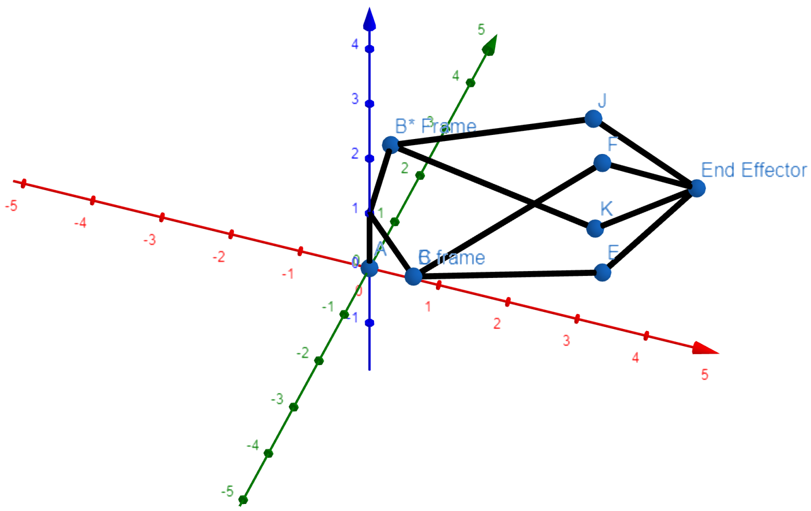


Figure 14: Top down view of robotic arms incident with E from two configurations.

It is also important to note that this is only possible if θ_2, θ_3 have complete movement $[0, 360^\circ]$ as well. Finally, note that it is well-known that a 2DOF planar revolute system has two arm configurations (elbow up and elbow down) to reach the same end effector position. As a result, there are 4 total configurations that a particular end effector position. Or if all configurations are drawn on the same plot,





Obtain the 3×3 Jacobian Matrix

First note the position of the end-effector in terms of the O frame,

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} \frac{(\cos(\theta_1+90)(\ell_3\cos(\theta_2+\theta_3)+\ell_2\cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)-\ell_1\cos(\theta_1)} \\ \frac{\ell_1\sin(\theta_1)+(\sin(\theta_1+90)(\ell_3\cos(\theta_2+\theta_3)+\ell_2\cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ \ell_0+\ell_2\sin(\theta_2)+\ell_3\sin(\theta_2+\theta_3) \\ 1 \end{bmatrix}$$

Consider the linear velocity Jacobian matrix,

$$J_v = \begin{bmatrix} \frac{\partial x}{\partial\theta_1} & \frac{\partial x}{\partial\theta_2} & \frac{\partial x}{\partial\theta_3} \\ \frac{\partial y}{\partial\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\partial y}{\partial\theta_3} \\ \frac{\partial z}{\partial\theta_1} & \frac{\partial z}{\partial\theta_2} & \frac{\partial z}{\partial\theta_3} \end{bmatrix}$$

And so the linear velocity Jacobian matrix becomes,

$$J_v = \begin{bmatrix} \ell_1\sin(\theta_1)-\frac{(\sin(\theta_1+90)(\ell_3\cos(\theta_2+\theta_3)+\ell_2\cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\frac{(\cos(\theta_1+90)(\ell_3\sin(\theta_2+\theta_3)+\ell_2\sin(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\frac{(\ell_3\cos(\theta_1+90)\sin(\theta_2+\theta_3))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ -\frac{\ell_1\cos(\theta_1)-(\cos(\theta_1+90)*(\ell_3*\cos(\theta_2+\theta_3)+\ell_2\cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & \frac{(\sin(\theta_1+90)*(\ell_3\sin(\theta_2+\theta_3)+\ell_2\sin(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & \frac{(\ell_3\sin(\theta_1+90)\sin(\theta_2+\theta_3))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ 0 & -\ell_3\cos(\theta_2+\theta_3)-\ell_2\cos(\theta_2) & -\ell_3\cos(\theta_2+\theta_3) \end{bmatrix}$$

```
jj = [theta1, theta2, theta3];
J = jacobian(endEffector,jj);
```

Forces $F_x = 10N$, $F_y = 0N$, and $F_z = 0N$ act at the endpoint, when joint angles are $\theta_1 = 0$, $\theta_2 = 45°$, and $\theta_3 = 90°$. Assume that $\ell_2 = \ell_3$. Obtain the joint torques needed for bearing the force acting at the endpoint. Discuss the physical sense of the result.

Note, for the manipulator to maintain position under the end effector force, it should exert enough torque reach a velocity that cancels the force on the end effector. As such, consider the target velocity,

$$\dot{X} = \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} = \begin{bmatrix} -10 \\ 0 \\ 0 \end{bmatrix}$$

And so, note the expression for the linear velocity at the end effector,

$$\dot{X} = J_v \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

And so,

$$\begin{bmatrix} -10 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \ell_1 \sin(\theta_1) - \frac{(\sin(\theta_1+90)(\ell_3 \cos(\theta_2+\theta_3)+\ell_2 \cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\frac{(\cos(\theta_1+90)(\ell_3 \sin(\theta_2+\theta_3)+\ell_2 \sin(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & -\frac{(\ell_3 \cos(\theta_1+90) \sin(\theta_2+\theta_3))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ -\frac{\ell_1 \cos(\theta_1)-(\cos(\theta_1+90)*(\ell_3*\cos(\theta_2+\theta_3)+\ell_2 \cos(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & \frac{(\sin(\theta_1+90)*(\ell_3 \sin(\theta_2+\theta_3)+\ell_2 \sin(\theta_2)))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} & \frac{(\ell_3 \sin(\theta_1+90) \sin(\theta_2+\theta_3))}{(\cos(\theta_1+90)^2+\sin(\theta_1+90)^2)} \\ 0 & -\ell_3 \cos(\theta_2+\theta_3) - \ell_2 \cos(\theta_2) & -\ell_3 \cos(\theta_2+\theta_3) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Or rather,

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J_v^{-1} \dot{X}$$

And so,

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 1800/((pi * 2^{1/2} * \ell_2)/2 - (\pi 2^{1/2} * \ell_3)/2) \\ (900)2^{1/2} * \ell_1)/((2^{1/2}\pi\ell_2^2)/2 - (2^{1/2}\ell + 3\pi\ell_2)/2) \\ -(1800\ell_1((2^{1/2}\ell_2)/2 - (2^{1/2} * \ell_3)/2))/((2^(1/2)\ell_2\pi\ell_3^2)/2 - (2^{1/2}\ell_2^2\pi\ell_3)/2) \end{bmatrix}$$

1 Appendix

```
clear
%% Problem 1 Part A
syms t;

q0 = 0;
q1 = 3;

a0 = q0;
a1 = 0;
a2 = (3/4)*q1-(3/4)*q0-(1/2);
a3 = (1/4)*(q0-q1+1);

qt = a0 + a1*t + a2*t^2 + a3*t^3;
vt = a1 + 2*a2*t + 3*a3*t^2;
at = 2*a2 + 6*a3*t;

hold on
fplot(qt, [0,2]);
fplot(vt, [0,2]);
fplot(at, [0,2]);

%% Problem 1 Part 2
clear
q0 = 0;
q1 = 2.6;

ta = 0.5;
qv = (q1-q0)/(2-ta);

syms t

a0 = q0;
a1 = 0;
a2 = qv/(2*ta);

b0 = q0 - (qv/2)*ta;
b1 = qv;

%c0 = q0 + 2*qv - ta*qv + ta + ((-ta^2 + 3*ta - 4qv - 2)/(2*ta)) - 2;
c0 = q0 + 4*qv - ta*qv + ta + ((-ta^2 + 4*ta - 4*ta*qv - 4*qv + 4)/(2*ta)) - 4;
c1 = 1 + 4*(qv - 1)/(2*ta);
c2 = -1*(qv-1)/(2*ta);



qat = a0 + a1*t + a2*t^2;
vat = a1 + 2*a2*t;
aat = 2*a2;

qbt = b0 + b1*t;
vbt = b1;
abt = 0;

```

```
qct = c0 + c1*t + c2*t^2;
vct = c1 + 2*c2*t;
act = 2*c2;


hold on
fplot(qat, [0,ta], 'b');
fplot(vat, [0,ta], 'r');
fplot(aat, [0,ta], 'g');

fplot(qbt, [ta,2-ta], 'b');
fplot(vbt, [ta,2-ta], 'r');
fplot(abt, [ta,2-ta], 'g');

fplot(qct, [2-ta,2], 'b');
fplot(vct, [2-ta,2], 'r');
fplot(act, [2-ta,2], 'g');

%% Problem 2
clear
A = [1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 2 0 0 0; 1 10 100 1000 10000 100000; 0 1 20 300 4000 50000; 0 0 2 60
    1200 20000];
B = [10; 0; 0; 100; 0; 0];
Matrix = [A B];
M = rref(Matrix);
Y = mldivide(A,B);

q0 = 10;
qf = 100;

t0 = 0;

tf = 10;

a0 = Y(1);
a1 = Y(2);
a2 = Y(3);
a3 = Y(4);
a4 = Y(5);
a5 = Y(6);

syms t

qt = a0 + a1*t + a2*t^2 + a3*t^3 + a4*t^4 + a5*t^5;
vt = a1 + 2*a2*t + 3*a3*t^2 + 4*a4*t^3 + 5*a5*t^4;
at = 2*a2 + 6*a3*t + 12*a4*t^2 + 20*a5*t^3;
jt = 6*a3 + 24*a4*t + 60*a5*t^2;


hold on
fplot(qt, [t0,tf]);
fplot(vt, [t0,tf]);
fplot(at, [t0,tf]);
fplot(jt, [t0,tf]);
```

```
%% Problem 3
clear
syms t
t0 = 0;
t1 = 2;
t2 = 4;
t3 = 8;
t4 = 10;

a0 = 10;
a1 = 0;
a2 = 15/2;
a3 = -5/2;

b0 = -80;
b1 = 120;
b2 = -45;
b3 = 5;

c0 = 100;
c1 = -58.75;
c2 = 10.625;
c3 = -0.546875;

d0 = 540;
d1 = -193.75;
d2 = 23.75;
d3 = -0.9375;

qat = a0 + a1*t + a2*t^2 + a3*t^3;
vat = a1 + 2*a2*t + 3*a3*t^2;
aat = 2*a2 + 6*a3*t;

qbt = b0 + b1*t + b2*t^2 + b3*t^3;
vbt = b1 + 2*b2*t + 3*b3*t^2;
abt = 2*b2 + 6*b3*t;

qct = c0 + c1*t + c2*t^2 + c3*t^3;
vct = c1 + 2*c2*t + 3*c3*t^2;
act = 2*c2 + 6*c3*t;

qdt = d0 + d1*t + d2*t^2 + d3*t^3;
vdt = d1 + 2*d2*t + 3*d3*t^2;
adt = 2*d2 + 6*d3*t;

hold on
fplot(qat, [t0,t1], 'r');
fplot(vat, [t0,t1], 'b');
fplot(aat, [t0,t1], 'g');

fplot(qbt, [t1,t2], 'r');
fplot(vbt, [t1,t2], 'b');
fplot(abt, [t1,t2], 'g');

fplot(qct, [t2,t3], 'r');
```

```
fplot(vct, [t2,t3], 'b');
fplot(act, [t2,t3], 'g');

fplot(qdt, [t3,t4], 'r');
fplot(vdt, [t3,t4], 'b');
fplot(adt, [t3,t4], 'g');


%% Problem 4 Part 1 {Open Loop Poles}
clear

s = tf('s');

Jl = 10;
Jm = 2;
Bl = 1;
Bm = 0.5;
k = 100;

plant = 1/((Jl/k)*s^2 + (Bl/k)*s + 1);

G = ((Jl/k)*s^2 + (Bl/k)*s + 1)/((Jm*s^2 + Bm*s + k)*((Jl/k)*s^2 + (Bl/k)*s + 1   ) - k);

Gm = 1/((Jm*s^2 + Bm*s + k)*((Jl/k)*s^2 + (Bl/k)*s + 1   ) - k);

Kp = 0.5;
Ki = 0;
Kd = 1;

C = pid(Kp,Ki,Kd,0);
% Load angle feedback
%LoadClosedLoop = feedback(C*G*plant,1);
%step(feedback(C*G*plant,1));
hold on
%step(feedback(C*G*plant,(1/plant)));

%rlocus(C*G*plant)

rlocus(Open);
%rlocus(C*G);
%pzmap(G)


%% Problem 4 Part 2 {PD Controller}
clear

mdl = 'RoboticsFinalExamProblem4MotorAngleFeedbackController';
open_system(mdl);

io(1) = linio('RoboticsFinalExamProblem4MotorAngleFeedbackController/Out 1',1,'input');

io(2) = linio('RoboticsFinalExamProblem4MotorAngleFeedbackController/Plant L',1,'openoutput');

linsys1 = linearize(mdl,io);
```

```
%% Problem 4 Part 3 {State Feedback and LQR}
clear
Jl = 10;
Jm = 2;
Bl = 1;
Bm = 0.5;
k = 100;

A = [0 1 0 0
     -k/Jl -Bl/Jl k/Jl 0
     0 0 0 1
     k/Jm 0 -k/Jm Bm/Jm];
B = [0 0 0 1/Jm]';

C = [1 0 0 0];

D = [0];

sys = ss(A,B,C,D);

CC = ctrb(sys);

r = rank(CC);

Q = C'*C;
R = 1;
N = 1;

[K, S, CLP] = lqr(sys,Q,R);
Ac = (A-B*K);
Bc = B;
Cc = C;
Dc = D;

sys_lqr = ss(Ac, Bc, Cc, Dc);

step(sys_lqr)

%pzmap(sys_lqr)




%% Problem 5 {Rotation Matrix}
clear
syms l1 l0 l2 l3 theta1 theta2 theta3

A = [1 0 0 0
     0 cosd(90) -sind(90) 0
     0 sind(90) cosd(90) 0
     0 0 0 1];
B = [cosd(90+theta1) -sind(90+theta1) 0 0
     sind(90+theta1) cosd(90+theta1) 0 0
     0 0 1 0
     0 0 0 1];
```

```
C = [1  0  0  l1*cosd(theta1)
     0  1  0  l1*sind(theta1)
     0  0  1  l0
     0  0  0   1];

T = A*B*C;



QQ = [l2*cosd(theta2) + l3*cosd(theta2 + theta3)
      l2*sind(theta2) + l3*sind(theta2 + theta3)
      0
       1];

endEffector = inv(T)*QQ;


jj = [theta1, theta2, theta3];

J = jacobian(endEffector, jj);


T = inv([J(1:3, :) ; J(5:length(J), :)])*[-10;0;0];

T = subs(T,[theta1, theta2, theta3],[0,45,90]);
```