

TECHNISCHE UNIVERSITÄT BERLIN



ROBOTICS COURSE

---

## Assignment 2

---

*Group Members:*

Boris Bubla

Anupama Rajkumar

Alladi Janesh

*Student ID:*

414016

415252

415349

14.11.2019

# 1 Forward Kinematics

## 1.1 Transformation Between Frames

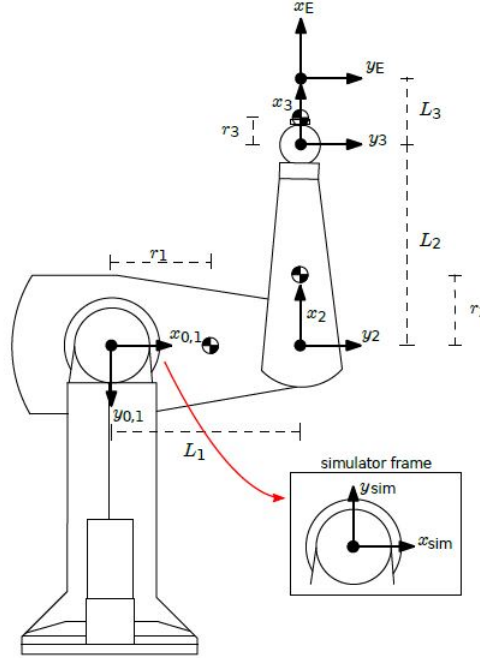


Figure 1: RRR Puma in zero configuration.

Figure 1: Puma RRR planar manipulator in zero configuration.

Table 1: A table showing the DH parameters for the given zero configuration.

	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$q_1$
2	0	$L_1$	0	$q_2 - \frac{\pi}{2}$
3	0	$L_2$	0	$q_3$
4	0	$L_3$	0	0

$$T_{i+1}^i = \begin{bmatrix} c\theta_{i+1} & -s\theta_{i+1} & 0 & a_i \\ c\alpha_i s\theta_{i+1} & c\alpha_i c\theta_{i+1} & s\alpha_i & s\alpha_i d_i \\ s\alpha_i s\theta_{i+1} & s\alpha_i c\theta_{i+1} & c\alpha_i & c\alpha_i d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The equation below shows the homogenous transform  $T_E^0$  for implementation in the PumaSim. Note that this includes a rotation of  $\pi$  about the x-axis as shown in Figure 1. This equation is

derived by multiplying adjacent transforms according to the matrix above. Ie  $T_E^0 = T_1^0 T_2^1 T_3^2 T_E^3$

$$T_E^0 = \begin{bmatrix} s_{123} & c_{123} & 0 & s_{123}L_3 + s_{12}L_2 + C_1L_1 \\ c_{123} & -s_{123} & 0 & c_{123}L_3 + c_{12}L_2 - s_1L_1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$T_1^0 = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} s_2 & c_2 & 0 & L_1 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_E^3 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

using sim frame instead of base\_0

4/5

## 1.2 End Effector Position in Operational Space

4/5

$$F(q) = \begin{bmatrix} s_{123}L_3 + s_{12}L_2 + c_1L_1 \\ c_{123}L_3 + c_{12}L_2 - s_1L_1 \\ q_1 + q_2 + q_3 \end{bmatrix}$$

(also using sim frame)missing -90 deg-1 p

## 1.3 Compute the End Effector Jacobian

$$J(q) = \begin{bmatrix} c_{123}L_3 + c_{12}L_2 - s_1L_1 & c_{123}L_3 + c_{12}L_2 & c_{123}L_3 \\ -s_{123}L_3 - s_{12}L_2 - c_1L_1 & -s_{123}L_3 - s_{12}L_2 & -s_{123}L_3 \\ 1 & 1 & 1 \end{bmatrix}$$

(cascading error with sim frame)

5/5

## 1.4 Understanding the Jacobian Matrix and Pose Singularities

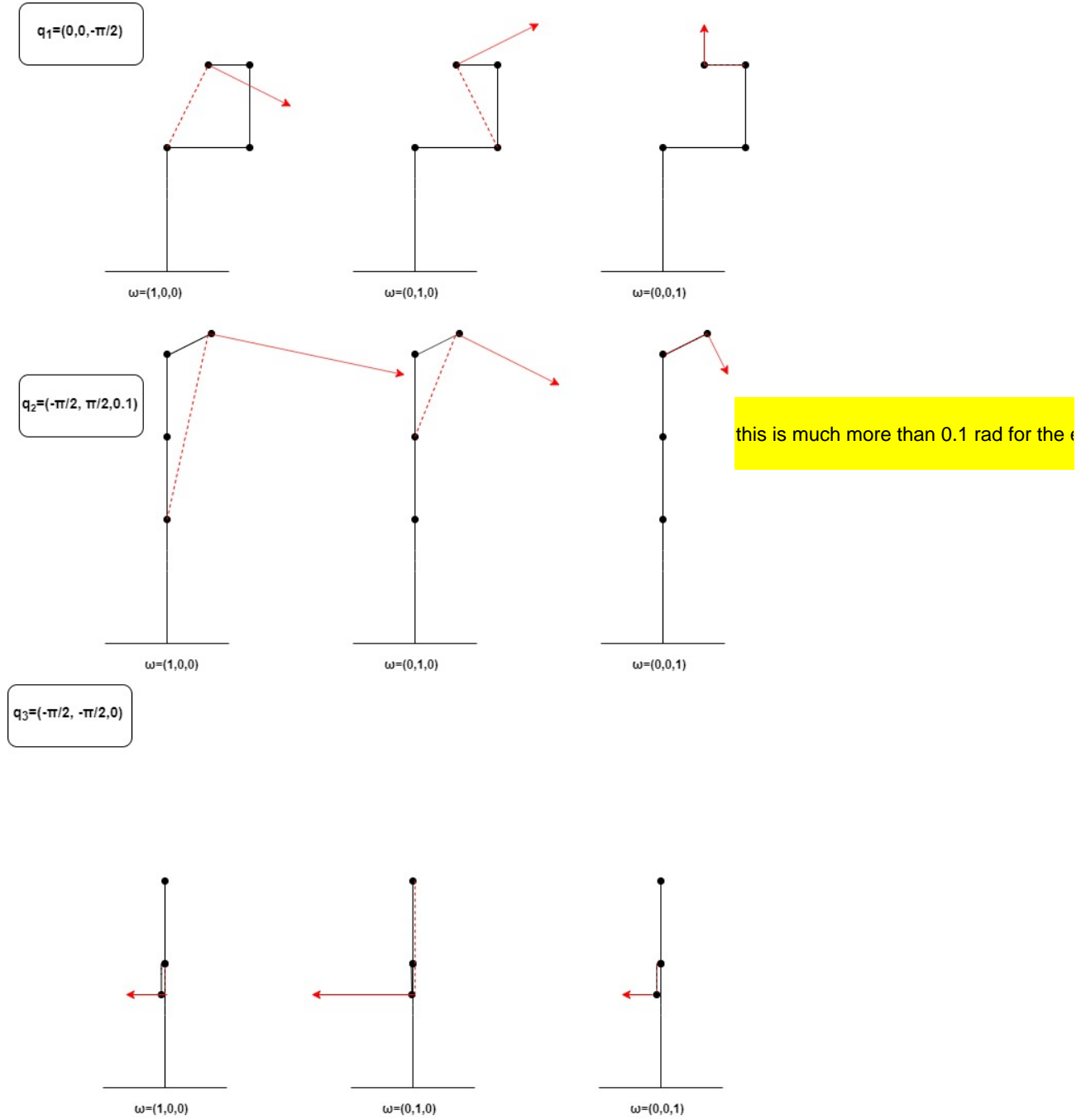


Figure 2: An image showing the three sets of velocity vectors according to the positions shown.

For  $q_1 = (0, 0, -\pi/2)$ , the configuration is not in a singularity and is movable in all directions and orientations.

For  $q_2 = (-\pi/2, \pi/2, 0.1)$ , the configuration is not in a singularity and is movable in all directions and orientations.

For  $q_3 = (-\pi/2, -\pi/2, 0)$ , the configuration is in a singularity and is only movable in the x-direction. This is evident because for every column vector applied to this configuration you can only move in the x-direction as seen from the direction of the velocity vectors in the drawings. A singularity implies that a degree is lost - in this case we cannot move in the y-direction.

q2 \_would\_ have been close to a singularity, if you had the correct pose....

4/5

## 2 Trajectory Generation in Joint Space

### 2.1 Generation of Smooth Trajectories with Polynomial Splines

$$\begin{aligned} u_1(t) &= a_0 + a_1(t_{via} - t_0) + a_2(t_{via} - t_0)^2 + a_3(t_{via} - t_0)^3 \\ u_2(t) &= b_0 + b_1(t_f - t_{via}) + b_2(t_f - t_{via})^2 + b_3(t_f - t_{via})^3 \end{aligned}$$

According to these equations, spline parameters can be calculated as:

$$\begin{aligned} a_0 &= u_0 \\ a_1 &= \dot{u}_0 \\ a_2 &= \frac{3}{t_{via}^2}(u_{via} - u_0) - \frac{2}{t_{via}}\dot{u}_0 - \frac{1}{t_{via}}\dot{u}_{via} \\ a_3 &= -\frac{2}{t_{via}^3}(u_{via} - u_0) + \frac{1}{t_{via}^2}(\dot{u}_0 + \dot{u}_{via}) \\ b_0 &= u_{via} \\ b_1 &= \dot{u}_{via} \\ b_2 &= \frac{3}{(t_f - t_{via})^2}(u_f - u_{via}) - \frac{2}{(t_f - t_{via})}\dot{u}_{via} - \frac{1}{(t_f - t_{via})}\dot{u}_f \\ b_3 &= -\frac{2}{(t_f - t_{via})^3}(u_f - u_{via}) + \frac{1}{(t_f - t_{via})^2}(\dot{u}_{via} + \dot{u}_f) \end{aligned}$$

Given:

$$\begin{aligned} t_0 &= 0 \\ t_{via} &= 2.5 \\ t_f &= 5 \\ u_0 &= (0, 0, 0)^T \\ u_{via} &= \left(-\frac{\pi}{4}, \frac{\pi}{2}, 0\right)^T \\ u_f &= \left(-\frac{\pi}{2}, \frac{\pi}{4}, 0\right)^T \\ \dot{u}_0 &= 0 \\ \dot{u}_{via} &= 0 \\ \dot{u}_f &= 0 \end{aligned}$$

use the heuristic!

Substituting these values, spline parameters are:

$$\begin{aligned} a_0 &= (0, 0, 0)^T \\ a_1 &= (0, 0, 0)^T \\ a_2 &= (-0.37, 0.75, 0)^T \\ a_3 &= (0.1, -0.2, 0)^T \\ b_0 &= (-0.78, 1.57, 0)^T \\ b_1 &= (0, 0, 0)^T \\ b_2 &= (-0.37, -0.37, 0)^T \\ b_3 &= (0.1, 0.1, 0)^T \end{aligned}$$

a2[0] = -0.25a

## 2.2 Trajectory to a Specific Point

Substituting the spline parameters calculated in 2.1, we get following trajectory equations.

From time  $t_0 - > t_{via}$

For joint 1 :

$$u_1(t) = -0.37(t_{via} - t_0)^2 + 0.1(t_{via} - t_0)^3$$

For joint 2 :

$$u_1(t) = 0.75(t_{via} - t_0)^2 - 0.2(t_{via} - t_0)^3$$

For joint 3 :

$$u_1(t) = 0$$

From time  $t_{via} - > t_f$

For joint 1 :

$$u_2(t) = -0.78 - 0.37(t_f - t_{via})^2 + 0.1(t_{via} - t_0)^3$$

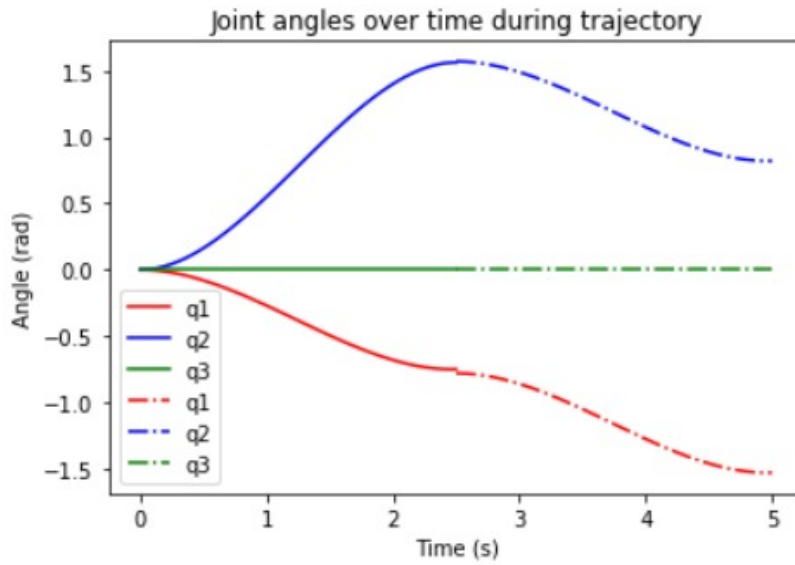
For joint 2 :

$$u_2(t) = 1.57 - 0.37(t_f - t_{via})^2 + 0.1(t_f - t_{via})^3$$

For joint 3 :

$$u_2(t) = 0$$

Plotting these equations over time, we get joint trajectory



4/4

Figure 3: Joint angles with respect to time.

## 2.3 Trajectory Generation

Total duration of trajectory  $t_f$  can be calculated using following two approaches:

1. Approach 1: Planning trajectories made of segments joint by parabolic arcs

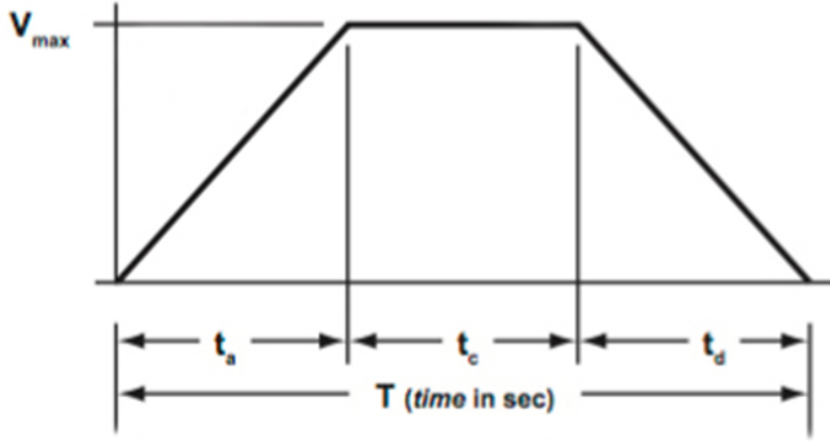


Figure 4: Segment joint by parabolic arcs.

$$\begin{aligned}
 t_0 &= \text{Currenttime} \\
 t_a &= \frac{\dot{q}_{max}}{\ddot{q}_{max}} = \frac{dq_{max}}{ddq_{max}} \\
 t_d &= t_a \\
 t_c &= \frac{\text{distance}}{\dot{q}_{max}} = \frac{(q_d - q)}{dq_{max}}
 \end{aligned}$$

Hence,

$$t_f = t_0 + t_a + t_c + t_d$$

2. Approach 2: Finding maximum value from velocity and acceleration equation

$$\dot{u} = 2a_2t + 3a_3t^2$$

$$\ddot{u} = 2a_2 + 6a_3t$$

As per constraints:

$$2a_2t + 3a_3t^2 \leq \dot{q}_{max}$$

$$2a_2 + 6a_3t \leq \ddot{q}_{max}$$

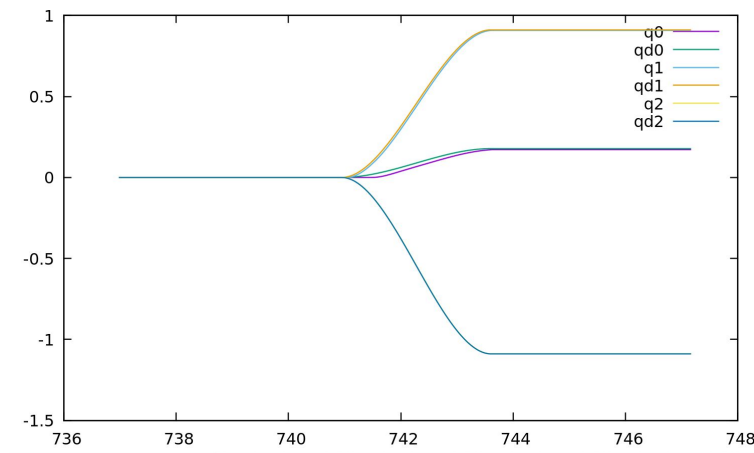


Solving for the maxima of these equations, we get

$$t_f = \max\left(\frac{3(q_f - q_0)}{\dot{q}_{max}}, \sqrt{\frac{6(q_f - q_0)}{\ddot{q}_{max}}}\right)$$

We have computed  $t_f$  using approach 1  
Trajectory of q over time

computeTf is wrong this is not what a cut



axis labels missing-1 p

Figure 5: Plot of desired and actual joint angles

Trajectory of tau over time

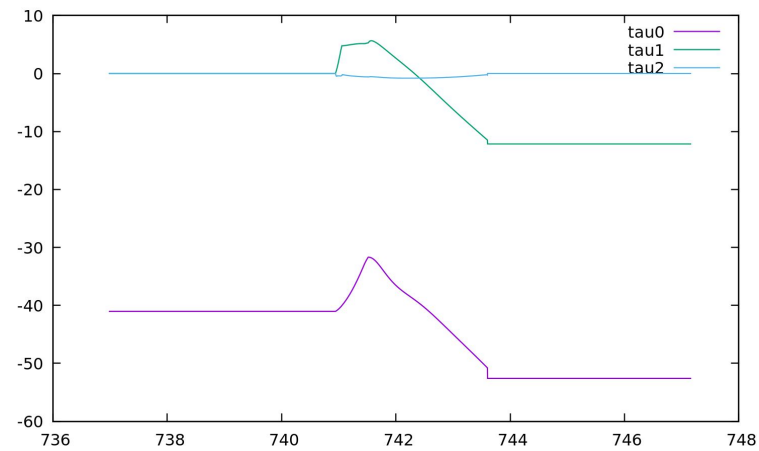


Figure 6: Plot of desired and actual joint angles

### 3 Operational Space Control

35/40

#### 3.1 proj2Control

The gains used for proj2Control are seen below.

Table 2: The gains used for proj2Control. Each  $K_p$  and  $K_v$  value have one gain for each joint: J1,J2,J3.

	<b>J1</b>	<b>J2</b>	<b>J3</b>
<b>K<sub>p</sub></b>	5000	4500	180
<b>K<sub>v</sub></b>	150	40	11

##### 3.1.1 Plots

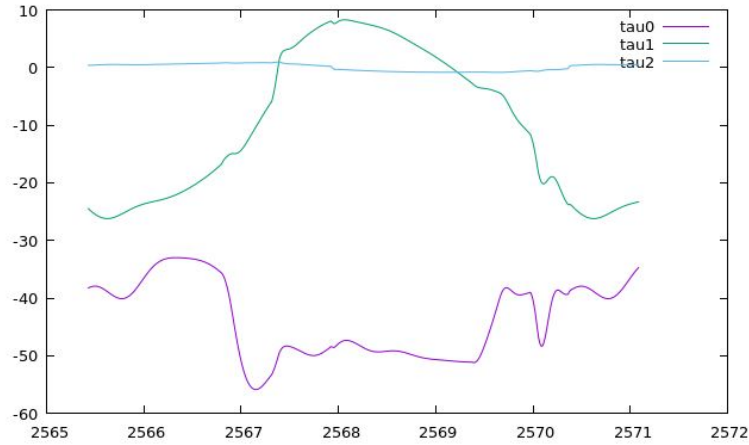


Figure 7: Plot of torques during one full circle.

axis labels missing-1 p

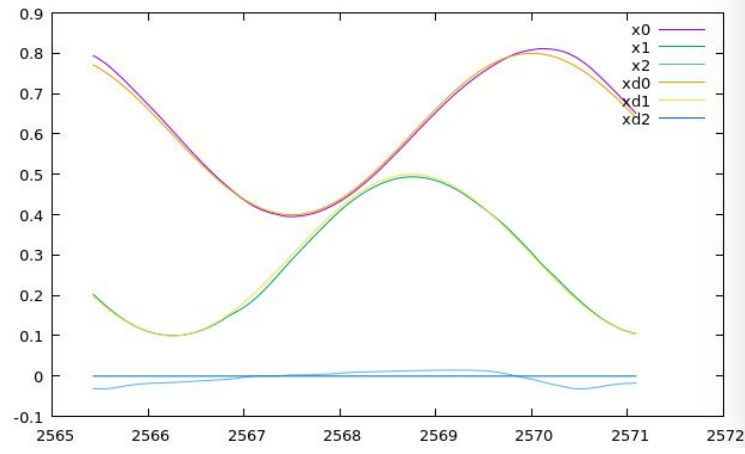


Figure 8: Plot of positions  $x,y,\theta$  as  $x_0,x_1,x_2$  and their desired positions  $dx_0,dx_1,dx_2$  during one full circle.

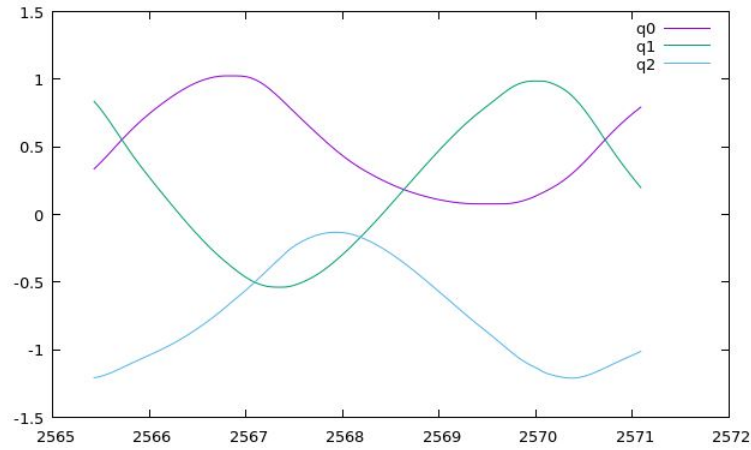


Figure 9: Plot of joint angles during one full circle.

### 3.1.2 Error Plots

The error plots of  $x,y,\theta$  are seen as error plots  $x_0,x_1,x_2$  as shown below.



Figure 10: Plot of error  $x_0$  over one full circle.

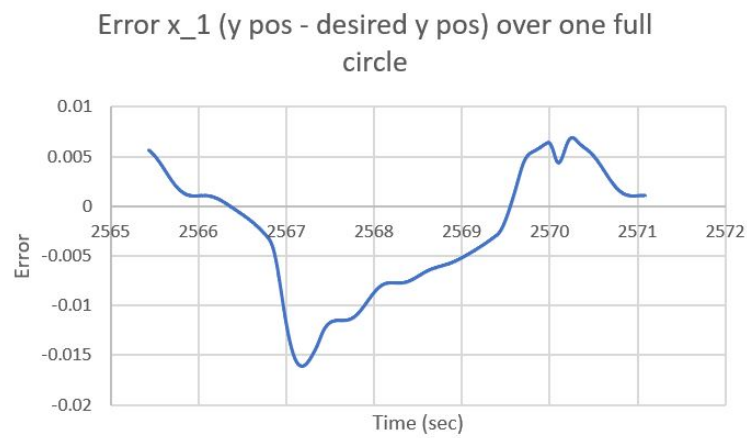


Figure 11: Plot of error  $x_1$  over one full circle.

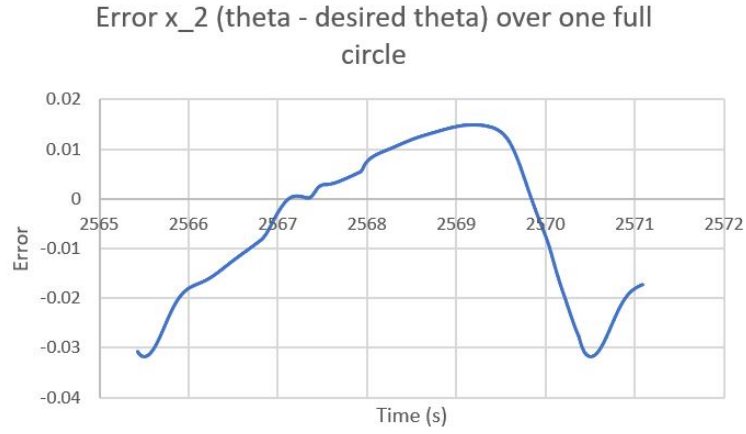


Figure 12: Plot of error x2 over one full circle.

### 3.1.3 How does the error change as $K_p$ changes?

As the  $K_p$  values decrease, the error graph will increase in magnitude. This can be seen by the increased differences in the plots of the desired positions and the actual positions, as shown below.

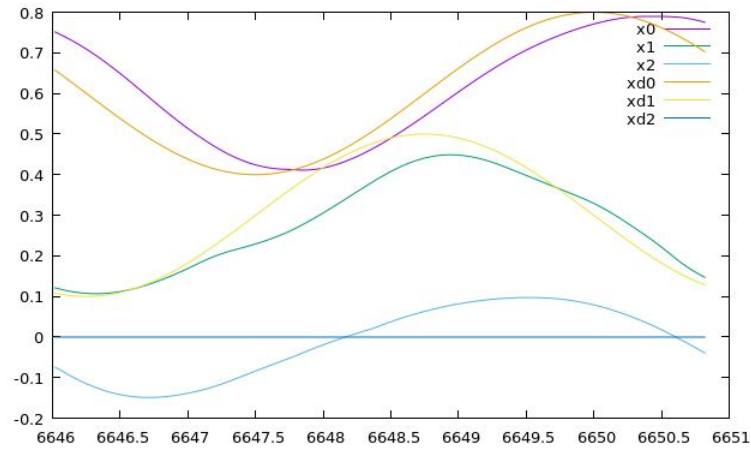


Figure 13: Plot of error with 10x reduced  $K_p$  gains of 500,450,21

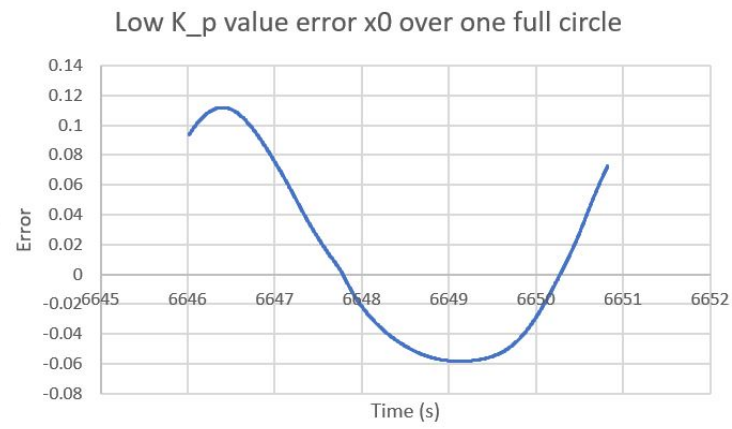


Figure 14: Plot of error with 10x reduced  $K_p$  gain of 500

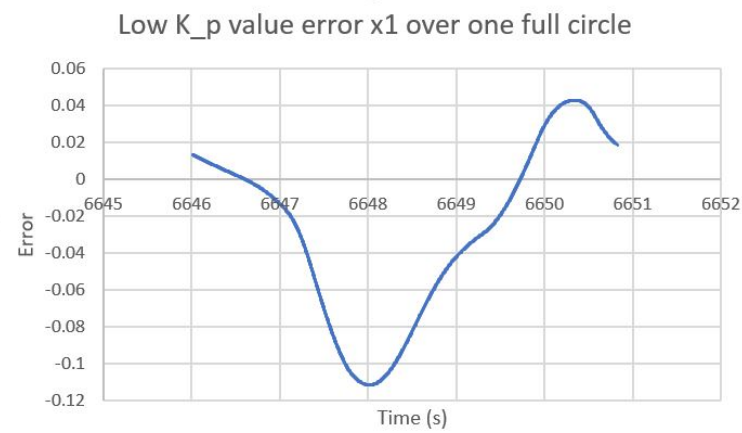


Figure 15: Plot of error with 10x reduced  $K_p$  gain of 450

where is the 2D plot of  $x$  and  $\dot{x}$ ?

24/25

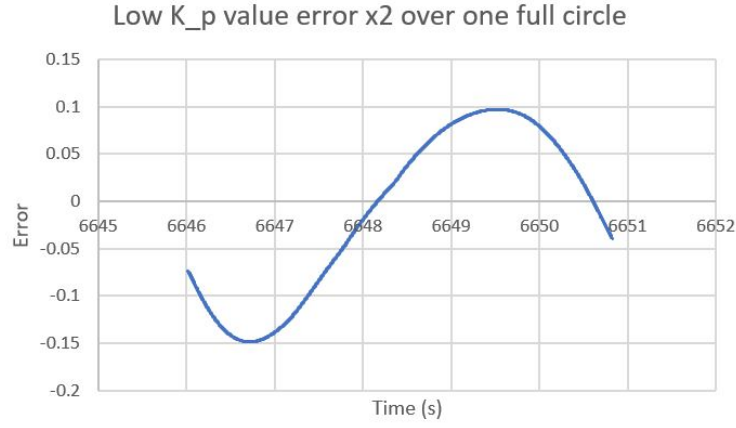


Figure 16: Plot of error with 10x reduced  $K_p$  gain of 21

### 3.2 proj3Control

The parabolic blends method involves one acceleration period ( $0 < t < t_b$ ), a constant velocity period ( $t_b < t < t_f - t_b$ ) and a deceleration period ( $t_f - t_b < t < t_f$ ).

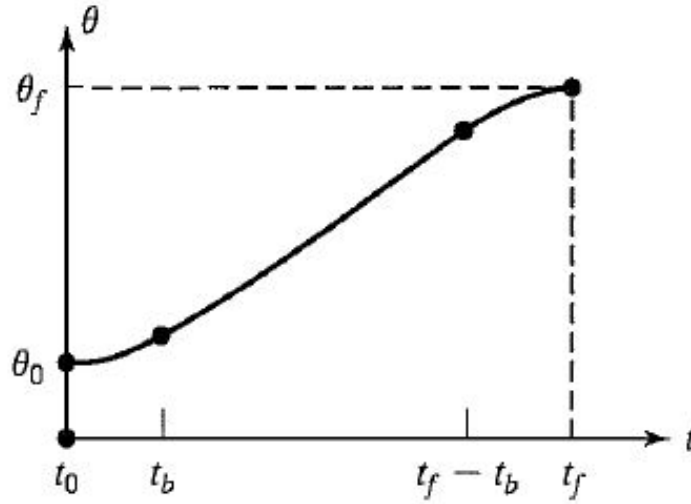


Figure 17: An image showing the different intervals in a parabolic blend.

By using the final position ( $\theta_f = 6\pi$ ) and initial position ( $\theta_0 = 0$ ) as well as the acceleration limit ( $\ddot{\theta} = \frac{2\pi}{25}$ ), one can solve for  $t_f$  as seen below.

$$t_f^2 \geq \frac{4(\theta_f - \theta_0)}{\ddot{\theta}}$$

$t_f \geq 10\sqrt{3}$  and so we choose  $t_f = 20$ s so that we are well within the limit. We can then calculate  $t_b$ , the blend time, as shown below.

$$t_b = \frac{t_f}{2} - \frac{\sqrt{\ddot{\theta}^2 t_f^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

Yielding  $t_b = 5$ s

This allows us to find the equations describing the three intervals, as described in the equations below. Equations 1 and 2 describe the acceleration period, 3 and 4 describe the constant velocity period and 5 and 6 describe the deceleration period.

$$\theta(t) = \theta_0 + 0.5\ddot{\theta}t^2 \quad (1)$$

$$\dot{\theta}(t) = \ddot{\theta}t \quad (2)$$

$$\theta(t) = \theta_{t_b} + \dot{\theta}_{t_b} * (t - t_b) \quad (3)$$

$$\dot{\theta}(t) = \dot{\theta}_{t_b} \quad (4)$$

$$\theta(t) = \theta_{t_f - t_b} - 0.5\ddot{\theta} * (t_f - t)^2 \quad (5)$$

$$\dot{\theta}(t) = -\ddot{\theta} * (t_f - t) \quad (6)$$

We feed these rotational positions and velocities into the x,y, $\theta$  vectors as done in proj2Control.

## 4 Task Implementation Table

Table 3: Task Implementation Table

Student Name	A1	A2	A3	A4	B1	B2	C1	C2	C3	C4	C5
<b>Boris Bubla</b>	x	x	x	x			x	x	x	x	
<b>Alladi Janesh</b>											
<b>Anupama Rajkumar</b>					x	x	x	x	x	x	

plots for beta and beta\_dot missing-3

bug in code for y-position, missing radiusg

11/15