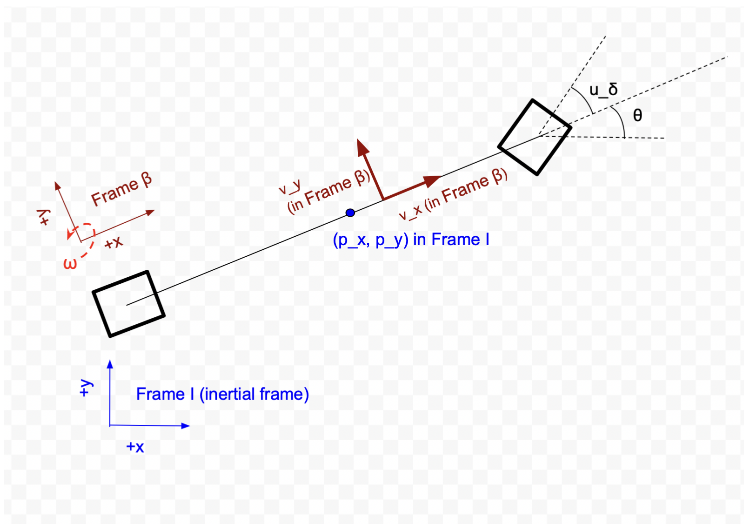
CDS110	**November 5, 2024**

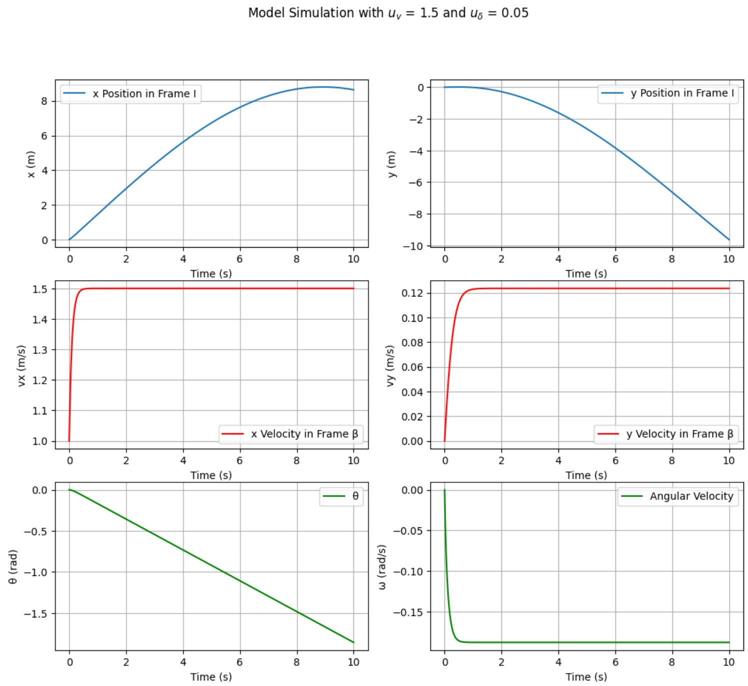
Pset 05

Problem 1

Task 0



Task 1
With nonzero control inputs $u_v = 1.5$ and $u_\delta = 0.05$ the simulation of the equations of motion is shown below.



Intuitively, as shown in the plots above, the simulated dynamics do match with the behavior of a real car. Because we have a low constant steering angle u_δ and a nonzero u_v, the car should be gradually turning. From the the plots above we we that this is the case as theta is linearly decreasing. Furthermore, since our steering input is constant, we should eventually reach a constant angular velocity (the plot as shows this). Lastly, for the first few seconds we should see the x position increasing at a decreasing rate and the y position decreasing since we are gradually turning (this agrees with the plots above) Note that if the model were to be simulated for longer, eventually the car would turn enough such that the x position begins to decrease and the y position would begin to increasing. Lastly, since our u_v is positive and constant with u_s also constant, we would eventually reach a steady state for both velocities (which the plots agree with above).

Task 2

Given $\dot{v}_x^\beta = -\frac{1}{\tau}v_x^\beta + \frac{1}{\tau}u_v$, we choose the controller $u_v = v_d^\beta - ke + \tau\dot{v}_d^\beta$ where the error $e = v_x^\beta - v_d^\beta$ and where $k > 0$. To see show convergence we use Lyapunov stability theory. We define the Lyapunov function to be $V(e) = e^2$. First, note that the Lyapunov function $V(e)$ is positive definite for all values of e except $e = 0$. Now taking the derivative of the Lyapunov function we have $\dot{V}(e) = 2e\dot{e}$. The derivation of $\dot{e}$ is shown below:

$$\dot{e} = \dot{v}_x^\beta - \dot{v}_d^\beta$$

By substitution we have:

$$\dot{e} = -\frac{1}{\tau}v_x^\beta + \frac{1}{\tau}u_v - \dot{v}_d^\beta = -\frac{1}{\tau}v_x^\beta + \frac{1}{\tau}(v_d^\beta - ke + \tau\dot{v}_d^\beta) - \dot{v}_d^\beta$$

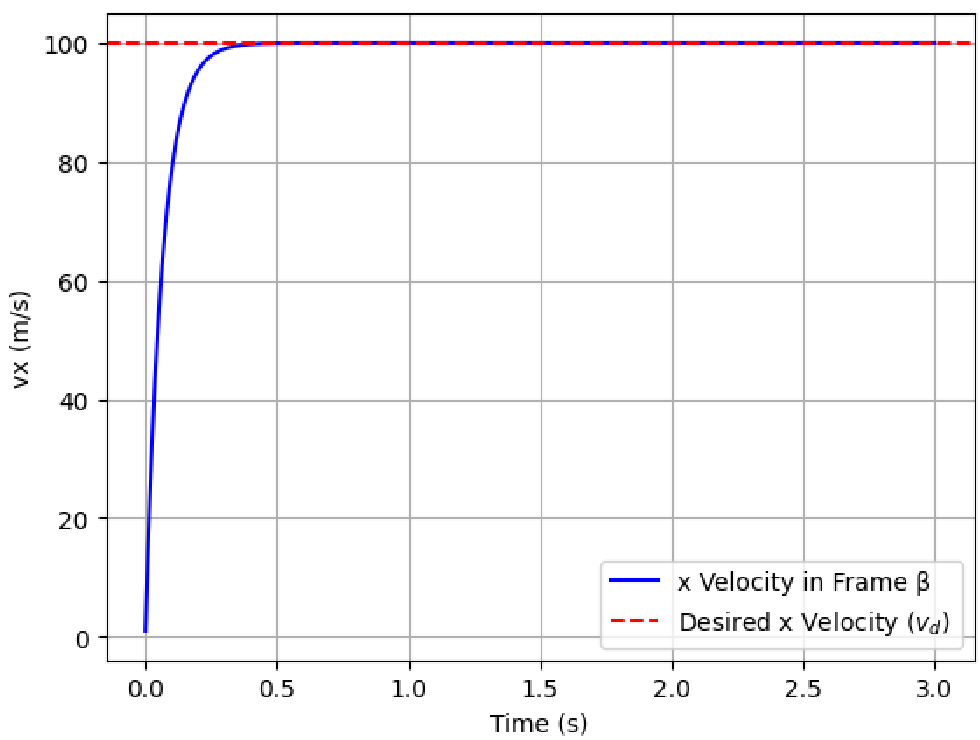
Simplifying:

$$\dot{e} = -\frac{1}{\tau}(v_x^\beta - v_d^\beta) - \frac{1}{\tau}ke + \dot{v}_d^\beta - \dot{v}_d^\beta = -\frac{1}{\tau}e - \frac{1}{\tau}ke = -e\left(\frac{1+k}{\tau}\right)$$

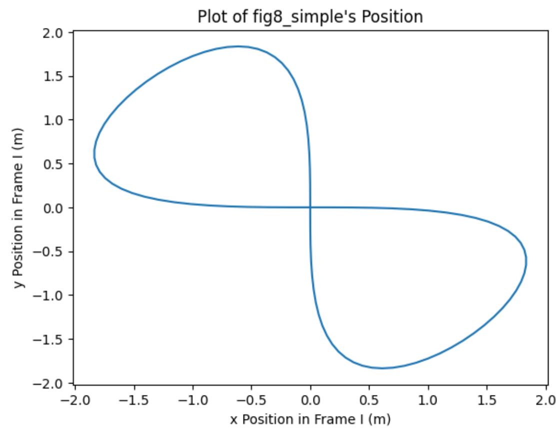
Thus we have $\dot{V}(e) = -2e^2\left(\frac{1+k}{\tau}\right)$. Therefore because both k and τ are positive then $\dot{V}(e) < 0$ for all e except $e = 0$ and at e=0 we have $\dot{V}(e) = 0$. Therefore, using Lyapunov stability theory, we have shown that the control input $u_v = v_d^\beta - ke + \tau\dot{v}_d^\beta$ yields asymptotic stability such that v_x^β converges to our desired x velocity v_d^β. Note, if $\dot{v}_d^\beta = 0$ (i.e. we are tracking a point), then our controller becomes $u_v = v_d^\beta - ke$

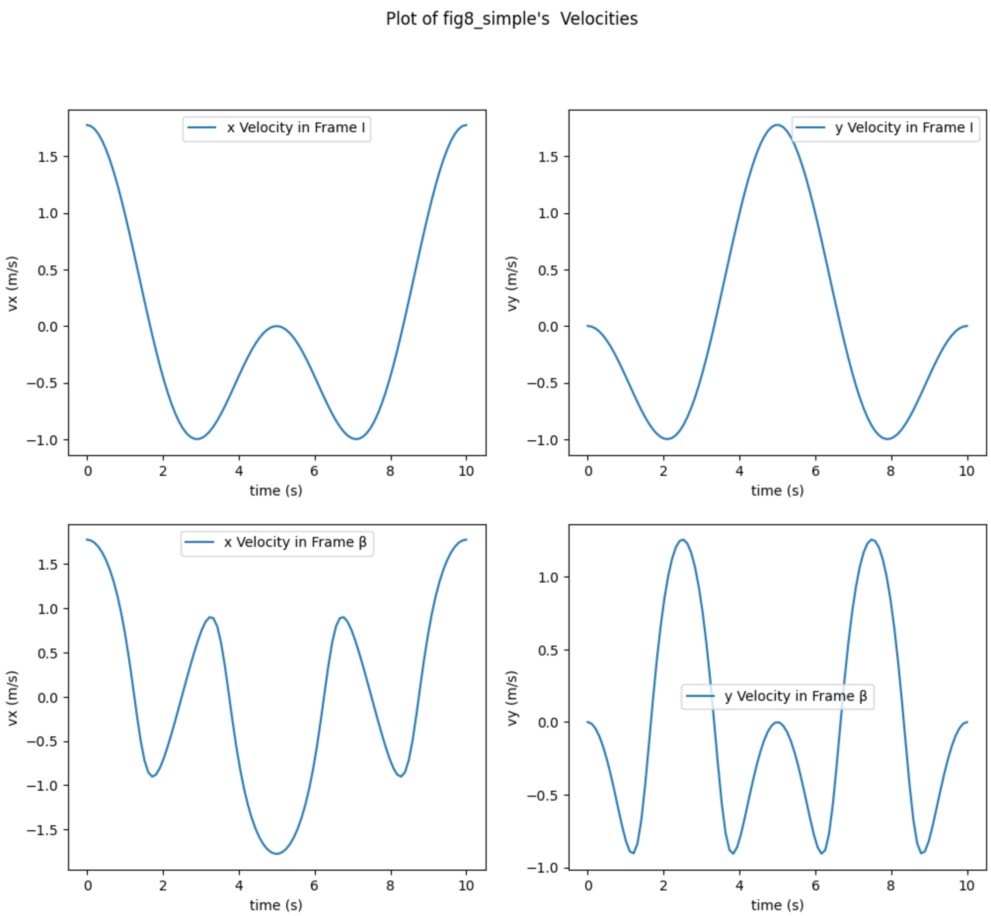
Task 3

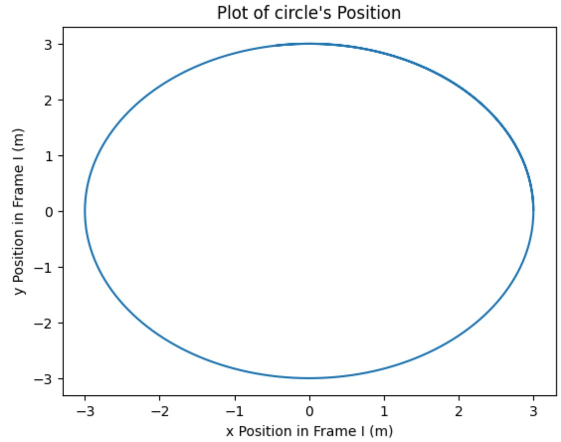
With $k = 0.5$, the plot of the controller applied to the simulation is shown below:

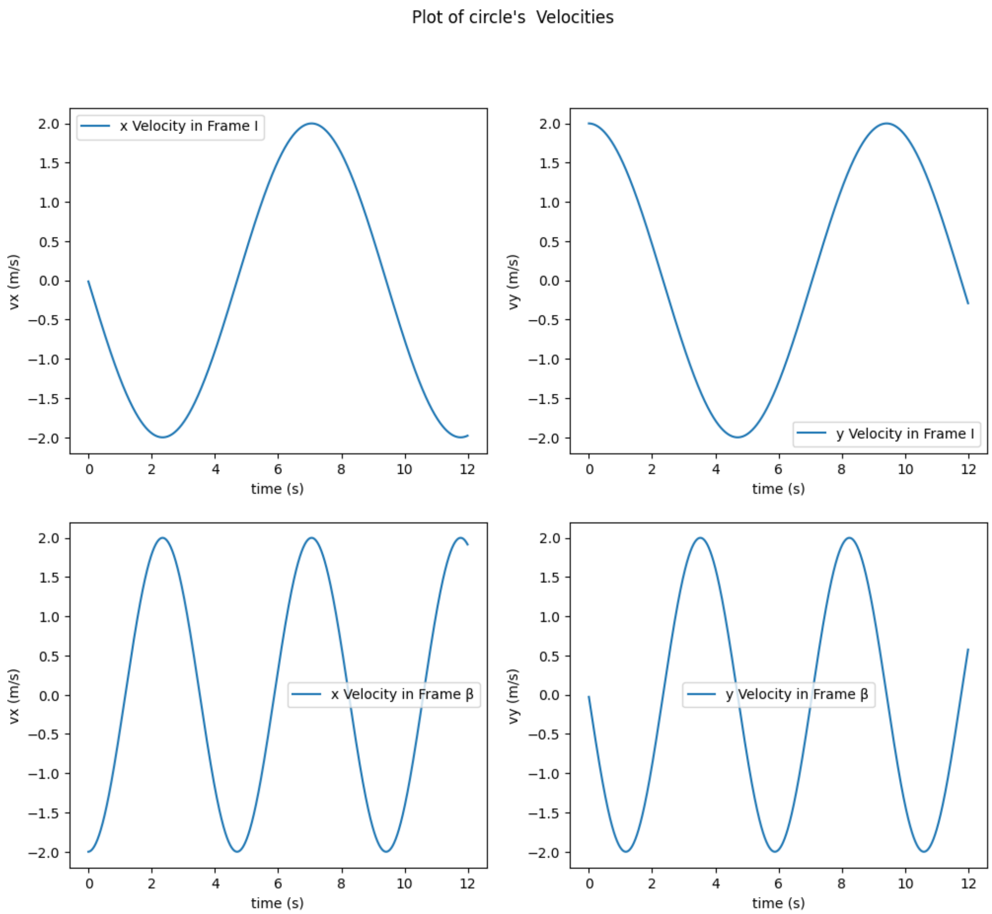


Task 4


Plot of fig8_simple's Position
y Position in Frame I (m)
x Position in Frame I (m)


Plot of fig8_simple's Velocities
x Velocity in Frame I
y Velocity in Frame I
x Velocity in Frame β
y Velocity in Frame β
vx (m/s)
vy (m/s)
time (s)


Plot of circle's Position
y Position in Frame I (m)
x Position in Frame I (m)


Plot of circle's Velocities
x Velocity in Frame I
y Velocity in Frame I
x Velocity in Frame β
y Velocity in Frame β
vx (m/s)
vy (m/s)
time (s)

Problem 2

Task 1
As given in the problem, we have

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u = Ax + Bu$$

$$J = \int_0^\infty \left(q_1 x_1^2 + q_2 x_2^2 + q_u u^2\right) dt$$

Now let $Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}$ and $R = q_u$. Thus we have:

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$$

Minimizing J with respect to u (such that Q ≥ 0 and R > 0) yields the solution $u = -Kx$ where $K = R^{-1}B^T P$ with P (positive definite symmetric matrix) being the solution to the Algebraic Riccati equation (shown below):

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

Now because P is symmetric then we let $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$. Now substituting A, B, R, Q and P we have:

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} - \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{q_u} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} + \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & p_{11} \\ 0 & p_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ p_{11} & p_{12} \end{bmatrix} - \frac{1}{q_u} \begin{bmatrix} 0 & p_{12} \\ 0 & p_{22} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} + \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & p_{11} \\ p_{11} & 2p_{12} \end{bmatrix} - \frac{1}{q_u} \begin{bmatrix} p_{12}^2 & p_{12}p_{22} \\ p_{12}p_{22} & p_{22}^2 \end{bmatrix} + \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} = 0$$

From the equation above, we derive three equations:

$$\frac{-p_{12}^2}{q_u} + q_1 = 0 \implies p_{12}^2 = q_1 q_u$$

$$p_{11} - \frac{1}{q_u} p_{12}p_{22} = 0 \implies p_{11} = \frac{p_{12}p_{22}}{q_u}$$

$$2p_{12} - \frac{1}{q_u} p_{22}^2 + q_2 = 0 \implies p_{22} = \sqrt{(2p_{12} + q_2)q_u}$$

Now solving for K:

$$K = R^{-1}B^T P = \frac{1}{q_u} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} \frac{p_{12}}{q_u} & \frac{p_{22}}{q_u} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{q_1 q_u}}{q_u} & \frac{\sqrt{(2\sqrt{q_1 q_u} + q_2)q_u}}{q_u} \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

Now finding the eigenvalues of the closed loop system:

$$sI - A + BK = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix} = \begin{bmatrix} s & -1 \\ k_1 & k_2 + s \end{bmatrix}$$

$$det(sI - A + BK) = s(k_2 + s) + k_1 = s^2 + k_2 s + k_1$$

Substituting in the values for k_1 and k_2 we have:

$$det(sI - A + BK) = s^2 + \frac{\sqrt{(2\sqrt{q_1 q_u} + q_2)q_u}}{q_u}s + \frac{\sqrt{q_1 q_u}}{q_u}$$

When we compare the above characteristic polynomial with the standard second-order polynomial $s^2 + 2\zeta_0\omega_0 s + \omega_0^2$, we see that as q_1 increases so does ω_0^2. In other words, increasing q_1 leads to a the system oscillating more and responding in a more aggressive manner. Now if q_2 increases we see that $\zeta_0\omega_0$ also increases. So by increasing q_2, our settling time decreases and we have more damping. Now increasing q_u will decrease both $\zeta_0\omega_0$ and ω_0^2. So increasing q_u leads to a greater settling time and less oscillations (i.e less aggressive response). In summary, q_1 controls the oscillations of the system (the aggressiveness of the response), q_2 controls the settling time and damping, and q_u controls both the aggressiveness/oscillations and the settling time and damping.

Task 2

Implementing the LQR in Python we verify that the K obtained analytically is the same as the K obtained in Task 1. To see this, the values of q_1, q_2 and q_u were set and the result of K was compared to the value of K obtained in Task 1. The values of q_1, q_2 and q_u were all set between 1 and 20 and the euclidean norm (l2 norm) was calculated between K obtained in task 1 and the K obtained with the LQR python implementation. Below is a plot of the errors, as the errors are on the order of 10^{-14} (this is due to numerical error in python's operations). Therefore, the K obtained analytically is the same as K obtained in Task 1.