SYSTEMS AND CONTROL ENGINEERING

SC649: EMBEDDED CONTROLS AND ROBOTICS

Assignment 2

Team
PRANAV GUPTA (22B2179)
ROHAN MEKALA (22B2106)
SAHIL SUDHAKAR (210010055)

Instructor
PROF. LEENA VACHHANI
leena.vachhani@iitb.ac.in

September 11, 2024

Contents

Work Distribution

1. **Pranav Gupta** (22B2179): Simulink model and stability analysis
2. **Rohan Mekala** (22B2106): exponential velocity control strategy
3. **Sahil Sudhakar** (210010055): ROS 1 Noetic - TurtleBot3 simulation

1 Block Diagram Setup

All the simulations in this assignment have been run with a fixed step size of 50 μs using `ode4` solver with a stop time of 10 seconds.

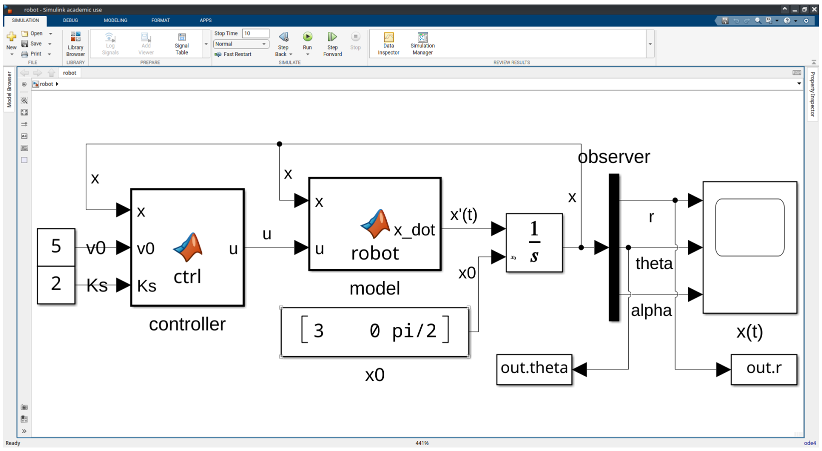


Figure 1: screenshot of simulink model used to simulate and conduct analysis

2 Plant and Controller

The plant of the kinematics of the robot model is:

$$
\begin{aligned}
\dot{r} &= v\cos(\alpha - \theta) \\
\dot{\theta} &= \frac{v}{r}\sin(\alpha - \theta) \\
\dot{\alpha} &= \omega
\end{aligned}
$$

The control law that we have used is as follows:

$$
\begin{aligned}
v &= v_0 \\
\omega &= -K_s \cdot \mathrm{sgn}(\alpha - \theta - \pi)
\end{aligned}
$$

3 Implementation

We have successfully implemented the steering control as the system response is converging and the robot settles down at the home position.

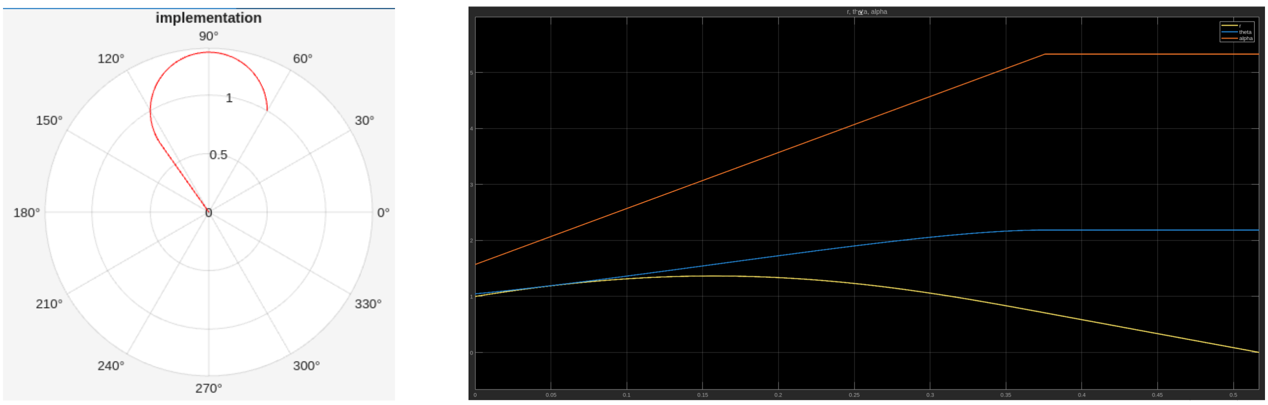
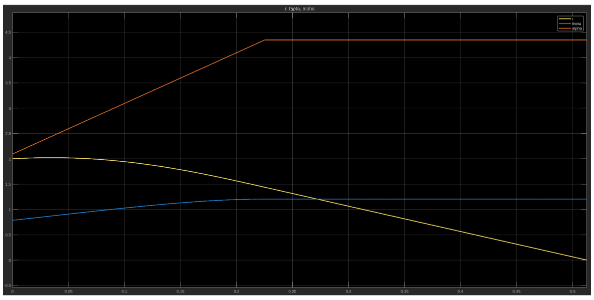
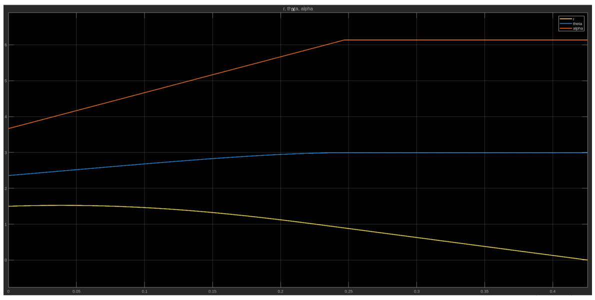


Figure 2: system response for initial conditions $\{R, \theta, \alpha\} = \{1, \frac{\pi}{3}, \frac{\pi}{2}\}$

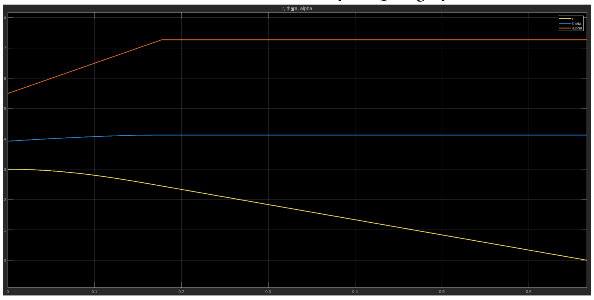
4 Response for Different Initial Positions in Each Quadrant:



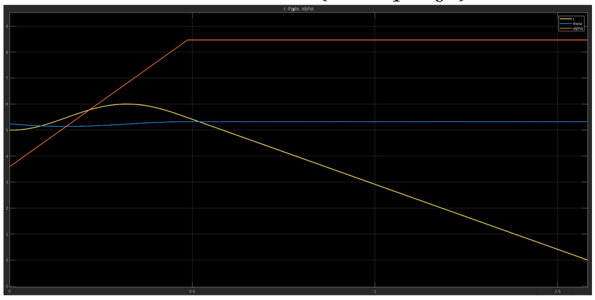
(a) $\{R, \theta, \alpha\} = \{2, \frac{\pi}{4}, \frac{2\pi}{3}\}$



(b) $\{R, \theta, \alpha\} = \{1.5, \frac{3\pi}{4}, \frac{7\pi}{6}\}$



(c) $\{R, \theta, \alpha\} = \{3, \frac{5\pi}{4}, \frac{7\pi}{4}\}$



(d) $\{R, \theta, \alpha\} = \{5, \frac{5\pi}{3}, \frac{8\pi}{7}\}$

Figure 3: system response for initial conditions

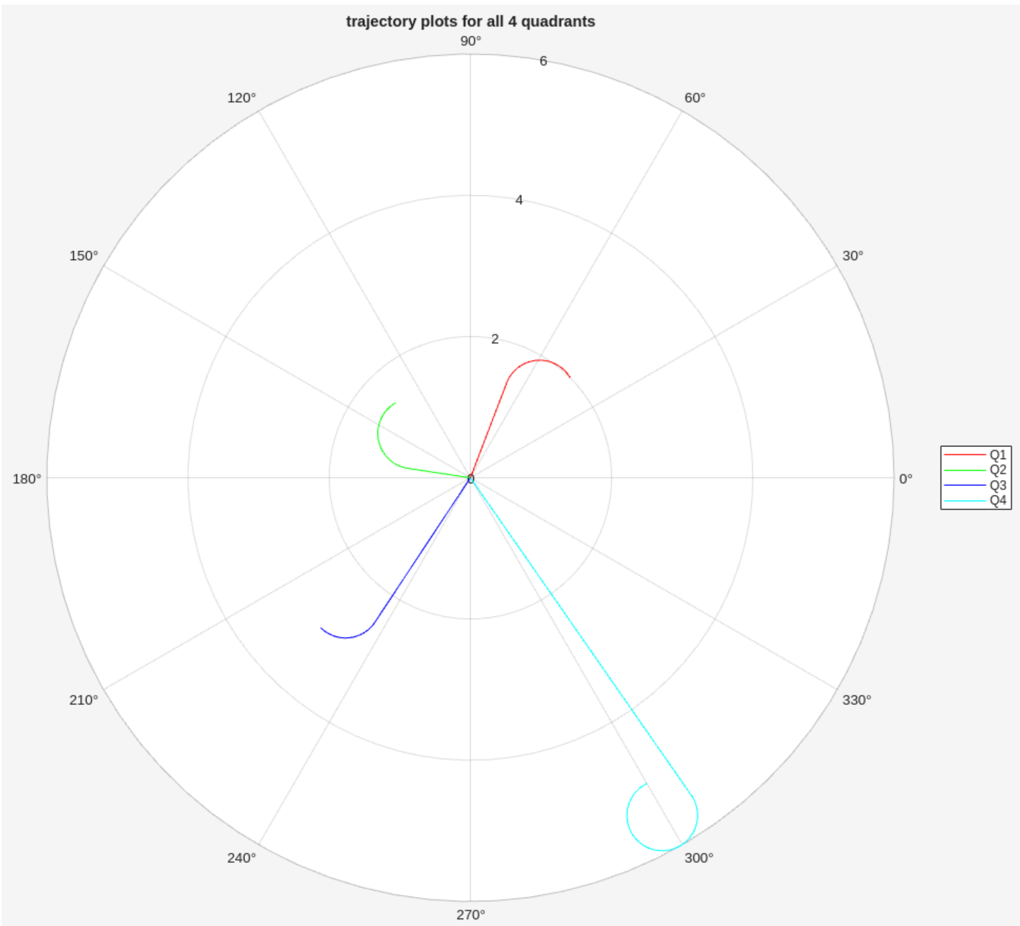
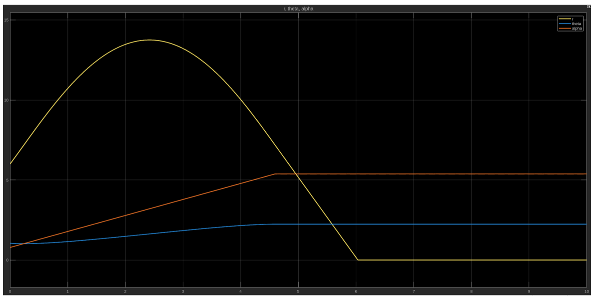


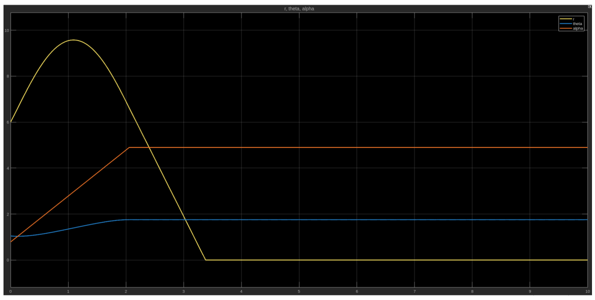
Figure 4: trajectories for all for quadrant initial conditions

5 Response for Different Control Gains

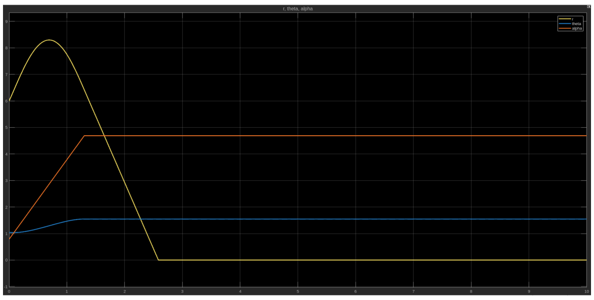
The initial conditions for this section are: $\{R, \theta, \alpha\} = \{6, \frac{\pi}{3}, \frac{\pi}{4}\}$ and $v_0 = 5, r_0 = 0.1$



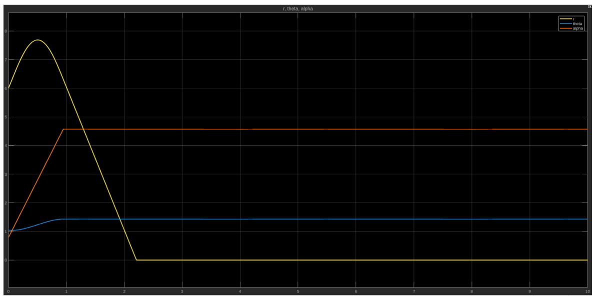
(a) $K_s = 1 \implies t_s = 6.032$



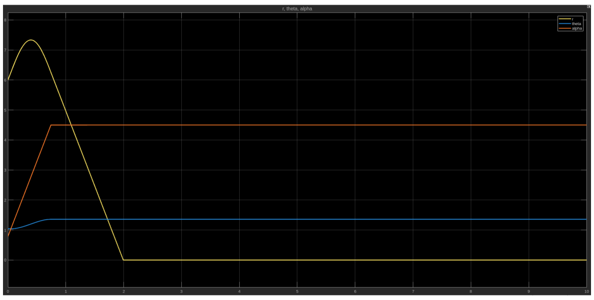
(b) $K_s = 2 \implies t_s = 3.380$



(c) $K_s = 3 \implies t_s = 2.584$



(d) $K_s = 4 \implies t_s = 2.210$



(e) $K_s = 5 \implies t_s = 1.994$

Figure 5: system response for $K \in \{1, 2, 3, 4, 5\}$, robot being forcefully stopped when r

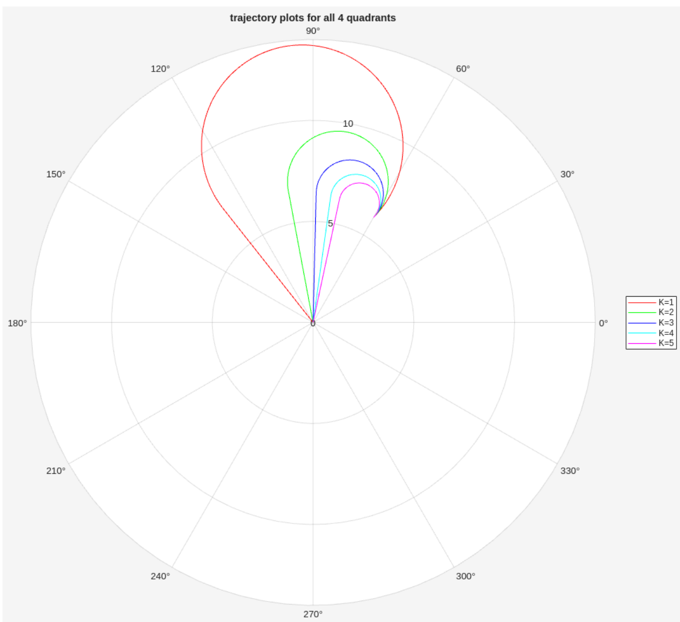


Figure 6: trajectories for all for quadrant initial conditions

Increasing K_s decreases the settling time t_s and also decreases the maximum overshoot. Physically, this means the robot's maneuver towards the home position is tighter because $\|\omega\| = K_s$, consequently dropping the settling time. One can also however notice that increasing K_s corresponds to installing a more powerful motor for the steering control, but at marginal returns on settling time for higher values of K_s.

6 Comments on Stability

We were unable to identify any inital conditions for which the controller is unstable in the sense of divergence. However, we were able to identify a class of initial conditions for which the controller is marginally unstable in the sense of getting stuck in a loop without converging to the home position, where if the initial relative heading of the robot $(\alpha_0 - \theta_0)$ is such that the control law doesn't effectively reduce the angle error between the robot's current heading and the target heading (home position), the robot could end up in a loop where it perpetually adjusts its heading in a circular manner without ever reducing the radial distance R to the home position.

One such trivial case is when the initial relative heading $\alpha_0 - \theta_0 = \frac{\pi}{2}$ and the initial radial distance $r_0 = \dfrac{v_0}{K_s}$ for which robot forever revolves about the origin in a fixed circle of radius r_0.

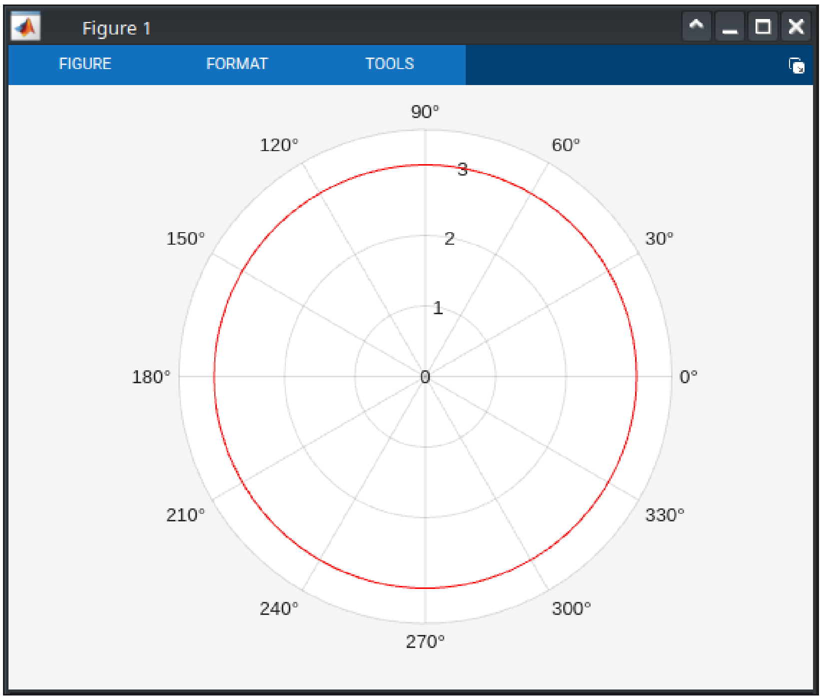


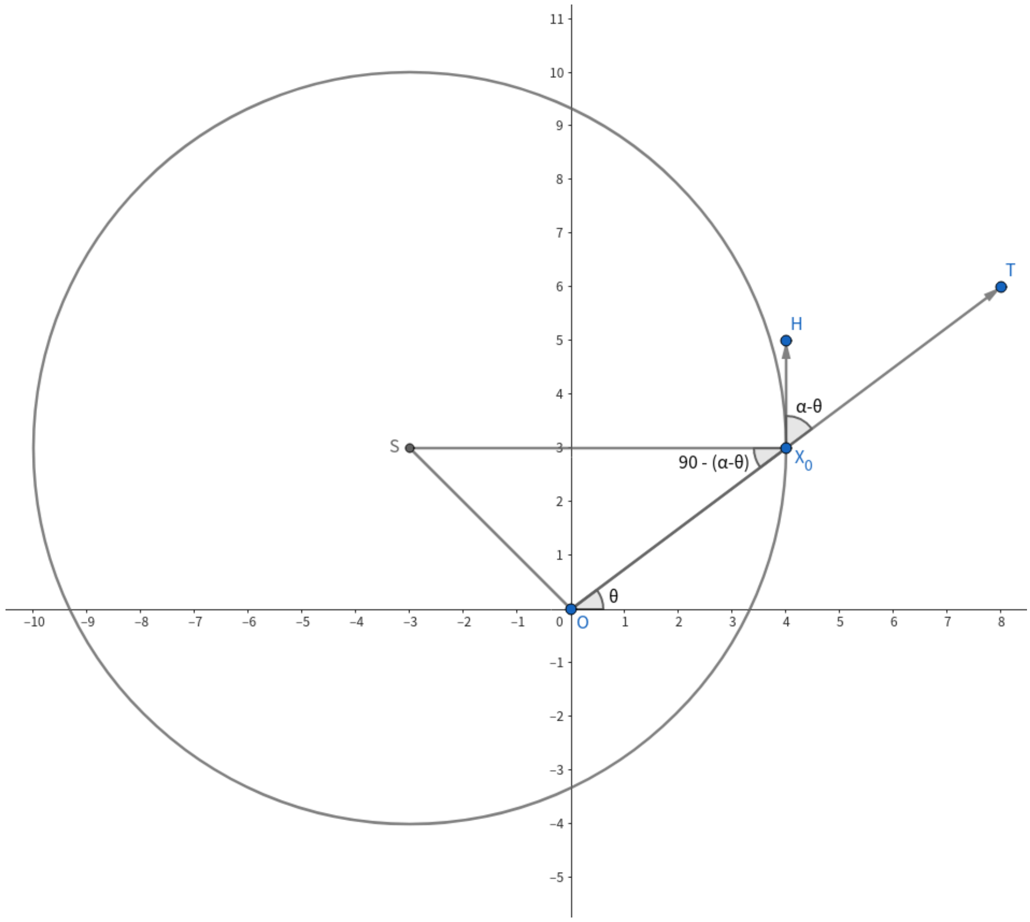
Figure 7: example of robot's marginally stable fixed circular motion

There are other non-trivial cases too, where certain combinations of initial radial distance r_0 and initial relative heading $\alpha_0 - \theta_0$ for which the robot forever revolves about a point in a fixed circle of radius $R = \dfrac{v_0}{K_s}$ not centered at the origin.

The desired relative heading of the robot to head towards home is $\alpha - \theta = \pi$. Considering this, we can break down the general motion of the robot under given controller into 2 phases:

1. An initial phase of circular motion of the robot in a circle of radius $\frac{v_0}{K_s}$ until its heading is aligned towards the home position, i.e. $\alpha - \theta = \pi$
2. Straight line motion of the robot towards the origin until it reaches home.

The switch from the 1st phase to the 2nd is triggered when its heading aligns towards home. Our hypothesis is that if the robot's initial state and control parameters are such that its heading during the 1st phase never aligns towards the home, it will forever remain stuck in circular motion. This would happen only when one cannot draw a tangent from the origin to the virtual circle of the 1st phase of the robot's motion, which is only possible if the origin lies within said circle. This happens when the distance of the virtual circle's centre from the origin is $< R = \frac{v_0}{K_s}$.



Hence, to compute the criteria for marginal stability, one can imagine a triangle ΔOX_0S where O is the home position, $X_0 \equiv (r_0, \theta_0)$ is the initial position of the robot and S is the centre of the virtual circle.

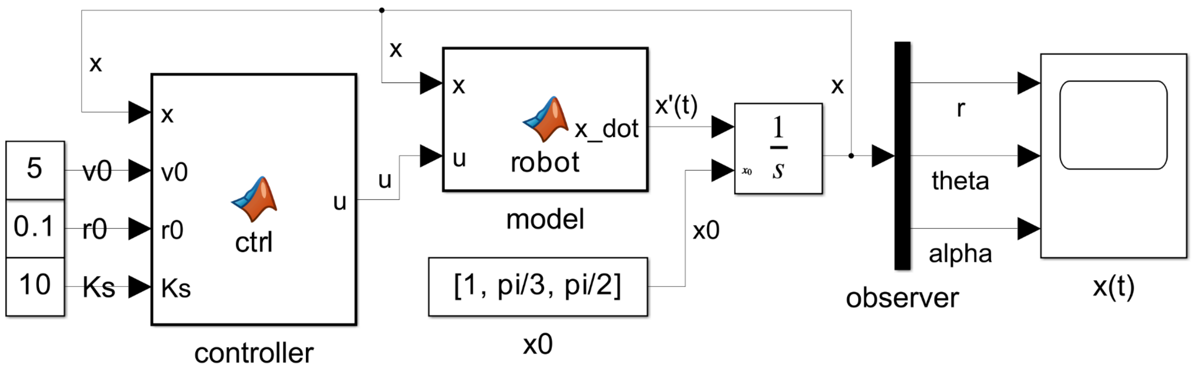
$\angle OX_0S = \frac{\pi}{2} - (\alpha - \theta)$ and sides $\|OX_0\| = r_0$ and $\|X_0S\| = R = \frac{v_0}{K_s}$. Marginal instability occurs when $\|OX\| = l < R$. By the cosine law,

$$r_0^2 + \cancel{R^2} - 2rR\cos\left(\frac{\pi}{2} - (\alpha_0 - \theta_0)\right) < \cancel{R^2} \implies r_0^{\cancel{2}} < 2\cancel{r_0}R\sin(\alpha_0 - \theta_0) \implies \boxed{r_0 < 2\frac{v_0}{K_s}\sin(\alpha_0 - \theta_0)}$$

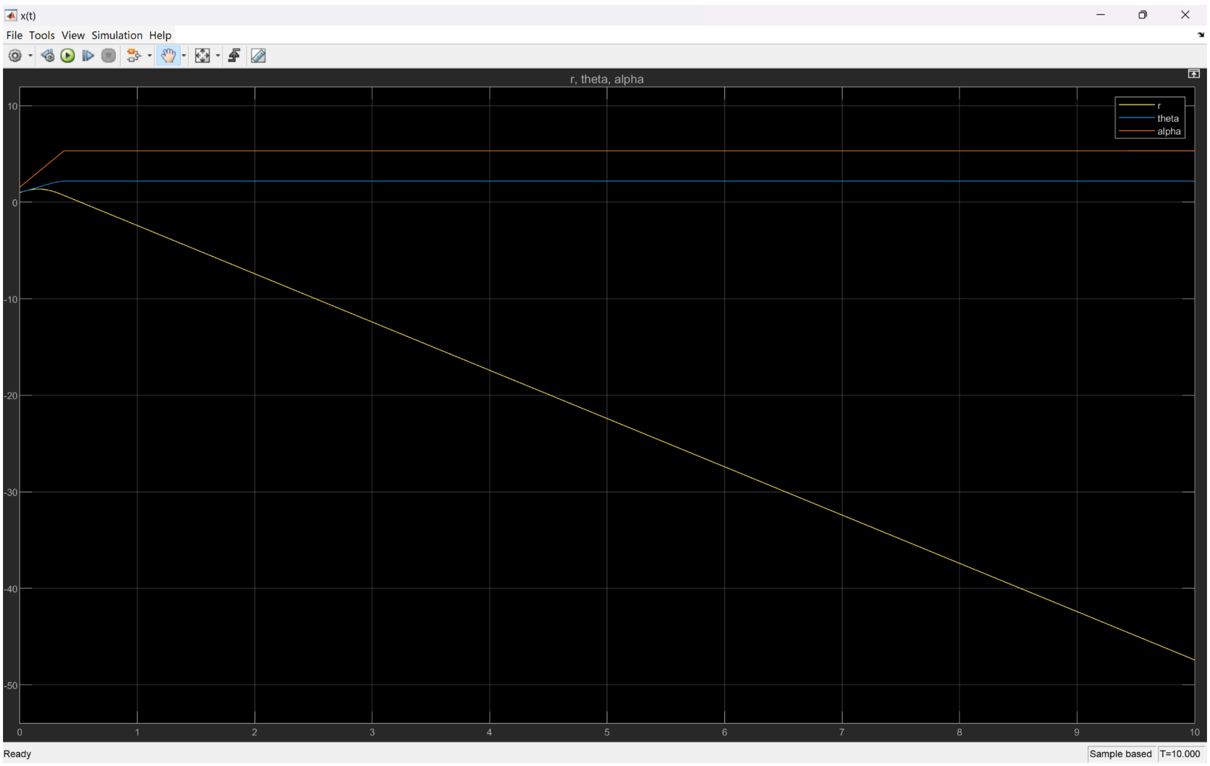
Hence, the class of initial conditions satisfying the above inequality will always lead to a marginally unstable response of fixed circular motion.

Assignment 2

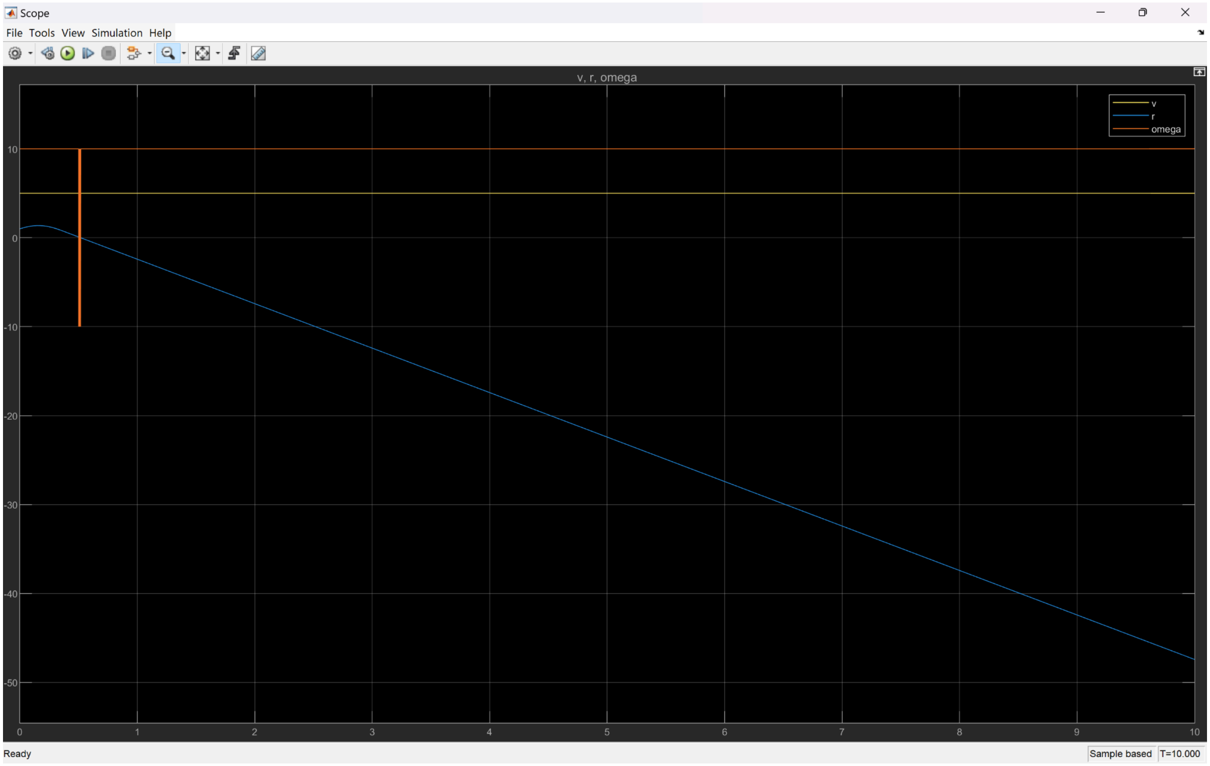
Rohan Suresh Mekala (22B2106)
Pranav Gupta (22B2179)
Sahil Sudhakar (210010055)

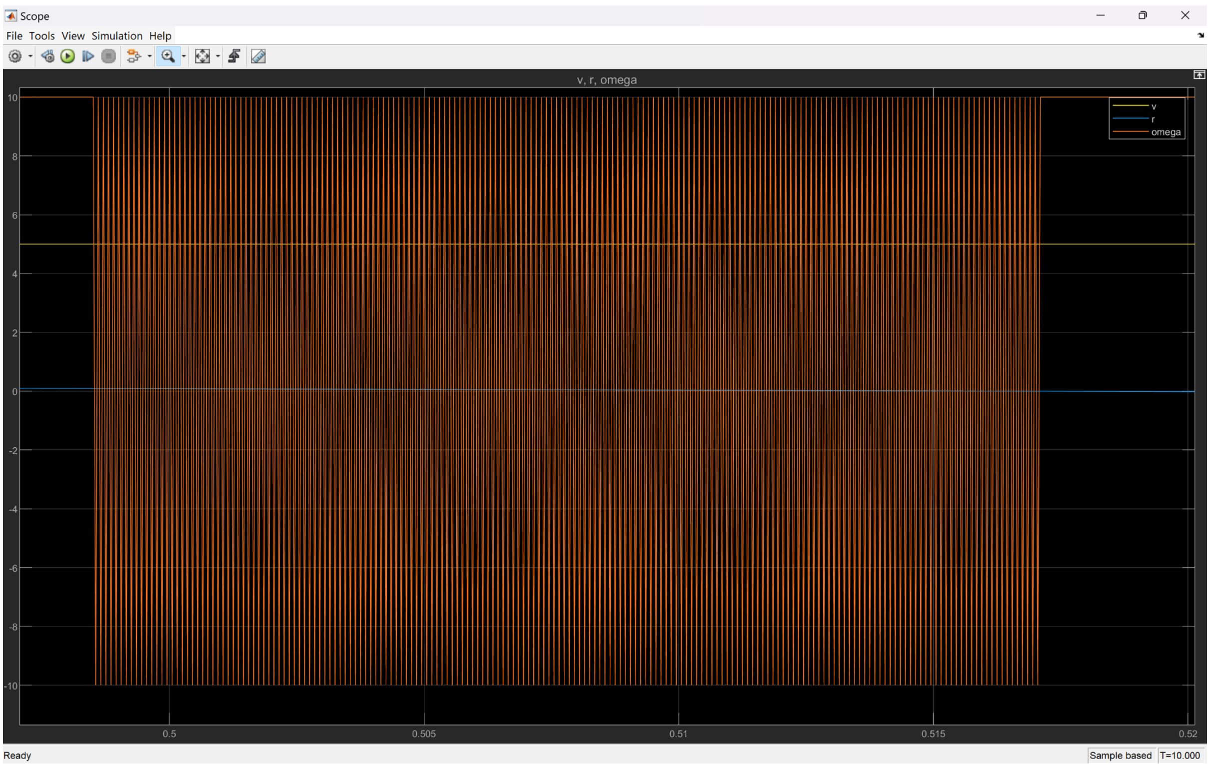


This is the block diagram setup that we have implemented in Simulink.
Note: In the assignment, it is mentioned that v is constant, however, with that assumption, the system response kept diverging for all possible initial conditions and cases. Here is the response plot for one random case:-



Note: All the simulations in this assignment have been run with a fixed step size of 5e-05 using ode4 solver with a stop time of 10 seconds.
We can clearly observe from this sample simulation that R diverges drastically implying the robot will never settle at the home position. This is the inherent issue with keeping v as constant which causes ω to fluctuate with a very high frequency between +Ks and -Ks at alternate simulation timesteps when the robot first comes very close to home position (R~0) which causes the forestated numerical instability.



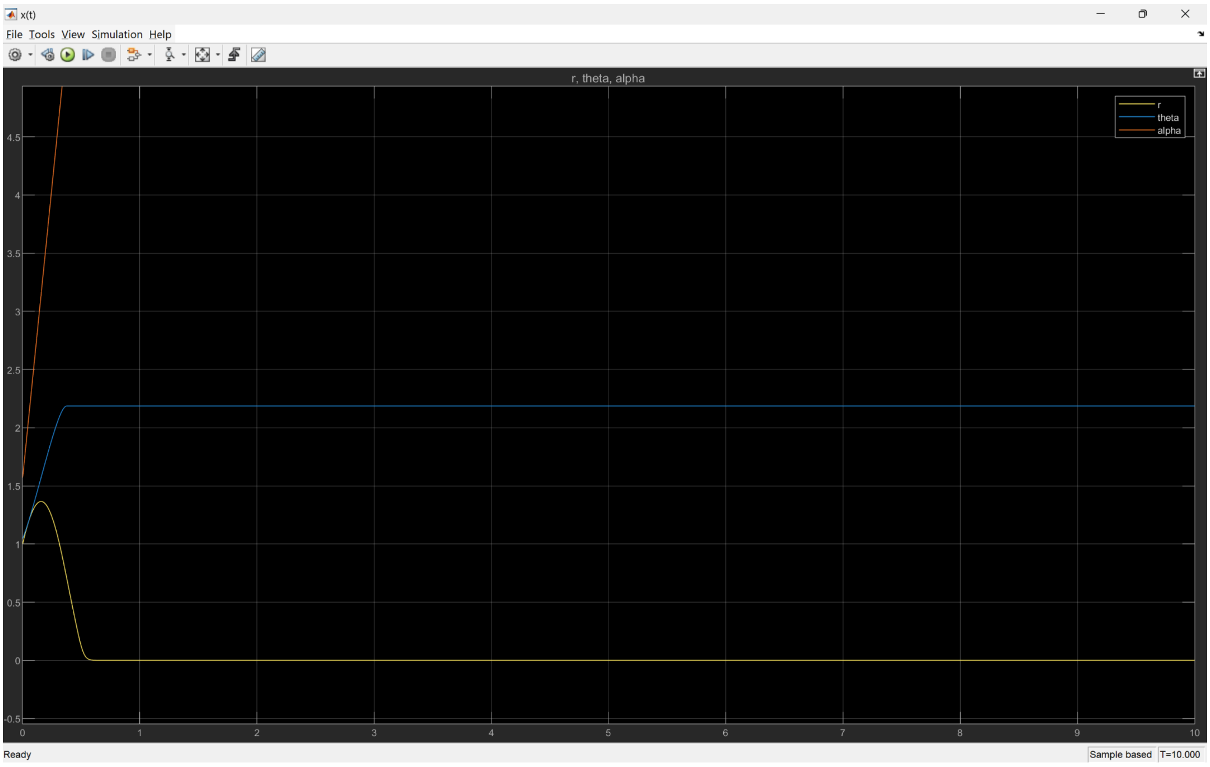


High frequency ω fluctuations at R=0

Owing to this issue, we decided to tweak the control law to control v to vary monotonically with r to ensure that the robot slows down as it approaches the home position and comes to rest at the home position. The primary goal was to pick a decaying function (with r as the independent variable) for v. We decided to proceed with the exponential decay function $(1-e^{-r/r0})$ as that was giving a very short settling time. Here is the final control law that we implemented:-

$v = v_0 * (1-\exp(-r/r_0))$
$\omega = -K_s * \mathrm{sgn}(\alpha - \theta - \pi)$

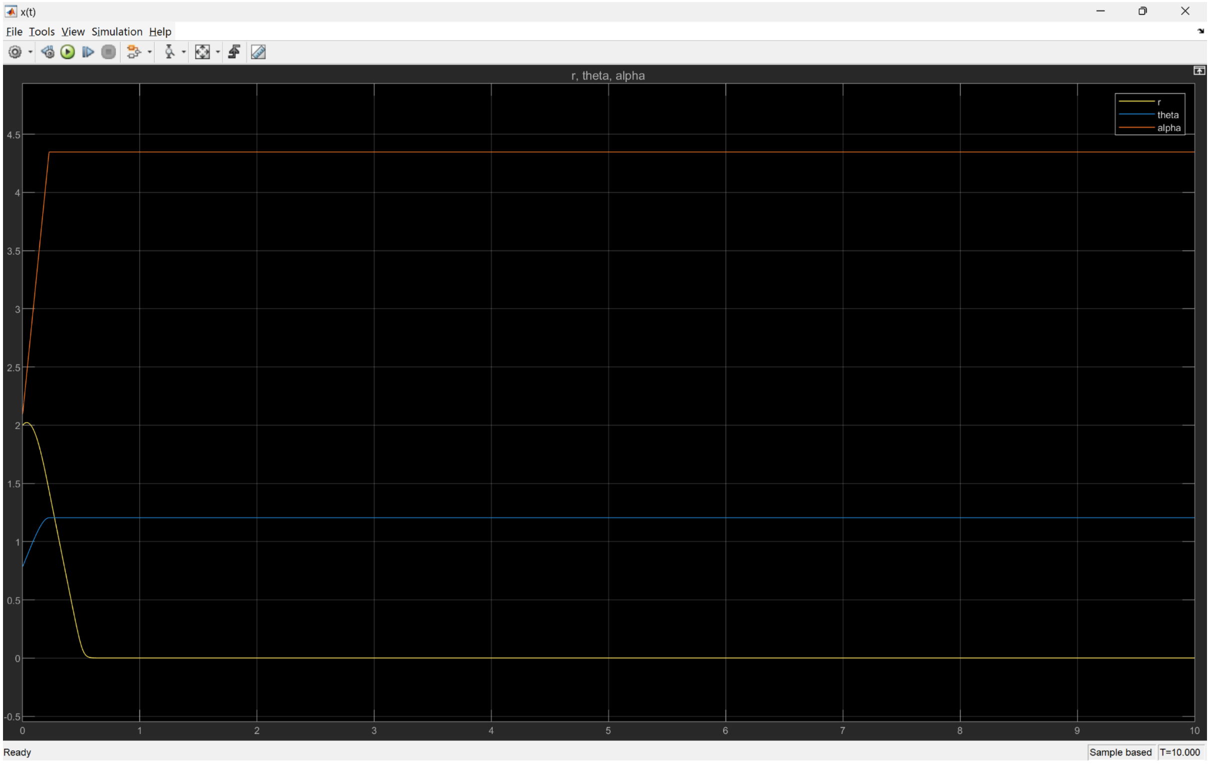
This is how we reached the final block diagram setup as depicted in the first figure.



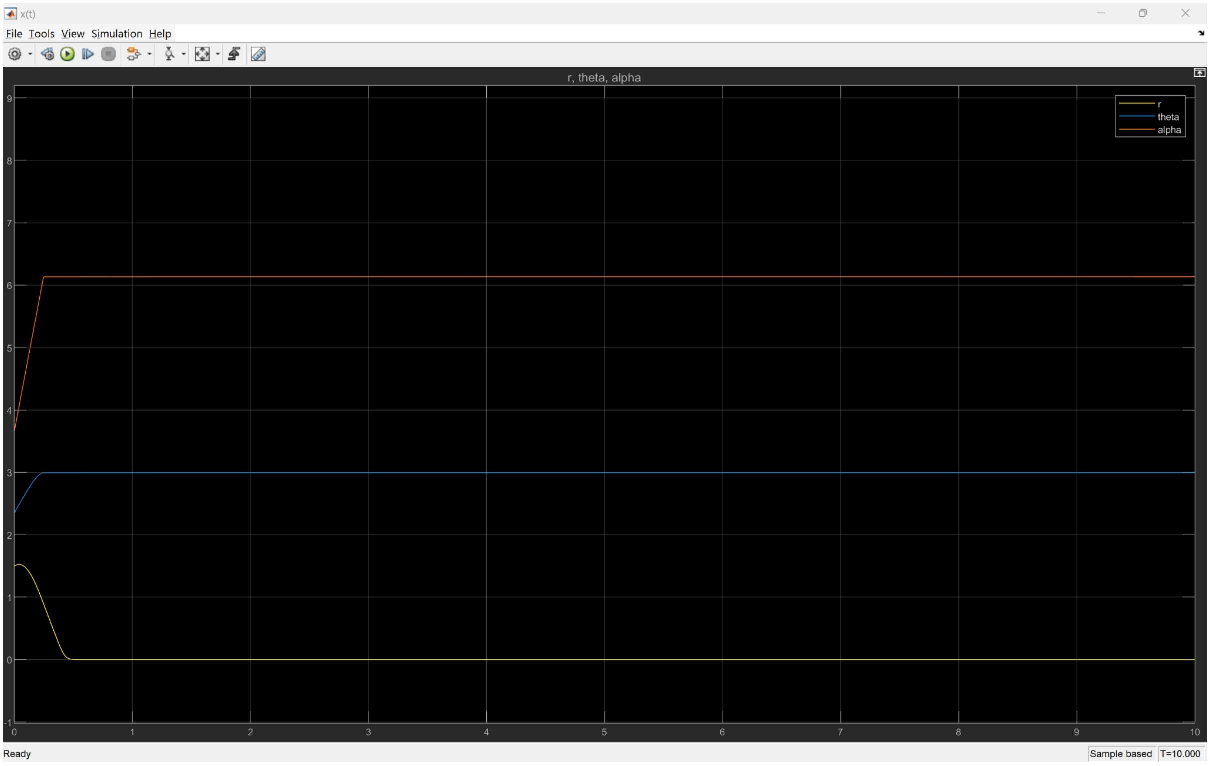
Therefore, with the help of this tweaked control law, we have fixed the issue of divergence and we can observe that the system response converges with a very short settling time and the robot is able to settle at the home position. This response was for initial conditions {R, θ, α} = {1,π/3, π/2}.

Let’s check the system response for different initial positions of the robot in each quadrant:-

1st quadrant:-
{R, θ, α} = {2,π/4, 2π/3}

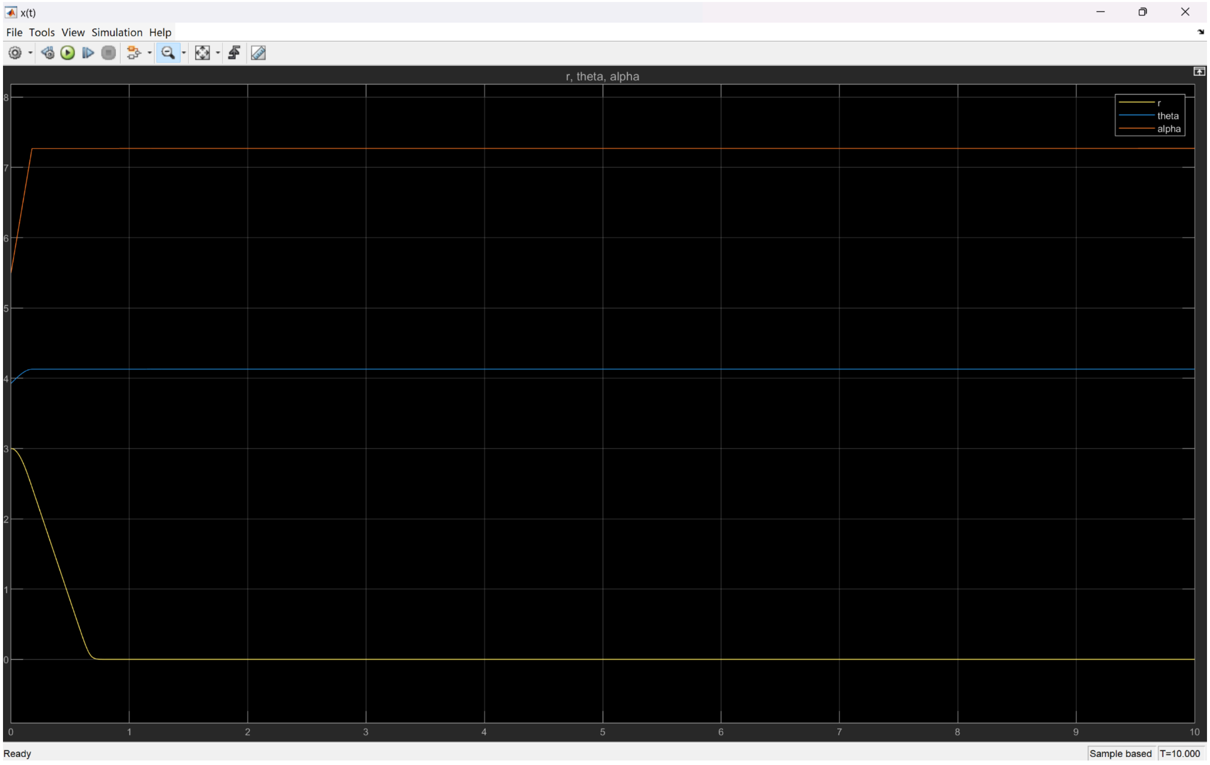


2nd quadrant:-
{R, θ, α} = {1.5,3π/4, 7π/6}



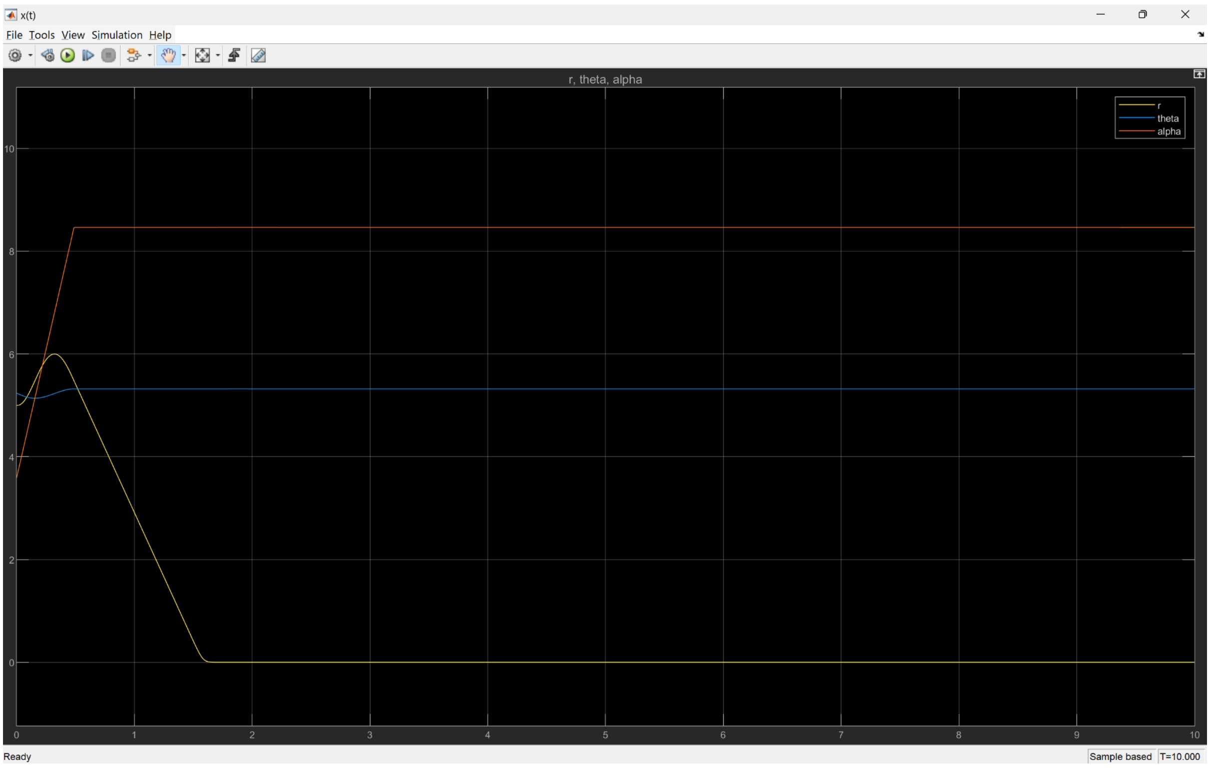
3rd quadrant:-

{R, θ, α} = {3,5π/4, 7π/4}



4th quadrant:-

{R, θ, α} = {5,5π/3, 8π/7}

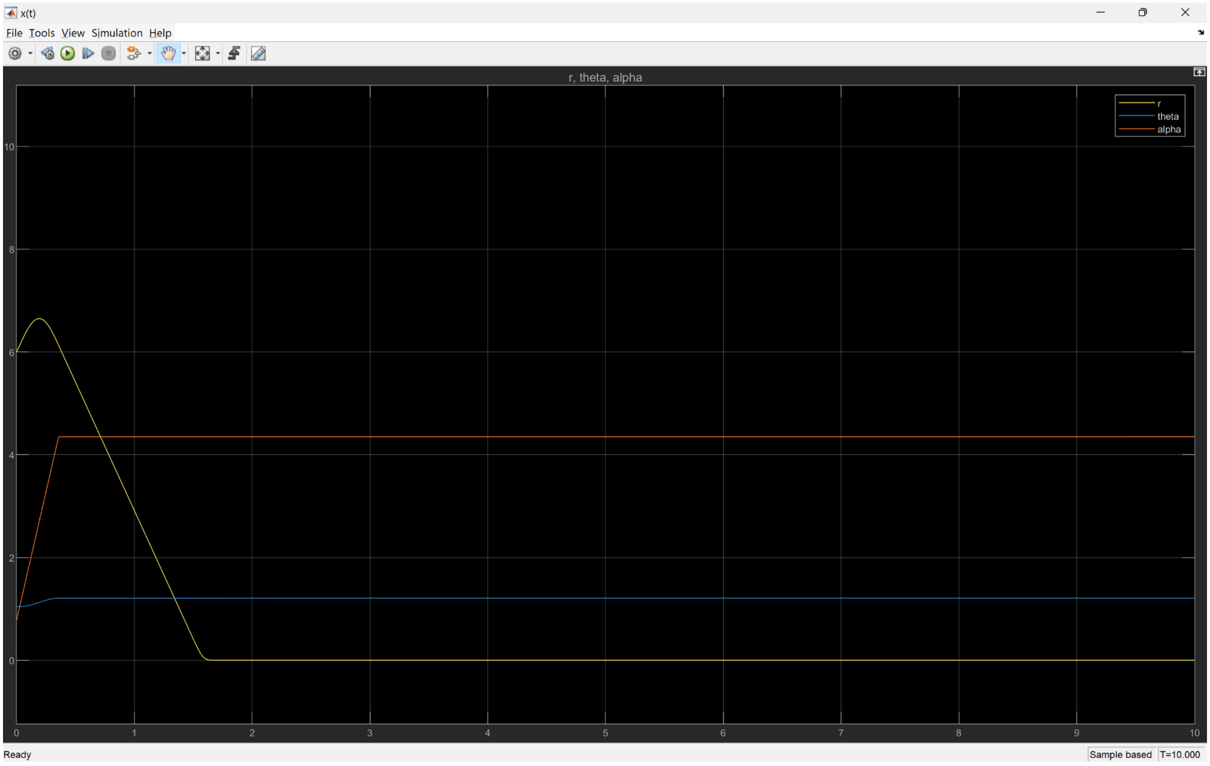


Let's check the system response for different control gains:-

As per our tweaked control law, we have 3 control gains: v_0, r_0 and K_s. Let's vary each of them and check the system response.

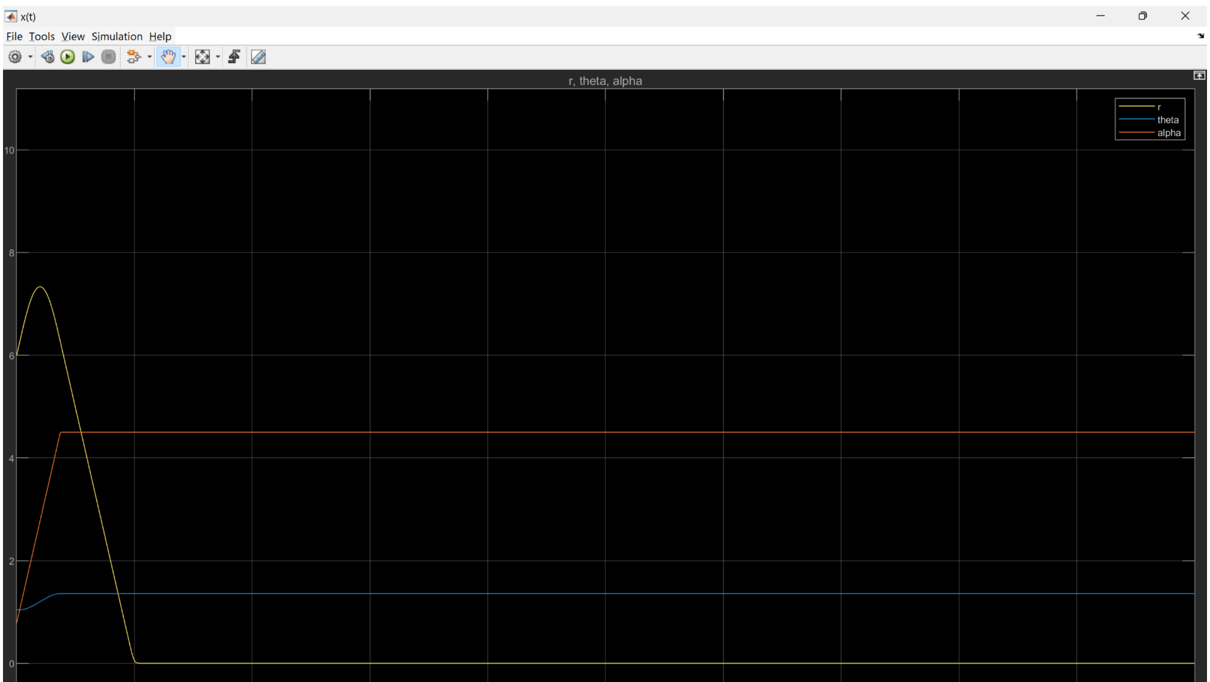
$\{R, \theta, \alpha\} = \{6, \pi/3, \pi/4\}$

a) Varying v_0:-
$r_0 = 0.1$
$K_s = 10$

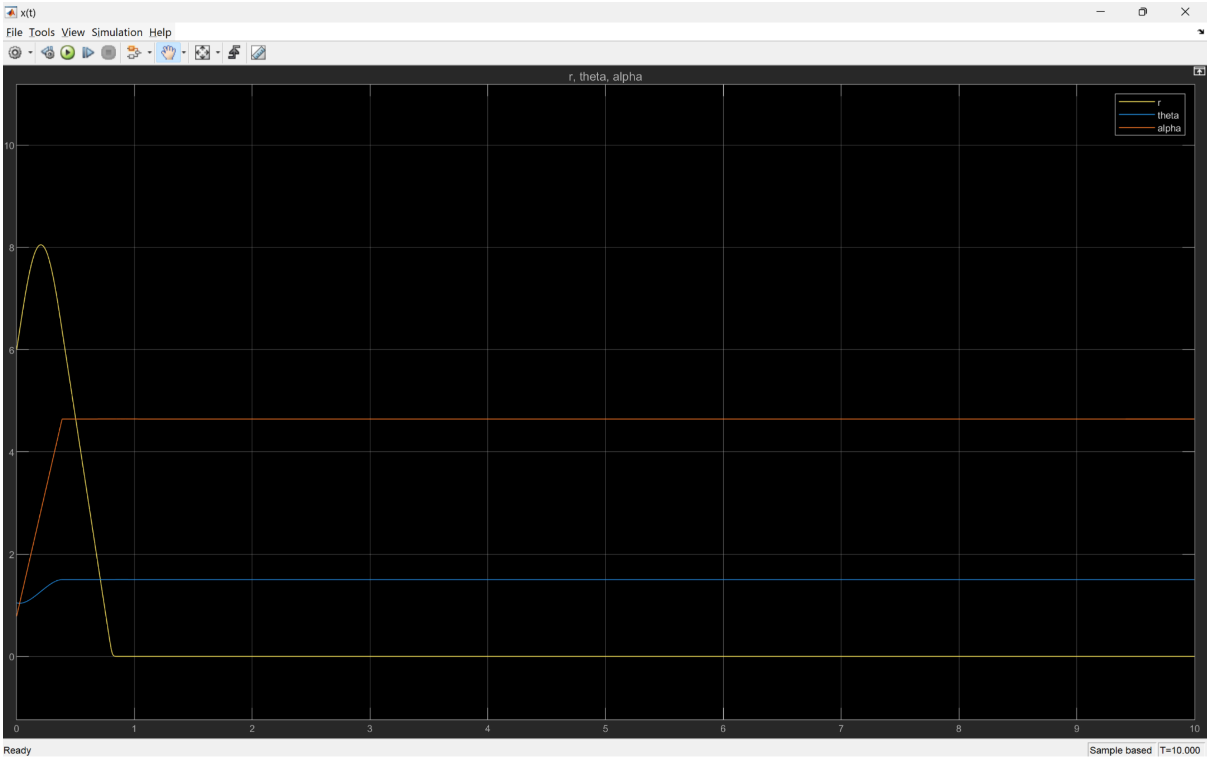


1) $v_0 = 5$

2) $v_0 = 10$



3) $v_0 = 15$



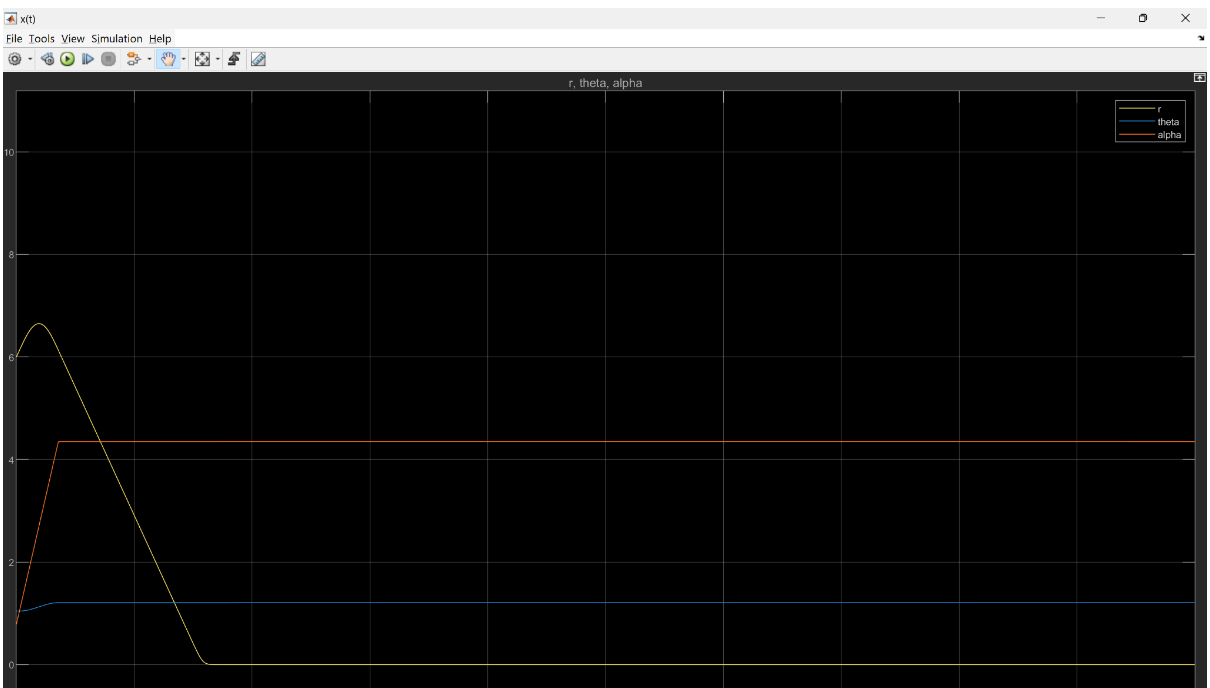
On increasing v_0, we can observe that the setting time decreases but at the cost of increasing the maximum peak overshoot. Moreover, we can observe an increase in the change of θ on increasing v_0. This is compliant with the physical system as increasing v_0 essentially increases the velocity which should decrease the time required to reach the home position while also causing increases in change of its orientation (θ) and inertia (which can be linked to the maximum peak overshoot).

b) Varying r_0:-

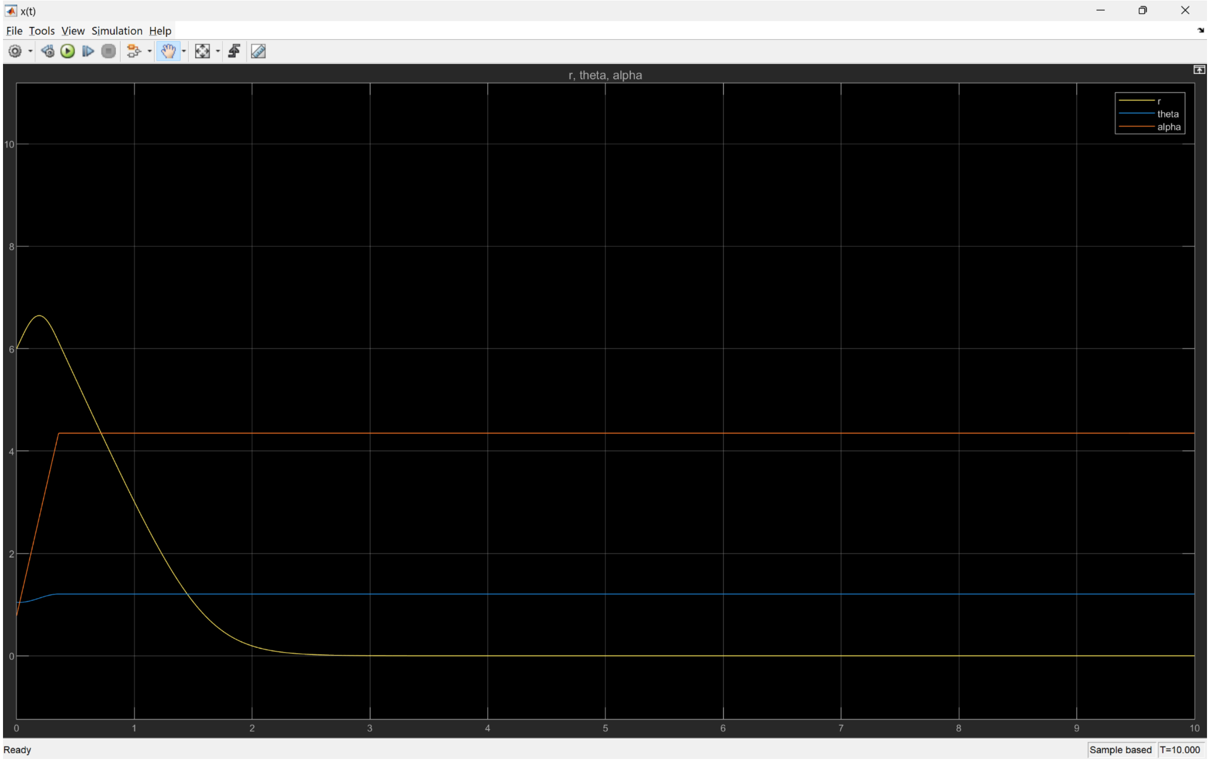
$v_0 = 5$

$K_s = 10$

1) $r_0 = 0.1$

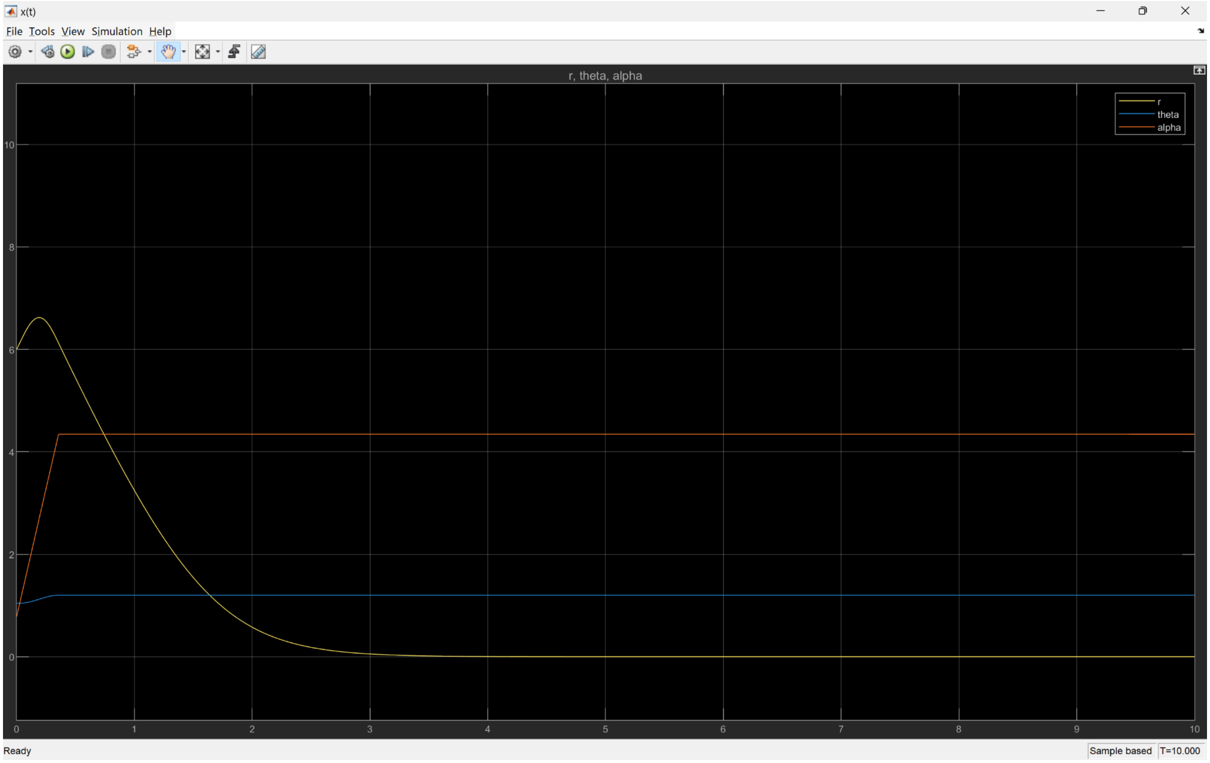


2) $r_0 = 1.2$



x(t)
File Tools View Simulation Help
r, theta, alpha
r
theta
alpha
Ready
Sample based
T=10.000

3) $r_0 = 2$

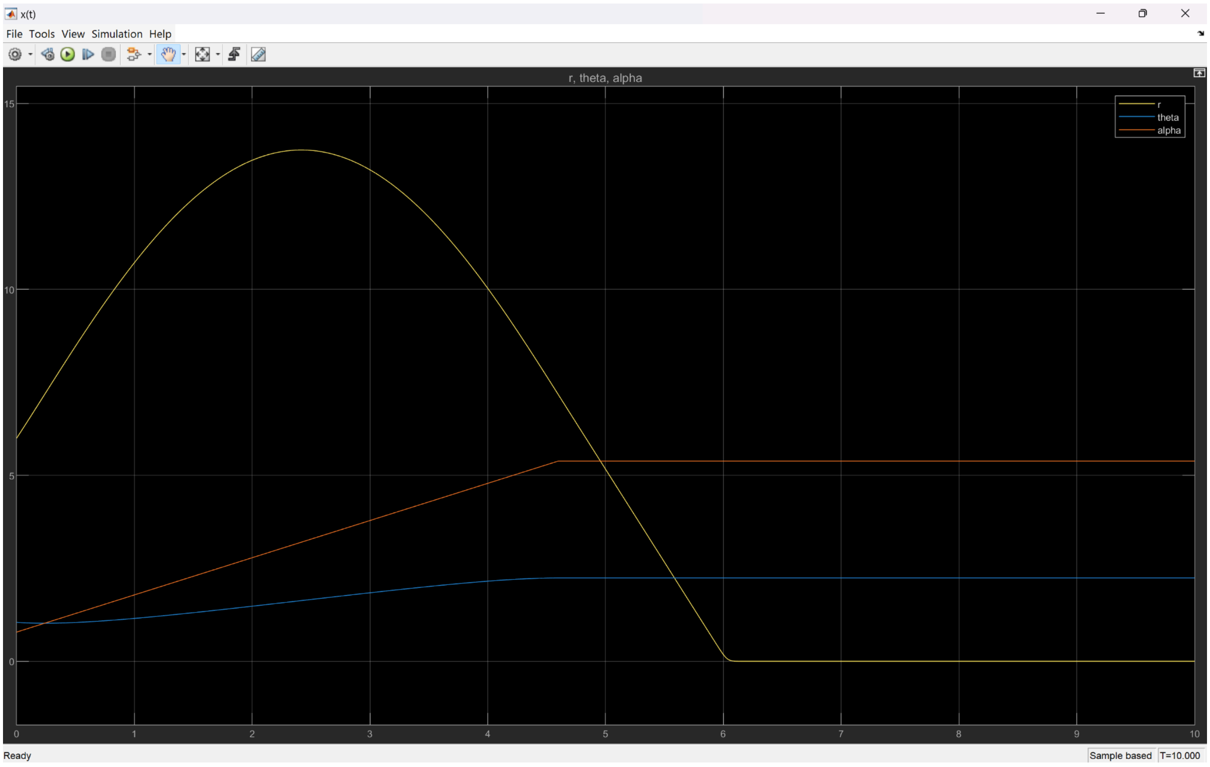


x(t)
File Tools View Simulation Help
r, theta, alpha
r
theta
alpha
Ready
Sample based
T=10.000

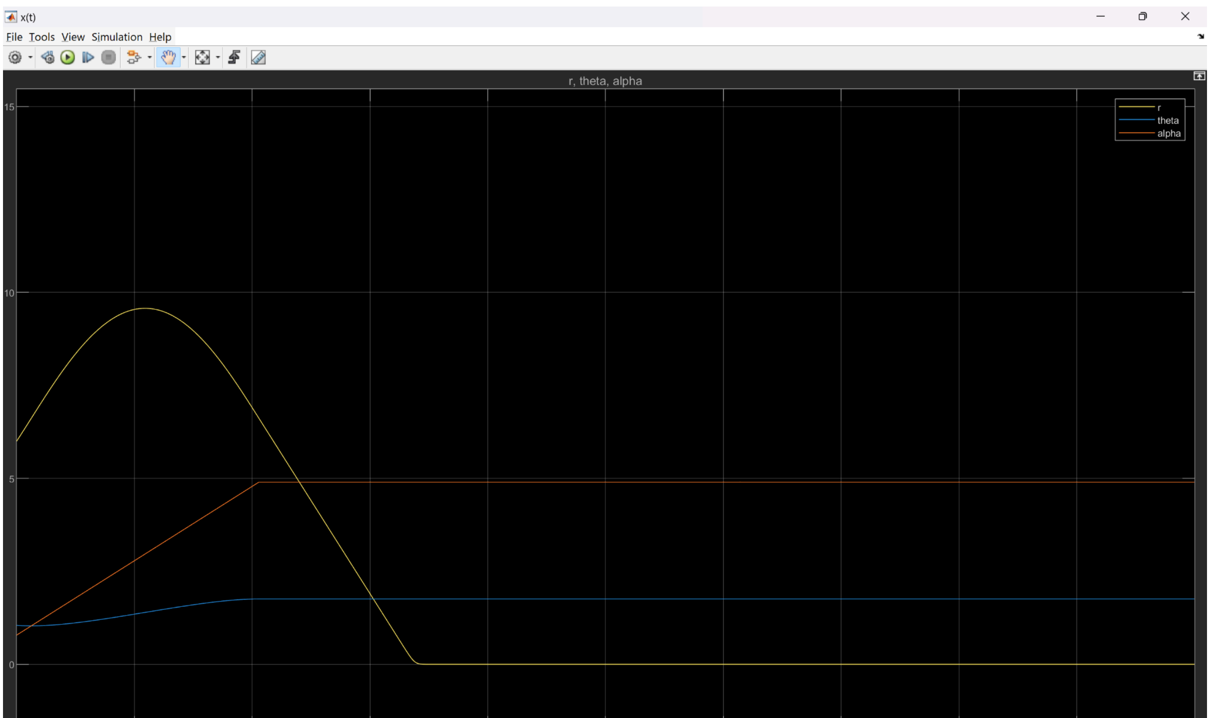
We can observe that on increasing r_0, the settling time will increase as the exponential decay decreases. Therefore, we should choose a small value of r_0 for our control law to reduce the settling time as much as possible.

c) Varying K_s:-
$v_0 = 5$
$r_0 = 0.1$

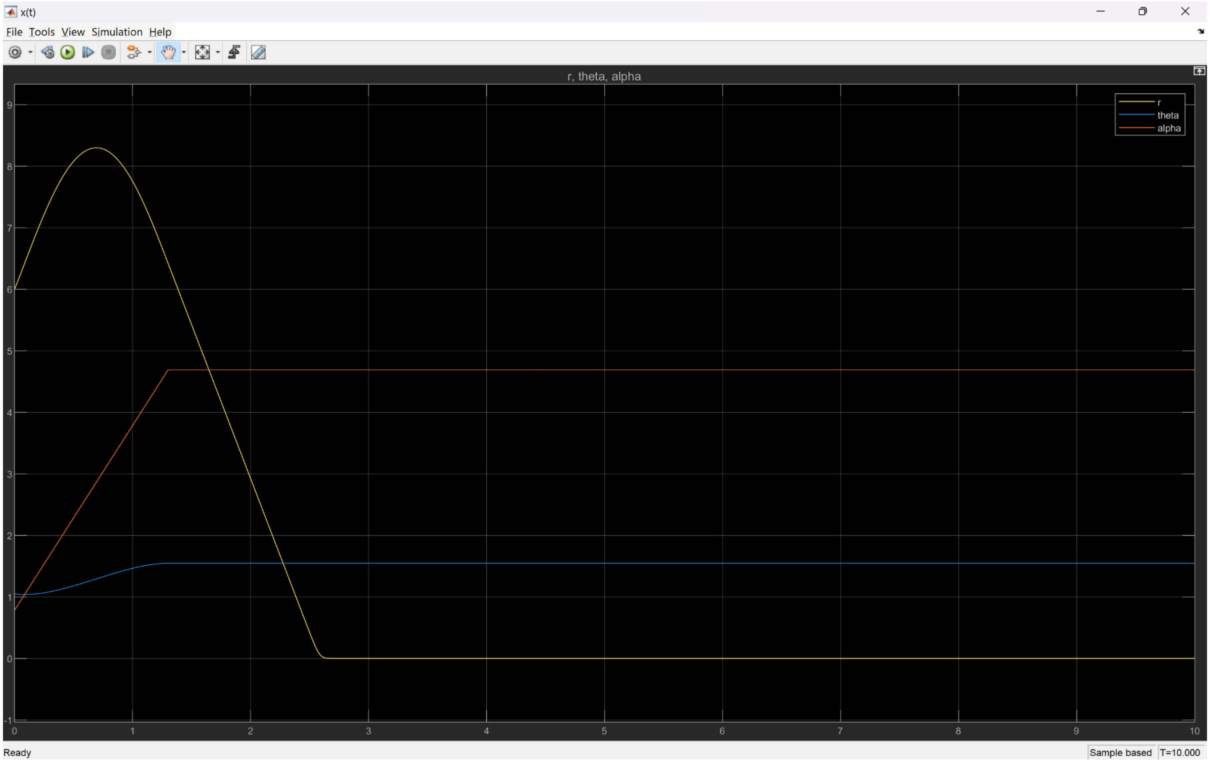
1) $K_s = 1$



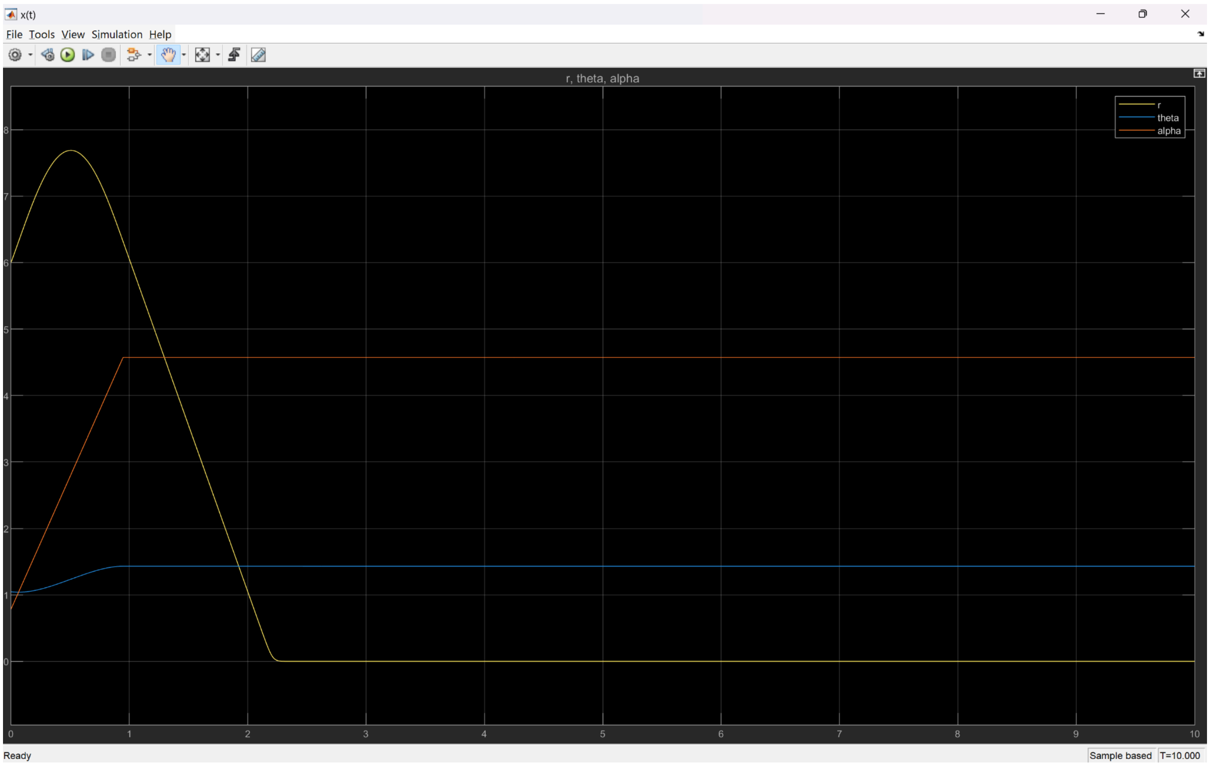
2) $K_s = 2$



3) $K_s = 3$


x(t)
File Tools View Simulation Help
r, theta, alpha
r
theta
alpha
Ready
Sample based
T=10.000

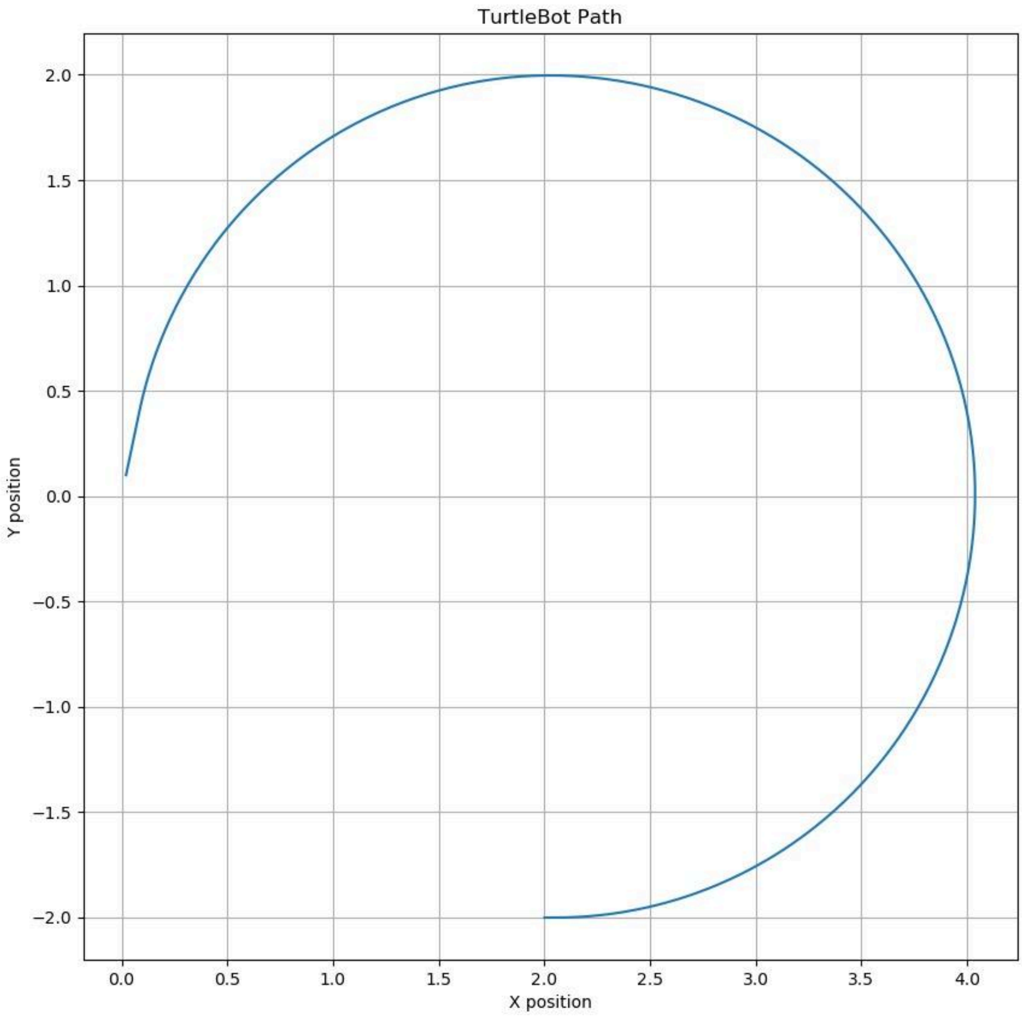
4) $K_s = 4$

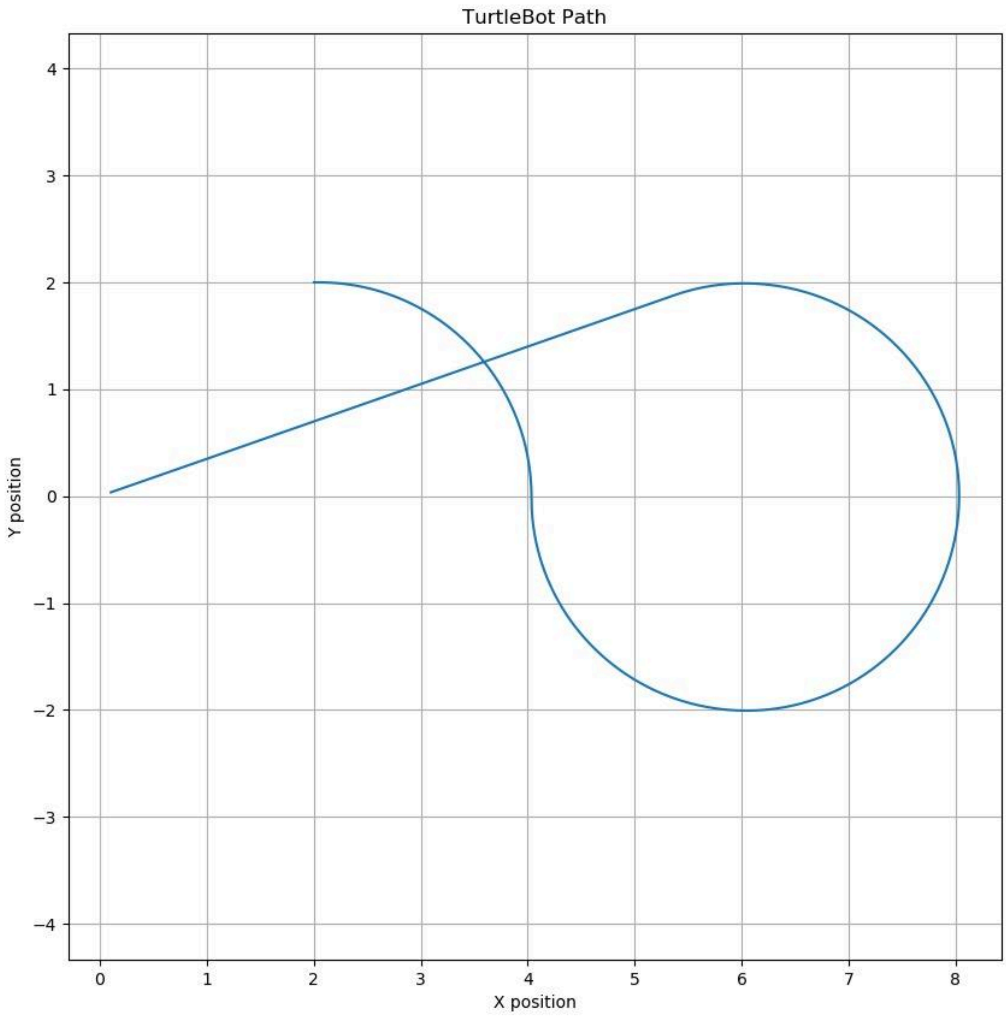

x(t)
File Tools View Simulation Help
r, theta, alpha
r
theta
alpha
Ready
Sample based
T=10.000

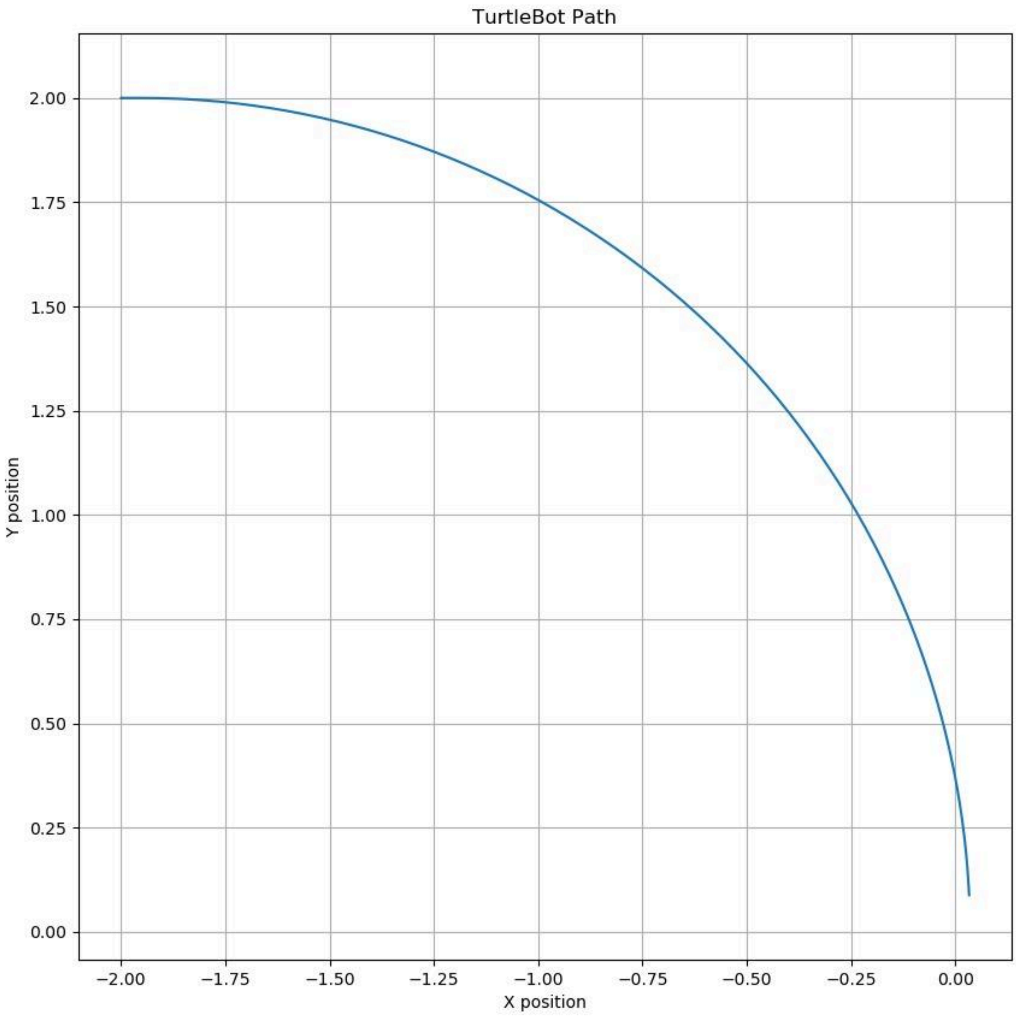
Increasing K_s decreases the settling time slightly, but the more important effect that it has is decreasing the maximum peak overshoot. Physically, what this means is that the robot is more responsive to control with a high latency to counter any existing inertia and direct the robot towards the home position. We can observe that as K_s increases, the states (R, θ, α) reach their steady state values faster which further highlights this point.

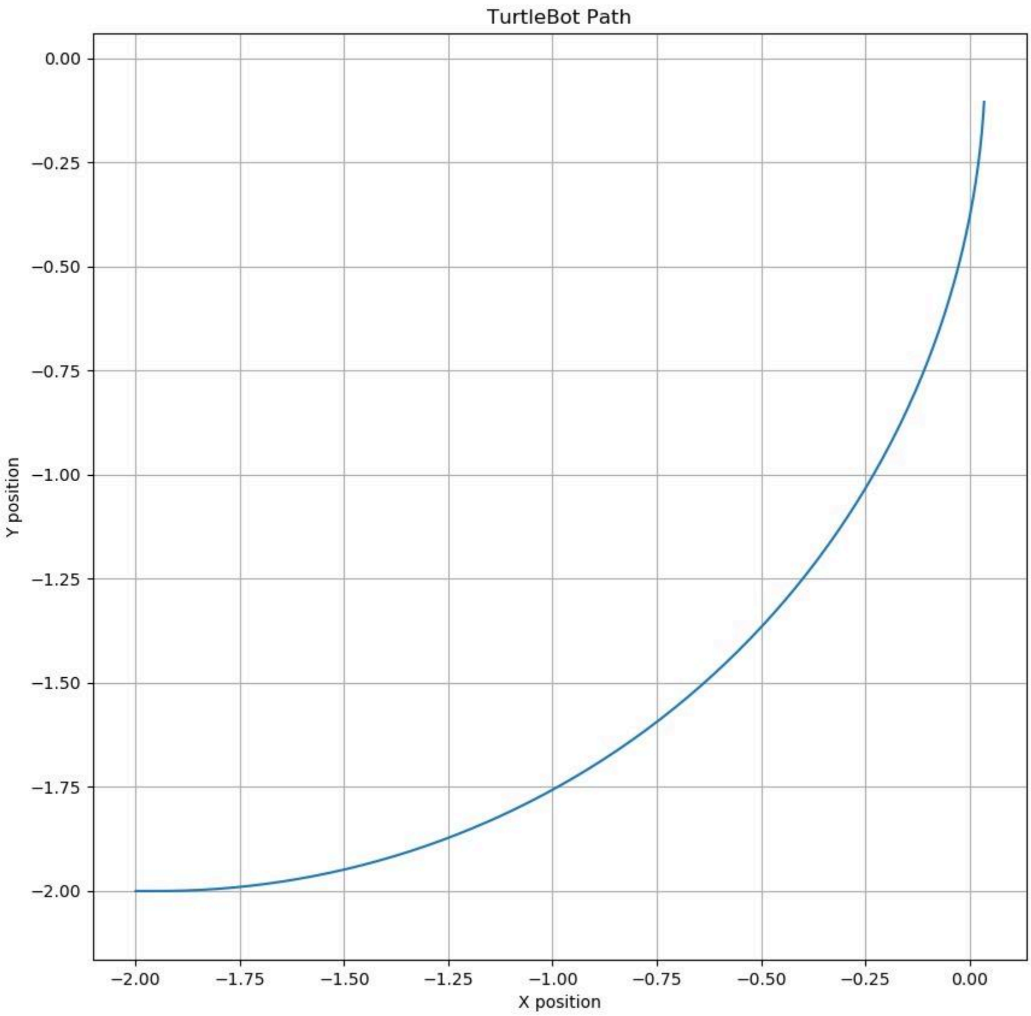
Stability of the controller:-

As observed from the above sections, the exponential decay control law used to control v to vary with r has ensured that inspite of changing the initial position or the control gains, the system response converges with a short settling time as compared to the constant v control strategy and therefore this controller is stable for all conditions.


TurtleBot Path
Y position
X position
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


TurtleBot Path
Y position
X position
4
3
2
1
0
−1
−2
−3
−4
0
1
2
3
4
5
6
7
8


TurtleBot Path
Y position
X position
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−2.00
−1.75
−1.50
−1.25
−1.00
−0.75
−0.50
−0.25
0.00


TurtleBot Path
Y position
X position
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
−2.00
−1.75
−1.50
−1.25
−1.00
−0.75
−0.50
−0.25
0.00