# Adaptive Control of Autonomous Vehicles

Dhruv Shah , Tejal Barnwal

November 23, 2021

## Introduction

In this paper we have presented an adaptive control design for an autonomous car. We have made a few simplifying assumptions which are explained subsequently. We have demonstrated via simulations that our design can track any trajectory which consists of straight lines connected at an angle of atmost 15 degrees when travelling at 10m/s i.e 36km / h. We have also demonstrated tracking for a circular trajectory.
We also present the control design for a three wheeled vehicle which is adopted directly from a reference paper and explain why that design cannot be applied to our system.

## State Space Model

For the Vehicle Model we utilize the dynamic bicycle model. Further, we utilize the following assumptions: (i) each tire shares the same parameters (vertical load, stiffness, etc.), and the lateral forces on left side and right side tires are symmetric; (ii) air resistance is negligible; and (iii) the vehicle is front-wheel drive (and front-wheel steered), and that the slip angle equals the steering angle. Using vehicle mass as $m$, and orientations speed by $\phi$, the simplified dynamical model, with acceleration $a$ and steering angular speed $\omega$ as control inputs $\boldsymbol{u} = [a, \delta]^T$. We have the following state space

$$\boldsymbol{\xi} = [x, y, v, \theta, \phi]^T$$

where $x$ and $y$ are the coordinates of the COM of the vehicle,$v$ is the velocity of the vehicle and $\delta$ is the steering angle.

$$\dot{\xi} \;=\; \begin{bmatrix} v\ sin(\theta) \\ v\ cos(\theta) \\ cos(\delta)a \;-\; \frac{2}{m}\, F_{yf}\ sin(\delta) \\ \phi \\ \frac{1}{J}(L_a(ma\ sin(\delta) \;+\; 2F_{yf}\ cos(\delta)) \;-\; 2L_bF_{yr} \end{bmatrix}$$

where $L_a$ is the distance between the centers of the front wheels and the vehicle's center of mass, $L_b$ is the distance between the centers of the rear wheels and the vehicle's center of mass, and $J$ is the rotational momentum. $F_{yf}$ is the front tire lateral force, $F_{yr}$ is

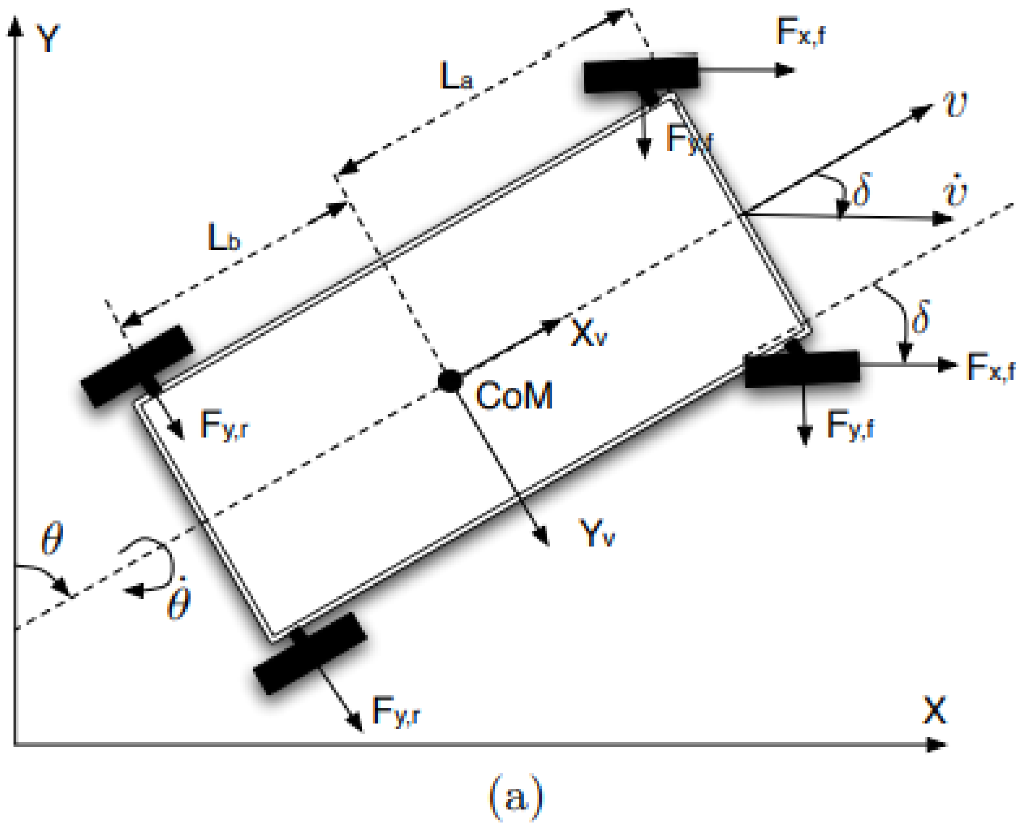


Figure 1: Vehicle Dynamic Model

the rear tire lateral force.
These forces can be computed from:

$$F_{yf} \; = \; C_y \left( \delta \; - \; \frac{L_a \phi}{v} \right)$$

$$F_{yr} \; = \; C_y \left( \frac{L_b \phi}{v} \right)$$

where $C_y$ is the lateral tire stiffness.

## Adaptive Backstepping Control for Autonomous Car

From the state space model defined above, we define

$$\alpha \ = \ cos(\delta)a \ - \ \frac{2}{m} \ F_{yf} \ sin(\delta) \tag{1}$$

$$\beta \ = a \ sin(\delta) \ + \frac{2F_{yf}}{m} \ cos(\delta) \tag{2}$$

as our virtual control inputs. We would be design the controller and getting expressions for $\alpha$ and $\beta$ and then use a numerical solver to obtain the $a$ and $\delta$ signals.
Therefore we obtain the following system dynamics,

$$\dot{x} = v \ sin(\theta) \tag{3}$$

$$\dot{y} = v \ cos(\theta) \tag{4}$$

$$\dot{v} \ = \ \alpha \tag{5}$$

$$\dot{\theta} \ = \ \phi \tag{6}$$

$$\dot{\phi} = \ \frac{L_a \ m}{J}\beta \ - \ 2 \ L_b F_{y,r} \tag{7}$$

The reference trajectories are given by bounded trajectories $x_r(t), \ y_r(t)$. Our objective is to track them asymptotically.

## Known Parameter Case

We have the tracking error for the system,

$$e_1 = x - x_r \tag{8}$$

$$e_2 = y - y_r \tag{9}$$

$$\dot{e_1} = vsin\theta - \dot{x_r} \tag{10}$$

$$\dot{e_2} = vcos\theta - \dot{y_r} \tag{11}$$

Let $v_{des}$ and $\theta_{des}$ be the desired values of $v$ and $theta$, so they should satisfy:

$$\dot{e_1} \ = \ v_{des} \ sin\theta_{des} \ - \ \dot{x_r} \ := \ -k_1 e_1 \tag{12}$$

$$\dot{e_2} \ = \ v_{des} \ cos\theta_{des} \ - \ \dot{y_r} \ := \ -k_2 e_2 \tag{13}$$

We define the backstepping variables, we will choose appropriate expression for $\phi_{des}$ later

$$z_1 = v \ - \ v_{des} \Longrightarrow \ \dot{z_1} \ = \ \alpha \ - \ \dot{v_{des}} \tag{14}$$

$$z_2 = \ \theta \ - \ \theta_{des} \ \Longrightarrow \ \dot{z_2} \ = \ \phi \ - \ \dot{\theta_{des}} \tag{15}$$

$$z_3 = \phi - \phi_{des} \implies \dot{z_3} = \frac{L_a}{J}\beta - 2L_bF_{y,r} - \dot{\phi_{des}} \tag{16}$$

We define candidate lyapunov function as,

$$V_1 = \frac{1}{2}\left(e_1^2 + e_2^2 + z_1^2 + z_2^2\right)$$

We have,

$$\dot{V_1} = e_1\dot{e_1} + e_2\dot{e_2} + z_1\dot{z_1} + z_2\dot{z_2}$$

$$\begin{aligned}\dot{e_1} &= v\ sin\theta - \dot{x_r}\\ &= v_{des}\ sin\theta + z_1\ sin\theta - \dot{x_r}\\ &= v_{des}\ sin\theta_{des}cosz_2 + v_{des}cos\theta_{des}\ sinz_2 + z_1sin\theta - \dot{x_r}\end{aligned}$$

In order to proceed further, we have no choice but to assume that $sinz_2 \to z_2,\ cosz_2 \to 1$

Therefore, we obtain, $\dot{e_1} = -k_1e_1 + v_{des}\ cos\theta_{des}\ z_2 + z_1sin\theta$

Similarly, $\dot{e_2} = -k_2e_2 + z_1cos\theta - z_2\ v_{des}\ sin\theta_{des}$

$$\begin{aligned}\dot{V_1} &= e_1\dot{e_1} + e_2\dot{e_2} + z_1\dot{z_1} + z_2\dot{z_2}\\ &= -k_1e_1^2 + -k_2e_2^2 + z_1(\dot{z_1} + e_1sin\theta + e_2cos\theta) + z_2(\dot{z_2} + e_1v_{des}cos\theta_{des} - e_2v_{des}sin\theta_{des})\end{aligned}$$

We now choose $\alpha$ to introduce stabilizing terms so that,

$$\dot{z_1} = \alpha - \dot{v_{des}} = -k_3z_1 - (e_1sin\theta + e_2cos\theta)$$

for some $k_3 > 0$

We thus obtain,

$$\alpha = \dot{v_{des}} - k_3z_1 - (e_1sin\theta + e_2cos\theta) \tag{17}$$

Similarly we chose $\phi_{des}$ so that, $\dot{z_2} = \phi_{des} - \dot{\theta_{des}} = -k_4z_2 - (e_1v_{des}cos\theta_{des} - e_2v_{des}sin\theta_{des})$

Therefore,

$$\phi_{des} := \dot{\theta_{des}} - k_4z_2 - (e_1v_{des}cos\theta_{des} - e_2v_{des}sin\theta_{des}) \tag{18}$$

We define, $z_3 = \phi - \phi_{des} \implies \dot{z_3} = \frac{L_a\ m}{J}\beta - 2L_bF_{y,r} - \dot{\phi_{des}}$

Now to drive $z_3 \to 0$, we define the final candidate Lypapunov function as,

$$\begin{aligned}V_{final} &= V_1 + \frac{1}{2}z_3^2\\ &= \frac{1}{2}\left(e_1^2 + e_2^2 + z_1^2 + z_2^2 + z_3^2\right)\\ \dot{V_{final}} &= -k_1e_1^2 - k_2e_2^2 - k_3z_1^2 + z_2(-k_4z_2 + z_3) + z_3\dot{z_3}\\ &= -k_1e_1^2 - k_2e_2^2 - k_3z_1^2 - k_4z_2^2 + z_3(\dot{z_3} + z_2)\end{aligned}$$

We choose $\beta$ so that, $\dot{z}_3 = \frac{L_a\ m}{J}\beta - 2L_bF_{y,r} - \dot{\phi_{des}} = -k_5z_3 - z_2$ for some $k_5 > 0$
Thus we obtain,

$$\beta = \frac{J}{m\ L_a}\left(2L_bF_{y,r} + \dot{\phi_{des}} - k_5z_3 - z_2\right) \tag{19}$$

## Adaptation Law

We have $C_y, m, J$ as unknown parameters which do not appear linearly in the dynamics. We employ tuning function method to construct the adaptation law. The main idea is that if we have a dynamics of the form

$$\dot{\boldsymbol{x}}\ =\ F(\boldsymbol{x},\ u_1,\ u_2)\ +\ G(\boldsymbol{x},\ u_1,\ u_2)\ \Theta$$

where $\Theta$ is the unknown parameter vector. Also if we have an asymptotically stable design for the known parameter case i.e $V_c(\boldsymbol{x}, \Theta) \leq -W(\boldsymbol{x}, \Theta)$ for some $W > 0$ for all $\Theta$.

For the unknown parameter case we define new Lyapunov Function as,

$$V\ =\ V_c(\boldsymbol{x},\ \hat{\Theta})\ +\ \frac{1}{2}\ \tilde{\Theta}^T\Gamma^{-1}\tilde{\Theta}$$

Therefore,

$$\dot{V} = \frac{\partial V_c}{\partial x}(F(\boldsymbol{x},\ u_1,\ u_2) + G(\boldsymbol{x},\ u_1,\ u_2)\ .\ \hat{\Theta}) + \frac{\partial V_c}{\partial x}G(\boldsymbol{x},\ u_1,\ u_2)\tilde{\Theta} + \frac{\partial V_c}{\partial \hat{\Theta}}\dot{\hat{\Theta}}\ - \tilde{\Theta}^T\Gamma^{-1}\tilde{\Theta}$$

where,
$\hat{\Theta}$ represents the estimated parameters,
$\tilde{\Theta}$ represents parameter error and
$\dot{\hat{\Theta}} =\ \Gamma\ \tau(\boldsymbol{x},\ \hat{\Theta})$ for some positive define matrix $\Gamma$

From the known case our Lyapunov function is independent of $\Theta$ i.e $\frac{\partial V_c}{\partial \Theta} = 0$, if we set

$$\tau(\boldsymbol{x},\ \hat{\Theta}) = \left(\frac{\partial V_c}{\partial x}\ G(\boldsymbol{x},\ u_1,\ u_2)\ \right)^T$$

Then, $\dot{V}\ \leq -W(\boldsymbol{x}, \hat{\Theta})$. Thus asymptotic stability can be proven using the LaSalle Yakashinov Theorem.
Therefore the parameter update law is given by,

$$\dot{\hat{\Theta}} =\ \Gamma\left(\frac{\partial V_c}{\partial x}\ G(\boldsymbol{x},\ u_1,\ u_2)\ \right)^T$$

For our system, we carefully factorize the dynamics to get,

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{v}\\ \dot{\theta}\\ \dot{\phi}\end{pmatrix} = \begin{pmatrix}v\ sin(\theta)\\ v\ cos(\theta)\\ a\ cos(\delta)\\ \phi\\ 0\end{pmatrix} + \begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ -2\left(\delta\ -\ L_a\frac{\phi}{v}\right) & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & L_aa\ sin(\delta) & 2L_a\left(\delta\ -\ L_a\frac{\phi}{v}\right)cos(\delta) & -2\ L_b^2\ \frac{\phi}{v}\end{pmatrix}\begin{pmatrix}C_y/m\\ m/J\\ C_y/J\\ C_y\end{pmatrix}$$

Thus, we have overparametrized the unkowns but we do not care as we don't have to estimate the parameters.
All the calculations for implementation are carried out using the symbolic toolbox of matlab and then copy-pasted in appropriate functions. As the expressions are explosive, nothing is presented here. Please refer to our github repository for the codes.

## Simplications to control design

Obtaining $a, \delta$ from $\alpha, \beta$ is a computationally heavy operation and it is not possible to perform simulations as the iterative solver slows down or stops the simulations with errors.
Hence, we simplify our problem and assume that $\alpha, \beta$ are our controls and thus consider (10 - 14) as our state space model. For this model we assume that $\frac{m}{J}$ and $C_y$ are unknown parameters and derive an adaptive control law exactly similar to that explained above.
We obtain

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{v} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} v\ sin(\theta) \\ v\ cos(\theta) \\ \alpha \\ \phi \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ L_a\ \beta & \frac{-2L_b^2\ \phi}{v} \end{pmatrix} \begin{pmatrix} m/J \\ C_y \end{pmatrix}$$

## Simulations

We have performed majorly simulations in which the system tracks a ramp input. We have always kept the initial position of the car at the origin and the reference trajectory is the line $y = x$. This implies that $\theta_{ref} = \pi/4$. We vary the initial orientation with each simulation and also the reference velocity to be tracked. Such a simulation tells us that if a vehicle is moving at a constant velocity on a straight line, can it track a change in its trajectory by a constant angle.
We first present the known parameter simulations.

## Known Parameter Case Saturated Inputs

As in most of the simulations the control inputs have extremely high values in the beginning, we have added a saturation block to limit the value for this simulation.

Ramp Input : $x_r(t) = 10t, y_r(t) = 10t$
Initial Conditions $t = 0$ : $x(0) = 0, y(0) = 0, v(0) = 10m/s$, $\theta(0) = 0.7rad$

Here, in figure 2 we observe that the errors oscillate and dampen and hence it takes a long time for the error to die to zero. This is because of saturating the control input. However we are able to achieve asymptotic tracking of the ramp input. There is a mismatch of $pi/4 - 0.7 = 5degrees$ which the system is able to compensate for while moving at 10m/s with saturated control inputs.

## Known Parameter Circular Trajectory

Sinusoidal Input : $x_r(t) = 100cos(0.01t), y_r(t) = 100sin(0.01t)$
Initial Conditions : $x(0) = 100, y(0) = 0, v(0) = 1m/s,\ \theta(0) = 2\pi rad, \phi(0) = -0.01$
Tuned Gains : $k_1 = 0.2, k_2 = 2.0, k_3 = 1.0, k_4 = 1.0, k_5 = 1.0$

In this simulation we demonstrate the ability of the system to track a circular trajectory. The error does not die to zero but attains a constant value asymptotically. The results are visible in figures 3 and 4.

## Unknown Parameter Case

We have assumed two unknown parameters with true values : $\frac{m}{J} = 0.5$ and $C_y = 11500$. We initialize our estimators with $\frac{m_o}{J_o} = 0.4$ and $C_{yo} = 11000$ and demonstrate that system can track a change of atmost 15 degrees when moving at 10m/s.

### 10 Degree Mismatch Unknown Parameters

Ramp Input : $x_r(t) = 10t, y_r(t) = 10t$
Initial Conditions $t = 0$ : $x(0) = 0, y(0) = 0, v(0) = 10m/s,\ \theta(0) = 0.6rad$

We have a mismatch of $\pi/4 - 0.6 = 10degrees$. However the system is able to track the reference trajectory. The plots are shown in figure 5. It takes about 4 seconds for the errors to reach negligible values.

### 15 Degree Mismatch Unknown Parameters

Ramp Input : $x_r(t) = 10t, y_r(t) = 10t$
Initial Conditions $t = 0$ : $x(0) = 0, y(0) = 0, v(0) = 10m/s,\ \theta(0) = 0.5rad$

This is simulation is similar to the previous one, but we are able to track a mismatch of about 15 degrees. The errors drop to to negligible values in about 6 seconds. The plots are shown in figure 6.
All the simulations were done in simulink and figure 7 shows the model used by us.

# Backstepping Control for Rickshaw

In order to explore the control of nonholonomic systems we present the control design of a 3 wheeled robot (Rickshaw). We present the design followed in [1]. The motivation for doing this is that this system is very similar to our system and hence we could use their results in our design. However there are a few minute errors in the paper that are clarified here, we also explain why the results mentioned in [1] cannot be directly applied for our system. The state space model is as follows

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{pmatrix} = \begin{pmatrix}cos\theta & -d\ sin\theta\\ -sin\theta & d\ cos\theta\\ 0 & 1\end{pmatrix}\begin{pmatrix}v_1\\ v_2\end{pmatrix} \tag{20}$$

One would be tempted to think that if we put $d = 0$ then this reduces to the initial part of our original system. However there is a subtle reason why putting $d = 0$ in the results of the paper doesn't work for our system.(Presented below)
The reference trajectories to be tracked are given by bounded trajectories $x_r(t),\ y_r(t),\ \theta_r(t)$

$$\dot{x_r} = v_r\ cos(\theta_r)$$
$$\dot{y_r} = v_r\ sin(\theta_r)$$
$$\dot{\theta}_r = \omega_r$$

We carefully choose the tracking error for the system as

$$e = \begin{pmatrix}e_1\\ e_2\\ e_3\end{pmatrix} = \begin{pmatrix}cos\theta & sin\theta & 0\\ -sin\theta & cos\theta & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x_r - x\\ y_r - y\\ \theta_r - \theta\end{pmatrix} \tag{21}$$

This error dynamics is the first trick in their design which allows the use of back-stepping. As the matrix is invertible, $e$ going to zero is equivalent to achieving tracking.

Computing the derivative we obtain,

$$\dot{e} = \begin{pmatrix}v_2e_2 + v_rcos(e_3) - v_1\\ -v_2e_1 + v_rsin(e_3) - dv_2\\ \omega_r - v_2\end{pmatrix} \tag{22}$$

Note, that $-dv_2$ term is missing in the original paper, which is crucial in implementation and is the reason why the results cannot be blindly applied to our system.(Explained below)

If we choose the auxiliary control input $v_c$ such that for some $k_1,\ k_2,\ k_3 > 0$,

$$v_c = \begin{pmatrix}v_rcos(e_3) + k_1e_1\\ w_r + k_2v_2e_2 + k_3v_rsin(e_3)\end{pmatrix} \tag{23}$$

Now if we assume that $v_c = v$ then the proof for asymptotic stability of the error is as follows:
Consider the Lyapunov Function

$$V = k_1\left(e_1^2 + e_2^2\right) + 2\frac{k_1}{k_2}(1 - cos(e_3))$$

Upon taking the derivative and putting $v_c = v$ we obtain,

$$\dot{V} = -2k_1^2e_1^2 - 2k_1de_2(w_r + k_2v_re_2 + k_3v_3sin(e_3)) - 2\frac{k_1k_3}{k_2}v_rsin(e_3)^2$$

Note that if $d = 0$ then the above Lyapunov function becemes negative semidefinite and would not guarantee asymptotic stability. However if it is not 0, then we use completing the squares and choose the gains appropriately to ensure asymptotic stability.

The rest of the formulation is same as presented in the paper and hence not reproduced here. The key result is that $u = \dot{v_c} + k_4 I_{2\times 2}\,(v_c - v)$ ensures asymptotic tracking of all three state variables.
We have implemented the above control design in simulink without any adaptation due to lack of time. The codes can be found at our github repository. We have shown the response of the system while tracking a ramp input in Figure 8.
The system is unable to drive the error to zero inspite of proving asymptotic stability theoretically. The reason for this still has to be explored.

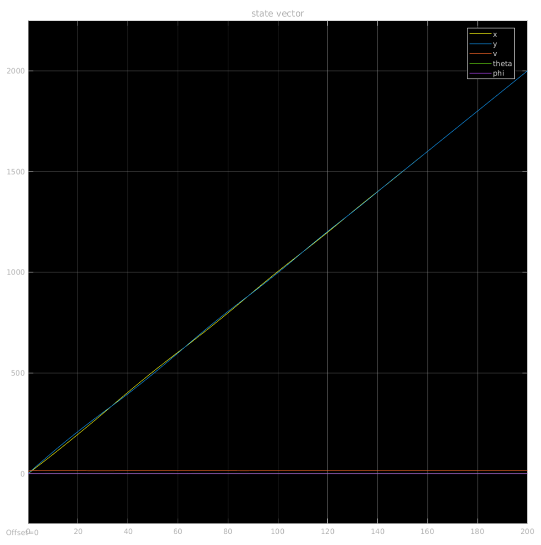
## Conclusion

We have presented the control design for the original system using backstepping and making an assumption that $z2$ which is the angle backstepping variable remains very small. However, one must note that this is not a bad assumption as we can always plan the reference trajectory in a way that the angles are almost matched
We have also simplified our design and used $\alpha, \beta$ as control inputs. We have demonstrated that the system is able to track rack inputs very well with an initial angle mismatch. However the control inputs grow very large and thus such a design is not physically implementatble. We have also presented how the system would behave if we use a saturation block at the control output for the known parameter case. We have also demonstrated that the system can track a circular trajectory reasonably well.
Finally, we have presented the control design for a 3 body vehicle from [1]. Here we have shown why the results are not applicable to our system. We have also implemented the control design from [1] and tested its performance while tracking ramp inputs.

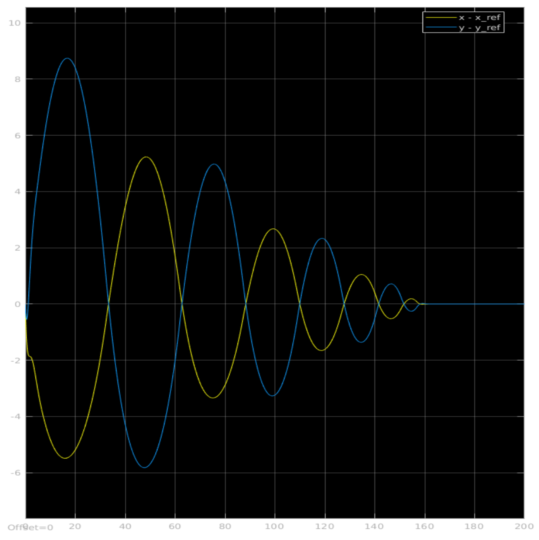
## References

1] R. Fierro and F. Lewis, “Control of a nonholonomic mobile robot:backstepping kinematics into dynamics,” J. Robot. Syst., vol. 14, no. 3,pp. 149–163, 1997

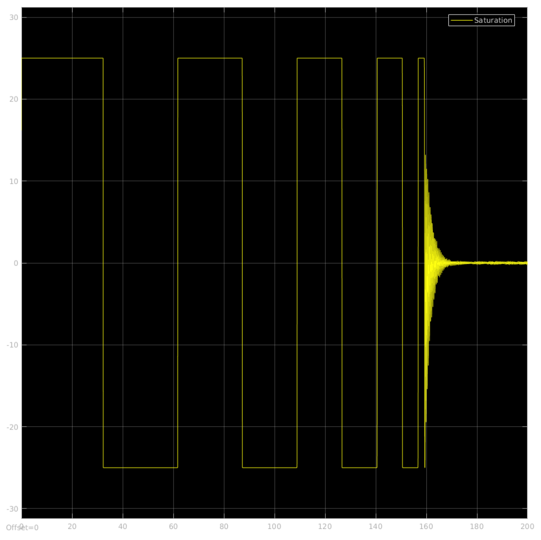
2] Github Repository



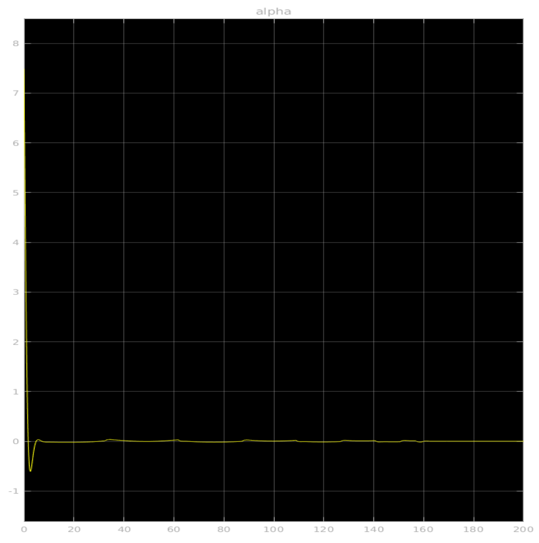
(a) Evolution of States



(b) Evolution of errors

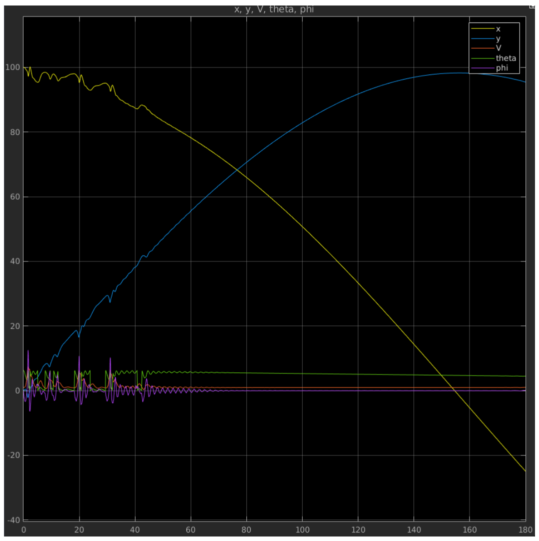


(c) Evolution of input $\beta$

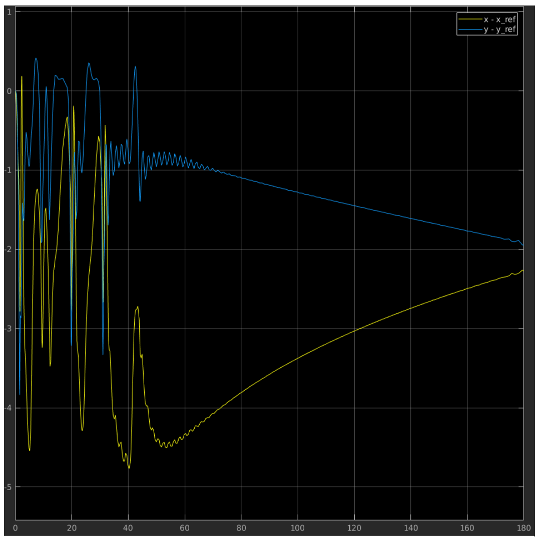


(d) Evolution of input $\alpha$

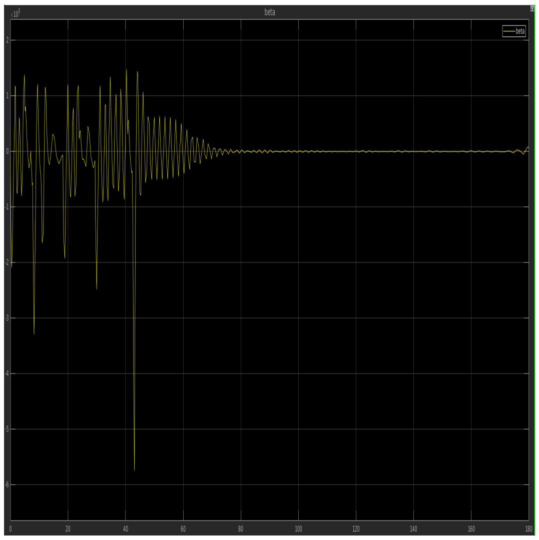
Figure 2: Ramp Input for Known Parameter Case and Saturated Inputs



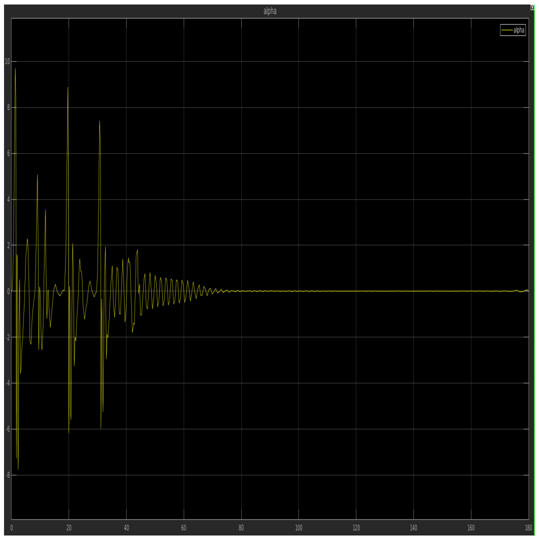
(a) Evolution of States



(b) Evolution of errors



(c) Evolution of input $\beta$



(d) Evolution of input $\alpha$

Figure 3: Circular Input for known Parameter Case

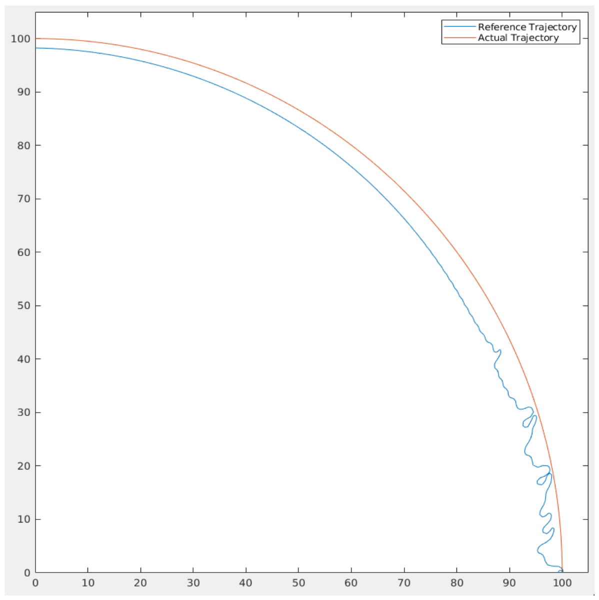
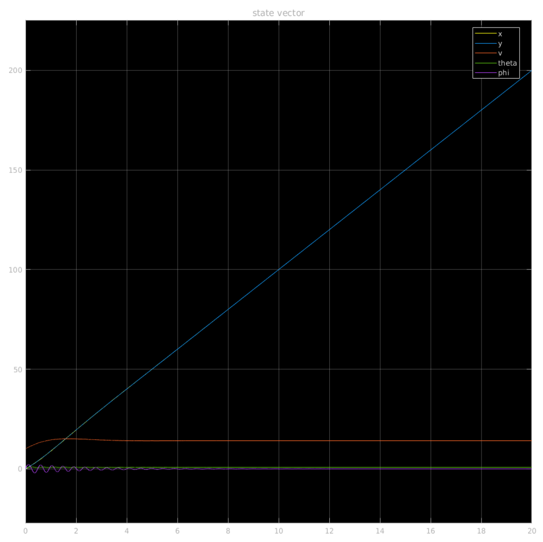
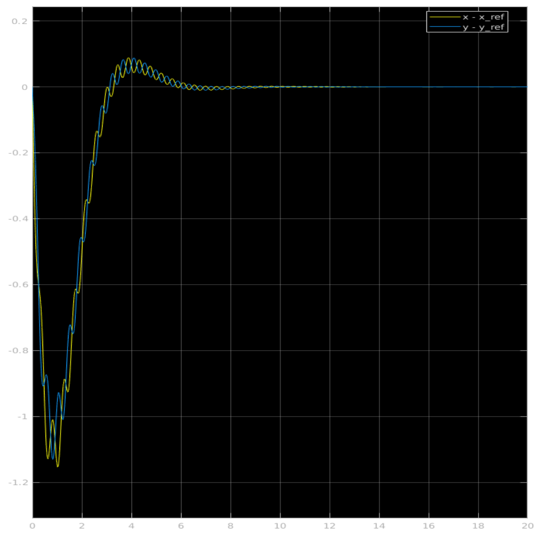


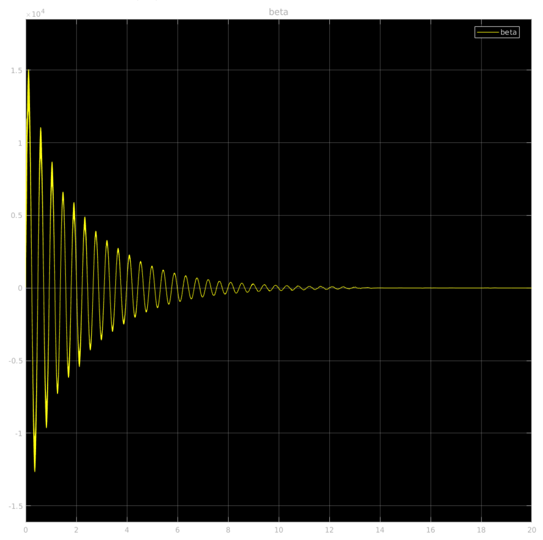
Figure 4: Performance Comparison of Reference Trajectory Tracking



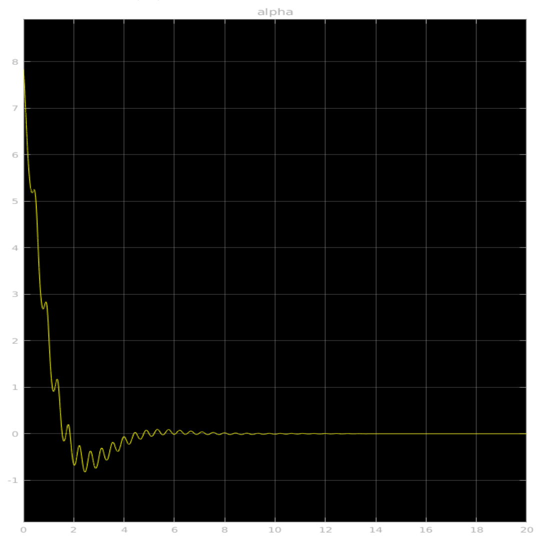
(a) Evolution of States



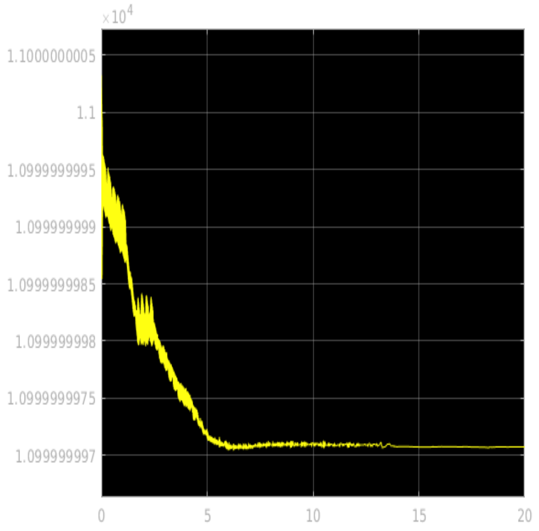
(b) Evolution of errors



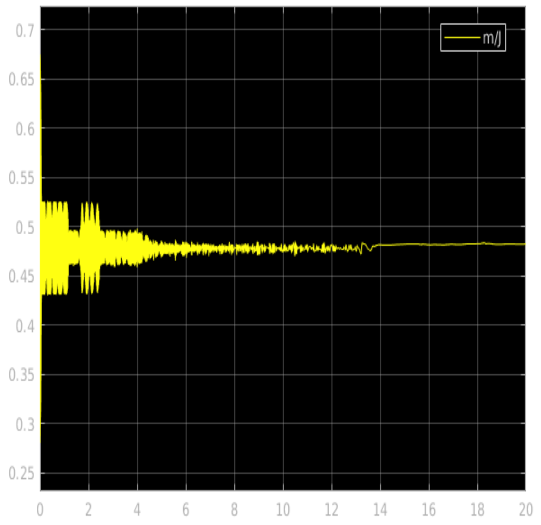
(c) Evolution of input $\beta$



(d) Evolution of input $\alpha$

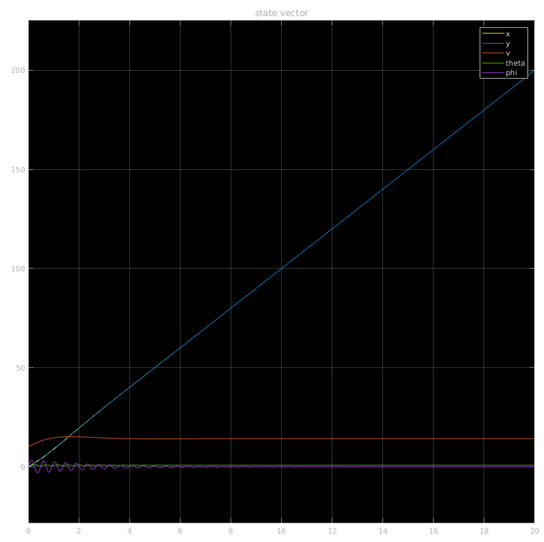


(e) Evolution of estimate $C_y$

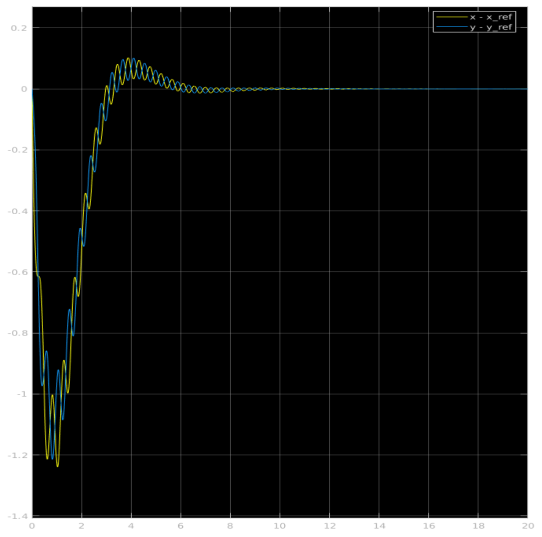


(f) Evolution of estimate $m/J$

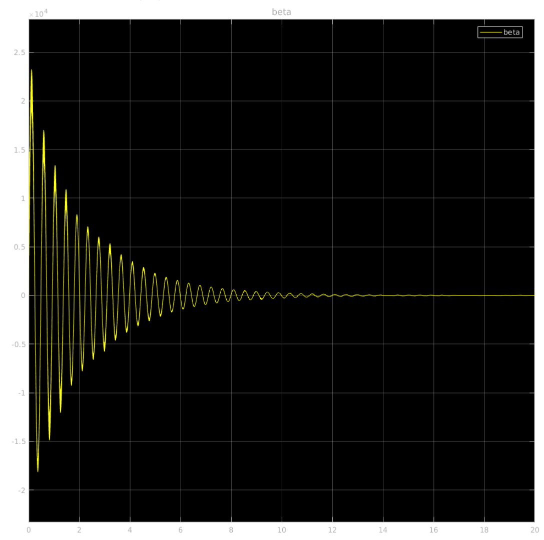
Figure 5: Ramp Input with 10 degree initial heading angle difference



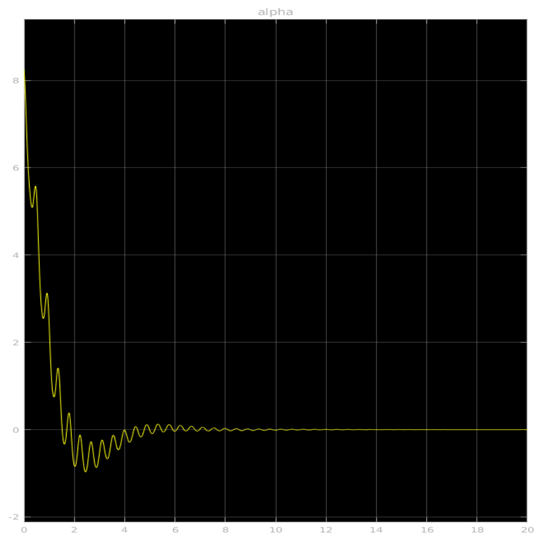
(a) Evolution of States



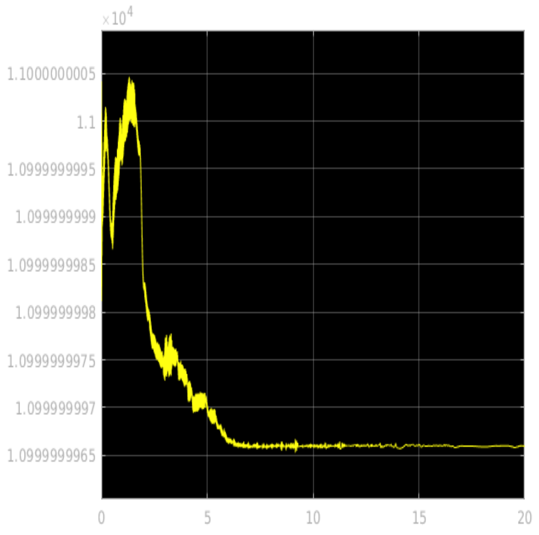
(b) Evolution of errors



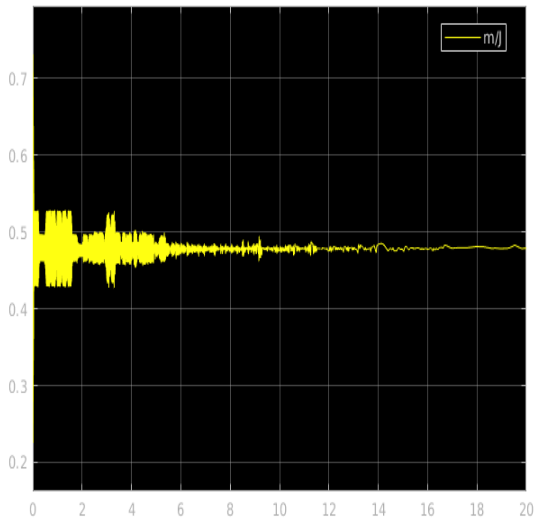
(c) Evolution of input $\beta$



(d) Evolution of input $\alpha$



(e) Evolution of estimate $C_y$



(f) Evolution of estimate $m/J$

Figure 6: Ramp Input with 15 degree initial heading angle difference

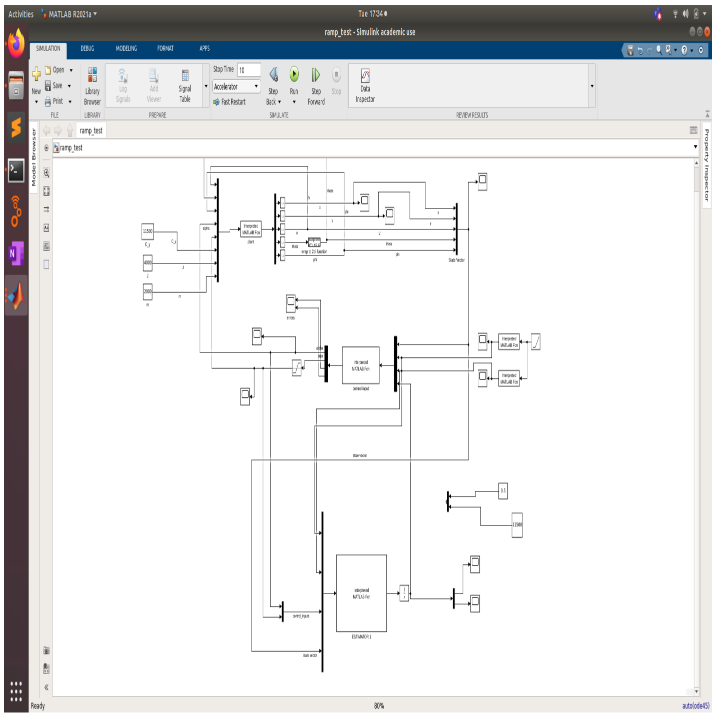
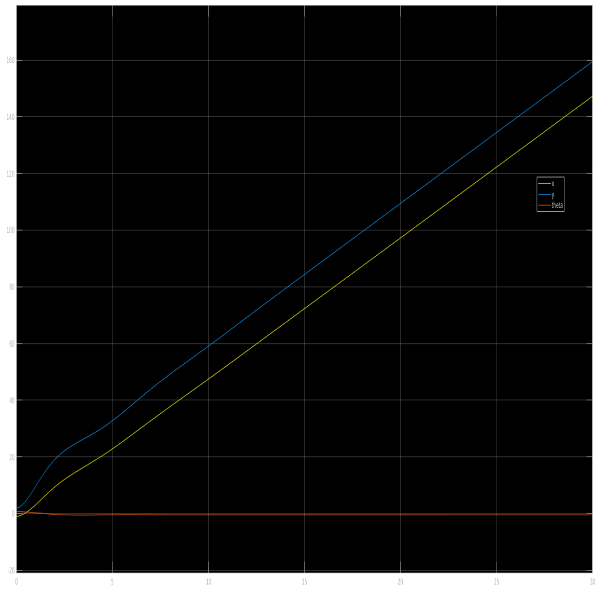
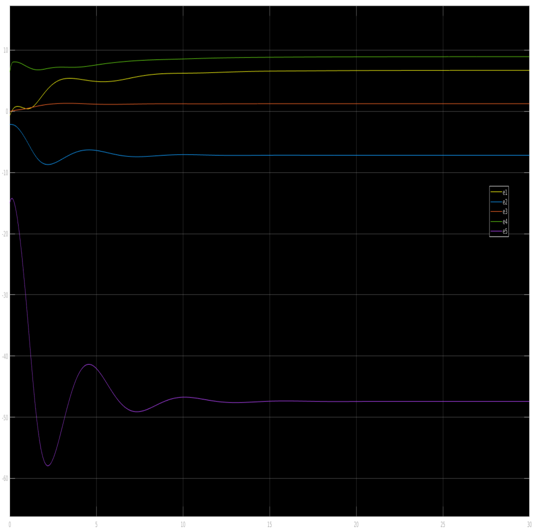


Figure 7: Simulink Model



(a) Evolution of States with time



(b) Evolution of errors with time

Figure 8: Non-holonomic Control Ramp Response Known Case