Autonomous Driving Mobility Assignment

16-665: Robot Mobility on Air, Land, and Sea

Assignment Parameters

This is an individual assignment. Conceptual collaboration is encouraged but students may not exchange work or code nor copy information from another source. Cite all references. This assignment is worth 20% of the total grade for this class. Please type or neatly write your solutions. Submission will be in .pdf form. Points will be given for correct work and correct solutions.

Learning Objectives

1. Use analytic techniques to make informed parametric comparisons.
2. Exercise vehicle control strategies in an example maneuver.

Submission Instructions

- Create a single .pdf file with your solutions for all the sections named hw1.pdf.
- Ensure that your .pdf contains all work, explanations, images, and codes for all the questions.
- Submit to Canvas using the proper assignment link by the date specified in the course schedule.

Contents

1 Kinematic Bicycle Model **3**
- 1.1 Pepy Model [1 point] . . . 4
- 1.2 Kong Model [1 point] . . . 4
- 1.3 Discussion [2 points] . . . 4
- 1.4 Implementation [3 points] . . . 4

2 Dynamic Bicycle Model **6**
- 2.1 Model Set Up [0 points] . . . 6
- 2.2 Lane Change [2 points] . . . 6
- 2.3 Desired and Actual Plot [1 point] . . . 7
- 2.4 Curves [2 points] . . . 8
- 2.5 Desired and Actual Plot [1 points] . . . 8
- 2.6 Discussion [1 point] . . . 8

3 Pure Pursuit Controller **9**
- 3.1 Pure Pursuit Controller [1 point] . . . 10
- 3.2 Plot Result[2 points] . . . 11
- 3.3 Discussion[1 point] . . . 11

Bibliography **12**

1 Kinematic Bicycle Model

In class, we derived the full kinematic bicycle model (KBM) based on Rajamani, Chapter 2. The resultant model and accompanying diagram are given below.

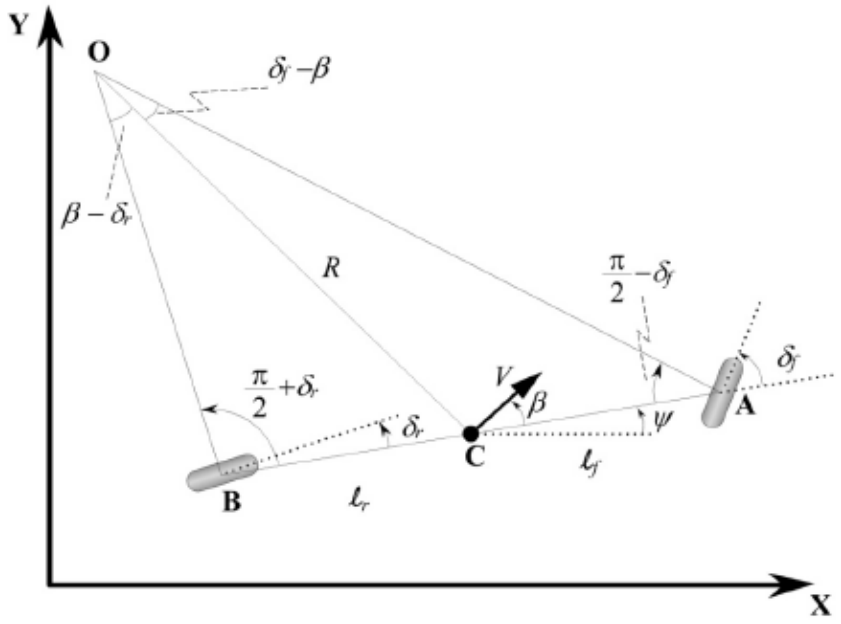

Figure 1: Kinematic Bicycle Model.

$$\dot{X} = V cos(\psi + \beta) \tag{1}$$

$$\dot{Y} = V sin(\psi + \beta) \tag{2}$$

$$\dot{\psi} = \frac{V cos\beta}{l_f + l_r}(tan\delta_f - tan\delta_r) \tag{3}$$

The KBM is usually given in simplified form in research papers. Two examples are given below from Pepy [1] and Kong [2]. The Pepy model measures the position of the vehicle at the rear wheels (Point B in the above diagram), whereas the Kong model uses the center of gravity (Point C in the above diagram). In both cases, a vehicle without steered rear wheels is considered. Derive these models, either by simplifying the Rajamani model, or from first principles. State all assumptions and show all steps.

1.1 Pepy Model [1 point]

The Pepy model:

$$\dot{X} = V cos\psi \tag{4}$$

$$\dot{Y} = V sin\psi \tag{5}$$

$$\dot{\psi} = \frac{V}{l_f + l_r} tan\delta_f \tag{6}$$

1.2 Kong Model [1 point]

The Kong Model:

$$\dot{X} = V cos(\psi + \beta) \tag{7}$$

$$\dot{Y} = V sin(\psi + \beta) \tag{8}$$

$$\dot{\psi} = \frac{V}{l_r} sin\beta \tag{9}$$

1.3 Discussion [2 points]

A. The vehicle slip angle at the center of mass β no longer appears in the Pepy model. Does this mean that the vehicle slip angle is zero? If not, write an equation for it in terms of known/measurable vehicle parameters/variables.

B. What is the difference between vehicle slip and tire slip? Are they related to one another? Can you have one without the other?

1.4 Implementation [3 points]

Implement the Pepy model in Matlab or your favorite software platform with velocity V equal to a constant and zero initial conditions for X, Y, and ψ. Let $l_r = l_f = 1.5$ meters. Use the three different steering inputs δ listed below. Provide one plot and one or two sentences to answer the questions for each part.

A δ = a non-zero constant. Try various constants. Write a mathematical expression for the relationship between the steering angle and the path radius or curvature.

B δ = a sinusoid. Before doing this, try to predict what the trajectory will look like. Did it come out as you expected? Why or why not? Try different velocities V and different amplitudes for the steering sinusoid and comment on the effects.

C δ = a square wave. Why is this unrealistic? How would you handle this lack of realism in the model?

INCLUDE YOUR CODE AT THE END OF YOUR ASSIGNMENT

2 Dynamic Bicycle Model

In class we derived the Dynamic Bicycle Model lateral dynamics based on lateral position error e_1 and yaw angle error e_2:

$$\frac{d}{dt}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix} \begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix} \dot{\psi}_{des} \quad (10)$$

2.1 Model Set Up [0 points]

Implement this model in Matlab or your favorite software platform. Use the following typical vehicle parameters: $V_x = 30\ m/sec$, $m = 1573\ kg$, $I_z = 2873\ kg - m^2$, $l_f = 1.1\ m$, $l_r = 1.58\ m$, $C_{\alpha f} = C_{\alpha r} = 80,000\ N/rad$.

The open-loop system, which can be written as $\dot{x} = Ax + B_1\delta + B_2\dot{\psi}_{des}$, has two eigenvalues at the origin and is therefore unstable. You can use state feedback to form the steering input $\delta(t) = -Kx$, leading to $\dot{x} = (A - B_1K)x + B_2\dot{\psi}_{des}$ for the closed-loop system.

2.2 Lane Change [2 points]

For this task, you are asked to switch from following a straight line below the global X-axis to following a straight line along the X-axis.

The vehicle's initial conditions are (X,Y) = (0,-5m) with ψ=0. Vx=30 m/sec as above and remains constant. (In other words, 5m below the origin pointing and traveling in the X-direction.) The vehicle's final conditions are (X,Y) = (100m,0m) with ψ=0. Use a diagonal line to connect the desired Y coordinates over 90 meters in the x axis. The vehicle must stay at Y = -5m until it reaches the diagonal portion of the path at X = 5m. Make sure to add a trajectory along the x-axis at the end so you can examine the maximum error and settling time.

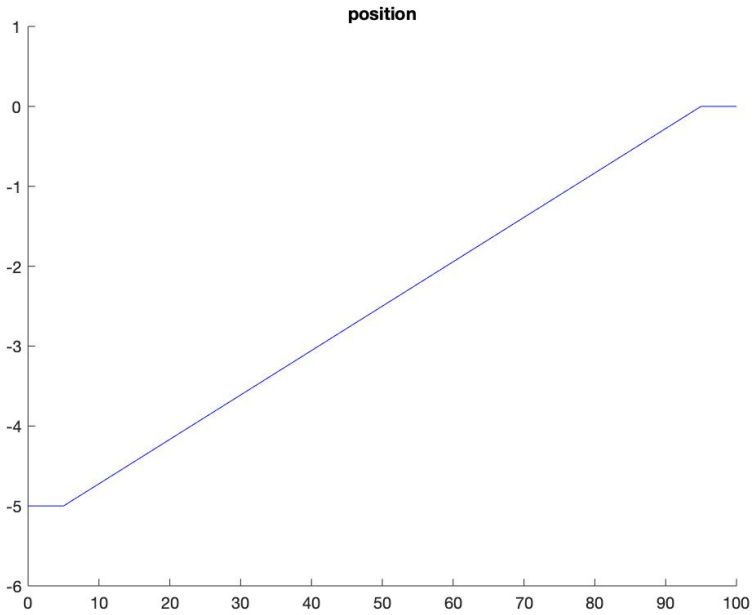

Figure 2: Desired vehicle trajectory

Create a series of inputs $\dot{\psi}_{des}$ that follows this path and use pole placement or LQR to achieve a maximum error abs(e2)$\leq$0.01 rad and reach within abs(e1)$\leq$0.002 m and abs(e2)$\leq$0.0007 rad within 1 second at the transition points (i.e., leaving the initial lane and entering the new lane). State whether you used pole placement or LQR and list the resultant poles. The time step of your simulation should be 0.01 seconds or larger, and the derivative of the steering input $\frac{\delta(t)}{dt}$ should not exceed 25 $\frac{rad}{sec}$. Please include a plot of $\frac{\delta(t)}{dt}$ over the course of the lane change maneuver.

Plot the resultant lateral position error e_1 and yaw angle error e_2 versus time, and make it clear that your simulation meets all of the specifications.

2.3 Desired and Actual Plot [1 point]

Plot the true global vehicle path in the XY plane, along with the desired vehicle path. Show separate plots zooming in on the transition points (i.e., leaving the initial lane and entering the new lane).

2.4 Curves [2 points]

Create an input $\dot{\psi}_{des}$ corresponding to following a straight path for 1 second, then a positive-curvature circular arc with radius 1000 m for 5 seconds, then a straight path for 1 second, and finally a negative-curvature circular arc with radius 500 for 5 seconds.

Use pole placement or LQR to achieve a maximum error abs(e1)$\leq$0.01 m and abs(e2)$\leq$0.01 rad. State whether you used pole placement or LQR and list the resultant poles.

Plot the resultant lateral position error e_1 and yaw angle error e_2 (these should both converge near 0 when the $\dot{\psi}_{des}$ is 0) versus time.

2.5 Desired and Actual Plot [1 points]

Plot the true global vehicle path in the XY plane, along with the desired vehicle path.

2.6 Discussion [1 point]

Create error plots as in Subsection 2.4 with two different values for V_x. Describe the effect on e_1 and e_2 in one to two sentences.

INCLUDE YOUR CODE AT THE END OF YOUR ASSIGNMENT

3 Pure Pursuit Controller

In the class we learned the basics of a Pure Pursuit Controller. Pure pursuit is the geometric path tracking controller that tracks a reference path using only the geometry of the vehicle kinematics. This tracking is done based on a look-ahead point on the reference path. Thus, the pure pursuit controller computes the steering angle required to move the robot from its current position to reach this look-ahead point in front of the robot.

For this assignment, please download the file "purepursuit_16665.py". A reference trajectory will be given which must be tracked using a PID controller and a pure pursuit controller. The PID controller which tracks the target velocity has been implemented in the starter code given. You will have to implement the pure pursuit controller that computes the steering angle required to reach each target point on the reference trajectory.

The Pure Pursuit Algorithm is as follows:

1. Determine the current location of the vehicle.
2. Find the target point (based on the look-ahead distance) on the trajectory.
3. Calculate the curvature to reach the target point from the current location.[reference:[3]]
4. Find the vehicle steering angle to obtain the desired path curvature.
5. Update the vehicle position.

Refer to the following image for better understanding of the pure pursuit controller:

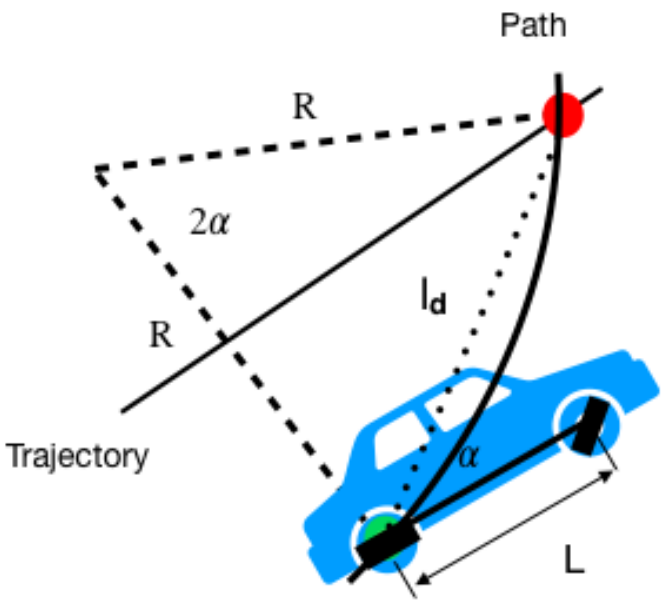

Figure 3: Pure pursuit geometric relationship
Source: Pure pursuit controller

The first two points of the algorithm have been implemented in the starter code. The last three points have to be implemented under the TODO sections which are the following sections of the code:

1. Update method inside the Vehicle class
2. PurePursuitcontrol method inside the Controller class
3. Yaw error inside the main method

The reference trajectory as well as the final trajectory followed by the vehicle have to be plotted. The code for plotting has been implemented.

3.1 Pure Pursuit Controller [1 point]

List the equations used to implement the pure pursuit controller.

3.2 Plot Result[2 points]

Submit the screenshot of the plot showing the reference trajectory and the final trajectory of the vehicle.

3.3 Discussion[1 point]

Vary the look-ahead distance parameter and observe the performance of the pure pursuit controller. Submit the corresponding plots for three look-ahead distances. Discuss the effect of the look-ahead distance on the pure pursuit controller.

INCLUDE YOUR CODE AT THE END OF YOUR ASSIGNMENT

Bibliography

[1] *Path Planning Using a Dynamic Vehicle Model*. Proceedings of the 2nd International Conference on Information & Communication Technologies. 2006. URL: https://ieeexplore.ieee.org/abstract/document/1684472/.

[2] *Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design*. Proceedings of the IEEE Intelligent Vehicles Symposium. 2015. URL: https://ieeexplore.ieee.org/document/7225830/.

[3] *Implementation of the Pure Pursuit Path Tracking Algorithm*. The Robotics Institute, Carnegie Mellon University. 1992. URL: https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf.