ME227: Vehicle Dynamics and Control **Spring 2021**

Assignment 1: The Kinematic Model

Due Thursday, April 8 at 5:00pm PDT

Purpose

There is a lot we can learn about the design and control of vehicles using simple models. To start off, we have a few exercises to familiarize yourself with the notation of coordinate transformations and rotation matrices introduced on the first day of class and to gain some more insight into the kinematic model. You will also build a simple simulation and test controllers capable of tracking a path with the kinematic model.

Instructions

This and following homework assignments will be submitted using two different tools, Gradescope and MATLAB Grader. Throughout the assignment, each prompt will be marked with either **Gradescope** or **MATLAB Grader** to make it clear where you should be submitting the answer to that problem. MATLAB Grader questions will also be shown in **Blue** to make them easy to see.

All written portions must be turned in through Gradescope. We will provide three different homework formats on Canvas (a latex template, a pdf template, and a condensed pdf template). Whatever format you decide to use, please **BOX** all of your final answers.

Some problems will make use of the MATLAB Grader suite discussed in class. These problems are available directly from Canvas by clicking on each individual question. You are allowed to submit code to MATLAB Grader as many times as you want before the due date without penalty. In this way you can be sure each function or script passes all of the assessments that go along with it before moving on to the next problem. We encourage you to write your own test cases as well to ensure your code is working as expected.

Problem 1 – Rotation Matrices

Rotation matrices are an important component of any control of dynamical systems. They provide easy transformations of locations, velocities, and commands that occur in different frames. This problem will help you gain some intuition into rotation matrices as you practice creating them.

Question 1.A – Combining Rotation Matrices (Gradescope)

Show that rotating from the vehicle frame to the inertial frame and then into the path frame is equivalent to rotating directly from the vehicle frame to the path frame. In other words, demonstrate the result we used in class that

$$A_{PO}A_{OV} = A_{PV}$$

Write the expression showing $A_{PO}A_{OV} = A_{PV}$***. Please include your work.***

Question 1.B – Creating Rotation Matrices (MATLAB Grader)

Implement a function that returns the rotation matrix A_{vo} for some heading angle Ψ_{VE}. You can use this function to translate a point from the inertial frame, O, to the vehicle frame, V.

Problem 2 – Lateral Velocity and Sideslip Angle

In the kinematic bicycle model, the lateral velocity at the rear axle is zero under the assumption that the tire velocity is aligned with the tire's longitudinal axis. The lateral velocity is nonzero at other points along the centerline of the kinematic vehicle, however.

Question 2.A – Velocity (Gradescope)

Derive an expression for the velocity at any point along the centerline of the kinematic model starting at the rear axle ($x = 0$) and ending at the front axle ($x = L$). You should express this only in terms related to the vehicle (the steer angle, the velocity and the wheelbase) and not the radius of the path. In our notation, we want an expression for the vector

$$\underset{\sim}{v}_{ox,V}$$

at any point from 0 to L along the centerline.

Write the expression for $\underset{\sim}{v}_{ox,V}$. Please include your work.

Question 2.B – Sideslip Angle (Gradescope)

Another way to think about the lateral velocity is in terms of the sideslip angle, β, of the vehicle which is the angle between the velocity vector and the vehicle's centerline:

$$\beta = \arctan\left(\frac{U_y}{U_x}\right)$$

Using the kinematic bicycle model and assuming small angles, calculate an expression for the sideslip angle at any point on the vehicle centerline from $x = 0$ to $x = L$.

Write the expression for β. Please include your work.

Question 2.C – Model Intuition (Gradescope)

Based on this velocity scenario and the geometry of the model, if the vehicle is tracing out a constant radius with point v on the centerline of the rear axle, is the front axle tracing the same path? If not, what is the shape of the path traced by the front axle and does it lie inside or outside the path traced by the rear axle?

Select the correct answers below and give your explanation for these answers.

The front axle will track a path which is _____.

(Circular / Ellipsoidal / Same As The Rear Axle / Other)

The front axle will trace a path which is _____the path traced by the rear axle.

(Inside / The Same As / Outside)

Problem 3 – Tracking a Path

Let's start off on a straight road and assume that the vehicle will be close enough to the desired path that small angle assumptions are valid. Assume that we begin pointing along the centerline of the road ($\Delta\Psi(0) = 0$) but with an offset from our desired lateral position, ($e(0) \neq 0$).

Question 3.A – Lookahead Controller (Gradescope)

A common method for path tracking with vehicles is to use a lookahead controller. This can either look at the projected error some distance ahead on the path or, more commonly, approximate this error as a linear combination of the lateral error and heading error:

$$\delta = K_{la} e_{la}$$
$$e_{la} = e + d_{la}\Delta\psi$$

Here d_{la} functions as the "lookahead distance" ahead of the reference point at which the error is calculated. This is exactly the projected error on a straight road and approximately the projected error on a curved road if the curvature is not too large relative to the lookahead distance.

During lecture, we also derived these equations assuming a small heading error and a small curvature:

$$\dot{s} = V$$
$$\dot{e} = V\Delta\psi$$
$$\dot{\Delta\psi} = \frac{V\delta}{L} - \kappa V$$

Perform several Laplace transforms assuming small angles, a straight road, and **constant velocity**. Arrange this into a form where we can look at the response to the lateral error from the initial condition. In other words, calculate the relationship:

$$\frac{E(s)}{e(0)}$$

The denominator of this relationship is the characteristic equation of the system that we can use to design the performance of our tracking controller.

Note: *Where does the initial condition $e(0)$ come from? Remember that initial conditions are a part of the Laplace transformation, we just often set them to zero when forming a transfer function. Leave them in your expression here.*

Write the resulting relationship $\frac{E(s)}{e(0)}$. Please include your work.

Question 3.B – Controller Stability (Gradescope)

If the lookahead distance (d_{la}) is a positive number, what can you immediately say about the sign of the lookahead gain (K_{la}) in order for the tracking system to be stable?

Select the correct answer below.

Solution:

The lookahead gain (K_{la}) must be _____.

(Positive / Negative)

Question 3.C – Lookahead vs PD Control (Gradescope)

Show that for a constant vehicle speed, the lookahead controller is equivalent to a PD controller of the form:

$$\delta = K_p e + K_d \frac{de}{dt}$$

Calculate the parameters K_p and K_d in terms of K_{la} and d_{la}. You should see that the lookahead control law provides a simple way of scaling, or gain scheduling, proportional and derivative gains as the vehicle speed changes.

Write the parameters K_p ***and*** K_d ***in terms of*** K_{la} ***and*** d_{la}***.***

Question 3.D – Controller Design (Gradescope)

Calculate values of K_{la} and d_{la} that will give the system a rise time of 2 seconds and a peak overshoot of 10% if the vehicle has a wheelbase, L, of $2.5m$ and travels at a speed, V, of $10m/s$.

Write the calculated values of K_{la} ***and*** d_{la}***. Please include your work.***

Problem 4 – Building a Simulator

Now we want to see how well this controller developed using simple analysis works in simulation. You will build up a simulation environment in MATLAB by following the MATLAB Grader prompts available in Canvas. The prompts are also shown for reference in the questions below.

Question 4.A – Equations of motion (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a function which calculates expressions for the vehicle state derivatives $\dot{s}$, $\dot{e}$, and $\Delta\dot{\Psi}$. Do **not** assume small angles when writing expressions for the state derivative. You are to modify the state derivatives `s_dot`, `e_dot`, and `dpsi_dot` in the function template on MATLAB Grader.

Question 4.B – Euler Integration (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a function which implements a simple Euler integration scheme with a given time step. For this function, don't use built in MATLAB functions such as `trapz`. You are to modify the expression for the state at the next time step `x1`.

Question 4.C – Lookahead Controller (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a function which implements the lookahead controller you designed in **Problem 3**. Use the lookahead gain and distance calculated in **Question 3.D**. You are to modify the lookahead controller parameters `K_la` and `d_la`, the lookahead error `e_la`, and the steering angle command `delta`.

In this function you should assume small angles, and that the road is straight.

Problem 5 – Simulate Various Road Geometries

Now we'll use the simulator built in **Problem 4** to analyze the controller's performance. Follow the MATLAB Grader prompts available in Canvas for each question below. The prompts are copied here for reference. Once the simulations are working, answer the qualitative questions about each simulation case.

Question 5.A – Simulation with a Straight Road (MATLAB Grader)

Now we'll use the simulator built in **Problem 4** to analyze the controller's performance. We'll begin on a straight road with no intial heading error ($\Delta\Psi$) and a small path-tracking error (e). The simulation will use parameters from our research vehicle **Niki** which is a Volkswagen GTI. The simulation is set to run for 10 seconds. After the simulation is complete, plot the lateral error (e), the heading error ($\Delta\Psi$), and lookahead error (e_{la}) as a function of time. Follow the prompts in MATLAB Grader to create and run this simulation.

Question 5.B – Simulation with a Straight Road (Gradescope)

Examine the plots you created in **Question 5.A**. Is the vehicle able to track the path with zero error?

Describe the controller's performance in terms of tracking error.

Question 5.C – Simulation with a Curved Road (MATLAB Grader)

Now we are going to let the road curve. Simulate your controller here for the same gains and initial conditions as **Question 5.A** on a curved road with a constant radius of 20 meters. After the simulation is complete, plot the lateral error (e), the heading error ($\Delta\Psi$), and lookahead error (e_{la}) as a function of time. Follow the prompts in MATLAB Grader to create and run this simulation.

Question 5.D – Simulation with a Curved Road (Gradescope)

Examine the plots you created in **Question 5.C**. Is the vehicle able to track the path with zero error?

Describe the controller's performance in terms of tracking error compared to Question 5.A.

Question 5.E – Simulation with an Undulating Road (MATLAB Grader)

Now we'll run the simulation on an undulating road which curves left and right. Simulate your controller here for the same gains and initial conditions as **Question 5.A**. After the simulation is complete, plot the lateral error (e), the heading error ($\Delta\Psi$), and lookahead error (e_{la}) as a function of time. Follow the prompts in MATLAB Grader to create and run this simulation.

Question 5.F – Simulation with an Undulating Road (Gradescope)

Examine the plots you created **Question 5.E**. Is the vehicle able to track the path with zero error?

Describe the controller's performance in terms of tracking performance compared to Question 5.A and Question 5.C.

Question 5.G – Compensating for Non-Zero Curvature (MATLAB Grader)

One way to handle tracking in the event of non-zero (and possibly changing) curvature is to use nonlinear feedback to compensate. The yaw dynamics take the form of:

$$\frac{d\Delta\Psi}{dt} = \left(\frac{V}{L}\right)\delta - \frac{\kappa V}{1 - e\kappa}$$

Compensate for the curvature term in this equation by changing the commanded steer angle to:

$$\delta = K_{la} e_{la} + \frac{L\kappa}{1 - e\kappa},$$

Create a function called `lookahead_lateral_control_curvature` that implements this modified lookahead controller. Follow the MATLAB Grader prompts to create this function.

Question 5.H – Simulation with Curvature Compensation (MATLAB Grader)

Now use your updated version of the lookahead controller with curvature compensation to control the vehicle on the undulating road. Simulate your controller here for the same gains and initial conditions as **Question 5.A**. After the simulation is complete, plot the lateral error (e), the heading error ($\Delta\Psi$), and lookahead error (e_{la}) as a function of time. Follow the prompts in MATLAB Grader to create and run this simulation.

Question 5.I – Simulation with Curvature Compensation (Gradescope)

Examine the plots you created in **Question 5.H**. Is the vehicle able to track the path with zero error?

Describe the controller's performance in terms of tracking error compared to Question 5.E.

Question 5.J – Understanding Curvature Compensation (Gradescope)

Explain how the additional term in our control law, from **Question 5.G**, is compensating for curvature.

Please provide both a mathematical and qualitative explanation of our curvature compensation.

Appendix A – Vehicle Parameters

Variable Name	**Value**	**Units**	**Description**
`veh.m`	1926.2	kg	Mass (Includes 4 passengers)
`veh.Iz`	2763.49	$\mathrm{kg\,m^2}$	Yaw Moment of Inertia
`veh.a`	1.264	m	Distance from Center of Mass to Front Axle
`veh.b`	1.367	m	Distance from Center of Mass to Rear Axle
`veh.L`	2.631	m	Wheelbase

Table 1.1: Vehicle Parameters and Values