

## Assignment 2: Tire Modeling and Experiments

Due Thursday, April 15 at 5:00pm

©Chris Gerdes, 2017

## Purpose

In this assignment, we will look at the bicycle model experimentally and compare the response when using linear and nonlinear tire models. All four problems involve simulation on MATLAB Grader so code your simulations and ask us questions early in the week if you have problems.

## 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. See the Piazza post on homework guidelines for more instructions on the different homework resources available to you. 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 – The Bicycle Model with Linear Tires

Over the course of the quarter, we will develop increasingly detailed models of vehicle motion using a straightforward Eulerian integration scheme you will code in MATLAB. Last week, you implemented a simulator in MATLAB Grader to look at the kinematic model and path following. This week, we will use the same technique to implement a simple form of the bicycle model developed in class (though not yet trying to follow a path with it). For the rest of **Problem 1**, assume the following holds:

- The vehicle moves at constant longitudinal speed ( $\dot{U}_x = 0$ )
- Small angle approximations are appropriate
- The tire forces are linearly related to slip angles through the cornering stiffness

When writing functions and running simulations, use the set of parameters given to you for Niki. These will be available in each MATLAB Grader problem where they are appropriate, and they are given in Appendix A and B at the end of this document.

Problem 1 contains most of the coding problems in the assignment. It may seem very long, but you will be able to re-use most of these functions for the rest of the assignment.

### Question 1.A – Understeer Gradient (MATLAB Grader)

To give our simulation results later in the problem a little more context, let's find Niki's understeer gradient.

Follow the prompts in MATLAB Grader to create a script to calculate Niki's understeer gradient. Find the understeer gradient in units of rad/m/s<sup>2</sup>, then find the understeer gradient in units of deg/g. Is Niki understeering, oversteering, or neutral steering?

Use the tire parameters `f_tire.Ca_lin` and `r_tire.Ca_lin` for the front and rear cornering stiffnesses, respectively.

You are to modify the understeer gradient outputs `K_radpmps2` and `K_degpg` in the function template in MATLAB Grader. Additionally, uncomment your choice of understeering, oversteering, or neutral steering.

### Question 1.B – Slip Angles (MATLAB Grader)

The next few questions will help you build up a simulator using the dynamic bicycle model.

Follow the prompts in MATLAB Grader to create a function to calculate the slip angles,  $\alpha_f$  and  $\alpha_r$ , at the front and rear tires, respectively. Assume small angles here. You are to modify the slip angle outputs `alpha_f` and `alpha_r` in the function template in MATLAB Grader.

### Question 1.C – Linear Tire Model (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a function to calculate the lateral force at one tire,  $F_y$ , using a linear tire model. Use the parameter `tire.Ca_lin` for the cornering stiffness. You are to modify the force output `Fy` in the function template in MATLAB Grader.

### Question 1.D – Dynamic Equations of Motion (MATLAB Grader)

The assumptions we made at the beginning of the problem result in a system with two state variables - the yaw rate,  $r$ , and the lateral velocity,  $U_y$ . You can then take the equations of motion:

$$\begin{aligned} ma_y &= F_{yf} + F_{yr} \\ I_z \dot{r} &= aF_{yf} - bF_{yr} \end{aligned}$$

and write them in the form

$$\begin{aligned} \dot{U}_y &= \dots \\ \dot{r} &= \dots \end{aligned}$$

to have two state equations.

Follow the prompts in MATLAB Grader to create a function to calculate expressions for the vehicle state derivatives  $\dot{U}_y$  and  $\dot{r}$ . You are to modify the state derivatives `Uy_dot` and `r_dot` in the function template in MATLAB Grader.

### Question 1.E – Taking a Single Time Step in Simulation (MATLAB Grader)

To effectively simulate a vehicle's dynamics, you need to do the following things in order at each time step:

1. Calculate slip angles from vehicle states and the steer angle
2. Calculate tire forces from slip angles
3. Calculate vehicle state derivatives from equations of motion using current states and tire forces
4. Update the vehicle states (and steer angle if necessary)

Follow the prompts in MATLAB Grader to create a function to take one time step in your simulation, updating the vehicle states  $U_y$  and  $r$  (we're only looking at velocity states in this assignment). Calculate the lateral acceleration  $a_y$  in this function as well. We'll use it later in our analysis. We have included tested versions of the functions you have written so far and the `integrate_euler` function you wrote last week for you to use here.

To make this function useful in later parts of the assignment, we want to be able to change tire models simply. To do this we're including an option flag `tire_mode` as an input to the function. This will check for either `'linear'` or `'fiala'` and use the respective model. We are only using the linear tire model in this assignment, so you only need to fill in the section for `'linear'` for now.

You are to modify the vehicle states and lateral acceleration `Uy_1`, `r_1`, and `a_y` in the function template in MATLAB Grader.

### Question 1.F – Simulating a Step Steer at Various Speeds (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a script to simulate the response of your model with linear tires to a step steer input of 5 deg (in other words, the initial conditions of the vehicle are travelling straight ahead and the steer angle is set at 5 deg throughout the simulation) at speeds of 10m/s, 20m/s, and 30m/s.

We have split up the data into cell arrays, with cell 1 corresponding to the 10m/s run, etc. If you're not familiar with MATLAB cells, you access them using bracket notation like `ux_mps_{1}`. Each cell holds a vector of data where each element represents one time step. To access a specific time step during a specific experiment you can use `ux_mps_{1}(20)` which would give you the longitudinal velocity at time step 20 during the 10m/s simulation. The MATLAB Documentation is very good if you are still unsure about how to use cell arrays.

Plot the simulated yaw rate,  $r$ , vs. time for all of the speeds on the same plot over the period of 0 to 2 seconds. Create similar plots for the simulated lateral velocity,  $U_y$ , and the simulated lateral acceleration,  $a_y$ .

**Question 1.G – Trends in Yaw Rate (Gradescope)**

Let's look at yaw rate first. Examine the plots you created in **Question 1.F**. As the vehicle speed increases, does the system take more or less time to reach steady state? Does the percent overshoot increase or decrease as speed increases?

*Describe the trends in settling time and percent overshoot of yaw rate with respect to speed.*

**Description:**

**Question 1.H – Trends in Lateral Velocity and Acceleration (Gradescope)**

Now let's look at lateral velocity and acceleration. Examine the plots you created in **Question 1.F**. As the vehicle speed increases, does the steady-state value of the magnitude of lateral velocity increase or decrease? Take a look at the steady-state lateral acceleration at each speed. Do all of these seem feasible for Niki?

*Describe the trend in steady-state lateral velocity with regard to speed and discuss how realistic the simulation results are.*

**Description:**

**Question 1.I – Direction of Lateral Velocity (Gradescope)**

Again, examine the plots you created in **Question 1.F**. What is the sign (positive/negative) of your lateral velocity at low speed, and at high speed? Does this match your intuition? Why, or why not?

*Describe the trend in the sign of lateral velocity, and why this trend occurs.*

**Description:**

## Problem 2 – Tire Nonlinearities

Now let's take a look at the linear tire force approximation as compared to the nonlinear model we derived in class.

### Question 2.A – Nonlinear Tire Model (MATLAB Grader)

Follow the prompts on MATLAB Grader to create a function to calculate the lateral force for one tire,  $F_y$ , using the **Fiala tire model** discussed in class. Using this model, the force at a given tire is a function of the slip angle,  $\alpha$ , and four parameters: the cornering stiffness  $C_y$ , the normal load  $W$ , the peak coefficient of friction  $\mu$ , and the sliding coefficient of friction  $\mu_s$ . Use the parameters `tire.Cy`, `tire.mu`, and `tire.mu_s` for the cornering stiffness, peak friction, and sliding friction, respectively. You are to modify the force output `F_y` in the function template on MATLAB Grader.

### Question 2.B – Plotting Tire Force Curves (MATLAB Grader)

Let's take a moment here to compare the differences in the linear and nonlinear tire models we have implemented so far.

Follow the prompts in MATLAB Grader to create a script which plots the tire force curves for both the linear and fiala tire models. Plot the nonlinear tire curves (lateral force  $F_y$  vs. slip angle  $\alpha$ ) for the front and rear tires on the same plot over a slip angle range of 0 to 12 deg. On the same plot, plot the tire curves for the linear models of the front and rear tires for a slip angle range of 0 to 4 deg.

### Question 2.C – Simulation with Different Tire Models (MATLAB Grader)

Now let's compare how different tire models behave in simulation. Follow the prompts in MATLAB Grader to create a script simulating a step steer of 5 deg with the vehicle travelling at a constant longitudinal speed of 30m/s. Use the `simulate_step` function from Problem 1. We have included a tested version that already has a tested function for the Fiala tire model as well. This way you can swap between tire models using the `tire_mode` option flag input.

Plot the simulated yaw rate,  $r$ , vs. time for both tire models on the same plot over the period of 0 to 2 seconds. Create similar plots for the simulated lateral velocity,  $U_y$ , and the simulated lateral acceleration,  $a_y$ .



**Question 2.D – Comparing Lateral Acceleration (Gradescope)**

Examine the plots you created in **Question 2.C**. How does the peak lateral acceleration with nonlinear tires compare with that of the linear tires? Does this seem more reasonable for Niki?

*Compare the peak lateral acceleration reached with each tire model. Explain if the nonlinear tire model yields a lateral acceleration value that is reasonable for Niki.*

**Description:**

## Problem 3 – Evaluating our Models Against Experimental Data

Before plunging deeper into the development of vehicle dynamics models, we should see how well these simple models predict the measurements we can obtain from a vehicle. We have some data of Niki doing a double lane change maneuver in the lot off Searsville road on Stanford's campus at various speeds. A high-precision GPS/INS system in the car gives us accurate measurements of lateral velocity, yaw rate, lateral acceleration, and vehicle speed. We also measure steer angle through the actuators built into the car. In this problem, we're going to compare these measurements to what we predict using the models developed in class. While there are a lot of simplifications involved, you should find the results at least somewhat impressive. If not, chances are you have a bug somewhere...

Three experiments were run, one at low speed, one at intermediate speed, and one at high speed. We have separated all of the data into MATLAB cell arrays, with the first cell corresponding to the low speed test, etc.

The following MATLAB Grader questions will walk you through modifying your simulations so that they can use recorded steer angle and longitudinal velocity as inputs. With this modification made, you can now compare the experimental data to three different models.

*NOTE: Though we have written our simulator to execute at 1kHz, our data logger records data on Niki at 200Hz. In research and industry, we frequently need to resample data that has been logged. One way to address this here is to post-process our recorded data early in our script by creating two new vectors of steering angle and velocity that correspond in time to our simulation time vector. You can use the MATLAB function `interp1` along with the vector of simulation time to interpolate a vector of steering angles and velocity from the recorded data. We have implemented this for you in MATLAB Grader, but it's a great trick to know if you work with experimental data in the future.*

### Question 3.A – Linear Bicycle Model, Constant Speed Assumption (MATLAB Grader)

Follow the prompts in MATLAB Grader to create a script which simulates the double lane change maneuver captured in the provided dataset. You should use the linear bicycle model (linear tire model), and assume that speed is constant throughout the maneuver.

Three experiments were run, one at low speed, one at intermediate speed, and one at high speed. We have separated all of the data into MATLAB cell arrays, with the first cell corresponding to the low speed test, etc. We have trimmed the dataset to contain only the maneuver in each cell (trimmed out lining the car back up, etc.) Find the average longitudinal speed of the vehicle over the entire maneuver. Set the longitudinal speed in your simulation to this average value.

Plot the yaw rate (in deg/s), lateral velocity (in m/s), and lateral acceleration (in  $\text{m/s}^2$ ) from each experimental run together with the corresponding simulated values to see how well your model predicts the vehicle's behavior.

### Question 3.B – Linear Bicycle Model, Variable Speed (MATLAB Grader)

It is possible to allow the speed to vary in your simulation. We are going to assume that we still have two state variables (yaw rate and lateral velocity) and simply update the longitudinal velocity with measured data at every time step.

*NOTE: Here we're fixing the mismatch in timing with the recorded longitudinal velocity in exactly the same way we handled steer angle data before.*

Follow the prompts in MATLAB Grader to create a script which simulates the double lane change maneuver captured in the provided dataset. You should still use the linear bicycle model (linear tire model).

Three experiments were run, one at low speed, one at intermediate speed, and one at high speed. We have separated all of the data into MATLAB cell arrays, with the first cell corresponding to the low speed test, etc. We have trimmed the dataset to contain only the maneuver in each cell (trimmed out lining the car back up, etc.) Use the measured longitudinal speed in your equations of motion in this simulation instead of the constant speed you assumed previously.

Plot the yaw rate (in deg/s), lateral velocity (in m/s), and lateral acceleration (in  $\text{m/s}^2$ ) from each experimental run together with the corresponding simulated values to see how well your model predicts the vehicle's behavior.

### Question 3.C – Nonlinear Bicycle Model, Variable Speed (MATLAB Grader)

Finally, let's look at the difference we might get with a nonlinear tire model. Using the same variable speed scheme from **Question 3.B**, set the option flag in `simulate_step` to use the Fiala tire model instead of the linear model.

Follow the prompts in MATLAB Grader to create a script which simulates the double lane change maneuver captured in the provided dataset. You should use the nonlinear bicycle model (Fiala tire model).

Three experiments were run, one at low speed, one at intermediate speed, and one at high speed. We have separated all of the data into MATLAB cell arrays, with the first cell corresponding to the low speed test, etc. We have trimmed the dataset to contain only the maneuver in each cell (trimmed out lining the car back up, etc.) Use the measured longitudinal speed in your equations of motion.

Plot the yaw rate (in deg/s), lateral velocity (in m/s), and lateral acceleration (in  $\text{m/s}^2$ ) from each experimental run together with the corresponding simulated values to see how well your model predicts the vehicle's behavior.

**Question 3.D – Analysis of Different Models (Gradescope)**

Look back over the plots you generated in **Question 3.A**, **Question 3.B**, and **Question 3.C**. Under what conditions do your simpler models stop accurately predicting the vehicle's behavior? Why is this happening? What other driving maneuvers would more prominently show the shortcomings of our linear tire model?

*Analyze where simple models stop working well and explain why this happens.*

**Description:**

## Appendix A – Vehicle Parameters

Variable Name	Value	Units	Description
<code>veh.m</code>	1926.2	kg	Mass (Includes 4 passengers)
<code>veh.Iz</code>	2763.49	kg m <sup>2</sup>	Yaw Moment of Inertia
<code>veh.a</code>	1.264	m	Distance from Center of Mass to Front Axle
<code>veh.b</code>	1.367	m	Distance from Center of Mass to Rear Axle
<code>veh.L</code>	2.631	m	Wheelbase
<code>veh.Wf</code>	9817.9	N	Static front axle weight
<code>veh.Wr</code>	9078.1	N	Static rear axle weight

Table 2.1: Vehicle Parameters and Values

## Appendix B – Tire Parameters

### Linear Tire Model

Variable Name	Value	Units	Description
f_tire.ca_lin	80,000	N/rad	Front Cornering Stiffness
r_tire.ca_lin	120,000	N/rad	Rear Cornering Stiffness

Table 2.2: Linear Tire Model Parameters

### Fiala Tire Model

Variable Name	Value	Units	Description
f_tire.cy	110,000	N/rad	Front Cornering Stiffness
f_tire.mu	0.90	Unitless	Front Peak Coefficient of Friction
f_tire.mu_s	0.90	Unitless	Front Sliding Coefficient of Friction
r_tire.cy	180,000	N/rad	Rear Cornering Stiffness
r_tire.mu	0.94	Unitless	Rear Peak Coefficient of Friction
r_tire.mu_s	0.94	Unitless	Rear Sliding Coefficient of Friction

Table 2.3: Fiala Tire Model Parameters