Final Group (4 member) Project
Control Law and Presentation Due December 2nd 8:00 AM
Presentations Will Be Scheduled for 12/2 and 12/3

The purpose of the final project is to develop an autonomous steering control system for the MKZ that you modeled in the mid-term project. Control laws from the top teams will be tested on the autonomous MKZ shown below. Students must provide their control law by Wednesday December 1st for evaluation. A new script, `run_MKZ_fp.p` has been developed for testing your controllers in simulation. See the help file for details. The new script will allow you to select one of three predefined routes used for testing the controller. You will also test your control capability with constant and sinusoidal desired headings as described below. Follow the steps below to design your controllers and prepare a presentation describing your process and controller performance (include your code as text, not images). Including the required information defined in each step in your presentation.

Figure 1: GAVLab Autonomous MKZ

a) Develop/verify the model from the yaw response of the vehicle assuming a constant speed of 30 m/s. In the mid-term project you developed a model with input steer angle (δ) and output yaw rate (r). There are two changes to the model for the final project. First, your original yaw rate model should be updated to provide yaw, ψ (the integral of yaw rate, $\psi = \int r$) as the output of the plant. You will also need to model actuator dynamics in the full model of the system. The steering actuation (with input command steer angle (δ_{com}) and output actual steer angle (δ)) has a first order transient response with a time constant of 0.1 seconds and a DC gain of one. Your complete dynamic model should have input “command steer angle” and “output yaw” (i.e. heading).

   Requirement: Dynamic model with input command steer angle and output yaw – provide derivation in presentation. Provide plots of eigenvalues, step response, and Bode plot.

   Goal: Validate the model that you developed with system identification in the mid-term project with the full dynamic model derived using the attached FBD and parameter values.

b) Develop a feedback control system to determine the commanded steering angle to control the heading (yaw) of the vehicle while driving at 30 m/s. You may choose any control law, but the controller should track a constant desired heading (yaw) with zero steady state error. It should track low frequency sinusoids with small amplitude errors and phase shifts.

   Requirement: Derivation of a heading controller. Provide closed loop eigenvalues, expected step response and Bode plot.

   Goal: Design controller to have a settle time of less than 0.6 seconds and a maximum percent overshoot less than 5%. Show validation of the expected performance with using your model of the system.

c) Evaluate the step response of your control system on the "actual" MKZ using the `run_MKZ_fp.p` function. Verify that the controller can achieve a response similar to the expected response found in part b). Also, evaluate the performance of the control for lower speeds (e.g. 10 m/s).

Requirement: Provide plots of the step response of the control system on the actual vehicle at 30 m/s. Discuss any differences between the expected and actual response. Provide plots of the step response of the at 10 m/s. Discuss how the low-speed response compares to the high-speed response. Does this make sense?

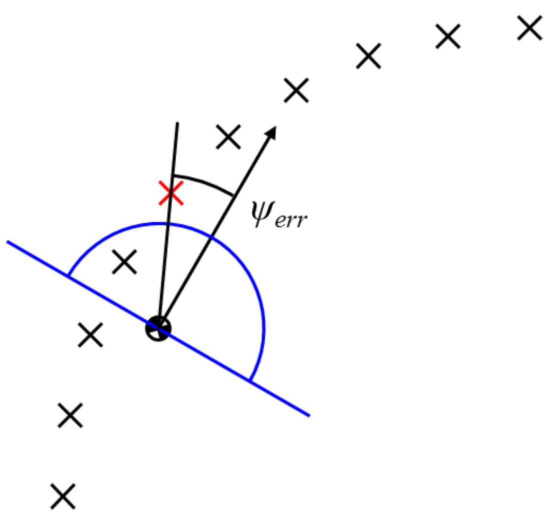
Goal: Show that the control can achieve a settle time of less than 0.6 seconds and a maximum overshoot less than 5% on the actual vehicle (i.e. using `run_MKZ_fp.p`*). Determine if the settle time and overshoot increase or decrease for higher or lower speeds.*

d) Evaluate the frequency response of your control system on the actual MKZ. Does the actual performance match the expected performance from part b? At what frequency does the vehicle only achieve 50% of the desired heading amplitude?

Requirement: Provide plots of the response of the control system for sinusoidal inputs with different frequencies. Show desired and actual heading on the same plot for each frequency chosen. Discuss the similarities and/or differences between the actual and expected performance.

Goal: Describe what changes you would make to your controller to improve tracking performance at higher frequencies? How would these changes affect the step response.

e) Use the provided trajectories to evaluate the performance your control in real-world scenarios. You can choose a trajectory by entering 1 or 2 as the last input to `run_MKZ_fp.p` (see help for example code). The function will return a "waypoint" for you to steer the vehicle towards to track a path. The figure below shows a series of waypoints that the vehicle will attempt to drive to follow a path. The red X is the waypoint that the vehicle should drive towards now (this is the point returned by `run_MKZ_fp.p`). The heading error (ψ_{err}) is the difference between the current heading and the desired heading towards the red X. The heading error can be calculated as shown in the equation below.



$$\psi_{err} = \tan^{-1}\left(\frac{E_{des} - E}{N_{des} - N}\right) - \psi$$

Note: for the arctan, use `atan2`. Also, this heading error, ψ_{err}, needs to be wrapped (i.e., limited to) $\pm 90°$ $\left(\pi/_2\, radians\right)$ using the `wrap_angle.m` function provided. Test your controller on both trajectories. Be sure to record the vehicle position returned by `run_MKZ_fp.p` to evaluate the effectiveness of the controller. Using the instructions below to plot the vehicle position on Google Earth. Was the vehicle able to follow the desired path.

Requirement: Test the control on all the given trajectories. Provide plots of the vehicle positions on Google Earth. Discuss the performance on the three paths.

Goal 1: How fast can you do the trajectories?

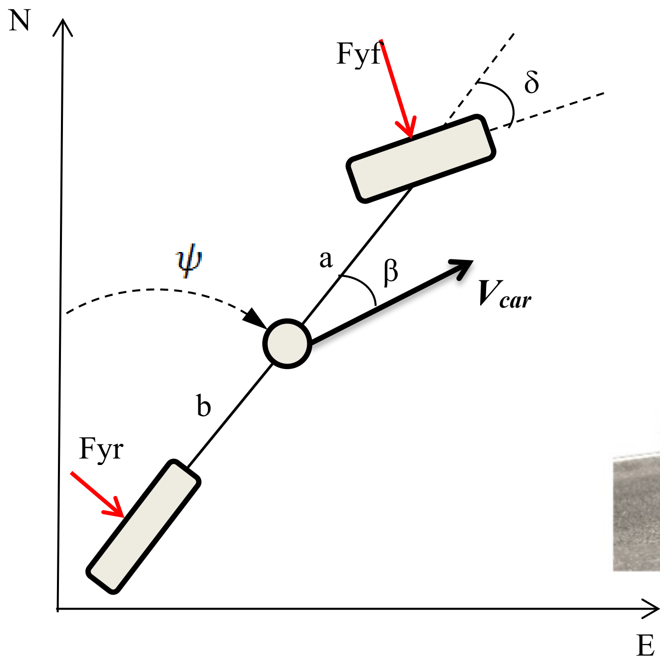
Goal 2: Develop your own waypoint path to test your controller and show the performance.

Plotting position on Google Earth:

1. Instructions to be provided.

Vehicle Dynamics of Lincoln MKZ

Derive the system model for the autopilot system.



The tire force is related to the lateral velocity at the tire through the following approximation:

$$Fy = \frac{-C}{V_{car}} \times V_y^{tire}$$

Parameter	Name	Units	Value
a	Front CG distance	m	1.257
b	Rear CG distance	m	1.593
C_f	Front Tire Cornering Stiffness	N/rad	120,000
C_r	Rear Tire Cornering Stiffness	N/rad	184,600
I_z	Yaw Mass Moment of Inertia	Kgm^2	4292
V_{car}	Vehicle Speed	m/s	User Defined
m	Vehicle Mass	kg	1856

Note that you must include centripetal acceleration when applying Newton's lateral equations:

$$\Sigma F_y = m\ddot{y} = m\left(V_{car}\dot{\psi} + \dot{V}_y\right)$$

Design a control system with the following interface control document (ICD):
Recall that $\vec{V}_B = \vec{V}_A + \vec{r}_{A/B} \times \vec{\omega}$

Measurements: GPS East, North, and Heading (ψ) and gyroscope yaw rate ($r = \dot{\psi}$), steer angle.
Controllable Variables: Steer angle.

Note that lateral position can be defined from North or from any arbitrary line by rotating the car into that frame by the desired heading (ψ_{des}) as shown below. The GPS heading provided will be measured from North in radians.