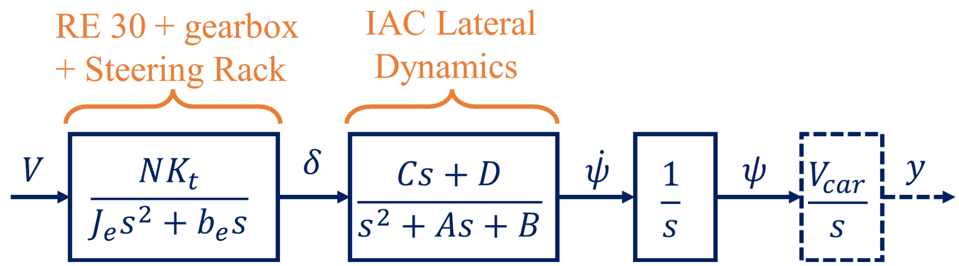
Final Group (4 member) Project
Control Law and Presentation Slides Due 12/3/2025 9:00 AM
Presentations on 12/3-12/5

The purpose of the final project is to develop a steering control system for the autonomous IndyLight race car (Dallara IL-15 used in the Indy Autonomous Challenge) pictured here. The steering wheel is turned with the Maxon RE 30 and gear box from the mid-term project. To complete this project, you must 1) model the system dynamics and 2) develop a model-based controller. Similar to the midterm assignment, you will use a `run_Indy_car_pcode.p` script that will allow you to select the inputs, initial conditions, and predefined routes for performing system identification or testing your controller(s) performance. The dynamics of the system are shown below in a block diagram.

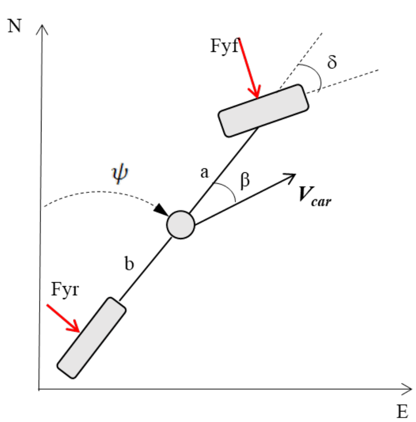


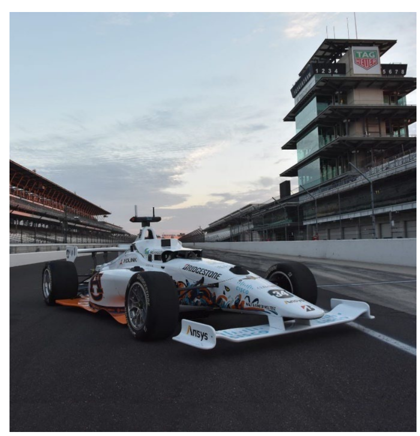
To control the car around a track, you can use heading (ψ) or lateral position (y). Note the block diagram shows an approximation of lateral position: $\dot{y} = V_{car}\sin(\psi + \beta) \approx V_{car}\psi$

To complete this project, we suggest using both Modeling and System Identification techniques.

1) Modeling Techniques:

To begin, you need to sum forces in the lateral direction and sum moments about the center of gravity of the vehicle to find equations of motion for yaw accerelation $(\ddot{\psi})$ and lateral acceleration $(\dot{V}_y)$. Then combine the equations to develop a transfer function with input steer angle (δ) and output yaw (ψ).Use the notes below to develop your model. Derive the system model for the autopilot system using the modeling steps and Newton's laws of motion.





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

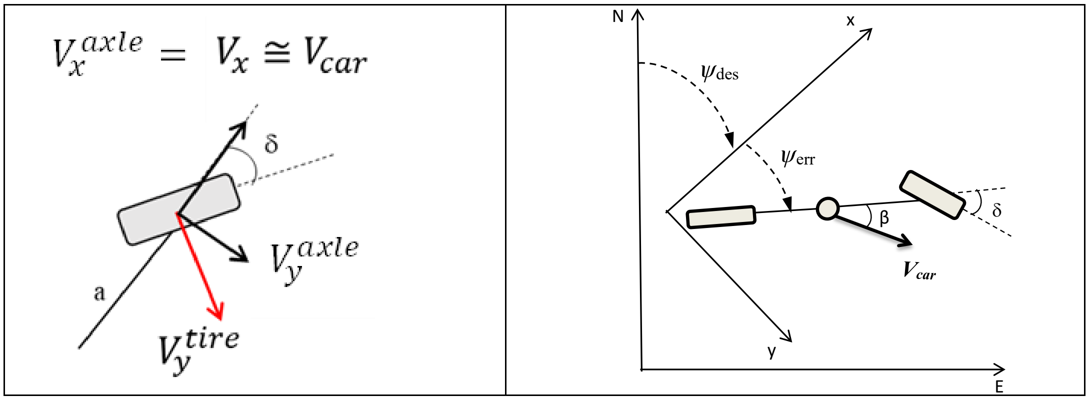
$$Fy = \frac{-C}{V_{car}} \times V_y^{tire}$$

You must include centripetal acceleration when applying Newton's lateral equations:

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

Note that V_y^{tire} is the lateral velocity in the tire frame (as shown in the figure below on the right):

Final Group (4 member) Project
Control Law and Presentation Slides Due 12/3/2025 9:00 AM
Presentations on 12/3-12/5



The velocity at the tire (V_y^{tire}) has components of V_x^{axle} and V_y^{axle}. Note that V_y^{axle}will have components of V_{car} and due to the rotation of the vehicle. This can be found by recalling that $\vec{V}_B = \vec{V}_A + \vec{\omega} \text{ x } \vec{r}_{A/B}$.

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.

The table below contains the approximate values needed to model the yaw dynamics of the car.

Parameter	Name	Units	Value
a	Front CG distance	M	1.72
b	Rear CG distance	M	1.25
C_f	Front Tire Cornering Stiffness	N/rad	100,000
C_r	Rear Tire Cornering Stiffness	N/rad	100,000
I_z	Yaw Mass Moment of Inertia	Kgm^2	1200
V_{car}	Vehicle Speed	m/s	User Defined
m	Vehicle Mass	Kg	720
η	Steering Rack Ratio		15

Note that the steering rack ratio (η) is the overall steering ratio of 15:1 for the car (i.e., the gearbox output at the steering wheel turns 15 degrees for 1 degree the tires turn).

2) **System Identification Techniques:**

a. Use system identification techniques to develop a model of the steering rack dynamics. It may be helpful to have known dynamics of the motor+gearbox, potentially closed loop control on motor position, to determine steering rack dynamics.

b. To perform system identification on the lateral dynamics, we suggest that you implement a feedback control system on steer angle. This will allow you to better isolate the lateral dynamics from the steer angle (RE30+gearbox+steering rack) dynamics.

c. Provide useful/appropriate inputs to the vehicle (`run_Indy_car_pcode.p`) in order to identify key parameters. As you can see from the dynamics above, the lateral dynamics are a function of speed, so you will need to perform system identification at the main speed (8 m/s) as well as any higher speeds you may wish to drive at.

At the end of the process, always validate your model against `run_Indy_car_pcode.p` for a few key speeds (or any speeds you plan to run your car).

Final Group (4 member) Project
Control Law and Presentation Slides Due 12/3/2025 9:00 AM
Presentations on 12/3-12/5

Follow the steps below to design your controllers and prepare a presentation describing your process and performance, including the information defined in each step. Note that meeting the requirements achieves 50% of the possible points, with performance on the goal steps earning the remaining 50% of the points.

a) Develop a model of the IAC car from input voltage (V) to yaw rate $(\dot{\psi})$ using modeling (lateral dynamics) and system identification (steering rack) techniques. Develop your own MATLAB simulation of this system. Validate that your model matches `run_Indy_car_pcode.p` using constant and variable inputs. Note: you have outputs of steer angle, yaw, and yaw rate. Having these measurements should help you identify if your errors are in your steering model or lateral dynamics model. Also, you may wish to use the homogeneous (free) response (i.e. $V = 0$) to better isolate the lateral vehicle response from the steering response.
Requirement: Dynamic model with input voltage and output yaw rate– provide the derivation or system identification approach in presentation. Provide plots of eigenvalues, step response, Bode plot, and verification of the Bode plot with multiple sinusoidal inputs.
Goal: Validate (match) your model against the `run_Indy_car_pcode.p` *function.*

b) Develop a feedback control system to calculate the desired voltage to control the heading (yaw) or lateral position (depending on your control system design) of the vehicle. A simple speed to aid in the development is at 8 m/s. Note that the full feedback system includes both the lateral dynamic model and the steering model from the mid-term project. You may choose any control law, but the controller should track a constant desired reference (heading or lateral position) with zero steady state error. It should track low frequency sinusoids (e.g. $\psi_{des} = \sin(1t)$) with small amplitude errors and phase shifts.
*Requirement: Derivation of a heading controller **for a speed of 15 m/s**. Provide closed loop eigenvalues, expected step response, and Bode plot. Show validation of the expected performance with using your simulation model of the system. Include plots of inputs – unreasonable inputs will count against you. **Note: you may need to redesign your controller as you increase speed**!*
Goal: Design controller to have a settle time of less than 0.6 seconds and a maximum percent overshoot less than 5%.

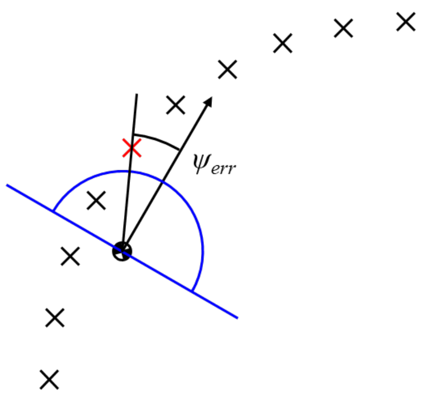
c) Evaluate the step response (constant desired reference) of your control system on the "actual" IAC car using the `run_Indy_car_pcode.p` function. Verify that the controller can achieve a response similar to the expected response found in part c. Also, evaluate the performance of the control for higher speeds (e.g. >60 m/s). Note the IL-15 has posted speeds over 200 mph at IMS ([Indy Lights tops 200 mph at Indy]), along with competitions at Las Vegas: ([https://youtu.be/1W99UskyQqs?t=3846]) and Laguna Seca: ([https://www.youtube.com/watch?v=JW12EsBhkco])
Requirement: Provide plots of the step response (and all of the control system inputs) on the actual vehicle at 8 m/s. Discuss any differences between the expected and actual response. Provide plots of the step response of the at >60 m/s. Discuss how the low-speed response compares to the high-speed response. Does this make sense? Include plots of inputs – unreasonable inputs will count against you.
Goal: Show that the controller can achieve a settle time of less than 0.6 seconds and overshoot less than 5% on the actual vehicle (i.e. using `run_Indy_car_pcode.p`*). Determine if the settle time and overshoot increase or decrease for higher or lower speeds.*

d) Use the provided trajectories to evaluate the performance of your control in real-world scenarios. You can choose a trajectory by entering 1, 2, or 3 as the last input to `run_Indy_car_pcode.p` (see example code in the help file).
*Requirement: Complete a lap at Indianapolis Motor Speedway (WP_File 2) **at 15 m/s.** Provide plots of the vehicle positions on Google Earth. Discuss the performance.*
Goal: Your grade will be determined by your fastest stable lap (determined by your velocity).
Next Steps: Test your controller on other tracks and develop your own waypoint path to test your controller and show the performance. For example, drive a route around campus.

Final Group (4 member) Project
Control Law and Presentation Slides Due 12/3/2025 9:00 AM
Presentations on 12/3-12/5

Techniques for waypoint and lateral error control:

The *`run_Indy_car_pcode.p`* 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_Indy_car_pcode.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 $\pm$ 180 degrees using *`wrap_angle.m`* function provided on canvas or the class website. Test your controller on all three given trajectories.

Instructions for plotting vehicle position on Google Earth.

Using a web browser, go to: https://www.gpsvisualizer.com

Click on: Google Earth KML or go to: https://www.gpsvisualizer.com/map_input?form=googleearth

Upload your `waypoint_file_for_GPSVisualizer.txt` generated by the .p code where it says "File #1 Browse…" You may upload multiple files (but these files should all be for the same "WP_FILE" runs using *`run_Indy_car_pcode.p`* to ensure they are all starting in the same location (i.e. each WP_FILE has a different starting location). Remember that *`run_Indy_car_pcode.p`* over writes the same file name, so you must rename the txt file if you desire to save it as a separate run. This could be done to overlay results from multiple runs using different control gains for example.

Once you have your file(s) loaded, hit "Create KML File" This will bring up a new page with a .kmz file. Click on the KMZ file and hit "save to computer" (you can also hit "open with" if you have Google Earth on your computer.

If you do not have Google Earth downloaded, then go to: https://earth.google.com

Click on the projects icon on the left tab (looks like a pin on a map and is the tab option from the top). Then click on "Open" then "Open->Import KML file from computer" Select the file you just save from GPS Visualizer (or whichever file you want to display) and hit "Open" This will then display your data onto Google Earth.