ECE484 FA23 MP2 Walkthrough
Vehicle Model & Control

Yan Miao, Yeji Mun, John Pohovey, Hongyi Li

9/15

Outline

- Logisticis
  - Highbay Visit (9/19)
    - AB1, AB2 -- 11:00
    - AB3, AB4, AB5 -- 11:40
  - MP1 Demo (9/22)
  - Midterm 1 (10/3)
    - Announcement on details will be made soon
  - MP2 Demo (10/6)
  - Project starts around mid October
    - Timeline will be announced soon
- HW2
- MP2

HW2

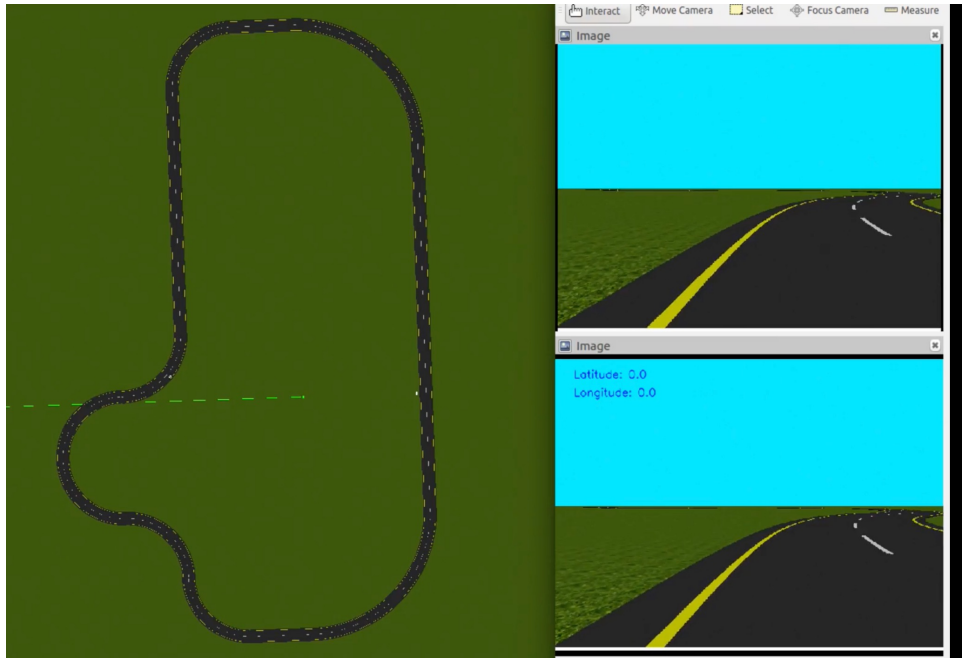
- Equilibrium Point (Problem 1)
- Lyapunov Stability (Problem 2, 3)
  - Hurwitz Matrix
  - Lyapunov Theorem
- Feedback Control (Problem 3)

- Also be prepared for exams on those topics

MP2

- Overall Objective: Implement a controller to navigate your car around the track(a list of waypoints to follow).

- Task 1: Learn ROS
- Task 2: Longitudinal Control (Set Velocity)
- Task 3: Lateral Control (Set Heading)



Task 1: Learn ROS

- Recall *rosnode* and *rostopic*, *subscriber* and *publisher*
- ROS also have its own message type, E.g. *ModelState*
- You need to read the documentation to learn how to extract info(E.g. position) from those ROS message

File: gazebo_msgs/ModelState.msg

Raw Message Definition

```
# Set Gazebo Model pose and twist
string model_name           # model to set state (pose and twist)
geometry_msgs/Pose pose     # desired pose in reference frame
geometry_msgs/Twist twist   # desired twist in reference frame
string reference_frame      # set pose/twist relative to the frame of this entity (Body/Model)
                            # leave empty or "world" or "map" defaults to world-frame
```

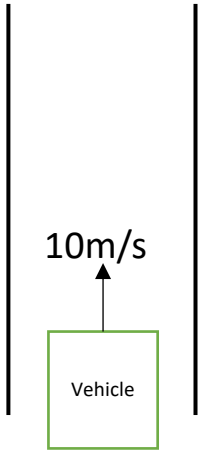
Compact Message Definition

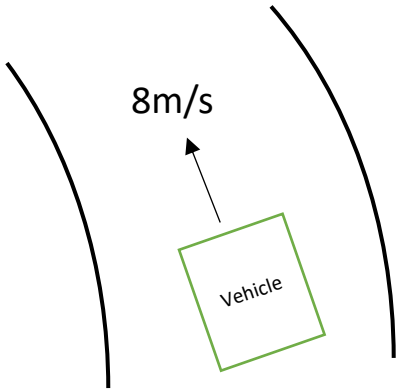
```
string model_name
geometry_msgs/Pose pose
geometry_msgs/Twist twist
string reference_frame
```

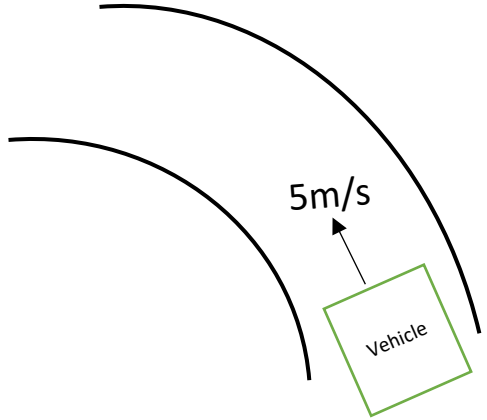
Task 2: Longitudinal Controller

- Based on a list of future waypoints, return the corresponding target speed.
- You need to design the <state, curvature, target_speed>mapping.
  - speed = f(curvature, vehicle_state), f is a function up to you to design, e.g. could be piece-wise linear

- For example, drive fast during straights and slow during turns







Tasks 3: Lateral Controller (Pure Pursuit)

- Derivation of the underlying mathematics is not a requirement for this MP, but strongly recommended for exams and projects.
- In short, you need to implement the below formula to get the steering value

$$\delta = \arctan\left(\frac{2L\sin(\alpha)}{l_d}\right)$$

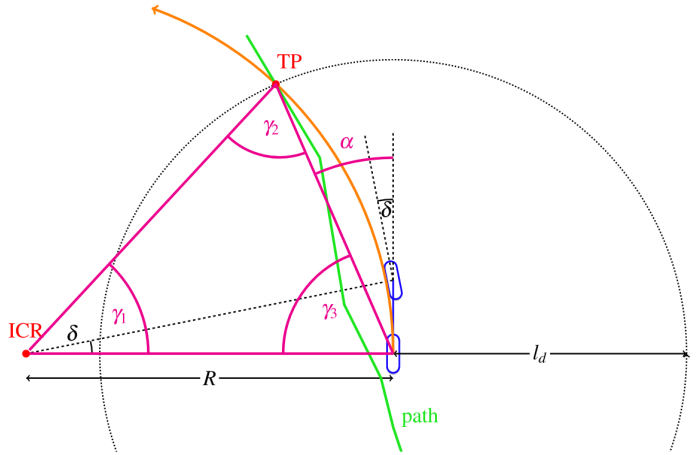


Fig. 28 The magenta triangle helps us to establish a formula for δ. #

- Several Methods to find the lookahead Target Point

(check documentation for details)

  - Directly setting as the nearest waypoint
  - Setting a fixed lookahead distance and interpolate between provided waypoints
  - Setting a dynamic lookahead distance (as a function of vehicle states) and interpolate between provided waypoints

Metric

- **Safety**: measured by number of waypoints you can follow
- **Efficiency**: measured by time to finish(TTF) the track (note difference between simulation time and calendar time)
- **Comfort**: (de)acceleration values

Grading

- HW2 (100 pts)
- MP2 (100 pts)
  - Autograding 20 pts
  - Report 70 pts
  - Demo 10 pts

- All details please refer to the documentation