TERM PROJECT

DESIGN AUTONOMOUS SHUTTLE CIRCULATOR FOR EASTON TOWN CENTER IN SMART COLUMBUS

ECE 5553 – AUTONOMY IN VEHICLES

EVAN STODDART, DEREK LONGSHORE, YIFAN WU, TEAWON HAN, SHIHAO LIU

04/23/2018

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

SHUTTLE VEHICLE AND AUTONOMOUS ARCHITECTURE

DEREK LONGSHORE

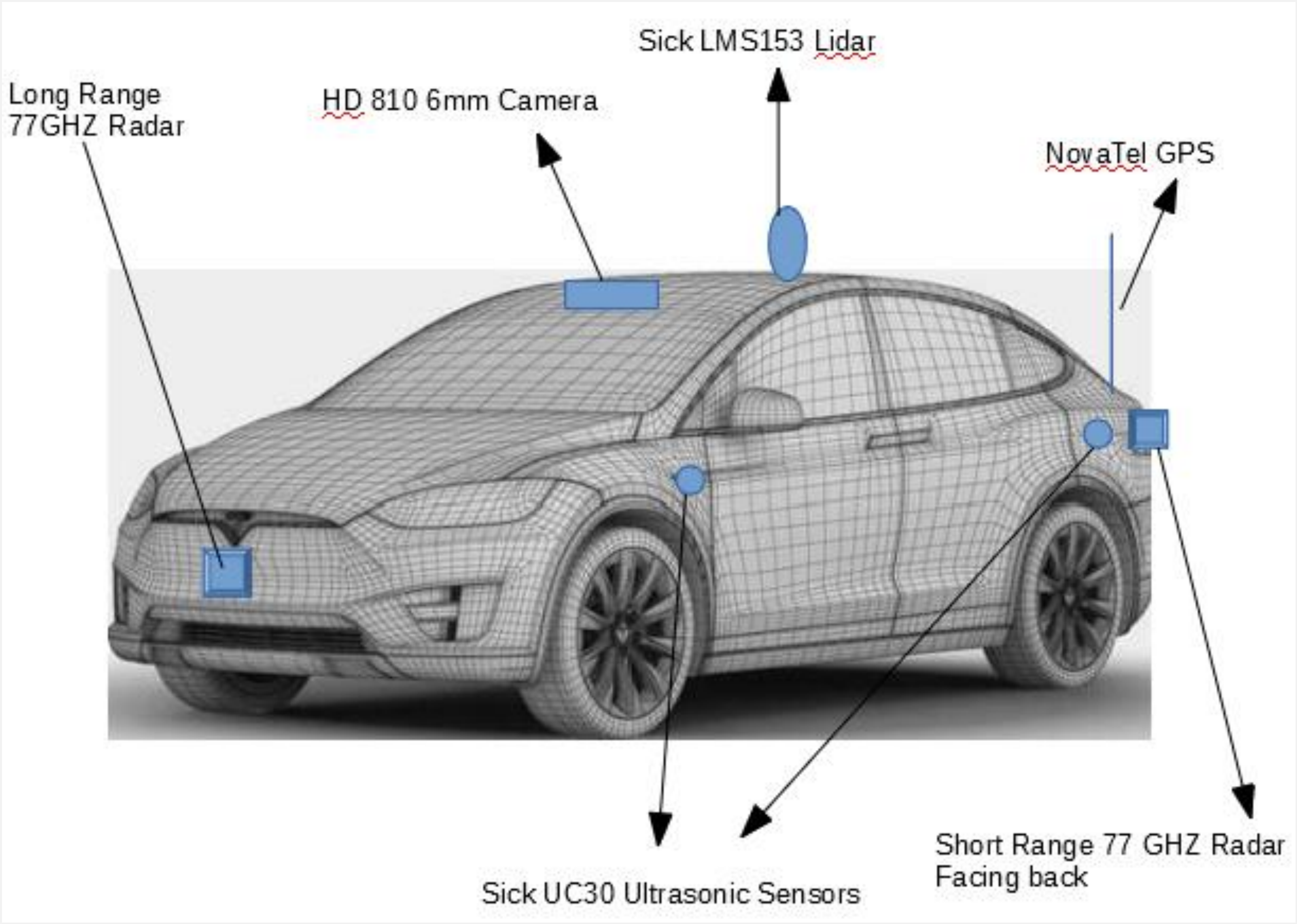
AUTONOMOUS ARCHITECTURE: DIMENSIONS

- Vehicle specifications:
  - Tesla Model X
  - M = 2458.924; (5421 lbs)
  - Weight distribution is 50% front and 50% rear
  - cf=2*1.5e5; (Cornering stiffness)
  - lr=2.96418*0.51; (Rear axles)
  - lf=2.96418*0.49; (Front axles)
  - cr=2*1.5e5; (Cornering stiffness)
  - L =2.96418 ; (Inter axle distance)
  - Car length = 198.3" L x 81.5" W x 66" H (Inches)
- Assumptions:
  - The vehicle is controlled by wire and there is no passengers. Hence, the mass of the vehicle is the total mass with no passengers.
  - The width is with mirrors folded in.
  - The specifications are without the added sensors.
  - The information can be found at https://www.tesla.com/support/model-x-specifications

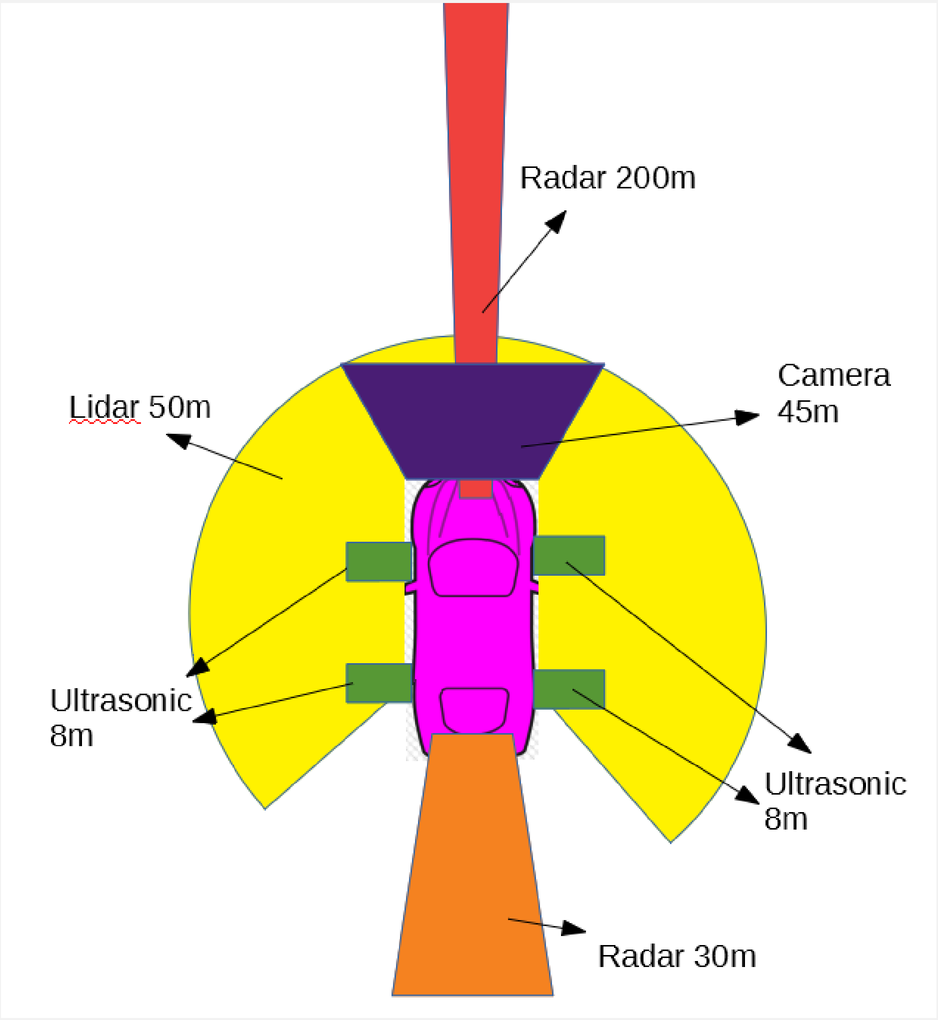
AUTONOMOUS ARCHITECTURE: SENSORS

Sensor	Characteristics
HD 810 6mm Camera	Fov: 45 degrees Range: 45m Sampling Rate: N/a
77 GHZ Radar	Fov: 10 Range: 200m Sampling Rate: 77GHZ
77 GHZ Radar	Fov: 30 Range 30m Sampling Rate: 77GHZ
NovaTel GPS	N/a
Sick LMS153 Lidar	Fov: 270 Range: 50m Sampling Rate: 50HZ
Sick UC30 Ultrasonic Sensor	Fov: 15 Range: 8m Sampling Rate: 20HZ

AUTONOMOUS ARCHITECTURE: SENSOR LAYOUT

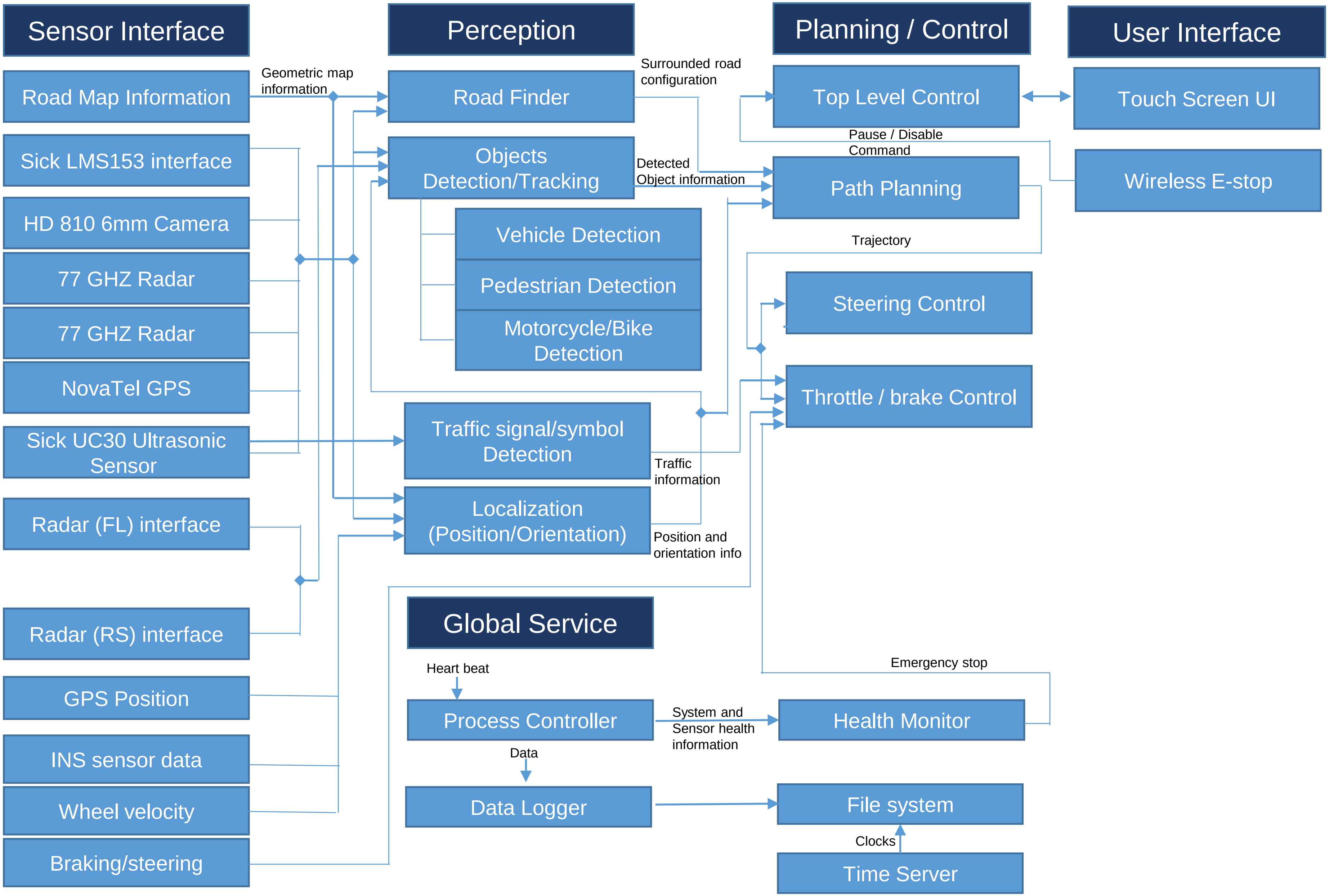


AUTONOMOUS ARCHITECTURE:
MAP OF SENSORS



BLOCK DIAGRAM FOR AUTONOMOUS VEHICLE SYSTEM

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH



THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

LOW LEVEL LONGITUDINAL CONTROL

YIFAN WU

DETAILS USED FOR THE MODEL

- Dugoff Tire Model

- μ : tire friction coefficient (0.7)
- F_z: vertical force = M/4(2458.924 kg/4)*g(9.81N/m^2)
- ε_r: 0
- C_x: longitudinal stiffness (3e5N)
- C_y: cornering stiffness (1.5e5N/rad)

- Wheel Dynamics

- η_t : Transmission Efficiency (0.9)
- λ_t : Gear ratio (1.0)
- λ_f : Final drive ratio (4.1)
- R_w: effective wheel radius (0.3 m)
- I_w : Inertia of the wheel (1 kgm^2)
- M: mass of the vehicle (M =2458.924)
- M_{eq} : equivalent mass factor - Load forces
- ρ : Air density (1.225 kgm^3)
- A_f: Frontal area of vehicle (2.8 m^2)
- C_d: Aerodynamic drag coefficient (0.29)
- C_{rr}: Rolling resistance coefficient (0.015)

- Equations used for modelling
  - $T_{MW}(t) = \eta_t \lambda_t \lambda_f T_M$
  - $I_w \frac{dw}{dt} = T_{MW}(t) - T_b - F_x R_W$
  - $s = \frac{wR_W - V}{\max(V, wR_W)}$, where $w = \int I_w \frac{dw}{dt}$
  - $F_{total}(t) = F_a(t) + F_g(t) + F_r(t)$
    - Aerodynamic drag force
      - $F_a(t) = \frac{1}{2}\rho C_d A_f V_{eff}$, where $V_{eff} = V + V_{wind}$
    - Rolling resistance
      - $F_r(t) = Mg\cos\alpha\, C_r(V)$, where $C_r(V) = C_{rr}V$
    - Road Grade
      - $F_g(t) = Mg\sin\alpha$
  - Vehicle longitudinal dynamics equation
    - $(M + M_{eq})\frac{dV}{dt} \equiv F_x - F_{total}$
  - Dugoff tire model equation

VEHICLE SPECIFICATION AND ASSUMPTION

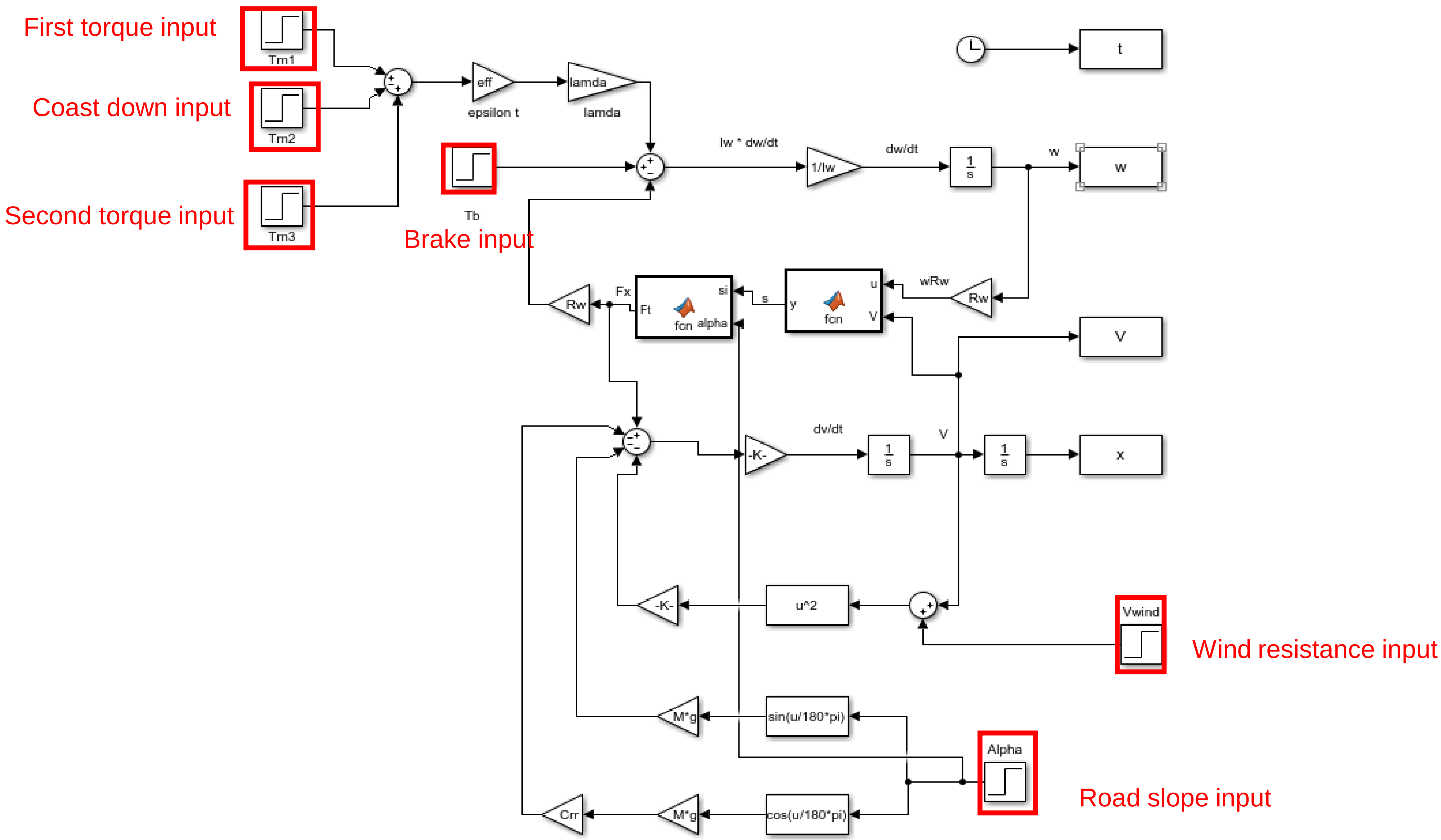
- Vehicle specification (Tesla Model X)
  - M = 2458.924;   %5421 lbs
  - cf=2*1.5e5;   %cornering stiffness
  - lr=2.96418*0.51;   %rear axles
  - lf=2.96418*0.49;   %front axles
  - cr=2*1.5e5;   %cornering stiffness
  - L =2.96418 ;   %Inter axle distance
  - Car length =5.03682   %198.3 inches
- Assumptions
  - The vehicle is controlled by wire and there is no passage. Hence, the mass of the vehicle is the total mass.
  - No extreme weather condition occur.
  - Vehicle is operating in a sunny day so that friction during rainy day does not apply to the test .
  - The speed limit is either 25 mph or 40 mph with multiple stop sign that requires vehicle to fully stop.
  - Vehicle drives on flat or slopped road without any construction or bumper.

DETAILS USED FOR THE MODEL

- Input and Output of the model
  - Input: Tm, Tm2, Tb, Vwind, and Alpha
  - Output: V

- Simulation Parameters
  - ODE Solver used: auto(ode45)
  - Time steps: 0.001
  - Duration of Simulation: 3000sec

SCREEN SHOT OF THE LONGITUDINAL SIMULINK MODEL



SCREEN SHOT OF THE SIMULINK MODEL

- There are two MATLAB Functions in the model
  - Function 1
    - Input : wRw and V
    - Output: Si

```
function y = fcn(u,V)
A = max([V,u]);
y=(u-V)/A;
```

  - Function 2
    - Input. : Si
    - Output: Ft

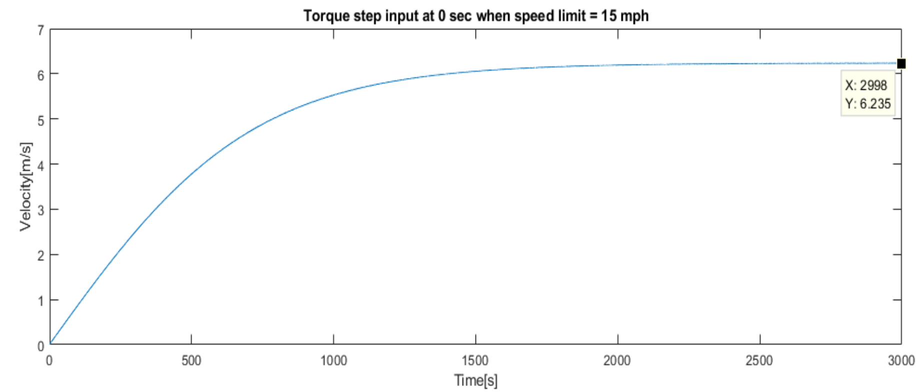
```
function Ft = fcn(si,alpha)
m = 2458.924;
g = 9.81 ;
mu = 0.7;     %Dry road friction coeff
Fz = m*g/4;
C_s = 3e5;
C_alpha = 1.5e5;
    S= mu*Fz*(1-si)./(2*sqrt(C_s^2*si.^2 + C_alpha^2*tand(alpha)^2));
;%since er=0
    if S <1
            f = S.*(2-S);
    else
            f = 1 ;
    end
    Ft = C_s*si.*f./(1-si);
```

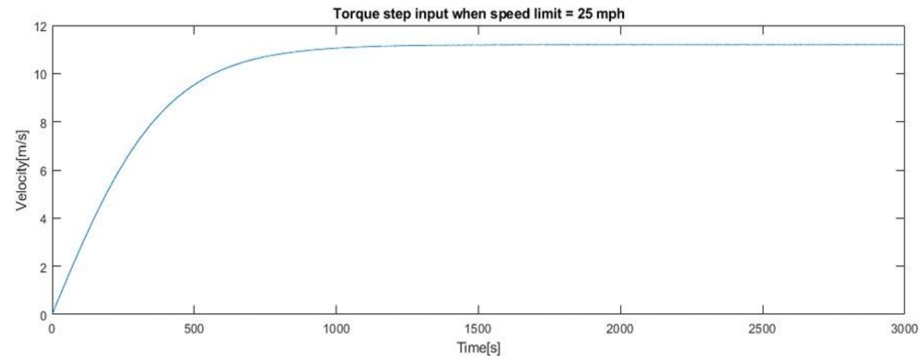
SIMULATION RESULTS

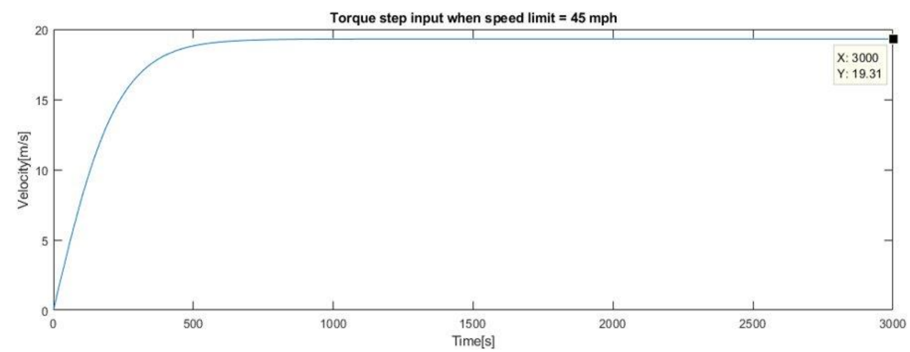
CASE: SPEED LIMIT 15MPH, 25MPH, 45MPH

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

- Three major speed limits in the Easton area
  - *15mph*, 25mph and 40 mph.
- By setting the first torque (step input) to 25.5 Nm, the velocity is converged to 6.235 m/s (13.94 mph)
- By setting the first torque (step input) to 29 Nm, the velocity is settled down to 11.2 m/s (25.05369 mph)
- By setting the first torque (step input) to 39 Nm, the velocity is settled down to 19.31 m/s(43.195 mph)
- It takes a longer time to reach the speed when having a low torque.

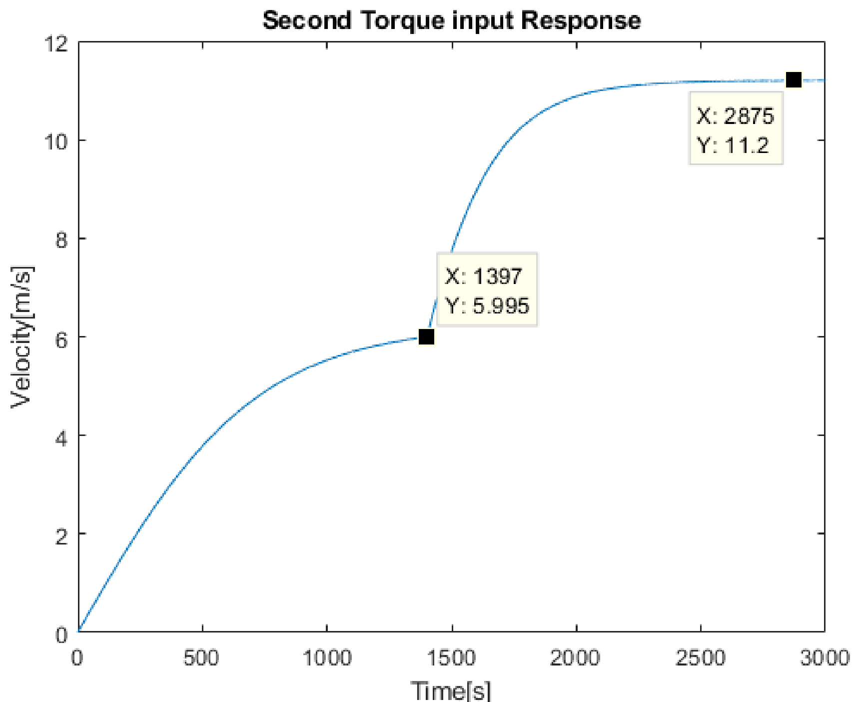






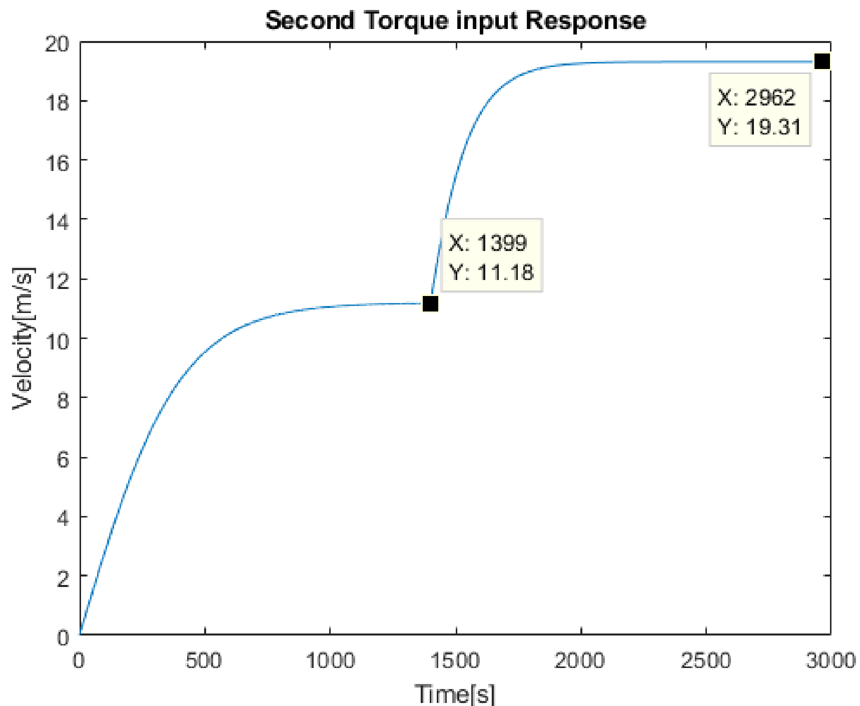
CASE1: THE SPEED LIMIT IS CHANGED FROM 15MPH TO 25 MPH

- When the speed limit is changed, the longitudinal speed will be increased by increasing torque (Tm).
  - V is proportional to Tm.
- Figure below shows that vehicle first increase velocity with 15 mph speed limit. Then, it accelerates again to meet 25 mph speed limit.
- By setting the first torque (step input) to 25.5 Nm at 0 sec, the velocity is settled down to 5.995 m/s(13.41 mph). By setting the second torque to 3.5 Nm at 1400 sec, the velocity is settled down to11.2 m/s(25.05 mph)



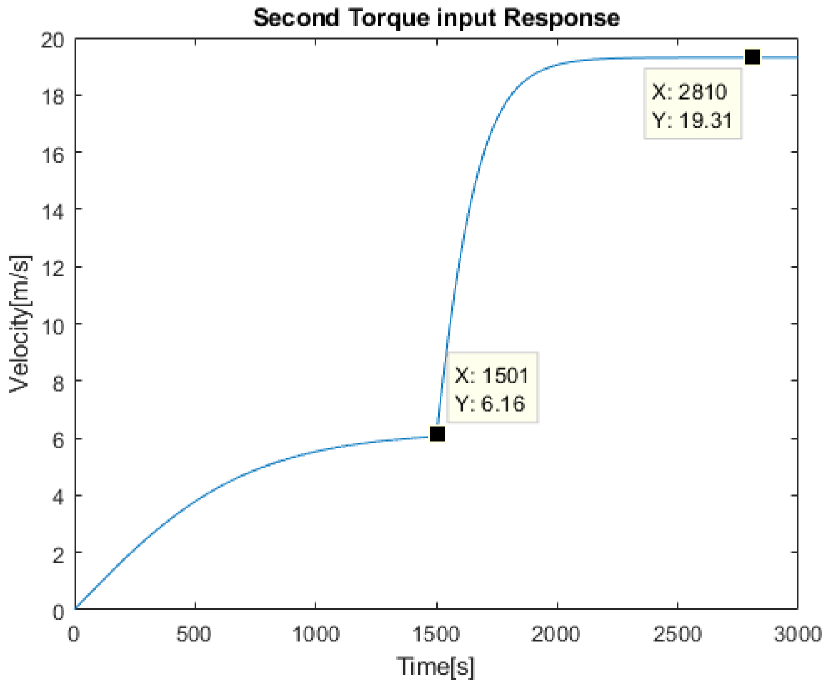
CASE2: THE SPEED LIMIT IS CHANGED FROM 25MPH TO 45 MPH

- When the speed limit is changed, the longitudinal speed will be increased by increasing torque (Tm).
  - V is proportional to Tm.
- Figure below shows that vehicle first increases velocity with 25 mph speed limit. Then, it accelerates again to meet 45 mph speed limit.
- By setting the first torque (step input) to 29 Nm at 0 sec, the velocity is settled down to 11.2 m/s(25.05 mph). By setting the second torque to 10 Nm at 1400sec, the velocity is settled down to19.31 m/s(43.195 mph)



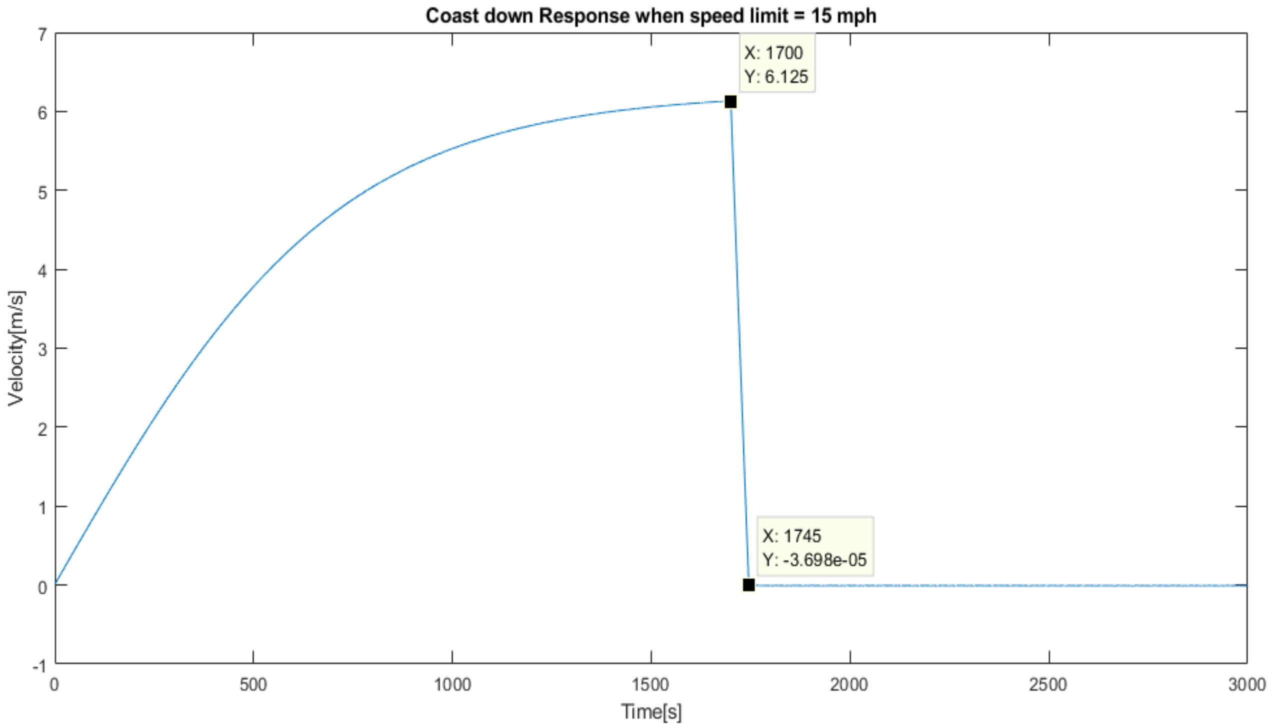
CASE3: THE SPEED LIMIT IS CHANGED FROM 15MPH TO 45 MPH

- When the speed limit is changed, the longitudinal speed will be increased by increasing torque (Tm).
  - V is proportional to Tm.
- Figure below shows that vehicle first increases velocity with 15 mph speed limit. Then, it accelerates again to meet 45 mph speed limit.
- By setting the first torque (step input) to 25.5 Nm at 0 sec, the velocity is settled down to 6.16 m/s(13.78 mph). By setting the second torque to 13.5 Nm at 1500sec, the velocity is settled down to19.31 m/s(43.195 mph)



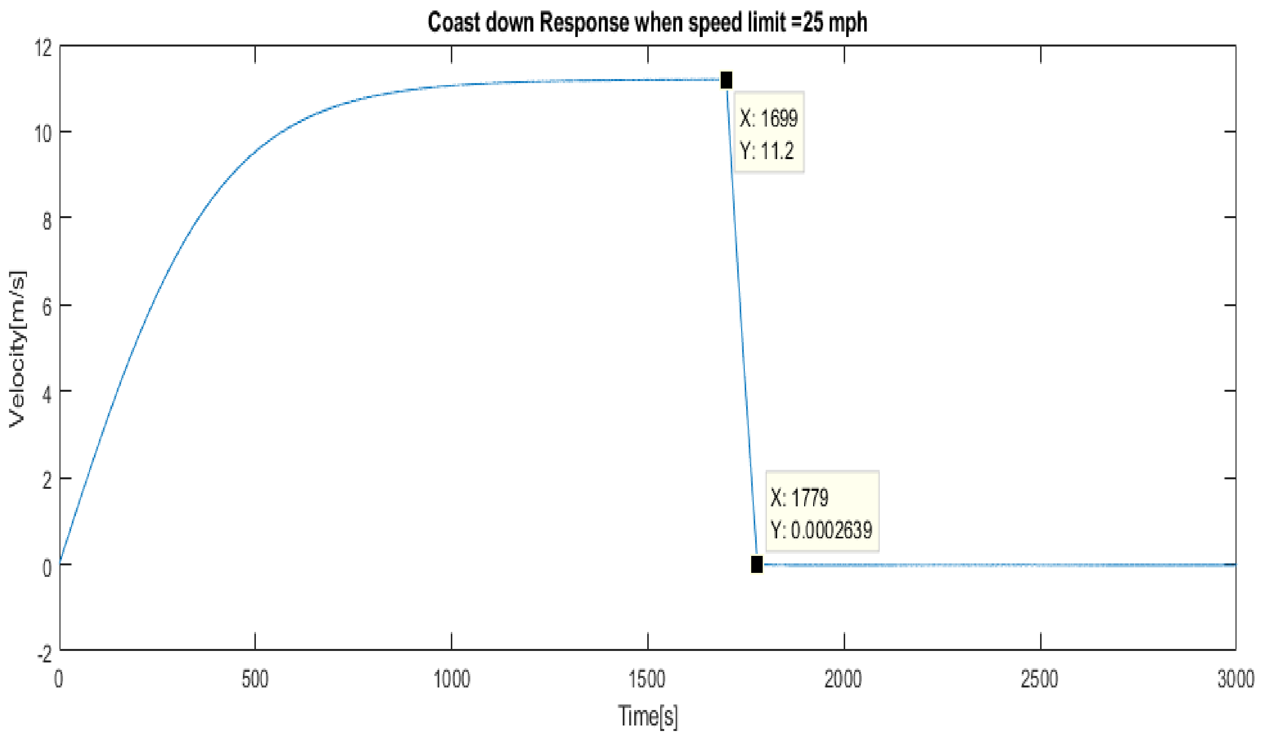
CASE 4: COAST DOWN AT SPEED LIMIT 15 MPH

- Tm1 (first torque input: 25.5Nm) is applied at 0 sec.
- There is no torque applied at 1700 sec.
- Tm2 (Coast down input: 25.5Nm) is applied at 1700 sec
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2}$, it goes to 0 at 1700 sec
- The speed is decreased to zero at 1745 sec (taking 45sec) without braking.

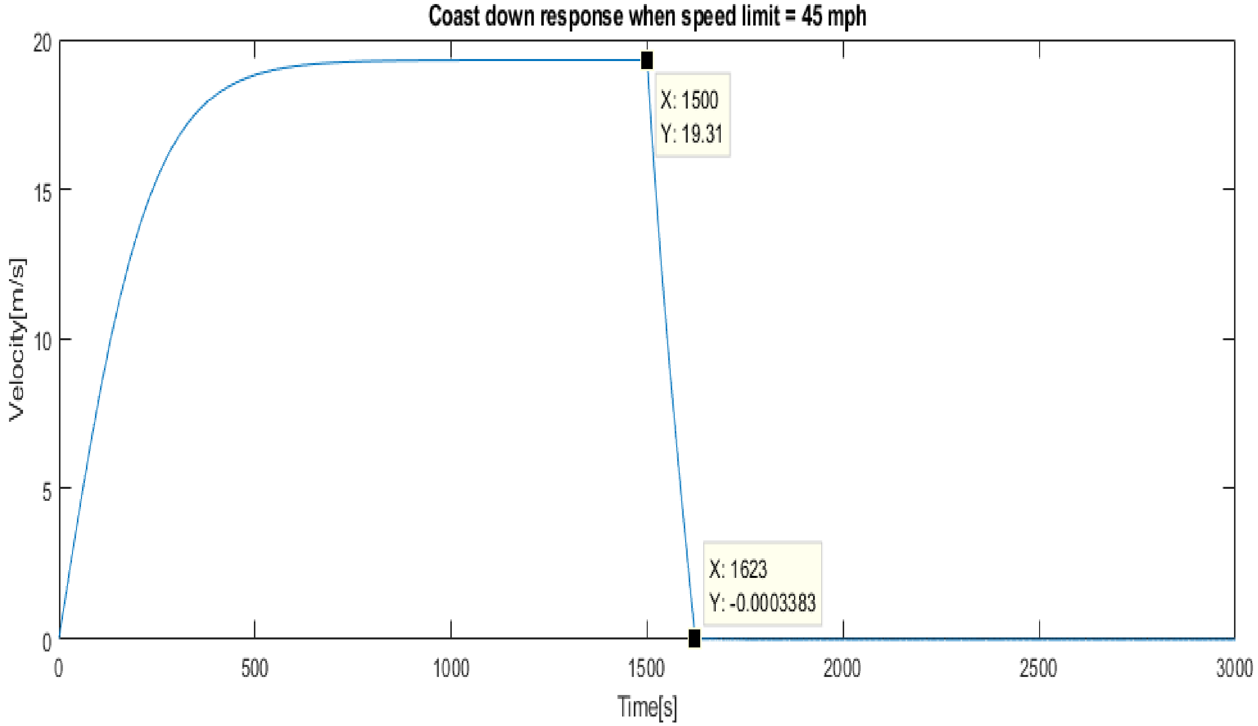


CASE 5: COAST DOWN AT SPEED LIMIT 25 MPH

- Tm1 (first torque input: 29Nm) is applied at 0 sec.
- There is no torque applied at 1700 sec.
  - Tm2 (Coast down input: 29Nm) is applied at 1700 sec
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2}$, it goes to 0 at 1700 sec
- The speed is decreased to zero at 1779 sec (taking 79sec) without braking.

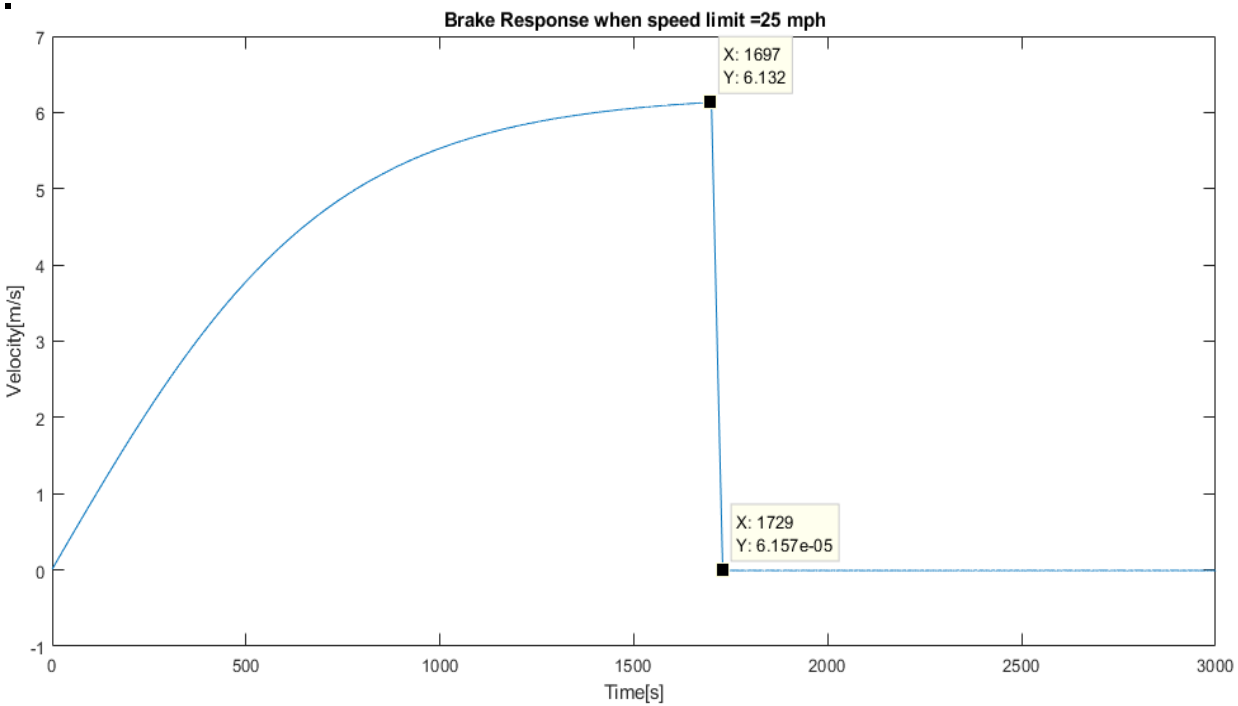


- Tm1 (first torque input: 39Nm) is applied at 0 sec.
- There is no torque applied at 1500sec.
  - Tm2 (Coast down input: 39Nm) is applied at 1500sec.
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2}$, it goes to 0 at 1500sec
- The speed is decreased to zero at 1623 sec (taking 123sec) without braking.

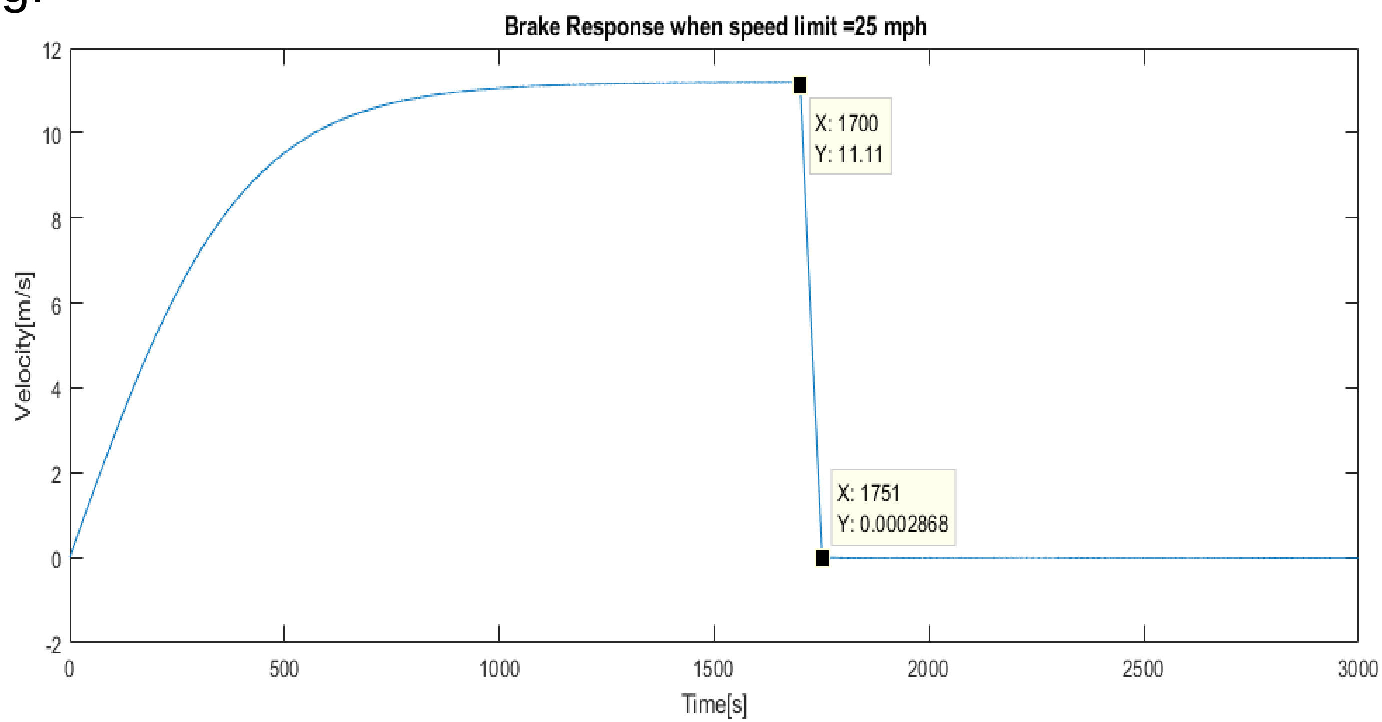


CASE 7: BRAKE AT 15 MPH

- Tm1 (first torque input: 25.5Nm) is applied at 0 sec.
- There is negative torque applied at 1700sec.
  - Tm2 (Coast down input: 25.5Nm) is applied at 1700sec.
  - Tb (brake torque input: -50Nm) is applied at 1700
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2} \ - \mathrm{Tb}$, it goes to -50 at 1700sec
- The speed is decreased to zero at 1729 sec (taking 29sec) with braking.

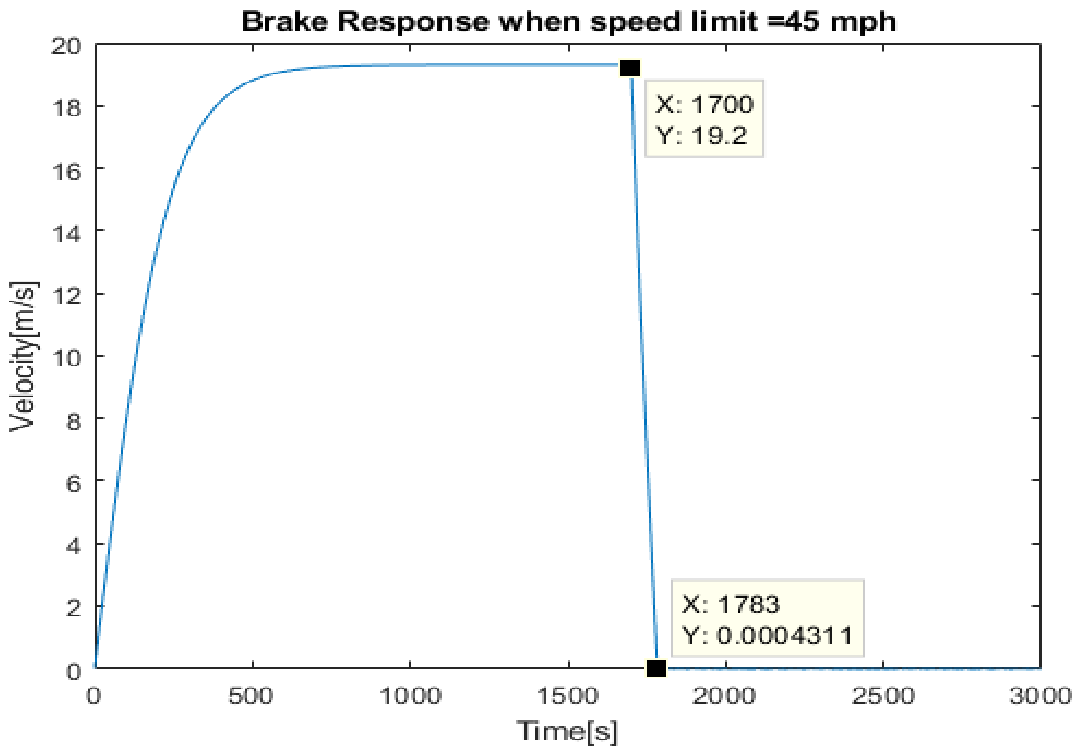


- Tm1 (first torque input: 29Nm) is applied at 0 sec.
- There is negative torque applied at 1700sec.
  - Tm2 (Coast down input: 29Nm) is applied at 1700sec.
  - Tb (brake torque input: -50Nm) is applied at 1700
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2} \ - \mathrm{Tb}$, it goes to -50 at 1700sec
- The speed is decreased to zero at 1751 sec (taking 51sec) with braking.



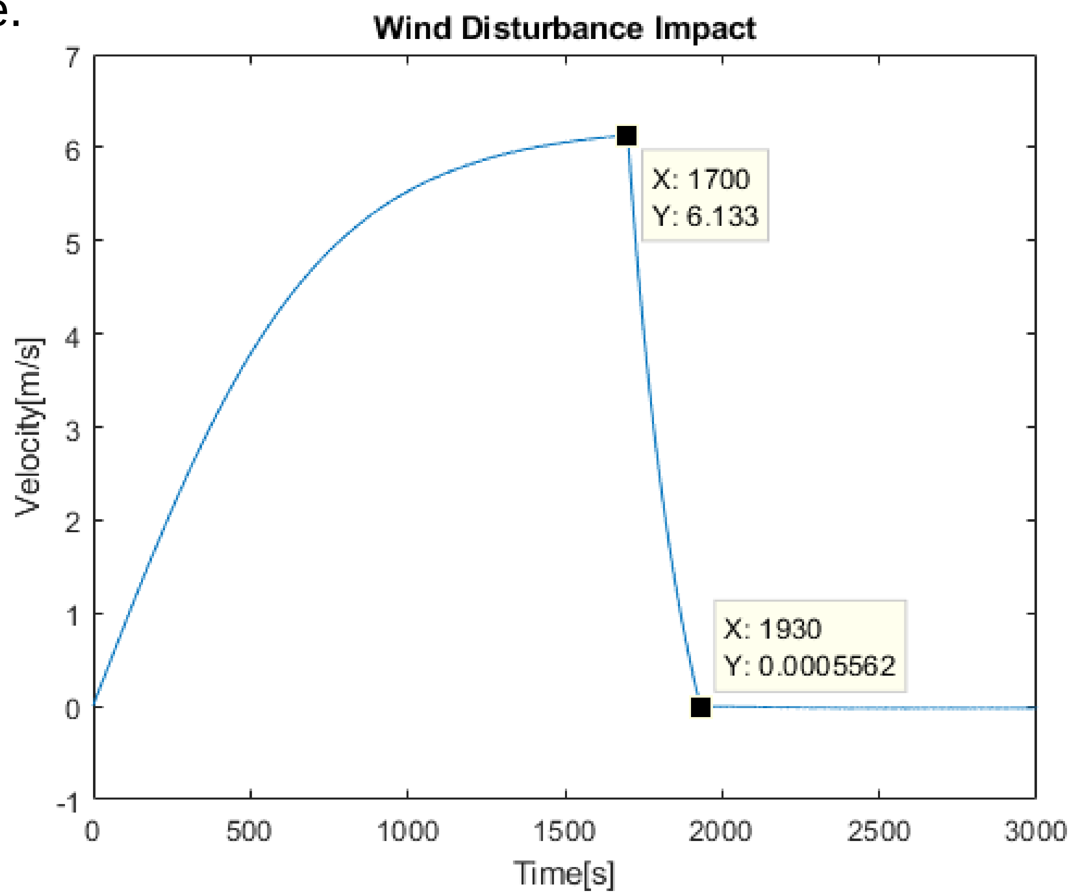
SIMULATION RESULTS
CASE 9: BRAKE AT 45 MPH

- Tm1 (first torque input: 39Nm) is applied at 0 sec.
- There is negative torque applied at 1700sec.
  - Tm2 (Coast down input: 39Nm) is applied at 1700sec.
  - Tb (brake torque input: -50Nm) is applied at 1700
  - $\mathrm{Tm}_{total} = \mathrm{Tm1} - \mathrm{Tm2} - \mathrm{Tb}$, it goes to -50 at 1700sec
- The speed is decreased to zero at 1783 sec (taking 83sec) with braking.



CASE 10 : WIND DISTURBANCE IMPACT AT 15 MPH

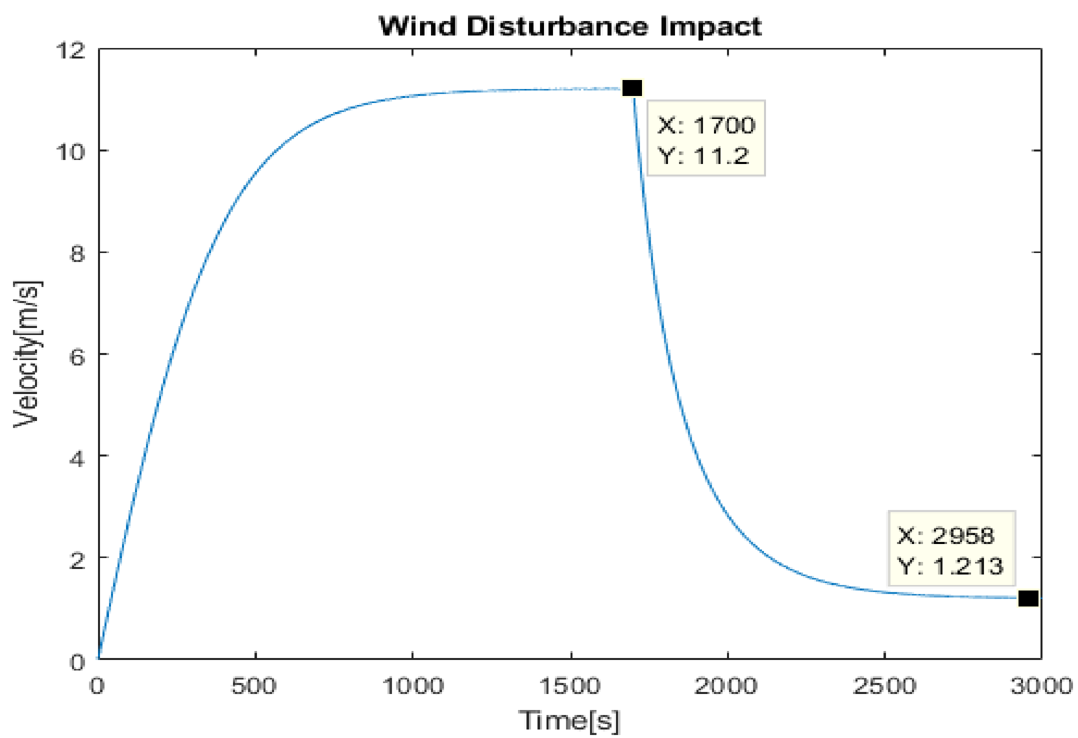
- Tm1 (first torque input: 25.5Nm) is applied at 0 sec.
- There is negative wind torque applied at 1700sec.
  - Vwind (wind disturbance input:10Nm)
  - The speed is decreased to zero at 1930 sec (taking 130sec) with wind disturbance.



SIMULATION RESULTS
CASE 11: WIND DISTURBANCE IMPACT AT 25 MPH

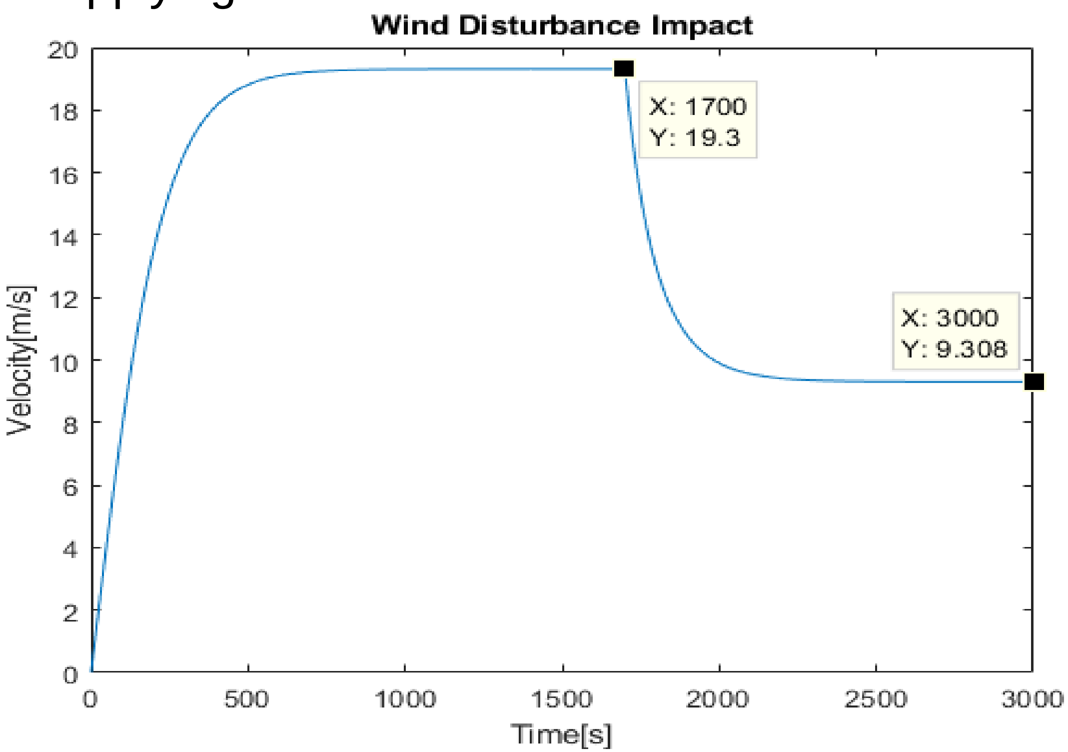
THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

- Tm1 (first torque input: 29Nm) is applied at 0 sec.
- There is negative wind torque applied at 1700sec.
  - Vwind (wind disturbance input = 10Nm)
  - The speed is decreased to approximately 1.2 m/sec at 2958 sec (taking 258sec) with wind disturbance.
  - The wind resistive force is not strong enough to stop the car while positive torque is still applying



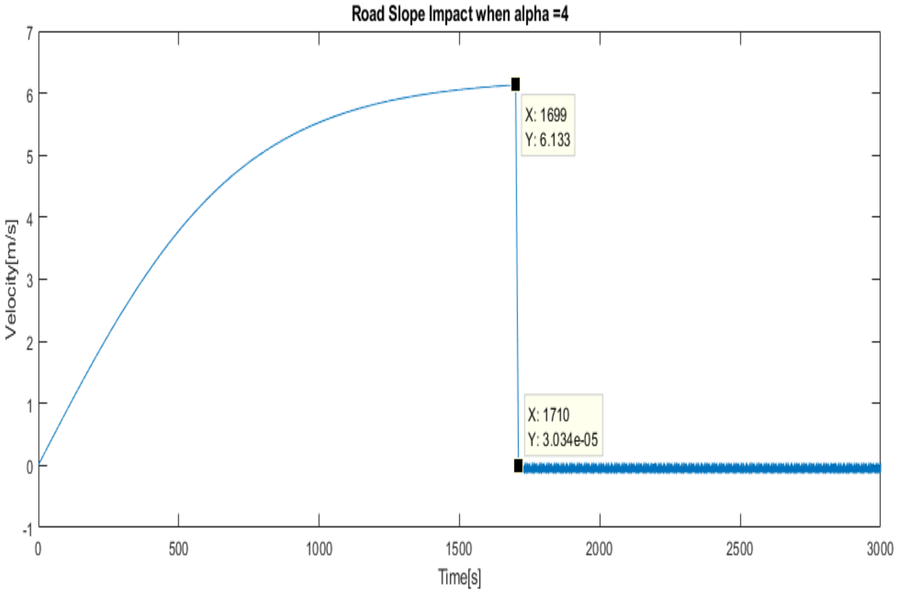
CASE 12: WIND DISTURBANCE IMPACT AT 45 MPH

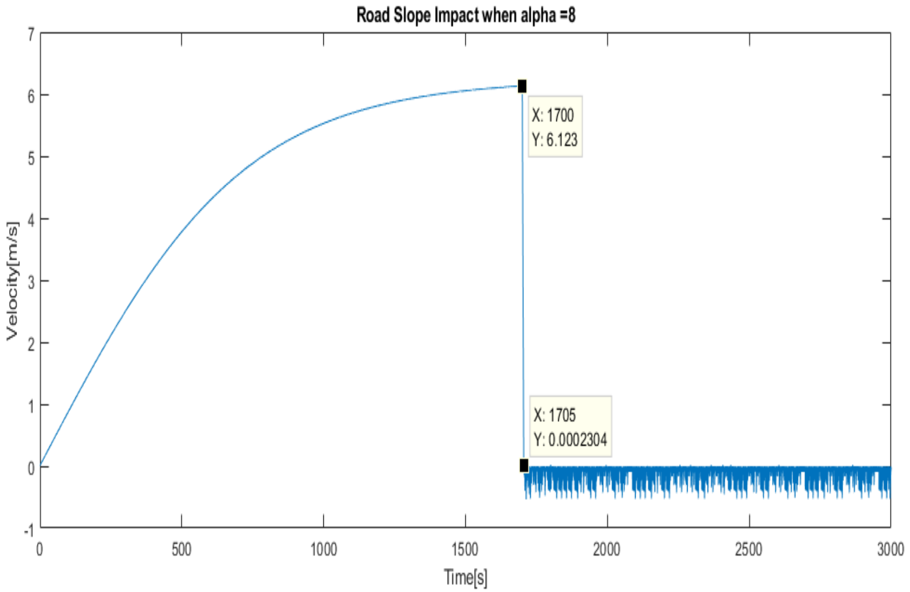
- Tm1 (first torque input: 39Nm) is applied at 0 sec.
- There is negative wind torque applied at 1700sec.
  - Vwind (wind disturbance input = 10Nm)
  - The speed is decreased to approximately 9.3 m/sec at 2500 sec (taking 800sec) with wind disturbance.
  - The wind resistive force is not strong enough to stop the car while positive torque is still applying



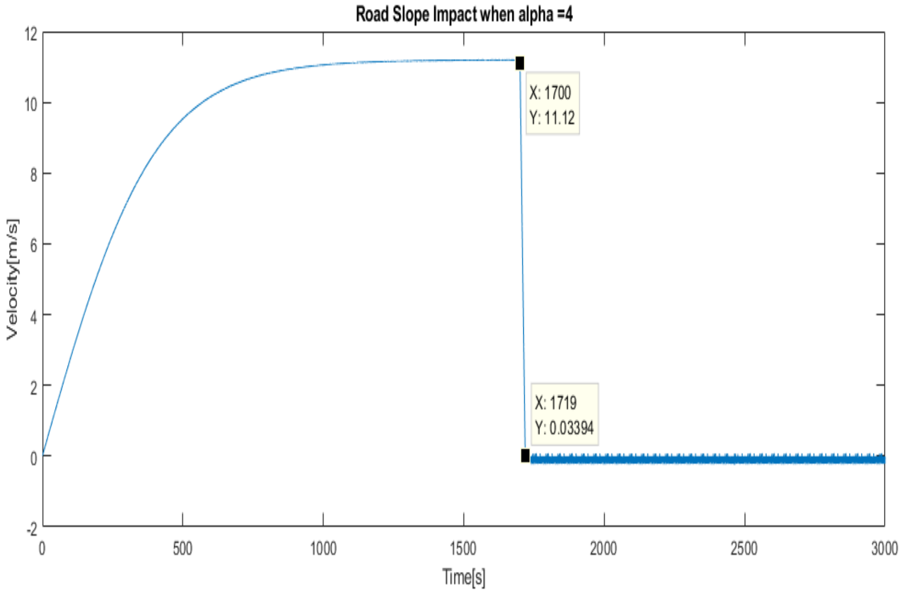
CASE 13: ROAD SLOPE IMPACT AT 15 MPH

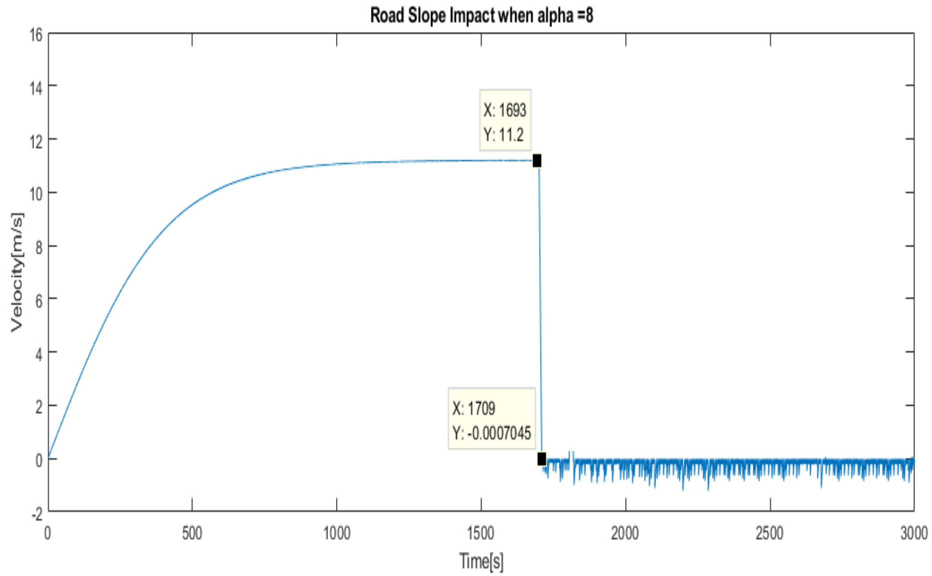
- Tm1 (first torque input: 25.5Nm) is applied at 0 sec.
- The slope1 is encountered at 1700sec.
    - alpha (road slope input: 4 degree) is applied at 1700 sec.
    - The speed is decreased to zero at 1710 sec (taking 10sec) .
- The slope2 is encountered at 1700sec.
    - alpha (road slope input: 8 degree) is applied at 1700 sec.
    - The speed is decreased to zero at 1705 sec (taking 5sec) .
- As slope increase, car slows down more rapidly.





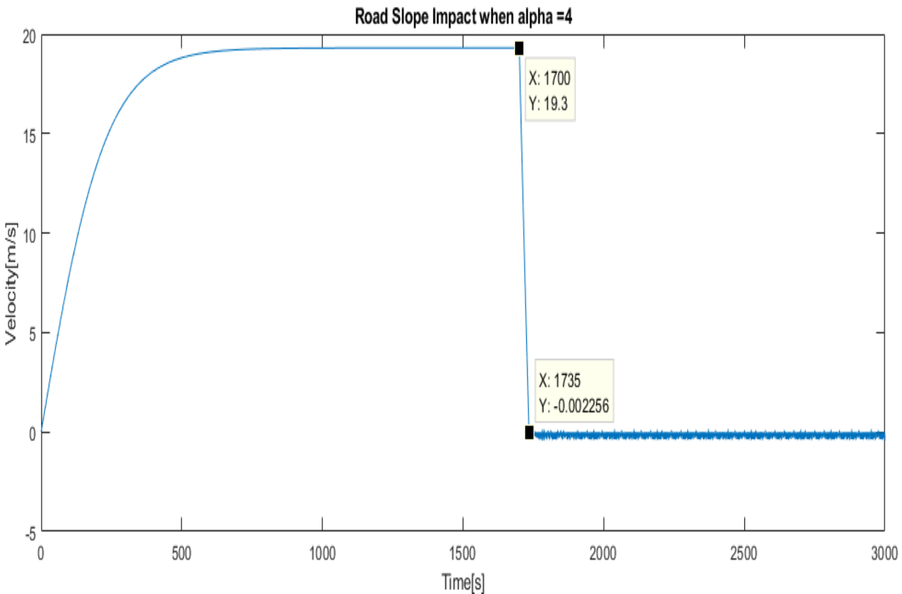
- Tm1 (first torque input: 29Nm) is applied at 0 sec.
- The slope1 is encountered at 1700sec.
  - alpha (road slope input: 4 degree) is applied at 1700 sec.
  - The speed is decreased to zero at 1719 sec (taking 19sec) .
- The slope2 is encountered at 1700sec.
  - alpha (road slope input: 8 degree) is applied at 1700 sec.
  - The speed is decreased to zero at 1709 sec (taking 9sec) .
- As slope increase, car slows down more rapidly.

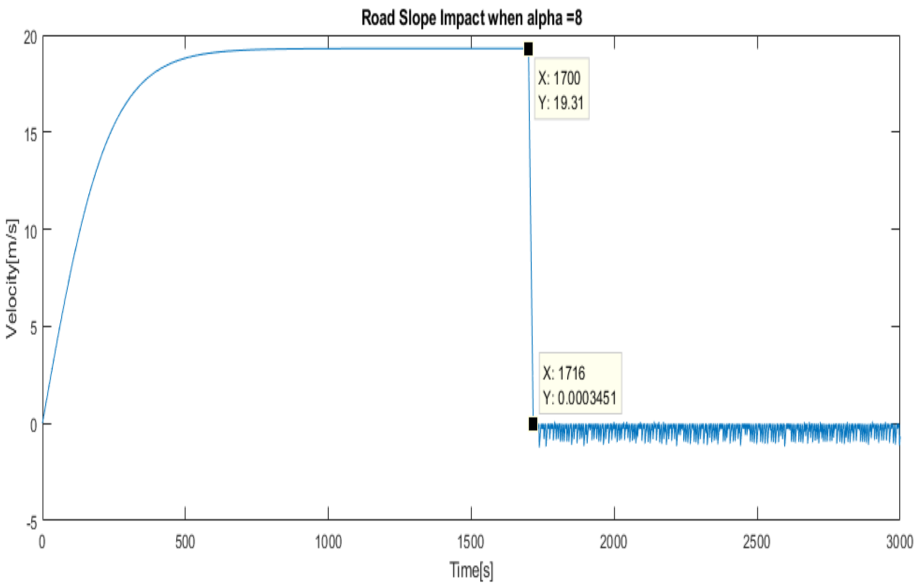




CASE 15: ROAD SLOPE IMPACT AT 45 MPH

- Tm1 (first torque input: 39Nm) is applied at 0 sec.
- The slope1 is encountered at 1700sec.
  - alpha (road slope input: 4 degree) is applied at 1700 sec.
  - The speed is decreased to zero at 1735 sec (taking 35sec) .
- The slope2 is encountered at 1700sec.
  - alpha (road slope input: 8 degree) is applied at 1700 sec.
  - The speed is decreased to zero at 1716 sec (taking 16sec) .
- As slope increase, car slows down more rapidly.





THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

LOW LEVEL LATERAL CONTROL

SHIHAO LIU

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

- Constants used in the models (Parameters)

- Dugoff Tire Model
μ : tire friction coefficient (0.7)
F_z: vertical force = M/4(2458.924 kg/4)*g(9.81N/m^2)
ε_r: 0
C_x: longitudinal stiffness (3e5N)
C_y: cornering stiffness (1.5e5N/rad)
Wheel base: 2964.18 mm
Turning circle: 5917 mm
Vehicle length: 2270.76 mm

- Wheel Dynamics
η_t : Transmission Efficiency (0.9)
λ_t : Gear ratio (1.0)
λ_f : Final drive ratio (4.1)
R_w: effective wheel radius (0.3 m)
I_w : Inertia of the wheel (1 kgm^2)
M : mass of the vehicle (M = 2458.924kg)
M_{eq} : equivalent mass factor - Load forces
ρ : Air density (1.225 kgm^3)
A_f: Frontal area of vehicle (2.8 m^2)
C_d: Aerodynamic drag coefficient (0.29)
C_{rr}: Rolling resistance coefficient (0.015)

- Load forces
ρ : Air density (1.225 kgm^3)
A_f: Frontal area of vehicle (2.8 m^2)
C_d: Aerodynamic drag coefficient (0.29)
C_{rr}: Rolling resistance coefficient (0.015)

- Equations used for modelling
  - Steering angle projection

$$\begin{bmatrix} \Sigma F_x \\ \Sigma F_y \\ \Sigma M_z \end{bmatrix} = \begin{bmatrix} -\sin\delta_f & -\sin\delta_r \\ \cos\delta_f & \cos\delta_r \\ l_f\cos\delta_f & -l_r\cos\delta_r \end{bmatrix} \begin{bmatrix} F_f \\ F_r \end{bmatrix}$$

  - Kinetics/Geometry

$$\beta_f = \tan^{-1}\left(\tan\beta + \frac{\dot{\Psi} l_f}{v\cos\beta}\right)$$

$$\beta_r = \tan^{-1}\left(\tan\beta - \frac{\dot{\Psi} l_r}{v\cos\beta}\right)$$

  - Dynamics

$$\begin{bmatrix} mv\left(\dot{\beta} + \dot{\Psi}\right) \\ m\dot{v} \\ J\ddot{\Psi} \end{bmatrix} = \begin{bmatrix} -\sin\beta & \cos\beta & 0 \\ \cos\beta & \sin\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma F_x \\ \Sigma F_y \\ \Sigma M_z \end{bmatrix}$$

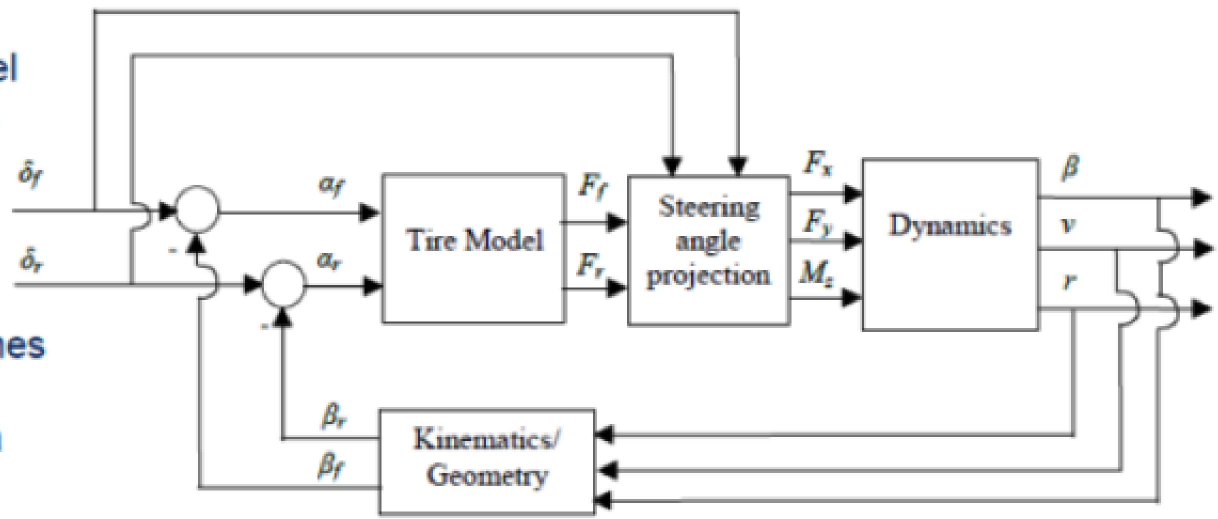
- Equations used for modelling
  - Linear single track model

$$\begin{bmatrix}\dot{\beta}\\ \dot{r}\end{bmatrix}=\begin{bmatrix}\frac{-C_f-C_r}{mV} & -1+\left(\frac{C_r l_r-C_f l_f}{mV^2}\right)\\ \frac{C_r l_r-C_f l_f}{I_z} & \frac{-C_f l_f^2-C_r l_r^2}{I_z V}\end{bmatrix}\begin{bmatrix}\beta\\ r\end{bmatrix}+\begin{bmatrix}\frac{C_f}{mV} & \frac{C_r}{mV}\\ \frac{C_f l_f}{I_z} & \frac{C_r l_r}{I_z}\end{bmatrix}\begin{bmatrix}\delta_f\\ \delta_r\end{bmatrix}+\begin{bmatrix}0\\ \frac{1}{I_z}\end{bmatrix}M_z$$

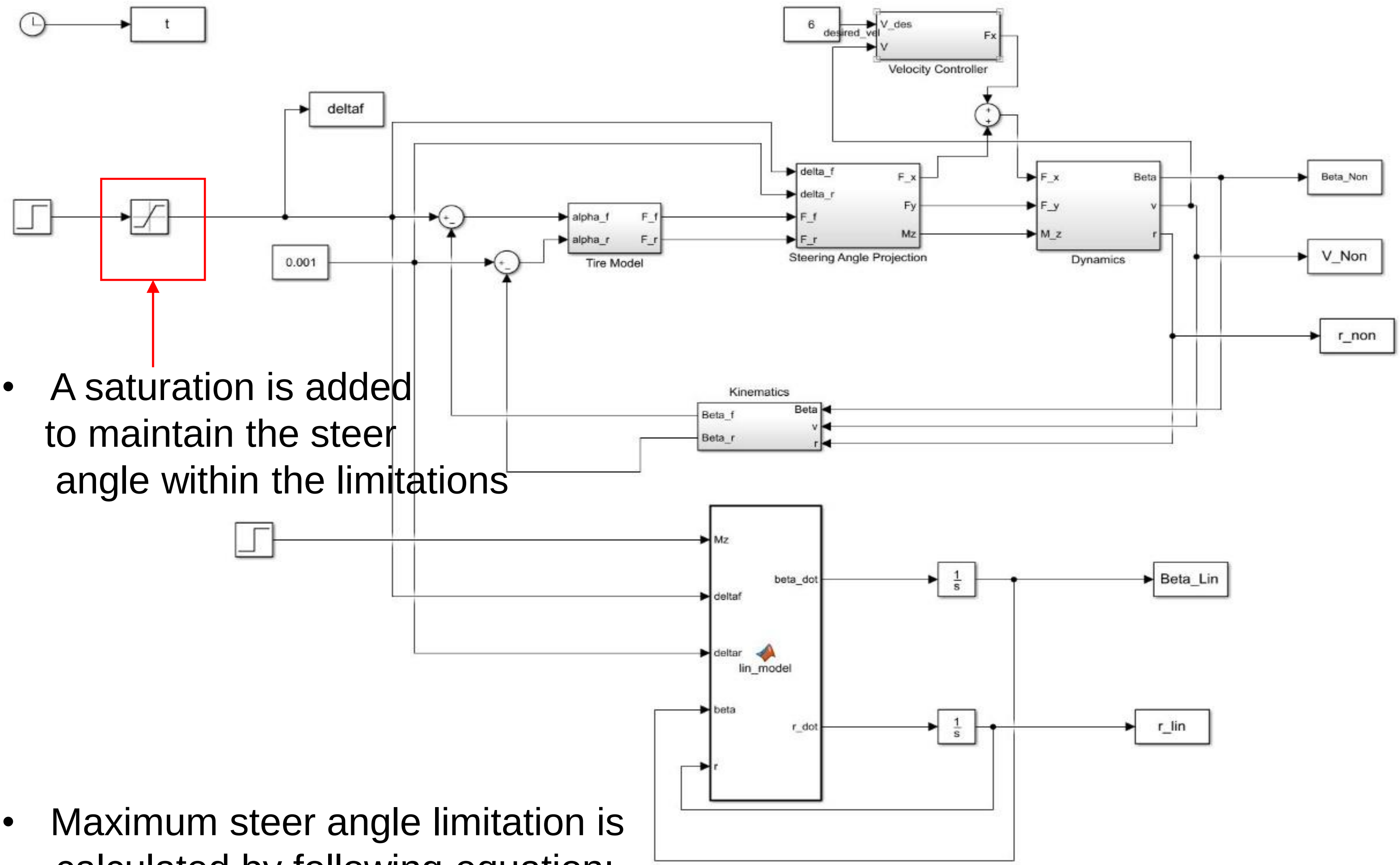
  - Lateral Vehicle Model design reference

Steering wheel commands δ_f.

δ_r usually comes from control system or is a fraction of δ_f.



SCREEN SHOT OF THE SIMULINK MODEL



- A saturation is added to maintain the steer angle within the limitations

- Maximum steer angle limitation is calculated by following equation:
  - atan(wheelbase/(turning circle – car width))=0.6826 rad

SIMULATION PATH

Figure 1a: Map of Easton Trolley Stop Locations and Route

- At this curve, the vehicle is going to turn left and go straight. So there will be a steering angle and then the steering angle will be zero.
- For this curved path, 20ft turning circle is applied for 30 seconds. 0ft turning circle is applied for the rest of the simulation.
- The speed limit is 15mph (6.83 m/s).
- The vehicle is expected to have a lower velocity in this sharpen right turn. So the velocity desired is set to 3 m/s.

MATLAB CODE FOR TESTING UNDERSTEERING

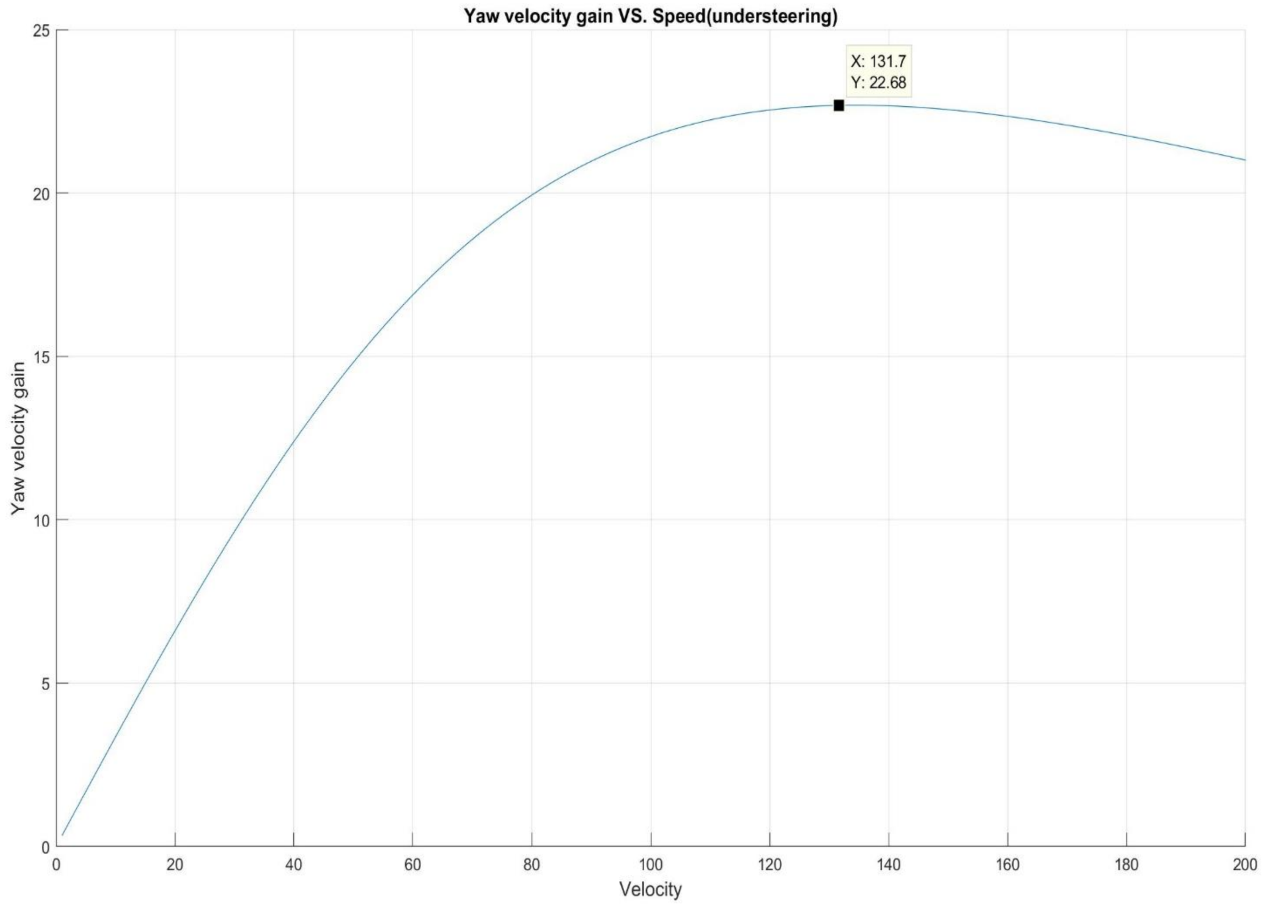
```
v=linspace(1,200);
g=9.81;
m=2458.924;
cf=2*1.5e5;
cr=2*1.5e5;
lf=2.96418*0.49;
lr=2.96418*0.51;
L=2.96418;

Kus=((m*g*lr)/(cf*L))-((m*g*lf)/(cr*L)); %understeering
gradient
Gyaw=v./(L+Kus*v.^2/g);

figure;
hold on;
grid on;
title('Yaw velocity gain VS.
Speed(understeering)','fontsize',12);

plot(v,Gyaw);
ylabel('Yaw velocity gain','fontsize',12)
xlabel('Velocity','fontsize',12)
```

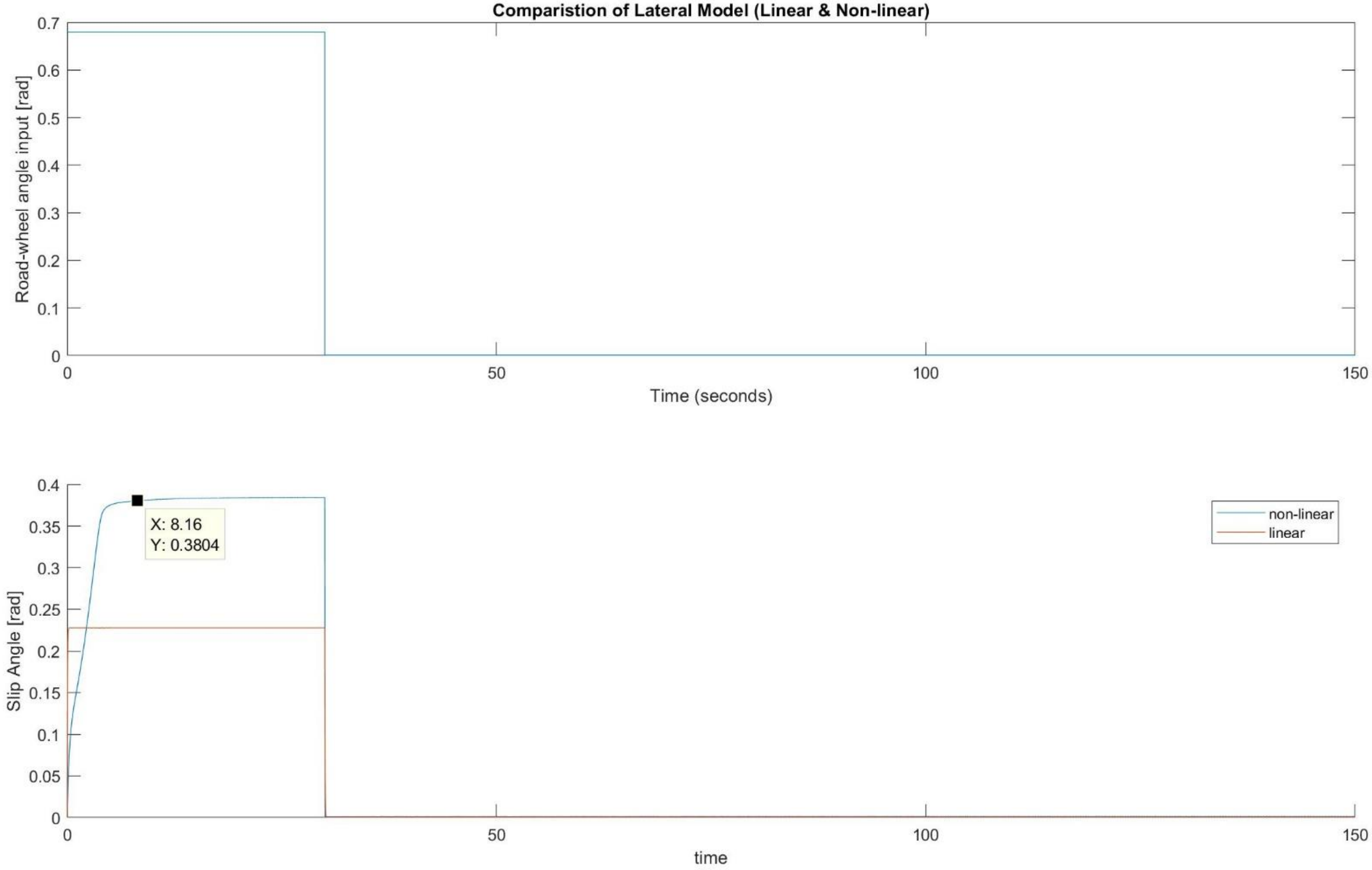
SIMULATION RESULTS IN TESTING UNDERSTEERING



- According to the simulation result with vehicle parameters, we can see that the vehicle is in understeering.

SIMULATION RESULTS TO QUESTIONS ASKED

PATH SIMULATION

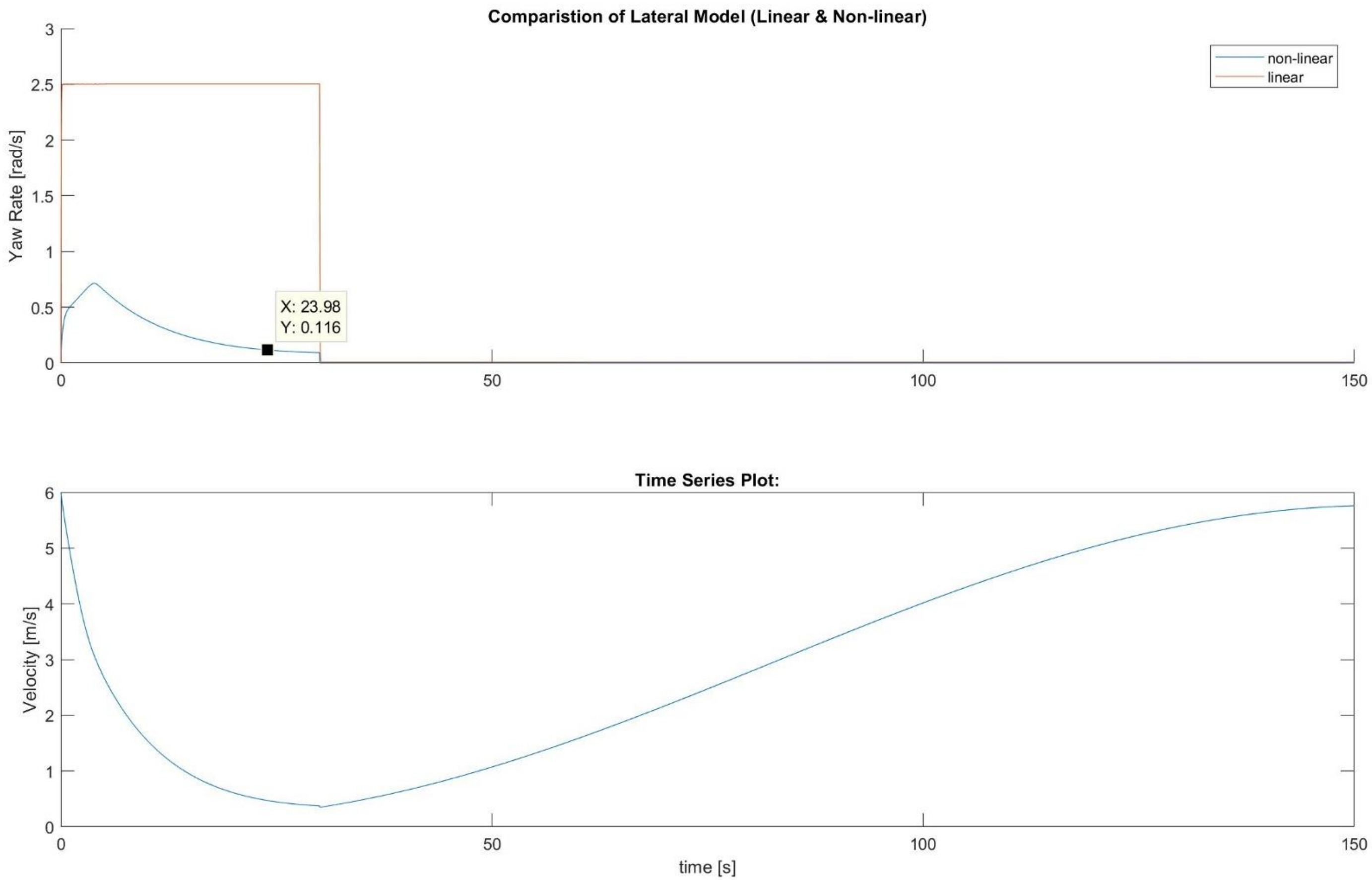


- The turning range is 20 feet. And the calculated steering angle is 0.68 rad. So the road-wheeling angle input is set to be 0.68 for the first 30 seconds.

- It takes 8.16 second for the slip angle of non-linear model to be converged while the linear model is converged almost immediately.

SIMULATION RESULTS TO QUESTIONS ASKED

PATH SIMULATION

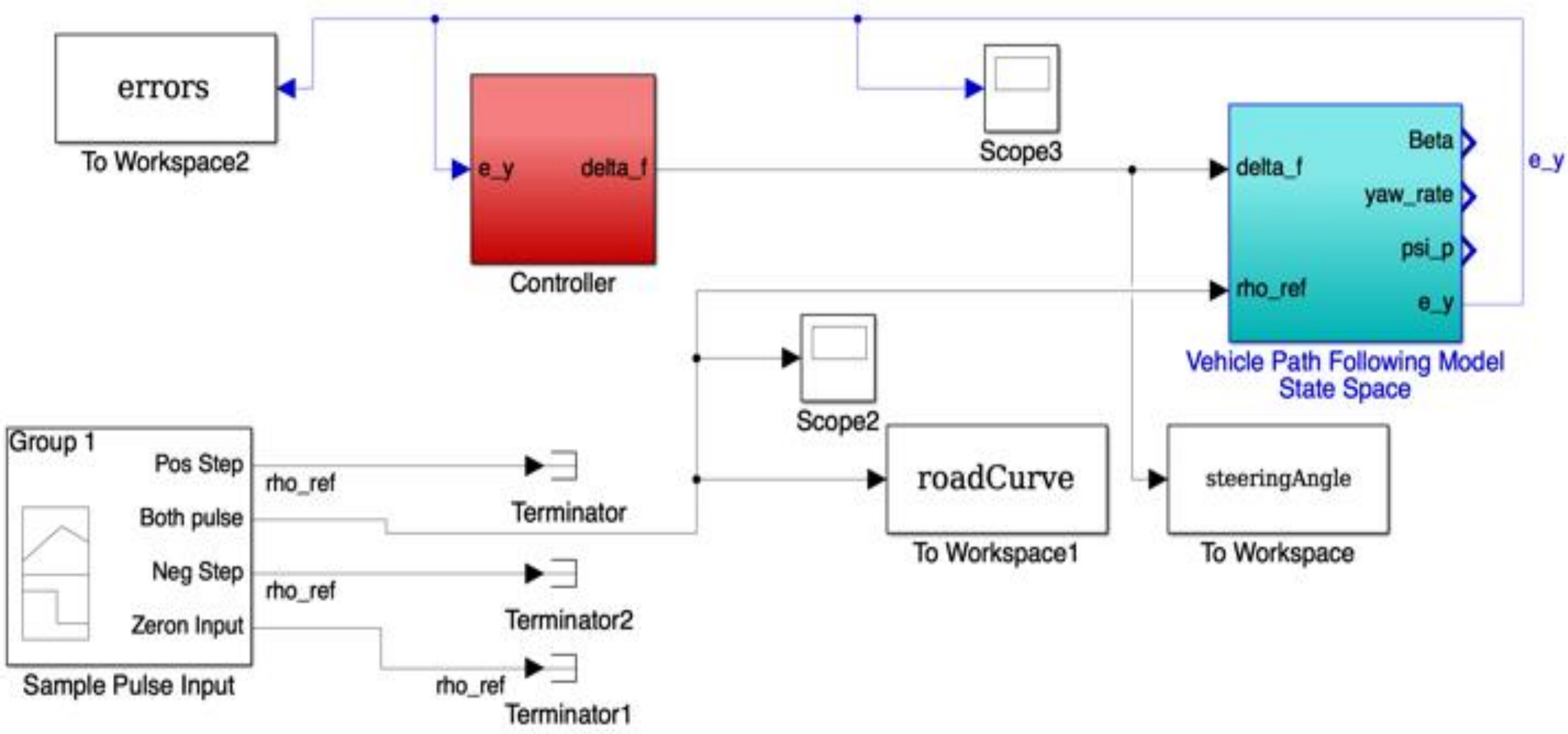


- It takes 23.98 second for the yaw rate of non-linear model to be converged while the linear model is converged almost immediately.
- The maximum of the velocity will be higher that the desired velocity but still lower that the legal speed limit.

PATH PLANNING AND FOLLOWING

TEAWON HAN

- Screen shot of the model created for PID tuning



PID GAINS

- What was your final PID gains that you have chosen?
  - Gain values
    - Proportional (P) : -1.65165435076644
    - Integral (I) : -1.56883467288441
    - Derivative (D) : -0.00627956510850705
    - Filter coefficient (N) : 1142.8279760416

- What was the constrain/conditions used for the tuning?
  - Bandwidth (rad/s) : 10
  - Phase Margin (deg): 60
  - Velocity of the vehicle (V) : 5m/s
  - Preview length : 2 m

- Explain your curve fit methodology with curvature and the slope of the curvature graph
  - Given map, Longitudinal and Lateral(LL) coordinates were obtained.
  - The set of LL coordinates is transformed to UTM coordinates
  - Beizer Curve (13 segments) were implemented.

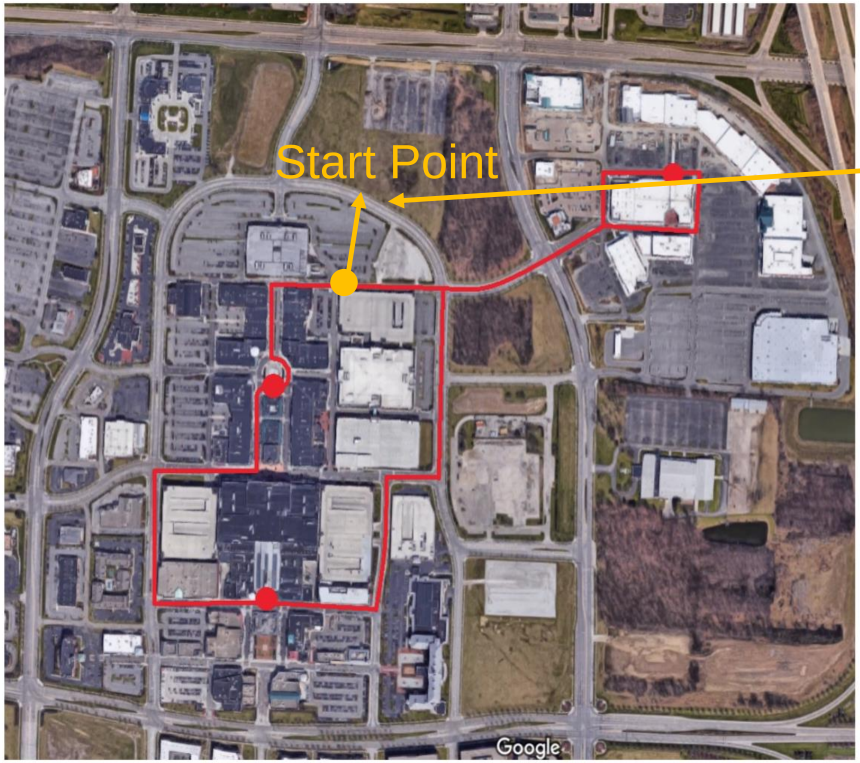
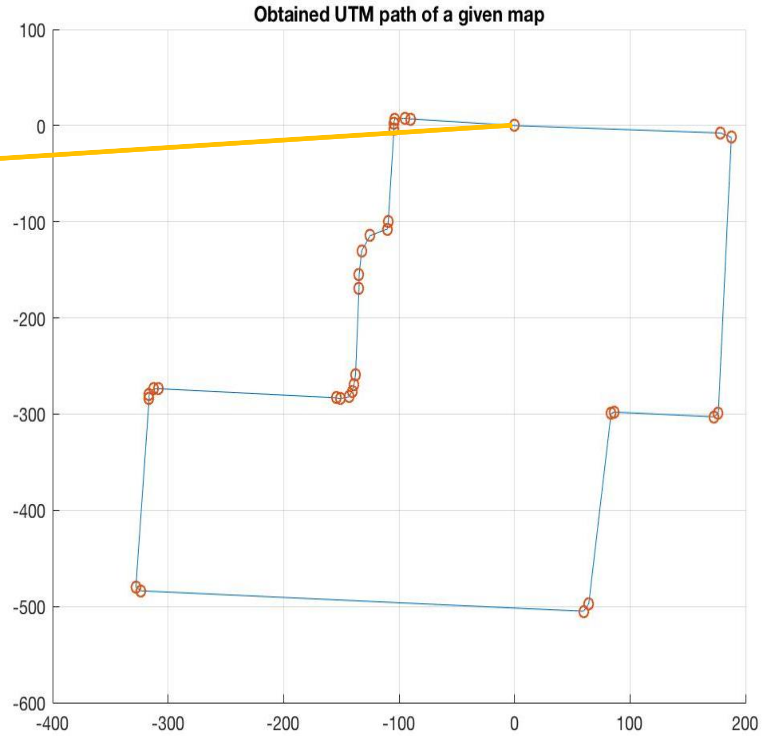
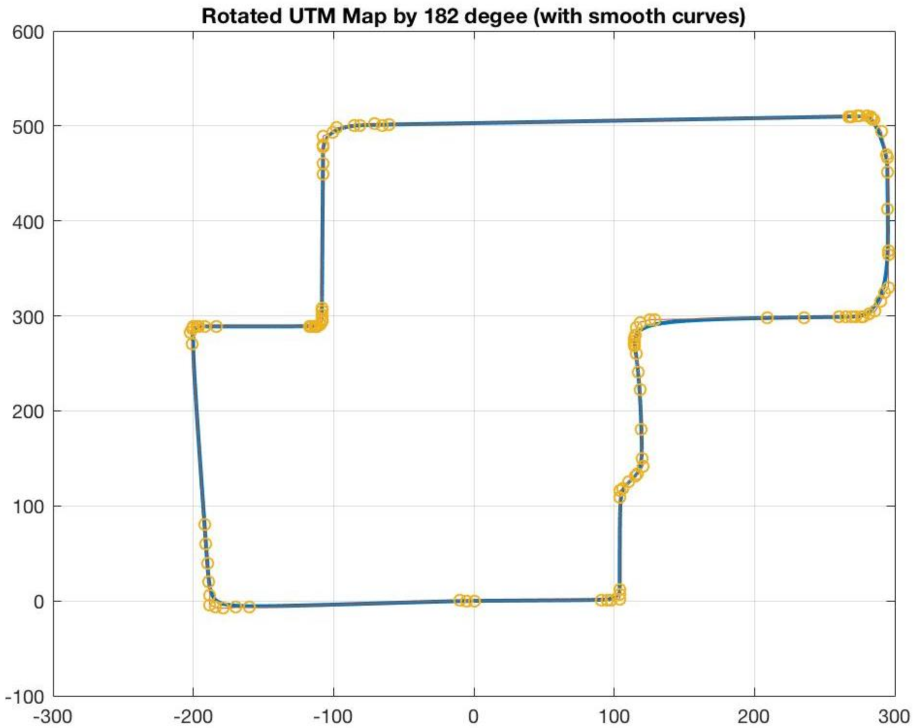
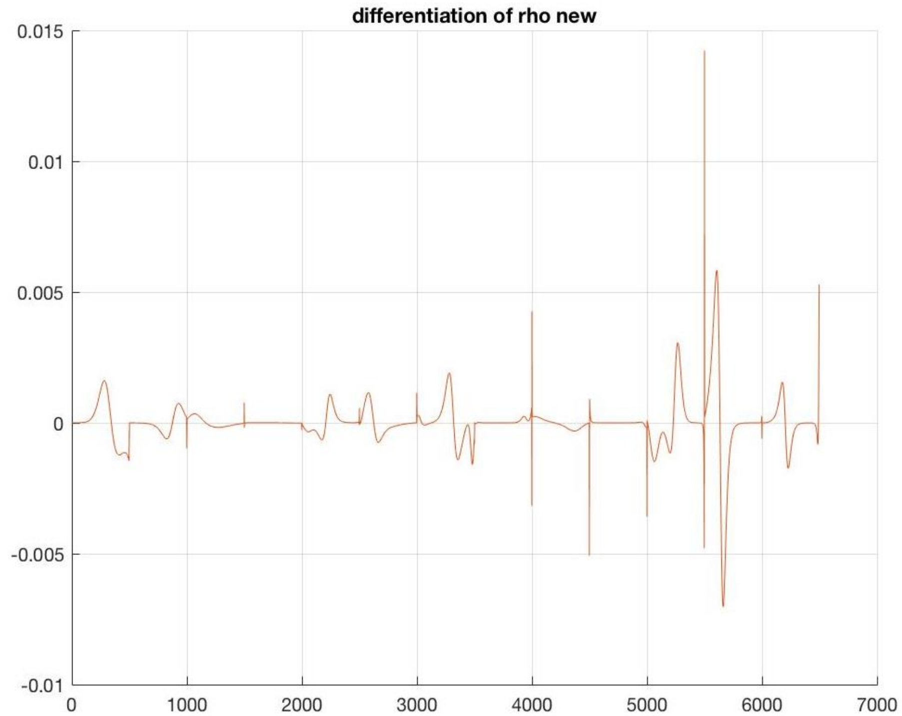


Figure 1a: Map of Easton Trolley Stop Locations and Route



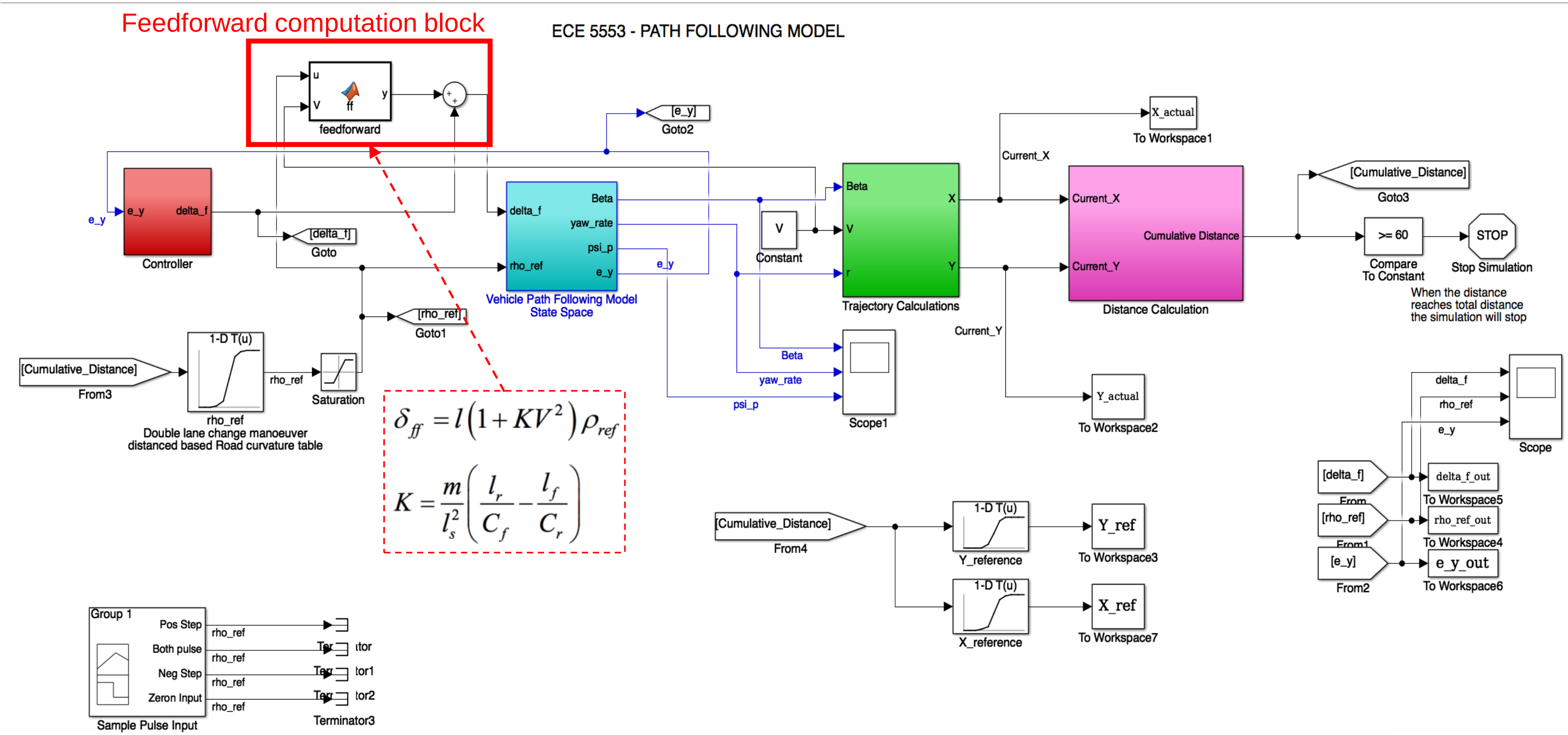
- Smoothing Curves
  - Genetic Algorithm was implemented to find appropriate points such that minimize the slope of road curvature.
  - UTM map is rotated (CCW) by 185 degrees for convenience.
  - Differentiation of curvatures in path is less than 0.015.



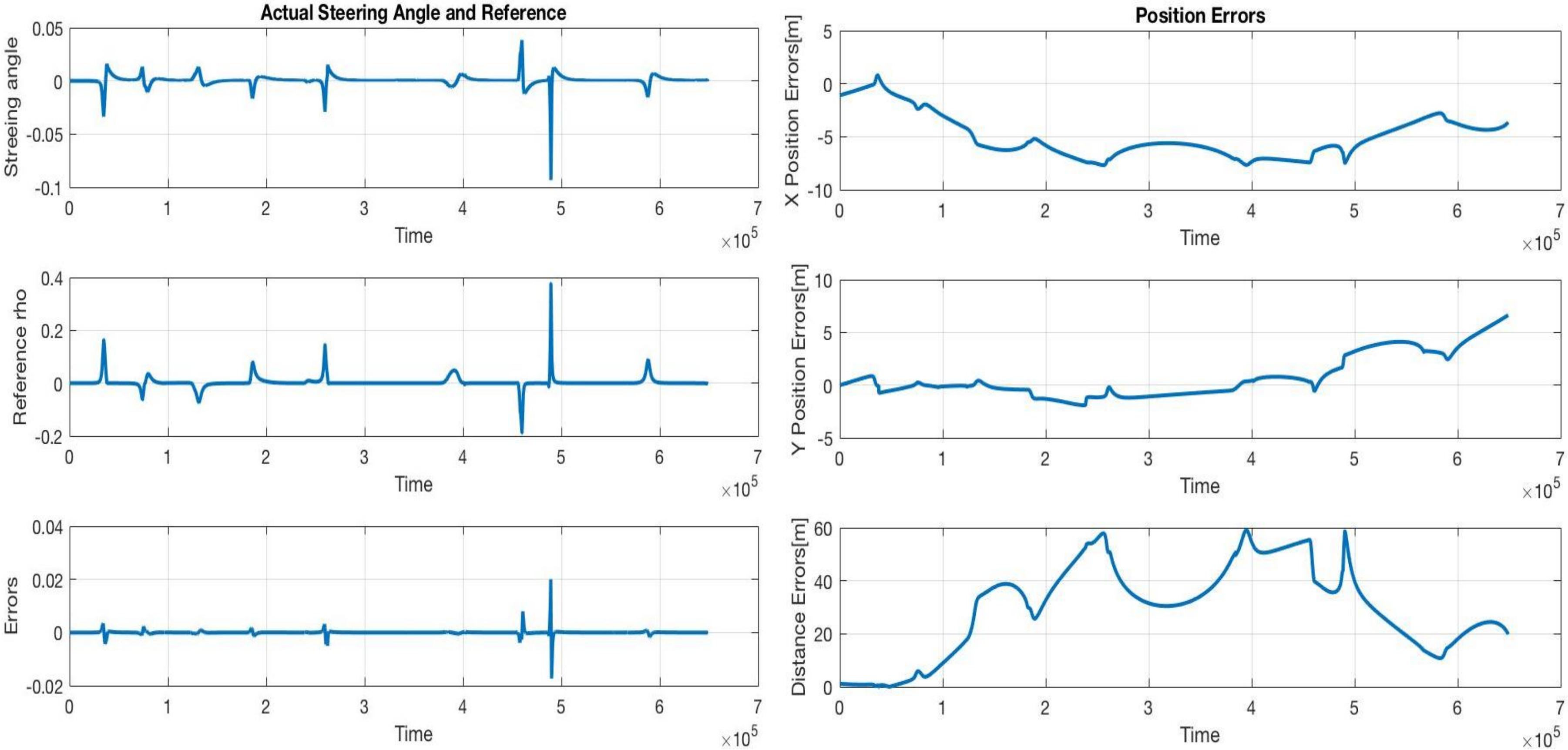


PATH FOLLOWING MODEL

- Feedforward computation part is added to the path following model which was given in ECE5553 HW4
- Velocity and preview length are set 2m/s and 10 m respectively.
- Saturation range is extended

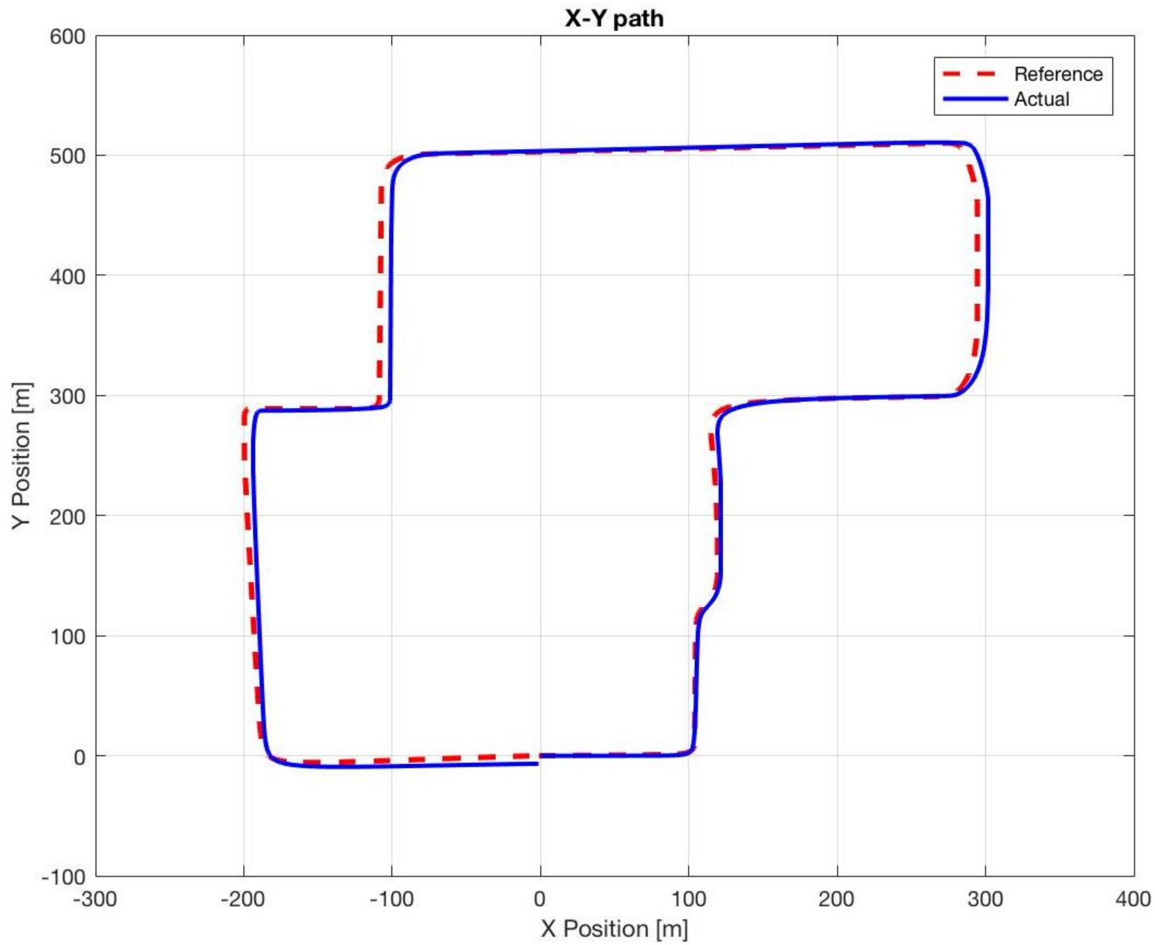


SIMULATION RESULTS



- Controller follows the path, but steering errors are accumulated resulting huge difference errors over time.

SIMULATION RESULTS



- Currently, controller is just following predefined steering angle, but the autonomous vehicle can generate more control inputs to compensate errors by using perceived information and localization techniques. (e.g., lane detection, 3D map feature matching)

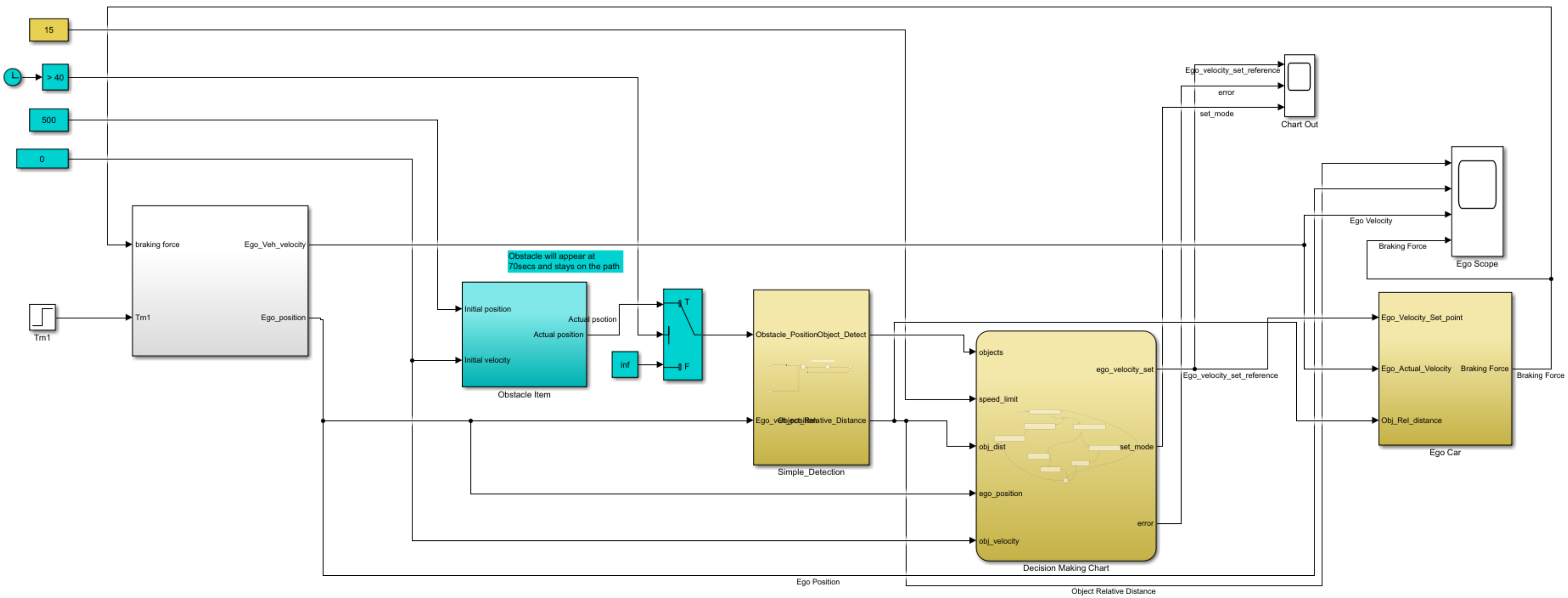
THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

COLLISION AVOIDANCE

EVAN STODDART

COLLISION AVOIDANCE

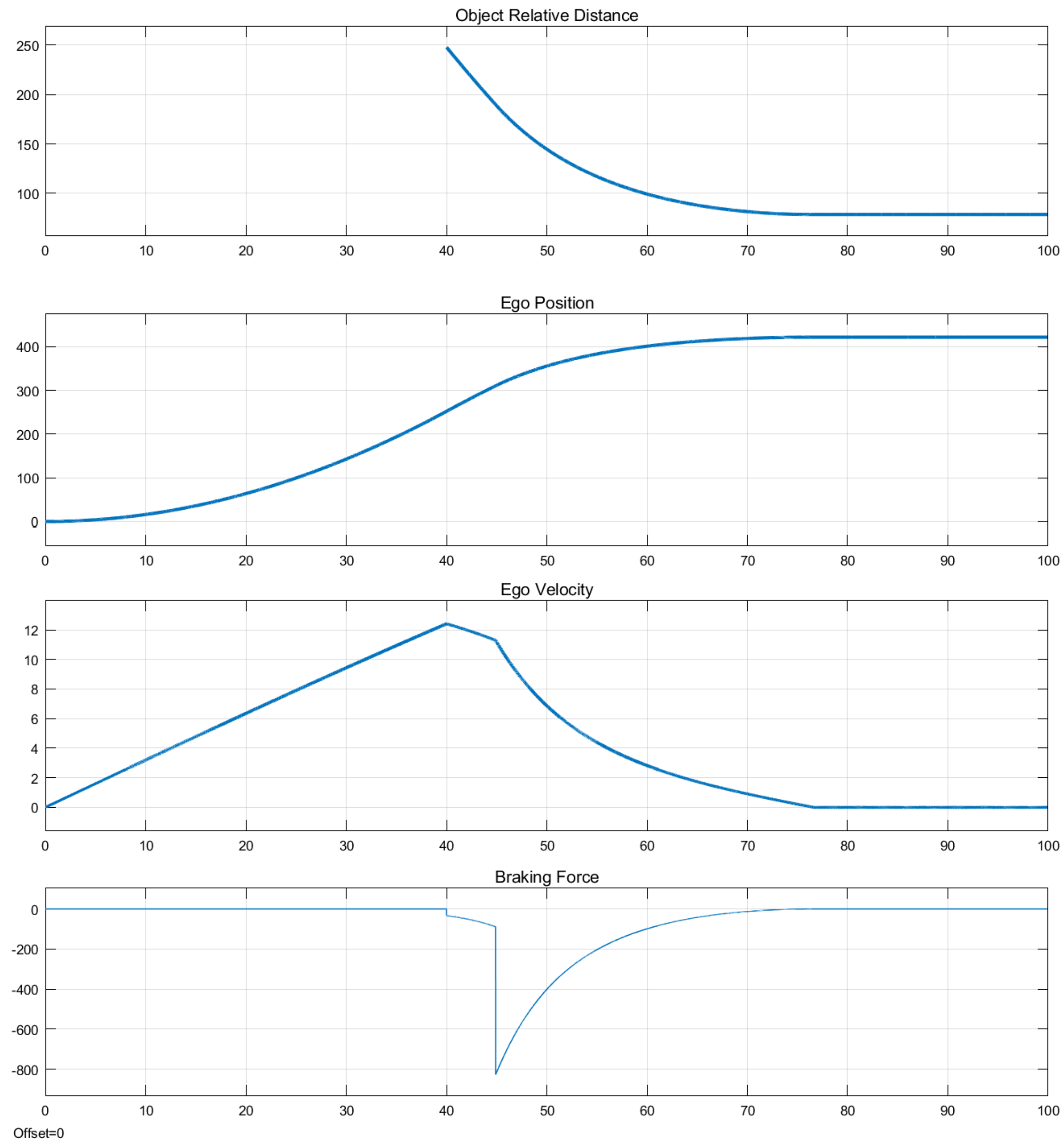
AUTONOMOUS SHUTTLE COLLISION AVOIDANCE SYSTEM



- **Model simulates the autonomous shuttle encountering a stopped object**
  - A vehicle at a red light
  - A vehicle dropping off a passenger at the shopping mall
  - A pedestrian crossing the street
- **Scenario**
  - The shuttle receives a step torque input
  - The shuttle detects a stopped vehicle starting at a detection range of 190m
  - The shuttle will make a decision based on the situation
    - If obstacle is a moving object, autonomous shuttle will maintain a safe distance.
    - If obstacle is a static object, autonomous shuttle will stop immediately to avoid collision.

COLLISION AVOIDANCE

- Simulation:
  - The shuttle receives an arbitrary torque request and begins accelerating to the speed limit
  - The vehicle becomes visible at 40 seconds
  - The shuttle sensor identifies the vehicle shortly after (when in range)
- Results:
  - The shuttle enters emergency braking
  - The shuttle applies a brake force to the longitudinal model
  - The shuttle descends towards a stop behind the target object
- Conclusion:
  - This model simulates a collision avoidance scenario using the shuttles longitudinal model parameters and a collision avoidance decision making strategy .



The Ohio State University
CENTER FOR AUTOMOTIVE RESEARCH

LOCALIZATION AND PERCEPTION

EVAN STODDART

LOCALIZATION

GPS with Google Maps Application Programming Interface (API) Integration will be the primary localization method for the shuttle. GPS can provide accurate road positions for the shuttle during its route. Although excessive for such a small route, implementing GPS framework allows the shuttle concept to be ported to other shopping malls in other cities easily due to the robustness of GPS systems.

The shuttle will utilize a gyroscope and accelerometer in the form of an inertial measurement unit (IMU). This will allow the shuttle to track its own movement and utilize dead reckoning interpolation in between GPS updates. Furthermore, this tactic can be used as a secondary positioning system in case the GPS sensor loses connection.

In MATLAB and Simulink

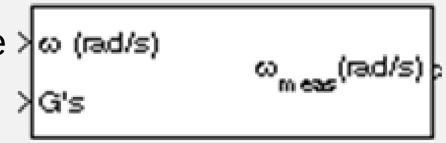
The GPS can be modeled using the 3rd party **GPS Navigation Toolbox.** This will be useful for modeling trilateration for GPS navigation systems as well as error and simulation.

Source:

https://www.mathworks.com/matlabcentral/fileexchange/41364-gps-navigation-toolbox

In MATLAB and Simulink

A Gyroscope sensor can be simulated in Simulink using the following module.



Parameters:

Second order dynamics - Select to apply second-order dynamics to gyroscope readings.

Natural frequency (rad/sec) - The natural frequency of the gyroscope.

Damping ratio - The damping ratio of the gyroscope.

Scale factors and cross-coupling - The 3-by-3 matrix used to skew the gyroscope.

Measurement bias (rad/s) - three-element vector bias

G-sensitive bias (rad/s/g-unit) – bias due to change in rates.

Update rate (sec) - Specify gyroscope update rate

Noise On - apply white noise to gyroscope readings.

Noise seeds - The scalar seeds for the Gaussian noise generator for each axis of the gyroscope.

Noise power (rad/s)^2/Hz - The height of the PSD of the white noise for each axis of the gyroscope.

Lower and upper output limits (rad/s) - angular rates in each of the gyroscope axes.

This Module is a comprehensive solution to gyroscopic sensor processing.

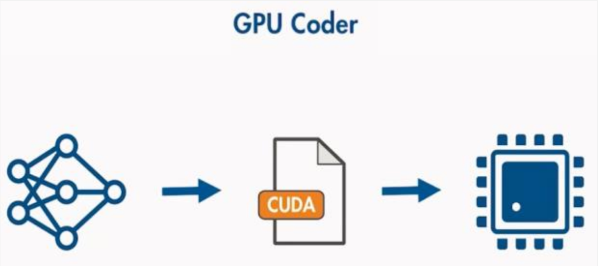
Source:

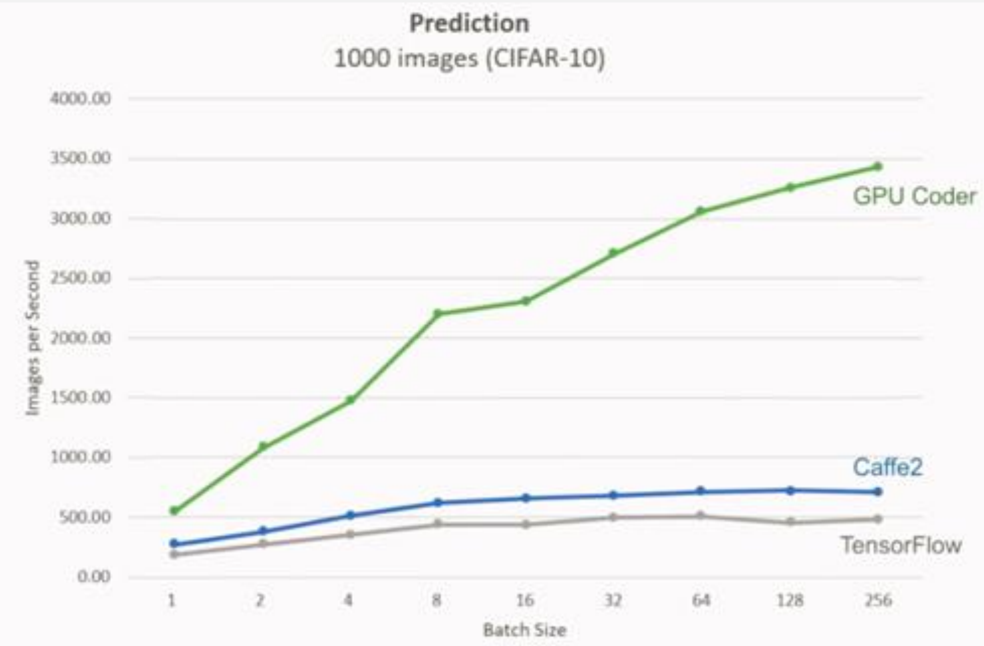
https://www.mathworks.com/help/aeroblks/threeaxisgyroscope.html

PERCEPTION: OBJECT DETECTION USING NEURAL NETWORKS

A highly parallel embedded computer will run cross-compiled MATLAB object detection code using the **Neural Network Toolbox™.** This supports quick, deployable, state-of-the art Detection algorithms using pretrained models Such as Google's Inception model. This model Will detect both cars and pedestrians as well as dozens of other objects.

The toolbox also supports MATLABs **GPU Coder.** This will allow the team to deploy networks with inherent Nvidia CUDA GPU optimization. This approach far exceeds other competing deep learning solutions such as Tensorflow and Caffe.

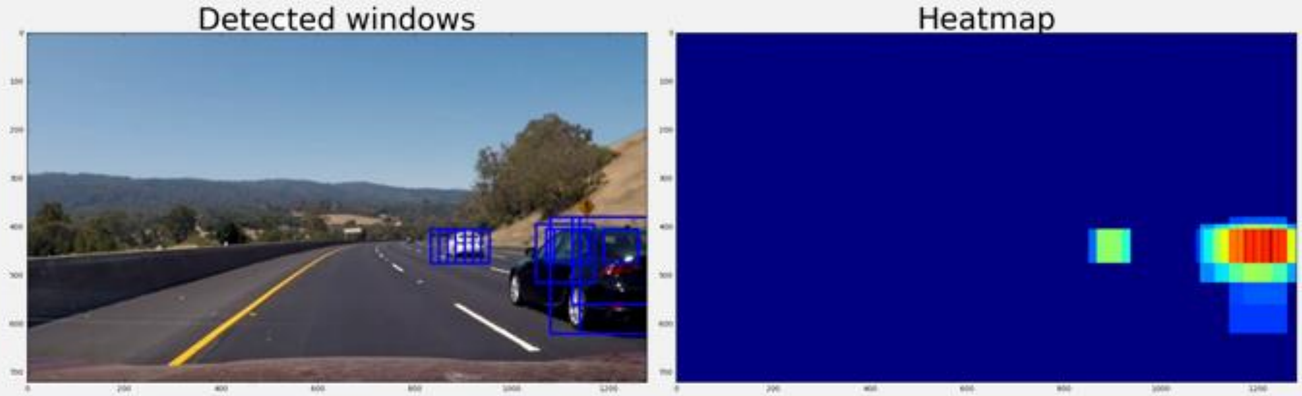




PERCEPTION: TRACKING

THE OHIO STATE UNIVERSITY
CENTER FOR AUTOMOTIVE RESEARCH

To eliminate false positive, a heat map filter will be implemented to reduce the "false alarm" rate in which the autonomous shuttle would accidentally think an obstacle exists. A heat map filter is a low pass digital filter that outputs only high-concentrations of object detections from a Classifier such as the neural network discussed on the last frame.



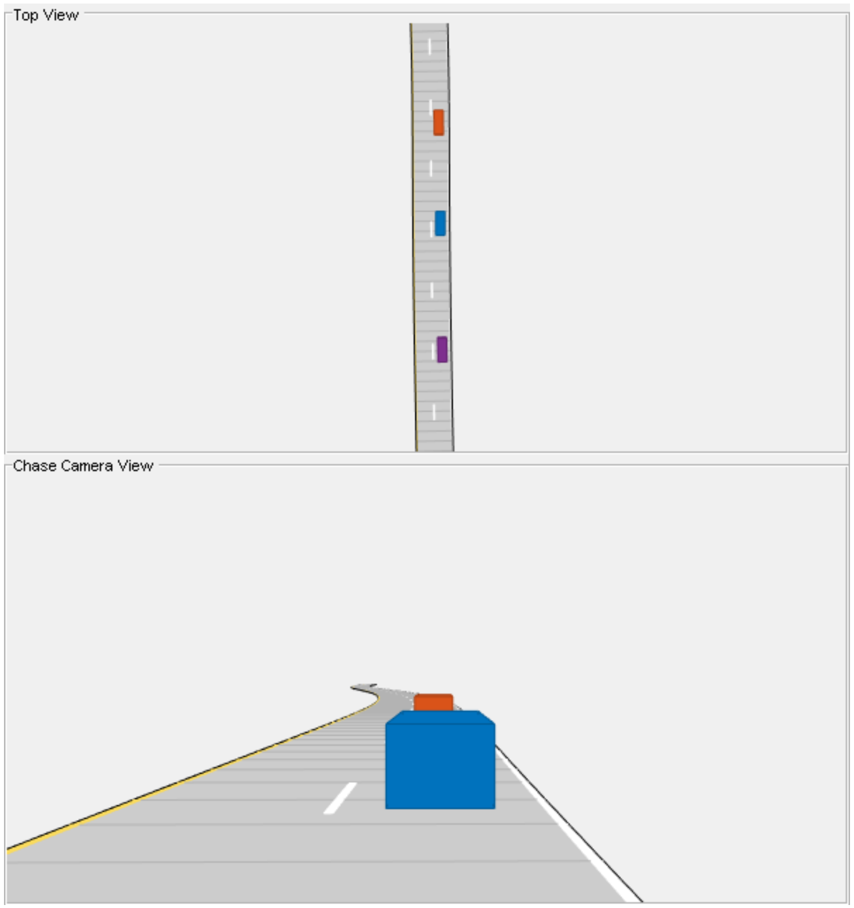
Optical Flow is an algorithm to compare pixel-by-pixel from frame-to-frame to see how an object has moved throughout a video stream. In MATLAB, the **Computer Vision System Toolbox** includes a framework for optical flow. This is useful because objects identified in the neural network will be run through the heat map filter above and then tracked using optical flow. This will give a robust solution to track objects using a camera.

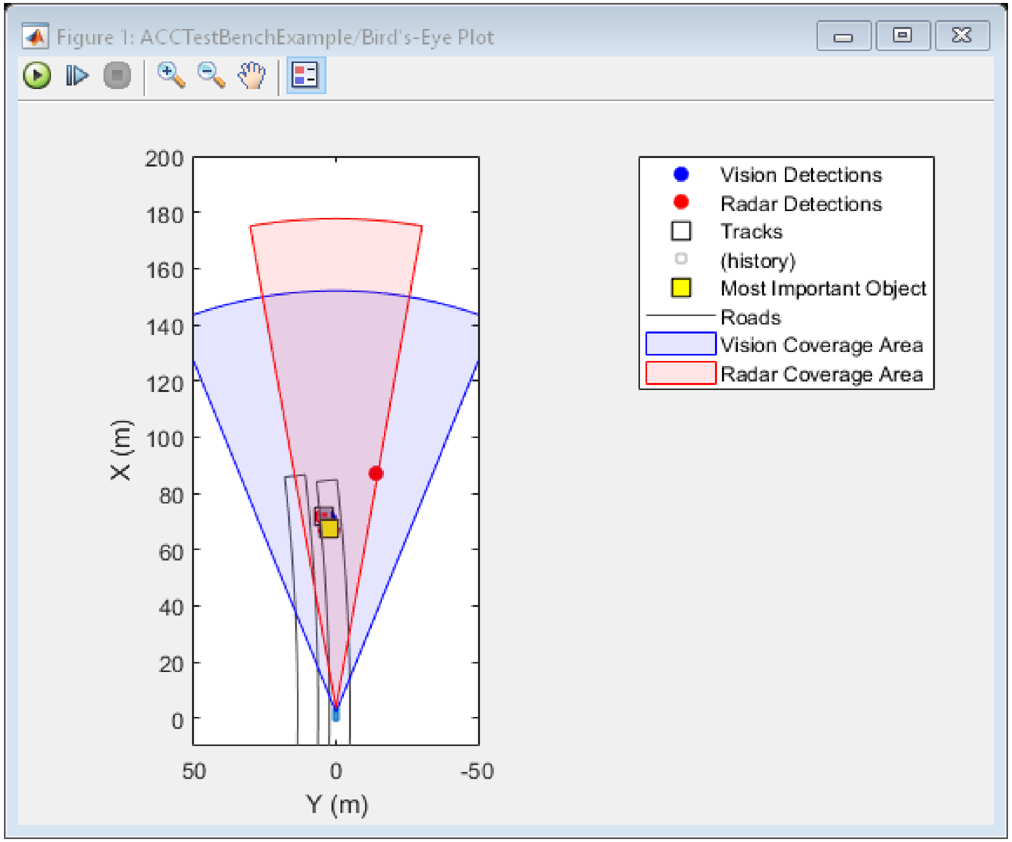
PERCEPTION: SENSOR FUSION AND RANGING

The Ohio State University
CENTER FOR AUTOMOTIVE RESEARCH

Radar provides the range estimate for sensor fusion. The radar can be simulated in MATLAB using synthetic data in a driving scenario.

Example:
https://www.mathworks.com/help/driving/examples/sensor-fusion-using-synthetic-radar-and-vision-data.html





With a model set to process camera and radar data, as well as other sensors that may be in the system, sensor fusion techniques can then form meaning results by combining the datasets. Sensor fusion algorithms can be prototyped in MATLAB using the **Automated Driving Toolbox**.

THE OHIO STATE UNIVERSITY

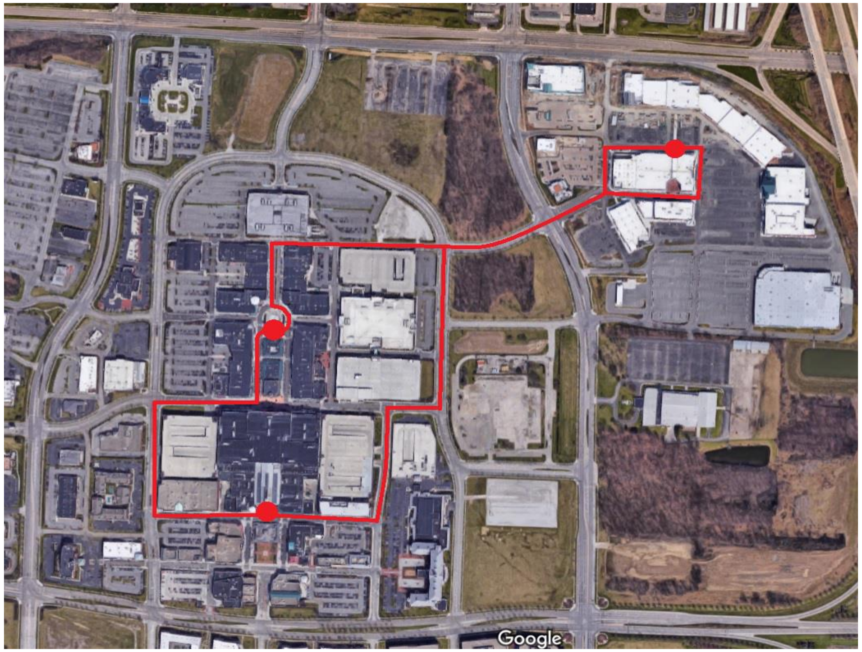
CENTER FOR AUTOMOTIVE RESEARCH

STATE FLOW DECISION MAKING

DEREK LONGSHORE

FINITE STATE MACHINE

The Ohio State University
CENTER FOR AUTOMOTIVE RESEARCH

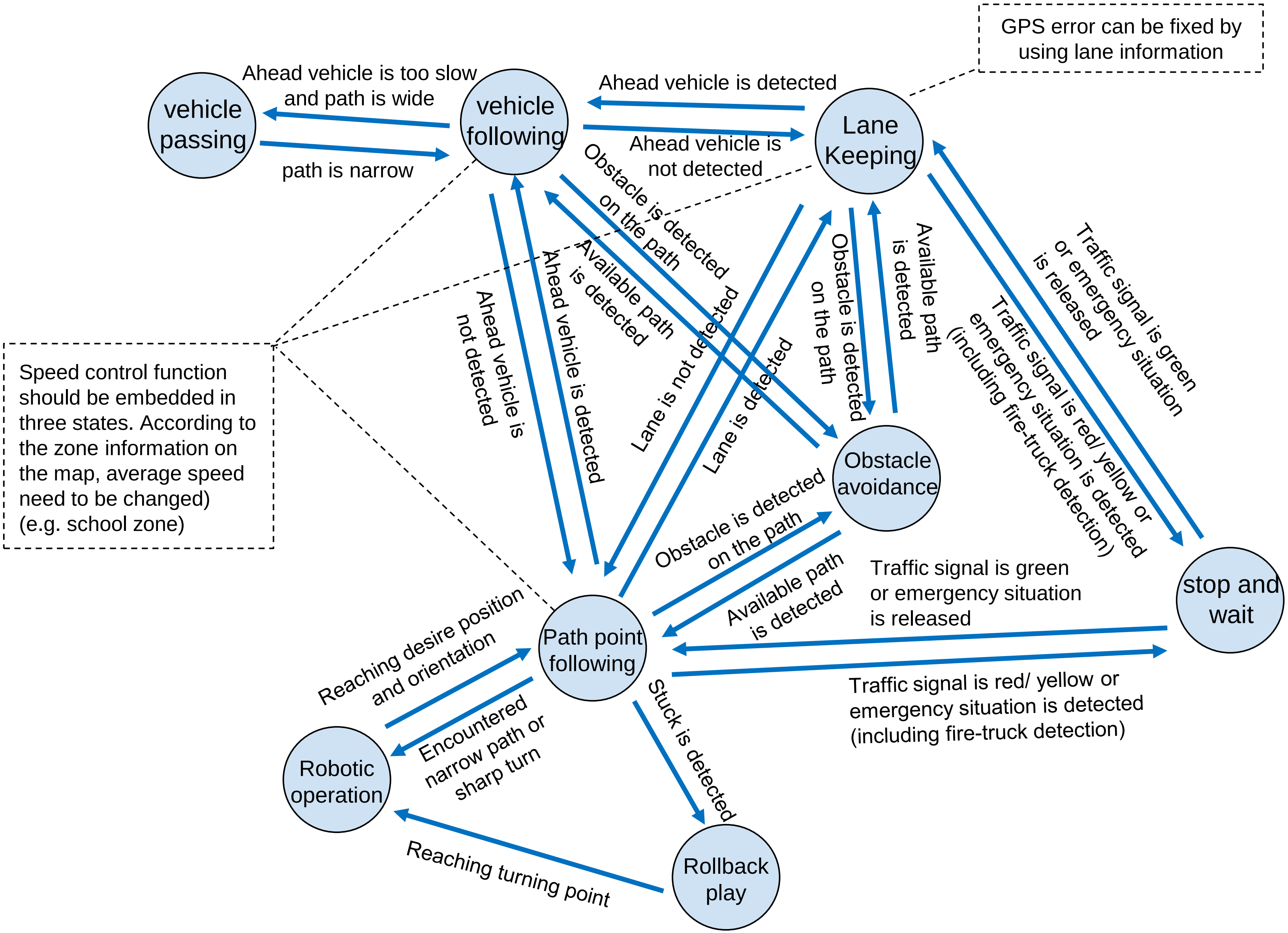


The figure shows the route of the trolley that the 6 autonomous shuttles will be replacing.

The route the shuttle will follow for this specific project is highlighted in purple. The figure also shows the stops, intersections, and crosswalks that have to be considered.

FINITE STATE MACHINE

The Ohio State University
CENTER FOR AUTOMOTIVE RESEARCH



FINITE STATE MACHINE

Vehicle passing: This would be in instances when a car turns hazards on or is blocking the road maybe waiting to pick someone up from the mall. The shuttle vehicle would check to see if there is nearby traffic that would make it unsafe to pass while also considering if the road is wide enough to safely make the maneuver.

Vehicle following: The default state that is active when there is a car in front of the shuttle. The shuttle will keep a safe distance from the car in front of it and stay in lane.

Point-path following: Takes over when the road is undetectable and there is no vehicle ahead to follow. Also will make sure the vehicle is following the preset desired route.

Lane keeping: This mode is active while on the road to make sure the vehicle stays in lane specially when there is no vehicle ahead to follow.

Obstacle avoidance: This state takes precedence whenever there is an obstacle on the road or path. The vehicle will stop and search for safe alternatives to the original route and once a path around the obstacle is detected the state is changed and the car continues on the new path. For example, if the normal route is blocked with construction, but a new path is outlined by cones; the vehicle will recognize this as the new route and follow the new path.

Stop and wait: This state will be important since there are a lot of stops on the route. This state is used for traffic signals, stop signs, pedestrian crosswalks, pedestrian pickup locations, and emergency situations. In the instances the vehicle is to stop for traffic reasons the vehicle is to follow the laws of the road until it is safe to proceed. In the instance of an emergency situation the vehicle needs to stop and pull over in a safe manner to make room for emergency vehicle to pass. For the vehicle pickup locations the vehicle will stop and wait until a number of checks are cleared to signal the passengers are ready to leave.

In the instances the vehicle could be stuck or the normal path is narrow due to some unforeseen circumstances the vehicle will try to rollback and find a position and orientation to get unstuck.

REAL WORLD IMPLEMENTATION CHALLENGES

DEREK LONGSHORE

REAL WORLD IMPLEMENTATION CHALLENGES

Challenges	Risk Level
Snow and Ice road conditions	High
Pedestrians crossing the road in unpredictable ways	HIgh
A car does not follow the traffic laws correctly	High
Interpreting pedestrian actions on crowded days	High
Cyber Security	High
How to manage vehicle issues (flat tire, engine malfunction, etc.)	Medium
Cyclists being unpredictable and using hand signals	Medium
Low visibilty weather (fog or heavy rain)	Medium
Street sign is vandalized or missing	Medium
Construction zones and traffic cones	Medium
Vehicle Accident in path	Medium
Cop directing traffic	Medium
How to manage an unexpected low fuel or low power situation	Low
Too many pedestrians get in vehicle exceeding the safety limit for passengers	Low
Interpreting 4 way stops	Low

CONCLUSION

CONCLUSION

- Based on longitudinal and lateral model, we can control the vehicle appropriately.
  - By controlling the engine and braking torques, we could make the vehicle keep the speed limit in the given path under different road conditions
  - We could recognize the vehicle’s feature such as understeering based on vehicle parameters
  - Linear lateral model is converged much faster than nonlinear lateral model.
- Path can be generated with given partial waypoints
  - Beizer curve is used to generate full path
  - Smoothing between segments of Beizer curve results in better path following.
  - Path following can be improved by applying additional control inputs obtainable from perception and localization systems.
- Different sensors can be applicable for Localization and Perception
  - GPS and IMU data can be merged to get more accurate position data.
  - Image data with the trained NN can be used to recognize environmental traffic conditions.
- Stateflow and decision making
  - Stateflow chart can be generated through Finite State Machine, and it can be directly applicable to the system by using decision making tool in Matlab.
  - It manages driving maneuvers all time as a high level controller

PRESENTATION

- YouTube: https://www.youtube.com/watch?v=MKH4dqSAEdY

REFERENCE

REFERENCES

- ECE 5553_HW1: Design Your Own Autonomous Road Vehicle Architecture and State Diagram
- ECE 5553_HW2: Build and Simulate Your Tire, Longitudinal and Lateral Models
- ECE 5553_HW3: CACC & ACC Simulation and String Stability
- ECE 5553_HW4: Path Following Model
- ECE 5553_HW5: Decision Making and Collision Avoidance
- Tesla model X parameters: https://www.tesla.com/support/model-x-specifications
- Map of test path: https://www.google.com/maps/@40.0527792,-82.9169938,1185m/data=!3m1!1e3
- MATLAB Toolboxes for localization and perception:
  - https://www.mathworks.com/products/neural-network.html
  - https://www.mathworks.com/products/automated-driving.html

ECE 5553 – AUTONOMY IN VEHICLES

DEREK LONGSHORE	LONGSHORE.22@OSU.EDU
EVAN STODDART	STODDART.3@OSU.EDU
SHIHAO LIU	LIU.5401@OSU.EDU
TEAWON HAN	HAN.394@OSU.EDU
YIFAN WU	WU.2828@OSU.EDU

04/23/2018

THE OHIO STATE UNIVERSITY

CENTER FOR AUTOMOTIVE RESEARCH