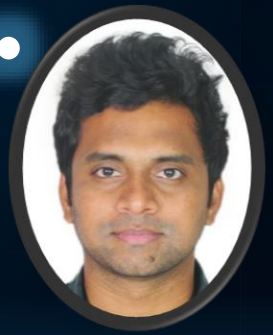
Data & AI Conference

08. - 10. April 2024

Hosted by AI CoE, Friends & Partners

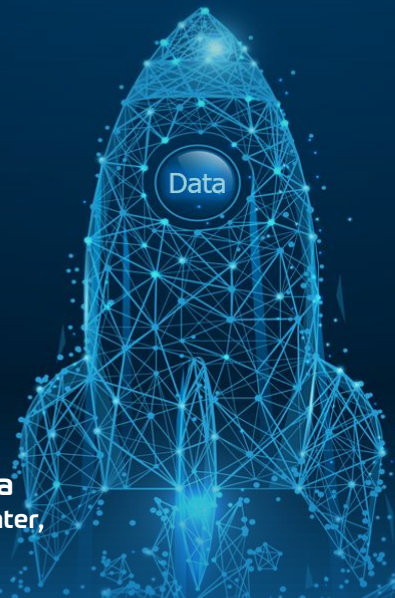
Supervised Learning-based Feed Forward Controller



Sivaraman Sivaraj
Senior Engineer
ZF Tech Center, HYD

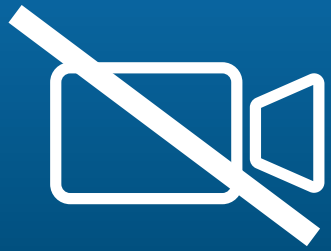
Dr. Michael Fleps-Dezasse
Function Lead
iVEE Innovation VMC, FRD

Dr. Elena Sapozhnikova
Chief AI Engineer, AI Tech Center, FRD



Session Netiquette

Session will be recorded



Please turn off video during presentation

Mute your self

Please post your questions in the chat or hold until the end!

(recording will be stopped)

Agenda

1. Introduction & Motivation
2. Data Generation
3. Supervised Learning Model and Training
4. Results & Concluding Remarks
5. Queries

Motivation :

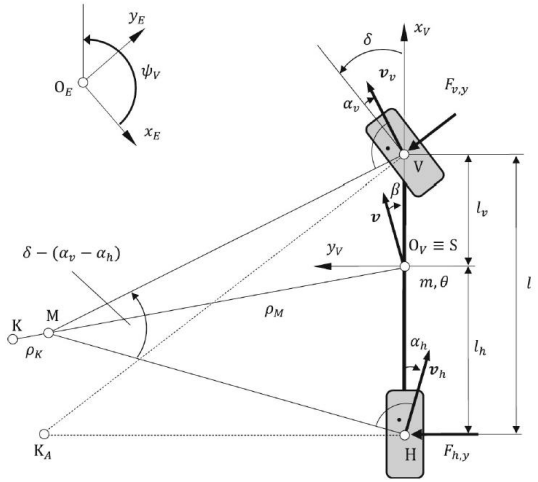
- IVEE (EE Motion under CubiX) project focuses on the development of curvature controller for ground vehicles based on linear single-track model.

- In general, controller composes of two units called feed-forward and feedback controls. Modern control theory-based algorithms such NeuroPID, IRL, MPC, etc., are developed for feedback control and its performances are compared against baseline controller.

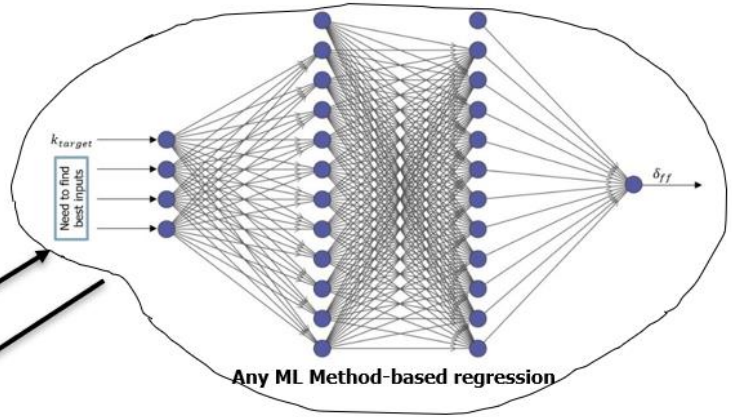
- Linear single-track model uses kinematic feed forward control by default. However, the vehicle's parameters and behaviors are highly sensible to velocity. When velocity increases, smaller change in steering angle could do larger impact in the vehicle position.

- The analytical method uses only the road curvature to computing the target wheel angle meanwhile the single-track model tends to be saturating its wheel angle for higher velocity. In other words, kinematic feed forward control is independent of vehicle's sensible parameter (velocity). The motivation of this work is to create a feed forward model based on data-driven method with accounting vehicle's kinematic parameters (curvature and velocity).

Control Architecture with Replacement :



$$k_{target} = \frac{1}{\rho_M}$$



Feed Forward Control
$\delta_{ff} = \arctan(l * k_{target})$

δ_{ff}

Position Controller

k_{target}

$\delta_{ff} + \delta_{feedback}$

Plant Dynamics

Present States

V_{actual}, k_{actual}

Feedback Control

$\delta_{feedback}$

Overview :

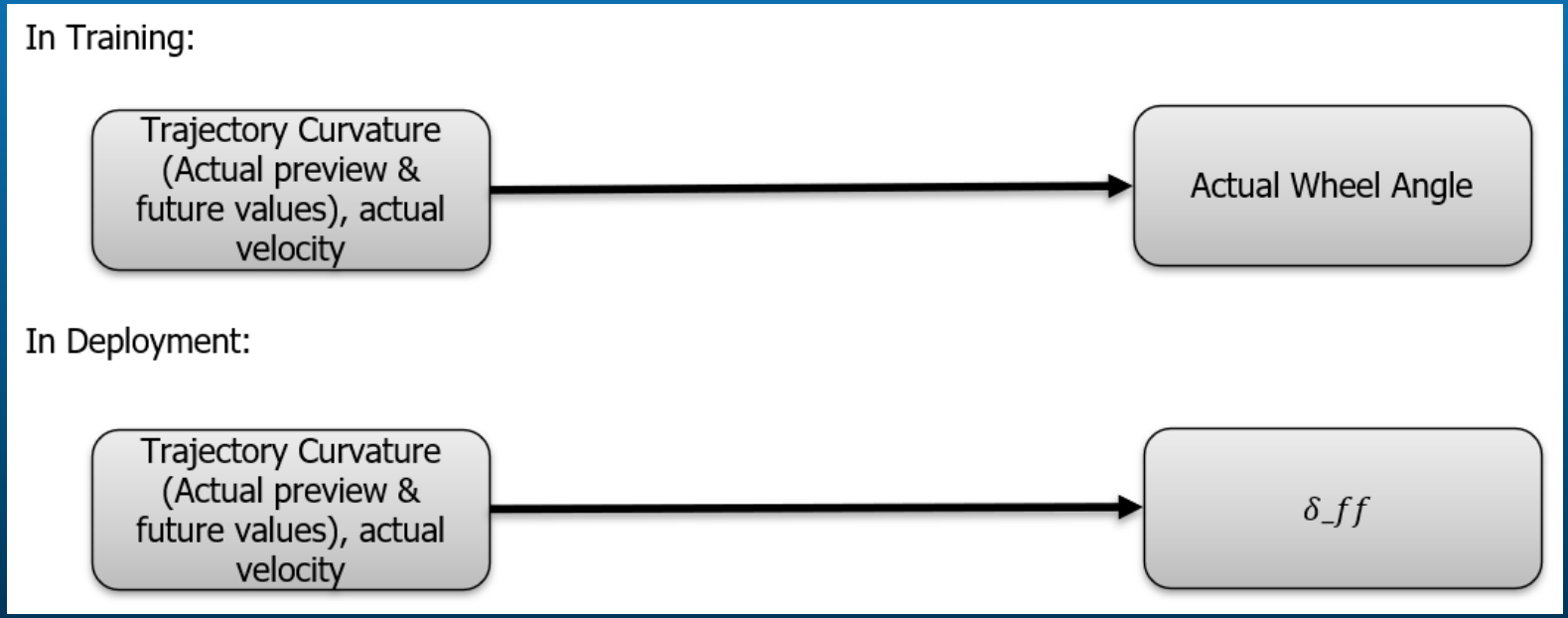
- The feedforward neural network and LSTM network are trained by supervised learning using an FMU vehicle model for data generation in Python.

- On accounting the vehicle velocity in computing the feed forward, the neural network model is developed based vehicle longitudinal velocity, current curvature, past and future curvatures. Based on the correlation between actual wheel angle, the input to the neural networks is selected.

- A Functional Mock-up Unit (FMU) of single-track model and velocity control are used for the data generation. Based on the velocity requirement, velocity control FMU will provide the braking and acceleration torque. Single track model takes target wheel angle, acceleration and braking torques value from the velocity control FMU.

- The data are generated based on brute force method of all combination of wheel angle and velocity and based on random wheel angle and random velocity profile. The generated data are preprocessed and trained. In addition to this, scenario data from the pipeline assessment also added. The trained pytorch model transferred to Simulink and the performance of the feedforward control unit is compared with baseline controller.

Development Strategy:



Therefore, the neural network unit is trained such that it predicts the wheel angle based on road curvature and vehicle longitudinal velocity. The neural network unit technically represents the inverse vehicle dynamics behavior. The net is an alternate one for analytical method of feed forward control.

Motivation in Data Generation

- Foremost aim in the data generation work is to capture the all-possible combination of states and expand the state space closer to the horizon.

- There are Two ways to gather the data for training
  - Synthetic data from computer simulation (car maker simulation, log data from pipeline assessment)
  - Experimental Data

- Experimental data are subjected to limited number of tracks. Covering up whole state space with experimental data are challenging one. So, we started to give much attention to synthetic data at this moment.

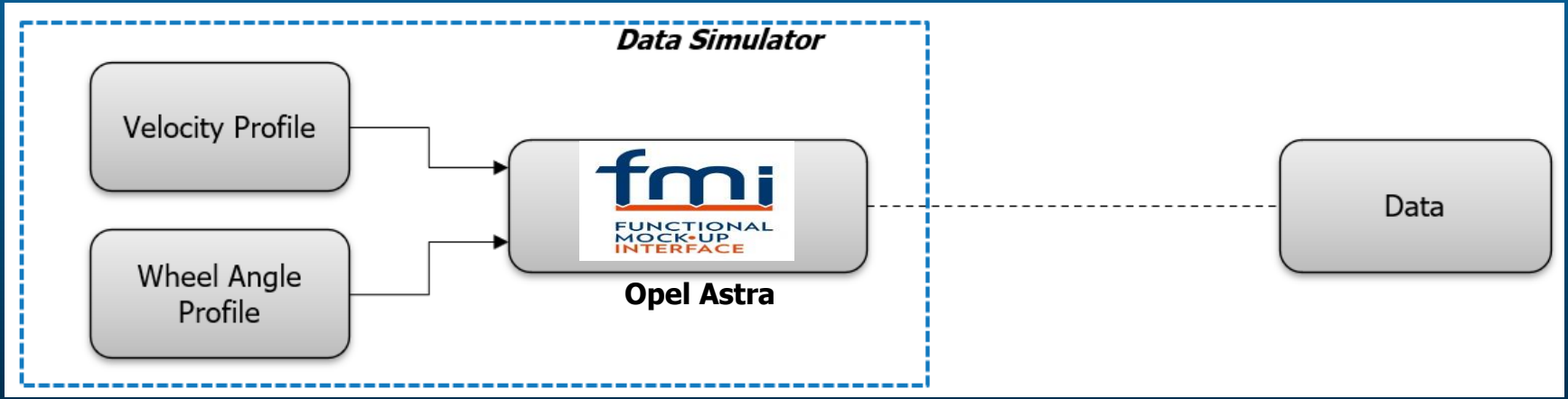
- Since the aim is to improving the existing kinematic model, the variables such as acceleration, curvature rate , side slip angle and side slip angle rate can also be accounted to developing the model. However, these parameters are sensitive to the noise. So, acceleration and curvature rate variables are not chosen for model input. But, these variables are good metric to observe the data spread over.

- Side slip angle can be used as input to the model which is planned as our future work.

Data Generation – Using FMU :

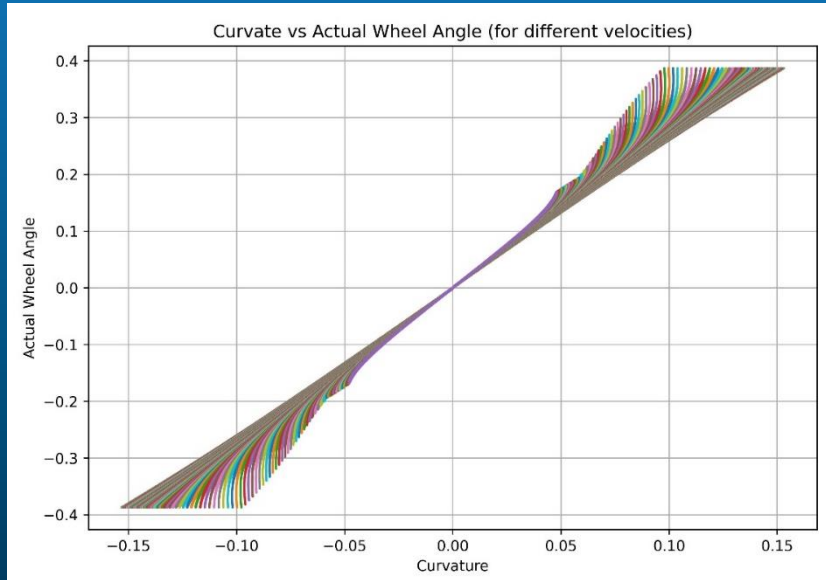
- Brut Force Method
  - Keeping the constant Velocity & Varying Wheel Angle
  - Sine signal on wheel angle constant & velocity

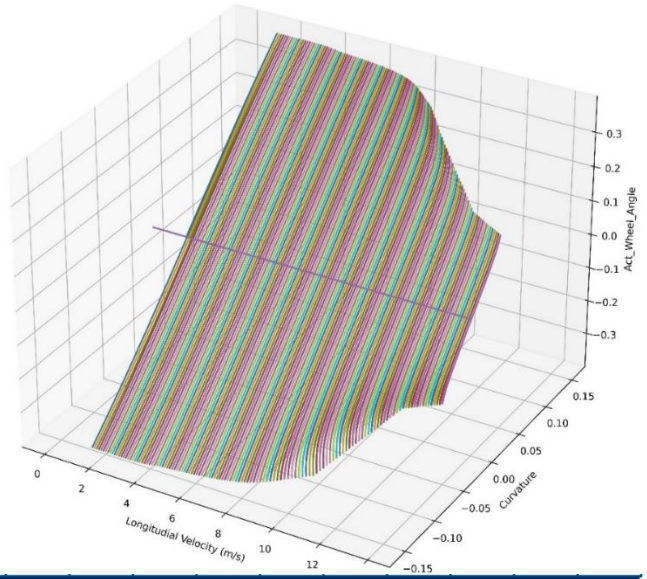
- Scenario Data



It is noticeable that velocity range is restricted from $1.5\ m/s$ to $14\ m/s$ and wheel angle is varying from $-22.9°$ to $22.9°$

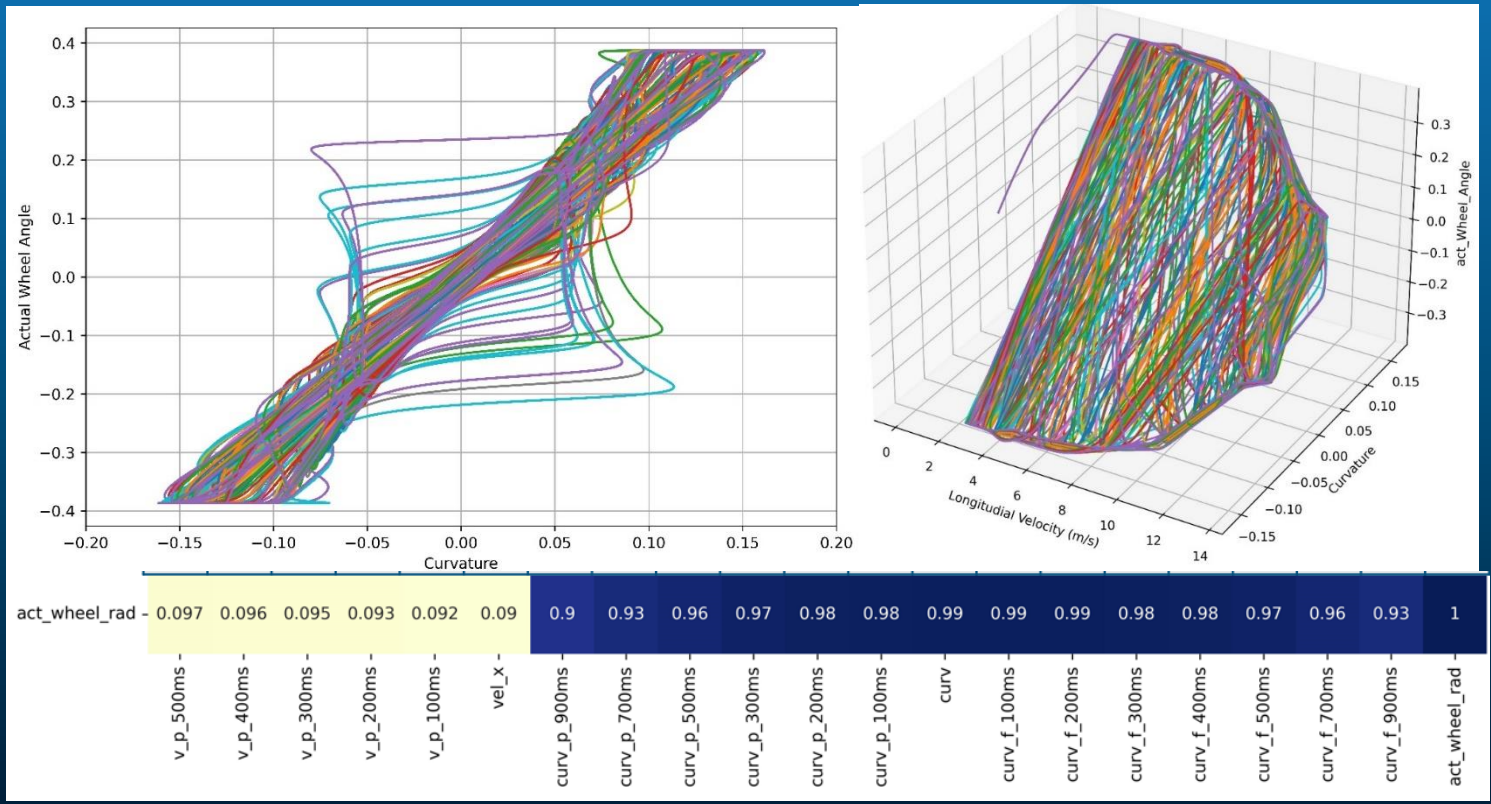
Brut Force Data : (constant velocity & varying wheel Angle)



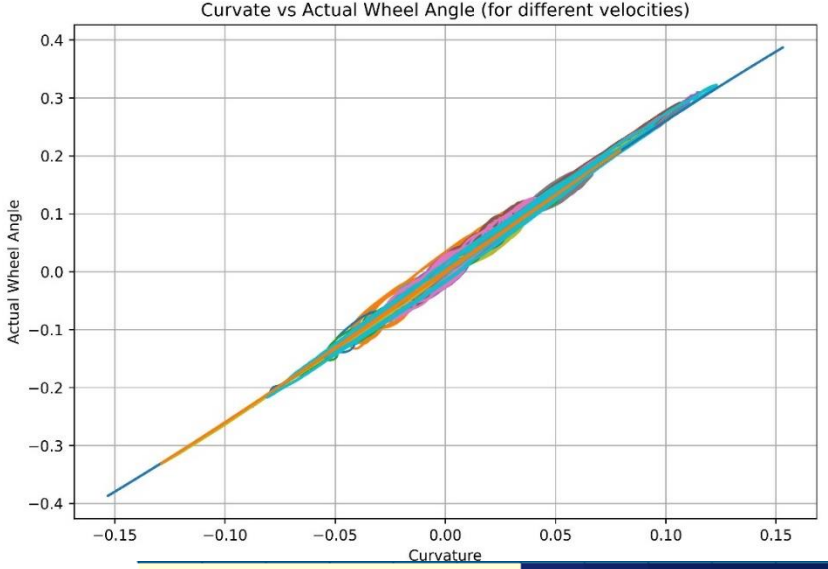


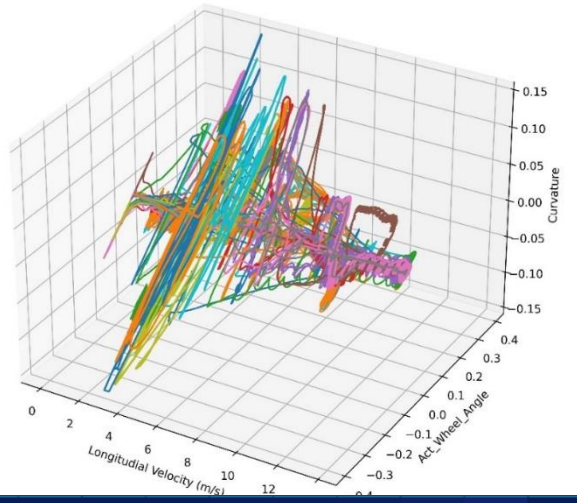
	v_p_500ms	v_p_400ms	v_p_300ms	v_p_200ms	v_p_100ms	vel_x	curv_p_900ms	curv_p_700ms	curv_p_500ms	curv_p_300ms	curv_p_200ms	curv_p_100ms	curv	curv_f_100ms	curv_f_200ms	curv_f_300ms	curv_f_400ms	curv_f_500ms	curv_f_700ms	curv_f_900ms	act_wheel_rad
act_wheel_rad	-1.9e-05	1.9e-05	1.8e-05	1.8e-05	1.8e-05	1.8e-05	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1

Sine Wave Signal's Data Generation : (Velocity & Wheel Angle)



Scenario Data:

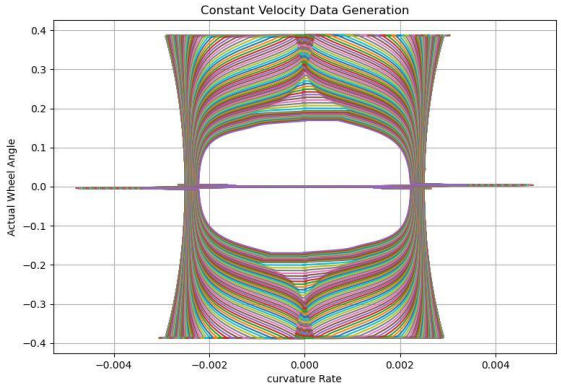


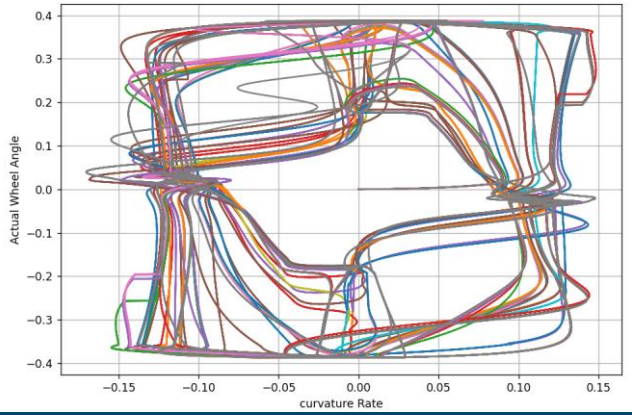


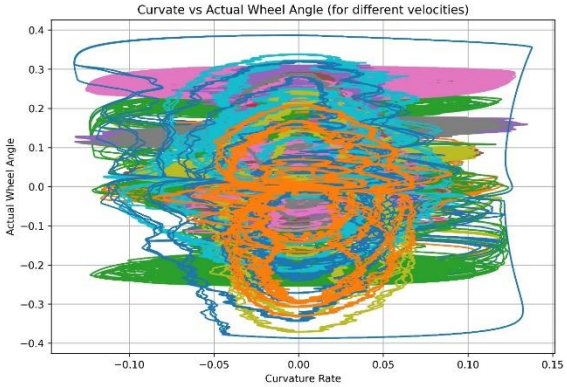
	v_p_500ms	v_p_400ms	v_p_300ms	v_p_200ms	v_p_100ms	vel_x	curv_p_900ms	curv_p_700ms	curv_p_500ms	curv_p_300ms	curv_p_200ms	curv_p_100ms	curv	curv_f_100ms	curv_f_200ms	curv_f_300ms	curv_f_400ms	curv_f_500ms	curv_f_700ms	curv_f_900ms	act_wheel_rad
act_wheel_rad	-0.026	-0.027	-0.027	-0.027	-0.027	-0.027	0.95	0.97	0.98	0.99	0.99	1	1	1	1	0.99	0.99	0.98	0.97	0.96	1

Curvature Rate Observation:

Brut Force Data	Sine Signal Data	Scenario Data







- Constant velocity has dense curvature rate spread, but with narrow range
- Sine wave signal-based dataset has spread from lower to maximum with lesser dense
- Scenario data has good spread and dense in the state space.

Data Inspection:

	v_p_500ms	v_p_400ms	v_p_300ms	v_p_200ms	v_p_100ms	vel_x	curv_p_900ms	curv_p_700ms	curv_p_500ms	curv_p_300ms	curv_p_200ms	curv_p_100ms	curv	curv_f_100ms	curv_f_200ms	curv_f_300ms	curv_f_400ms	curv_f_500ms	curv_f_700ms	curv_f_900ms	act_wheel_rad
kendall	0.019	0.019	0.018	0.018	0.017	0.017	0.836	0.862	0.887	0.906	0.917	0.93	0.943	0.947	0.94	0.929	0.917	0.906	0.885	0.861	1
Spearman	0.03	0.03	0.029	0.029	0.028	0.028	0.942	0.96	0.974	0.984	0.988	0.991	0.993	0.994	0.993	0.991	0.988	0.984	0.974	0.959	1
Pearson	0.033	0.033	0.032	0.032	0.031	0.03	0.939	0.957	0.971	0.981	0.985	0.988	0.99	0.99	0.99	0.988	0.985	0.981	0.971	0.957	1
Covariance	0.033	0.033	0.032	0.032	0.031	0.03	0.94	0.96	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.97	0.96	1

Since Covariance correlation is linear one, other non-linear correlations are computed and observed before selecting the suitable input to the neural networks. For training, the data created by different methods are combined. From the table, future curvatures (after 100ms and 200ms) are having higher correlation that the present curvature. Similarly, past velocities are also higher correlation than the current velocity.

NN Architecture

$v_{p_{200}}$ → velocity 200ms past

$v_{p_{100}}$ → velocity 100ms past

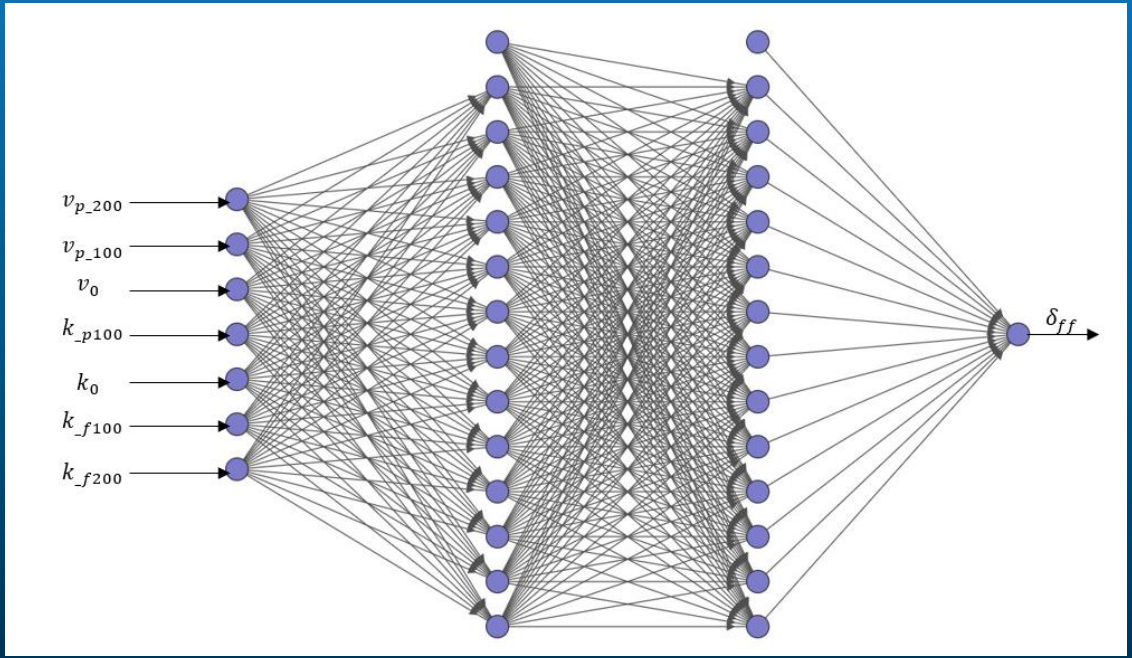
v_0 → velocity present

$k_{p_{100}}$ → curvature 100ms past

k_0 → curvature present

$k_{f_{100}}$ → curvature 100ms future

$k_{f_{200}}$ → curvature 100ms future



The data are $Gauss$ normalized. The neural network has two hidden with 512 neurons. tanh activation function is used in the hidden layers. Input and output layers are kept free from activation function.

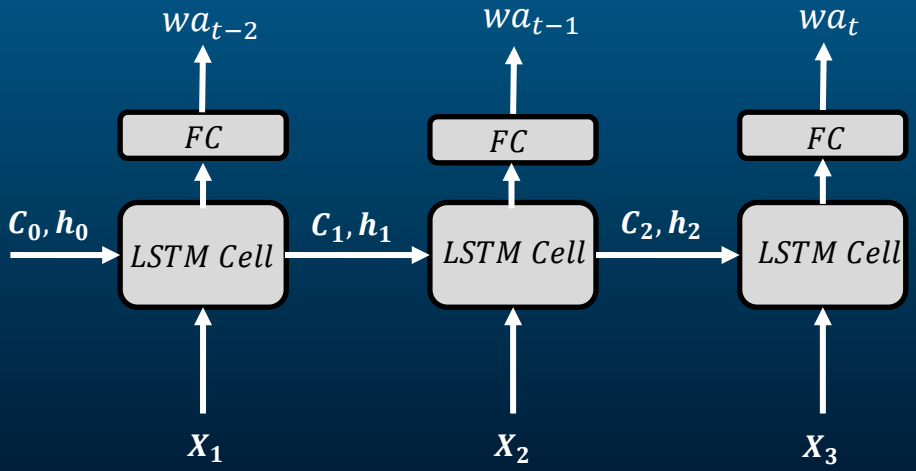
LSTM Model:

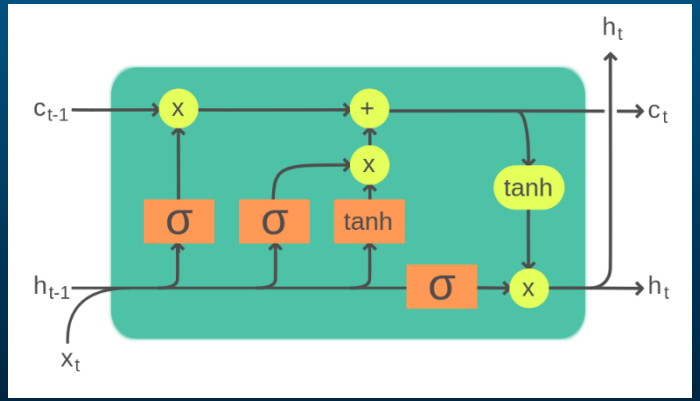
$$X_1 = [k_{p100}, k_0, k_{f100}, k_{f200}, v_{p200}, v_{p100}, v_0]_{t-2}$$
$$X_2 = [k_{p100}, k_0, k_{f100}, k_{f200}, v_{p200}, v_{p100}, v_0]_{t-1}$$
$$X_3 = [k_{p100}, k_0, k_{f100}, k_{f200}, v_{p200}, v_{p100}, v_0]_{t}$$

$$\begin{bmatrix} wa_{t-2} \\ wa_{t-1} \\ wa_t \end{bmatrix} := \begin{bmatrix} \delta_{t-2} \\ \delta_{t-1} \\ \delta_t \end{bmatrix} \rightarrow \delta_{ff}$$

Required Feed Forward Signal

512 hidden nodes are used in the LSTM Cell with linear activation function in the output layer.

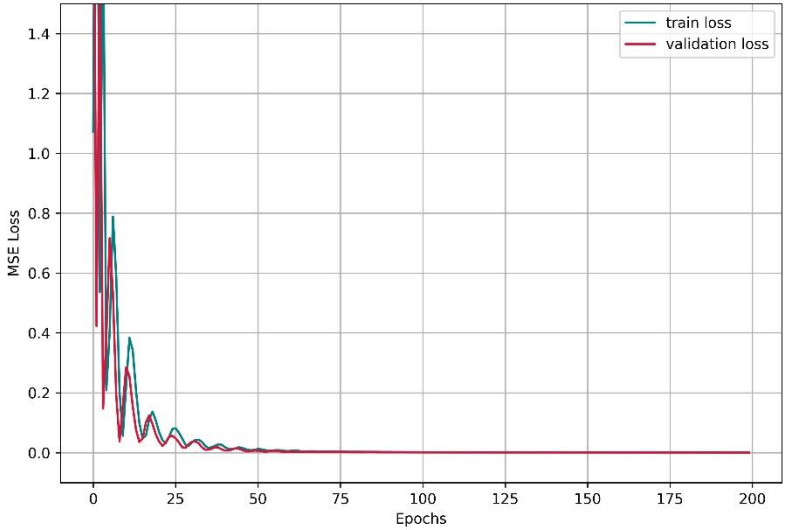


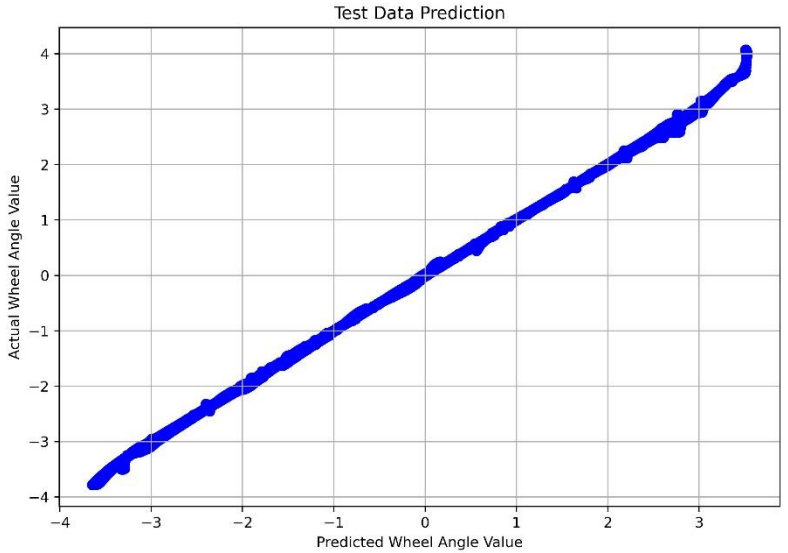


LSTM Cell Architecture
Source: Wikipedia

MLP Model's Train – Validation Loss & Test Data Prediction :

Data are split into 70 – 20 – 10 % for training, validation and testing

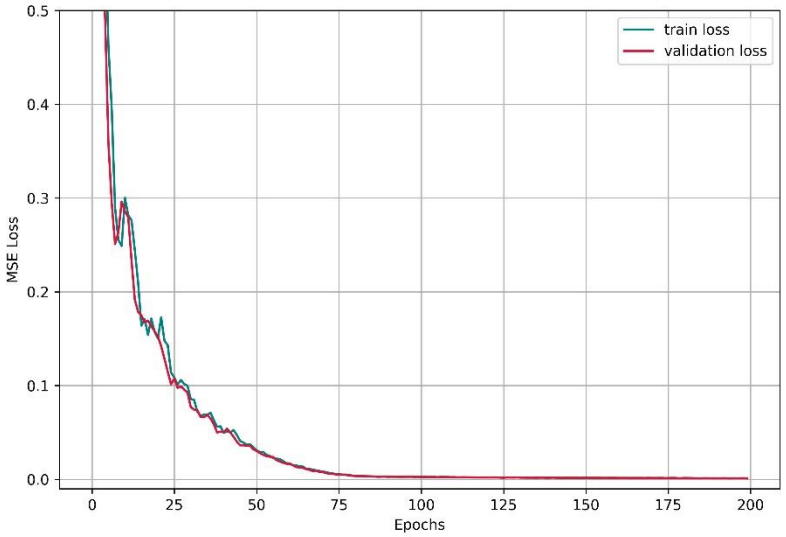


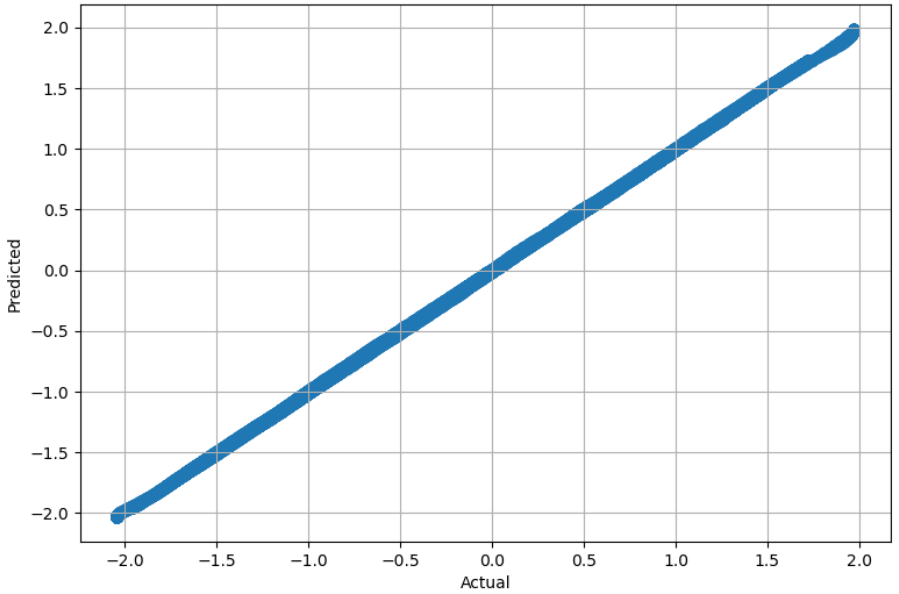


- Training Data Points : (4,692,061 * 8)
- Validation Data Points : (1,340,589 * 8)
- Test Data Points : (670,294 * 8)

LSTM Model's Train-Validation Loss & Test Data Comparison:

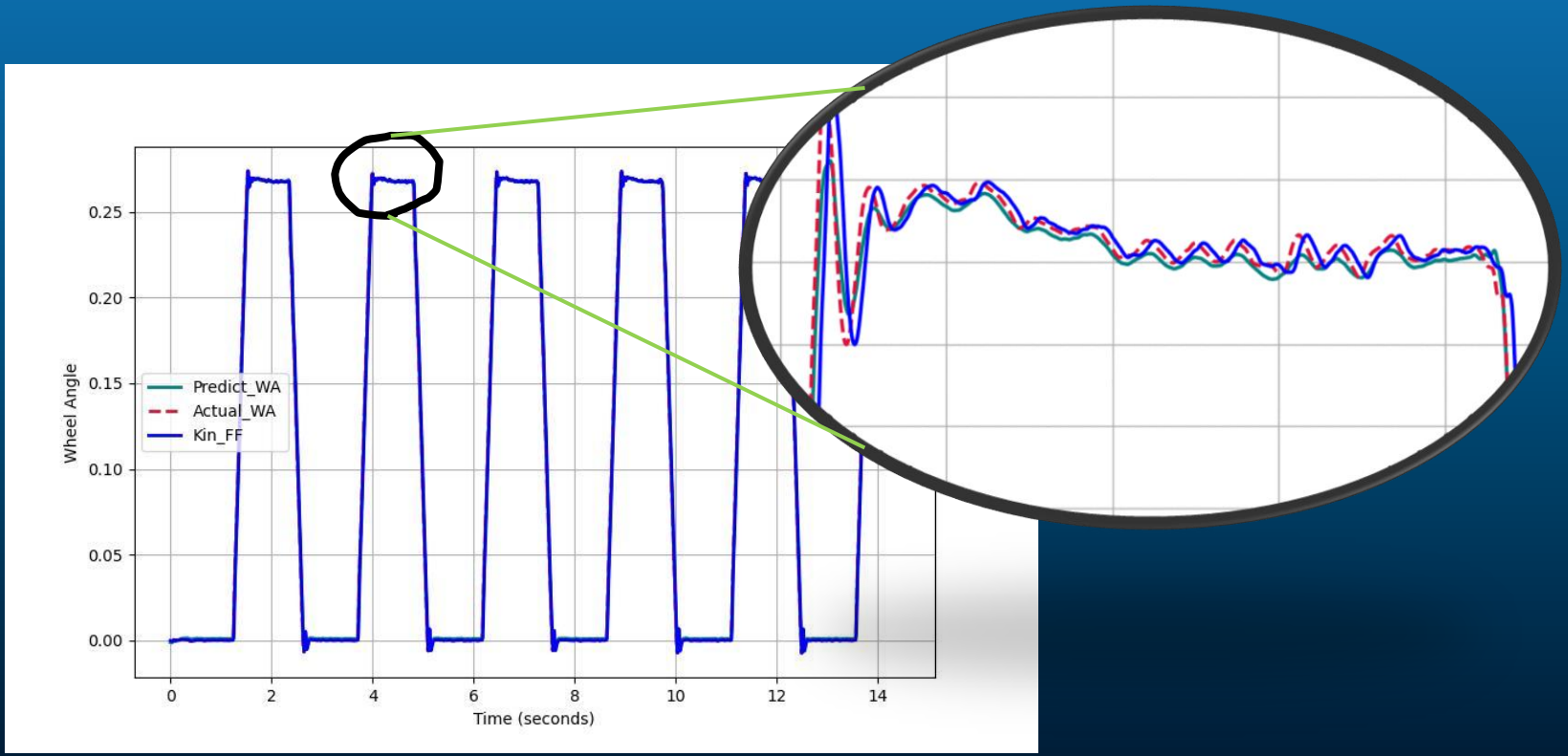
Data are split into 70 – 20 – 10 % for training, validation and testing



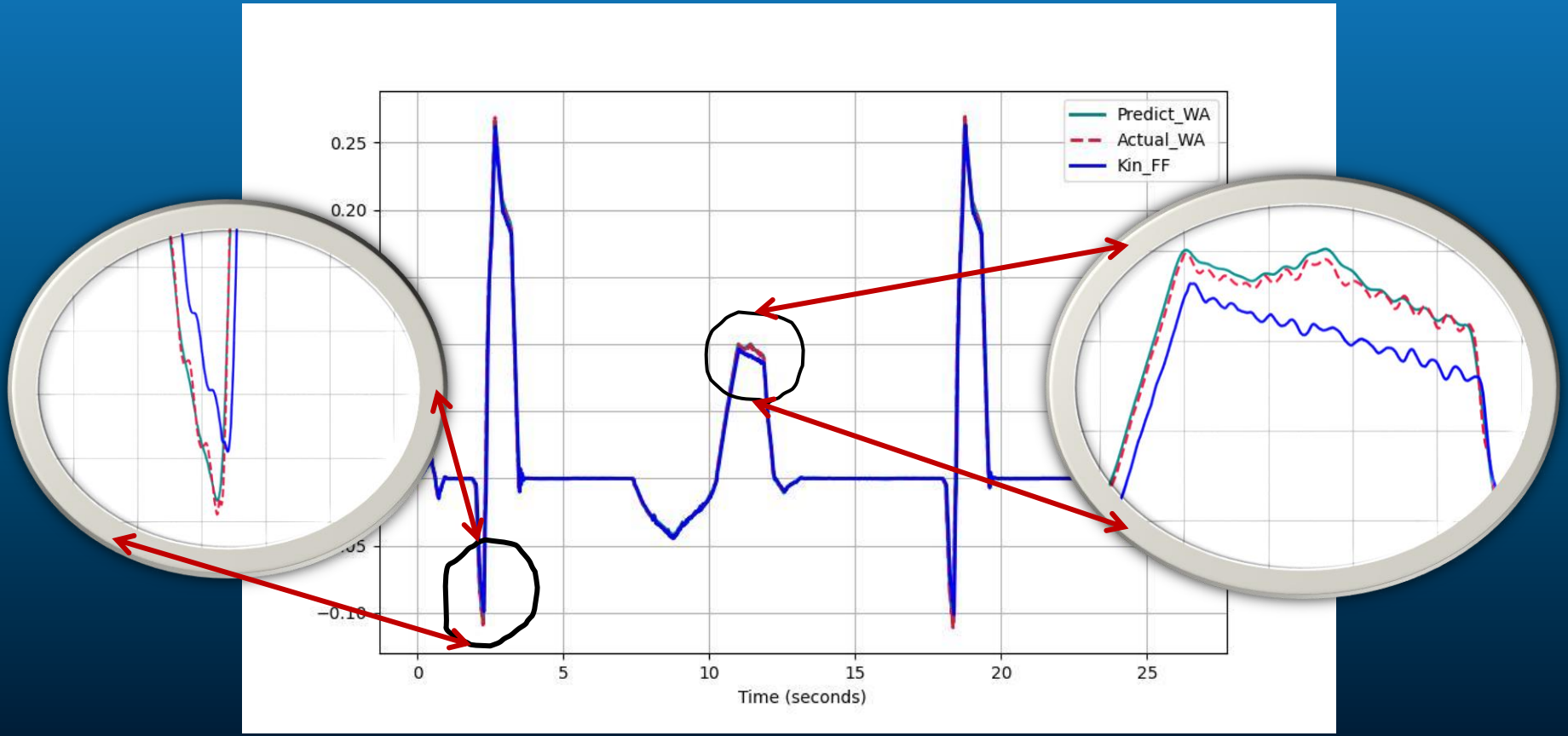


- Training Data Points : (4,692,061 * (4*8))
- Validation Data Points : (1,340,589 * (4*8))
- Test Data Points : (670,294 * (4*8))

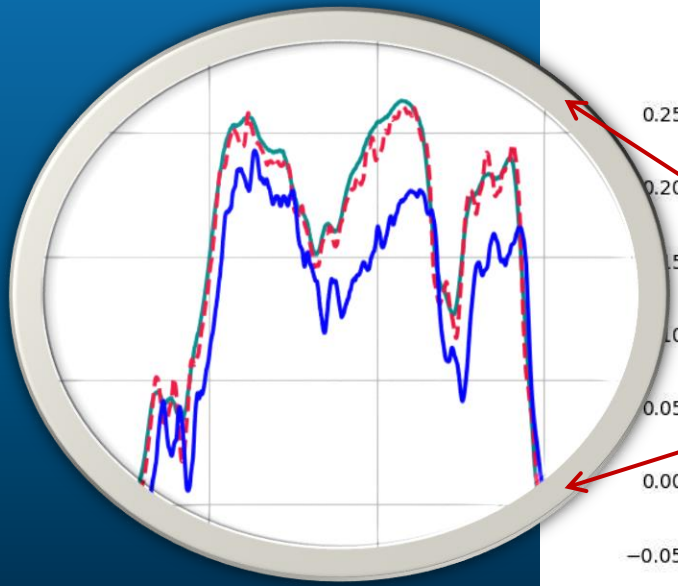
Testing Scenario in Python – uturn r10m 10kph s n

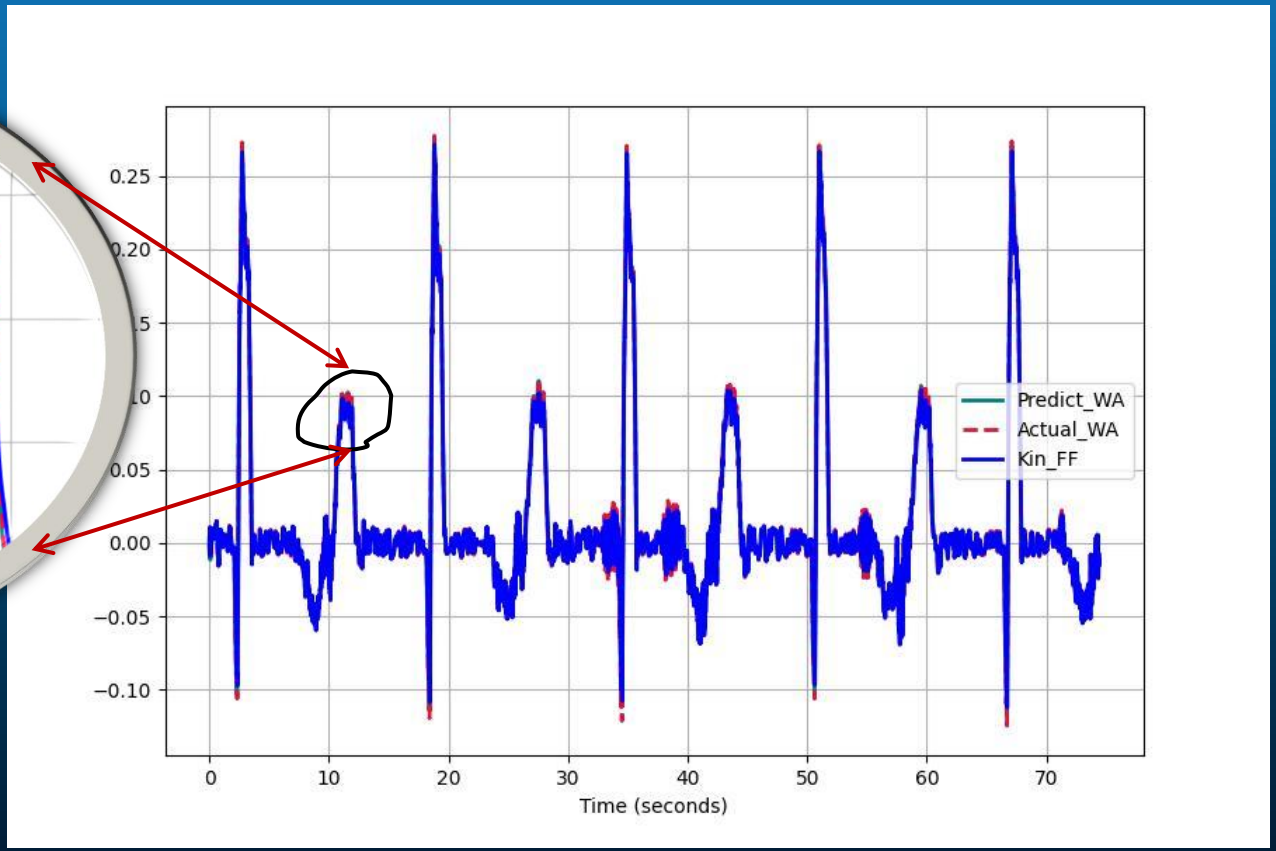


Testing Scenario in Python – fasttrack plant2 v1 s n

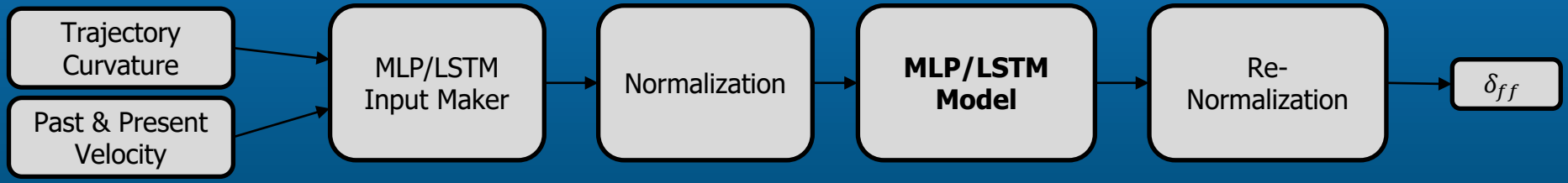


Testing Scenario in Python - fasttrack plant2 v1 s w



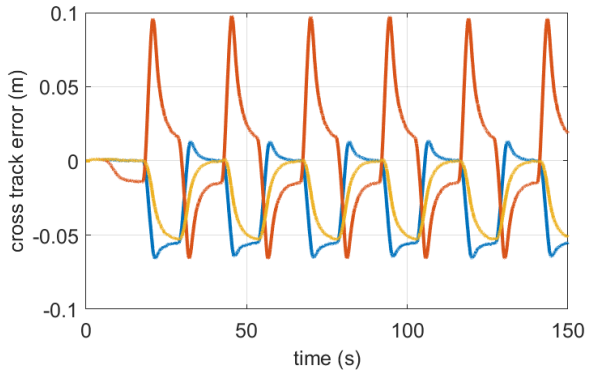


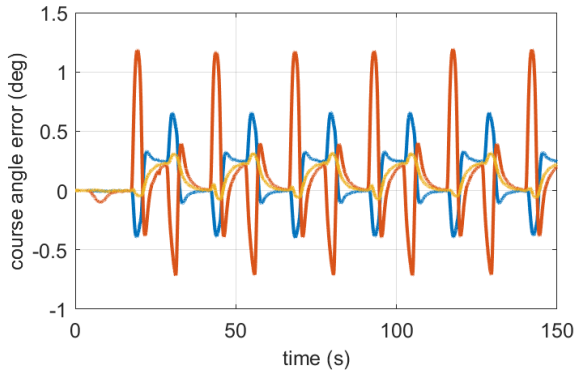
Transfer to Simulink Model :

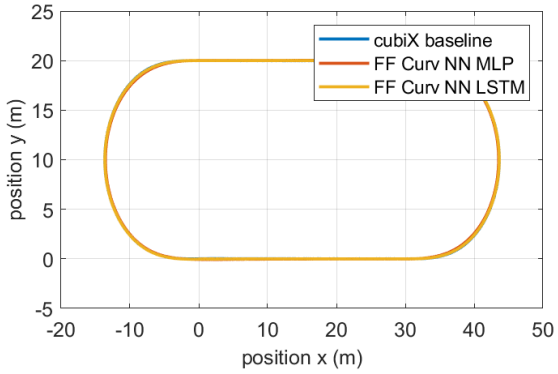


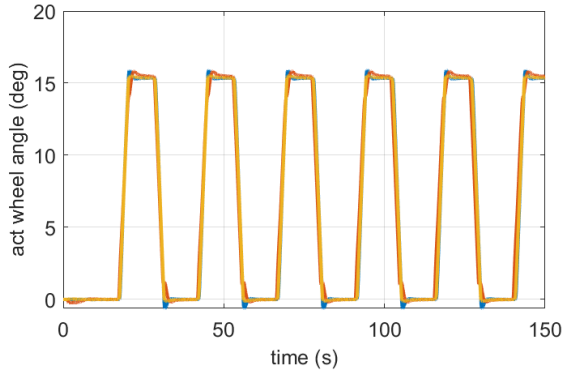
- The trained pytorch model's weights are saved into csv file and its gauss normalized values are used for normalize the input signals.
- Memory buffers are used to store the past information such as velocity.

Result : Scenario 01

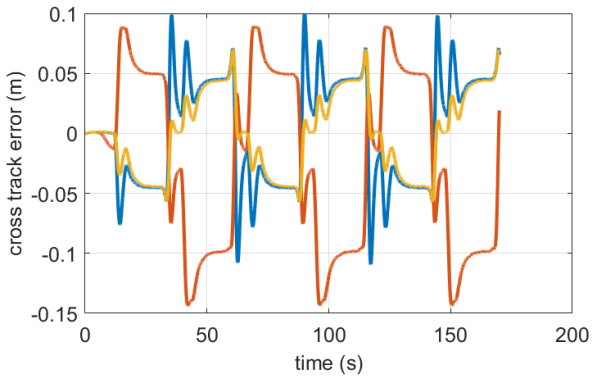


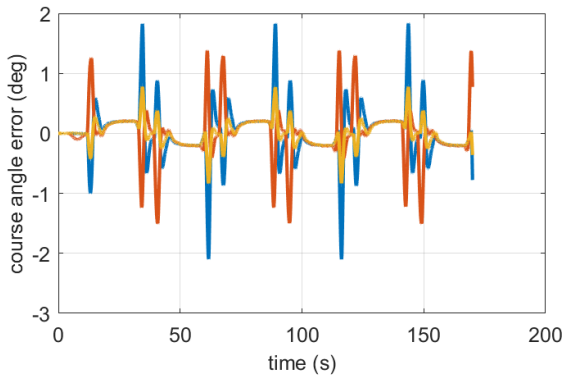


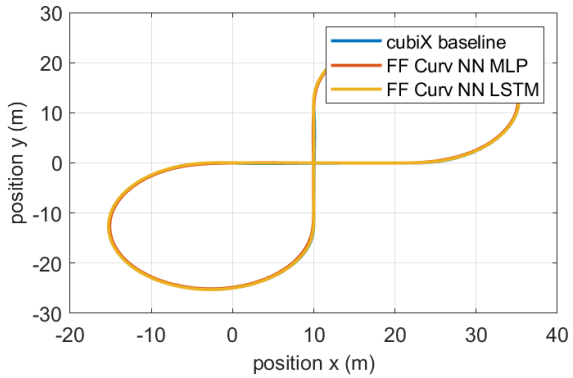


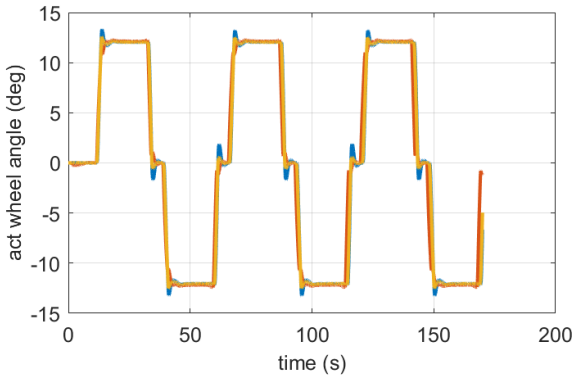


Result : Scenario 02

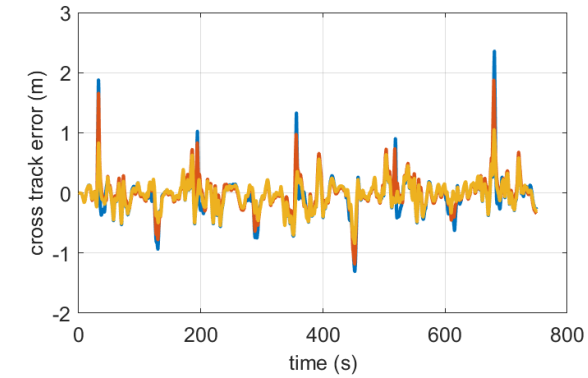


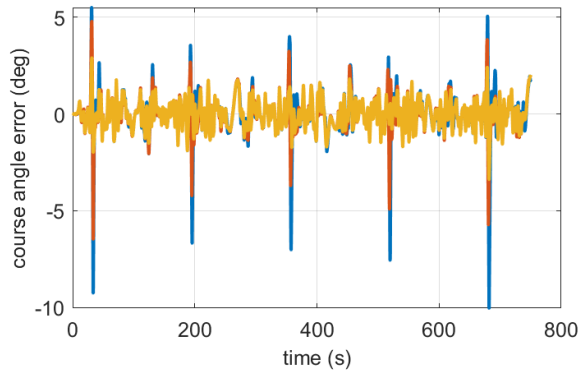


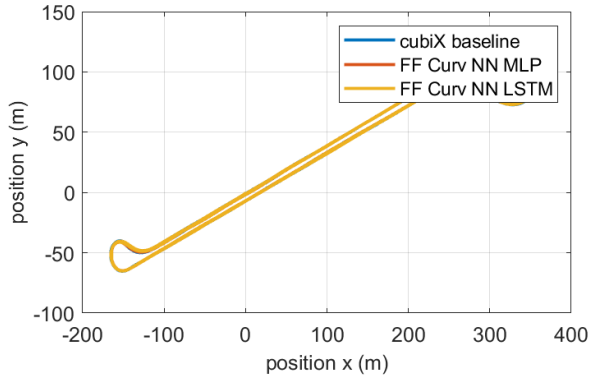


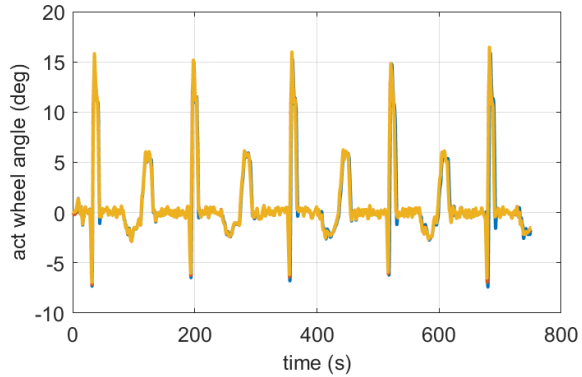


Result : Scenario 03









Results Comparison in Pipeline Assessment

Testing Scenario	Kinematic_Model	MLP_model	LSTM_model
uturn r10m 10kph s n	0.066	0.098	0.053
fasttrack plant2 v1 s n	1.093	0.922	0.422
fasttrack plant2 v2 s n	0.436	0.309	0.15
circle r10m 10kph s n	0.054	0.104	0.053
circle r20m 20kph s n	0.127	0.066	0.054
circle r50m 30kph s n	0.24	0.17	0.066
fasttrack plant2 v1 r n	1.428	1.226	0.369
infty 10_15kph r n	0.109	0.144	0.07
fasttrack plant2 v1 s w	1.343	1.903	0.891
circle r50m 30kph r w	0.65	0.581	0.478

Pipeline assessment has totally 33 different scenarios and developed models are compared against baseline controller with all scenarios. The mean of all scenario's error are computed as below.

Mean of Errors :	0.685	0.621	0.341

Conclusion Remarks:

- Motivation of replacing kinematic feed forward model by AI-based model is successfully carried out.
- Data are generated by Functional Mock-up Unit (FMU) by different set input signals and its correlations are computed.
- Based on the correlation's component, suitable input signals are selected and MLP and LSTM models are trained.
- Trained model are transfer to Simulink and compared against baseline controller.
- Finally, AI-based model is performing better than baseline controller (kinematic feed forward model)

Thanks….!
&
Questions Please…!