A Methodology for Development of Look Ahead Based Energy Management System Using Traffic In Loop Simulation

A Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Avinash Vallur Rajendran, B.Tech.

Graduate Program in Mechanical Engineering

The Ohio State University

2018

Master's Examination Committee:

Prof. Giorgio Rizzoni, Advisor

Prof. Marcello Canova

Dr.Qadeer Ahmed

Abstract

This thesis details efforts towards developing a methodology that enables the design of a look ahead based energy management system. It explores various technologies that are required to enable such a system to function on a physical vehicle. A new simulation framework known as 'Traffic-In-Loop' (TIL) simulation is developed to mimic real-world driving. It serves as a drive cycle independent controls development platform. The framework is enabled by combining microscopic traffic simulation with a detailed mathematical powertrain model. The TIL simulation technique facilitates emulation of on-board sensors, V2X communication and capture causal behavior of real-world scenarios. Data collected from these virtual sensors are used to forecast future drive scenarios – called 'Look ahead predictions'. Further a strategy to integrate future drive scenario forecasts with powertrain control is introduced. The above advances, catalyzed the design of a look ahead based energy management controller, called 'Delta Energy Controller'. It aims at improving a vehicle's fuel economy by utilizing available drive scenario forecasts. Simulation results are used to prove the optimality of this controller and study the improvement in fuel economy as a function of better look ahead predictions.

Dedicated to the future of sustainable transportation

Acknowledgments

I thank my advisor, Prof.Giorgio Rizzoni for having faith in my abilities and awarding me with an opportunity to work with his team on this research project. Inputs from Dr.Qadeer Ahmed and Bharathkumar Hegde have greatly helped me over the course of this work. I would also like to thank Prof. Marcello Canova for coaching me on simulation and modeling techniques. His teaching equipped me with required skill to complete this thesis.

I am grateful to have been blessed with great friends and staff at the Center for Automotive Research. I highly appreciate their unconditional support in solving any and all administrative difficulties that came my way. The relationships that I formed at this center have molded me to the person I am today. Finally, I must express my profound gratitude to my many friends for celebrating my successes and sharing my lows. They provided me with continuous encouragement and unfailing support throughout my years of study.

Vita

2009-2013	B.Tech. Mechanical Engineering National Institute of Technology Tiruchirappalli, India
2013-2015	Assistant Manager, Tata Motors Engineering Research Center Pune, India
2015-present	Graduate Research Associate, Center for Automotive Research The Ohio State University Columbus, Ohio

Publications

Research Publications

A. V. Rajendran, B. Hegde, Q. Ahmed and G. Rizzoni "Design and development of traffic-in-loop powertrain simulation" *IEEE Conference on Control Technology and Applications*,Aug. 2017.

A. V. Rajendran, Y. Khairate and C. Awate "Prescriptive Modeling, Simulation and Performance Analysis of Mild Hybrid Vehicle and Component Optimization". *Symposium on International Automotive Technology* ,Jan. 2015.

Fields of Study

Major Field: Mechanical Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Mankind was made to move. Over the past century humans have realized that mobility enables progress and equated mobility to power. However, environmental and sustainability concerns have damped this enthusiasm in recent years. Hybrid electric vehicles have been viewed as stepping stones in the path towards a greener and more sustainable future for transportation. Hybrid vehicles are those vehicles that have more than one energy sources that complement each other. Typically one of the energy source is a conventional fuel -in liquid or gaseous form and the other is rechargeable energy storage system. The former may be electrochemical, hydraulic or mechanical devices. Since energy required to propel the vehicle comes from two different types of energy sources, hybrid vehicles often need to use two types of prime movers to convert the stored energy into propulsive power. Batteries are a form of electrochemical energy storage systems and are most common in today's automotive powertrains. Such vehicles that use a combination of battery and fuel power are termed as 'Hybrid *Electric* Vehicles' (HEV). In a HEV, an electric motor is used to enable this conversion. Further, Internal combustion engines are used to convert chemical energy from conventional fuels to propulsive power. The presence of two prime moves allows a vehicle's internal computer (called controller) to optimally choose and split the total required propulsive power between the two sources. This split is done with the aim of maximizing overall system efficiency subjected to certain performance constrains. The nature and amount of split achievable is determined by the vehicle's powertrain architecture and its components sizes. This thesis, deals with a specific architecture known as 'Range Extended Electric Vehicle ' (REEV) as seen in Chapter 4.

As we create vehicles that are more fuel efficient, we rarely pay attention to factors external to the vehicle. These 'environmental factors'extend beyond the powertrain and often encompass the broader environment outside the vehicle. A vehicle's fuel economy extends beyond its components and is influenced by these external factors [10, 11, 12, 13, 14]. This includes the effect of a vehicle's interactions with traffic, infrastructure and variation in human-drivers. Studies show that lab based testing for fuel consumption fails to capture variations of upto 35% [15, 16, 17, 18, 19]. Hybrid electric vehicles are inherently more sensitive to such external variation and has been explored in detail by various entities [20, 21].

With powertrain components today nearing the peak of their respective efficiencies limits, considerable focus has been shifted towards understanding the effects of factors external to the powertrain that affect performance and fuel economy of the system. Several existing literature [14, 22, 23, 24] cite driver behavior for approximately 35% of variation in fuel economy. Other major factors that contribute towards variation of on-road economy include, effects of real-world traffic conditions and infrastructure. However by experience, we know that a human driver's behavior in real life scenarios is influenced by the behavior of other drivers on the road and presence of infrastructure elements like traffic lights or stop signs. Hence these 3 major factors - 1.External Factors, 2.Driver Behavior and 3.Traffic can be considered as being connected at some level. Road grade, accessory loads, ambient conditions, maintenance, trip duration and vehicle utilization (eg. driving with windows rolled down) also cause variation in on road economy from lab tested values. All of these above factors that directly or indirectly affect fuel economy of a vehicle are considered under the broad umbrella of 'Environmental Factors'.

Look Ahead Energy Management is a powertrain operation strategy that tries to capture the environmental factors a-priori in an attempt to optimize powertrain operation. Typically, a look ahead energy management strategy has 2 components a predictor and a control optimizer. The predictor aids in collecting information from an array of data sources and fusing them to create a velocity and power forecast for a given trip. The control optimizer is an adaptive algorithm that aims to maximize the powertrain's energy efficiency. The range, quality and reliability of the look ahead predictions depend on the sensors and algorithms being used. Chapter 2 covers in detail, the various sensors that can be used to enable look ahead predictions.

In the absence of a physical vehicle to develop and test the LEMS platform, a virtual test bed needed to be designed that captures external factors like traffic, infrastructure and related interactions. Additionally, the platform also needed to account for non-causality of real world systems. That is, an input at time 't' would affect future input at time 't+n'. Since most techniques used today in the domain of powertrain modeling are not capable of accounting for non-causal systems, its effects often go uncaptured. Hence a tool called 'Traffic In Loop' simulation was developed such that it accounts for causal behavior. Traffic In Loop simulation uses Simulink for mathematical modeling of the powertrain and vehicle dynamics. Parallelly, a traffic simulator called SUMO, is used to create a virtual test environment for the vehicle. Chapter 3 discusses this process in fine detail.

Look ahead predictions are made using sensor data as emulated in the TIL framework. A methodology to use such forecasts for powertrain control is discussed with the example of controller called ‘Delta Energy Control’. This control strategy is designed for Range Extended Electric Vehicles. The controller predicts the quantum of energy that needs to be added -to ensure that the terminal battery State Of Charge (SOC) is within a reasonable bounds of the desired trip-end SOC. This computation is done using available look ahead data and current SOC information. The algorithm is self executing at beginning of each trip and before every EGU state change. The overall goal of this algorithm is to add required energy with minimal engine crank events. It has been demonstrated through TIL simulations that algorithm delivers improved economy as the quality of look ahead data improves. The improvement is quantified by increase in fuel economy and reduction in number of engine cranks. The results of this activity are illustrated in Chapter 4.

1.1 Definition of Common Terms

- **Host Vehicle** or **Host :** Refers to a vehicle of interest in the traffic simulation environment(SUMO is used in this thesis), whose powertrain is being co-simulated by a specialized model (powertrain model is build in Simulink for this thesis). SUMO emulates the host vehicle’s interaction with the transportation system and other vehicles on the road. This vehicle’s sensor readings are accessible to the user and its behavior in the traffic simulation environment is controlled by the powertrain model. User defined control actions can be performed on this vehicle through Simulink.

- **Leader Vehicle** or **Leader:** Refers to a vehicle thats immediately ahead of the host vehicle, in the same lane.

- **Look Ahead Prediction** or **Look Ahead Forecast:** Predicted trajectories of vehicle velocity and wheel power, as functions of distance or time. These predictions are calculated utilizing data received from on-board sensors and/or external sources.

- **Look ahead horizon:** The maximum distance or time window upto which look ahead predictions can be made.

- **External factors:** Environmental events (like switching traffic lights) and elements (traffic behavior) that are external to the powertrain but affect its fuel economy are called External Factors.

- **Infrastructure Elements:** Traffic control and regulation infrastructure like stop signs, speed limits and traffic lights that form a part of the overall transportation infrastructure are called infrastructure elements.

- **Road** or **Edges:** The words 'Road' and 'Edge' is used interchangeably in this work, referring to virtual roads that are simulated in the traffic model.

- **Network** or **Map:** Refers to a virtual network of interconnected roads used for traffic simulation. The map (or network) consist of a set of road segments that are connected by intersections, and traffic flow between them is controlled by a traffic lights or priority based stop signs. Each road specifies maximum speed limits for vehicles traveling along it. The transportation network in any location can be catheterized in this fashion. For purposes of creating realistic driving scenarios, this information representing Columbus(Ohio,USA) is imported from a map database in this work.

- **Route:** Route is the user defined path assigned to the host vehicle that determines its travel from a given location to an other. The route is specified along various road segments that are defined in the Network being used for traffic simulation.

- **Effective Speed:** Presence of traffic congestions or dense traffic along a road leads to lower speed of travel than the maximum speed limit on that road. This average speed of traffic flow on a given road is called Effective Speed limit.

- **Real-world driving:** Drive cycle independent driving scenarios, where a vehicle's velocity dynamics is determined by its powertrain limits, speed limits on the current road and reaction to other vehicles in its immediate surroundings is called real-world driving.

- **Traffic:** Vehicles other than the host vehicle in the traffic simulation environment is called traffic.

1.2 Contribution and Organization of the Thesis

Each chapter of this thesis describes a unique contribution. The following list gives a brief insight into the contents and contribution of each chapter.

- **Chapter 2** records a high level literature survey of various sensors and technologies to perceive a vehicle's immediate surroundings and predict future driving scenarios. Subjective conclusions are drawn regrading the utility of each technology in enabling look ahead sensing.'Technology Enablers' are defined and philosophy behind constructing 'sensor suites' through a process called 'Preferential Technology Loading' is detailed. This involves transforming traditional classification of sensors and re-grouping them according to the information they capture. Six sample 'sensor-suites' are constructed by incrementally augmenting look ahead predictions with data from additional sensors, such that prediction from each successive suite is better than the previous.

- **Chapter 3** outlines the concept of traffic simulation and describes various aspects of 'Traffic In Loop' (TIL) simulation. It explores prior published attempts to achieve traffic integrated-powertrain simulation and points out the distinguishing features that makes this framework exclusive. Further TIL's role in various phases of developing powertrain controls and simulating connected vehicles is explored. The chapter concludes with a demonstrative discussion of results from the co-simulation that encompasses outputs from powertrain components and sensor emulations.

- **Chapter 4** introduces readers to the specific powertrain, and traffic simulation models used in this work. A onboard implementable lookahead based energy management strategy called 'Delta Energy Controller' (DEC) is developed for a Range Extended Electric Vehicle (REEV). A process to integrate look ahead information into powertrain control (DEC) is proposed. Simulation results show that DEC is robust to variation in traffic conditions and its performance approaches optimality as quality of look ahead information improves.

- **Chapter 5** summarizes contribution of this thesis and identifies areas where further investigation is necessary.

Chapter 2: Technology Enablers for Look Ahead Sensing

2.1 Introduction

Information plays a very crucial role in making a look ahead energy management strategy effective and help drive the system towards optimality. For the look ahead forecasts to be effective, information that would aid in constructing a trustworthy forecast of future drive scenarios needs to be collected. This requires looking beyond the powertrain and into the environment and the transportation system. These elements are called 'External Factors' and affect vehicle behavior in the present and future. Since these factors cannot be controlled, they can be viewed as noise or disturbances acting on the vehicle system. Examples of such 'External Factors' include traffic, grade, road speed limits, presence of stop signs, head wind etc. Apart from the mere availability of information, its reliability also plays a vital role in reaping the benefits of look ahead based energy management. The various techniques, sensors and sources that can furnish information beneficial towards understanding future driving conditions are termed as 'Technology Enablers', since they 'Enable' the construction of an informed prediction of future scenarios. The term 'enablers' is used interchangeably with 'sensors', both of which refer to''Technology Enablers'. The contribution of this chapter lies in classifying these 'external factors' and analyzing available technologies to capture them from a powertrain control and LEMS's stand point. Further, it summarizes results of an exploratory literature survey that was conducted to understand the types of information that can be obtained from these enablers and recognize the benefits associated with each source.A brief study on various fusion algorithms that are commonly used in modern vehicles has also been conducted. The last section of this chapter, introduces 'Technology Loading' and discusses the creation of sensor and information packages to enable various levels of LEMS.

2.2 External Factors and Their Classification

'External Factors' refers to those environmental elements that are external to the powertrain and have a potential to affect a vehicle's current or future behavior. It includes other vehicles on the road, road conditions, regulative traffic rules, traffic lights etc. Listed below are eight external factors which have been identified as being crucial to making look ahead forecasts. Table 2.1 lists out the various available sources for extracting a particular type of look ahead information.

External factors influence the host vehicle velocity profile and their effects create either a short or long-term variation on the host's velocity. Such variations are respectively termed as 'short term dynamics' or 'long term dynamics' as they cause the vehicle speed to vary from some set preferred speed. The preferred speed is considered to be a function of driver behavior and road speed limits. For example, the variation in vehicle speed triggered due to the behavior of a leader can be viewed as a short-term dynamic. However, a change in speed limit along a given section of a highway is said to be a long-term dynamic. Long term dynamics can be forecasted with considerable headway but short-term dynamics are those events that can only

External Factors and Technology Involved in Sensing		
Sl.No.	**External Factors**	**Technology Enablers**
1	Road Speed Limit	GPS+Offline Maps (or)Camera (or)I2V
2	Stop Sign and Traffic Light Locations	GPS+Offline Maps (or)Camera
3	Grade	GPS+Offline Maps
4	Traffic Density	Camera (or)V2X (or)LiDAR (or)RADAR
5	Leader Speed and Headway	RADAR (or)LiDAR (or)Camera (or)V2V
6	Leader Type	Camera (or)V2V (or)LiDAR
7	Traffic Light Status	I2V (or)Camera
8	Effective Speed Limits	Online Maps (or)I2V (or)Camera

Table 2.1: Various available sources for extracting a particular type of look ahead information.

be predicted once the vehicle has entered a given horizon, which is in the order of about 100 meters or 15 seconds [25]. The maximum horizon length for the short-range dynamics is determined by the range of sensors and other technologies that are involved. Another major difference between short and long-term predictions is that, short term predictions change at higher frequencies however long-term predictions change over a larger period of time (or distance). In other words, the perceivable effects of short term dynamics are felt over smaller time periods whereas the effects of long term dynamics are experienced over larger time period. For example, a short term variation like reactive variation in speed due to the behavior of a leader vehicle can cause continuous variation in the host vehicle speed. On the contrary, the change in speed limits on a highway are usually effective for at least a few miles (or few minutes of driving). Short term factors are affected by behavior of host's operator and other individual drivers and capture road randomness, have greater estimation confidence and influences the local minimum. From a powertrain control perspective, literature reports only a marginal improvement with a prediction horizon of more than 15s or 100m [25]. Long-term demand is affected by overall traffic flow and is dominated by traffic and driver's reaction to the future driving situations. They can be updated by short term predictions when event horizon approaches within range.

2.3 Technology Enablers for Look Ahead Prediction

This section deals with all the sensors that were studied for use as technology enablers in look ahead predictors to make velocity and drive power forecasts. The following subsection presents results of the literature study undertaken for each of these technologies. They explore various aspects of the technology including its physical components, shortfalls and strengths. Special attention has been dedicated towards understanding the technologies that contribute in improving look ahead forecasts. The concluding Section 2.3.7 offers a birds eye view of the study in Figure 2.3.7. It displays the benefits of each sensor from the perceptive of look ahead predictions and use in Advanced Driver Assistance Systems (ADAS). It is to be noted that the purpose of this study was to develop a basic understanding of various sensors to grasp their individual contributions in forecasting future driving scenarios from a powertrain control point of view. Subsequent sections of this chapter are designed to be a primer to various sensing technologies – providing a brief introduction to the working principles, classifications and measurement obtained from various sensing technologies .

2.3.1 Global Positioning Systems

Positioning systems are very basic and versatile technologies that provide absolute measurements. They aid the host vehicle in localizing itself and hence form the basis for any look ahead control application. Positioning systems are paired with maps to enable localization. In the domain of powertrain control, self-localization helps serves as a feedback to the controller - relating the vehicles current position with respect to the destination and/or other fixed references. Positioning helps measure trip progress and extract traffic, grade and other data from a different database etc. They are also used in combination with other sensors to estimate additional parameters like velocity, acceleration, grade, heading etc. With regards to velocity prediction, positioning systems can aid in remote sensing by keeping track of the vehicle positions and speeds along different routes. Such historical data can then be used by velocity predictors to create or improve future estimates.

Working Principle

There are various types of positioning systems operational today two of the most popular systems are USAs GPS (Global Positioning System) and Russias GLONASS. Though their setup may vary in various senses, all positioning systems operate via the basic principle of Trilateration. Trilaterization is position triangulation in a 3 dimensional space and can measure elevation data in addition to surface positioning. The process of trilateration is briefly explained with Figure 2.3.1

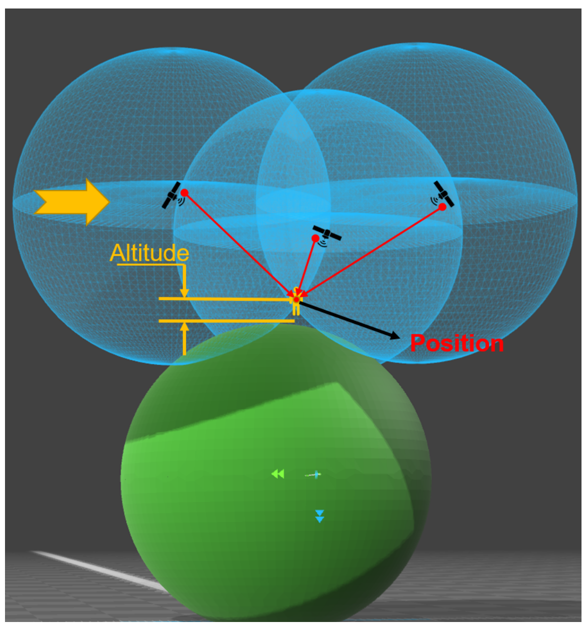

Figure 2.1: Illustration of trilaterization

To compute the position of an object(fitted with a GPS receiver), the GPS tries to measure the object's distance from at least three satellites. The location of these satellites can be computed precisely using the 'almanac' and 'ephemeris' data transmitted by the satellites and a mathematical model of orbital dynamics [26]. The transmissions from the satellites carry high precision time data thus inherently recording time of broadcast. Each satellite broadcast its respective signals by modulating its own unique binary sequence that serve to identify these satellites. By measuring the difference between time of broad cast and reception, the distance of the receiver from each of the satellites can be measured. Thus estimating the receivers position on the earth surface with reference to the satellites. More detailed introduction to GPS receivers can be found in [26].

The most commons forms of error in a positioning system can be classified into two categories, Random Noise (antenna noise, and noise due to numerical computation) and Atmospheric noise. Atmospheric noise is a slowly varying bias that can account for nearly 9-10m of location error [26]. Additionally, multi-path and diffraction errors are caused by buildings and other tall structures as satellite signals bounce-off from them. This phenomenon accounts for atleast 1.5m of error in urban areas. However it can be higher based on the satellite constellation, nature and density of such structures, and the location of the sensor. Path of GPS signals from the satellite to the onboard receivers can also be affected by urban canyons, tree covered regions, mountains and hills. Aforementioned factors can also cause a decline in number of visible satellites - further affecting the accuracy of measurement [27].

Types of GPS

There are multiple subcategories of the GPS technology; the salient features of each has been listed below [**?**, williams2012evaluation, corrections2004comparing]

1. **WASS Wide Area Augmentation System** is based primarily on the satellite infrastructure and covers entire US through geo-stationary satellites. This system yields nominal accuracy of 4-5m (95% i.e.., 1.95 σ)

2. **D-GPS Differential GPS** are based on satellite and ground infrastructure. Contrary to WASS sensors, D-GPS connects to satellites and grounded antennas (Base stations). The Ground antennas are immobile and have precise knowledge of its position. They measure GPS positioning errors by comparing the known actual position with their GPS estimated position every time instant. This error quantification information is then shared with nearby mobile receivers (vehicles) via FM carriers or cellular networks. D-GPS positioning is better than WASS estimations owing to This error corrections, which accounts for atmospheric noise. D-GPS yield estimates with a nominal accuracy of about 2-3m (95% i.e.., 1.95 σ) and their accuracy depends on the number of uncommon satellites used.

3. **RTK Carrier Phase Positioning** give the most precise and accurate position estimates in the family of GPS sensors. With a nominal accuracy 1 to 0.2m (95% i.e.., 1.95 σ). But their signals are generally weak, need complex hardware and require a higher processing time.

Application based selection of GPS for powertrain control

A given GPS technology can yield different accuracy in measurements depending on the number of parallel channels, error correction softwares used and antenna precision. Superior antenna systems ensure strong locks are maintained under circumstances that tend to create errors in accuracy of the receivers. The number of parallel channels influence the ability to quickly lock onto satellites when first turned on or when it comes out of a GPS black out; Thus influencing system accuracy for the first few seconds. As far as powertrain control and look ahead predictions are concerned, the classification of precision and accuracy can be viewed based on a sensors localization ability with respect to infrastructure elements as demonstrated below : [28]:

- **Road Level Realization** refers to identifying the road on which a vehicle is travelling. For example, this refers to the ability to distinguish if the host vehicle is on a service road parallel to a freeway or the freeway itself. Realizing road level accuracy, requires a sensor with positioning precision and accuracy of approximately 5m. Hence can be achieved with Wide Area Augmentation Systems (WAAS). These are economic and widely used standalone systems.

- **Lane Level Realization** involves identifying the host vehicles current lane on a given road. This kind of accuracy enables a velocity predictor to identify if a vehicle is traveling along a dedicated lane that may potentially affect speed limits. Lane level realization can localize a vehicle as being on a said lane but might not be able to estimate its closeness to the edges of that lane. It requires a sensing unit that is accurate and precise to at least 1.5 m.

- **Within Lane Realization** is often useful in autonomous vehicles and V2V based safety applications - including Electronic Emergency Brake Light EEBL), Forward Collision Warning (FCW), and Lane Change Advisor (LCA). This form of realization demands the highest levels of accuracy and precision in the sub-meter ranges. This form of localization can help a vehicle position itself within a given lane.

In summary, positioning accuracy requirement depends on the type of look ahead predictors being used and the level of data required by them. In the situation where remote sensing of vehicle speed is performed by an off-site server, the accuracy of onboard positioning system must be accounted. Given the specification of GPS sensor being used on a given vehicle, variance in perceived velocity can be estimated - as a function of its accuracy, sampling frequency and the vehicle speed. Variance thus calculated can be used to derive an approximate velocity profile that consist of a velocity band with varying widths. Figure 2.3.1 illustrates this effect, where measurement variance changes with the speed being measured.

The above figure 2.3.1 shows that width of the variance band is not constant, speed measurements from GPS sensors are inherently erroneous at lower speed and hence the estimate accounts for this difference. This can be numerically proved using statistical multi variance analysis. The basic formula for velocity is given as the rate of change in position (P) over time taken (T) as given in equation 2.1.

$$V = \frac{\Delta P}{\Delta T} \tag{2.1}$$

The multi variable variance ($sigma$) of a measurement 'x' is calculated as follows 2.2:

$$\sigma_{x_i}^2 = \sum_{n=1}^{n} \left[\sigma_{x_i} \left[\frac{\partial f(x_1, x_2, x_3......x_n)}{\partial x_i} \right] \right]^2 \tag{2.2}$$

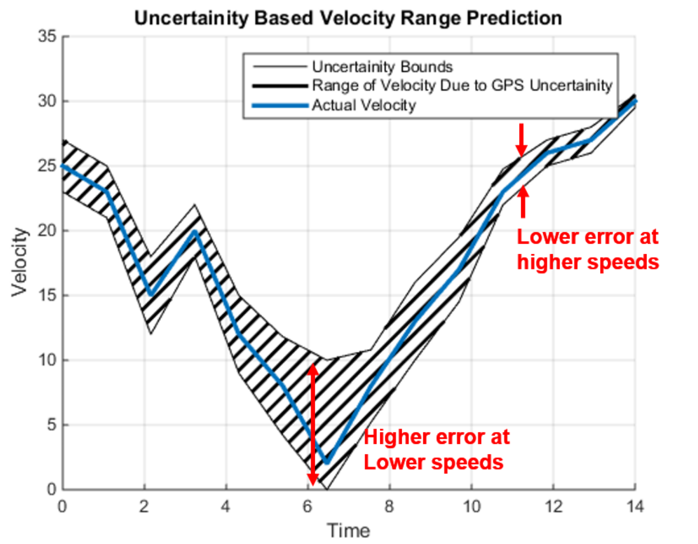

Figure 2.2: Remote sensing velocity from GPS measurements

Performing the operation from equation (2.2) on (2.1), we derive an expression (2.3) for variance in velocity measurement from measuring difference in position and assuming a straight line of travel.

$$\sigma_v^2 = \sigma_p^2\left[\frac{1}{\Delta T}\right]^2 + \sigma_t^2\left[-\frac{\Delta P}{(\Delta T)^2}\right]^2 \tag{2.3}$$

Where, $\sigma_v, \sigma_p\ and\ \sigma_t$ signifies the uncertainties in velocity, position and time respectively. The above approach uses only difference in position over a set time to calculate the velocity. However, in commercial positioning systems, velocity is derived based on a combination of Doppler effect and differentiation of position data. Doppler effect is the shift in radio frequency of the GPS satellite signals, that is introduced by the relative motion between the geostationary satellite and a mobile receiver. We can conclude from expression (2.3) that uncertainty of the velocity measurement and the accuracy of the GPS sensors are not linearly related. Assuming that the uncertainty in time caused by clock drift and clock inaccuracies is negligible. Equation (2.3) shows that a 3 fold improvement in accuracy ($\sigma_p/3$), we observe 10 times better estimate of velocity ($\sigma_v/10$).Whereas a 5 fold improvement in accuracy ($= \sigma_p/5$) translates to a velocity profile that is 24 times more accurate ($\sigma_v/24$). In conclusion, since the variance of measurement varies with speed of the vehicle, the uncertainty band is higher at lower speed and lower at higher speeds as shown in Figure 2.3.1.

2.3.2 Inertial Measurement Units

Inertial Measurement Units (IMU) are vehicle state measurement devices. They help determine dynamic states like acceleration and speed. Position estimation from GPS are subjected to various kinds of errors as discussed in section 2.3.1. Measurements from remote location methods like GPS are prone to interruption due to obstruction of satellites by buildings, trees overpasses etc. The primary role for IMU's in look ahead prediction application is to improving the performance of the positioning systems. This is achieved by a process known as 'dead reckoning'. Which can be defined as position tracking based on previously known position, given the instantaneous acceleration over a certain time period. IMUs can also be use for verifying of GPS position - this helps eliminate 'jump errors. Jump Errors are positioning inaccuracies that cause the host's perceived location to change at random - appearing as if the vehicle has jumped from one point to an other.

GPS sensors measure position whereas, IMU measures instantaneous acceleration. In order to compute vehicle position (given an initial position on a map), acceleration measurements from IMU are integrated twice. Thus while comparing position estimates from each of these sensors, we see that GPS provide low drift measurements that are updated at lower frequencies. While IMU's yield short interval measurements which are prone to drift heavily owing to noise integrative errors. However this complementary feature, is exploited by GPS-IMU integrated systems - called Inertial Navigation System (INS) and are widely used for cost effective positioning and navigation [29, 30, 31]. The amalgamation of GPS and IMU measurements help improve positioning accuracies by using complementary filters[29, 30, 31, 32].

Complementary Filters

These are special type of sensor fusion filters used to fuse two noisy measurements of the same signal with complementary properties. In this case, IMU provides good information in the short term (high frequency data and noise), while GPS provides good information over the long term (low frequency data and noise). The process of detecting a point in time when the velocity of an object is zero and resetting the integral of acceleration at that point, is termed a zero velocity update. These updates can be used to limit error growth in inertial tracking devices like INS. Most commonly used GPS sensors have an update frequency of about 1 second. In order to check the position estimates from the GPS sensor, INS uses the IMU to measure distance independently by integration of measured acceleration over time. Various algorithms are then used to determine a more accurate estimation of position by differentially weighing the two measurements. Distance measurements from IMU's are reset when the vehicle comes to a halt or a strong GPS fix is obtained that eliminates GPS estimation errors. Some INS systems try to bridge GPS gaps and reduce error growth by using various auxiliary sensors- such as compasses, inclinometers, tilt meters, and odometers [33, 34].

Accurate estimation of position using Inertial Sensing Units are important for look ahead forecasting specifically in closely packed urban roads. Occlusions caused by urban canyons can lead to noisy GPS positioning that tend to indicate the vehicle's presence on different roads between GPS updates. If the speed limit on the said roads are different, the predictor changes its prediction at every update interval. Such behaviors reduce the quality of look ahead prediction and thus the optimality of control obtained from those predictions. Using INS can help avoid such random errors and keep the vehicle locked on to a given road segment. An alternative to this is to use higher accuracy GPS sensors but the cost of upgrade might out weigh benefits. However correction to positioning system from the IMU cannot be performed over extended periods of time since the changes are not known accurately and the process is subject to cumulative errors.

Errors and Compensations Associated with IMU Sensing

Every inertial sensor involves measuring the movement or mechanical change of some internal component(s). This measured motion is then related to vehicle state (Roll, Pitch and Acceleration) based on known physical relations. Based on the physics used, one can expect different levels of accuracies to determine vehicle state. However, a major contributor in sensor uncertainty and error is the technique and internal sensor, employed to quantify the mechanical change induced in the sensing components. Specifying the errors caused by these internal sensors is outside the scope of this work. However IMU errors can be classified into four types as briefly listed below[35, 26] :

- Scale factor error is a ratio of change in sensor output with respect to true intended measurement. It can be estimated by the filters implemented in the inertial measurement unit.

- Cross coupling error is the result from non-orthogonality in the mounting of inertial measurement sensors along the X,Y and Z axis of the sensor unit. This error is estimated during the calibration - required corrections are applied to the sensor's processing unit at the time of manufacturing.

- Errors can also be caused due to imbalance in the proof mass of the spinning or vibratory mass within individual sensing elements

- Random errors are caused due to noise from electrical and mechanical instabilities. During static operation, this error follows Gaussian distribution. The same assumption is often used to filter noise over dynamic operating points as well.

When integrating acceleration to arrive at a position, an uncompensated accelerometer bias error will introduce an error proportional to the elapsed time in the velocity estimate. This error is proportional to square of elapsed time in the position estimate[32]. Generalizing these calculations as performed in [36], we can see that a bias error gives a position error e_P of $e_P = 17.66m/mg/min^2$. Which implies an uncompensated in-run bias stability of 0.05 mg can introduce positioning error(e_P) of 1 meter and velocity error(e_V) of $e_V = 0.589m/s/mg/min$. So external speed measuring devices like GPS and wheel speed sensors serve as resets for these integration operations and are valuable resource for bounding such errors.

2.3.3 Vision Based Sensors

Vision based sensors can be divided into single camera systems, stereo camera systems and RGB-D sensors. All three types can be used for Road sign detection, Traffic light detection and traffic density estimation. Some applications have also been identified in helping to improving GPS accuracy. All vision sensors are passive sensors and are affected by ambient conditions like lighting, weather and pollutants (dust, smoke). The sensor hardware is low cost but processing hardware requirements can increase the overall material and developmental cost of the system. Single camera systems are used commonly in lane and obstacle detection systems. These systems are often capable of measuring headway to a detected obstacle albeit with low accuracy [37] and [38]. However appropriate software can help build robust collision avoidance systems though generation of occupancy grids. Multi-camera systems on the other hand provide more accurate ranging capabilities [39]. 'Red Green Blue - Depth' (RGB-D) vision technology, provides traditional images and depth information for each pixel at a fixed frame-rate.

Single Camera Systems

Commonly used in lane departure waning systems and can aid look ahead controller in localizing the host vehicle on to a lane given the approximate position on road as shown in Figure 2.3.3. This becomes important for a look ahead velocity predictor when dealing with dedicated lanes on highways.

In single camera systems, ranging is achieved by using inverse perspective mapping [40]. This process involves transforming the camera's plane coordinates to a rectified birds-eye view. The input image pixels are mapped to the road reference

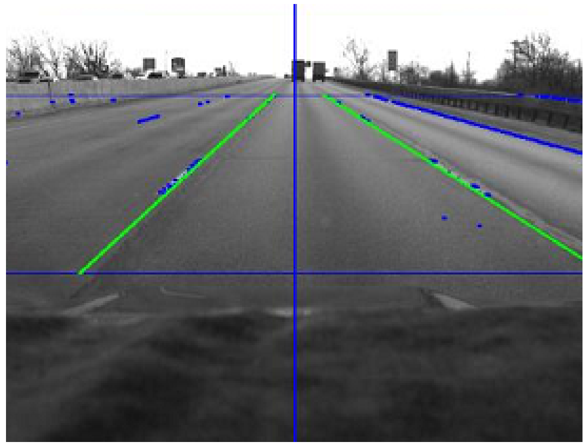

Figure 2.3: Lane detection using pruned hough transform [3]

system, and a new image is produced where perspective effects are removed. Resulting image becomes a top view of the road where each pixel on rectified image can be associated with a known distance on the road. This technique is based on an assumption that the road is flat and that there exists a fixed rigid body transformation from the camera to the roads reference frame. However the later assumption is not completely true as the angle between camera and the ground is affected by pitch and yaw when vehicle is in motion. Inverse perspective mapping of a given scene is shown in Figure 2.3.3, it can be observed that the transformation shortens range of the system. An alternative to this transformation is to use the apparent motion of portions of an image between frames and determine relative motion of the camera. This requires three basic operations, first, finding features in an image suitable for tracking. Second, Matching these features in a subsequent image and finally solving for the resulting camera motion. Thus this technique relies heavily on the image

Figure 2.4: Inverse perspective mapping [3]

processing tools (hardware and software) that is used hence a formal qualification of errors or confidence cannot be established.

State of art also report of methods to improve the host vehicle's self localization accuracy through use of an upward facing camera [1]. This technique is especially useful in urban canyons where GPS systems are prone to multi-path errors. The upward facing camera is used to detect those satellites who do not lie along the line of sight. In Figure 2.3.3, the satellite 'PRN7' is obstructed by a building. Signals from this satellite do not reach the GPS receiver along a straight line of sight but undergo multi path reflection before reaching the host vehicle's GPS receiver causing a bias in the position measurement. However when the signals from this particular satellite is ignored, the positioning becomes more accurate.

Figure 2.3.3 indicates that by avoiding measurements that cause large biasing errors in the positioning computation, the two-dimensionalpositioningaccuracy improves. The radius of uncertainty is observed to be 31.4m before correction and reduces to 7.6m after correction. (ie., 75.8% decrease in radius of uncertainty).

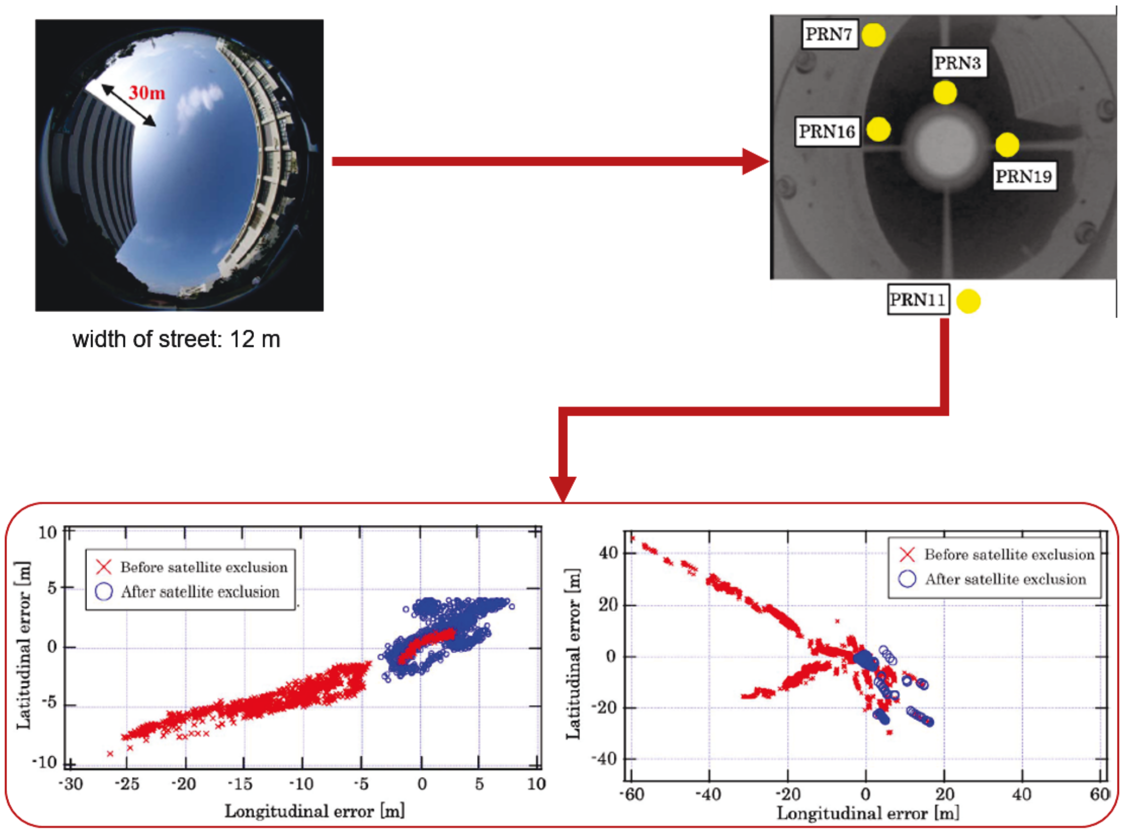

Figure 2.5: GPS multipath mitigation using upward facing camera [1]

Road sign detection can be carried out using single camera systems. Histogram of oriented gradients (HOG) is an effective way to capture shape information [41] and records upwards of 90% accuracy under ideal conditions but depends on various factors such as image processing capability, cameras parameters (frames per second, pixel density etc.),host vehicle's velocity and on road obstacles. Experimental data from literature [42, 43] suggestes successful detections of above 96% on cloudy and sunny days. This number drops to about 90% during night and rainy days.

The stages involved in any road sign detection algorithm are summarized in [44]. The preliminary step is called color segmentation and involves separating the segments from the image; based on color or Hue, saturation and value. Further, shape recognition and sign description is performed to recognize possible traffic sign patterns. Finally the detected patterns are classified with respect to known traffic signs using standard databases. Techniques such as neural networks are frequently used in the final step. Irrespective of the image processing technique used, all traffic sign recognition systems suffer from a few common weaknesses that make reliable detection difficult [44]. The most common amongst which are, variation in lighting conditions, presence of other objects in the scene that occlude the signs or create ambiguity, vandalism of sign boards, motion blur, and variation in orientation, specifications and location of the signboard. Detection for targets with a lateral distance of 10 m, vertical distance of 7m and at vehicle speeds of up to 250kph have been recorded by commercial systems at the time of compiling this document[45].

Single camera systems can play an important role in detecting traffic lights. However this would involve exercising more intelligence and image processing, since identification of relevant traffic lights can be relatively complicated. When used in conjunction with a look ahead system, upcoming braking events can be foreseen in addition to making start/stop decisions. Thus positively impacting the regeneration and acceleration management onboard. However, the range of detection is limited in comparison to a V2I system. Ranging sensors (like RADAR and LiDAR) give a reliable estimate of headway to a vehicle ahead but are unable to identify its type. Camera based systems can be used as a secondary unit in this scenario to identify the type of leader.

Knowledge of leader type can help short term acceleration predictions backed by statistical data. For example, a large vehicle such as a bus exhibits lower acceleration in comparison to a car, this everyday observation can be backed by statistical data that quantifies the same. Hence, knowledge of the expected acceleration of a leader can help in better prediction of the host vehicle's velocity profile over a short horizon. This technology can also be used to form an understanding of driver intent in case of a vehicle 'stop' event. For example, using a vision system can help a look ahead predictor understand if a given vehicle stop was a consequence of a traffic light, stop sign or driver intended stop. Such classification can help the predictor from isolating those factors that would affect all vehicles versus those that dont.

Image Processing Techniques

Figure 2.3.3 provides a high-level overview of some the most commonly used algorithms in onboard image processing systems for automotive applications [2, 3, 4, 5]. Each algorithm has its individual strengths and weaknesses in its original form. Improving them through various mathematical formulations and alterations is a topic of current research within the scientific community. Selection of appropriate algorithm for a given application is subjective. often more than one algorithm is used too analyze the same frames. Algorithms like matched filters which are computationally less intensive are used for primary analysis before advanced algorithms are used to analyze specific sections of the frame. Algorithm selection is also decided by constrains such as camera specifications and onboard processing capabilities. 'Canny Edge Detector' algorithms are often used to locate traffic signs like speed limit or stop signs as they are robust in bad weather and ambient light conditions. In this technique, a Gaussian blur is applied to the image are based of creating a brightness gradient

Algorithm	Description	Pros	Cons
Matched Filter	Contrast/brightness segmentation	• Lower processing load • Quick linear pattern matching like lane detection • Can Work with lower camera resolution and bad weather conditions	• False detections • Text recognition is difficult • Affected by local lighting
Canny Edge Detector	Brightness gradient - jacobian matrix generated	• Robust in bad weather / bad ambient lights • Simplicity of process for implementation • Find 'true edges' with noise resistance and minimal response if tuned	• Greater processing • Bias towards horizontal and vertical edges • Greater resolution camera needed • Tuning intensive
RANSAC	Feature based extraction	• Curve fit based feature recognition is simplified Lane marker detection under bad marker visibility	• Susceptible to noise • Best only for straight line and curve fitting • Text recognition is poor with low powered processor
Hessian Point Detector	Derivative of gradient - hessian matrix	• Responds well to textured scenes in which there are a lot of corner-like features	• Heavy processing • Edge orientation information is lost
Pruned Hough Transform	Every possible line is voted upon	• Handle uncertainties in vision • Noise filtering	• Text recognition is difficult need multi core processing

Figure 2.6: Overview of image processing techniques used commonly in vision sensors [2, 3, 4, 5]

for each pixel of the frame. when tuned appropriately, they can suppress noise and are not affected glaring caused by oncoming vehicle lights or sun near the horizon. RANdom SAmple Consensus (RANSAC) is a feature based extraction policy that is used for lane detection or fitting other straight lines and curves. This is a brute force technique and looks for predefined patterns in every frame. By virtue of it operation, this technique is not suitable for sign detection. Pruned Hough Transform is also similar to RANSAC in a sense that in this operation, each point is voted for being part of a line that passes through it. Thus it is similar to curve fitting a line on every frame. However, this algorithm is better than RANSAC as it can handle uncertainty and noise.

Stereo Camera Systems

In addition to the image of environment around the vehicle, stereo camera systems can help measure distance to detected objects. Unlike RGB-D, RADAR or LiDAR systems that broadcast a specific electromagnetic waves and study the reflected signals, these systems work in a passive manner through post processing information from multiple sources. It consist of two cameras each of which observe the environment from a slightly different perspective. Usually the two cameras are mounted horizontally with some separation. A depth or height map can be extracted by corresponding the detected features from each camera with respect to the other. The separation distance between the two cameras is related to the depth of field and resolution as shown in Figure 2.3.3 and Equation 2.4.

$$Z = f \times \frac{B}{(X_r - X_l)} \tag{2.4}$$

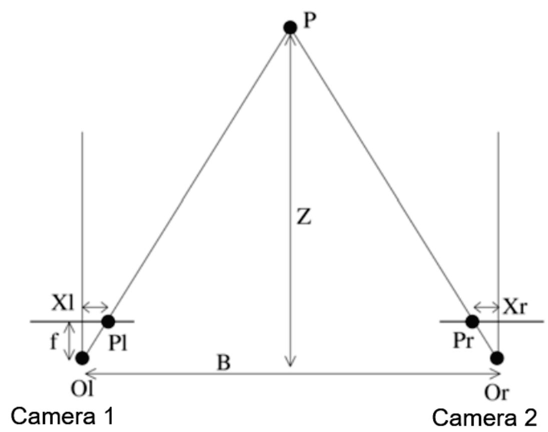

Figure 2.7: Depth calculation from stereo camera system

For automotive applications, vehicle width would limit the maximum depth of field and resolution of the system however better performance can be obtained by using a camera with more pixels. The precision of the distance measurements depend on accuracy with which pictures from the two cameras can be correlated. Error varies with distance of object from the camera. When a target object lies further away from the camera system, it is represented by a lower number of pixels. Hence the resolution of depth map decreases for objects that are further from the camera system.

Mono-cameras systems could only detect and classify objects, Stereo camera provide additional layer of information in the form of distance to detected objects. Hence, stereo camera systems can help monitoring headway and measuring speed of vehicle ahead. Additionally, they can aid in monitoring speed of other vehicles on a given road that are visible to it. Such information can be used to calculate average speed of traffic flow on a road, this infomraiotn can then be passed on to other vehicles upstream via V2V communication or a central cloud based server.

RGB-D Systems

RGB-D sensors yield conventional RBG (Reg Green Blue) color images but also provide pixel wise depth information. While 3D point clouds obtained from a ranging sensor like LiDARs are well suited for frame-to-frame alignment and 3D reconstruction of the environment, certain types of information can only be gained from a visual inspection of the environment. Outputs from RGB-D devices are similar to that obtained from the fusion of a camera and a ranging sensor and yield information about the distance to an observed object and specify various aspects of the nature of object being observed. Unlike a regular camera, depth information is derived by projecting an infrared speckle pattern on the environment and recording the reflected pattern with an infrared camera. The recorded speckles is compared part-by-part to reference patterns that were captured previously at known depths. The system then estimates the per-pixel depth for which the reference patterns and projected pattern match the best [46]. On account of their operating principle, these devices are highly susceptible to influence from dust and other environmental factors. Depth estimates are noisy and their field of view(60^o) is far more constrained than that of the specialized cameras and laser scanners commonly used for 3D mapping (180^o) [47]. Other types of RGB-D devices use either active stereo [konolige2010projected] or time-of-flight sensing [kisavcanin2014advances] to generate depth estimates for each pixel. Key drivers of this technology in todays world are computer gaming and home entertainment applications. The technology behind RGB-D sensors are not fully matured and continue to remain under development. In its current state, commercially available sensors provide depth only up to a limited distance, typically less than 5m [48]. However given an opportunity to use this technology in the domain of powertrain control, they could serve as a single alternative over using two independent technologies.

2.3.4 V2X Communication

V2X communication signifies any wireless communication between vehicles and other Vehicles(V2V), infrastructure(V2I) or a central cloud based Server(V2S). Such technology enables vehicles to be transformed into moving sensors capable of communicating various parameters like position, speed, brake activation, thus aid in real time decision making. Incorporation of V2X information can improve the velocity prediction significantly and promising improvements in the estimation performance have been demonstrated[25]. An illustration of a potential application is provided in Figure 2.3.4 Two major technologies have been employed to facilitate information transfer in V2X Communications Dedicated Short Range Communication (DSRC) radio and 5th Generation mobile networks (abbreviated as 5G). Market studies indicate a Steady increase in growth and adoption of V2X technology over the next decade [] at the end of which 5G will be the preferred technology over DSRC for V2X communications. Data enabled OBD-II adapters can enable V2X services for the 1.2 billion vehicles on the road today arent connected. Apart from DSRC and 5G technology, weather and traffic data communication has been demonstrated using AM and FM radio broadcasts. This section explores the various types of V2X networks, benefits and weakness associated with each, information flow mechanism involved and the types of information that can be exchanged via these routes. Introductory information on V2V communication can be found in [49]. Both 5G and DSCRC based communication techniques have individual strengths and weaknesses as discussed in [50]. Here the technologies are contrasted based on a powertrain controller/predictors view and physical implementation. 5G based systems are more energy efficient, provide better coverage and high down and uplink capacity. However, in the current form, their performance decorates at higher vehicle speeds and tend to have unreliable latencies in regions with high mobile network usage. Owing to the large coverage and centralized operation, identification of peers is often complicated and needs to be done offsite on the central server. DSRC however mitigates the drawback with respect to peer identification due to smaller coverage area and selective admission of peer vehicles to the network. DSRC based systems are also cheap and easy to implement since required technology have been developed to a good extent. Hence, they are best suited for dissemination of traffic signals, cooperative adaptive cruise control and other short range applications. Nonetheless DSRC based V2X communication is hampered by several technical disadvantages like channel congestion band leakage etc. In conclusion neither of the two technologies is superior to the other and technology selection for a given application should be done on a subjective basis bearing in mind that the selection will affect the look ahead horizon of the system. Irrespective of the communication technology used, the architecture of a typical V2X System is somewhat unchanged as described in [49] and briefed below:

1. **On-board unit :** An OBU is a device that resides in vehicle and helps in sharing information with RSUs or with other OBUs. Its components include a resource command processor (RCP), a user interface, a specialized interface to connect to other OBUs and a network device for short range wireless communication based on IEEE 802.11p radio technology. This device has network

capabilities which is used to send, receive and forward information without wires on the fly. Functioning capabilities of OBU includes routing, quality of service, security and IP mobility.

2. **Application unit :** An Application Unit (AU) is device that exists in vehicle and executes applications utilizing the commuting capabilities of the OBU. The AU can be either a dedicated device for specific application say security or a device with communication to run the Internet. An OBU can have multiple AUs embedded in it that can share OBUs resources and processing. AU may be embedded inside a OBU making it a single unit or both may be separately attached through a wired or wireless connection. There is only logical difference between AU and OBU.

3. **Road-side unit:** It is a road side infrastructure that may be fixed along roads junctions, parking slots, petrol pumps, eating joints, etc. It comprises of a device with networking capabilities that works for short range wireless communications using IEEE 802.11p.

V2V Communication

V2V communication networks allows direct information exchange between vehicles without fixed infrastructure or base stations. In this case the location and velocity of the nodes (vehicles) are changing constantly, and the communication range is short. The set of vehicles that can directly communicate changes continuously as vehicles continuously join and leave a said network. Two nodes must be able to communicate securely when are in range. The physical layer and the network must be ad-hoc, decentralized. They can be classified based on the networks architecture as Local and

Network based communications. **Local V2V communication** refers to direct communication between vehicles- enabled by DSRC technology and exhibits low latency. However, it is beneficial only when operating as a part of a group, have a limited range and require co-operation between various operators. Local V2V communication also needs to account for disturbance from non-connected vehicles. Alternatively, **Network based V2V** operation is individualistic and independent of other vehicles. All vehicles are connected to a global live server that coordinates the various vehicles and is not limited by range. I2V data can be consolidated as a part of the Network based V2V information packet, provided that the infrastructure elements share required information to the V2V servers. Owing to the involvement of a central server that is an offsite entity, these networks can exhibit high latency. Nevertheless, 5G communication technology is said to possess the capability to address this shortfall, but no commercial implementations of the same were identified in literature. In addition to Offline GIS information, Radar, Vision and other onboard sensors, Data from other vehicles that are ahead of the subject vehicle along a give route used to better predict the current traffic future conditions. An illustrative example of the same is provided in Figure 2.3.4 here the arrows indicate the direction of information travel and the 'Satellite represents an offsite server that facilitates communication.

In Figure 2.3.4, If a truck T1 has already gone past a road segment 'A-B , then information about the trip 'A-B can be sent to T2. This would can then be used in an optimal control sense to improve the control decision for T2. For example, if an Adaptive-Equivalent Consumption Minimization (A-ECMS) strategy is used to control torque split of in T2; historic velocity data from T1 would aid in arriving upon a better initial estimate of the equivalence factor through model based predictions.

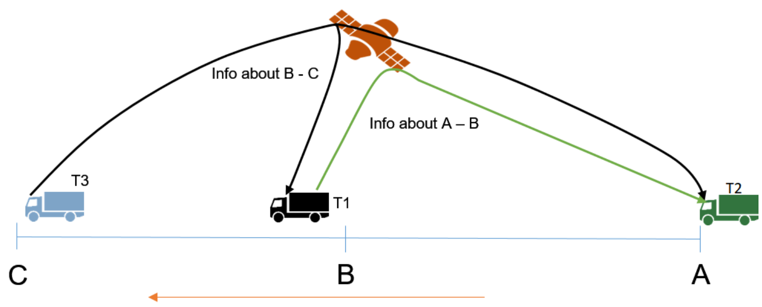

Figure 2.8: Illustrative example of V2V communication assisted by off-site servers (Adapted from [6])

This would mean T2 operates more optimally compared to T1. However, variations caused by difference in traffic and other external conditions may still affect the optimality of T2. Generally, data from V2X estimates can be used to approximate future velocity profile and make adjustments in control scheme for optimality. The range deficiency of local V2V communication is addressed by setting up Mobile Ad Hoc Networks (MANET) also called as multihop V2V communication. In conventional local V2V communication system, two vehicles that are close enough communicate directly in the form of a point to point broad cast. Whereas in multi-hop systems, Intermediate nodes forward messages between vehicles that cannot communicate directly owing to the separation between them being lager than their individual range [51]. However, this increases the latency of the communication system. Despite the increase in communication range introduced by multi-hop communication setting up a V2V network can be challenging at lower vehicle densities. In other words, V2V networks require an adequate technology penetration to make it beneficial. Additionally, on urban roads, signals subjected to interference from other sources and attenuation from buildings and other structures.

V2I Communication

V2I communication can be classified into two broad categories, Bidirectional and Unidirectional. In the former case, the vehicle communicates point to-point with a base station. The base station is responsible for coordinating the communication and can provide access control and congestion management. On the contrary, a unidirectional communication systems transfers information either from a broadcast station to the vehicle or vice versa (e.g. Automatic toll collection on highways). V2I information used by a powertrain controller or look ahead predictor primarily corresponds to traffic congestion, route and weather information. These parameters can help an intelligent controller identify upcoming velocity and acceleration profiles. For example, a V2I system may provide the look ahead controller with data regarding expected travel times to a given destination or a way point along the route. This data can then be used to calculate expected traffic flow speeds and hence the expected deviation in host vehicles speed from the road speed limit. This updated information can then be used by a controller to make optimal control decisions.

Infrastructure based Traffic Data Collection

Apart from speed, position and acceleration information from peer vehicles, V2I communication yields data about traffic along the route. This includes information on speed monitoring, traffic counting, presence detection, direction headway measurement, vehicle classification etc. Such measurements are carried out by either using Intrusive or Non-intrusive techniques. Sensors such as inductive loops, magnetometers, pneumatic road tubes or piezoelectric cables that are installed directly on the pavement surface directly measure traffic are called intrusive sensors. On the other hand, Non-intrusive Sensors like LiDAR, video image processing, RADARs, passive acoustic array etc. That are often mounted above or on the side of a roadway such that they can monitor multiple lanes of traffic and flow direction [52]. The operation, strengths and weaknesses of most commonly used traffic detection sensors[52, 53, 54, 55] are itemized below:

- **Pneumatic road tube:** - These are nature and well understood that is flexible to satisfy large variety of applications. It provides basic traffic parameters (e.g., volume, presence, occupancy, speed, headway, and gap) while being insensitive to unfavorable weather such as rain, fog, and snow. Its accuracy for traffic count data is comparable to other more advanced sensors hence it is a common standard for obtaining accurate occupancy measurements. However, the installation requires road work and involved cutting the pavement. Often multiple detectors are required to monitor a location. Classification of traffic based on vehicle types is challenging with this technique

- **Inductive loop :** This is similar to the pneumatic sensors described above. However they experience lesser stress due to traffic and have better life and can measure traffic speed.

- **Magnetic sensor :** Also called as induction coil magnetometers and can be used where loops are not feasible (e.g., bridge decks). They are also Insensitive to inclement weather such as snow, rain, and fog. Installation of these systems requires pavement cut or boring under. In their basic form, they cannot detect stopped vehicles but can measure speed of moving traffic.

- **Video image processor :-** Image processing can easily monitor multiple lanes when positioned at a 30-50ft height and offers speed measurement capability. Addition and modification of detection zones is simple. These are the only class of sensors that can classify perceived vehicles into their respective types. Nevertheless reliable operation at poor lighting conditions requires external lighting and their performance is affected by dust, fog , rain and snow. Camera based sensors demand higher maintenance in the form of periodic lens cleaning that increase operating costs.

- **RADARS :** being non-intrusive sensors they are often mounted such that they can measure traffic flow over multiple lanes. Additionally, they can directly measure traffic flow speeds and estimate lane occupancy. Some RADAR systems also offer the capability to estimate type of vehicles being sensed hence can aid in traffic classification. However, these advantages come with the higher cost of maintenance. Microwave based RADARs are insensitive to inclement weather, but Active Infrared RADARs can be affected by rain, dense fog or blowing snow.

- **Ultrasonic and Acoustic Array based sensing :** These sensors often function on time of flight measurements and can offer Multi lane detection. Since an acoustic signals time of flight varies based on the prevalent condition of the transmitting medium, environmental factors like temperature, and extreme wind can affect their performance. Moderate to high vehicle speeds, can affect the quality of measurements. They can not

V2S Communication

Vehicle to Server (V2S) is an umbrella term that includes some forms of V2V and V2I communication. Any form of communication between two nodes that takes place over an offsite server is considered to be a form of V2S communication. For example, V2V communication via a satellite link and traffic broadcast over a mobile network is said to be a form of V2S communication. However if the same traffic information was received through DSRC hotspots along the route then, it is considered to be a part of V2I communication. In Summary though the type of information being transmitted might be similar between V2I/V and V2S, the presence of a central server agent increases the range of communication. The server also works as an aggregator of information from various sources and performing offline computations thus add further value to the information that is being communicated. For example, a server that receives data from road side traffic measurement units regarding the traffic count on a specific road segment, servers can combine the count information with that collected from data received from various vehicles that are currently traversing that on-road to yield richer data corresponding to traffic densities and average flow speed of the traffic. Such combination of data can potentially help an adaptive controller estimate appropriate control actions for predicted future scenarios. Albeit the involvement of an external server increases the networks latency, a reasonable delay in the order of seconds is tolerable from a powertrain control perspective as it is not safety critical.

2.3.5 Specialized Sensors

This section discusses a variety of sensors that have not been used on board automobiles today, however, offer a unique perspective of the vehicles environment and hence have a potential to be useful in powertrain control and look ahead forecasting. On board sensor or Vehicle mounted sensors, rely on remote sensing techniques to detect near surface environmental properties. Vehicle mounted sensors allow operators to survey larger areas and detect real-time surface conditions. Various types of onboard sensors have been identified for the purposes of powertrain control and look ahead forecasting as listed below:

- Laser doppler velocimeter
- Anemometer
- Road roughness sensor
- Slippery road detection sensor
- Road temperature sensor
- Ice detection sensor

LDV Laser Doppler Velocimeter

The basic principle of LDV is the Doppler Effect. It utilizes the relationship between scattered and incident light frequencies when the laser beam strikes on a moving surface to estimate relative velocity. Hence, they can be used to obtain precise ground speed estimates. Literature reports two types of LDV sensors 'Reference Beam Sensors and 'Split Reuse Type sensors. Reference Beam sensors are self-adapted to different vehicle to road surface height as long as the emission angle is fixed [56]. In Split Reuse Type LDV, scatted light power is twice that of the reference-beam configuration for the same scattering surface. Hence signal to noise ratio of the instrument is improved in comparison. However, since the measurement is 1D in this case, velocity measurement accuracy is compromised on bumpy roads. From a powertrain control perspective, LDV pointed towards the ground can be used for accurate measurement of vehicle velocity and enhance dead reckoning accuracy. Prior art [56] shows that the positioning error from dead reckoning with LDV sensors can be limited at less than 20m in two hours of driving. Hence LDVs can be useful in urban canyons and along underground tunnels where localization is difficult through GPS. Alternatively, it can be used for obstacle velocity measurement if mounted to face direction of travel. LDV have a wide dynamic range, high measurement accuracy, good linearity and fast dynamic response non-contact measurement. However, the Sensitivity of detectors and frequent signal loss due to poor reflectivity of road surface are weaknesses of this technology that needs to be addressed. Additionally, Cost of precise lasers with non-varying frequency, size and robustness of these sensors make this technology unsuitable for automotive application in their present state.

Anemometers

Anemometers are devices used for measuring wind speed and direction. Ultrasonic, Hot Wire, and vane type anemometer are most common types of anemometers in todays market. Hot wire anemometer is based on current required to offset thermal convection while vane type anemometers are based on the movement of mechanical vane. Given the lack of moving parts, ultrasonic anemometers fare better in terms of longevity and environmental robustness and thus best suited for vehicle-mounted applications. Ultrasonic anemometers are based on time of flight of sound waves between 2 transducers. They work on the principle that sound waves are faster in the direction of wind than against it. They measure time taken for an ultrasonic pulse of sound to travel from the North transducer to the South transducer, and compares it with the time for a pulse to travel from ‘South to ‘North transducer. Likewise, time of flight is compared between ‘West and ‘East transducers. The wind speed and direction is calculated from the differences in the times of flight on each axis. Such anemometers are currently used in wind measurement at weather observatories, airports, wind Tunnels etc. Though automotive applications have not been demonstrated in literature yet, onboard head wind measurement can be used in power prediction and grade estimation (given engine torque and vehicle load). Realtime wind measurements can also be communicated via a V2X network

Road Condition Measurement Roughness, Slip, Temperature, Ice

Road condition measurements can be made either onboard or information gathered from infrastructure based elements like ‘Embedded pavement sensors and ‘remote surface sensors can be broadcast to vehicle using V2X networks. Sensors that are in direct contact with the pavement surface are designed to endure harsh conditions and high traffic flows. Their face mounts flush to the pavement surface and measures near surface properties like presence of ice, surface temperature etc. Remote surface sensors are pole mounted or otherwise installed above the measured surface and offer a non-invasive measurement of road surface conditions.

Road Roughness is measured by the international roughness index (IRI). From powertrain performance standpoint, road roughness impacts vehicle speed and fuel consumption [57]. Two types of roughness Measurement devices that can be used on vehicles [57, 58] :

1. **Response-type roughness measuring systems:** uses the measured vehicle suspension dynamics to correlate vehicle motion and road profile. As suspension

systems characteristics change with wear and tear, there is no simple correction that can be applied to correct measurements. Instead, the entire road-meter-vehicle system must be recalibrated. Thus, rigorous calibration and testing is required as a part of normal maintenance procedures when using these sensors onboard.

2. **Direct profile measurement systems:** make physical measurements with dedicated contact or non-contact devices called profilometers. Profile is measured as the basis for direct computation with high-speed profilometers. Compared to the former technique, this offers minimal re-calibration and higher quality measurements. The downside to this technology being, the cost and complexity of high-speed profilometers.

From everyday experience, one can draw a conclusion that a majority of human drivers tend to drive at lower speeds on rough roads. Thus, measuring road roughness can help a look ahead system associate low speeds at a given location to traffic versus bad road conditions. Determinations of this kind can be performed onboard and passed on to a central server which could then broadcast the same to other vehicles passing through the given road segment. This would help predictors better estimate upcoming effective speed limits and future vehicle velocity.

Slippery Road Detection

A multi sensor approach is generally used for slippery road detection [59]. Exterior vehicle mounted sensors are used to measure pavement temperature, air temperature and dew point. These measurements are then fused with on-board data like engine and vehicle speed, gas pedal position, steering wheel angle etc., to determine changes in road friction. For Example, road slip determination, can be performed by monitoring rotational rates of driven versus un-driven wheels. These measurements are then used to determine rotational differential over time and thus predict relative friction. Information about slippery roads can be used by powertrain controllers and look ahead predictors in a manner like road roughness measurements. Slippery roads make drivers exercise more caution while driving and hence result in traffic moving at a slower pace than usual. This reduces a roads effective speed limits. Such effects can be predicted ahead of time by monitoring friction conditions on the road.

Road Temperature Sensors

Temperature sensors are used to detect possible freezing temperatures on the pavement. Infrared sensors can be used for measuring pavement temperature, and measures air temperature. They are mounted outside the vehicle and continuously monitor road surface temperatures. An early warning system using pavement temperature sensors alerts the driver when the road surface temperature dips below 37 degrees. It is currently used at Airports to Detect possible icing conditions on taxiways and by state and Municipal transportation boards to detect road surface temperature for applying salt brine or other anti-icing techniques. However, it is proposed that these sensors can be used When dealing with possible icing conditions as drivers generally drive at speeds lower than the speed limit. This information can be used by LEMS system in future velocity prediction. The sensed data can also be communicated via such that other vehicles make appropriate changes in control for efficiency and safety.

Ice Measurement

Non-Contact Onboard Detection of liquid water and ice on the road surface suitable for on-board applications. The system is based on the different optical responses of water and ice to radiation of near infrared wavelengths. The presence of water on the surface of the road and the thickness of the water layer affects the reflection coefficient hence the state of the dry/wet state of the road can be identified by Illuminating the road ahead with near infrared light and measuring the reflected light [60, 61]. Presence of ice on the road also causes a behavioral change in drivers who often do milder accelerations and drive at lower speeds compared to regular scenarios. This expected change in behavior can be utilized by the LEMS system to make effective predictions regarding future driving scenarios. If icing information is broadcast by a vehicle is available to another vehicle upstream, look ahead forecasting can account for the upcoming expected variation in the vehicle behavior.

2.3.6 Ranging Sensors

They have typically been used for safety oriented functions, however recent literature [62, 63, 64, 65, 66] indicate evolution of these sensors to encompass their utility in powertrain control. Ranging sensors can be commonly classified into 3 categories: RADAR, LiDAR, Ultrasonics All three are active sensors. They transmit energy signals in a specific pattern and study the returned signals to interpret the measured quantity. Ranging sensors used commonly in todays automotive applications typically use time-of-flight and/or frequency phase shift to measure the distance and speed of a tracked objects [67].

RADAR

Millimeter-Wave Radar (MMWR) are robust to weather and are popularly used for automotive applications [67]. MMWR can be further classified pulse radar and continuous wave radar. Pulse radars are complex in implementation and hence not commonly used in automotive applications [68]. Frequency of RADARS used in automotive applications are determined by legislation and typically operate at 77Ghz or 94 Ghz [7]. Though most RADAR systems in production today seem to have a good range resolution, measuring the angular position of an object with respect to the sensor continues to be a challenge. Most RADARS today use either 'sequential lobing [69] or 'monopulse [70] techniques for electronic measurement of an objects detected position. However, these techniques lack angular resolution and thus rely on tracking and filtering to resolve different objects when more than one objects have same range but different positions[7]. This implies that to use these sensors in traffic detection, advanced algorithms need to be developed that can count traffic objects and track their speeds to judge average traffic parameters. Owing to higher frequencies, MMWR can reduce interference from unwanted reflections and yield stable. That is they are not disturbed by shape, color or texture of targets surface detections [68]. MMWR are prone to interference from other radar equipments and need to be provided with appropriate physical and software based protection which has significant impact on the systems cost [71].

RADAR signals attenuate with distance owing to atmospheric absorption[72]. Like other electromagnetic wave based devices, RADARs are affected by environmental conditions like rain and fog that introduce reflective particles along the radar beams path of travel [72]. Water droplets scatters radar waves and hence further attenuates the signal strength while reflecting some waves back to the receiver. This results in two erroneous effects it reduces the operational range of the RADAR and created a back scatter that obscures the actual obstacle [7]. Though these environmental factors decreases the received signal to noise ration and lead to signal attenuation, it is not strong enough to compromise operation of the sensor. Literature [73, 74] discusses various processing techniques to improve RADARs performance under rain and other noisy environmental conditions.

LiDAR

Light Detection and Ranging (LiDAR) sensors use laser light to measure distance to an object. LiDAR sensors can be broadly classified into 2 types: Scanning LiDAR and Multibeam LiDAR. Multibeam LiDAR employ an array of transmitting elements that illuminate different angular sections of the environment and a corresponding array of receiving elements. On the contrary, scanning LiDARS typically use one transmitter and receiver. Light beams are mechanically rotated to scan various angular sections of the environment. The rotation is achieved by using a rotating mirror or prism system. In both cases, distance to an object is determined by measuring the difference in time between the transmitted and received pulses [75].

Owing to wave properties, LiDARs have small divergence angle, better resolution and sensitivity in comparison to radars range [7, **?**]. However, automotive LiDARS use incoherent light detection. Which implies that, velocity measurement using the difference in frequency between transmitted and reflected wave (Doppler effect) is not possible as in the case of RADARs. Thus, velocity measurements in LiDARs are performed by tracking movement of detected points over time[75].

With most modern LiDAR systems operating near peak efficiencies, the measurement range is influenced only by lens aperture and size of the sensor[75]. Automotive LiDARS operate in the near infrared wavelength of lasers. Since these sensors employ laser lights to perform distance measurements, the reliability of these measurements can be affected by environmental factors like snow, rain and fog. Additionally, such external conditions can also impact a LiDARs maximum range [7, **?**]. Water sprays from other vehicles introduces ghost targets and influence the sensor performance. Infrared light is not permeable to dust and mud hence deposition of the same on LiDAR sensors over the course of field use can affect their performance.

Ultrasonic Sensors

Ultrasonic use mechanical waves sound waves. Hence are considered to be different from RADARs and LIDARs that use electromagnetic waves as the measuring medium. The ultrasonic ranging unit consist of a transmitter, a receiver and a processing unit [76]. Ultrasonic wave pulses of 20 kHz or higher are emitted from the transmitter. Time gap between the transmitted and reflected pulses of the ultrasonic is used to measure the distance to an object. Ultrasonic waves good penetration through rain, snow and fog and hence measurements from these sensors are relatively robust to weather conditions[68]. Owing to wave propagation dynamics associated with ultrasonic pulses, they have poor directional characteristics and exhibit large divergence. The slower propagation speeds of sound waves render these waves highly susceptible to frequency shift from the doppler effect thus disqualifying them from use on highspeed applications [68].

Comparison of Ranging Sensors

Figure 2.3.6 provides a comparison of various ranging sensors used for ranging in automotive applications. Though LiDARS outperform RADARS in angular measurement, it is clearly visible that RADAR sensors are better suited for a wider range of applications. Ultrasonic and camera based recognition techniques exhibit significant drawbacks under unfavorable weather and/or light conditions. Though they offer better lower range measurement capability unlike RADARs. However, from the perspective of powertrain control application as explained in Section 2.3.6 very short range measurements (<2m) are not useful. Hence long range RADAR systems are most preferable to address the Look Ahead Prediction problem for an LEMS controller.

Use of Ranging Sensors in Powertrain Control

Measuring the speed dynamics of the vehicle ahead, can be useful in forecasting imminent acceleration and deceleration behavior of the host vehicle. This information can be treated as inputs to improve short term velocity prediction as derived from other sources. Prior research proves the link between short time variation in vehicle velocity and fuel economy [64, 65, 66] and quantifies the effect of predicting upcoming traffic scenarios. Along similar lines, Manzie et al. [62] demonstrated a potential fuel economy benefit of 13-35% when 50seconds of future traffic information is visible. Additionally, Verma [63] shows that short term traffic predictions aids controllers in rejecting unnecessary throttle variations and hence prevent transient losses in the combustion engine. Thus resulting in a 6% improvement in fuel economy of trucks as proved via experimentation.

2.3.7 Summary

A tabulated summary of the sensors studied above is presented in Figure 2.3.7. The table lists out the 'Use' of these sensors in look ahead prediction for power train control. The table also provides a high level overview of the information that can be obtained from the respective technologies - which is further elaborated in Figure 2.3.7. Additionally, alternate applications for the studied sensors in safety and ADAS applications are also presented. Figure 2.3.7 differentiates a sensor's pri-

	Short Range Radar	Long Range Radar	Lidar	Ultrasound	Video Camera	3D-Camera	Far IR Camera
Range Measurement < 2m	o	o	o	++	-	++	-
Range Measurement 2..30m	+	++	++	-	-	o	-
Range Measurement 30..150m	n.a.	++	+	- -	-	-	-
Angle Measurement < 10 deg	+	+	++	-	++	+	++
Angle Measurement > 30 deg	o	-	++	o	++	+	++
Angular Resolution	o	o	++	-	++	+	++
Direct Velocity Information	++	++	- -	o	- -	- -	- -
Operation in Rain	++	+	o	o	o	o	o
Operation in Fog or Snow	++	++	-	+	-	-	o
Operation if Dirt on Sensor	++	++	o	++	- -	- -	- -
Night vision	n.a.	n.a.	n.a.	n.a.	-	o	++

++ : Ideally suited / + : Good performance / o : Possible, but drawbacks to be expected;

- : Only possible with large additional effort / - - : Impossible / n.a. : Not applicable

Figure 2.9: Strength and weaknesses of automotive ranging sensors [7]

Technology	Information	Use	Safety and ADAS Application
GPS	Global position Velocity Heading	Localization Feedback	Included in maps and V2V
IMU	Acceleration – longitudinal Acceleration – lateral, Acceleration - Vertical	Localization Feedback, Speed Detecting turns Road Condition measurement	ESP, Traction Control Slip Detection
Camera	Recognize landmarks Obstruction – ahead and lateral Traffic light (visible range) Lane localization	Localization Indicator of traffic Increase regen time Driver Intent Recognition	Collision Avoidance Lane departure waning, Lane Keep Assist Pedestrian detection Speed limit and sign detection
Radar	Relative speed of the vehicle ahead Headway Obstruction – ahead and lateral	Efficient Adaptive Cruise Control Indicator of traffic Identify or predict Lane change / passing	Adaptive Cruise control Collision Avoidance Pedestrian Detection Lane Change Assist Beam Control Blind Spot Monitoring Turning Assistant
Lidar	Relative speed of the vehicle ahead Headway Obstruction – ahead and lateral Lane localization (specialized cases)	Efficient Adaptive Cruise Control Indicator of traffic Identify or predict Lane change / passing	Adaptive Cruise control Collision Avoidance Pedestrian Detection Lane Change Assist Beam Control Blind Spot Monitoring Turning Assistant
Maps - offline	Grade on road ahead Grade profile for multiple routes Routes to destination Speed limits	Road load estimation Optimal routing	Hill descent systems Adaptive Cruise Control (Speed Limit) Wrong-way driving warning
Maps - online	Routes, Weather, Wind Traffic information Construction zone	Online routing update Effective speed limit Optimal routing	Adaptive Cruise Control (based on road condition, Speed Limit) Crosswind and Weather warning
V2I/S	Traffic light broadcasts Traffic density information Traffic speed broadcast Weather and road conditions	Velocity optimization / increase regen time Indicator of traffic Ramp-up shaping	Intersection Assistant (with GPS) Turning Assistant Traffic Light Control Crosswind and Weather warning
V2V	Braking pedal signal Accelerator pedal signal Self classification	Indicator of traffic Ramp-up shaping	Collision Avoidance, Adaptive Cruise Control Lane Change Assist Beam Control Blind Spot Monitoring Turning Assistant

Figure 2.10: Summary of technology enablers for look ahead forecasting

mary measurements and those that can be derived from the primary measurement. This distinction between primary and derived measurements is significant since errors in primary measurement can be propagated to derived measurements by various mechanisms though some mechanism might transform the errors in primary measurement to noise in the derived measurements, certain other mechanisms may cause an integrative or cumulative errors that cause the derived quantities to endlessly drift from physical values. For example, when measuring speed from position estimates from GPS, the error in instantaneous velocity estimate may manifest itself as random bias. However, when integrating longitudinal and lateral acceleration from IMUs to determine distance traveled, the uncertainty propagates differently. By continuously double integrating erroneousness values the estimated value of distance traveled might continually diverge from the actual value in the absence of an external reset. The table in figure 2.3.7 also ranks the various sensors in terms of the noise associated with the measurements, their measurement's tendency to drift, short term accuracy and if they are robust to inclement weather. The measurement characteristics in the above chart is divided into 4 categories. 'Low', 'High', 'Not Applicable (N.A)' and 'Env.Dependent'. Where the last classification signifies that the noise levels expected in these measurements are highly dependent on the operating environment. For example, GPS measurements can be very noisy on cloudy days when line of sight path to the GPS satellites are obstructed by the clouds. Certain vehicle and environmental measurements like 'Traffic detection' from V2V sensors need to satisfy special conditions to be applicable, in this case V2V technology needs to have sufficient market penetration before it can be used to make traffic related estimates. Similarly, RADARs and LiDARs are traditionally pointed parallel to the ground surface to

Sensor System	Vehicle Measurement					Environment Measurements					Measurement Characteristics			
	Position	Velocity	Accel'n	Grade	Heading	Leader Velocity	Head way	Leader Classification	Traffic	Infrastructure	Noise	Drift	Short Term Accuracy	Robust for Weather
GPS +Map	**✓✓**	✓✓	✓✓	✓	✓	X	✓	X	X	X	Env. Dependent	Low	Low	x
IMU	✓	✓✓	**✓✓**	✓✓	✓✓	X	X	X	X	X	High	High	High	✓
Vision	✓	✓	✓	X	X	✓✓	**✓✓** (Stereo Camera)	✓✓	✓✓	✓✓	Env. Dependent	N.A	Env. Dependent	x
RADAR	X	✓ (downward facing)	✓ (downward facing)	X	✓ (stereo downward facing)	**✓✓**	✓	X	✓	X	Low	Low	High	✓
LiDAR	X	✓ (downward facing)	✓ (downward facing)	X	✓ (stereo downward facing)	✓✓	**✓✓**	✓✓	✓✓	X	Low	Low	High	✓
Ultrasonic	X	X	X	X	X	✓✓	**✓✓**	X	X	X	High	High	Medium (Low Range)	X
V2V	X	X	X	X	X	**✓✓**	✓	**✓✓**	✓✓ (high penetration)	X	Low	N.A	High	✓
V2I	X	X	X	**✓✓**	X	X	X	X	**✓✓**	**✓✓**	Low	N.A	Low	✓
V2S	X	X	X	**✓✓**	X	X	X	X	**✓✓**	**✓✓**	Low	N.A	Low	✓

✓✓ - Primary Measurement
✓✓ - Capable of Individually achieving
✓ - Better accuracy needs co-operation with other technologies
X - Not Achievable

Figure 2.11: Benefits of various technology enablers for look ahead forecasting

monitor the vehicle's surroundings. However, to enable ground speed detection for precision dead reckoning, they need to be pointed downwards.

2.4 Overview of Sensor Fusion for Look Ahead Control and Predictions

Sensor fusion is necessary to address the challenge of fusing data coming out of several sensors in real-time to aid in accurate determination of a vehicle's environment. The notion behind utilizing different types of enablers to measure the same quantity was to increase the redundancy of data and hence arrive at a better estimate of the measured quantity. Sensor conflicts are caused by increasing complexity of sensor outputs, measurements of different reliabilities and varying update frequencies . This further underlines the need for a robust sensor fusion algorithm.This section aims to give a high level overview of various sensor fusion algorithms and understand how various sensors can be fused together to get a good overall understanding of the drive scenario.

Sensor fusion can take place at various levels. The type of sensor fusion that needs to be adopted is application and sensor depended. Some common basic topologies of fusion are itemized below [77, 78] :

- **Sensor Layer Data Fusion -** Each sensor has an independent processing unit which performs feature extraction, target classification and tracking. Ultimately providing object information to a central processing unit(CPU). Since each sensor or (sensor unit/cluster) is equipped a processor, the processing load on the central unit is relieved but over all system cost increases. This architecture is scalable and robust to hardware change as primary structure and data fusion arithmetic remain unchanged

- **Center Layer Data Fusion -**Each sensor has an independent processing unit which performs basic tasks like filtering, magnification and reorganization. But unlink 'Sensor Layer Data Fusion' - feature extraction, target classification and tracking is performed in a central processor. Since sensor layer processing is absent (as in 'Sensor Layer Data Fusion') the data processor experiences additional computation load. However it yields a simpler structure with better reliability and lower cost. Allowing optimizing the CPU processing for different sensor and enhance utilization and efficiency of information. However, flexibility of the system to hardware changes and scalability are compromised.

- **Mixed Data Fusion** is a hybrid of sensor and central level data fusion. Here, data from sensors can be provided to the center processing unit directly or by each sensor-processor. Therefore the architecture is very adaptable but complex [78].

- **Distributed multi-sensor data:** In this case, the CPU is not directly connected with sensors but rather connected to 'local data fusion units which fuses native data from each sensor set. Data can he fused on different layers and in different fashions based on objects and requirements. Each layer has absolute fusion result that is a part of entire fusion. This kind of structure preserves the efficiency of centralized data fusion. However this technique demands a some extra computation to orient the sensor data and results of local data fusion center[79].

- **Decentralized Data Fusion-** In this technique, no global or local fusion units exist. Hence very different from the above 4 topologies. Each sensor output connects with other sensors via nodes, the inputs to which can be sensor data or data fusion results from other nodes. The nodes output current result of local data fusion. Due to absence of a centralized data fusion unit, the output of the system can be obtained from any node. In the structure a node can connect with all other nodes or part of them. Communication and data fusion computation is relatively complicated. However, satisfactory performance can be obtained from a set of relatively lower powered cores [80, 81].

- **Three-Level Layer Data** is a parallel structure constituted with signal layer fusion, reasoning layer fusion and dynamics layer fusion. Sensor integration and data fusion progress via correlative and learning mode in signal layer. It is usually used in elicitation, correlation and training network (such as neural networks) to realize integration of sensor without mathematic model of process and phenomenon. The trailing process cannot be quantified. In reasoning layer statistic model is usually required. The results are then fed to the CPU that ultimately controls the system. Despite its simple architecture, the structure is complicated as function of each layer is not well defined and their relation is not very compact making it difficult to apply in practical applications[80, 80].

Conventionally data fusion algorithms can be divided into 3 levels : Estimation methods (or Low Level Fusion) , Classification methods (or Medium Level Fusion) and Inference methods (or High Level Fusion) [79]. Estimation methods are used primarily in preliminary estimation of low level incoming signals. Classification methods are employed to classifying the extracted features. Whereas Inference methods are

Methods		Advantages	Limitations
Estimation Methods	Kalman Filter	• High computational efficiency. • Easy to implement.	• Restricted to linear and Gaussian assumptions. • Lower accuracy and additional phase delays in distributed tracking problems.
	Extended Kalman Filler	• Computationally efficient. • Intuitive and easy to use. • Stable in practical estimation problems	• Non-Gaussian noise processes are not al IO\ved . • System and measurement models need to be differentiable
	Covariance Intersection	• Can fuse independent sensory data with unknown correlation that KF is awkward with. • Can fuse any pair of Gaussian probability density functions	• Constrained to be Linear form of local sensor estimates. • Cannot perform a partial update in estimates.
	Covariance Union	• Can avoid the information corruption and eliminate spurious estimations. • Tolerant of inconsistent data sources	• High computationally demanding. • Inappropriately conservative fusion result may occur.
Inference Methods	Support Vector Machine	• Can deal with non-linear and non-monotonic data. • Can deliver a unique solution.	• With the lack of transparency of results. • Problems in the choice of the kernel function. • High complexity and extensive memory requirements in large-scale tasks.
Inference Methods	Bayesian Inference	• Can provide more intuitive and meaningful inferences. • Can provide a convenient setting for a wide range of models.	• The needed prior distributions of unknown parameters are always available. • High computational cost comes with a large number of parameters in models.
	Particle Filter	• Able to represent arbitrary density. • Dealing with nonlinear/non-Gaussian dynamic systems.	• High computational complexity • Difficult in determining the optimal number of particles. • Sample degeneracy and in1poverish1nent.
	Dempster-Shafer evidence theory	• Able to explicitly model the degree of ignorance which is calculated improperly in Bayesian methods. • Can model belief not only on elementary hypotheses but also on composite ones. • More flexible than probabilistic approaches.	• Computational complexity increases exponentially with the growing number of elements in the frame of discernment. • Conflicting beliefs management. • The assumption of independent evidences dose not always hold true.

Figure 2.12: Sensor fusion architecture for look ahead forecasting

utilized for decision-making at a high level. Table in Figure 2.4 shows a comparison between a few commonly used basic fusion algorithms that have been classified according to their fusion level.

The set of sensors considered in earlier sections of this chapter can be integrated as per the architecture proposed in Figure 2.4. The ultimate aim of such sensor fusion process is to enable velocity and power prediction for future drive scenarios. The system has both realtime and offline predictive components. Most predictions within the vehicle's immediate surroundings are predicted and updated in real-time. Realtime predicted components are those from ranging sensors, wheel speed sensors, IMUs, Vision sensors (cameras) and V2V reception information. These elements help identify the instantaneous vehicle states and variations to predicted velocities due to driver's reaction to external factors. These predictions act as short term, high frequency corrections to the a prior long term prediction. Traffic and infrastructure based forecasts of speed and power demand are predicted off-line and updated at lower frequencies; these are called long term predictions. This concept of classifying predictions into short and longterm are discussed in section 2.5.1.

Owing to the various technology enablers acting at different time scales, update frequencies, precisions, and confidences - a sensor fusion algorithm is essential to make look ahead predictions while drawing from as many diverse sources as possible. IMU sensors are used to measure vehicle acceleration and yaw rates. Using the INS's kinematic equations, these measurements can be converted into velocity, altitude and displacement estimates. Sensor characterization studies for a specific sensor can be performed on the IMU sensors that yield sensor biases that be used as an other input into the INS equations to improve accuracy of IMU estimates. This technique of (displacement) position and velocity estimation can be used in conjunction with GPS systems for dead reckoning or to correct GPS readings in urban canyons. GPS acts in tandem with map databases to furnish grade and infrastructure related measurements. Infrastructure measurements include location of traffic lights, and stop signs along the route and speed limits of roads that are part of a given route. Remote sensing of a vehicle states like speed, acceleration and velocity is also possible through GPS sensors. Though vehicle state measurements are not very useful for on-board look ahead - they can be used by a remote server based system connected by some wireless medium to estimate on road conditions. Such estimates can then be used by the other vehicles upstream. Ranging sensors like RADAR and LiDAR offer the capability to measure distance and relative velocity of other vehicles on the road and help in creating a look ahead prediction thats valid over the next few seconds. In order to effectively calculate the speed of other vehicles, the host vehicle's speed needs to be known and hence wheel speed sensors are also included in the full sensor suite. Once the speed of vehicles surrounding the host is determined, it can serve as an estimate of the average traffic flow speed on the given road segment. The difference between the road speed limit and average traffic flow speed can further be used an indicator to traffic conditions on the road and communicated to a remote server or other vehicles upstream. Image sensors when used in union with appropriate image processing algorithms can be used to generate ranging data but also classify the vehicles on the road. It can also be used to read road signs and other infrastructure related information in real-time that may be used by look ahead controller to make changes to predicted estimates. For example, speed reduction due to an unexpected speed limit variation due to construction zones can be accounted for by image sensors.

Given the inputs available to the fusion algorithm, it undertakes a operations like time synchronization, coordinate transformation etc. before applying more complicated intelligence in analyzing the measured quantities to yield reliable look ahead information. It is to be borne in mind that Figure 2.4 describes a generic sensor suite that includes all the sensors studied. The following section 2.5 describes the process of incrementally adding new technology enablers to a vehicle. Such that the benefits associated with each in terms of look ahead control can be studied.

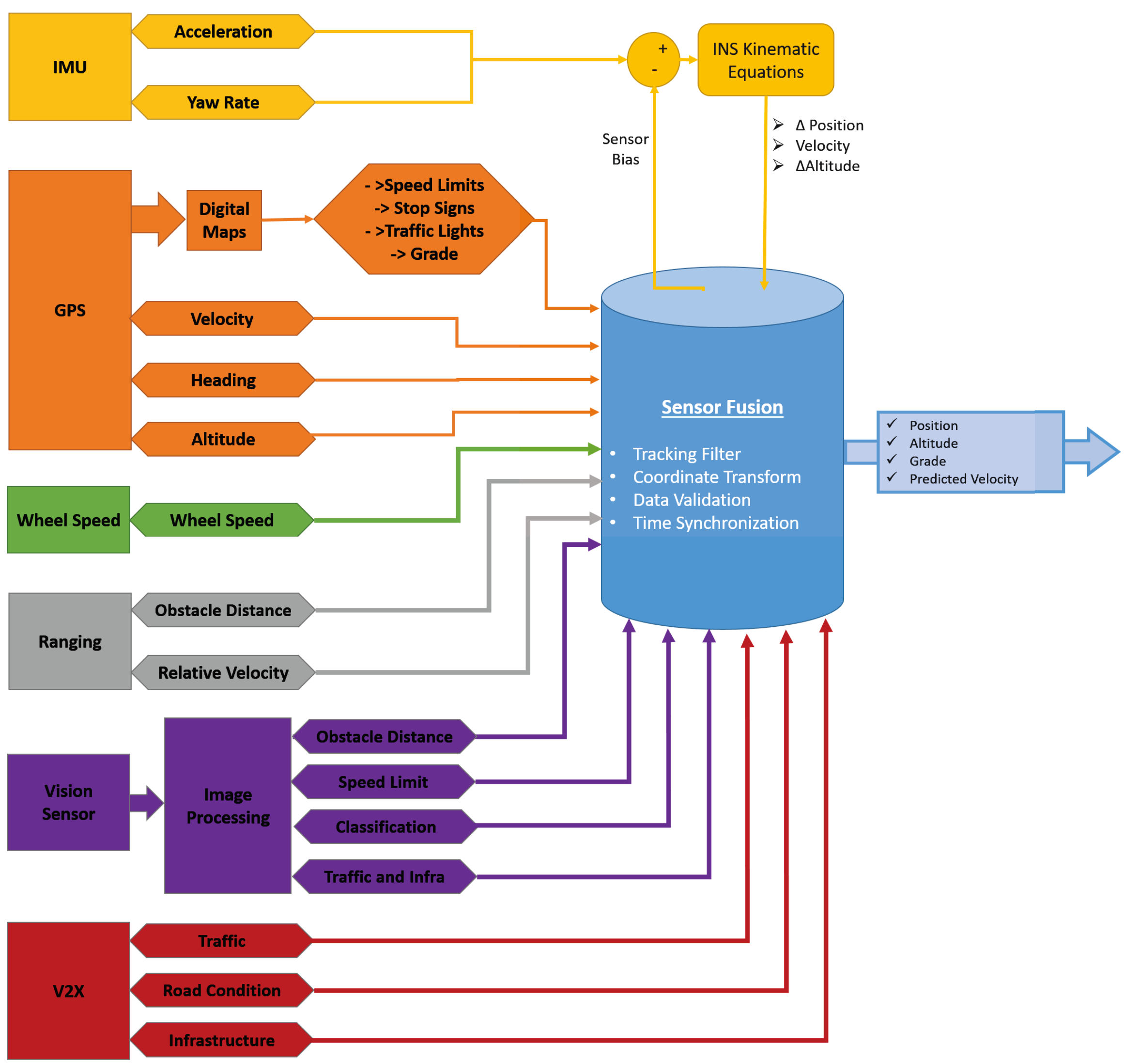

Figure 2.13: Sensor fusion architecture for look ahead forecasting

2.5 Creation of Sensor Suites

In order to decide which sensors get the precedent over the others in terms of being included first and which ones get added later, a subjective evaluation gleaned from studying these sensors, as presented in Section 2.3 is used. The subsequent parts of this section introduces 'Technology Loading', and classification of various sensors and information sources.

2.5.1 Classification of Technology Enablers to Capture External Factors

The above Subsection 2.2, distinguishes the types of external factors based on their range of detection and rate of change involved. This classification is unconventional and arrived upon from the view point of a look ahead predictor (or powertrain controller). 'Technology Enablers' that facilitate collection of related data can also be classified on the same basis, corresponding to the type of information that they yield.

From the standpoint of a look ahead powertrain control and look ahead predictions, the sensors described in 2.3 are used in some combination to obtain a forecast of the future power requirements. This is influenced by two major factors- grade and velocity. Grades data is static and can either be obtained form maps or estimated from GPS or IMU. Velocity data on the other hand is predicted in two steps - long term and short term forecasts .Long term forecasts are in the form of effective speed limits as predicted based on on-road conditions. They have a slow time varying (long distance varying) dynamics in their variation. For example, potential vehicle slow down caused 5miles down the road due to construction. Short term forecasts on the other hand are a estimation of the reactive variations in powertrain operation due to a perceived change in the vehicle's immediate surroundings i.e., they forecast potential changes in operation of the powertrain that are reactive to environmental conditions.

The following figure gives a graphical illustration of the same. In Figure 2.5.1, the blue truck represents the 'Host Vehicle' and the other vehicles are representative of traffic around it. The orange symbols correspond to vehicles in its immediate vicinity and the green ones represents traffic downstream along the vehicle's route.

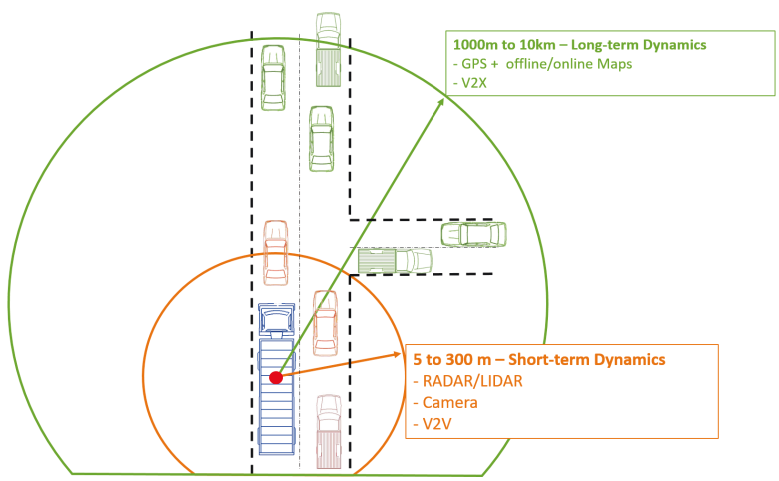

Figure 2.14: Graphical illustration of short term versus long term dynamics in external factors and the sensors used to detect them

Technology Enablers such as the RADARs and Cameras have an fixed observation range up-to which they can sense external factors. This is determined by the range of individual sensors. From Section 2.3 we know that most such sensors have a dependable range of about 300 meters. Hence such technologies are said to be adept at capturing short term dynamics in vehicle speed and power. On the contrary, over the air information sources like V2I communication and or traffic information from a server (online maps) the 'host vehicle' retrieves information on external factors that might affect its behavior thousands of meters later. Thus these kinds of data sources are considered to yield information that potentially affect 'host vehicle's' long term velocity profile.

2.5.2 Preferential Technology Loading

Technology Loading refers to incrementally appending sensors and other technology enablers to an onboad look ahead system to guide the prioritization of technology. This considers the market readiness of each system and how they improve look ahead performance. For example, technologies that yield look ahead forecasts that are more reliable, but are not widely used in the market is given a lower priority in being included in a sensor package, compared to those technologies that yield less reliable information but are commonly used. For example, I2V communication from a traffic light enroute can predict an upcoming stop event accurately compared to forecasting stops using onboard RADAR based on detecting a stopped vehicle. Successful I2V communication in this case, needs governments to install I2V transponders on traffic lights and hence the accessibility to such data is limited as of now. RADARs on the other hand are more widely used, are standalone onboard technology and does not depend on the sensed element (traffic light in this case) being appropriately instrumented. Hence RADARs are more preferred and get a higher priority in terms of technology loading. The process of technology loading was carried out for all sensors studied in Section 2.3. The result of this activity is depicted graphically in figure 2.5.2. There each successive step represents addition of new information to the existing data set to improve look ahead system's performance.

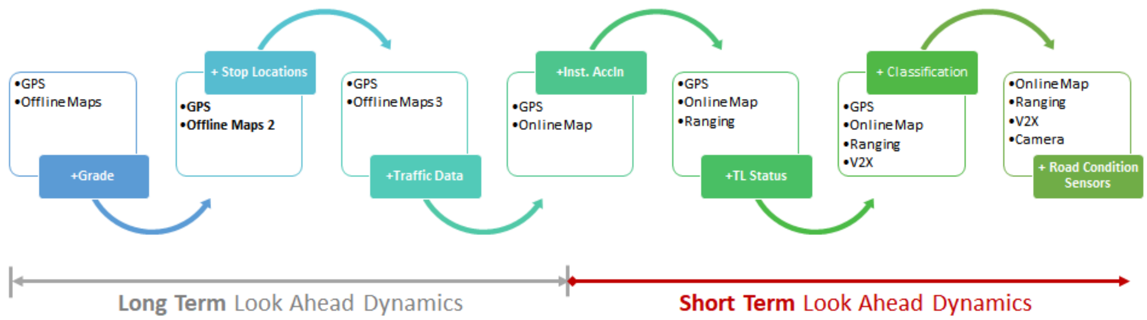

Figure 2.15: Graphical illustration of technology loading

2.5.3 Information Package Creation

Using the above philosophy of perceiving the technology enablers as information sources, six technology packages have been proposed. Each package introduces a new type of information though addition of additional sensors. In real life implementation, this would mean each package is not only more expensive than the previous, but the quality of prediction also increases. The most basic form of look ahead information are Maps coupled with GPS information. This is readily available in today's world and there exists various consumer products like Google Maps and Here Maps, that integrate this information, and aid in self localization. Offline maps also offer information on road speed limits, number of lanes on a road and presence of intersections. This forms the most rudimentary form of look ahead and helps in localizing the host vehicle along its route. This is the first proposed package. Further this information can be improved with knowledge of stop signs and traffic light locations. Grade information can also be obtained from most mapping services, though this does not directly affect velocity predictions from the look ahead system, it would significantly affect the calculated power trajectory. Hence yielding the third package in the illustrated list. Traffic conditions along the route also have a profound impact on the vehicle efficiency and such information can often be gained from online maps that connect to central servers, to retrieve associated data like average moving speed of the traffic (effective speed limits). This forms the fourth sensor suite. The implementation of these sensor suites are relatively easy and related technologies are already integrated into many modern vehicles as a part of the infotainment system. These basic technologies yield data concerning the long term dynamics of the system. Since all above packages use Map based data, they yield results concerning the Long term forecasts that vary slowly. As we look into addition of more information from short term predictions, the most commonly used technology are ranging sensors like RADARs and LiDARs, followed by V2X communication, this results in the fifth and sixth sensor combinations. The reason for prioritizing radars over V2X systems has been explained earlier in this section as an example. Additionally, camera based systems can also be used in advances systems to detect deviation from expected behavior like change in road speed limits due to construction activities, classification of the leader vehicle to better estimate its acceleration behavior, detect traffic light states etc. Finally, road condition sensors like those described in 2.3.5 can be used to make better predictions of future velocity and power. Despite the concept of using these sensors is fairly simple, they have been given low priority as automotive applications for these sensors are not well established and very few case studies exist on their performance aboard a vehicle.

2.5.4 Creating Technology Suites

The previous Subsection 2.5.3 discussed various 'Information Packages', where each package delivered a specific set of data to the look ahead controller. Nevertheless, the same types of information can be achieved from different physical sensors. For example, headway distance can be measured from RADAR sensors or using a stereo camera based system or calculated using known GPS coordinates of the two vehicles (lead and follow vehicles) where the information is shared via V2V communication. Each of these measurements would carry a different degree of uncertainties and can be quantified by experimental studies or model based predictions.

From the perspective of look ahead predictors, the source of data is not as important as the measured data itself. Thought the dependability of the predictions may be affected, the predictions themselves are expected to be similar with a given selection of external factor's measurements as inputs. Thus the physical technology and the measured quantity are dissociated. In other words, we group the sensor packages by the information provided and not by the sensors used. The reasoning behind this approach being, look ahead predictors consider various data points and their corresponding confidence estimates when making predictions. They do not consider the actual physical technology that is used to derive the predictions. By dissociating the measured quantity from the sensors themselves, creating information packages can be simplified as it allows disregarding the uncertainties introduced by different sensors and their combinations. This would allow future implementers of look ahead control take sensor suite decisions based on various factors including market forces and technology developments. For example, sensor package option 5 includes information on headway distance and leader speed. However the same information can be obtained from a radar or a lidar or stereo camera. Given the nature and range of the measurement involved, Section 2.3.7 suggests using a RADAR sensor. However, if a design decision is made based on business forces or other secondary constrains, to use Camera or LiDAR based systems, the information package utilized remains the same thought the physical sensors are different. This variation in technology can be accounted for by using error corrections.

It is important to note that this point that though the physical implementation of each of these Utility Packages may be achieved through multiple technological routes, from the simulation and modeling perspective they can be condensed into the external factor that they measure. This data on the measured 'External Factor' is fed into the Look Ahead predictor and hence a prediction of future drive scenarios is arrived upon. Figure 2.5.4 discusses the various information packages that have been formed by following the technology loading process. These packages are termed 'Utilities'. The first package (Utility 1) uses an offline map in conjunction with GPS. This corresponds to the most basic form of look ahead and yields data regarding trip progress and speed limits on the roads. However, for purposes of simulation, having speed limit information can emulate the presence of this package on the simulated vehicle. IMU can be optionally added to the system to help in dead reckoning when the GPS system experiences a loss in connectivity. Similarly, in Utility 2 we add grade information to this. Utility 3 uses a more advanced map that contains information on location of stop signs and traffic lights. Such data help predicting acceleration, deceleration and stop events in addition to obtaining speed limit information and monitoring trip progress. These locations can be precomputed for a given route and used for velocity prediction. Starting from Utility 4, the speed limit information is replaced by ‘effective speed limits’. Effective speed limits corresponds to average speed of traffic on a given road segment. This data is obtained by moving to online maps that use crowd sourced data. Traffic information in this case is communicated by some wireless medium from a central server that hosts the online map. Utility 5 includes a ranging sensor that yields information on the speed of a leader and the headway between the host vehicle and the leader, and the V2X communication delivers a data driven historic velocity profile.

2.5.5 Creating A Framework for Virtually Simulating Sensor Inputs

‘Traffic in Loop’ simulation was a framework that was build to facilitate control development process. with virtual sensors and technology packages. It helps emulate technology measurements from a‘virtual’ host vehicle’s perspective as it interacts with a real-world like virtual environment. The next Chapter 3 talks in detail about this effort.

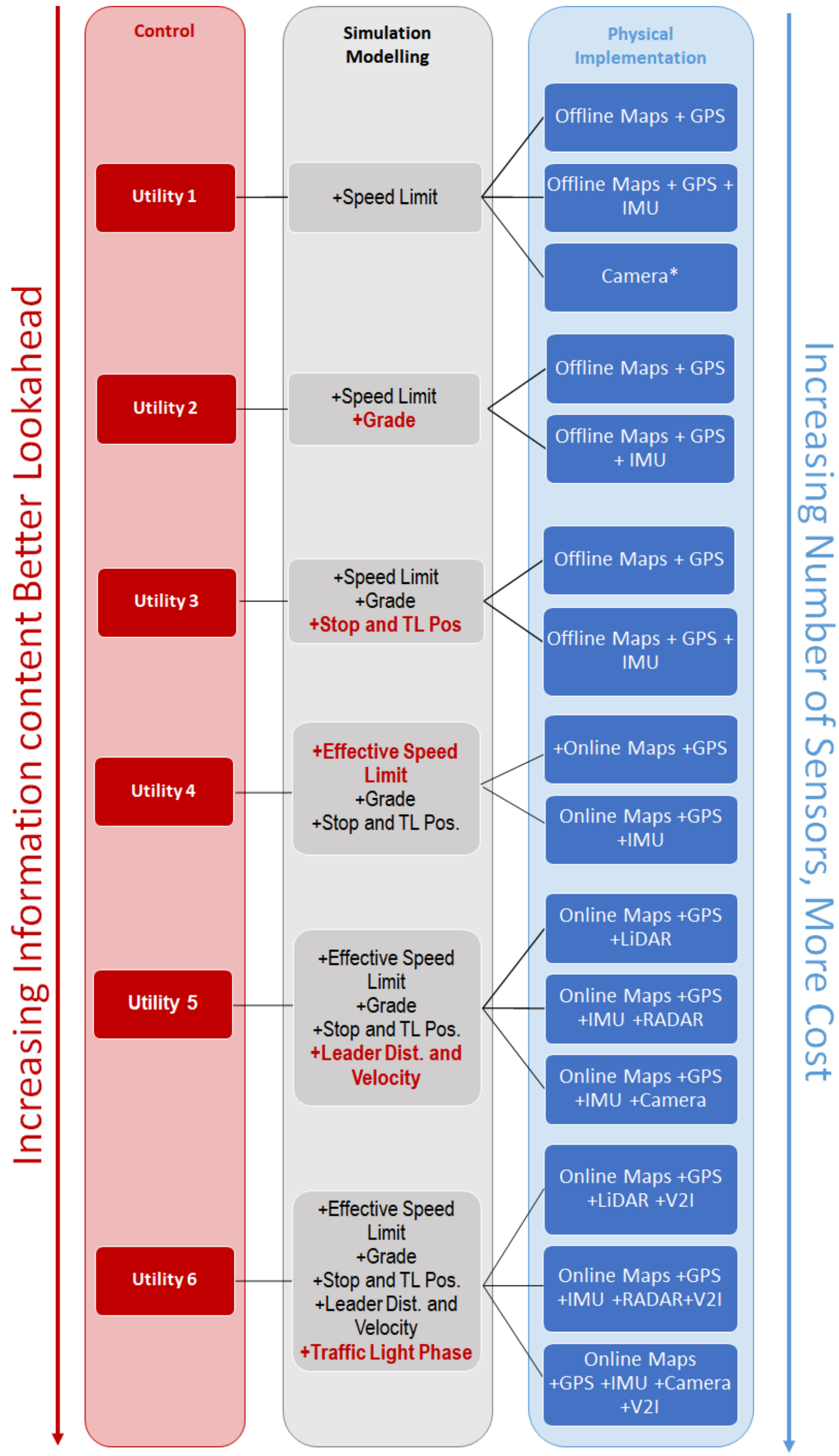

Figure 2.16: Proposed technology suites

Chapter 3: Introduction and Development of Traffic In Loop Simulation

This chapter details the 'Traffic in Loop' simulator and its framework. The simulation technique is termed 'Traffic-In-Loop' (TIL) as it shows how a vehicle and its controller responds, to realistic virtual stimuli. TIL is similar to existing concepts like Model-In-Loop (MIL) and Hardware-In-Loop (HIL) simulation. In this case a Simulink model is used as a mathematical representation of the vehicle plant, and a traffic simulation software called SUMO (Simulation of Urban MObility) is used as an environmental proxy to generate virtual roads and infrastructure networks. TIL simulation aims to integrate real world factors such as causality, vehicle-infrastructure interaction, and traffic behavior into a Simulink based powertrain simulator; to capture the dynamics that are associated with real-world systems. It helps users study the effect of factors external to the powertrain, on the vehicle‘s overall performance (like battery throughput, engine operating points etc.) and energy efficiency. The proposed framework is capable of emulating on-board sensors, V2X communication and capturing non-causal behavior of the real-world scenarios. Owing to these capabilities, the driving scenarios hence generated are more realistic; replete with sensor measurement information. A sample of the results is presented in Section 3.6.

3.1 Motivation

The motivating objective behind the creation of this system is the realization that vehicle fuel economy extends beyond the realm of powertrains and is influenced by many factors that are external to the vehicle. These factors are also known as 'environmental factors' and encompass the broader environment outside the vehicle [10, 11, 12, 13, 14]. However, isolating their effects have been difficult due to their seemingly random behavior and non-causal nature. As detailed in Section 3.1.1, given the inability of drive cycle based powertrain simulation techniques to fully represent effects related to vehicle interaction with infrastructure, traffic, and its non-causal approximation of driving profiles. A need was felt to develop a simulation platform that allows for evaluating a powertrain under realistic traffic scenarios. The TIL framework accomplishes this by creating a realistic driving environment using traffic simulators that are integrated to the powertrain simulation in Simulink. It marks a departure from the popular cycle-based simulation for powertrain analysis and provides a methodology to analyses vehicle systems, controls or components from a transportation system's perspective. Additionally, the framework also offers an open opportunity to develop, calibrate and validate controllers in lifelike drive scenarios, thus supporting in accelerating the design process for intelligent powertrain and safety controllers.

3.1.1 Shortfalls of Drive Cycle Based Analysis

Chapter 2 explores the various sensors and technologies that can be used to glean an understanding of the environmental factors that are external to the powertrain, but affect the vehicle's future velocity and power profiles. The unpredictable nature of these factors is due to the transportation system's overall interdependency with various elements. For example, the velocity profile of a vehicle is affected due to its interactions with traffic lights, behavior of other vehicles, road profiles, human-driver variations etc. Standard drive cycles fail to take into consideration such random events as they subject the vehicle system to a temporally fixed set of inputs every trip. This has a profound impact on the results of such tests as summarized below:

Driver Induced Randomness of Real-world Systems

Literature [15, 16, 17, 18, 19, 14, 22, 23] records up to 35% variation in fuel economy owing to driver behavior during on-road driving. This is the top contributing factor that leads to variation of on-road economy from estimates. The second and third major factors are, interaction of vehicles with other members on the road (traffic), and interaction with infrastructure elements (e.g. stop signs, speed limits, traffic lights etc.). However, by experience it is worth noting that a human driver's behavior in real life scenarios, is influenced by the actions of other drivers (social interactions) around him and presence of infrastructure elements along the route. Hence these 3 major factors can be considered as being connected at some deeper level.

Such social and infrastructure related interactions on driver behavior create a non-causal wave of events that ultimately affect driving patterns and hence the fuel economy of a vehicle. Drive cycle based simulation fails to capture variation in trip parameters like trip-time and fuel economy caused due to such random events. TIL setup allows users to break out of test cycle based simulation techniques and allows studying trip parameters as a statistical distribution. The accuracy of on-road fuel economy estimations is only constrained by fidelity of powertrain models and calibration of the traffic simulation.

Causality of Real-world Systems

The real world is non-causal in a sense that an input at time 't' will also affect future input at time '$t + n$'. For example, increase in throttle opening to accelerate before a traffic light may lead to a necessary application of brake a few meters ahead of a traffic light that is 'red'. Alternatively, if a steady input was maintained ahead of the same traffic light, the future braking event may have been avoided- given the vehicle speed is such that it yields sufficient time for the lights to turn 'green'.

In real life, non-causal effect can be introduced by driver behavior, powertrain limitations, traffic and vehicle's interaction with the infrastructure. An example of powertrain limitation induced causality is when, vehicle acceleration or speed is artificially limited due de-rated operation of the motor on an electric vehicle. Control actions like adaptive cruise controller or speed limitations caused by intelligent vehicle controls can also induce dynamic effects that fail to get captured when using conventional drive cycle based powertrain simulation. However, the TIL framework offers the capability to capture and quantify such behavior as shown in Figure 3.6.

3.2 Contribution

Given the inability of current simulation techniques to fully represent effects related to vehicle interaction with infrastructure and other vehicles and its non-causal approximation of driving profiles. A strong need was felt to develop a simulation framework where the vehicle drives in a virtual real-world condition and is subjected to interaction with surrounding vehicles and infrastructure. This work has developed a framework that aims to integrate the real word factors such as causality, vehicle-infrastructure interaction and traffic behavior into a conventional powertrain simulator in an effort to capture the dynamics that are associated with real-world systems and study the effect of environmental factors external to the powertrain on the overall system efficiency and performance. In the proposed framework, the on-line coupling allows the adaptation of vehicles behavior during simulation runtime. Hence in a statistical sense, the frame work of the simulation that is described in this paper helps to accurately capture the mean and variance in on-road fuel economy, constrained only by the fidelity of the models used without the need for on road tests. Hence enables the capture of small economy improvements such as the 5% improvement through optimal mild hybridization shown by [82] or similar improvements demonstrated by Chen et al. [83] through active NOx control in diesel engines.

3.2.1 Traffic In Loop (TIL) Simulation

The simulation framework proposed in this thesis overcomes these drawbacks and allows for running virtual road tests of vehicles replete with sensors and control algorithms. This is achieved through the use of traffic simulations. Provided that the traffic and infrastructure elements within the environment is tuned with appropriate data, it can generate driving scenarios that are far more realistic compared to standard test cycles. Various attempts have been made towards utilizing the concept of traffic simulation in conjunction with numerical powertrain simulators for fuel economy prediction. A summary of these attempts is presented in section 3.4.1. However the TIL framework differs from these prior attempts since it forms a closed loop co-simulation where the powertrain model and the traffic simulation environment interact dynamically through an exclusive two way communication link. In this simulation architecture the environmental factors like infrastructure and vehicles that form traffic are simulated and controlled by the traffic simulation software. Whereas an external powertrain model simulates with high detail the internal and vehicle level dynamics of a host vehicle. The co-simulation is setup such that the host's powertrain dynamics affects the its behavior in the traffic simulator at every time step and vice versa.

The subsequent sections of this chapter describe various components of Traffic In Loop Simulation, including its architecture and concludes with a demonstration of results obtained from it.

3.3 Traffic Simulation

Traffic simulation is the mathematical modeling of transportation systems. These simulated environments accommodate for the presence of stop signs, traffic lights, lane changes and traffic with a spread of different vehicles; driven by drivers of various types. It also imposes varying properties for roads including, road based speed limits, lane widths and prohibited lanes. The behavior of member vehicles to the above parameters are described by discrete rules or mathematical formulations that are often differential equations [84].

Additionally, velocity profiles followed by vehicles on a traffic simulator are not predefined. Each vehicle's behavior is determined by its acceleration limits, driver behavior model, and road speed limits. Given a traffic free virtual road, vehicles in this environment accelerate to a steady speed and maintain that speed in the absence of any external disturbances like traffic or stop signs. This steady state speed for each vehicle, depends on the vehicle properties and type of driver assigned to it. Certain driver models may cause a minor oscillation of this steady state speed, about a given mean with a pseudo-random amplitude and frequency. Further, member vehicles decelerate to avoid collisions in the presence of other vehicles or when approaching traffic lights and stop signs. This steady state speed for each vehicle, depends on the type of driver assigned to it. Further, the vehicles members decelerate to avoid collisions in the presence of other vehicles or ahead of traffic lights and stop signs. Traffic simulation techniques can be classified into 3 basic types: Macroscopic, Mesoscopic and Microscopic. In this thesis, the author limits discussion to microscopic traffic simulators where each individual vehicle is treated as an independent element that obeys traffic laws and whose behavior is modeled by sub-models known as 'behavioral models', as detailed in [85]. Lieberman et.al.[84] provide a comprehensive explanation regarding types of traffic simulators and their classifications. Additionally, prior literature [86, 87, 88] also present a more detailed analysi s and study comparing the above-mentioned simulators. Given below is a short explanation of types of traffic simulations along with graphical illustrations:

Microscopic Traffic Simulation

Microscopic simulations track individual vehicle movements on a second by second or sub-second basis. In such simulations, there will be a 'warm-up' period before the system reaches a desired traffic density on its network. SUMO, the traffic simulation package used in this work falls under this category. The computation load for such simulation techniques increase with the number of vehicles being simulated.

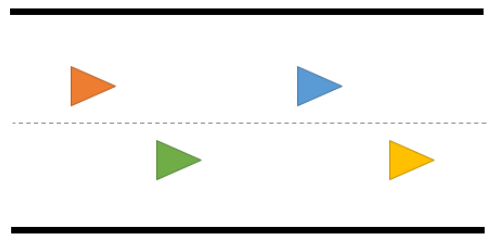

Figure 3.1: Graphical representation of a microscopic traffic simulation

Mesoscopic Traffic Simulation

When dealing with larger simulation areas or higher traffic count, microscopic simulation techniques can become slow as they are process intensive and compute the behavior of every vehicle individually. To simplify this problem, transportation elements can be analyzed in small groups. Within each group the elements are considered homogeneous. Vehicles can be grouped according to various factors but mostly based on direction of traffic flow. This is illustrated in Figure 3.2.

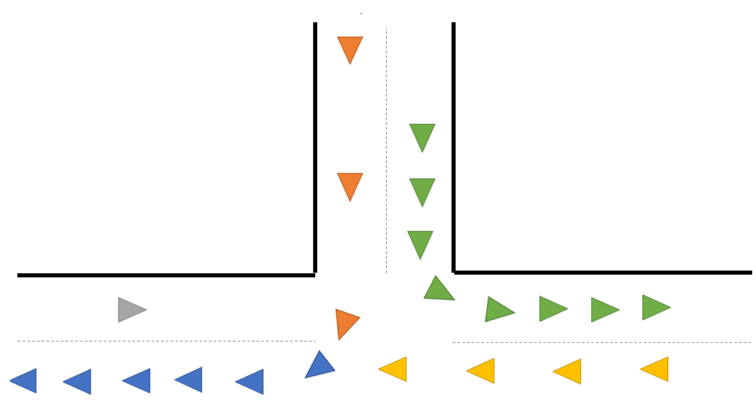

Figure 3.2: Graphical representation of a mesoscopic traffic simulation

Macroscopic Traffic Simulation

Macroscopic traffic simulations deal with aggregated characteristics of transportation elements, such as aggregated traffic flow dynamics and zonal-level travel demand analysis. Figure 3.3 shows an example of macroscopic traffic simulation, a snap shot of Google maps with predicted traffic on various roads.

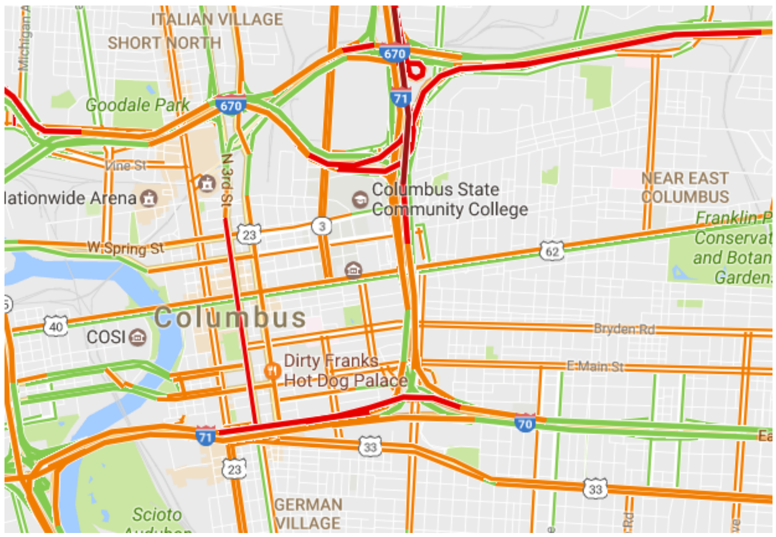

Figure 3.3: Graphical representation of a macroscopic traffic simulation

3.3.1 SUMO Traffic Simulator

SUMO ('Simulation of Urban MObility') is one of the many traffic simulators that exist today and was chosen by the authors for this work. It is an open source multi-modal microscopic traffic simulator that is available under public license (GPL) since 2001 [89]. SUMO is a space-continuous, discrete time-microscopic traffic simulation platform. It features a suite of tools that include a road network importer, algorithms for demand generation, vehicle routing utilities etc. It is capable of modeling realistic traffic conditions on a given road segment or whole cities. Here SUMO is linked to MATLAB via an intermediate layer known as TraCI that aids in integrating the two softwares by establishing an intermediate communication pathway.

TraCI

TraCI [89, 90] ('Traffic Control Interface') is an interface that facilitates online retrieval and setting attributes to the simulated objects in SUMO and available under a public license. TraCI was further adapted to enable integration into MATLAB by Acosta et. al. [91], this version of TraCI is known as TraCI4MATLAB and is available through an open source license. The online adaptability of SUMO via TraCI4MATLAB was an integral reason behind the choice of the SUMO for this activity. Section 3.5.1 discusses in finer detail about the utilization of TraCI interface to achieve co-simulation between Simulink and SUMO.

3.3.2 Structure of Input Data for Traffic Simulation

To perform a simulation in SUMO, the user needs to define three major types of information, each of which is furnished to SUMO as independent '.xml' files. This is displayed as an info-graphic in Figure 3.4. The 'Route' file defines the vehicle types (e.g. car, bus, truck etc.) the driver behavior for each of these types (following distance, aggressiveness, acceleration characteristics etc.), and vehicle's speed and acceleration limits. This file also defines the route that each vehicle in the network will follow and their start times. Hence it can be said that this file defines the dynamic components in a simulation. The second type of '.xml' file that is required is the 'Network' file. It defines the region that is being simulated and refers to a set of roads connected by intersections that permit the movement of traffic from one point to the an other. A 'Network' is akin to a map and specifies the location of Edges(roads), edge width, travel speed limits, lane widths and direction of travel. The file also specifies how a 'Node' (intersection) connects different edges and defines infrastructure elements, that control these intersections. It is to be noted that 'traffic lights' are physical objects on the SUMO map, but 'stop signs' are implemented by assigning priority numbers to various roads at an intersection. The network file also contains information on the priority numbers for different edges and the location of traffic lights. The third and last type of '.xml' file required are the 'Additional' files. City limits, locations of schools and work centers, leisure spots, population statistics, origin-destination matrices (OD Matrices), operating policies of the traffic lights etc., are defined as additional files.

3.4 Traffic Integrated Powertrain Simulation

By combining traffic simulation with a physics based powertrain model, the simulation framework proposed in this chapter overcomes drawbacks of drive cycle based powertrain simulation as detailed above. It allows users to run a virtual road test of a vehicle, in a simulated environment with various sensors and control algorithms in loop. Given the nature of vehicle behavior in traffic simulators (as described in Section 3.3), velocity profiles generated from such techniques are more realistic, provided the traffic and infrastructure elements are tuned appropriately with accurate data. Once calibrated, infinite number of drive scenarios can be obtained by varying 'seed numbers' corresponding to each simulation run- in the traffic modeling software.

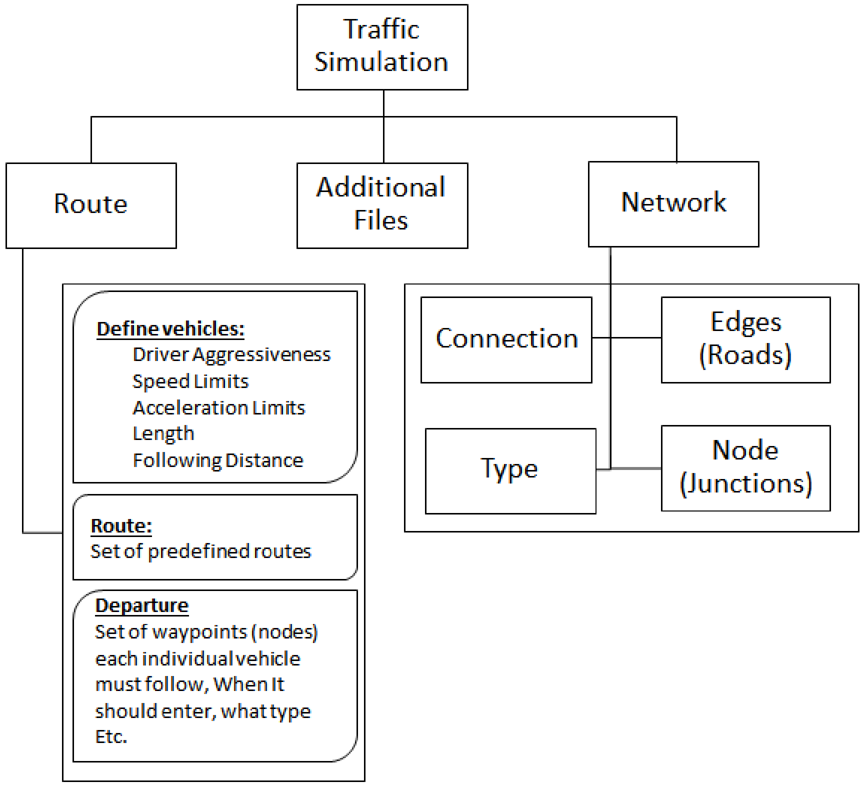

Figure 3.4: Visualization of file structure and data required for SUMO simulation

3.4.1 Literature Survey

Though the concept of traffic based powertrain analysis is a relatively recent development, various attempts have been made towards utilizing the concept of traffic simulation in conjunction with numerical powertrain simulators for various reasons. The most common of which are towards fuel economy prediction, route optimization, and design of eco-driving algorithms. Most of such efforts can be broadly classified into two categories:

1. **Open Loop Simulation:** In this simulation technique, powertrain model and the traffic model are coupled in a sequential one directional manner. An example of such implementation is described in [92] and [93].

2. **Post Processing:** In this method, velocity traces (known as drive scenarios) are generated through traffic simulation. These traces are used to then fed into a numerical powertrain simulator as static input files, or some form a curve fitting approximation is applied - . Such an approach is demonstrated by [94], [95], and [96].

The traffic simulation packages used in above cases all have a simplified vehicle point-mass dynamics model. This model can predict acceleration and deceleration behavior based on user defined limits and vehicle mass. Based on empirical data for a given vehicle type, some simulation packages can approximate a general value for CO_2 emissions and fuel consumption. However, these models are incapable of capturing detailed phenomenon like gear changes, controller action, engine operating points, accurate fuel economy prediction etc. Another major drawback associated with the above approaches is their negligence of causality. Powertrain control algorithms that affect vehicle speed or acceleration tend to have profound an impact on the vehicle's future velocity profile as demonstrated in Figure 3.6 and 3.5. The techniques described in prior art can be viewed as being a form of open loop system. Since, they are unidirectional and the powertrain model only follows commands from the traffic simulator. The powertrain's actual behavior is not communicated to the traffic simulation environment. Implying that the effect of differences between the two models are neglected. But the difference caused by the variation in model behaviors can

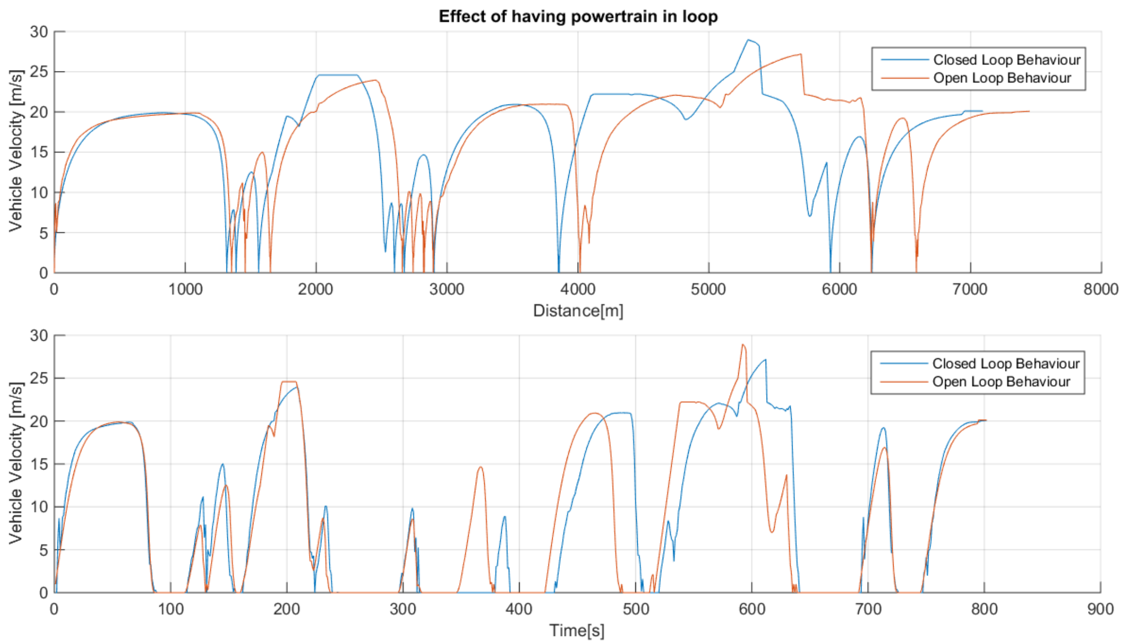

Figure 3.5: Information flow between simulink and SUMO

have a significant impact on causality of the system. This effect is illustrated with a simulation based example in Figure 3.5.

Figure 3.5 illustrates the differences in velocity profiles obtained by open loop and closed loop simulation of a host vehicle in a given map. The Closed loop results have been obtained from the TIL framework. In both cases, the vehicle begins its trip at the exact same time and location on the map. All other parameters in-terms of traffic simulation are kept constant. Upon visual observation, a clear difference between the open loop and closed loop simulation can be observed. These are captured well in the form of noticeable variation in the stop locations and number of stops between the two runs. in this case the variations can be noticed starting from after 1000m of travel. The minor variations in vehicle behavior has a butterfly effect that trickles across the cycle, further highlighting the importance of capturing causality during a drive cycle. At about 5900m from the trip's origin, it can be observed that the close loop model comes to a halt; whereas the open loop model does not. This could be attributed to the fact that owing to prior variation in velocity, the closed loop model passes a traffic light in 'green' whereas the open loop model had to wait for the traffic light to turn 'green'. Similarly, at about 6500m the opposite phenomenon is observed where the open loop model comes to a halt, but the closed loop model does not.

It can be noticed from Figure 3.5 that though the time trace of velocity is similar, they are not the same. This is caused by the differences between the longitudinal acceleration dynamics of the two models. The open loop model is controlled only by the acceleration limits. Whereas, the closed loop traffic simulation that has a Simulink powertrain model in loop is affected by a wide range of dynamics such as (but not limited to) turbo lag, torque loss due to gear change, tire slip etc. Hence, it can be concluded that the closed loop model is more realistic in its representation of a vehicle's acceleration behavior. These real-world like factors affect the vehicle's acceleration differently compared to the point mass model used in the traffic simulation packages, causing the effects of causality to build up and leading to a similar but not congruent velocity trace. It is worth noting that these effects can build up over time and differences become more apparent in longer routes.

To illustrate the effect of causality the velocity trace from four cases, each with a different upper bounds of accelerations are shown in Figure 3.6. A visual inspection of which reveals that no two traces are the same. The results for this plot have been generated by assuming that vehicle starts at the exact same origin point in every run and follows the same route to the destination. The change in acceleration limits

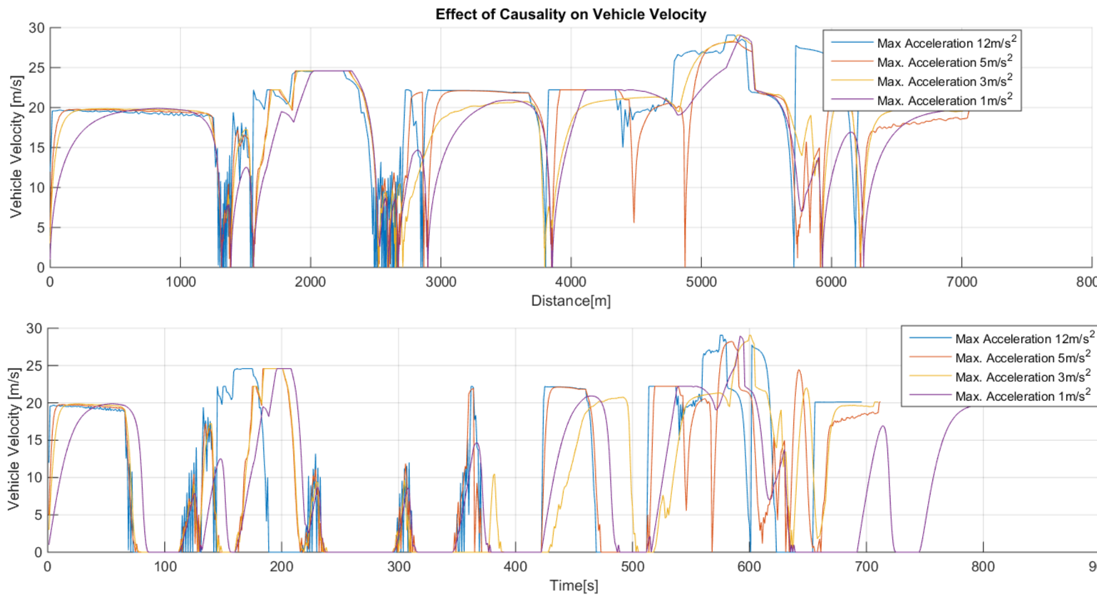

Figure 3.6: Effect of causality on vehicle velocity

affects the acceleration behavior of the vehicle. This difference propagates throughout the cycle in the form of a causal wave of events. For example, the case with $5m/s^2$ acceleration, slows down more than the others at about $4500m$, whereas the other cases do not. This could be caused by a change in the leader vehicle, i.e. if the host vehicle followed a car in other cases but end up following a slower truck in this particular case. Further this slow down causes the vehicle to come to a complete halt at about $4900m$. The reasons for this halt could range from traffic lights to a stopped vehicle on the road.

Given the above limitations, existing simulators from literature make the testing and calibration of communication or sensor depended controllers like, traffic light adaptive cruise control, active re-routing etc. difficult and cannot reveal all associated effects. First, the effect of control on practical trip parameters like idle time at traffic lights and trip time cannot be captured and the input to vehicle models remain unchanged temporally irrespective of the vehicle model's behavior. Second, these techniques discount the effect of random noise from the presence and behavior of other vehicles on the road. Third, both of the above types of frameworks only allow a one directional flow of information. Thus, it is not possible to immerse control algorithms into a simulated dynamic real-world like scenario to study control induced causality nor validate controllers that are sensitive to non-causal behavior. Finally, the effects of sensor noise, its effect on the controller and its interaction with a dynamically changing environment in a non-causal system are difficult to explore and understand with standalone powertrain or traffic simulators alone.

3.5 Traffic in Loop Simulation

To overcome the deficiencies in current practices pertaining to powertrain simulation as discussed earlier, a framework that aims to integrate the real world factors such as non-causality, vehicle-infrastructure interaction and traffic behavior into a conventional powertrain simulator is presented here. This enables the user to capture dynamics that are associated with real-world systems and study the effect of factors external to the powertrain on the overall system efficiency and performance.

In the proposed framework, the online coupling of SUMO and Simulink allows the adaptation of vehicles behavior during simulation runtime. Further, TIL helps to capture the mean and variance of on-road fuel economy without the need for physical road testing. In theory, the precision of such estimations is constrained only by the fidelity of models used. The primary goal of this co-simulation is to describe a 'host vehicle' in the SUMO environment, whose dynamics and controls are governed by a Simulink based powertrain model. As this host vehicle moves in the virtual world; a Simulink based dynamic powertrain plant model is utilized at the back-end, to compute the parameters such as, but not limited to: Host speed for next time step, instantaneous fuel consumption, engine operating points, gear shifts etc., while simultaneously exercise various control actions (e.g. acceleration limiting, torque split etc.). The speed computed by the powertrain model is then fed into SUMO as an 'command input' before every simulation step. This would make sure that the behavior of vehicle model within SUMO is realistic, and within dynamical bounds as defined in the vehicle model. This closed loop simulation allows all associated causality to be strictly maintained. The following Section 3.5.1 describes the powertrain simulator's integration with a traffic simulation package that helped achieve the goals of this work.

3.5.1 Framework and Architecture

The integration of Simulink based powertrain simulator with SUMO traffic simulation has been accomplished in a two-step manner. First, the TraCI4MATLAB interface is used to link MATLAB and SUMO in a 'server-client' fashion [90] [91]. Where, SUMO acts as the server with MATLAB acting as the client. TraCI4MATLAB uses a Java API to establish a TCPIP communication link between SUMO and MATLAB. To ensure connectivity, the authors have chosen to utilize an explicitly specified virtual port to establish this communication passage. Further, this architecture ensures that multiple vehicle models running on various different physical/virtual machines can simultaneously access the simulation parameters in a centrally operating SUMO server by connecting to the same port via TraCI. However, this thesis deals with a

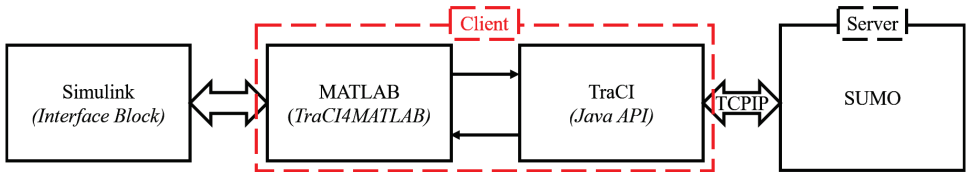

Figure 3.7: Information flow between simulink and SUMO

single vehicle model. In Figure 3.7, the dotted lines enclosing the 'TraCI' and 'MATLAB' block signify that they operate as one integral unit since, TraCI4MATLAB is implemented as MATLAB functions that reside within the MATLAB environment. The second step in the integration process is establishing a communication link between the Simulink model and TraCI4MATLAB. This is done via an interface that resides within the Simulink based powertrain model. This interface supports two way flow of information and works towards closing the loop between SUMO and the powertrain model. Hence embedding a more detailed and customizable powertrain and control model into the traffic simulation. These two steps in the integration process is illustrated graphically in Figure 3.7.

The overall flow of information during every time-step is depicted by figure 3.8 . The overall goal behind organizing the simulation framework as shown is that, given a-priori knowledge of the 'host' vehicle's speed in the next time step; the vehicle model can treat it as a 'virtual' driver speed request. This speed request ($\bar{\vartheta}_{k+1}$) is then subject to control and vehicle dynamic limitations, hence resulting in modified vehicle speed ϑ_{k+1}. Subsequently, speed ϑ_{k+1} from the vehicle model is then sent to SUMO as a speed input. This method of Simulink leading the simulation and SUMO following the speed input, ensures that the SUMO and Simulink models are always tightly coupled. However, vehicle velocity in a future time step is not available as an output from SUMO. Thus, the authors set the Intelligent Driver Model (IDM) model as the driver model for the host vehicle within SUMO environment and implemented the same equations in Simulink. φ_k carries all the required parameters from SUMO to be used by the IDM model shown above to predict the vehicle's velocity in the next time step $\bar{\vartheta}_{k+1}$. Since the model equations for the IDM is same in both SUMO and Simulink environment, the velocity prediction for step 'k+1' is accurate. This predicted velocity $\bar{\vartheta}_{k+1}$ is fed into a vehicle controller as desired velocity. Subsequently the controller applies a set of inputs, given by Γ_{k+1} to the vehicle model. Based on control and component level dynamics, the velocity ϑ_{k+1}, attained by the vehicle is computed by the vehicle model; given the latest control inputs and powertrain limits. This now forms the velocity input which is then set as an input to the SUMO vehicle via the interface. Upon receiving this input, SUMO carries out one simulation step and the whole process repeats for the next time step. Hence achieving a closed loop co-simulation with traffic in the loop. From a systems perspective, one can consider this integration process to be akin to replacing the driver in the vehicle model with the traffic simulator.

The co-simulator framework developed by the authors is depicted in Figure 3.9. This high level illustration depicts all associated components and information flows. There are 4 basic components in this simulator:

1. **SUMO and associated support structures:** Facilitate the communication between SUMO and Simulink through MATLAB.

2. **Sensor Modules:** Outputs from SUMO such as (but not limited to) distance to and speed of vehicle ahead, current phase of upcoming traffic light en-route

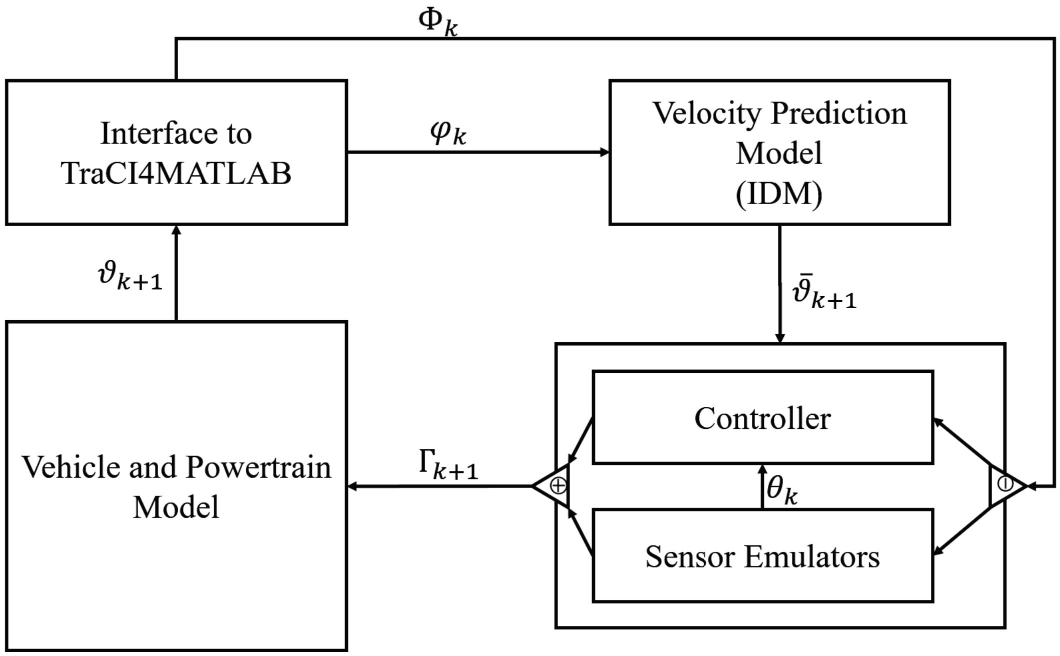

Figure 3.8: Information Flow in Developed Closed Loop Simulation Setup

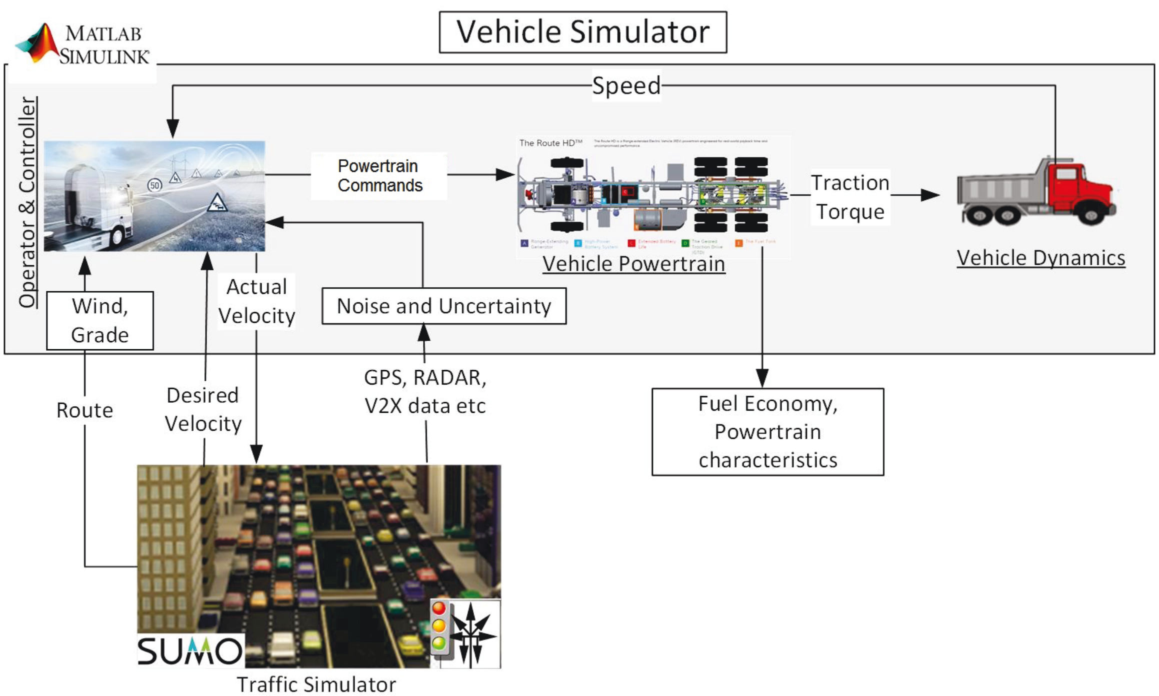

Figure 3.9: Birds-eye view of simulation architecture

etc. a can be considered as sensor outputs. In this case, these values can be considered as output from a RADAR sensor and a V2I receiver respectively. Sensor noise can then be modeled separately by understanding individual sensors behavior and creating a custom noise pattern that can be added to the sensor outputs from the SUMO before being fed into the controller.

3. **Powertrain Controller:** The powertrain controller can range from a simple start stop controller for mild hybrids that use the sensor data to perceive the environment to make on-off decisions to more complex controllers where sensor data and robustness to uncertainty is crucial for optimal operation. These controllers process inputs from SUMO and yield control inputs to the powertrain.

4. **The powertrain and vehicle dynamics model:** Helps calculate energy consumption and fuel economy values. Based on the fidelity of the system model, one can choose to observe dynamics involved in various components and study their effects on the environment or vice versa.

3.5.2 Route Generation and Traffic Modeling Using Real-world Data

As discussed in Section 3.3.2, the process of traffic and transportation system simulation involves two categories of elements that need to be defined, 1.) Static components like a network of roads, infrastructure elements (like speed limits and location of stop signs), route definitions etc. 2.) Dynamic Components include, vehicle definitions (like acceleration limits and top speed) and driver behavior attributes (reaction time, preferred head way, aggressiveness etc.). Both these components together determine the flow of traffic across the simulated network and hence affect the velocity traces of all vehicle in the simulation. This section provides a brief overview of the various techniques that can be used to create a realistic network. This includes importing a map of a roadways from a desired location from a web based repository and generate traffic from real world measurements.

Network Generation

In order to achieve a realistic traffic simulation one needs to have a real-world like network for the simulation. SUMO allows the user to design their own network or import it in the form of a '.xml' file from other sources. Map databases contain comprehensive and updated maps of most cities and can help in creating a detailed network for the purposes of traffic simulation. In this work 'Open Street Maps' is used to import a network, that is representative of the road transportation system in Columbus Ohio. 'Open Street Maps' (OSM)[8] is a web based repository of open source maps that enables maps to be saved offline '.xml' files. A SUMO utility called 'NETCONVERT.exe' can be used to process the map for use in simulation.

Route Generation Techniques

Once a network is imported, a travel route needs to be defined for each vehicle in the simulation. This process of defining routes for constituent members are detailed in Section 3.5.2. However, it is often desired that the host vehicle that is being simulated in the TIL framework needs to be tested with a specific route that meets certain criteria. This is achieved by manually defining the required route for the 'Host' vehicle. For purposes of this work three methods have been explored:

1. **Extract Route From GPS Trace :**A Global Positioning System (GPS) trace can be obtained for a given route by either driving along the said route in real

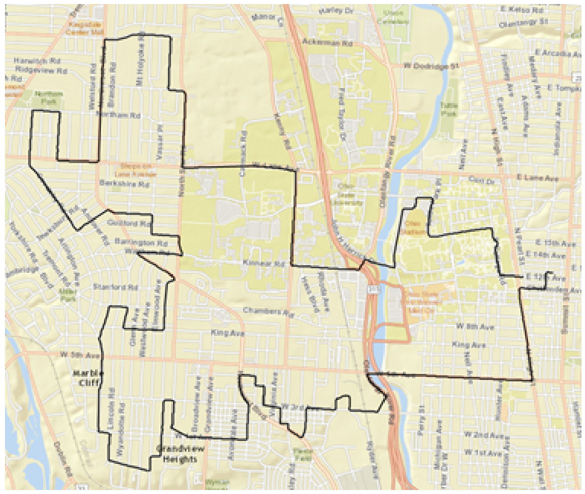

Figure 3.10: Route generation by real-life GPS data logging

life and logging GPS data or by using advanced mapping tools like Google Earth. The extracted GPS coordinates can be associated with respective road segments with help of SUMO and their corresponding identification numbers (edge ID) can be extracted. The edge ID numbers are unique within a given map database, hence a time sorted unique sequence of these edge ID numbers form the route. If GPS data is collected by data logging through experimentation, additional steps might be required to post process the acquired GPS data to snap them onto nearest road segments and account for data collection errors. Figure 3.10 displays the post processed result of route creation through GPS data logging from a real-life test.

2. **Generate route from random section of edges:** Arbitrary routes can also be created by selecting random edges and routing the vehicle to these edges via a shortest path algorithm. This process of picking random edges and creating

a travel route between them can be repeated and concatenated with the result from the previous trial, until total trip length has reached a said value. The output for such process is displayed in Figure 3.11. This technique can yield infinite number of different routes and serves as a good validation tool when used with TIL.

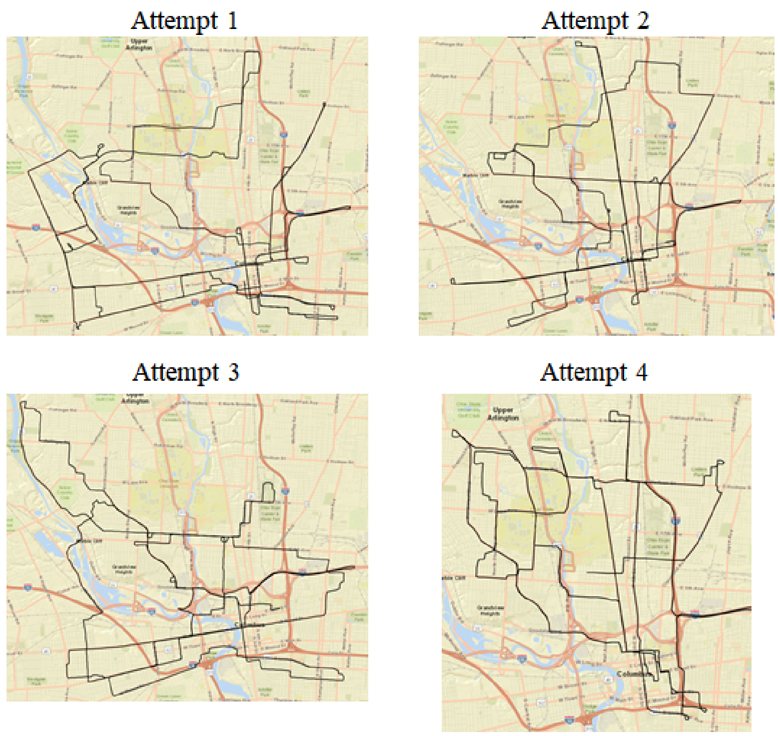

Figure 3.11: Arbitrary route generation by selection of random edges

3. **Generate route by manual selection of constituent edges:** If the exact route to be tested is known, its route can be created manually by selecting edges along the specified path. Figure 3.12 illustrates a route that has been created

using this technique. In this case the selected route refers to the proposed setup for the ‘Columbus Smart City Corridor’.

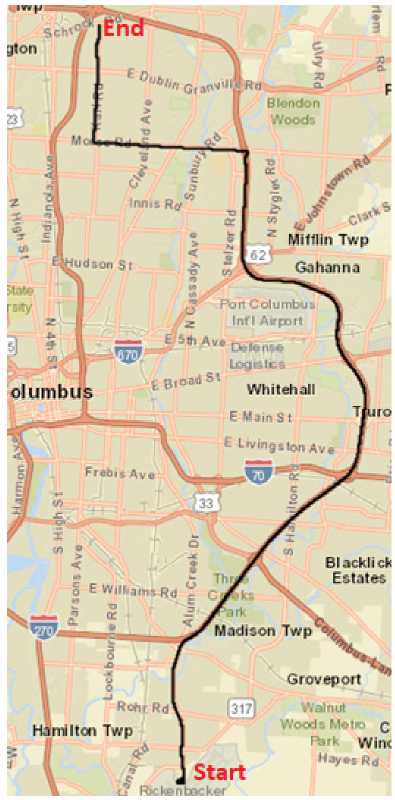

Figure 3.12: Creating the columbus smart city corridor’s route by manually selecting edges

4. **Route Creation through turn probability:** This technique is not explored in this work. However, given the turning probabilities for the ‘host’ vehicle at a given intersection (subject to the turn being lawful as per the defined network). A route can be generated though a random process of selecting a starting edge, and choosing the direction of turns (left, right or straight) at a every intersection.

Traffic Calibration

A network only defines the roadways, speed limits, stop signs, traffic lights, and direction of vehicle movement within the environment. So as to introduce traffic, the user needs to define the member vehicles that collectively constitute traffic. This involves deciding the number, vehicle types (this determines the acceleration limits, maximum speed and other vehicle properties), driver types, individual routes and distribution of the vehicles that move within the network. This process of introducing traffic on the virtual roads within the SUMO environment is called 'Traffic Demand Modeling'. To create a virtual environment that is realistic, the user needs to calibrate the 'Traffic Demand Model' such that it imitates real life. Various tools are provided along with the SUMO package to aid in this process. Each of these tools use different types of data to create traffic, these data types and tools are summarized below:

1. **Demand Modeling through Trip Definition :** Here the user creates a 'trip' for individual vehicles that constitute traffic. A 'trip' is SUMO's definition for a vehicle movement from a set origin to a prefixed destination. It is defined by the point of origin, destination and trip time. Users can either choose to create these trip files manually or use inbuilt tools like 'randomTrips.py' to generate a set of random trips on a given network. The trips are then expanded to 'Routes' that contain information on the exact path traversed by a vehicle from its origin to destination. For realistic results, the trips need to be chosen carefully such that it introduces an appropriate number of vehicles headed along different sections of the network.

2. **Demand Modeling through Flow Definitions:** 'Flows' refer to a common direction of movement for a group of vehicles. This method is similar to technique 1 described above,but here the user defines origins and destinations to traffic 'Flows'. Member vehicles that constitute a flow can depart at different times with the departure time being periodic or following a said distribution. Alternatively, a user can also choose to not specify a flow's destination, but define 'turning ratios' - that specify the probability of a member vehicle turning left, right or proceeding straight at a given intersection. Through experience, this technique is found to be useful when dealing with traffic along large networks with a combination of interstates, highways, main roads and residential roads; since it offers a direct control on the number of vehicles that pass along major roads and hence their traffic conditions. Additionally, defining flows along with turning ratios reduces the manual effort towards creating independent trips to populate the network as desired.

3. **Demand Modeling through Detector Data:** Detectors are traffic counting sensors used by urban planners and government agencies to monitor the usage of roads. These entities study data hence obtained and calculate a scalar value known as the 'Average Annualized Daily Traffic' (AADT). This number indicates the number of vehicles that pass through a given point on the road on a daily basis over a twenty four hour period. This data is often published as public record and can be accessed via the world wide web. AADT numbers can be used by SUMO to generate traffic in a given network. Since AADT numbers are freely available and are derived from real-world measurements, it has been used as one of the methods for traffic generation in this work. AADT data from the

Ohio Department of Transportation for the city of Columbus that is available at [97] is used to model traffic flow along Columbus's smart city corridor.

4. **Demand Modeling through OD Matrices:** Certain government agencies provide public data in the form of Origin-Destination Matrices. These tables are often used in the domain of urban planing and traffic modeling. They specify the number of trips originating from a given road (origin) destined to an other road (destination) withing the network . SUMO offers a set of utilities to import such matrices and convert them into traffic.

5. **Demand Modeling through Population Statistics :** Population statistics within a network can be used to generate traffic demand by accounting for the two major groups of activities ie., Work and School based traffic, and leisure traffic. This method takes into account time of day, city limits, school times, work and residential locations in a network etc. Demand Modeling through population statistics uses probabilistic knowledge of traffic behavior and does not give the user a direct control over the traffic densities. Also, owing to the nature of this technique, it presents a unique challenge that the user needs to posses knowledge about the region, in order to specify work,school, leisure and other areas within a given network.

The above list provides a concise overview on the available techniques to create and calibrate traffic. More information on traffic calibration and demand modeling can be found on SUMO's user manual, a digital version of which can be found in [98].

3.5.3 Sensor Emulation

Section 2.3 details various sensors and systems that can be used onboard a vehicle that makes it possible to understand its surroundings. However, SUMO does not have inbuilt sensor models; hence it cannot independently publish such measurements. The TIL framework however, allows the emulation on onboard sensors such that it yields sensors measurements as though they were acquired from sensors aboard the host vehicle. This is achieved by using an interplay of TraCI and SUMO. In the context of implementing a TIL simulation this is done by querying SUMO for different types of information using predefined TraCI functions. For example, to emulate a ranging sensor; we first list the types of outputs that a real-life sensor like RADAR or LiDAR would provide. From Section 2.3.6 we know that ranging sensors typically provide Distance to leading vehicle and its speed. With this knowledge, appropriate TraCI functions that can be used to query this information. Often the host vehicle's is explicitly specified (by specifying their corresponding ID numbers) in these functions so that the information is collected from the host vehicle's perspective in the SUMO environment. The sensor emulation is implemented by mapping sensors and information they provide to appropriate TraCI functions, as shown in Figure 3.13.

In figure 3.13 'TestVehicleID' refers to the identification number of the host vehicle. Similarly, 'LeaderID' corresponds to ID number of the vehicle immediately ahead of the 'host vehicle'. In some cases, multiple functions calls are required to extract said data. In such instances the'$->$' symbol has been used to indicate that output of first function is passed to a second function and so on.

Information	GPS	Maps	IMU	Camera	V2V	V2I/S	Ranging	Traci Functions
Self Localization	■							traci.vehicle.getPosition('TestVehicleID')
Other Vehicle Location					■			traci.vehicle.getPosition('VehicleID')
Grade	■					■		Simulink, distance based Look up table
Self Speed	■		■					traci.vehicle.getSpeed('TestVehicleID')
Longitidinal Acceleration	■		■					**d/dt** {traci.vehicle.getSpeed('TestVehicleID')}
Lateral Acceleration	■		■					**d/dt** {traci.vehicle.getAngle('TestVehicleID')}
Leader Ranging				■				traci.vehicle.getLeader_osu('TestVehicleID')
Leader Speed				■			■	traci.vehicle.getLeader_osu('TestVehicleID')
Leader Acceleration				■			■	traci.vehicle.getLeader_osu('TestVehicleID') ->traci.vehicle.getAccel(LeaderID)
Leader Calssification				■				traci.vehicle.getLeader_osu('TestVehicleID') ->traci.vehicle.getVehicleClass(LeaderID)
Lane Identification				■				traci.vehicle.getLaneID(TestVehicleID) (or) traci.vehicle.getLaneIndex(vehID)
Number of Lanes		■		■				Simulink edge ID based Look up table
Road Width		■		■		■		traci.lane.getWidth(traci.vehicle.getLaneID('TestVehicleID'))
Road Localization				■				traci.lane.getLength(laneID)
Speed Limit		■		■	■	■		traci.lane.getMaxSpeed(LaneID)
Road Traffic number		■			■	■		traci.edge.getLastStepVehicleIDs(edgeID) -> Count
Road Utilization %		■		■	■	■		traci.edge.getLastStepOccupancy(edgeID)
Road Demographics					■			getLastStepVehicleIDs(edgeID) -> traci.vehicle.getVehicleClass(VehicleIDs)-> count (or) traci.edge.getLastStepLength(EdgeID)
Edge Average Length					■			traci.vehicle.getLastStepLength(edgeID)
Edge Average Speed					■	■		traci.vehicle.getLastStepMeanSpeed(edgeID)
Number of vehicles in jam		■			■	■		traci.vehicle.getLastStepHaltingNumber(edgeID)
TLS Position		■			■	■		traci.vehicle.getVehicleLights() -> Post Processed in Matlab
TLS Phase						■		traci.trafficlights.getPhase(TrafficLightID)
TLS Time to green						■		traci.vehicle.getVehicleLights() Customized Code
TLS Distance		■				■		traci.vehicle.getVehicleLights() -> Post Processed in Matlab
Distance to Destination		■						traci.simulation.getDistanceRoad(Current Edge, Current Pos, Destination Edge, Destination Pos, isDriving)

Figure 3.13: Sensor emulation with TraCI

This implementation is a product of the realization that; from the vehicle level control, modeling and simulation standpoint, it becomes more important about what realistic sensor information is available and not about how that information is derived. The outputs from SUMO are exact representation of current state in the virtual environment. Nonetheless, if a need is felt to introduce noise or make the sensor outputs more lifelike, a mathematical model of the sensor can be implemented in Simulink and outputs from SUMO can be post-processed before being fed into the controller. Sensor based results presented in Section 3.6 are obtained from emulations that followed the above methodology but without the addition of any external noise.

3.6 Results from Sensor and V2X Emulation

A traffic-in-loop co-simulation was run for vehicle model using the above framework. The results from this co-simulation have been presented in the following section. The route chosen was a 4.5 Mile long circuitous loop in Columbus, Ohio. Map required for this simulation was obtained from 'Open Street Maps'. A snap shot of the selected region is shown in Figure 3.14. The route stars and ends at the point marked with the red star. The vehicle follows a predetermined route along along the direction indicated by the arrows. The other 'stars' on the map indicate user defined way-points. The route has been chosen such that the host vehicle, i.e., the vehicle that is being tested, undergoes portions of urban and highway driving.

The simulation is setup in a manner such that the vehicle is subjected to random traffic at every iteration. In this context, 'random' refers to a situation where the average traffic density remains approximately the same; but the vehicle and driver types encountered by the 'host' varies every simulation run. This situation is analogous to a case where a person driving to work every day leaves home at the approximately the same time hence facing approximately same density of traffic. However, he/she is subjected to variations in their velocity trajectories owing to other vehicles that they encounter on the road. This particular scenario is designed to capture such variations. To ensure that the simulation is realistic, a need was identified to reference the results to some standard metric. In this case the authors have attempted to calibrate the traffic density on the given map such that the average speed over 50 runs of the TIL simulator (when subjected to the above mentioned 'random' traffic scenarios) match

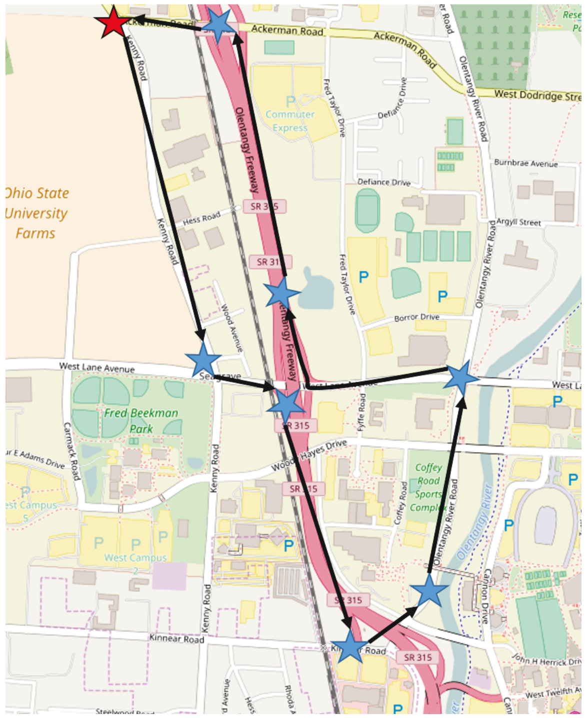

Figure 3.14: Simulation route (Map source:Open Street Maps[8])

with the average speeds over NEDC and FTP Cycles. The results of the calibration is presented in Table 3.1.

Table 3.1: Statistical validation of driving scenario

Parameters	**NEDC**	**FTP75**	**SUMO**
Average Speed (MPH)	20.87	21.2	19.75
Normalized MPG	0.9487	1	1.0037 (±8%)

3.6.1 Simulation Results-Traffic Data and Sensor Emulations

Over the course of the simulation, the TIL simulation framework also records various sensor outputs. A few selected outputs of interest are displayed in Figures 3.15 through 3.19. These outputs correspond to a random simulation run along the given route. It is worth mentioning that, these results correspond to a randomly selected simulation run and do not relate to either the best or worst case fuel economy scenario.

Figure 3.15 illustrates the GPS position trace of the 'host' vehicle. In Figure 3.15, the red markers signify the presence of a traffic light controlled intersection. This form of information is said to be emulating a GPS sensor and an map. The positions of vehicle is localized using a GPS system and traffic light positions are obtained from the map database that is accessible to the vehicle in some form. TraCI offers a functionality that helps obtain the Cartesian coordinates of the host vehicle in real time. The author utilizes this functionality and converts the Cartesian system data to latitude-longitude data to emulate a GPS sensor's behavior.

A virtual V2I communication capability is demonstrated in Figure 3.16. Here the traffic light is said to be actively communicating its location, phase and time to turn 'green' to the vehicle. In this case we assume that the traffic lights are capable of communicating via some form of wireless communication and the host vehicle is appropriately instrumented to receive the same. The discontinuities in data presented by figure 3.16 corresponds to time instants where the vehicle did not detect an upcoming traffic light. This situation can arise when the vehicle is on a road segment that is not controlled by a traffic light.

A few mapping services provide over the air information on traffic density and average vehicle speeds on roads to clients in real-time. The authors implemented such

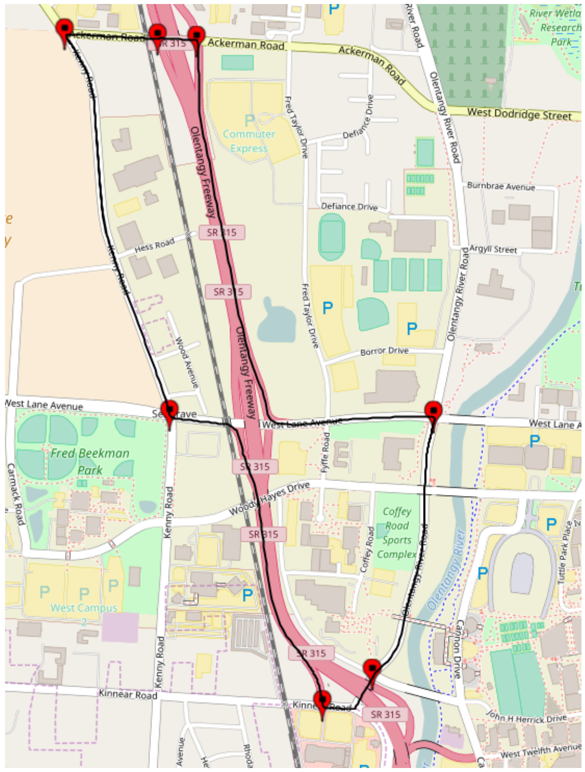

Figure 3.15: Position tracking with virtual GPS sensor and emulation of map based navigation system

a system within the TIL framework. The results of which are displayed in Figure 3.17 and 3.18. Utilization refers to the level of occupancy of a given road segment and can be treated as a measure of traffic density on a given road. Figure 3.17 displays the results from an emulation of such a service which is available to the host vehicle

Similarly, figure 3.18 displays a comparison between the average speed over all vehicles on a given road as communicated from a central server, with respect to the actual speed achieved by the 'host' vehicle on that given edge at a given instant of time.

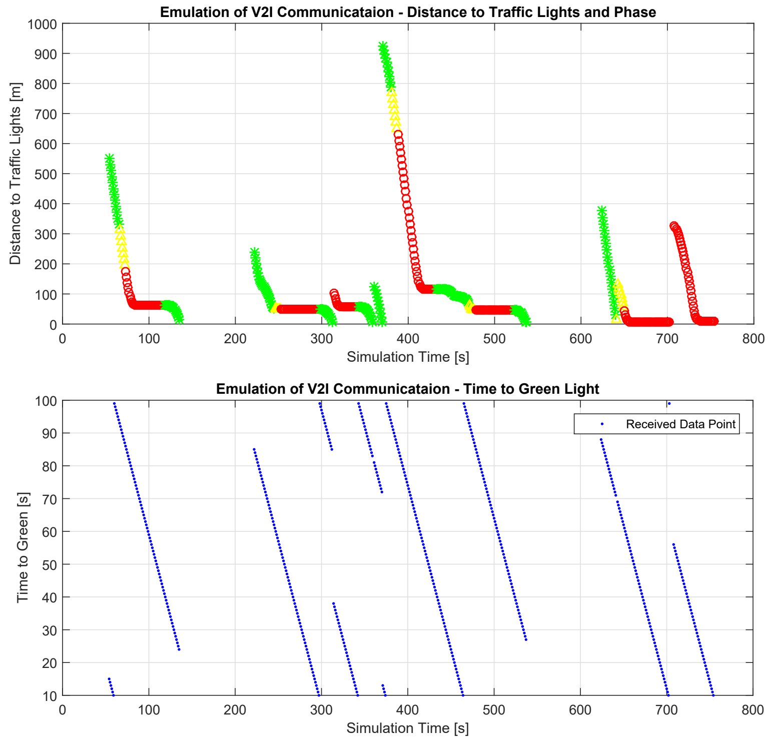

Figure 3.16: Virtual V2I communication with traffic light

Additionally, through the use of TIL simulation the authors were also able to emulate the presence of a ranging sensor like RADAR or LiDAR and obtain the instantaneous distance to, and speed of their immediate leader as displayed in Figure 3.19. This data is subjected to discontinuities since there occurs certain instances where the host is free to move and no leader is detected. Also, if the leader vehicle is beyond a sensing threshold distance, the its presence is not observed. From SUMO's perspective this is implemented by querying SUMO for the vehicle id of the vehicle that is directly in front of the host using TraCI. Further, by utilizing the obtained ID, the leader's speed and position are calculated.

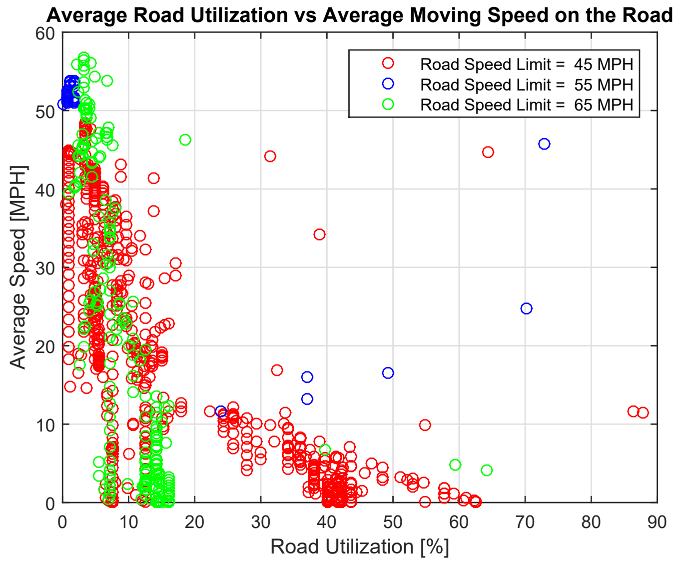

Figure 3.17: Road traffic density for current road versus expected vehicle speed

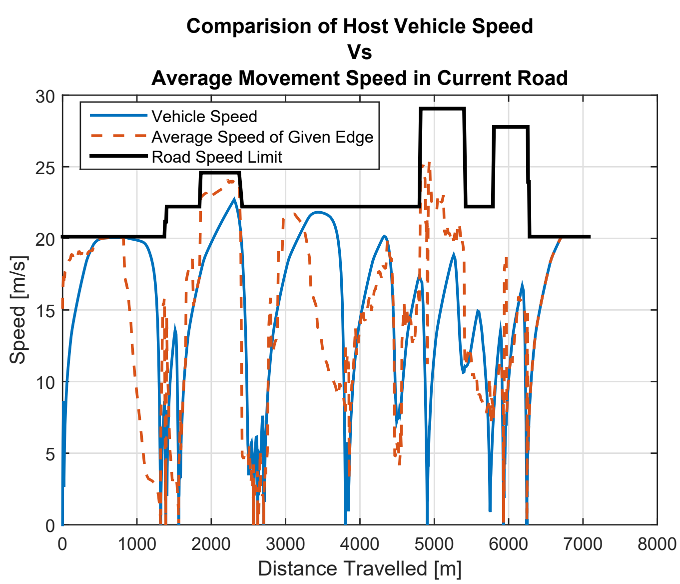

Figure 3.18: Average speed of traffic on current Road Vs host vehicle‘s speed

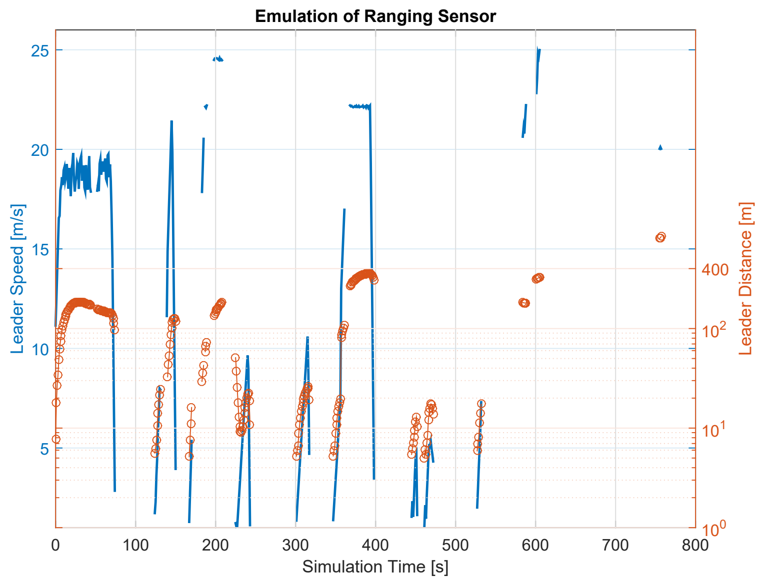

Figure 3.19: Obstacle (leader vehicle) tracking and classification with virtual ranging and vision sensors

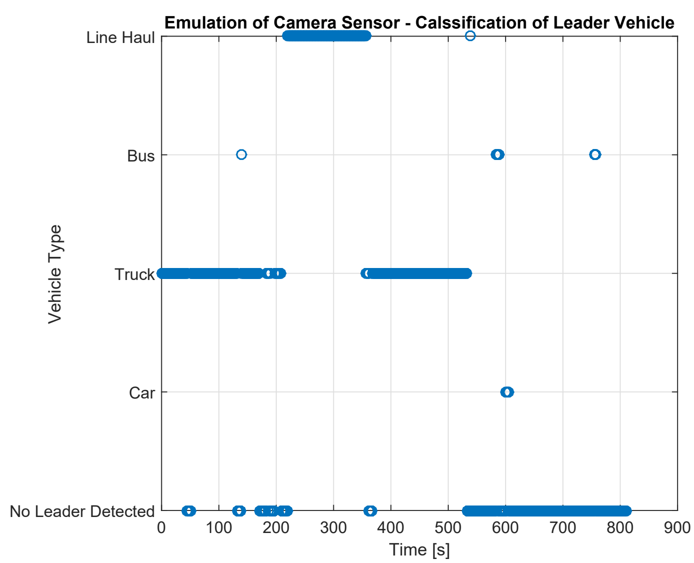

Figure 3.20: Obstacle (leader vehicle) tracking and classification with virtual ranging and vision sensors

Since a ranging sensor is incapable of identifying/classifying the lead vehicle, through the implementation of a virtual vision sensor the authors are also able to record the of vehicle that preceded the host vehicle as shown in subplot 2 of Figure 3.19. The vision sensor is implemented by using the lead vehicle's ID to query its type in SUMO.

from the above plots it is evident that the TIL simulations can yield virtual data that resembles the vehicle electronic- horizon look ahead data. Given various sensor packages as described in Chapter 1 the length of this horizon and its accuracy can vary. This can be implemented by selectively corrupting data from certain sensors or not record them at all.

3.6.2 Simulation Results-Powertrain

Figure 3.21 depicts the variation of predicted fuel economy over the 50 cycle. This shows a 15% variation in fuel economy (between the minimum and maximum values) with a pronounced peak at the mean fuel consumption. In Figure 3.21 the x-axis represents the variation in the fuel economy with respect to the mean indicated as '1'. The fuel economy numbers seem to follow a normal distribution. Figure 3.22 on the other hand shows a Poisson like distribution for the spread of trip times. This is an expected behavior since the trip time cannot go below a specific value owing to road speed limits.

In order to explicitly study the causes for the variation in fuel economy, Figure 3.23 illustrates the time trace of host vehicle's speed. It is readily observable from the figure 3.23 that in the worst case scenario (the case that produced the least fuel economy), the host vehicle idles for a longer duration. It is apparent from the distance

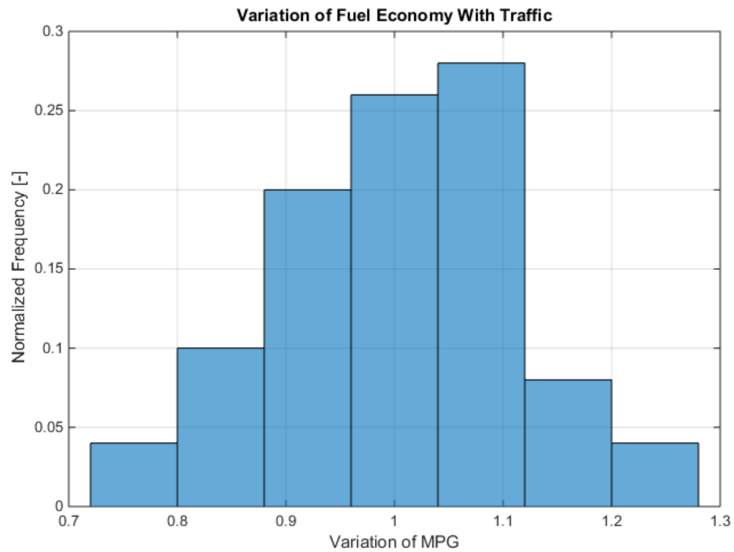

Figure 3.21: Variation of fuel economy due to traffic

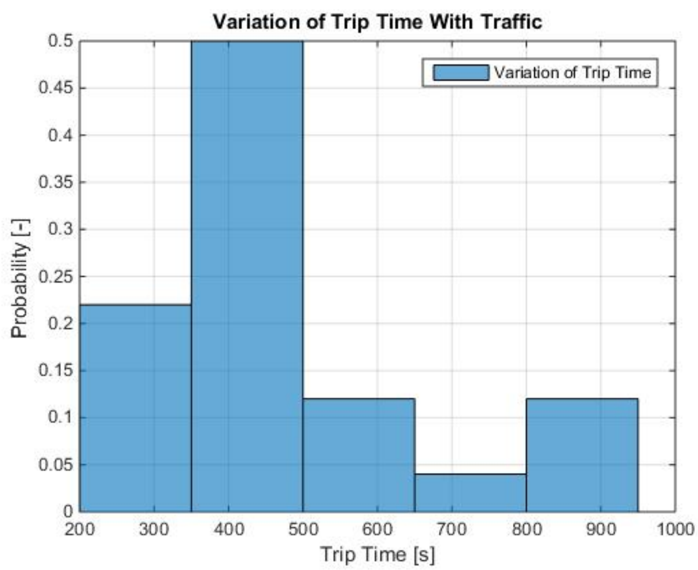

Figure 3.22: Variation of trip time due to traffic

versus speed plot that vehicles in both cases came to a halt at a certain distance; indicating that the stop was owing to traffic lights or a stop sign. In this particular case, the halt can be attributed to a traffic light; the vehicle inches forward for a brief period but remains stuck in traffic for an extended duration there after, owing to congestion and traffic light phasing.

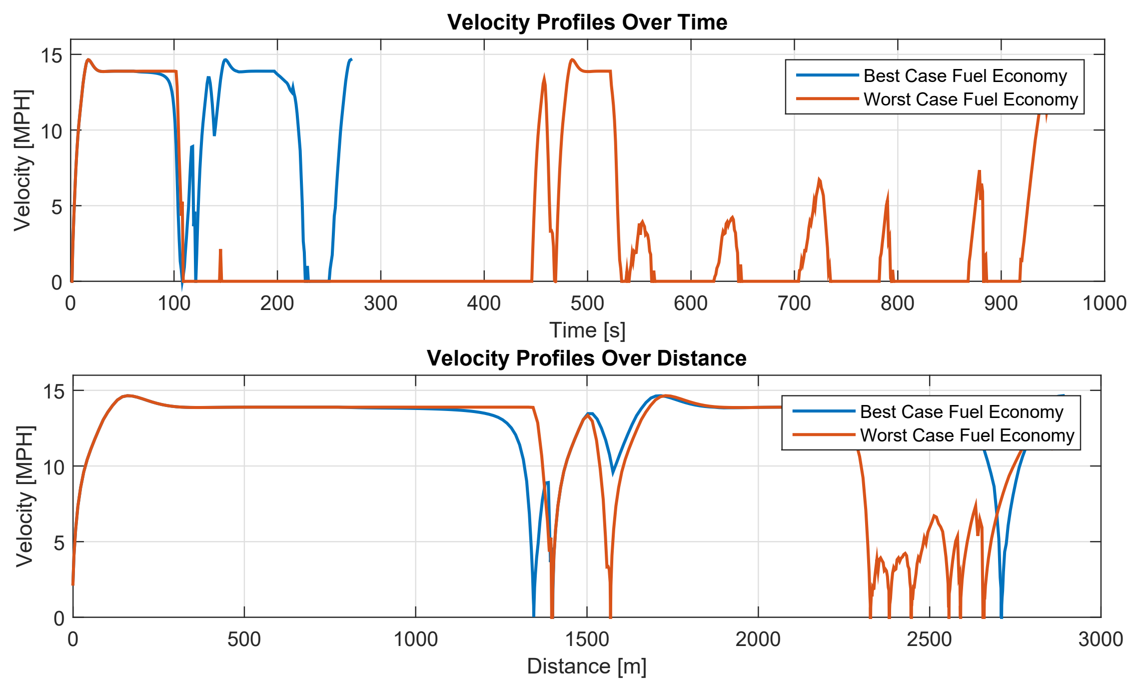

Figure 3.23: Speed profiles of best and worst fuel economy cases

Figure 3.24 indicate the operating points of the engine over the given simulation run as obtained from the powertrain model. Given a specific shifting strategy, gear shift are also recorded and the instantaneous gear indexes are reported by the powertrain model. The results of which are illustrated in Figure 3.25. It is observed that the vehicle undergoes a total of 120 gearshifts during the simulation period.

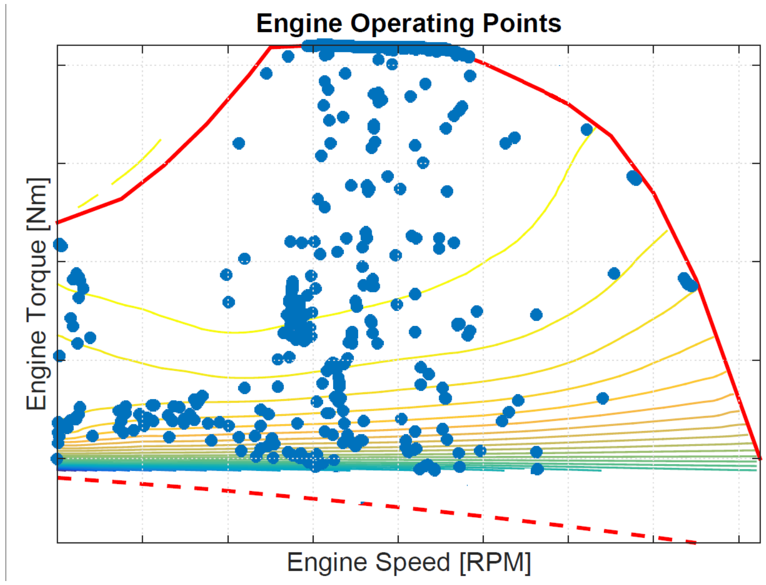

Figure 3.24: Engine operating points

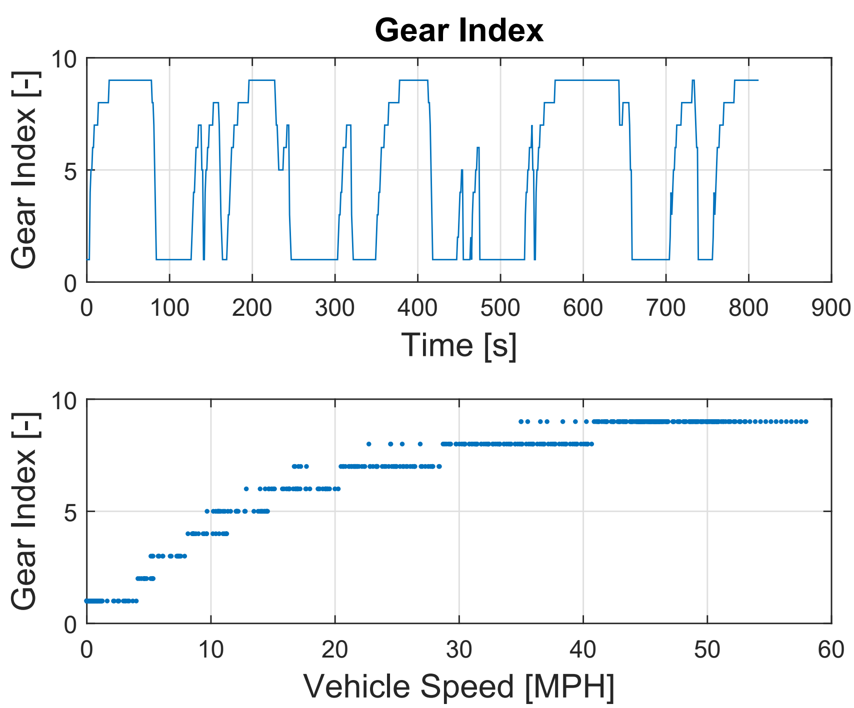

Figure 3.25: Gear shift behavior

3.7 Chapter Summary

A traffic integrated powertrain simulation framework called 'Traffic In Loop' simulation was introduced. It facilitated the simulation of environmental and infrastructure elements like grade, stop signs, traffic etc.. Additionally, the framework's set up enabled drive cycle independent evaluation of powertrain performance and emulation of virtual sensors. Figure 3.26 illustrates the integration of various elements and their interaction in the TIL framework.

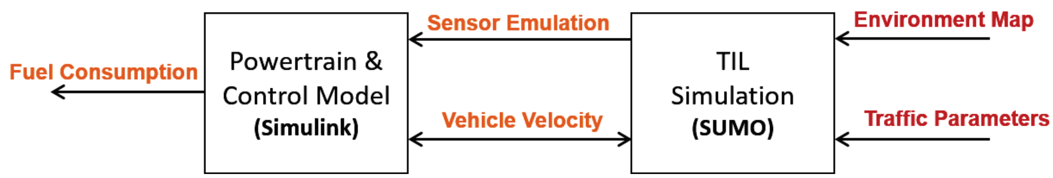

Figure 3.26: Structure of Traffic-In-Loop simulation

In Figure 3.26, environment map and traffic parameters refer to - the description of road networks, location of stop signs,speed limits, grade, and traffic distribution.

Chapter 4: Look Ahead Energy Management Strategy for Range Extended Electric Vehicles

This chapter focuses on a look ahead based energy management system that has been developed for a 'Range Extended Electric Vehicle' (REEV) however some of these concepts can be carried over into other hybrid architectures. An introduction is provided to vehicle and traffic models used for the simulation. In sections 4.4.1 different levels of look ahead data are described and their integration into powertrain control algorithms are explained. Ultimately, a energy management strategy called 'Delta Energy Controller' (DEC) is outlined. Fuel economy benefits derived from DEC with various levels of look ahead information are compared.

4.1 Range Extended Electric Vehicles

REEV's are a type of hybrid electric vehicles which have a large re-chargeable battery pack coupled with an appropriately sized motor. The REEV architecture allows the batteries to be recharged from an electric grid. The battery pack is sized appropriately such that, these vehicles can travel considerable distances solely on electric power stored onboard. However owing to current technology limitations, charging these large battery packs are more time consuming than topping a vehicle's fuel tank with gaseous or liquid fuel to give the same travel range. The lack of charging

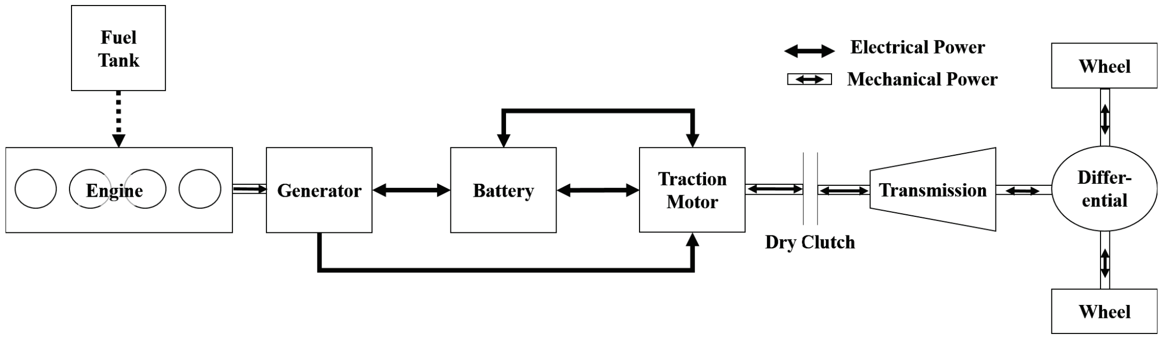

Figure 4.1: Powertrain architecture

infrastructure in many parts of the county further add towards limiting the pure electric range. However, the REEV addresses this concern with the combustion engine that serves as a back up power source and can propel the vehicle with satisfactory performance for the reminder of a trip. The battery pack is recharged from the power grid when the vehicle is not in use. If the grid energy is clean and sustainable, this mode of propulsion can significantly reduce vehicular carbon emissions and pave the way towards a greener future. Further details about the flow of energy inside the vehicle is clarified in Section 4.2.

4.2 Introduction to Vehicle Model

For purposes of this work, the vehicle being modeled is a class 6 truck. Its component specifications are listed in figure 4.2. Figure 4.1 illustrates its architecture. This vehicle is assumed to be operating as a delivery truck for the purposes of this work.

In order to accurately estimate fuel consumption and battery state of charge (SOC), a simplified vehicle plant model was built in Simulink. The model replicates

Parameter	Description
Engine	6.7L – diesel
Max Power	246 kW
Max Torque	1018 Nm
Idle Speed	800 RPM
Cylinder	6
Transmission	4 speed
Gear Ratios	3.49 ,2.03 ,1.47, 1
Differential	5.57
Clutch Type	Dry clutch
Vehicle mass	9392.5 kg
Payload	957 kg
Vehicle (C_d)	0.75
Vehicle (A_f)	5.69 m^2
Tire radius	0.5162 m
Rolling resistance co-eff	0.007
Traction Motor Type	Permanent Magner DC
Max Power	227 kW
Max Torque	1200 Nm
Generator	Permanent Magner DC
Max Power	130 kW
Max Torque	690 Nm
Max Current	126 A
Battery	112 kW-h, 600 V, Li-ion
Series	182
Parallel strings	86 (188 Ah)

Figure 4.2: Host vehicle's component specifications

energy flows within the vehicle and powertrain components through a simplified representation of the powertrain components. It ignores certain phenomena such as temperature variation in engine, battery dynamics, effect of after-treatment system, battery aging, pollutant emissions, etc. . Conversion losses like that during flows through a connection device, friction losses and other inefficiencies are modeled using steady state efficiency maps - as functions of operating conditions. In this simplified model, the vehicle is considered as a point mass object and equations describing its motion can be written from the computation of equilibrium forces. The vehicle and powertrain components models are built following modeling philosophy and equations described by S.Onori et al. [9]. The engine, generator, and traction motor units have been modeled by steady state maps whereas the battery is described by a second order model. Inertial masses of the crankshaft, Engine-Generator coupling and the generator's rotor have been lumped together and represented in the coupling shaft. A similar approximation has also been made downstream of the traction motor, where the gearbox and differential's inertias are lumped into the propeller shaft. While the each axel's shafts and wheels inertias are lumped together.

It has been seen from Chapter 3 that the TIL framework accounts for the non-causal behavior of real world systems. To exploit this capability, its imperative that the vehicle plant model respects limits of the powertrain components and represent the vehicle's real life dynamics as much as possible. Hence a forward dynamic modeling approach is adopted. This simulation technique bases the operation of each powertrain component through physics based dynamic equations, that describe its state evolution. Hence, the plant computes vehicle speed as the result of a dynamic

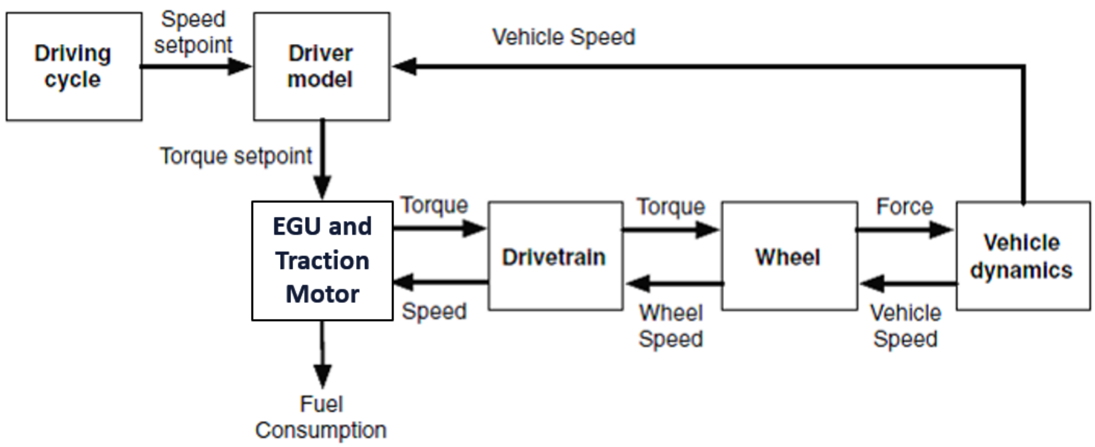

Figure 4.3: Structure of vehicle plant model [9]

simulation according to physical relations. The flow of information the overall structure of the vehicle model is illustrated in 4.3.

Owing to powertrain's component limitations, the plant model may not exactly follow the desired velocity trace. This introduces an error between the actual and desired velocities. The TIL framework accounts for this difference through a speed feedback 'ϑ_{k+1}' as shown in Figure 3.8

REEV architecture decouples the engine from wheels. Hence the engine can be operated at any speed-torque point while the electric motor delivers power required to propel the vehicle. For sake of simplicity of problem formulation, the engine is operated at a fixed load point in this application. The selection of this operating point is done in consideration of overall efficiency curves of the engine-generator unit. This point is found to correspond to an engine torque of 350 Nm and 172 rad/s - producing a net electrical power of 54.8kW from the generator.

The battery pack sizing for the given vehicle is such that it can support upto 50 miles of pure electric propulsion. However in the interest of numerical computation and reducing simulation time, the route used for TIL simulation is limited to 18 miles. Hence to account for the shorter trip length, the battery pack's operating SOC range has been modified to make it commensurate with designed trip length. The minimum SOC is thus limited to 77%, which corresponds to a 8% maximum depth of discharge.

The baseline,Rule based control strategy tries to achieve a linear battery discharge, such that the pack reaches a target SOC of 77% after 19 miles of driving. This is achieved by creating a reference SOC trajectory as a function of travel distance. This strategy is a rule based thermostatic SOC tracking control, where reference SOC is a function of distance. According to this strategy, the Engine-Generator Unit (EGU) is turned 'ON' when the battery pack SOC falls below 1% of the reference value. Similarly, the EGU is turned 'OFF' when the battery has been charged to 0.5% more than the reference SOC. Here we also assume that the engine's On-Off status is solely an output of the powertrain control algorithm and is independent of the vehicle's 'ignition key' state. Each engine start is penalized with a constant fuel quantity that corresponds to 40 seconds of idle fuel consumption for each start event. This rule based controller does not utilize look ahead information. An adaptive suboptimal controller called 'Delta Energy Controller' is introduced in Section 4.5 which utilizes look ahead data to improve the over all fuel economy.

4.3 Application Specific Calibration of Traffic In Loop Simulation

The Simulink based forward dynamic vehicle model described in Section 4.2, was integrated into the Traffic In Loop simulation platform. The integration was done by following the architecture detailed in Section 3.5.1. Since the said vehicle is assumed to operate as a delivery truck, route and traffic scenarios created for purposes of developing and testing a LEMS controller needed to reflect the same.

As a delivery truck, the vehicle is bound to undergo multiple start-stop events. Most delivery truck operators turn 'off' the vehicle at stops and start it back 'on' once the delivery is complete. In a conventional powertrain, the engine crank is a transient event and consumes a finite amount of battery charge. and accounts for a transient spike in the fuel consumption. Thus for a conventional combustion engine powered vehicle, it is imperative to model these start-stop events and the duration of each stop. Thus it can be said that every delivery stop negatively impacts the overall fuel economy of a day's trip. If the engine is not turned off, the idle fuel consumption adds to reducing the overall trip's fuel economy. However owing to the test vehicle's REEV architecture; losses at idling, start up, and engine crank phases are zero. The only energy consumer during a delivery stop are the auxiliary loads - which we assume to be non existent as the driver turns off the vehicle.

The above reasoning allows 'delivery stops' to be removed from the vehicle's duty cycle and hence do not need to be simulated during TIL. However, stops at traffic lights and stop signs remain significant as the vehicle is not turned off and auxiliary loads continue to draw power from the vehicle's electrical grid. Thus the route needs to account for these infrastructure based stops whereas it does not need to include delivery stops.

4.3.1 Need for Traffic In Loop Simulation in this Application

In order to develop a look ahead energy management system, data from various on-board sensors and remote sensing technologies need to be emulated. These data points can then be used to generate velocity and power predictions that is to be used by an onboard controller. To test the robustness of the controller, various traffic scenarios needed to be simulated while maintaining the non-causal behavior of the real world. The TIL platform helps immerse the simulated vehicle, its controllers and sensors into a virtual world, that contains pseudo-random traffic and infrastructure elements.

4.3.2 Description of Route and Traffic

In order to effectively develop, validate and test the LEMS algorithm, a route was created such that it captured the driving conditions experienced by a piratical delivery truck. Techniques discussed in Section 3.5.2 describe various methods to create a simulation route from a real world map. Given the author's familiarity with the area around Ohio State University's campus, a suitable route was decided and designed around the same. GPS data was collected by an onroad experiment - using a GPS logger and driving along the shown route. The route is shown in Figure 4.4. It is 18.75 miles long and passes through urban and residential areas. A vehicle on this route interacts with 16 Traffic lights and has many turns. Since the route primarily pases through residential areas, it frequently encounters stop signs and has a average speed limit of 22 MPH. The velocity profile of the host vehicle is ultimately influenced by traffic conditions and phases of traffic lights that the vehicle encounters.

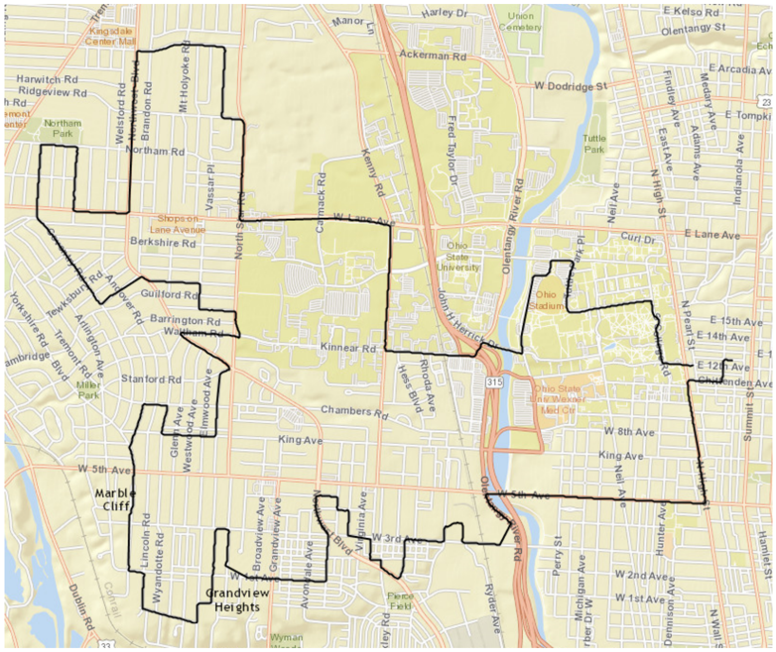

Figure 4.4: Delivery truck route used for development and testing of LEMS controller

Section 3.5.2 discusses various techniques that can be used to generate traffic on a given network (map). Traffic data from detectors and other sources are available only for arterial roads and highways. But the route decided for this specific activity primarily consist of residential roads, where reliable traffic data is not available in public domain. Therefore, the traffic flow along the route is calibrated manually to reflect average density of traffic as observed by experience.

4.4 Velocity and Power prediction Enabled by Look Ahead Technology

4.4.1 Types of Velocity Predictors

This thesis inherits results from velocity predictor related work performed by Hegde.B. [Conversation at Center for Automotive Research, The Ohio State University, August 2017]. Hegde describes various techniques that can be used to derive velocity and power predictions from available look ahead data. Its understood that the reliability of look ahead forecasts correspond to sensors on-board the vehicle. Four types of look ahead predictors are studied in this thesis. They are named in increasing order of information richness and complexity. Vehicle and nominal driver models are inbuilt in the look ahead predictors; they help forecast wheel power demand and gear index as a function of distance in addition to vehicle velocity as shown in Figure 4.10. The model based nature of these predictions ensure that powertrain limits and driver behavioral consternations are respected in the forecasts. However, a human driver's behavior and unexpected onroad variations and trip time predictions are uncertainties in the system. The effect of distance based predictions on trip time estimates and its effect on control is detailed in Section 4.4.2. The vehicle is assumed to be access grade data from some map database. Figure 4.5 illustrates the grade profile assumed for the route described in Figure 4.4 is plotted in .

The different types of velocity predictors considered for purposes of this thesis are enumerated below. Each predictor consecutively adds information to a prior predictor's output. For example, Type 3 predictors add effects of turns on vehicle speed over and above the prediction from Type 2 predictors. Figure 4.6 illustrates the outputs from the four types of predictors described in the following list.

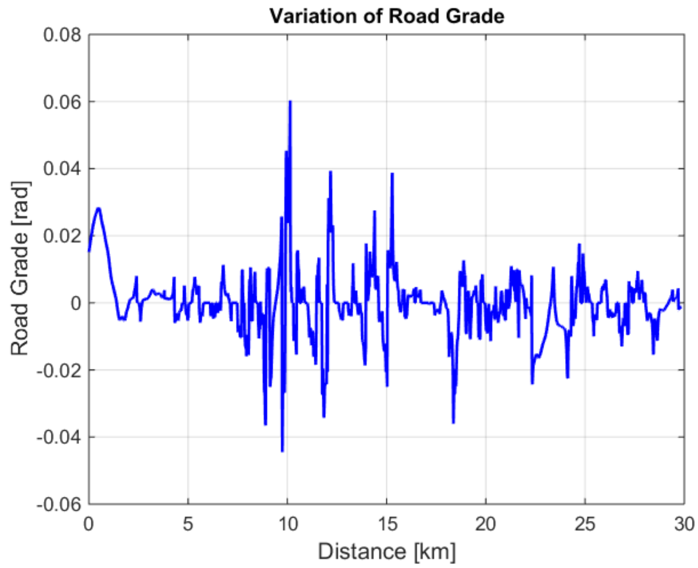

Figure 4.5: Variation of road grade along selected route

1. **Type 1 os SL-Predictors** : These predictors rely on road speed limit (SL) and grade information from maps. They use speed limit data of different roads along the route to forecast future speed. As shown in figure 4.6, the inbuilt powertrain model recognizes the infeasibility of step increase in speed and predicts a gradual acceleration that is in accordance with powertrain component's limitations and a predefined driver behavior model. Grade information is used to make wheel power predictions. However, It does not account for vehicle stops at stop signs and intersections.

2. Type-2 or SLSS-Predictors: Type 2 adds to Type 1 predictors. These predictors include Stop Sign (SS) information and account for presence of stop signs and traffic light controlled intersections along the route. They also use a-priori

knowledge of the vehicle specifications to predict acceleration and deceleration performance.

3. Type 3 or SLSSLT: Type 3 improves speed forecasts from Type 2 predictors by using known information about the route and expected turns enroute. This predictor is designed based on observations that human drivers reduce speed prior to sharp turns following which they accelerate and try to match speed limit of the new road. Type 3 predictors acknowledge this behavior and improves upon the prediction from Type 2 predictor.

4. Type 4 or V2X : This predictor directly uses the velocity trajectory information from an other vehicles that transversed a given route at some point in the past. Predictions are made either by extracting the speed of a prior vehicle as a function of location, or data from multiple vehicles can be fused together to yield a prediction. The fusion process may be defined such that the speed data is weighted differently based on vehicle type, time of travel etc. The underlying assumption of this predictor is human drivers are best indicators of an other human's on road behavior. In this case, actual speed speed information from a similar vehicle that transversed a given road in the past is used to detect other factors that cause drivers to reduce speed that apart from stop signs, traffic lights and turns. This includes the effect of crossing intersections with larger roads (where drivers reduce speed for safety reasons) or effect of curvy roads along the route. The exchange of speed information is facilitated by some form of intermediate medium like an ofline server as illustrated in Figure 2.3.4.

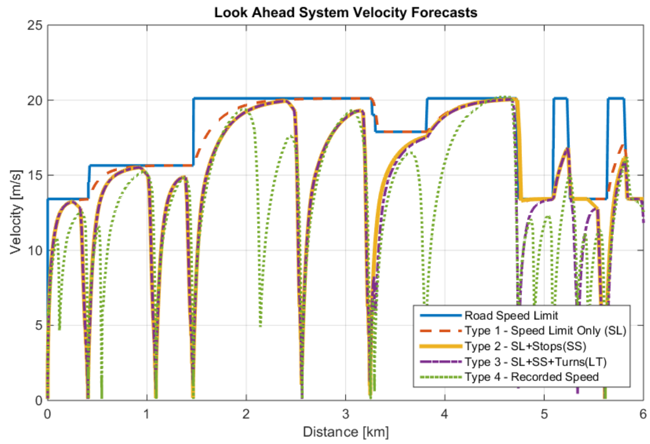

Figure 4.6: Velocity outputs from four types of predictors

Road speed limit shown in Figure 4.6 is shown for illustrative purposes and is not considered a form of prediction. The outputs of the Predictor Type 1 through 4 follow trends as expected from theory explained above. Type 4 predictors represents the historical velocity profile of a similar vehicle that traversed a given route at a past time. Owing to the nature of predictions, the velocity forecast from predictor type 1,2 and 3 remains the same at all times for a given route. However based on traffic conditions, the velocity trajectory from type 4 might vary.

4.4.2 Uncertainty in Trip Time Estimation

Different types of velocity predictors have been discussed in Section 4.4.1. It can be noticed from Figure 4.6 that the predictions are made as a function of trip distance. This implies that trip time is no longer a known quantity and has to be estimated based on predicted speed and distance. Trip time estimation is not straight forwards since the speed vs distance plot does not directly yield time information. As a result different predictors estimate different trip durations as shown in 4.8. Nevertheless,

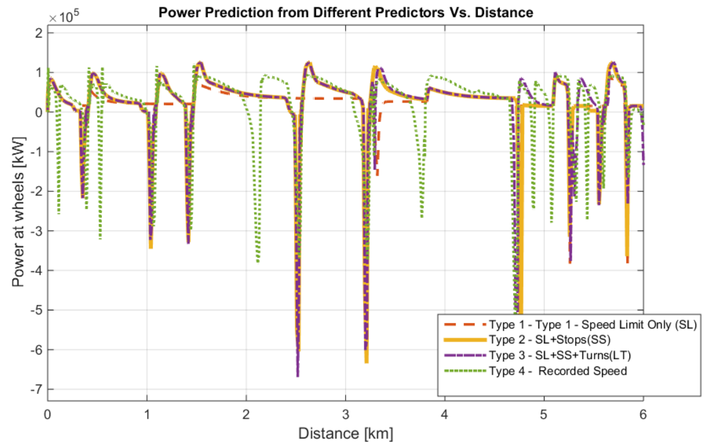

Figure 4.7: Wheel power outputs from four types of velocity predictors

with increase in the level of look ahead information - trip time estimates approach the true value. Figure 4.8 plots trip time as a function of progress towards destination. In this plot, 'progress' value of 1 indicates that the host vehicle needs to travel 1 full trip. Similarly, a 'progress' value of 0.25 implies the vehicle needs to cover $(1/4)^{th}$ the trip's overall length to reach the end. Trip 'progress' of 0 indicates the end of trip.

The trip time estimates are performed based on equation 4.1, derived from basic laws of motion.

$$t(i) = 2 \times \frac{(d(i) - d(i-1)}{v(i) + v(i-1)} \tag{4.1}$$

Where, $d(i) - d(i+1)$ is the difference in distance between two consecutive points on the look ahead predictions. Similarly, $\dfrac{v(i) + v(i-1)}{2}$ is the mean velocity of two consecutive predictions. In Figure 4.9 the velocity and corresponding distance from a arbitrary simulation run is used to reconstruct velocity as a function of time. The original velocity-time trace is then compared to the reconstructed trace. The two data tips on the plot correspond to the perceived end of the simulated trip. It is noticed

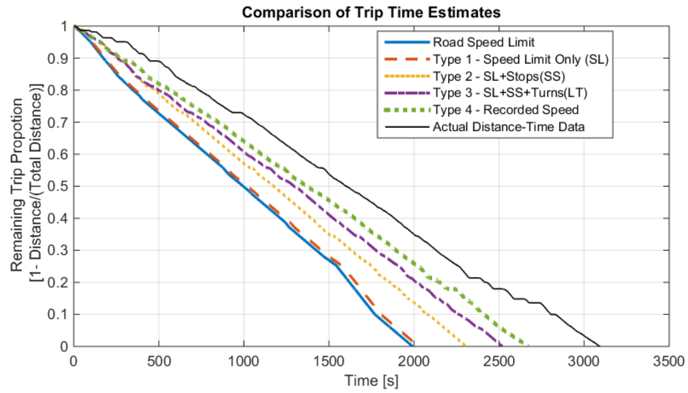

Figure 4.8: Effect of prediction type on trip time estimate

that the reconstruction of time from distance information leads to loss of stop time data and thus erroneous estimates of trip time. If the predictor does not account for this behavior or yield trip time estimates, any time based control algorithm using this data might fail to achieve optimal results.

4.4.3 Integration of Look Ahead Predictions into Powertrain Control

As discussed in the previous section, look ahead information is presented to a controller in the form of wheel power demand, and velocity forecasts as function of distance. This section details a methodology to utilize this data in a powertrain controller to assist making optimal control decisions.

The look ahead predictor uses data obtained from various sensors and technologies to estimate future velocity $(V_1, V_2, ...V_k)$ and Grade $(\theta_1, \theta_2, ...\theta_k)$ as function of distance $(d_1, d_2, ...d_k)$. Distance based predictions are constant and do not vary based on vehicle speed. Thus making the predictions static and more reliable. An inbuilt vehicle-powertrain model utilizes known vehicle parameters to determine wheel

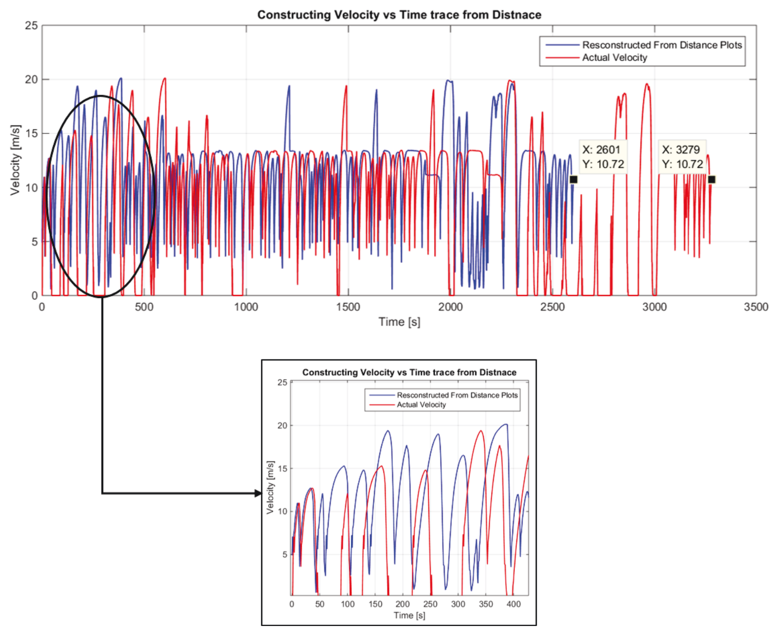

Figure 4.9: Effect of transforming velocity profile from distance to time domain

power demand $(W_{p_1}, W_{p_2}, ...W_{p_k})$ from predicted velocity and grade data. A backwards gear estimation algorithm estimates gear predictions given grade and velocity requirements. Hence the two major outputs of the predictors, gear ratio and wheel power are obtained from the predictors. All prediction uncertainties can thereby be lumped into one of the above outputs. The gear and wheel power prediction from the look ahead controller, is used to calculate battery power demand by using a backward powertrain and battery model. This accounts for efficiencies and limits of various intermediate powertrain components.

Figure 4.10 diagrammatically depicts the process of integrating look ahead data with powertrain controls. The information available from the predictor are velocity profile, grade, and gear forecasts. This implies that the predictors need to have a knowledge of vehicle's properties such as mass, gear ratios, shifting schedules etc.

Ultimate the goal of process described in Figure 4.10 is to calculate the net energy requirement from the powertrain. This quantity is represented as 'Powertrain Energy Trajectory'. A controller such as DEC utilizes this prediction to compute optimal control actions. In this case, the 'Powertrain Energy Trajectory' corresponds to energy drained from the battery and the control action is ignition state of EGU.

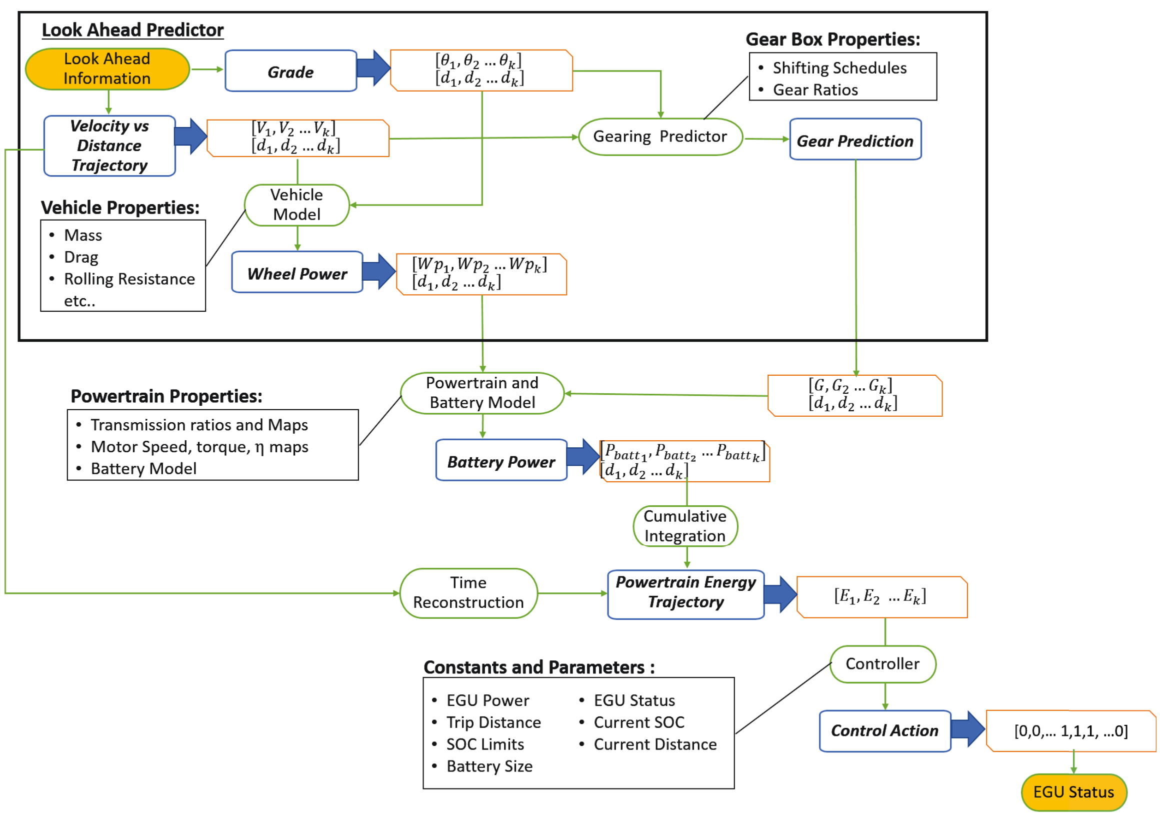

Figure 4.10: Integration of look ahead prediction into powertrain control

Calculation of Energy Requirement from Look Ahead Information

Numerical steps involved in calculating 'Powertrain Energy Trajectory' utilizing the above mentioned set of look ahead information is presented below. It is to be noted that the 'trip time' estimation is derived from look ahead data, using technique described in Section 4.4.2.

First, wheel power (W_{p_n}) is calculated from road load force (F_{wheels_n}). The calculation is done based on the known values of aerodynamic, inertial, and friction properties of the vehicle as shown :

$$
\begin{aligned}
F_{drag_n} &= -\Big[C_{aero} \cdot V_n^2 \cdot (V_n > 0)\Big] + \Big[C_{aero} \cdot V_n^2) \cdot (V_n \leq 0)\Big] \\
F_{roll_n} &= -\Big[(C_{roll_{st}} + C_{roll_{dyn}} \cdot V_n) \cdot cos(\theta_n) \cdot (V_n > 0)\Big] \\
&\quad + \Big[(C_{roll_{st}} + C_{roll_{dyn}} \cdot V_n) \cdot cos(\theta_n) \cdot (V_n \leq 0)\Big] \\
F_{grade_n} &= -\Big[M_v \cdot g \cdot sin(\theta_n) \cdot (V_n > 0)\Big] + \Big[M_v \cdot g \cdot sin(\theta_n) \cdot (V_n \leq 0)\Big] \\
F_{accel_n} &= -\Big[M_v \cdot \dot{V}_n \cdot (\dot{V}_n > 0)\Big] + \Big[M_v \cdot \dot{V}_n \cdot (\dot{V}_n \leq 0)\Big] \\
F_{wheels_n} &= -\Big[(F_{drag_n} + F_{roll_n} + F_{grade_n} + F_{accel_n}\Big] \\
\Rightarrow W_{p_n} &= F_{wheels_n} \cdot V_n
\end{aligned}
$$

Where,

F_{drag_n}	=	Aerodynamic friction force
F_{roll_n}	=	Rolling friction force
F_{grade_n}	=	Force caused by gravity when driving on non horizontal roads
F_{accel_n}	=	Inertial force
F_{wheels_n}	=	Net tractive force at wheels
C_{aero}	=	product of aerodynamic drag coefficient and vehicle's frontal area
$C_{roll_{st}}$	=	Coefficient of static friction between tire and road
$C_{roll_{dyn}}$	=	Coefficient of static friction between tire and road
g	=	acceleration due to gravity ($9.81 m/s^2$)
M_v	=	Mass of vehicle
V_n	=	Vehicle velocity (predicted)
θ_n	=	Road grade (inclination)

Next the required torque at the wheel (T_{wheels_n}) to meet road load requirements is computed:

$$T_{wheels_n} \quad = F_{wheels_n} \cdot r_{tyre}$$

Where,

$$\begin{array}{ll} T_{wheels_n} & = \text{Wheel torque required to meet road load } F_{wheels_n} \\ r_{tyre} & = \text{Tire radius} \end{array}$$

In order to compute the torque produced by the traction motor, gear selection is treated as a static map. The map defines discrete values of gear ratio (G_n)- as a function e of vehicle speed (V_n) and grade (θ_n). Knowing the value of final drive ratio ($gRatio_{FD}$), allows computing an overall transmission ratio ($transRatio_n$) between the traction motor and wheels. Torque(T_{EM_n}) produced by the motor is calculated by reflecting the wheel torque demand to the motor shaft - given the transmission ratio ($transRatio_n$). The computed motor torque is subjected to the maximum propulsion torque ($T_{EM_{max}}$) and minimum regeneration torque ($T_{EM_{min}}$) limits. $T_{EM_{min}}$ and $T_{EM_{max}}$ are defined as a function 'f' of motor shaft speed (w_{EM_n}).

$$\begin{aligned}
\Rightarrow G_n &= e(V_n, \theta_n) \\
transRatio_n &= G_n \cdot gRatio_{FD} \\
w_{EM_n} &= \frac{V_n \cdot transRatio_n}{r_{tyre}} \\
T_{EM_{min}}, T_{EM_{max}} &= f(w_{EM}) \\
T_{EM_n} &= \begin{cases} \dfrac{T_{wheels_n}}{transRatio_n \cdot trans\eta_n} & ; T_{EM} \leq T_{EM_{max}} \\ T_{EM_{max}} & ; T_{EM} > T_{EM_{max}} \\ T_{EM_{min}} & ; T_{EM} < T_{EM_{min}} \end{cases}
\end{aligned}$$

Where,

G_n	= Gear ratio (predicted)
e	= Map of gear ratio as function of vehicle speed, and grade
$transRatio_n$	= Overall transmission ratio
$gRatio_{FD}$	= Final drive ratio
w_{EM_n}	= Traction motor speed
$curr_{transRatio}$	= Gear ratio
$T_{EM_{min}}$	= Negative torque limit of traction motor
$T_{EM_{max}}$	= Positive torque limit of traction motor
f	= Map of torque limits as a function of traction motor speed

The traction motor's electric power (P_{EM_n}) is computed as a function of torque (T_{EM_n}), speed (w_{EM_n}) and efficiency (η). Where the motor's efficiency is defined as 2-dimensional map (g) of speed and torque.

$$\begin{aligned}\eta_{EM_n} &= g(w_{EM_n}, T_{EM_n}) \\ P_{EM_n} &= \left[\frac{T_{EM_n} \cdot w_{EM_n}}{\eta} \cdot (T_{EM_n} > 0)\right] + \left[T_{EM_n} \cdot w_{EM_n} \cdot \eta_{E_n} \cdot (T_{EM_n} \leq 0)\right]\end{aligned}$$

The battery model being used in this framework is a simplified first order model. In order to simplify commutations, the battery's charge and discharge capacity is ignored. Hence the traction motor's electric power (P_{EM_n}) is equated to the battery power(P_{batt_n}). The powertrain energy trajectory (E_n in watt-second) is eventually computed at this stage by integrating battery current (I_{batt_n},in ampere) and voltage (v_{batt_n}, in volts). Since this is a discrete integration process, the sum is multiplied by sample time (Ts).

$$
\begin{aligned}
\Rightarrow P_{batt_n} &= P_{EM_n} \\
v_{batt_n} &= \mathit{\Lambda}(OCV_{n-1}, SOC_{n-1}) \\
R_{batt} &= \Upsilon(P_{batt_n}, SOC_{n-1}, OCV_{n-1}) \\
\eta_{batt_n} &= \Phi(P_{batt}, SOC_{n-1}, OCV_{n-1}) \\
I_{batt_n} &= \eta_{batt_n} \cdot \left[v_{batt_n} - \frac{\sqrt{v_{batt_n}^2 - 4 \cdot R_{batt_n} \cdot P_{batt_n}}}{2 \cdot R_{batt_n}} \right] \\
\Rightarrow E_n &= \left[\sum_0^n (I_{batt_n} \cdot v_{batt_n}) \right] \cdot Ts
\end{aligned}
$$

Given the energy trajectory and battery capacity, the net energy (E_n) used by the powertrain in calculated as shown above. Here a rechargeable battery is the main source of energy, its SOC trend can be calculated as shown in Eq. 4.2. Using SOC prediction is advantageous over using predicted energy, as it allows the incorporation of battery charging efficiency.

$$
SOC_n = SOC_{n-1} - \frac{I_{batt_n}}{C_{nom}} \tag{4.2}
$$

For the drive cycle specified in Section 4.3.2 the predicted Battery and Wheel energy trends are shown in Figure 4.11.

From a numerical perceptive SOC is a representation of energy consumed in terms of a battery capacity. Hence in this framework, SOC is not subjected to minimum limits at this point. For example, if the initial SOC for a specific cycle is assumed to be 100% and the predicted ending SOC is -20%; it implies that total energy consumed by the vehicle is equal to 1.2 times the battery capacity.

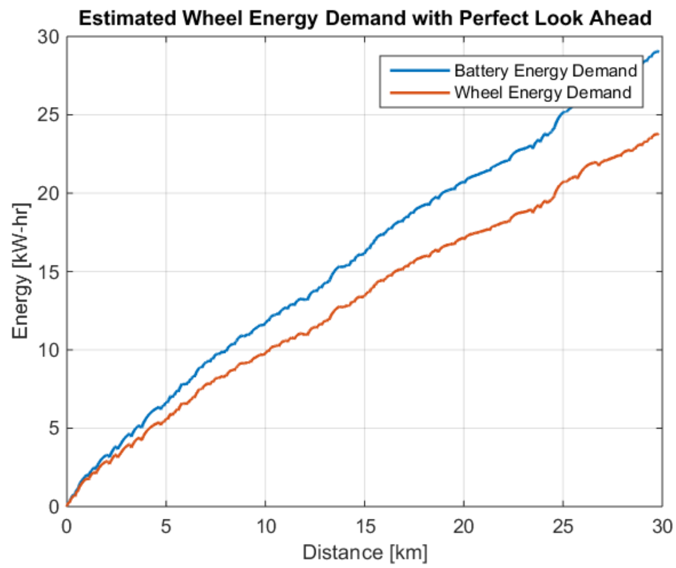

Figure 4.11: Integration of look ahead prediction into powertrain control

4.5 Suboptimal Controller - Delta Energy Controller

4.5.1 Introduction to Delta Energy

The delta energy controller is a control algorithm that utilizes look ahead information in order to maximize the energy efficiency for REEV's. This algorithm is framed to work with look ahead information in the form of velocity and wheel power predictions. The methodology to derive powertrain energy trajectory (E_n) from has been discussed in Section 4.4.3. For a REEV with fixed engine operating points, the primary factor in optimizing energy efficiency is reducing the number of engine crank events. DEC utilizes look ahead data to determine the optimum point at which engine should be turned ON and how long it should stay ON - such that the number of cranks is minimized, while adhering to SOC bounds of the battery. This section provides a formal representation of this strategy and demonstrates the effect of look ahead prediction quality on the vehicles fuel economy.

The DEC controller works on the reasoning that total energy required for cycle can be estimated from look ahead prediction. Net expenditure of battery energy is computed in the form of SOC. Given a desired end-of-trip SOC for the battery pack, the net energy that needs to added into the battery system from EGU is calculated from equation 4.3. This net energy is called '$DeltaEnergy$' ans is graphically illustrated in Figure 4.12.

$$\Delta E_{batt} = C_{nom} \times \Delta SOC \times \eta_{chrg} \tag{4.3}$$

In equation 4.3, ΔSOC is the difference between the predicted cycle-end-SOC and the desired SOC level at the end-of-trip. η_{chrg} is the average charging efficiency of the battery and C_{nom} is the nominal charge capacity of the battery.

An important distinction to be made is that 'Delta Energy' corresponds to the amount of energy that is added to the vehicle system over a trip and not the amount of energy added to the battery. In other words, 'Delta Energy' includes that proportion of EGU energy which is used towards propulsion and that which gets stored in the battery.

In this implementation, DEC considers a static charging and discharging efficiency for the battery and fixed EGU operating point -producing constant electrical output of 54.95kW. With this knowledge, DEC works to optimize the duration and phasing of energy addition. In the case illustrated above, the engine-generator unit (EGU) comes ON at about 7km after the trip begins and remains ON until the required amount of energy has been added to the system.

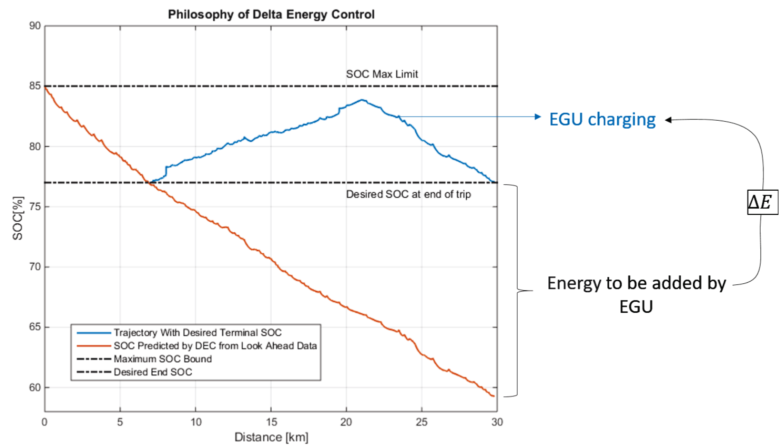

Figure 4.12: Illustration of the DEC algorithm with perfect look ahead information

4.5.2 Feasible Range for Energy Addition

One of the main objectives of DEC is to ensure the addition of required fuel energy (Delta Energy) in a manner such that SOC bounds are not violated. Figure 4.13 demonstrates 4 different cases of energy addition. In the first case, the energy addition begins very early in the trip, however given the initial SOC of the battery pack, its the maximum bounds is exceeded. On the contrary, if engine generation was activated after the boundary case - Case 4, the terminal SOC bounds cannot be satisfied as the trip concludes before required energy is added. Hence there exists a clearly defined 'feasible region' bounded by Case 2 and Case 4 within which the energy addition process should be done.

The feasible region is primarily determined by trip time since the EGU's output is fixed. Time taken to add a certain amount of energy remains fixed. Hence when determining the feasible region the lower bound (corresponding to Case 4 in

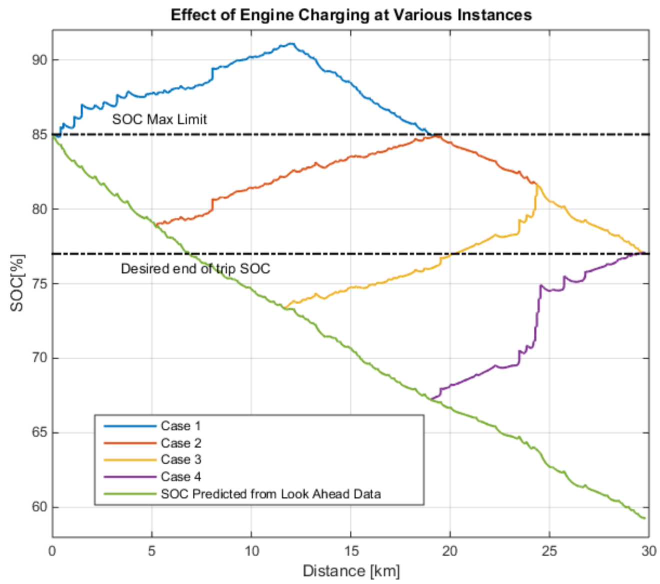

Figure 4.13: Feasible range for EGU charge addition

Figure 4.13) is constrained by the estimated time remaining in the trip. The algorithm has been defined in a manner such that, if a situation is encountered where the amount of charge to be added is such that it is infeasible to be accommodated at one engine Turn ON event, the charging process is split up into multiple phases as shown in Figure 4.14.

4.5.3 Mathematical Formulation of DEC

The problem of optimal control to minimize fuel consumption over 'N' steps, can be represented mathematically as shown below:

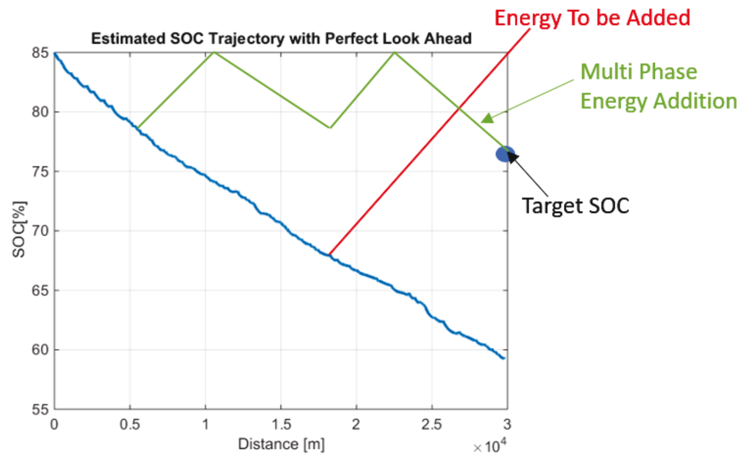

Figure 4.14: Multiphase EGU charge addition

$$
\begin{aligned}
J^* &= \underset{U}{\operatorname{argmin}}\{J\} \textit{ such that,} \\
J &= \sum_{k=n}^{N} \Big[\dot{m}_f(k) + \dot{m}_s(k) \Big] \\
\Rightarrow J &= \sum_{k=n}^{N} \Big[\dot{m}_f(\omega_e^*, T_e^*) x_2^2(k) + \dot{m}_s(k) H(x_2(k) - x_2(k-1)) \Big] \\
\Rightarrow J &= \dot{m}_f(\omega_e^*, T_e^*) \sum_{k=n}^{N} \Big[x_2^2(k) \Big] + \dot{m}_s \sum_{k=n}^{N} \Big[H(x_2(k) - x_2(k-1)) \Big]
\end{aligned}
$$

$$
\therefore J = \dot{m}_f(\omega_e^*, T_e^*) \sum_{k=n}^{N} \Big[x_2^2(k) + q \cdot H(u_p(k)) \Big] \tag{4.4}
$$

System states x_1 and x_2 are given by :

$$
\begin{aligned}
x_1(k+1) &= x_1(k) + x_2(k) \cdot \alpha_e(\omega_e^*, T_e^*) - W(k) \\
x_2(k+1) &= x_2(k) + u_p(k)
\end{aligned}
$$

Where,

$$
\begin{array}{rcl}
\dot{m}_s & = & \text{Start fuel penalty} \\
\dot{m}_f & = & \text{Fuel flow rate at } (\omega_e, T_e) \\
x_2 & = & \text{EGU State} \\
x_2 & \in & \{0, 1\} \\
T_e^*, \omega_e^* & \rightarrow & \text{Optimal Torque and Speed of EGU} \\
H & \Rightarrow & \text{Heaviside step function} \\
m & = & \{0, 1, 2, \ldots\} \text{ Determined iteratively by the controller} \\
U & = & \left\{ [u_0^T, u_1^T, u_2^T, ..., u_{p=2m}^T] \in \{0, 1\} : \sum_{p=1}^{2m} size(u_p) = N \right\} \\
q & = & \dfrac{\dot{m}_s}{\dot{m}_f} \\
x_1 & = & SOC \\
SOC_{min} & < & x_1 \ < \ SOC_{max} \\
\alpha_e & = & \text{Equivalent EGU power expressed as battery SOC} \\
W & \rightarrow & \text{Predicted SOC trajectory from equation 4.2}
\end{array}
$$

In the DEC strategy, the set of control inputs ‘U’ corresponds to EGU cranking status over ‘N’ points. Here a crank status ‘1’ corresponds to a ‘On’ command, on the contrary ‘0’ status corresponds to ‘Off’ command. The control input ‘U’ is divided into $2m + 1$ components $([u_0, u_1, u_2, ..., u_{2m}])$. ‘$m$’ is determined iteratively by the controller. The parameter ‘m’ defines the number of engine cranks required to add ΔE quantum of energy without exceeding SOC bounds of the battery. Each element ‘u_p’ in U is a column vector.Elements of ‘u_p’ (‘a’) can take one of the discrete values $\{1, 0\}$ as shown below:

DEC defines,

$$u_p = [a_{0_p}, a_{1_p} \ldots a_{j_{p_p}}]^T$$

Such that ,

$$a_{0_p}, a_{1_p}, \ldots a_{j_{p_p}} = \begin{cases} 0 & \text{; p is an odd number} \Rightarrow \sum [u_p] = 0 \\ 1 & \text{; p is an even number} \Rightarrow \sum [u_p] = j_p \end{cases}$$

$$\sum_{p=0}^{2m} j_{p_p} = N$$

Where,

$$
\begin{array}{rcl}
j_{p_p} & = & size(u_p) \\
p & = & 0, 1 \ldots 2m
\end{array}
$$

An initial value of '1' is assumed for 'm' and through numerical iteration over the cost function defined in equation 4.4 an minimum value is arrived upon. The computation of optimality is simplified since control can be defined with only $4m-1$ parameters, ie., m,$j_1, j_2, j_3 \ldots j_{2m-1}$, and value of $\sum(u_{j_1}), \sum(u_{j_2}), \ldots \sum(u_{2m-1})$. This is made possible with the realization that we only need to determine the lengths of vectors in u_p (Note : size(u_p)= j_p), however the values of elements in a given u_p can only assume one of the discrete values 0 or 1 based on the numerical value of 'p'. Optimization problem defined in Equation 4.4 is solved numerically using the DEC strategy described above.

For example, in the case illustrated in figure 4.12, to define the 1 charging event, the algorithm sets $m = 1$.

$$
\begin{aligned}
m &= 1 \\
\Rightarrow 2m-1 &= 3 \\
\therefore p &= \{0,1,2\} \\
\text{U can be defined as} & \\
U &= [u_0, u_1, u_2] \\
\text{Where } u_0, u_1, u_2 \text{ are such that :} & \\
size(u_0) &= j_0 \\
size(u_1) &= j_1 \\
size(u_2) &= j_2 \\
j_0 +_j 1 + j_2 &= N
\end{aligned}
$$

The values of $[u_0], [u_1]$ *and* $[u_2]$ are determined numerically as :

$$
\begin{aligned}
u_0 &= [0_{0_0}, 0_{1_0}, 0_{2_0} \ldots 0_{j0_0}] & \Rightarrow \sum[u_0] = 0 \\
u_1 &= [1_{0_1}, 1_{1_1}, 1_{2_1} \ldots 1_{j1_1}] & \Rightarrow \sum[u_1] = j_1 \\
u_2 &= [0_{0_2}, 0_{1_2}, 0_{2_2} \ldots 0_{j2_2}] & \Rightarrow \sum[u_2] = 0
\end{aligned}
$$

In the above example, system performance has been defined by parameters m, j_0, j_1, j_2, $\sum[u_0]$, $\sum[u_1]$ *and* $\sum[u_2]$. Thereby reducing the problem complexity from determining 'N' values for the input to only 7 parameters.

4.6 Simulation Results

Results from Traffic In Loop simulation of a Simulink based powertrain model along with traffic simulation in SUMO are presented here. The power train model is as described in section 4.2. The traffic distribution, simulation maps and route are defined in section 4.3. TIL is performed first on a REEV that operates with no look ahead capability, based on a thermostatic controller. The results from this simulation is presented in Section 4.6.2. Further, the rule based controller is replaced with DEC that draws look ahead information from various types of predictors as described in Section 4.4.1 and 4.4.3.

4.6.1 Trip Characterization

The predictions are made using known route information. Since the effect of traffic and other dynamic elements are not considered in the above predictors, the predictions remain constant for a given route. Upon defining the route and calibrating the traffic density along this route, as given in section 4.3 randomness was introduced into the driving scenarios by randomizing the phasing of traffic lights, order of vehicles introduced into the route etc. This causes variations between two driving scenarios. The randomization is performed by varying 'Seed' numbers before every simulation run in SUMO. By using this technique, infinite number of drive scenarios can be generated. For purposes of this work, we limit the study to 50 different scenarios. TIL co-simulation is performed over the 50 scenarios and results are collected from the simulations.

A sample velocity profile of the host vehicle in the is presented in figure 4.15. It is important to note that the trajectory described in Figure 4.15 refers to one of the 50 drive scenarios described above. Though the route remains the same, differences in the randomness of the virtual world in TIL causes variations in the velocity profiles of different scenarios. However, owing to lack of clarity in its representation, that behavior is not illustrated in this work. The above mentioned randomness causes variation in trip parameters between different scenarios. These parameters include total trip time, average velocity, number of stops and time spent following an other vehicle (leader time). Figure 4.16 elucidates the same.

This is seen as being akin to making multiple trips along a given route where the traffic density remains approximately the same at all times. Implying that the only variation between different trips is the duration of stop at each traffic light, the number of traffic lights that force a stop, and short term dynamics in speed introduced by leader vehicles. Hence each individual drive scenario is considered to be a trip. The words 'Trip' and 'Scenario' are used interchangeably in the remainder of this work. Subsequent sections of this thesis deals with studying the effects of incorporating look ahead information on the fuel economy of the vehicle. The statistical fuel economy benefits obtained through look ahead based control is then compared to economy values obtained from implementing a rule based controller that operates without any look ahead information.

4.6.2 Traffic In Loop Simulation Results with No Look Ahead

The first part of this section (section 4.6.3)demonstrates the working of a thermostatic rule-based controller using one trip. The same controller is then applied across 50 test trips as discussed above. The results from this activity are presented in section 4.6.4

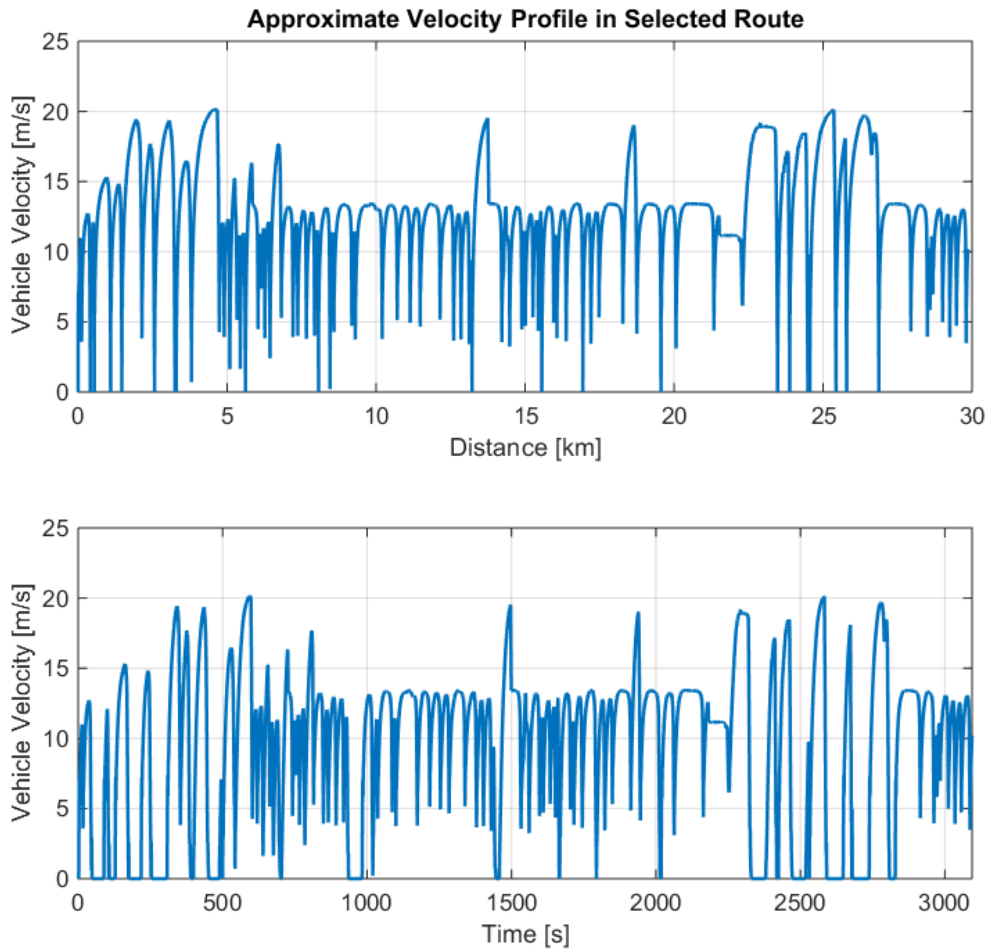

Figure 4.15: Sample of host vehicle's velocity profile

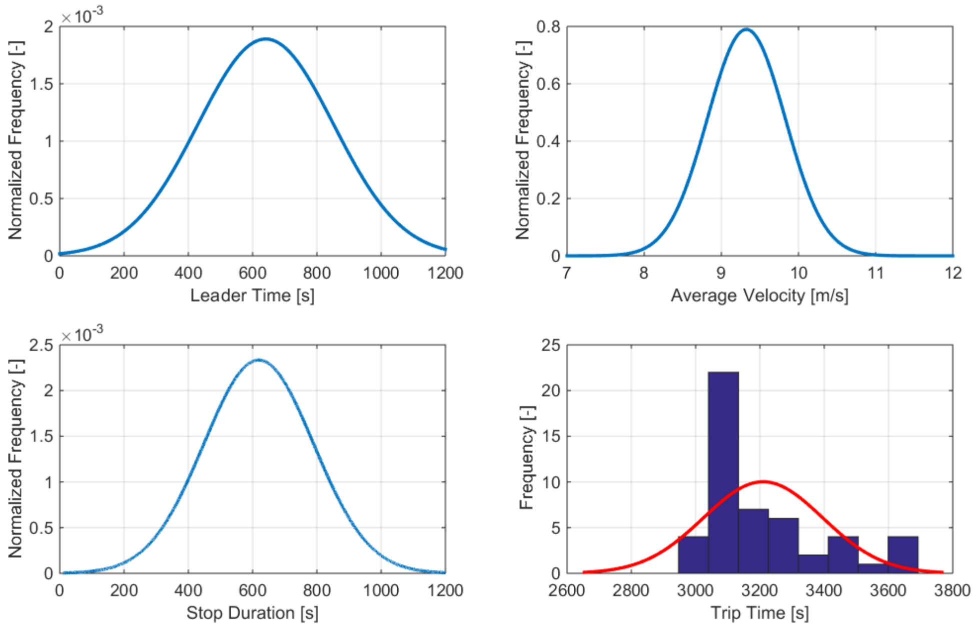

Figure 4.16: Variation of trip parameters across 50 drive scenarios

4.6.3 Thermostatic Controller

The thermostatic control used in this work is defined with through a reference curve for battery SOC. The reference is constructed to achieve a linear discharge from initial SOC of 85% to a desired 77% ending SOC. Reasoning behind choosing the end SOC as such is discussed in Section 4.2. The reference curve ('SOC Reference') is defined as a function of distance and is shown in Figure 4.17. Thermostatic controller is an ON -OFF in nature and tries to follow the given reference with in a bound of -1% and +0.5% by appropriately tunning ON and OFF the Engine-Generator Unit (EGU). When the battery SOC drops below 1% of the reference SOC at a given distance, the EGU is turned ON. As the vehicle continues to move, electric propulsive power is drawn from both the EGU and battery. Since the EGU operates at a fixed point,

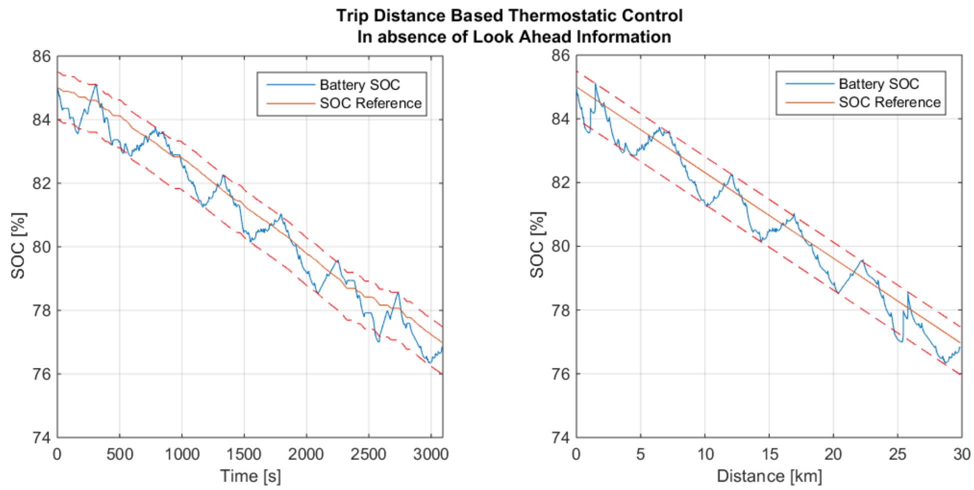

Figure 4.17: Actual SOC dynamics and distance based SOC reference curve

excess power output is used to charge the battery. When SOC has been recharged to +0.5% of the reference SOC at a given distance, the EGU is turned off. This behavior is illustrated in Figure 4.17.

Following the operation principle listed above, the thermostatic control tunrs on the EGU 7 times as shown in Figure 4.18. This can also be seen as sections with charging phases in the battery SOC dynamics curve shown in figure 4.18.

4.6.4 Fuel Economy Analysis

Figure 4.19 demonstrates the variation in fuel economy over 50 runs of the TIL simulation. The spread in economy numbers are normalized around their mean. The figure shows a ±2% variation around the mean. This spread in fuel consumption computed along the due to the variation in traffic and other external conditions that are introduced by randomization of the driving scenarios.

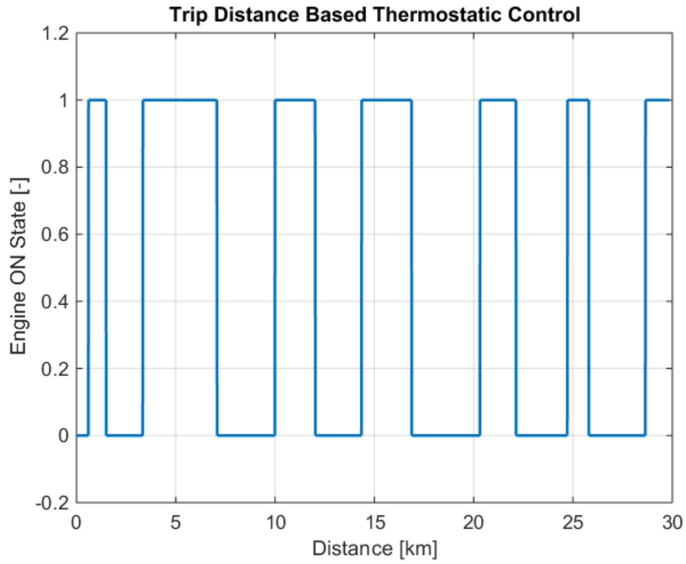

Figure 4.18: Actual SOC dynamics and SOC reference curve

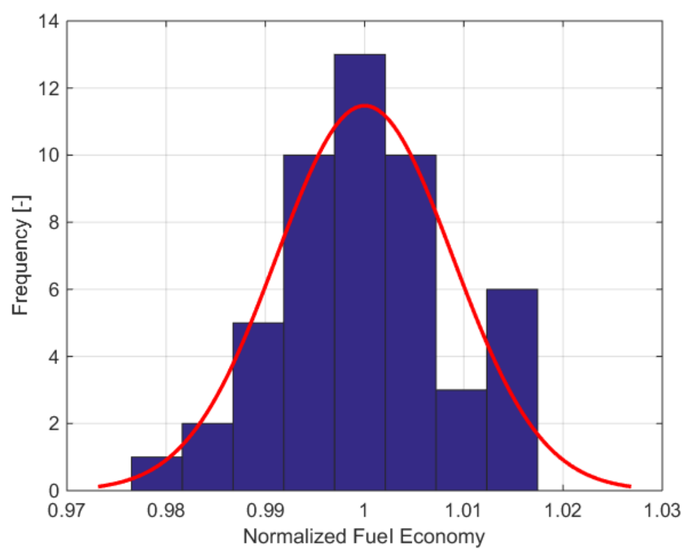

Figure 4.19: Fuel economy variation due to traffic and other external conditions

Given a particular trip, the energy required to complete it remains the same. Further, since SOC at end of trip is constrained the overall energy added by engine also remains constant. In this particular architecture, we assume a fixed operating point for the Engine. However each engine crank event requires more fuel than that consumed during constant operation as described in Section 4.2. This implies that net fuel consumption is primarily determined by the number of engine ON events that take place during a trip. We can thereby conclude that the primary objective behind further improving the fuel economy of this vehicle can be achieved by minimizing the number of engine cranks while maintaining the net ON time as a constant - in order to add said amount of battery charge and meeting SOC constrains (SOC at end of constrained at to be 77%).

4.6.5 Traffic In Loop Simulation Results with Delta Energy Controller

Section 4.4.1 examines various types of look ahead predictions that can be made by utilizing different types of data regarding vehicle's route. The effects of adding look ahead information on fuel economy is explored in this section. First, effects of incorporating look ahead information with the help of DEC is illustrated for an individual trip. Further the statistical distribution and variation of fuel economy is studied given different types of look ahead information. The types of predictors and the abbreviation used in the description of results are similar to that used in Figure 4.6 and are reiterated in table 4.1.

Look Ahead Types	
Type No.	**Description**
Type 1	(SL) Based of road speed limits.
Type 2	(SLSS) Based on road speed limits and locations of Stop signs and intersections.
Type 3	(SLSSLT) Based on road speed limits, locations of Stop signs and intersections, and turns.
Type 4	(V2S Enabled) Based on speed and power information as communicated by a prior vehicles along same route

Table 4.1: Summary of look ahead information types used in this thesis

4.6.6 Effect of Look Ahead Information on Individual Trip

It can be seen in Figures 4.20 that in the absence of any form of look ahead information, the engine is cranked 7 times in this particular trip. However the type 1 predictors despite begin the lowest form of look ahead information, reduce the number of cranks to 5 events for over the trip. Further as speed stop sign and turn information are added the number of engine ON events reduces to 2 and 1 times respectively. For this particular cycle, we notice that both type 4 and type 3 information yield only one engine crank event. Figure 4.21 illustrates the battery SOC trajectory of the vehicle with and without look ahead information. At first glance the SOC trajectories with look ahead information seem to share a similar form. But closer observation yields that they reflect the EGU status from Figure 4.20.

4.6.7 Fuel Economy Analysis With Different Type of Look Ahead Information

As a continuation of the above study, look ahead information based DEC controller is applied across a number of trips (random drive scenarios). The distribution of fuel

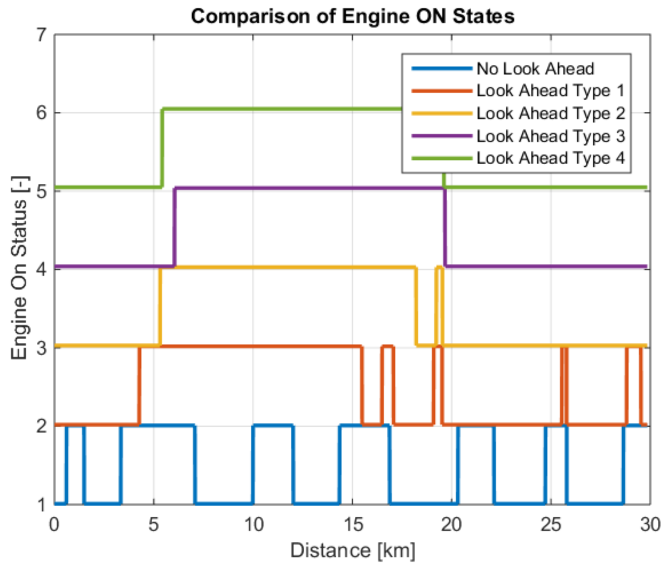

Figure 4.20: Reduction in number of engine cranks with addition of look ahead information

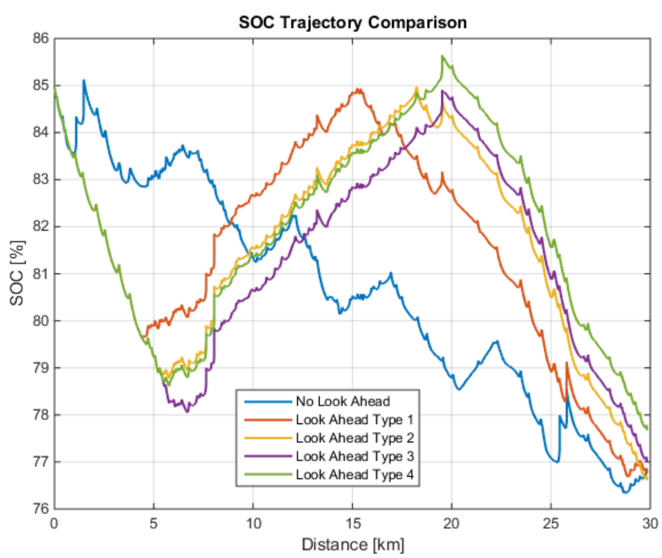

Figure 4.21: SOC trajectories with look ahead information

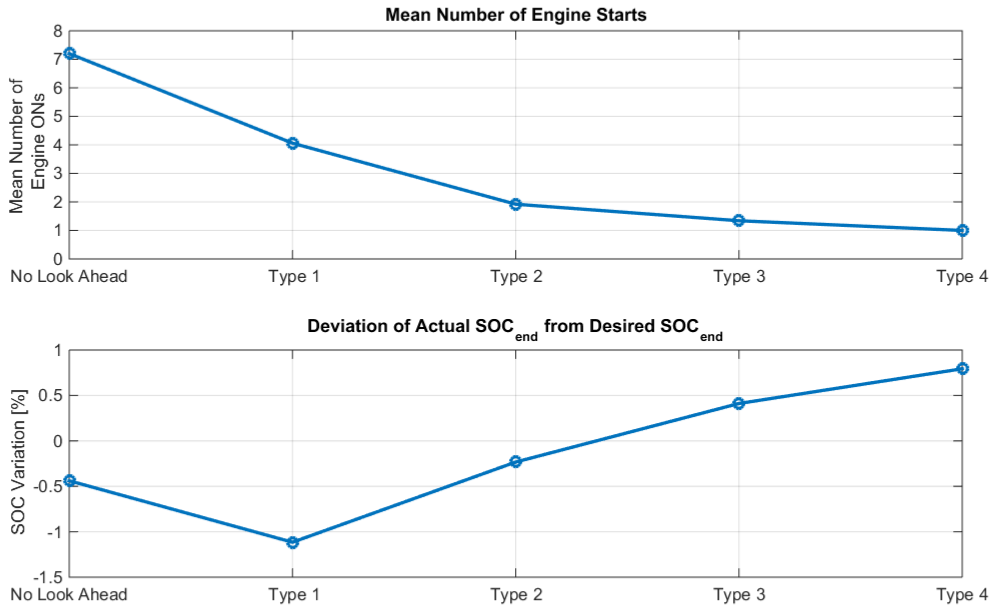

Figure 4.22: Mean number of engine startups and mean difference between the desired and actual SOC at the end of Cycle

economy from this is compared to that obtained in the absence of any look ahead information, as presented in figure 4.19. With improving the quality of look ahead information progressively from type 1 to type 4, the mean number of engine start ups over 50 tips reduces. However none of these predictors are capable of yielding perfect look ahead information for each trip leading to a quantifiable difference between the predicted and actual behavior of the system. Effect of these uncertainties manifest themselves as deviation in the final SOC compared to the desired SOC value.

The deviation in final SOC from desired end SOC is compensated by computing a static equivalent fuel consumption. The process for re-calculating vehicle's fuel economy for each case is done by using equations as shown in equations 4.5 to 4.9 :

$$\Delta E_{batt} = C \times \Delta SOC \times \eta_{chrg} \tag{4.5}$$

Equations 4.5 gives the overall difference in the actual versus desired battery energy level. Where, ΔE_{batt} is the equivalent battery energy and C is battery capacity in Ah. ΔSOC refers to difference between desired and actual end of trip SOC. This difference in energy is converted to equivalent engine ON time though equation 4.6

$$\Delta T_{EGU} = \frac{E_{batt}}{(P_{EGU} \times \eta_{Gen})} \tag{4.6}$$

In equation 4.6, η_{chrg} refers to efficiency of battery charging and η_{EGU} is the overall efficiency of generator. P_{EGU} is the EGU power output in W and ΔT_{EGU} corresponds to time required by EGU to generate ΔE_{batt} joules of energy and is measured in seconds. Hence given the fuel flow rate into the engine, $\dot{V}$; Δf the compensatory fuel mass can be calculated. However, if $\Delta T_{EGU} > 0$: then it can be said that the EGU needs to be run for extra time to compensate for lower Ending SOC than desired.

$$\Delta f = \Delta T_{EGU} \times \dot{V} \tag{4.7}$$

Else if $\Delta T_{EGU} < 0$, It can be considered that the extra energy in the battery could have been used to provide propulsive power. This is calculated with approximate estimates of the traction motor efficiency η_{tracEM} and battery discharge efficiency $\eta_{dischrg}$. Therefore resulting in the equation :

$$\Delta f = \Delta T_{EGU} \times \dot{V} \times \eta_{dischrg} \times \eta_{tracEM} \tag{4.8}$$

The Δf hence calculated is added to the overall fuel consumption of the trip and the fuel economy is recalculated to result in the equivalent fuel consumption MPG_{eq} as shown in equation 4.9.

$$MPG_{eq} = \frac{Dist_{trip}}{\Delta f + V_{fuel}} \tag{4.9}$$

All reference to fuel economy in this work refers to the equivalent fuel consumption thats as calculated using the above method. Figure 4.23 shows variation of improvement in fuel economy with look ahead information. For the sake of comparison, improvements are referenced to performance of the rule based controller as discussed in Section 4.6.3 and Figure 4.19.

The fuel consumption numbers plotted in Figure 4.23 correspond the equivalent fuel economy as computed based on equations 4.5 and 4.6 based on difference in the trip's ending SOC and desired ending SOC of 77%.

Figure 4.25 is a modified representation of the distributions given in Figure 4.23. It helps conclude that incorporation of look ahead information into powertrain control has quantifiable benefits. But the rate of improvement in system efficiency, decreases with increase quality of the look ahead information. This claim is supported by simulation based fuel economy improvement results shown in Figure 4.25. The improvement in fuel economy is caused by reduction in number of engine cranks with better look ahead information. This is shown in figure 4.22.

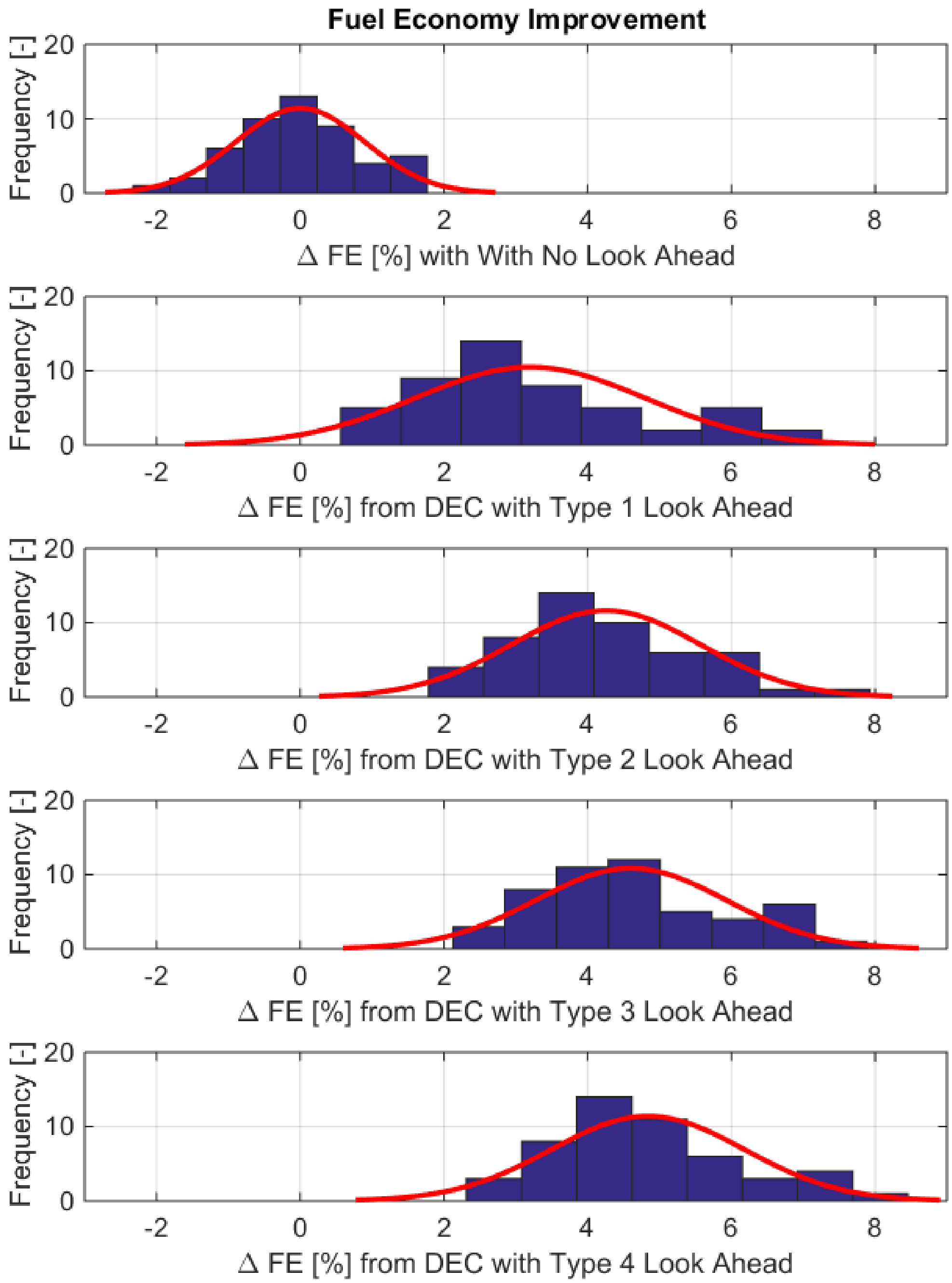

Figure 4.23: Distribution of improvement in fuel economy with look ahead information

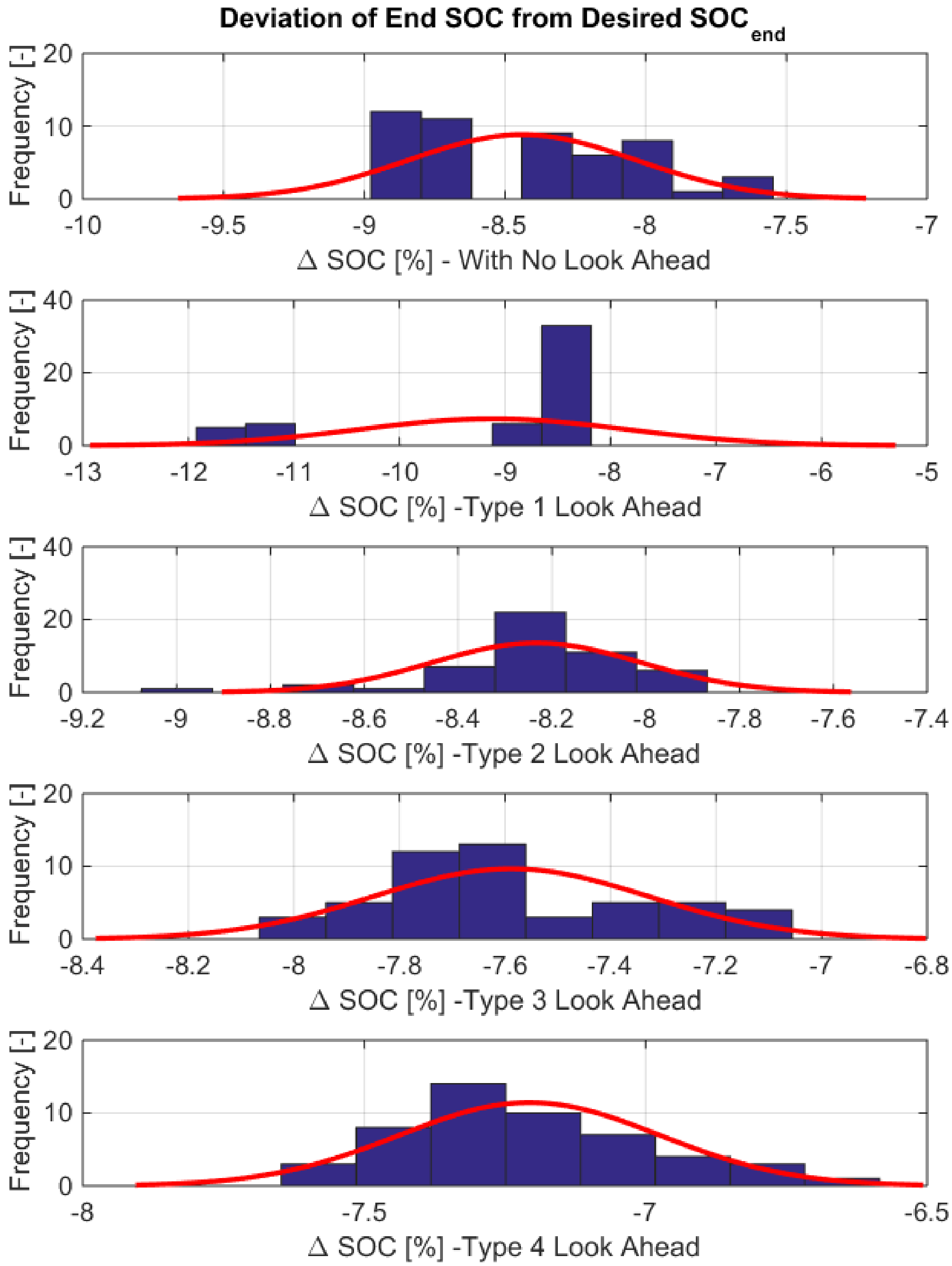

Figure 4.24: Distribution of end of trip SOC with look ahead information

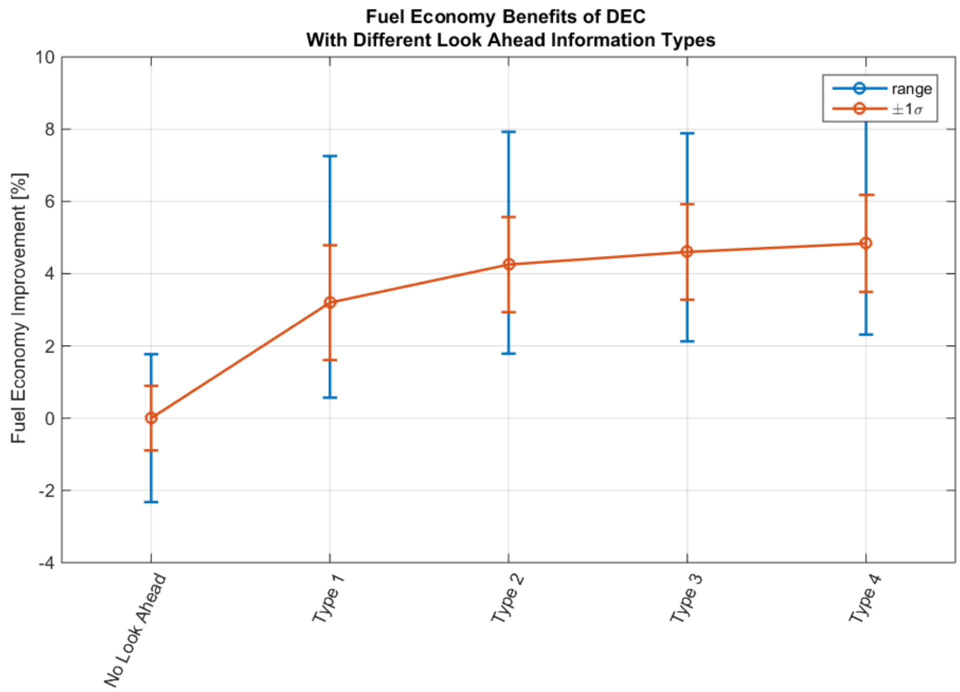

Figure 4.25: Fuel economy improvements with incremental addition of look ahead information

4.7 Chapter Summary

This chapter introduced a REEV model that was used to study the effect of look ahead data, on fuel economy improvement. First, a framework was introduced that enables the use of look ahead forecasts in powertrain control. Further 4 different types of look ahead predictors were discussed. Figure 4.26 gives an overview of simulation structure used.

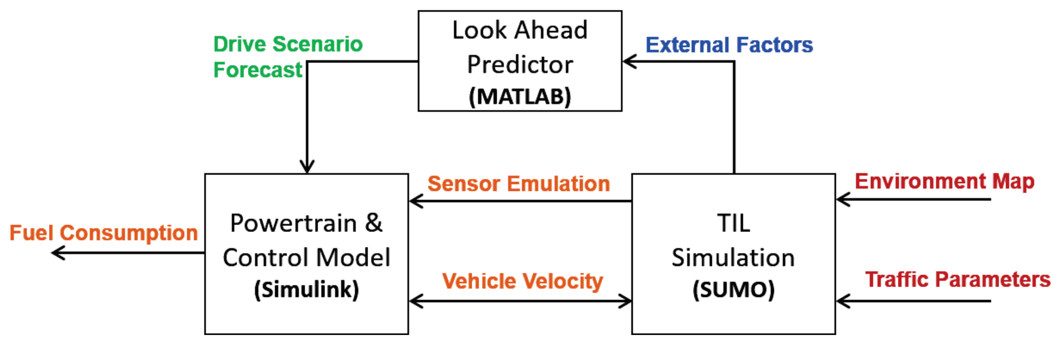

Figure 4.26: Structure of look ahead based powertrain simulation

A control strategy called 'Delta Energy Controller' was developed that uses look ahead data to improve a REEV's fuel economy. This controller tries to minimize engine start events to improve powertrain's efficiency. Fuel economy benefits derived from using different qualities of look ahead forecasts along with the DEC strategy were explored. Simulation results show that progressive improvement in look ahead data leads to a diminishing increase in fuel economy benefits, thus forming a Pareto front.

Chapter 5: Conclusion and Future Work

5.1 Conclusions

This thesis has described a methodology for the development of a look ahead based energy management system, for a range extended electric vehicle. In this control philosophy, environmental data is used to construct a forecast of the future driving scenario. Knowledge of these expected conditions is used to improve powertrain's fuel efficiency. Various technologies that can be used to enable environmental data collection on a vehicle, are termed as 'Technology Enablers'. A consolidated study of various technologies was conducted and presented in Chapter 2. The introduction of these technologies onto physical vehicle systems are guided by a process called 'Preferential Technology Loading' and was demonstrated through the creation of 'Sensor Packages'.

Powertrain simulation techniques used today were found to be deficient in mimicking real-world driving as detailed in Chapter 3. Hence the simulation of real world driving scenarios was facilitated by implementing a traffic integrated powertrain co-simulation technique, called Traffic In Loop (TIL) simulation as also disused in Chapter 3. The TIL framework accounts for factors external to the powertrain that influence fuel consumption. It provides a simulation platform that aids in virtual on-road testing for purposes of control development, and studying vehicle behavior from a transportation system's perspective.

By leveraging the TIL simulation, an effective look ahead based energy management controller has been designed in Chapter 4. A procedural framework to integrate look ahead predictions into powertrain control is also discussed. The effectiveness of this framework and optimality of the controller is proven through simulations, that show improvement in fuel economy with better look ahead data. However, the improvements were observed to diminish incrementally with better forecasts thus forming a pareto front.

5.2 Future Work

The optimality of DEC as been proven by logical reasoning and testing over 50 drive scenarios. However, in order to formalize the findings, a comparison needs to be drawn with reference to solutions obtained from dynamic programing; in the presence of perfect a-priori information. In its current sate, DEC algorithm operates the engine at a fixed point (most optimum point). Hence providing a binary control on charge addition. This control logic can be expanded to include a flexible engine operation region. Additional efforts need to be directed towards improving look ahead forecasts by accounting for traffic, and information from short term sensors like RADARs.

Look ahead energy management strategy has been explored for a range extended electric vehicle traveling along a specific route. However the benefits exhibited need to be verified over other routes under different traffic densities. The TIL platform enables mimicking realistic driving and hence can be used to create 'real-world like' duty cycles. Thus it can aid in performing component life cycle analysis under realistic scenarios.

Additionally, capability to include infrastructure in-loop with powertrain simulation, allows studying the effects of transportation infrastructure on vehicle performance and component sizing. For example, battery size determination of transit buses can be performed based on a city's transportation infrastructure, location of charging stations etc. Additionally, control development and testing for applications similar to 'geo-fencing' can also be performed using the TIL framework.

Bibliography

[1] J. I. Meguro, T. Murata, J. I. Takiguchi, Y. Amano, and T. Hashizume, "GPS multipath mitigation for urban area using omnidirectional infrared camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 22–30, 2009.

[2] J.-P. Carrasco, A. d. l. E. de la Escalera, J. M. Armingol, *et al.*, "Recognition stage for a speed supervisor based on road sign detection," *Sensors*, vol. 12, no. 9, pp. 12153–12168, 2012.

[3] Ü. Özgüner, T. Acarman, and K. A. Redmill, *Autonomous ground vehicles*. Artech House, 2011.

[4] C. Stiller, "Intelligent vehicle technologies, theory and applications," 2001.

[5] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1258–1271, 2002.

[6] G. Cebrat, "Collaborative strategic energy management of serial-hybrid electric urban buses in operation," in *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, pp. 713–718, IEEE, 2013.

[7] R. Rasshofer and K. Gresser, "Automotive radar and lidar systems for next generation driver assistance functions," *Advances in Radio Science*, vol. 3, no. B. 4, pp. 205–209, 2005.

[8] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[9] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid electric vehicles: Energy management strategies*. Springer, 2016.

[10] S. Huff, B. West, and J. Thomas, "Effects of air conditioner use on real-world fuel economy," tech. rep., SAE Technical Paper, 2013.

[11] H. Lohse-Busch, M. Duoba, E. Rask, K. Stutenberg, V. Gowri, L. Slezak, and D. Anderson, "Ambient temperature (20 f, 72 f and 95 f) impact on fuel and energy consumption for several conventional vehicles, hybrid and plug-in hybrid electric vehicles and battery electric vehicle," tech. rep., SAE Technical Paper, 2013.

[12] J. F. Thomas, S. P. Huff, L. G. Moore, and B. H. West, "Measurement of vehicle air conditioning pull-down period," tech. rep., Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). National Transportation Research Center (NTRC), 2016.

[13] J. Thomas, S. Huff, and B. West, "Fuel economy and emissions effects of low tire pressure, open windows, roof top and hitch-mounted cargo, and trailer," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 7, no. 2014-01-1614, pp. 862–872, 2014.

[14] R. Verma, N. Nahar, Z. Tang, and B. Saltsman, "A driver assistance system for improving commercial vehicle fuel economy," tech. rep., SAE Technical Paper, 2013.

[15] H. Huo, Z. Yao, K. He, and X. Yu, "Fuel consumption rates of passenger cars in china: labels versus real-world," *Energy Policy*, vol. 39, no. 11, pp. 7130–7135, 2011.

[16] N. S. C. for Environmental Publications, "Final technical support document - fuel economy labeling of motor vehicle revisions to improve calculation of fuel economy estimates," Tech. Rep. 420R06017, U.S. Environmental Protection Agency, 2006. Accessed : 15 February 2017.

[17] N. S. C. for Environmental Publications, "Response to comments: Fuel economy labeling of motor vehicles," Tech. Rep. 420R06016, U.S. Environmental Protection Agency, 2006. Accessed : 15 February 2017.

[18] ABMARC, "Real world driving fuel efficiency and emissions testing," tech. rep., Australian Automobile Association, 2016. Accessed : 15 February 2017.

[19] P. Mock, J. German, A. Bandivadekar, I. Riemersma, N. Ligterink, and U. Lambrecht, "From laboratory to road: A comparison of official and real-world fuel consumption and co2 values for cars in europe and the united states," tech. rep., International Council on Clean Transportation, 2013. Accessed : 15 February 2017.

[20] P. Sharer, R. Leydier, and A. Rousseau, "Impact of drive cycle aggressiveness and speed on hevs fuel consumption sensitivity," tech. rep., SAE Technical Paper, 2007.

[21] M. G. Shirk and B. M. Geller, "Factors affecting the fuel consumption of plug-in hybrid electric vehicles," in *The 25th International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exposition*, 2010.

[22] S. T. Partnership, "Smartway transport glance at clean freight strategies driver training," 2002. EPA Document Ref.No. EPA-420-F-16-030, Accessed: 2017-08-30.

[23] United States Environmental Protection Agency, *Smartway Transport Partnership: U.S. EPA and NRCan Official Signing of the Memorandum of Understanding and Licensing Agreement: Fact Sheet*, National Service Center for Environmental Publications (NSCEP), 2005. EPA Document Ref. No., EPA420-F-05-04.

[24] N. R. Council *et al.*, *Technologies and approaches to reducing the fuel consumption of medium-and heavy-duty vehicles.* National Academies Press, 2010.

[25] D. Lang, R. Schmied, and L. Del Re, "Prediction of preceding driver behavior for fuel efficient cooperative adaptive cruise control," *SAE International Journal of Engines*, vol. 7, no. 2014-01-0298, pp. 14–20, 2014.

[26] Ü. Özgüner, T. Acarman, and K. A. Redmill, *Autonomous ground vehicles.* Artech House, 2011.

[27] A. Janowski, A. Nowak, M. Przyborski, and J. Szulwic, "Mobile indicators in gis and gps positioning accuracy in cities," in *International Conference on Rough Sets and Intelligent Systems Paradigms*, pp. 309–318, Springer, 2014.

[28] T. Williams, P. Alves, G. Lachapelle, and C. Basnayake, "Evaluation of gps-based methods of relative positioning for automotive safety applications," *Transportation research part C: emerging technologies*, vol. 23, pp. 98–108, 2012.

[29] M. G. Petovello, *Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation.* University of Calgary, Department of Geomatics Engineering, 2003.

[30] S. Godha, "Performance evaluation of low cost mems-based imu integrated with gps for land vehicle navigation application," *UCGE report*, vol. 20239, 2006.

[31] S. Sukkarieh, "Low cost, high integrity, aided inertial navigation systems for autonomous land vehicles," *The University of Sydney*, 2000.

[32] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation techniques: principles and applications.* Springer Science & Business Media, 2012.

[33] R. S. Harvey, *Development of a precision pointing system using an integrated multi-sensor approach.* University of Calgary, 1998.

[34] J. Stephen, "Development of a multi-sensor gnss based vehicle navigation system," *UCGE Reports*, no. 20140, 2000.

[35] A. Nawrat, K. Jedrasiak, K. Daniec, and R. Koteras, "Inertial navigation systems and its practical applications," in *New Approach of Indoor and Outdoor Localization Systems*, InTech, 2012.

[36] A. Lawrence, *Modern inertial technology: navigation, guidance, and control.* Springer Science & Business Media, 2012.

[37] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, "Mapping large loops with a single hand-held camera.," in *Robotics: Science and Systems*, vol. 2, 2007.

[38] H. Kim, C.-S. Lin, J. Song, and H. Chae, "Distance measurement using a single camera with a rotating mirror," *International Journal of Control, Automation, and Systems*, vol. 3, no. 4, pp. 542–551, 2005.

[39] J. Mrovlje and D. Vran, "Distance measuring based on stereoscopic pictures," *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, vol. 2, no. October, pp. 1–6, 2008.

[40] M. Oliveira, V. Santos, and A. D. Sappa, "Multimodal inverse perspective mapping," *Information Fusion*, vol. 24, pp. 108–121, 2015.

[41] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognitionhow far are we from the solution?," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–8, IEEE, 2013.

[42] C. G. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff, "Real-time recognition of us speed signs," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 518–523, IEEE, 2008.

[43] W. Liu, Y. Wu, J. Lv, H. Yuan, and H. Zhao, "Us speed limit sign detection and recognition from image sequences," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, pp. 1437–1442, IEEE, 2012.

[44] M.-Y. Fu and Y.-S. Huang, "A survey of traffic sign recognition," in *Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on*, pp. 119–124, IEEE, 2010.

[45] Mobileye, "system-on-chip (SoC) the EyeQ® family," October 2017.

[46] K. Litomisky, "Consumer rgb-d cameras and their applications," *Rapport technique, University of California*, vol. 20, 2012.

[47] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *Experimental robotics*, pp. 477–491, Springer, 2014.

[48] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[49] Ü. Özgüner, T. Acarman, and K. A. Redmill, *Autonomous ground vehicles*. Artech House, 2011.

[50] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2377–2396, 2015.

[51] C. Schindelhauer, "Mobility in wireless networks," in *International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 100–116, Springer, 2006.

[52] L. E. Y. Mimbela and L. A. Klein, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems," tech. rep., Department of Transportation, Joint Program Office for Intelligent Transportation Systems, 2000.

[53] L. A. Klein, M. K. Mills, and D. R. Gibson, "Traffic detector handbook: -volume ii," tech. rep., U.S Department of Transportation, Federal Highway Administration, 2006.

[54] A. Rhodes, D. Bullock, J. Sturdevant, Z. Clark, and D. Candey, "Evaluation of stop bar video detection accuracy at signalized intersections," *Transportation Research Board*, pp. 05–0384, 2006.

[55] L. Klein, "Data requirements and sensor technologies for its," *Norwood, MA, Artech House*, 2001.

[56] Z. Jian and L. Xingwu, "Research on laser doppler velocimeter for vehicle self-contained inertial navigation system," *Optics & Laser Technology*, vol. 42, no. 3, pp. 477–483, 2010.

[57] N. Abulizi, A. Kawamura, K. Tomiyama, and S. Fujita, "Measuring and evaluating of road roughness conditions with a compact road profiler and arcgis," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 3, no. 5, pp. 398–411, 2016.

[58] Y. Du, C. Liu, D. Wu, and S. Jiang, "Measurement of international roughness index by using-axis accelerometers and gps," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[59] R. Robinson, "Slippery road detection and evaluation," tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 2012. Accessed on : 20 October 2017.

[60] M. Ruiz-Llata and P. Acedo, "Remote ice detection system for on-board applications based on fiber optics," in *Proc. of SPIE Vol*, vol. 7753, pp. 77535N–1, 2011.

[61] C. Ciamberlini, G. Innocenti, and G. Longobardi, "An optoelectronic prototype for the detection of road surface conditions," *Review of scientific instruments*, vol. 66, no. 3, pp. 2684–2689, 1995.

[62] C. Manzie, H. Watson, and S. Halgamuge, "Fuel economy improvements for urban driving: Hybrid vs. intelligent vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 1, pp. 1–16, 2007.

[63] R. Verma, "Look-ahead driver feedback and powertrain management," tech. rep., Eaton Corporation, Menomonee Falls, WI (United States), 2014.

[64] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, 2009.

[65] W. Huang, D. M. Bevly, S. Schnick, and X. Li, "Using 3d road geometry to optimize heavy truck fuel efficiency," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pp. 334–339, IEEE, 2008.

[66] N. J. Kohut, J. K. Hedrick, and F. Borrelli, "Integrating traffic data and model predictive control to improve fuel economy," *IFAC Proceedings Volumes*, vol. 42, no. 15, pp. 155–160, 2009.

[67] H. Durrant-Whyte, "A critical review of the state-of-the-art in autonomous land vehicle systems and technology," *Albuquerque (NM) andLivermore (CA), USA: SandiaNationalLaboratories*, vol. 41, 2001.

[68] M. Zhao, A. Mammeri, and A. Boukerche, "Distance measurement system for smart vehicles," in *New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on*, pp. 1–5, IEEE, 2015.

[69] J. Flacke, K. Heckl, and B. Kaiser, "Verfahren zur winkelbemessung mittels eines mehrfachantennenradarsystems," *European Patent EP 0*, vol. 989, no. 416, p. A2, 1999.

[70] D. Si, S. C. Luo, C. Pero, X. Wu, and R. Knox, "Millimeter-wave fmcw/monopulse radar front-end for automotive applications," in *Microwave Symposium Digest, 1999 IEEE MTT-S International*, vol. 1, pp. 277–280, IEEE, 1999.

[71] B. D. Cordill, S. A. Seguin, and L. Cohen, "Electromagnetic interference to radar receivers due to in-band ofdm communications systems," in *Electromagnetic Compatibility (EMC), 2013 IEEE International Symposium on*, pp. 72–75, IEEE, 2013.

[72] D. E. Kerr, *Propagation of short radio waves*, vol. 24. IET, 1951.

[73] B. R. Bean, E. Dutton, and B. Warner, "Weather effects on radar," *Radar Handbook," McGraw-Hill Book Company, New York*, pp. 24–1, 1970.

[74] I. S. Merrill *et al.*, "Introduction to radar systems," *Mc Grow-Hill*, 2001.

[75] R. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science*, vol. 9, no. B. 2, pp. 49–60, 2011.

[76] J. L. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 674–680, IEEE, 1989.

[77] H. Xinhan and W. Min, "Multi-sensor data fusion structures in autonomous systems: a review," in *Intelligent Control. 2003 IEEE International Symposium on*, pp. 817–821, IEEE, 2003.

[78] L. A. Klein, *Sensor and data fusion concepts and applications*. Society of Photo-Optical Instrumentation Engineers (SPIE), 1993.

[79] R. C. Luo, C. C. Chang, and C. C. Lai, "Multisensor fusion and integration: Theories, applications, and its perspectives," *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3122–3138, 2011.

[80] S. C. Thomopoulos, "Sensor integration and data fusion," *Journal of Field Robotics*, vol. 7, no. 3, pp. 337–372, 1990.

[81] A. G. Mutambara, *Decentralized estimation and control for multisensor systems*. CRC press, 1998.

[82] A. R. Vallur, Y. Khairate, and C. Awate, "Prescriptive modeling, simulation and performance analysis of mild hybrid vehicle and component optimization," tech. rep., SAE Technical Paper, 2015.

[83] P. Chen and J. Wang, "Integrated diesel engine and selective catalytic reduction system active no x control for fuel economy improvement," in *American Control Conference (ACC), 2013*, pp. 2196–2201, IEEE, 2013.

[84] E. Lieberman and A. K. Rathi, "Traffic simulation," in *Traffic Flow Theory - A State-of-the-Art Report*, ch. 10, pp. 10–1–10–25, Committee on Traffic Flow Theory and Characteristics, 2001. Accessed On : 30-Aug-2017.

[85] A. Bajpai and T. V. Mathew, "Development of an interface between signal controller and traffic simulator," in *Conference of Transportation Research Group of India (CTRG)*, (Bangalore, India), 2011. Accessed : 30-Aug-2017.

[86] P. A. Ehlert and L. J. Rothkrantz, "Microscopic traffic simulation with reactive driving agents," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 860–865, IEEE, 2001.

[87] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espié, "A behavioral multi-agent model for road traffic simulation," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1443–1454, 2008.

[88] J. Maroto, E. Delso, J. Felez, and J. M. Cabanellas, "Real-time traffic simulation with a microscopic model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 513–527, 2006.

[89] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility)-an open-source traffic simulation," in *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pp. 183–187, 2002.

[90] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th communications and networking simulation symposium*, pp. 155–163, ACM, 2008.

[91] A. F. Acosta, J. E. Espinosa, and J. Espinosa, "Traci4matlab: Enabling the integration of the sumo road traffic simulator and matlab® through a software re-engineering process," in *Modeling Mobility with Open Data*, pp. 155–170, Springer, 2015.

[92] J. Macedo, Z. Kokkinogenis, G. Soares, D. Perrotta, and R. J. Rossetti, "A hla-based multi-resolution approach to simulating electric vehicles in simulink and sumo," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pp. 2367–2372, IEEE, 2013.

[93] J. Soares, B. Canizes, C. Lobo, Z. Vale, and H. Morais, "Electric vehicle scenario simulator tool for smart grid operators," *Energies*, vol. 5, no. 6, pp. 1881–1899, 2012.

[94] A. Wegener, H. Hellbruck, C. Wewetzer, and A. Lubke, "Vanet simulation environment with feedback loop and its application to traffic light assistance," in *GLOBECOM Workshops, 2008 IEEE*, IEEE, 2008.

[95] P. MacNeille, O. Gusikhin, M. Jennings, C. Soto, and S. Rapolu, "Integration of traffic simulation and propulsion modeling to estimate energy consumption for battery electric vehicles," in *Simulation and Modeling Methodologies, Technologies and Applications*, pp. 3–19, Springer, 2013.

[96] D. Fredette, J. Jing, and U. Ozguner, "Intelligent vehicle fuel saving technologies: Comparing three primary categories of methods," in *ASME 2015 Dynamic Systems and Control Conference*, pp. V002T31A003–V002T31A003, American Society of Mechanical Engineers, 2015.

[97] "Transportation data management system." `http://odot.ms2soft.com/tcds/`, Accessed : 26 September 2017.

[98] "Introduction to demand modeling in sumo." `http://sumo.dlr.de/w/index.php?title=Demand/Routes_from_Observation_Points&oldid=9674`, Accessed : 26 September 2017.