Abstract

The EcoCAR3 competition challenges student teams to redesign a 2016 Chevrolet Camaro to reduce environmental impacts and increase energy efficiency while maintaining performance and safety that consumers expect from a Camaro. Energy management of the new hybrid powertrain is an integral component of the overall efficiency of the car and is a prime focus of Colorado State University’s (CSU) Vehicle Innovation Team. Previous research has shown that error-less predictions about future driving characteristics can be used to more efficiently manage hybrid powertrains. In this study, a novel, real-world implementable energy management strategy is investigated for use in the EcoCAR3 Hybrid Camaro. This strategy uses a Nonlinear Autoregressive Artificial Neural Network with Exogenous inputs (NARX Artificial Neural Network) trained with real-world driving data from a selected drive cycle to predict future vehicle speeds along that drive cycle. Various prediction windows are analyzed and compared to quantify tradeoffs between prediction window size and speed prediction error for a given drive cycle. To investigate the fuel economy (FE) improvement potential of this new control strategy, a high fidelity model of a Toyota Prius, developed by Colorado State University, is used. An optimal dynamic programming (DP) engine controller is implemented in the Prius model. Several exemplar controllers are studied for the specified drive cycle: the model baseline controller, a DP derived engine controller using NARX Artificial Neural Network speed predictions, and a DP derived engine controller using a 100% accurate speed prediction. These simulations allow for investigation into the tradeoffs between different prediction window sizes. Additionally, the results provide insight into what FE benefit can be expected from speed prediction compared to baseline and idealized conditions. This potentially achievable FE benefit is used as motivation to develop a predictive controller that can be implemented in real-time on the supervisory controller of CSU’s Plug-in Hybrid Electric (PHEV) Camaro.

Introduction

The need for cleaner and more efficient forms of transportation is well understood and documented. Hybrid Electric (HEV), Plug-in Hybrid Electric and Electric (EV) vehicles have been gaining momentum in the U.S. as well as globally [1, 2, 3, 4]. While current HEV and PHEV are more efficient and less damaging to the environment than conventional vehicles, there are significant improvements to hybrids that can be made. One improvement would be to have hybrids in a larger range of vehicle classes, including performance vehicles. The EcoCAR3 competition is working to address that issue. The EcoCAR3 competition involves 16 colleges across North America, including Colorado State University, and is funded by General Motors and the Department of Energy, among others. Schools are competing to redesign a 2016 Chevrolet Camaro into a hybrid. The goal is to reduce energy consumption and greenhouse gas emissions, while maintaining the performance, safety and utility that consumers demand in a Camaro. One of the main focuses of the project is the development of the Energy Management Strategy (EMS) for the new hybrid powertrain. The EMS is one of the most important factors in determining the FE of a hybrid [5].

One strategy to increase the FE of the Camaro is through real-time prediction. Research showing an increase in fuel economy due to prediction of vehicle operating conditions (such as vehicle speed, battery SOC, torque command) from environmental knowledge (such as traffic conditions, traffic signals, road types and grade) has been well documented [6, 7, 8, 9, 10]. These methods often don’t reach on-board implementation due to the difficulty of obtaining and communicating accurate environmental information to the vehicle.

Researchers are working to obtain many different environmental conditions on-board to improve fuel economy. Conditions such as traffic signals [10], road grade [11], road type, and driving style [12], among others, are being explored. Future vehicle speeds are complex and difficult to predict. The researchers in [12] calculate an initial, approximate speed trace based on speed limits and then calculate the deviation from that in real time. This study presents a more straightforward method to predict future speeds in real time using a NARX Artificial Neural Network.

Neural Networks (NN) are used in a wide range of applications in research related to improving FE in HEVs. [9] uses NN for optimal battery current levels in the EMS of PHEVs. Current research at
Colorado State University is showing NNs can be effective in replacing rules based models to predict fuel consumption. [13, 14, 15] all use NN as part of the intelligent online EMS. This current reseach leverages NN’s versatility to predict future vehicle speeds.

Dynamic programming as a means of deriving the optimal control for a given state space is well understood [16] and its application to the optimal HEV energy management is well documented in literature articles [6,17, 18, 19, 20] and textbooks [21-22]. The drawback of dynamic programming is that it is computationally costly and thus difficult to implement in real time HEV energy management [11]. As a result, research has moved towards more implementable versions of this theory such as stochastic dynamic programming [23, 24, 25, 26, 27] and model predictive control [7-8,28-29]. Because of the difficulties of real-world implementation of dynamic programming researchers now mainly use it as a convenient (offline calculable) measure of the globally optimal results [30-31].

This research utilizes a NARX Artificial Neural Network to make vehicle speed predictions. These predictions serve as an input to a predictive powertrain controller that uses dynamic programming to optimize engine control for the speed prediction. This prediction can be implemented in real time and only needs limited previous driving data, and knowledge of the destination to be implemented. The predictive powertrain controller is computationally expensive, but is suitable to evaluate the maximum FE benefit that can be realized from the speed predictions. These speed predictions have the capability to be implemented as a component of the fuel economy optimization control on the hybrid supervisory controller (HSC) of Colorado State University’s PHEV Chevrolet Camaro.

### Methods

Vehicle speed prediction aims to maximize fuel economy by predicting the vehicle speed for the upcoming segment of time and optimizing the engine control for that prediction window, all in real time. Speed predictions are made using a NARX Artificial Neural Network. To investigate the tradeoff between prediction window size and the deviation between predicted and actual vehicle speeds, a range of prediction window sizes are evaluated.

To evaluate the benefit of future vehicle speed predictions, a high-fidelity model of a generation 3 Toyota Prius, developed by Colorado State University, is used. The input to the simulation is the desired prediction window size and real world drive cycle data. The baseline HSC of the Prius model is adapted to include a NARX Artificial Neural Network and a predictive powertrain controller for speed predictions and to derive optimal engine control based on those speed predictions, respectively. This is completed repeatedly as the model is driving the drive cycle. The output of the simulation is the vehicle’s fuel consumption.

The goal of this study is to compare the vehicle FE for differing prediction window sizes, along with a baseline with no speed prediction and an idealized case with perfect speed predictions. It is important to note that this study is not related to Eco-routing, the driver is controlling the vehicle, and this work focuses on powertrain control.

### Baseline Vehicle Fuel Economy Modeling

The baseline vehicle controller and vehicle plant model operate on an equation-based algorithm. The vehicle plant and baseline controller are models of a generation 3 Toyota Prius in the MATLAB/Simulink language. To display its similarity to an actual generation 3 Toyota Prius, a simulation of three relevant EPA standard drive cycles was developed. The results were used to validate the performance of the FE model and are shown in Table 1 [22].

<table>
<thead>
<tr>
<th></th>
<th>UDDS</th>
<th>HWFET</th>
<th>US06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>69.6</td>
<td>67.3</td>
<td>43.5</td>
</tr>
<tr>
<td>Simulation</td>
<td>70.9</td>
<td>64.5</td>
<td>42.6</td>
</tr>
</tbody>
</table>

Based on the similarity to real world data, the results of the baseline simulation are considered validated for the purpose of predicitating fuel economy of a generation 3 HEV Prius.

### Drive Cycle Development

Existing EPA standard drive cycles aim to capture a mixture of generic city and highway driving. In order to attempt to capture a similar mix, through a shortened drive cycle, a custom cycle in Fort Collins, Colorado is developed. A shorter drive cycle is desired because the route needs to be driven multiple times. The route is shown in Figure 1.

---

### Table 1. Comparison of Baseline Model and Experimental FE

<table>
<thead>
<tr>
<th></th>
<th>UDDS</th>
<th>HWFET</th>
<th>US06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>69.6</td>
<td>67.3</td>
<td>43.5</td>
</tr>
<tr>
<td>Simulation</td>
<td>70.9</td>
<td>64.5</td>
<td>42.6</td>
</tr>
</tbody>
</table>

Figure 1. Custom Drive Cycle Route

Vehicle speed and GPS location are recorded from the Controller Area Network (CAN) bus during each trip along the drive cycle. An example speed trace is shown in Figure 2. Note that the letters show the correspondence between location and speed trace.

To collect data to train the NARX Artificial Neural Network, this drive cycle is driven a total of 10 times, on different days and at different times of day to get as much variation as possible. Data from 8 of the cycles is used to train, validate and test the NARX Artificial Neural Network and data from the other 2 is used in the simulations. To develop a baseline FE for use as a comparison to speed prediction...
FE, a simulation to drive the custom drive cycle using the baseline controller was developed. This baseline simulation provides an important comparison point for the optimized FE results.

Figure 2. Speed trace of custom drive cycle

**Neural Network Vehicle Speed Predictions**

As stated above, a NARX Artificial Neural Network is used to make vehicle speed predictions based on current, past and future GPS locations and past vehicle speeds. Since the NARX Artificial Neural Networks are trained with actual driving data, it should be noted that these predictions will be representative of how the driver drives. Thus, if the driver drives aggressively, the NARX Artificial Neural Network will predict aggressive driving behavior.

The exogenous inputs to the NARX Artificial Neural Network are the vehicle’s longitude and latitude. The output of the NARX Artificial Neural Network is the vehicle speed. Only one hidden neuron layer is used. This is shown in Figure 3. There was no pre-processing of the data before using it to train the NARX Artificial Neural Network.

![Figure 3. Closed Loop NARX ARTIFICIAL NEURAL NETWORK](image3)

A different NARX Artificial Neural Network is needed for each prediction window size. The method of developing each NARX Artificial Neural Network is the same, but the parameters are different. The general method for developing the NARX Artificial Neural Network will be explained, and then the method for determining the parameters to be used for each NARX Artificial Neural Network will be discussed.

The parameters which are to be changed for each NARX Artificial Neural Network are prediction window size, the number of hidden neurons, and the number of input and feedback delays. The number of input and feedback delays are set equal to each other in each NARX Artificial Neural Network. Each NARX Artificial Neural Network only has one hidden layer, uses Levenberg-Marquardt backpropagation to update weights, a hyperbolic tangent sigmoid transfer function for the hidden layer and a linear transfer function for the output layer.

Once the parameters are set for the NARX Artificial Neural Network, it is then trained using the entire cycle data from the 8 drive cycles on an open loop. In a closed loop NARX Artificial Neural Network, the output of the current time step is used as an input for the next time step. An open loop differs in that rather than using the output from the previous time step as the input, it uses the actual, known, target value. This makes training of the NARX Artificial Neural Network faster and more efficient. Once training is complete, the NARX Artificial Neural Network is then closed. The input of vehicle speed is now taken from the output of the NARX Artificial Neural Network from the previous time step. An example of an open loop NARX Artificial Neural Network is shown in Figure 4. It can be compared to the closed loop NARX Artificial Neural Network depicted in Figure 3.

![Figure 4. Open Loop NARX Artificial Neural Network](image4)

To decide the best NARX Artificial Neural Network parameters for each prediction window, many feasible combinations of hidden neurons and delays are used to predict the drive cycle in segments that correspond to prediction window size. The range of hidden neurons explored is from 6-16 and delays from 2-26 seconds. The combination which produces the lowest average mean square error over the entire drive cycle (not used at all for training) is chosen to be used for that prediction window size.

Once the training is completed, the NARX Artificial Neural Network is ready to make vehicle speed predictions over prediction windows. It is assumed that knowledge of the route which is being driven is available, and such future GPS locations are available to be used by the NARX Artificial Neural Network.

**Development of Predictive Powertrain Controller**

A predictive engine controller that was developed in previous research at Colorado State University [30,31] is leveraged as a foundation in this research to determine optimal engine control based on predicted vehicle speeds. The controller uses dynamic programming (DP) to evaluate all possible states and determine the optimal engine power for each state. The states are the State of Charge (SOC) of the battery and the time. The input of the DP algorithm is the speed trace and the output is a table of optimal engine power for all combinations of SOC and time steps. The optimal engine power is found by minimizing the fuel consumed over the drive cycle. One important constraint of the algorithm is that the SOC at the end of the speed trace is set to be the same as the SOC at the beginning, to simulate a charge sustaining situation.

**Implementation of Prediction and Predictive Powertrain Controller into Model**

To evaluate the benefit of predicting future vehicle speeds, the prediction and predictive powertrain controller are implemented into the HSC of the model so speed predictions and engine control can be developed as the simulated vehicle is driving. This is a main focus of the research, as we are seeking to make this algorithm real-world implementable with current technology.
The baseline controller in the model is adapted to have the capability to use the NARX Artificial Neural Network to make speed predictions for the upcoming vehicle speed using previous vehicle speed and GPS location. The predicted speed is then input into the DP algorithm to calculate the optimal engine power for each SOC and time over the upcoming prediction window size. This table is formatted as a two dimensional table lookup that the vehicle controller uses. The input to the lookup is the current SOC and time elapsed since the prediction was made. The output of the lookup table is the engine power demand. If the optimal engine power demand is zero, the engine demand is set to “off.” The engine demand is set to “on” when the engine power demand is above the threshold set for engine turn on.

This routine is repeated at 1 Hz to ensure that the maximum realizable FE potential is achieved. This does not diminish the benefits of having a longer prediction window. The DP algorithm is run over the entire prediction window at each time step, so it determines the optimal engine control for that entire prediction window. This ensures vehicle information is as up to date as possible and this same method is used for the idealized cases of perfect speed predictions. However, this method differs from making a prediction for the prediction window and then using that control for the entire prediction window.

Figure 5 shows the flow of information between the NARX Artificial Neural Network, the predictive powertrain controller, the modified baseline controller and the vehicle plant model.

These simulations make it possible to evaluate the trade-offs of different prediction window sizes. If the prediction window is very short, accurate speed predictions can be made. However, there is a limited benefit that the DP algorithm can realize over a short prediction window. Conversely, with longer prediction windows, the DP algorithm can find more optimal ways to operate the engine to minimize the fuel consumed over the prediction window. However, with longer prediction windows, the speed predictions will be less accurate and the DP algorithm will be optimizing for speeds which the vehicle may not actually travel.

Simulations of different prediction window sizes are developed to explore these trade-offs. Simulations for prediction window sizes of 3, 5, 8, 10, 15, 20, 30, 45 and 60 seconds are developed. In addition, some idealized cases are explored. Simulations where the speed prediction is removed and instead the actual speed trace is used as an input to the predictive powertrain controller is developed for the same array of prediction windows. These represent cases in which perfect speed predictions could be made.

Results and Discussion

Before evaluating the FE benefit of different prediction window sizes, a better understanding of the tradeoffs between prediction window size and prediction quality needs to be obtained.

**Tradeoffs Between Prediction Window Size and Prediction Quality**

As one might imagine, with a shorter prediction window, more accurate predictions can be achieved. Conversely, longer prediction window sizes result in larger speed prediction errors. Speed prediction error is calculated as the difference in the predicted and actual speed at the end of the prediction window. Figure 6 shows the speed prediction error distribution for different prediction window sizes, along with the standard deviation of prediction error.

![Figure 6. Error Distribution for Differing Prediction Window Sizes](image-url)

As the prediction window size grows, the prediction error distribution becomes larger. For the larger prediction window sizes, the prediction error reaches a saturation point because there is an inherent limit that is reached. For example, if the vehicle speed in the training dataset never eclipses 30 m/s, 80 m/s will not be predicted.
While the speed error reaches a saturation point, the distribution of the prediction errors continues to grow as the prediction window size grows. Figure 7 shows the distribution of prediction errors for a shorter and longer prediction window. For shorter prediction windows, the prediction errors are more concentrated, and are centered on 0 error. As the prediction window size increases, the errors become less concentrated, with a larger standard deviation in the error predictions.

Figure 7. Prediction Error Distribution of 5 and 60 Second Prediction Windows

While more accurate predictions are desirable, there is a drawback to shorter prediction window sizes. Shorter speed predictions result in less information to supply to the predictive powertrain controller. Thus, while shorter prediction windows provide more accurate predictions, they are inherently limited in the FE gain that can be achieved. If a prediction window is too short, there will not be enough information available to realize a FE benefit over the baseline controller. If the prediction window is too large and there is too much prediction error, the predictive powertrain controller will be optimizing for incorrect speed predictions and again, no FE benefit will be realized. It is also possible that this prediction can result in a FE decrease over the baseline controller. This could occur if the prediction window is so short that the optimization routine is too limited by the SOC constraint that it cannot find FE benefit. Also, a loss of FE can occur if the speed predictions are too erroneous that optimal engine power vastly differs from what is needed to power the vehicle. However, these are extreme cases, and only happen at very short or very long prediction window sizes. A moderate range of prediction window sizes are explored to find the prediction window size which yields the largest FE benefit.

**FE Benefit of Different Prediction Window Sizes**

With the implementation of the speed prediction and predictive powertrain controller and a better understanding of the tradeoffs between prediction window size and prediction error, we now seek to develop a simulation based quantification of the fuel economy benefit as a function of prediction window size. A simulation of the custom drive cycle using the baseline controller was developed as a baseline to be used as a comparison. Additionally, simulations of the custom drive cycle using the predictive powertrain controller with perfect speed “predictions” (the actual speed trace) were developed for each prediction window size. This serves as a best case scenario representing the FE benefit that could be realized from this predictive powertrain controller if perfect predictions could be achieved.

Then, simulations are developed for each of the prediction window sizes using speed prediction. As described in the *Methods* section, a NARX Artificial Neural Network, trained for each specific prediction window size, outputs speed predictions which the predictive powertrain controller uses as inputs to develop the optimal engine control for that speed prediction. This is done in real time at 1 Hz in the simulation, while the model is driving the drive cycle. A comparison of engine power between the baseline and the simulation using a 20 second prediction window is shown in Figure 8. Note this is a subset of the drive cycle, as the cycle is over 20 minutes long.

The predictive powertrain controller leverages the knowledge of future speeds to realize FE benefit by keeping the engine off as much as possible, which is displayed between 535-545 seconds. When the engine is turned on, it does so when more power is needed and operates along the ideal operating line (IOL). High efficiencies are
achieved at higher engine power. The result is that the engine tends to run at a higher power than the baseline when it is operating, displayed between 550-565 seconds in Figure 8.

To evaluate the FE benefit from each simulation, the fuel consumed during the drive cycle, along with the SOC and distance traveled, are extracted from each simulation. The SAEJ1711 recommended practice is used to calculate the miles per gallon equivalent (MPGe).

Due to the stochastic nature of Neural Network training, each time the NARX Neural Network is retrained, it will produce a slightly different NARX Artificial Neural Network. To account for this, each prediction window size had simulations completed 5 times to capture this variety in NARX Neural Network training outcomes.

The average MPGe for each of the prediction windows is calculated and compared to the baseline simulation. These results are displayed in Figure 9 and Table 2 (in the appendix). Figure 10 is a box and whisker plot to capture the variation introduced by NARX Artificial Neural Network training. This displays both the stochastic nature of NARX Artificial Neural Network as well as the overall trend of how the prediction window size impacts the FE benefit.

Figure 9. Perfect Speed Prediction and Average NARX Artificial Neural Network Prediction Percent System Energy Savings over Baseline Simulation

To understand how much of an impact speed prediction error has on FE benefit, the idealized case of simulations using the predictive powertrain controller with perfect speed predictions are also investigated and shown in Figure 9 and Figure 10. This represents a ceiling for the FE benefit that can be achieved with this predictive powertrain controller. Figure 9 and Figure 10 show that much of the FE benefit which can be achieved using this controller is actually achieved when making speed predictions with the NARX Artificial Neural Network.

Figure 10. Percent System Energy Saved over Baseline Simulation as a function of Prediction Window Size

It can be seen for this particular drive cycle, that 30 second prediction window results in the largest average FE benefit. However, 20 and 45 second prediction windows produce virtually the same FE benefit. These represent the best balance of prediction window size and prediction accuracy for the prediction windows explored. Prediction windows smaller than this realize less FE benefit because less information being supplied to the predictive powertrain controller. Prediction windows larger have a higher likelihood of predicting erroneously, which is shown by the larger maximum and minimum trends for the longer second prediction windows.

It should also be noted that there exists a saturation point where larger prediction windows, even with perfect prediction, will not realize larger FE benefits. This is caused by system limitations, such as the size of the energy storage system (ESS). Predicting further into the future than how much energy the ESS can hold, will not gain more FE because the predictive powertrain controller will have enough information to find the optimal solution for all possible ways to charge/discharge the ESS.

As was mentioned in the Methods section, there is data from two drive cycles that was acquired but not used in any way for the NARX Artificial Neural Network training. The same method as described above was completed on this other data set. The results are shown in the appendix in Table 4 and Table 5, Figure 11, and Figure 12. In general, this data set shows a similar trend. Short prediction windows are not as effective as prediction windows sizes around 30 seconds. In this drive cycle, extremely short prediction windows have an adverse effect on FE. However, these two drive cycles are not enough to make strong conclusions about all predictions and how robust they are.
Conclusions

This research shows that this method of real world implementable speed prediction can yield FE benefits for hybrid electric vehicles over the baseline controller when coupled with a predictive powertrain controller. The results also show that a large portion of the FE that could be realized from perfect speed prediction can be achieved with speed prediction from a NARX Neural Network trained with previous real world driving data from the drive cycle. This indicates that even with prediction error, FE gains can be realized.

This method can have a wide variety of applications. In this study, information about the entire drive cycle is used over the entirety of the cycle. However, the application is not limited by that. If there is previous data recorded for a segment of the route, this prediction could still be leveraged, thus information about the whole route is not necessarily needed. It should be noted that there are situations in which this method would not be fruitful. For example, if the SOC of the battery is too low, the predictive powertrain controller would be limited in how the engine can be operated. Similarly, during highway driving situations, it is typically necessary to have the engine operating the majority of the time.

The results of this study suggest that for this type of mixed driving prediction windows around 30 seconds provide the best potential for FE benefits in this vehicle. These represent the best tradeoff between gaining enough future information (prediction size) and making speed predictions which are accurate enough to still consistently realize FE benefits (prediction quality). However, it is difficult to make generalized claims based on how exactly prediction window size and prediction error will affect FE benefits from this study alone.

Additionally, technological advances, such as vehicle to vehicle or vehicle to infrastructure communication, should increase the capability to accurately predict future speeds for larger prediction windows. This methodology, which can be implemented using today’s technology, could continue to provide more FE benefit as predicting capability increases.

Future Work

This work serves as motivation to implement real time speed prediction and a predictive powertrain controller on the HSC of the PHEV Camaro developed at Colorado State University. To complete this, a predictive powertrain controller that is less computationally expensive than the DP algorithm currently used needs to be developed.

Additionally, an investigation of the FE benefit which could be realized from Vehicle to Vehicle communication is going to be explored. This will be compared to, and combined with, the current speed prediction method to investigate the impacts on the accuracy of the speed predictions.

References


Contact Information

David Baker
David.a.baker16@gmail.com
Ph: 716-353-5584

Acknowledgements

I would like to thank Dr. Thomas Bradley for all the guidance and support on this research. I would also like to thank General Motor, Argonne National Labs and all other sponsors of the EcoCAR 3 competition for providing all the resources to make this work possible. Thanks to all co-authors for their contributions and help writing this article.

Definitions/Abbreviations

CSU - Colorado State University
DP - dynamic programming
EMS - energy management system
ESS - energy storage system
EV - electric vehicle
FE - fuel economy
HEV - hybrid electric vehicle
HSC - hybrid supervisory controller
IOL - ideal operating line
MPGe - miles per gallon equivalent
NARX - nonlinear autoregressive with exogenous inputs
NN - neural network
PHEV - plug-in hybrid electric vehicle
SOC - state of charge
## APPENDIX

Table 2. Baseline and NARX Artificial Neural Network Prediction Simulation FE for different prediction window sizes

<table>
<thead>
<tr>
<th>Prediction Window Size</th>
<th>NN Prediction MPGe</th>
<th>Percent system energy saved over baseline simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (no prediction)</td>
<td>57.4</td>
<td>0.0%</td>
</tr>
<tr>
<td>3 second</td>
<td>65.1</td>
<td>1.3%</td>
</tr>
<tr>
<td>5 second</td>
<td>65.2</td>
<td>1.5%</td>
</tr>
<tr>
<td>8 second</td>
<td>65.7</td>
<td>2.2%</td>
</tr>
<tr>
<td>10 second</td>
<td>66.0</td>
<td>2.7%</td>
</tr>
<tr>
<td>15 second</td>
<td>67.1</td>
<td>4.4%</td>
</tr>
<tr>
<td>20 second</td>
<td>67.7</td>
<td>5.3%</td>
</tr>
<tr>
<td>30 second</td>
<td>67.8</td>
<td>5.5%</td>
</tr>
<tr>
<td>45 second</td>
<td>67.7</td>
<td>5.4%</td>
</tr>
<tr>
<td>60 second</td>
<td>67.3</td>
<td>4.7%</td>
</tr>
<tr>
<td>75 second</td>
<td>67.8</td>
<td>5.5%</td>
</tr>
<tr>
<td>90 second</td>
<td>67.2</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 3. Baseline and Perfect Prediction Simulation FE for different prediction window sizes

<table>
<thead>
<tr>
<th>Prediction Window Size</th>
<th>Perfect Prediction MPGe</th>
<th>Percent system energy saved over baseline simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (no prediction)</td>
<td>57.4</td>
<td>0.0%</td>
</tr>
<tr>
<td>3 second</td>
<td>64.5</td>
<td>0.2%</td>
</tr>
<tr>
<td>5 second</td>
<td>66.0</td>
<td>2.6%</td>
</tr>
<tr>
<td>8 second</td>
<td>67.0</td>
<td>4.1%</td>
</tr>
<tr>
<td>10 second</td>
<td>67.8</td>
<td>5.4%</td>
</tr>
<tr>
<td>15 second</td>
<td>68.6</td>
<td>6.6%</td>
</tr>
<tr>
<td>20 second</td>
<td>68.4</td>
<td>6.3%</td>
</tr>
<tr>
<td>30 second</td>
<td>68.5</td>
<td>6.5%</td>
</tr>
<tr>
<td>45 second</td>
<td>69.5</td>
<td>8.0%</td>
</tr>
<tr>
<td>60 second</td>
<td>69.2</td>
<td>7.6%</td>
</tr>
<tr>
<td>75 second</td>
<td>69.3</td>
<td>7.7%</td>
</tr>
<tr>
<td>90 second</td>
<td>69.0</td>
<td>7.2%</td>
</tr>
</tbody>
</table>
Table 4. Baseline and NARX Artificial Neural Network Prediction Simulation FE for different prediction window sizes for another drive cycle of same route

<table>
<thead>
<tr>
<th>Prediction Window Size</th>
<th>NN Prediction MPGe</th>
<th>Percent system energy saved over baseline simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (no prediction)</td>
<td>64.5</td>
<td>0.0%</td>
</tr>
<tr>
<td>3 second</td>
<td>64.2</td>
<td>-0.6%</td>
</tr>
<tr>
<td>5 second</td>
<td>64.4</td>
<td>-0.1%</td>
</tr>
<tr>
<td>8 second</td>
<td>65.3</td>
<td>1.2%</td>
</tr>
<tr>
<td>10 second</td>
<td>65.4</td>
<td>1.4%</td>
</tr>
<tr>
<td>15 second</td>
<td>66.2</td>
<td>2.6%</td>
</tr>
<tr>
<td>20 second</td>
<td>66.8</td>
<td>3.5%</td>
</tr>
<tr>
<td>30 second</td>
<td>66.4</td>
<td>3.0%</td>
</tr>
<tr>
<td>45 second</td>
<td>66.8</td>
<td>3.5%</td>
</tr>
<tr>
<td>60 second</td>
<td>66.7</td>
<td>3.3%</td>
</tr>
<tr>
<td>75 second</td>
<td>67.1</td>
<td>3.9%</td>
</tr>
<tr>
<td>90 second</td>
<td>67.0</td>
<td>3.8%</td>
</tr>
</tbody>
</table>

Table 5. Baseline and Perfect Prediction Simulation FE for different prediction window sizes for another drive cycle of same route

<table>
<thead>
<tr>
<th>Prediction Window Size</th>
<th>Perfect Prediction MPGe</th>
<th>Percent system energy saved over baseline simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (no prediction)</td>
<td>64.7</td>
<td>0.0%</td>
</tr>
<tr>
<td>3 second</td>
<td>64.1</td>
<td>-0.9%</td>
</tr>
<tr>
<td>5 second</td>
<td>65.4</td>
<td>1.1%</td>
</tr>
<tr>
<td>8 second</td>
<td>67.1</td>
<td>3.7%</td>
</tr>
<tr>
<td>10 second</td>
<td>67.2</td>
<td>4.0%</td>
</tr>
<tr>
<td>15 second</td>
<td>68.2</td>
<td>5.4%</td>
</tr>
<tr>
<td>20 second</td>
<td>67.8</td>
<td>4.8%</td>
</tr>
<tr>
<td>30 second</td>
<td>67.8</td>
<td>4.8%</td>
</tr>
<tr>
<td>45 second</td>
<td>68.1</td>
<td>5.3%</td>
</tr>
<tr>
<td>60 second</td>
<td>68.3</td>
<td>6.1%</td>
</tr>
<tr>
<td>75 second</td>
<td>68.7</td>
<td>6.2%</td>
</tr>
<tr>
<td>90 second</td>
<td>68.7</td>
<td>6.2%</td>
</tr>
</tbody>
</table>
Figure 11. Perfect Speed Prediction and Average NARX Artificial Neural Network Prediction Percent System Energy Savings over Baseline Simulation for another drive cycle on same route.

Figure 12. Percent System Energy Saved over Baseline Simulation as a function of Prediction Window Size for another drive cycle on same route.