

DSCC2014-6031

COMPARISON OF VELOCITY FORECASTING STRATEGIES FOR PREDICTIVE CONTROL IN HEVS

Chao Sun

National Engineering Laboratory for Electric Vehicles
Beijing Institute of Technology
Beijing, 100081, China
Email: chaosun.email@gmail.com

Xiaosong Hu

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, 41296, Sweden
Email: xiaosong@chalmers.se

Scott J. Moura

Department of Civil and Environmental Engineering
University of California
Berkeley, California 94720
Email: smoura@berkeley.edu

Fengchun Sun

National Engineering Laboratory for Electric Vehicles
Beijing Institute of Technology
Beijing, 100081, China
Email: sunfch@bit.edu.cn

ABSTRACT

The performance of model predictive control (MPC) for energy management in hybrid electric vehicles (HEVs) is strongly dependent on the projected future driving profile. This paper proposes a novel velocity forecasting method based on artificial neural networks (ANN). The objective is to improve the fuel economy of a power-split HEV in a nonlinear MPC framework. In this study, no telemetry or on-board sensor information is required. A comparative study is conducted between the ANN-based method and two other velocity predictors: generalized exponentially varying and Markov-chain models. The sensitivity of the prediction precision and computational cost on tuning parameters is examined for each forecasting strategy. Validation results show that the ANN-based velocity predictor exhibits the best overall performance with respect to minimizing fuel consumption.

1 INTRODUCTION

This paper introduces an ANN-based velocity forecaster into a predictive energy management controller for HEVs. The goal is to reduce the performance gap between deterministic dynamic

programming and MPC by more accurately forecasting future velocity and thus enhancing fuel economy. We leverage existing techniques in data-driven forecasting models, such as ANN [1] or Gaussian mixture models [2].

Over the past decade, sophisticated energy management strategies have been developed to fully exploit the advantages of a hybrid powertrain for increasing fuel economy [3, 4]. These strategies include dynamic programming (DP) [5], equivalent consumption minimization strategy (ECMS) [6], MPC [7]. This paper focuses on the MPC approach, which optimizes the decision variables over a finite moving horizon in real-time while maintaining computational load within a practical range [8]. Assuming the terrain information is known, the performance of MPC-based energy management strongly depends on future velocity forecasts [4]. This paper seeks to enhance the performance of predictive HEV energy management via precise velocity forecasting. Throughout this study, we assume the HEV is not equipped with telemetry or range-finding devices.

ANN is a powerful method for time series forecasting [9] across many application areas. Through appropriate training, an ANN can approximate any continuous measurable function to desired accuracy. It has also demonstrated strong capabili-

ties in predicting nonlinear dynamic behaviors [10]. To our best knowledge, ANN-based velocity predictors for MPC-based energy management of HEVs has not been investigated. Previously implemented methods include the exponentially varying predictor [7] and Markov-chain predictor [11]. The authors of [7] assumed power demand decreases exponentially over the prediction horizon. To systematically investigate this method, a generalized exponentially varying velocity predictor is considered in this paper. Markov-chain models are also often used for vehicle forecasting [11], and is effective for generating stochastic driving patterns [12]. This study extends the Markov-chain velocity predictor by developing multi-stage Markov chain models.

A comparative evaluation of the three classes of velocity predictors for MPC-based energy management of a power-split HEV is presented in this paper. The main contribution is an ANN-based velocity predictor for MPC implementation. The sensitivity of fuel economy to different network structures is analyzed. We also examine a generalized exponentially varying model and Markov-chain model. The three classes of velocity predictors are systematically compared in terms of prediction precision, computation time and resulting fuel economy.

The remainder of the paper is arranged as follows. Section 2 describes the ANN-based velocity predictor structures. Section 3 details the generalized exponentially varying and Markov-chain velocity predictors. The HEV powertrain control problem is formulated in Section 4, including the control-oriented vehicle model and nonlinear hierarchical MPC energy management strategy. Comparison results are illustrated in Section 5, followed by key conclusions in Section 6.

2 ARTIFICIAL NEURAL NETWORK VELOCITY PREDICTOR

ANNs can be trained to approximate a highly nonlinear input/output relationship [1]. Three typical ANN structures, back propagation neural network (BP-NN), layer recurrent neural network (Recurrent-NN) and radial basis function neural network (RBF-NN) are investigated with respect to short-term vehicular velocity forecasting. In the BP-NN structure, the input layer is used to receive and distribute the input pattern, followed by a hidden layer that depicts the nonlinearities of the input/output relationship. The output layer yields the desired output patterns [1].

For BP-ANN, the activation function is hyperbolic tangent sigmoid function. The basic formula of the hidden layer in the BP-algorithm is

$$a^1 = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (1)$$

$$n = Wa^0 + b \quad (2)$$

where a^1 and a^0 are neural outputs of the current layer and prior

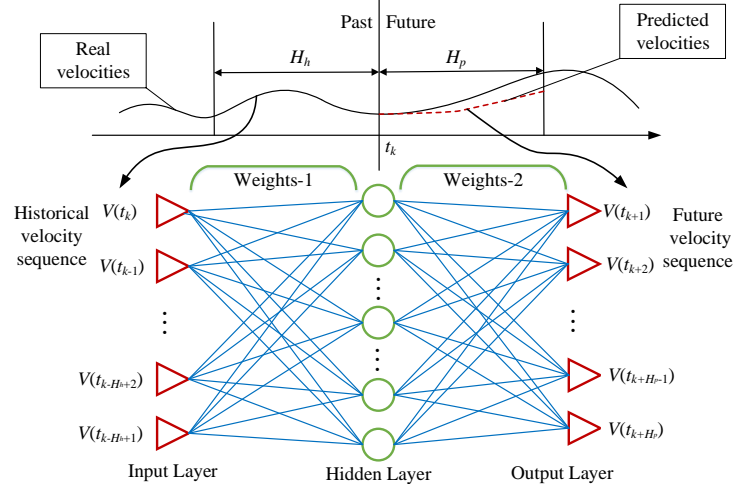


FIGURE 1. INPUT-OUTPUT PATTERN FOR ANN-BASED HORIZON VELOCITY PREDICTORS.

layer, respectively; n is accumulator output; W is weight and b is bias.

With a self-connected hidden layer, the Recurrent-NN has an internal state, which allows the network to exhibit temporal dynamic behavior [13]. However, this may increase the training convergence time. The basic formula of the recurrent layer in Recurrent-NN is

$$a^1(t) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (3)$$

$$n(t) = Wa^0(t) + W'a^1(t - \Delta t) + b \quad (4)$$

where W' is the feedback weight; $a^1(t - \Delta t)$ is the delayed output at time $(t - \Delta t)$.

The RBF-NN is also considered for comparison. In the standard approach to RBF-NN implementation, a radial basis function needs to be predefined at first. Then, the number of hidden layer neurons can be determined with a given training accuracy. The Gaussian function is used as the radial basis function in the hidden layer to activate the neurons, formulated as

$$a^1 = \exp\left(-\frac{\|n - \mu\|^2}{2\sigma^2}\right) \quad (5)$$

$$n = Wa^0 + b \quad (6)$$

where μ is the neural net center and σ is the spread width. Both μ and σ are typically fit using gradient descent.

The input-output structure of a NN-based velocity predictor is demonstrated in Fig. 1. The network inputs include historical

velocity sequences, and the outputs include predicted future velocity sequences. Each input-output pattern composes a moving window of fixed length. In Fig. 1, H_h is the length of the past velocity vector, H_p is the length of the forecasted velocity vector, and t is the time instant. All three ANN structures have the same input-output structure.

Six arbitrarily selected driving cycles are employed as training samples, including the WVUSCITY, JN1015, NEDC, US06, Artemis-highway and Artemis-suburban [14]. This training data set includes both urban and highway schedules, and therefore covers a relatively comprehensive set of driving scenarios. A total of 5930 input-output training samples are derived from this dataset. An arbitrary set of 85% of the data is used for training; while the remainder is used for validation. A sensitivity study has been conducted to avoid over-fitting. The ANN training process is well-studied [1], and therefore not discussed here.

3 EXPONENTIALLY VARYING AND MARKOV-CHAIN VELOCITY PREDICTOR

The exponentially varying velocity predictor is based on the empirical hypothesis that future vehicle velocity evolves exponentially. The Markov-chain velocity predictor is based on statistical analysis of driving data.

3.1 Exponentially Varying Velocity Predictor

For compactness of notation, denote $V_k = V(t_k)$. In each receding horizon, the exponentially varying velocity forecast is modeled as

$$V_{k+n} = V_k \times (1 + \varepsilon)^n, \text{ for } n = 1, 2, \dots, H_p \quad (7)$$

where V_k is the initial velocity at time step k , ε is the exponential coefficient. When ε is positive, future velocity increases; when ε is negative, future velocity decreases; when ε is zero, future velocity is constant and equal to V_k . Symbol ε is a tuning parameter, to be discussed later.

3.2 Markov-chain Velocity Predictor

The Markov states and emissions are defined on discrete-valued domains given by vehicle velocity \bar{V} (0 to 30 m/s) and vehicle acceleration $\bar{\alpha}$ in $(-1.5$ to $1.5 \text{ m/s}^2)$, respectively. For fair comparison, the same driving cycles used for ANN training are used to calculate the Markov emission probability matrix.

Suppose the vehicle velocity and acceleration are discretized into p and q intervals indexed by i and j , respectively. The current velocity is V_k , and possible future acceleration values are α_{k+n} . The Markov-chain process is defined by an emission probability matrix $T \in \mathbb{R}^{p \times q}$ with

$$[T]_{ij} = \Pr[\alpha_{k+n} = \bar{\alpha}_j | V_{k+n-1} = \bar{V}_i] \quad (8)$$

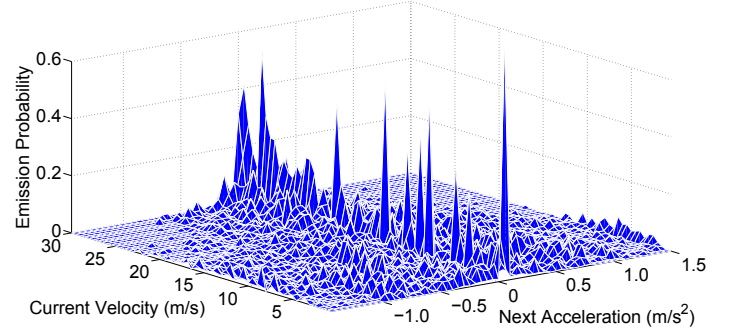


FIGURE 2. ONE-STAGE MARKOV EMISSION PROBABILITY MATRIX.

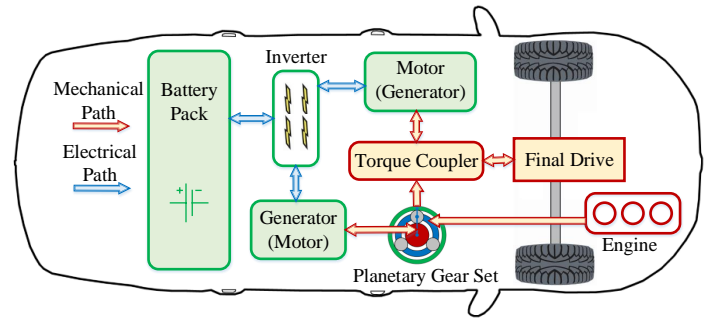


FIGURE 3. HEV POWER-SPLIT CONFIGURATION.

for $i \in \{1, \dots, p\}$, $j \in \{1, \dots, q\}$ and $n \in \{1, \dots, H_p\}$ [11]. Suppose $p = 60$ and $q = 60$, the probability matrix extracted from the sample driving cycle data is shown in Fig. 2. Note that large accelerations or decelerations are more likely at lower velocities than at high velocities.

To increase the prediction accuracy, a 1-stage Markov-chain process can be extended to multi-stage models [15], using the following probabilities

$$\Pr[\alpha_{k+1} = l_1, \alpha_{k+2} = l_2, \dots, \alpha_{k+m} = l_m | V_k = l_0, V_{k-1} = l_{-1}, \dots, V_{k-m} = l_{-m}] \quad (9)$$

where l_i 's are appropriate values from the corresponding discrete-valued domains.

4 PROBLEM FORMULATION

4.1 Control-oriented Power-split HEV Model

The structure of the power-split HEV is shown in Fig. 3, where the engine and generator are connected to the planet carrier and the sun gear, respectively. A torque coupler is used to combine the ring gear with the motor to power the final drive.

The kinematic constraint of the planetary gear set is given by

$$\omega_s S + \omega_r R = \omega_c (S + R) \quad (10)$$

where S and R are the radii of the sun gear and the ring gear, respectively. Angular speeds of the ring, sun, and carrier gears are denoted as ω_r , ω_s and ω_c . By neglecting the inertia of the pinion gears and assuming that all the powertrain shafts are rigid, inertial dynamics of the powertrain can be derived as

$$J_{gen} \frac{d\omega_{gen}}{dt} = T_{gen} + FS \quad (11)$$

$$J_{eng} \frac{d\omega_{eng}}{dt} = T_{eng} - F(S + R) \quad (12)$$

$$J_{mot} \frac{d\omega_{mot}}{dt} = T_{mot} - (T_{axle}/g_f) + FR \quad (13)$$

where J_{gen} , J_{eng} and J_{mot} are lumped inertias of the generator, engine and motor, respectively; $T_{gen} = T_s$, $T_{eng} = T_c$ and $T_{mot} = T_r$ are torques of the generator, engine, and motor, respectively; F represents the internal force on pinion gears; g_f is the gear ratio of the final drive; T_{axle} is the drive axle torque. To reduce the control-oriented model's complexity, we disregard the inertial dynamics, and use the steady-state values of (11)-(13). Motor torque and vehicle velocity are given by

$$\omega_{mot} = \frac{g_f}{R_{wheel}} V \quad (14)$$

$$m \frac{dV}{dt} = \frac{T_{axle} + T_{brake}}{R_{wheel}} + mg \sin(\theta) - \frac{\rho AC_d V^2}{2} - C_r mg \cos(\theta) \quad (15)$$

where R_{wheel} is the wheel radius; m is vehicle mass; T_{brake} is the friction brake torque; θ denotes the road grade and is assumed to be zero; $\frac{1}{2} \rho AC_d$ is the aerodynamic drag resistance; C_r represents the rolling resistance coefficient.

The engine fuel flow rate (\dot{m}_{fuel}) and power transfer efficiencies for the motor and generator (η_{mot} and η_{gen}) are:

$$\dot{m}_{fuel} = \psi_1(\omega_{eng}, T_{eng}) \quad (16)$$

$$\eta_{mot} = \psi_2(\omega_{mot}, T_{mot}) \quad (17)$$

$$\eta_{gen} = \psi_3(\omega_{gen}, T_{gen}) \quad (18)$$

where ψ_1 , ψ_2 and ψ_3 are empirical maps of the engine fuel flow, motor efficiency and generator efficiency, respectively.

Battery state of charge (SOC) is modeled as a single state. The internal resistance model is applied in the control-oriented model [16], described as

$$\dot{SOC} = -(I_{bat}/Q_{max}) \quad (19)$$

$$P_{bat} = V_{oc} I_{bat} - I_{bat}^2 R_{bat} \quad (20)$$

where I_{bat} and Q_{max} are battery current and charge capacity, respectively; P_{bat} and R_{bat} are the power and internal resistance; V_{oc} represents the open circuit voltage. Positive P_{bat} denotes discharge. Terminal battery power is described by

$$P_{bat} = P_{mot}/(\eta_{mot}\eta_{inv})^{k_{mot}} + P_{gen}/(\eta_{gen}\eta_{inv})^{k_{gen}} \quad (21)$$

where P_{mot} and P_{gen} are motor and generator shaft powers, respectively; η_{inv} is the inverter efficiency;

$$k_i = \begin{cases} 1, & \text{if } P_i > 0 \\ -1, & \text{if } P_i \leq 0 \end{cases} \text{ for } i = \{mot, gen\} \quad (22)$$

Eqn. (10)-(22) summarize the control model used for MPC, and more details can be found in [12]. Throughout this study, MPC is applied to a detailed plant model furnished by the QSS-toolbox developed at ETH Zürich (see [17] for details).

4.2 Nonlinear Model Predictive Control

Engine speed ω_{eng} and torque T_{eng} are chosen as decision variables. Denoting x as the state variable, u as the control variable, d as the system disturbance, and y as the output, the proposed control-oriented powertrain model can be represented as

$$\dot{x} = f(x, u, d), \quad y = g(x, u, d) \quad (23)$$

with $x = \text{SOC}$, $u = [\omega_{eng}, T_{eng}]^T$, $d = V_{predict}$, $y = [\dot{m}_{fuel}, P_{bat}, T_{mot}, \omega_{gen}, T_{gen}]^T$. $V_{predict}$ is a short-term future velocity sequence provided by the velocity predictor at each time step. Considering $\Delta t = 1$ second, at time step k , the cost function J_k is formulated as

$$J_k = \int_{k\Delta t}^{(k+H_p)\Delta t} [\dot{m}_{fuel}(u(t))]^2 dt \quad (24)$$

where H_p is the prediction horizon length, which is herein equal to the control horizon length for simplicity [7]. Additionally, the following physical constraints must be enforced:

$$\begin{aligned} & \text{SOC}^{\min} \leq \text{SOC} \leq \text{SOC}^{\max}, \\ & T_{eng}^{\min} \leq T_{eng} \leq T_{eng}^{\max}, \quad \omega_{eng}^{\min} \leq \omega_{eng} \leq \omega_{eng}^{\max}, \\ & T_{gen}^{\min} \leq T_{gen} \leq T_{gen}^{\max}, \quad \omega_{gen}^{\min} \leq \omega_{gen} \leq \omega_{gen}^{\max}, \\ & T_{mot}^{\min} \leq T_{mot} \leq T_{mot}^{\max}, \quad \omega_{mot}^{\min} \leq \omega_{mot} \leq \omega_{mot}^{\max}. \end{aligned} \quad (25)$$

In addition, HEVs must also keep the final SOC close to the initial SOC over the entire driving cycle. However, since future velocity trajectories are unknown in this paper, it is difficult

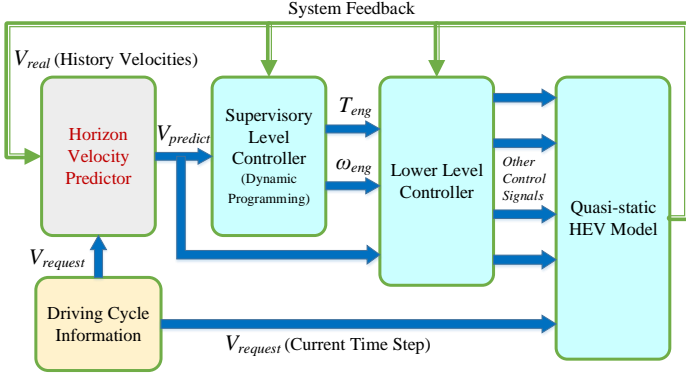


FIGURE 4. HIERARCHICAL CONTROL SYSTEM AND THE SIMULATION DIAGRAM.

to acquire the optimal SOC trajectory for the MPC controller to follow. Thus, we define a charge sustaining strategy as the battery energy usage policy. Namely, the terminal SOC is restricted within a pre-defined envelope in each control horizon.

$$\text{SOC}^{low}(k) \leq \text{SOC}((k + H_p)\Delta t) \leq \text{SOC}^{high}(k). \quad (26)$$

The horizon's terminal-SOC-bounds are given by

$$\begin{aligned} \text{SOC}^{low}(k) &= \text{SOC}^{min} + (\text{SOC}_f - \text{SOC}^{min}) \sin\left(\frac{k}{N} \cdot \frac{\pi}{2}\right), \\ \text{SOC}^{high}(k) &= \text{SOC}^{max} + (\text{SOC}_f - \text{SOC}^{max}) \sin\left(\frac{k}{N} \cdot \frac{\pi}{2}\right), \end{aligned} \quad (27)$$

where SOC_f is the desired final SOC and N is the total number of time steps in the driving cycle. These bounds converge to the desired final SOC as the drive cycle reaches the final time step. For comparative analysis, all simulations with different velocity predictors employ the same terminal-SOC-reference bound functions.

The architecture of the simulation diagram is illustrated in Fig. 4. The MPC controller is applied in the supervisory level and the optimal control problem is solved using DP [18]. The actual driving velocity $V_{request}$ is sent to the quasi-static HEV model. Velocity forecast $V_{predict}$ is provided from the velocity predictor. The simulation procedure is described as follows:

1. The horizon velocity predictor generates the forecasted velocity vector $V_{predict}$;
2. Given $V_{predict}$, the MPC controller calculates the optimal control policy that minimizes objective function (24);
3. Apply the first element of the optimal control policy to the lower level controller;
4. Measure the system states and update. Repeat the simulation procedure at each time instant.

5 SIMULATION RESULTS AND DISCUSSION

Eight driving cycles are used for comparing the performance of the three velocity predictors. These include certification driving cycles HWFET, UDDS [14] and real collected driving data WVUSUB, WVUINTER, Real HW2, Real HW3, Real UB2 and Real UB3. The real-world cycles in the testing data are from [19,20]. Note these cycles differ from those used for ANN training and Markov probability computation. All testing cycles are concatenated for ease of presentation in Fig. 5.

Deterministic DP (DDP) and deterministic MPC (DMPC) use full knowledge of the vehicle velocity schedule and serve as benchmarks. Vehicle parameters and empirical maps are obtained for the *Toyota Prius* model in ADVISOR 2002 [14]. Simulation was performed on a personal computer with an Intel Core i7-3630QM CPU @2.4GHz. Horizon length H_p is specified to be 10 as a compromise between computation and optimality. The optimal control policy is solved by DP, and the computation time is 0.67 seconds on average for each time step. Thus the control system runs in real time in Matlab. The initial SOC and final SOC in all simulations are set as 0.65. Upper and lower SOC bounds are 0.8 and 0.5, respectively. Four metrics are used to assess the velocity predictors:

1. Average root mean squared error (RMSE, in m/s) of the predicted velocities in all of the receding horizons.
2. Violating frequency of the velocity and acceleration constraints for the predicted velocities, represented by e .
3. Online computation time T (in microseconds) of the velocity prediction process at each time step.
4. Consequent vehicle fuel consumption (in grams).

Detailed simulation results for the UDDS cycle are shown in Sections 5.1 to 5.4. Comparative results for all testing cycles are shown in Section 5.5. Note that the terminal SOC deviation has been compensated for in the fuel consumption calculation.

5.1 Exponentially Varying Predictor for UDDS

As shown in Table 1, the minimum fuel consumption occurs when predicted velocity decreases slightly. Terminal SOC, represented by SOC_f , ascends as ϵ grows. This is because when the predicted velocity changes aggressively, the engine tends to provide more power. Thus, more engine power is absorbed by the battery through the generator. The minimum fuel consumption occurs for $\epsilon = -0.01$, which corresponds to relatively low predicted velocity RSME.

5.2 Markov-chain Velocity Predictor for UDDS

Next the Markov-chain velocity predictor is evaluated. The 1-stage Markov-chain model has the worst performance in characterizing future velocities, as shown in Table 2. It is, however, the most computationally efficient. For multi-stage Markov-chain models, one requires an often prohibitively large set of

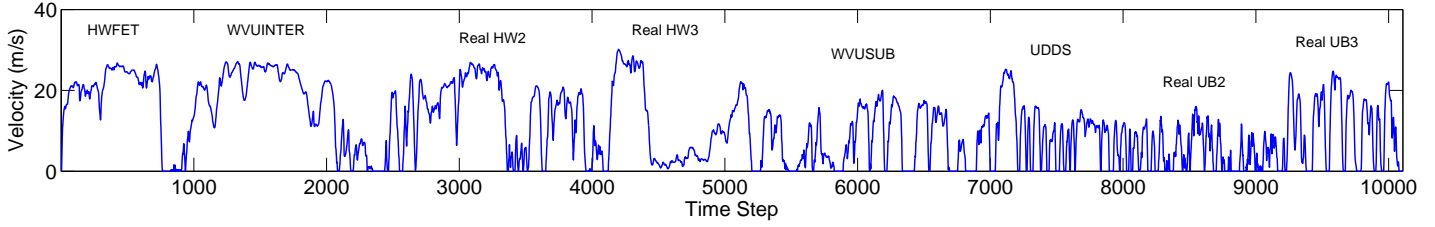


FIGURE 5. TESTING DATA. REAL HW: REAL COLLECTED HIGHWAY DATA; REAL UB: REAL COLLECTED URBAN DATA.

TABLE 1. EXPONENTIALLY VARYING PREDICTOR UDDS

ϵ	SOC_t^{MPC}	RMSE	Fuel	T
-0.03	0.6475	3.63	539.2	0.023
-0.02	0.6487	3.45	538.4	0.026
-0.01	0.6501	3.38	537.3	0.032
0	0.6515	3.44	538.8	0.034
0.01	0.6554	3.67	541.7	0.029
0.02	0.6588	4.07	545.0	0.023

TABLE 2. MARKOV-CHAIN PREDICTOR FOR UDDS

Stage	SOC_t^{MPC}	RMSE	e (%)	Fuel	T
1	0.6578	3.67	0	564.5	1.647
2	0.6642	3.23	0	529.8	2.821
3	0.6648	3.21	0	528.3	2.919
4	0.6646	3.14	0	526.7	2.831
5	0.6648	3.11	0	523.9	2.797

training cycles to identify each transition probability. This phenomenon elucidates the poor practicality of multi-stage Markov-chain speed predictors. To address this problem, either a sufficiently rich sample database is needed or a well-tuned probability map must be computed off-line. The maximum Markov stage in this simulation is 5. The extracted probability matrix changes negligibly for higher stages.

When a specific multi-stage Markov state is not found in the probability matrix, the Markov stage will decrease until an effective probability distribution is found. For example, if a 5-stage Markov state is extracted from the testing cycle and the exact same state cannot be identified from the assigned database, the Markov chain velocity predictor will abandon the last state element and reduce the stage to 4. For a new Markov state in the next prediction process, the Markov stage will recover to 5. It can be seen in Table 2 that more stages result in smaller RMSE. Hence, increased fuel efficiency can be achieved via multi-stage Markov chain speed predictors. However, the cardinality of the probability matrix for a multi-stage Markov-chain model scales exponentially. Therefore, Markov-chain models with greater than 3 stages are rarely used.

5.3 ANN Velocity Predictor for UDDS

For ANN velocity predictors, the dimension of the input velocity vector H_h is specified to be 10. The authors investigated

different neuron numbers. In order to gain the best prediction performance within an acceptable training time, the neuron numbers for BP-NN and Recurrent-NN are both set as 20 and 50. Since the RBF-NN has much faster training speed than BP-NN and Recurrent-NN, we can easily develop RBF-NN based velocity predictors with more neural nodes. Therefore the RBF-NN neuron number is set as 50 and 100.

Table 3 shows the comparison results of the three categories of ANN velocity predictors for UDDS testing. Note that the error caused by velocity/acceleration constraint violations (e) of the predicted velocity is included in the average RMSE computation. The RBF-ANN structure achieves better fuel economy than the Recurrent-NN and BP-NN velocity predictors, despite a similar RMSE, since there are substantially fewer constraint violations.

5.4 Comparison of the Velocity Predictors for UDDS

The 1-stage and 5-stage Markov-chain velocity predictors are selected to compare with the best velocity predictors from the exponentially varying and ANN classes. Comparison results are shown in Table 4.

5.4.1 Prediction Precision From Table 4, the average RMSE of 1-stage Markov-chain velocity predictor is about 10% larger than that of the exponentially varying predictor. This means the 1-stage Markov-chain process behaves poorly in modeling comprehensive driving behaviors. Additionally, the aver-

TABLE 3. ANN PREDICTOR FOR UDDS

Net-Node	SOC_t^{MPC}	RMSE	e (%)	Fuel	T
BP-20	0.6576	2.32	914 (6.67%)	500.9	0.22
BP-50	0.6562	2.29	528 (3.85%)	501.2	0.21
Re-20	0.6595	2.28	819 (5.98%)	497.9	0.21
Re-50	0.6571	2.29	933 (6.81%)	500.0	0.22
RBF-50	0.6566	2.28	26 (0.19%)	499.9	0.22
RBF-100	0.6580	2.27	24 (0.18%)	496.1	0.21

(e is the number (and percentage) of constraint violations.)

TABLE 4. VELOCITY PREDICTING COMPARISON FOR UDDS

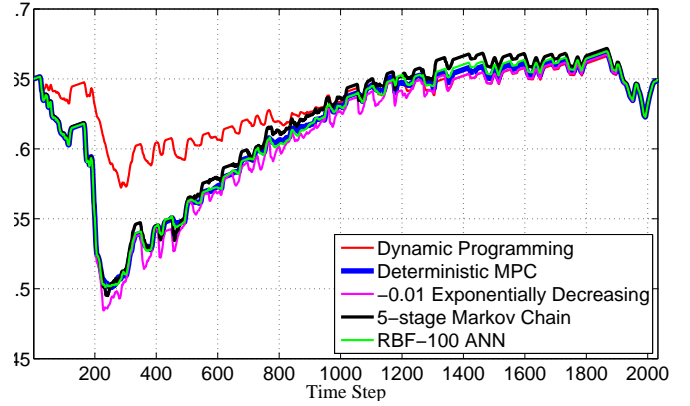
Methods	SOC_t^{MPC}	RMSE	e (%)	Fuel	T
DDP	0%	–	0	0%	–
DMPC	+0.68%	0	0	+4.86%	–
-0.01 EV	+0.02%	3.38	0	+14.45%	0.03
1-stage MC	+1.2%	3.67	0	+20.24%	1.65
5-stage MC	+2.28%	3.11	0	+11.60%	2.92
RBF-100	+1.23%	2.27	0.18%	+5.67%	0.21

(EV = exponentially varying; MC = Markov-chain. The terminal SOC and fuel consumption results have been normalized against DDP for an intuitive comparison.)

age RMSE of the 100-node RBF-NN velocity predictor is 27% less than that of the 5-stage Markov chain predictor. Both of them are better than the other two approaches. This indicates that the 100-node RBF-NN and 5-stage Markov-chain velocity predictors are preferable in predicting short-term future velocities.

The nature of each prediction method can be seen visually in Fig. 6. This figure elucidates how a 1-stage Markov chain is too short-sighted to capture velocity profiles longer than a few time steps. One can also observe how the 5-stage Markov chain roughly predicts constant acceleration. In contrast, the 100-node RBF-NN seems to capture micro-trip-like behaviors better than the other three. This is especially apparent when the velocity switches from acceleration to deceleration.

5.4.2 Fuel Consumption From Table 4, the 100-node RBF-NN predictor is the most energy-saving predictor. In particular, the fuel consumption of 100-RBF is only 4 grams higher

**FIGURE 7.** SOC TRAJECTORIES FOR UDDS TESTING.

than the deterministic MPC approach, which is only 0.8%. Fig. 7 shows the SOC trajectories for DDP, DMPC and the four velocity-predictor-based MPC controllers given in Table 4. The difference between the DP and MPC-based trajectories is due to DP being a global optimization; whereas the MPC controller yields a locally optimal solution for each control horizon. With the same battery energy consuming policy, the SOC trajectory of the 100-node RBF-ANN MPC is close to the DMPC benchmark.

5.4.3 Computation Time and Repeatability From Table 4, it is obvious that the Markov-chain method is computationally heavier than the ANN-based method and the exponentially varying method. This is because the process of generating a stochastic Markov emission needs additional calculations to form the probability distributions. In addition, the Markov-chain approach produces different velocity forecasts from the same input, which creates a repeatability issue. The other methods provide deterministic velocity forecasts.

5.4.4 Constraint Violation The exponentially-varying and Markov-chain approaches do not violate the speed and acceleration constraints at all. In the 100-node RBF-NN case, the constraints are violated 24 times, which is 0.18% of all the predicted velocities.

5.5 Simulation Results for All Testing Cycles

Similar outcomes are achieved for the other seven testing cycles. Fig. 8 provides the average values and standard deviations of fuel consumption (percentage) for MPC with different velocity predictors, relative to deterministic DP. We can see that ANN predictors generally provide the least increase in fuel consumption, both in terms of average values and standard deviation. Note these results are evaluated across both emission cer-

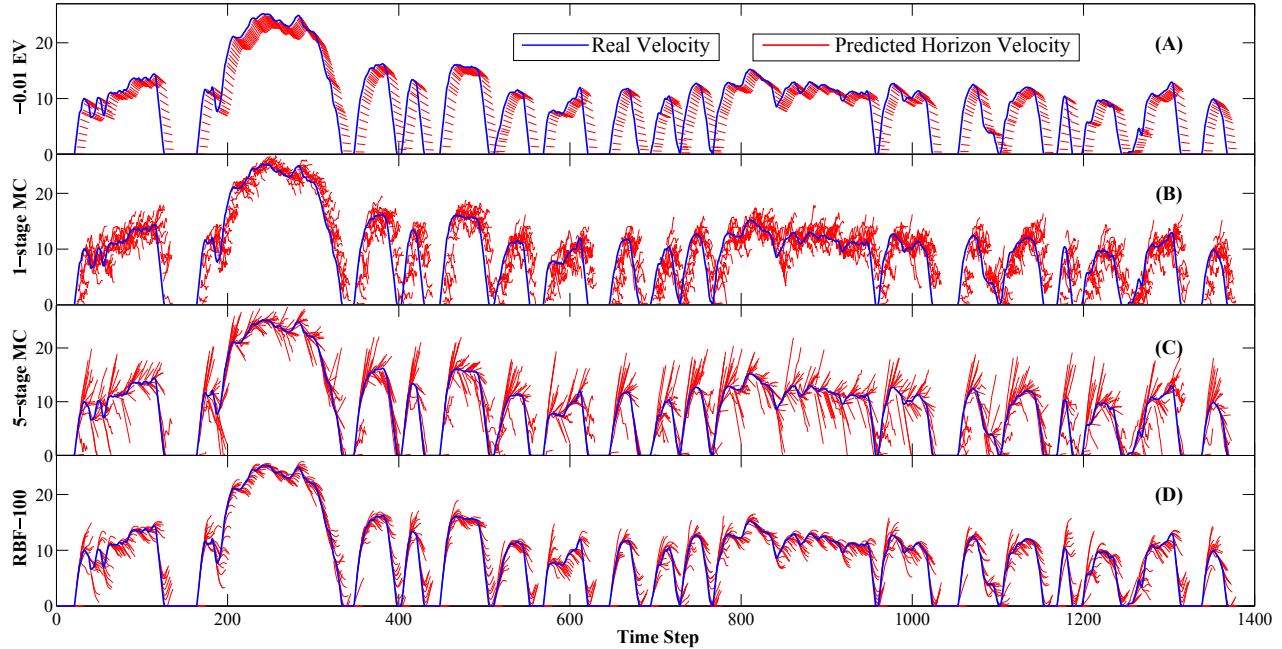


FIGURE 6. VELOCITY PREDICTION RESULTS FOR UDDS.

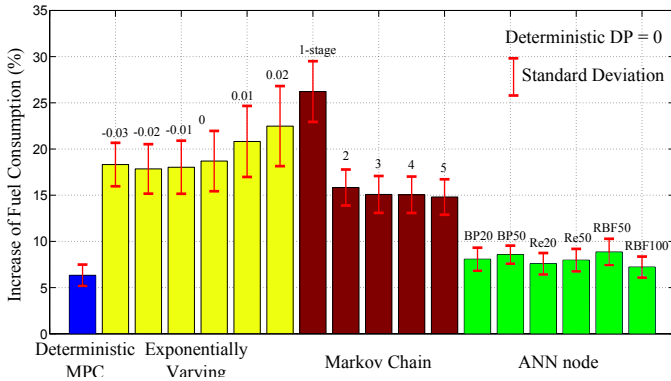


FIGURE 8. INCREASE IN FUEL CONSUMPTION COMPARED WITH DETERMINISTIC DP.

tification cycles and real-world drive cycle data. Consequently, we conclude the ANNs provide a promising blend of prediction capability and computational efficiency for MPC energy management in HEVs.

6 CONCLUSIONS

The performance of MPC-based HEV energy management strongly depends on accurate forecasts of vehicular velocities

over the receding horizon. This paper compares three classes of velocity forecasting algorithms. Generalized exponentially varying, stochastic Markov-chain and novel ANN-based velocity predictors are systematically described and evaluated in terms of prediction precision, computation burden and fuel economy. Simulation results indicate that the 100-node RBF-NN velocity predictor has the best overall performance. Namely, it provides the least average fuel consumption for eight testing cycles, and is computationally efficient.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Karl Hedrick, Prof. Francesco Borrelli at the University of California, Berkeley, USA, for substantial help and enlightening discussions on MPC theory and its implementation.

REFERENCES

- [1] Hagan, M. T., Demuth, H. B., Beale, M. H., et al., 1996. *Neural network design*. Pws Pub. Boston.
- [2] Angkititrakul, P., Ryuta, T., Wakita, T., Takeda, K., Miyajima, C., and Suzuki, T., 2009. "Evaluation of driver-behavior models in real-world car-following task". In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 113–118.
- [3] Kim, N., Cha, S., and Peng, H., 2011. "Optimal control

- of hybrid electric vehicles based on Pontryagin's minimum principle". *IEEE Transactions on Control Systems Technology*, **19**(5), pp. 1279–1287.
- [4] Sciarretta, A., and Guzzella, L., 2007. "Control of hybrid electric vehicles". *IEEE Control systems*, **27**(2), pp. 60–70.
- [5] Lin, C.-C., Peng, H., Grizzle, J. W., and Kang, J.-M., 2003. "Power management strategy for a parallel hybrid electric truck". *IEEE Transactions on Control Systems Technology*, **11**(6), pp. 839–849.
- [6] Serrao, L., Onori, S., and Rizzoni, G., 2009. "ECMS as a realization of pontryagin's minimum principle for HEV control". In American Control Conference (ACC), pp. 3964–3969.
- [7] Borhan, H., Vahidi, A., Phillips, A. M., Kuang, M. L., Kolmanovsky, I. V., and Di Cairano, S., 2012. "MPC-based energy management of a power-split hybrid electric vehicle". *IEEE Transactions on Control Systems Technology*, **20**(3), pp. 593–603.
- [8] Camacho, E. F., Bordons, C., Camacho, E. F., and Bordons, C., 2004. *Model predictive control*, Vol. 2. Springer London.
- [9] Zhang, G. P., 2003. "Time series forecasting using a hybrid ARIMA and neural network model". *Neurocomputing*, **50**, pp. 159–175.
- [10] Mazloumi, E., Rose, G., Currie, G., and Moridpour, S., 2011. "Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction". *Engineering Applications of Artificial Intelligence*, **24**(3), pp. 534–542.
- [11] Bichi, M., Ripaccioli, G., Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I. V., 2010. "Stochastic model predictive control with driver behavior learning for improved powertrain control". In Decision and Control (CDC), pp. 6077–6082.
- [12] Moura, S. J., Fathy, H. K., Callaway, D. S., and Stein, J. L., 2011. "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles". *IEEE Transactions on Control Systems Technology*, **19**(3), pp. 545–555.
- [13] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J., 2009. "A novel connectionist system for unconstrained handwriting recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**(5), pp. 855–868.
- [14] Wipke, K. B., Cuddy, M. R., and Burch, S. D., 1999. "ADVISOR 2.1: a user-friendly advanced powertrain simulation using a combined backward/forward approach". *IEEE Transactions on Vehicular Technology*, **48**(6), pp. 1751–1761.
- [15] Bolch, G., Greiner, S., de Meer, H., and Trivedi, K. S., 2006. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.
- [16] Hu, X., Li, S., and Peng, H., 2012. "A comparative study of equivalent circuit models for li-ion batteries". *Journal of Power Sources*, **198**, pp. 359–367.
- [17] Guzzella, L., and Amstutz, A., 1999. "CAE tools for quasi-static modeling and optimization of hybrid powertrains". *IEEE Transactions on Vehicular Technology*, **48**(6), pp. 1762–1769.
- [18] Sundstrom, O., and Guzzella, L., 2009. "A generic dynamic programming Matlab function". In Control Applications (CCA) & Intelligent Control (ISIC), IEEE, pp. 1625–1630.
- [19] Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., and Bayen, A. M., 2010. "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment". *Transportation Research Part C: Emerging Technologies*, **18**(4), pp. 568–583.
- [20] LeBlanc, D., 2006. Road departure crash warning system field operational test: methodology and results. Technical report, University of Michigan, Ann Arbor, Transportation Research Institute.