Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles

Chao Sun, Xiaosong Hu, *Member, IEEE,* Scott J. Moura, *Member, IEEE,* and Fengchun Sun

***Abstract*—The performance and practicality of predictive energy management in hybrid electric vehicles (HEVs) are highly dependent on the forecast of future vehicular velocities, both in terms of accuracy and computational efficiency. In this paper, we provide a comprehensive comparative analysis of three velocity prediction strategies, applied within a model predictive control (MPC) framework. The prediction process is performed over each receding horizon, and the predicted velocities are utilized for fuel economy optimization of a power-split HEV. We assume that no telemetry or on-board sensor information is available for the controller, and the actual future driving profile is completely unknown. Basic principles of exponentially varying, stochastic Markov-chain, and neural network based velocity prediction approaches are described. Their sensitivity to tuning parameters is analyzed, and the prediction precision, computational cost, and resultant vehicular fuel economy are compared.**

***Index Terms*—Energy Management, Hybrid Electric Vehicle, Model Predictive Control, Velocity Prediction, Artificial Neural Network, Comparison.**

I. Introduction

SOPHISTICATED energy management strategies have been developed to provide better fuel economy performance in HEVs [1], [2]. This paper intends to facilitate the performance of predictive energy management through evaluating different horizon velocity predicting approaches.

In the literature, dynamic programming (DP) and equivalent consumption minimization strategy (ECMS) are crucial in resolving the energy management problem for HEVs. Globally, DP can ensure an optimal result when complete knowledge of driving conditions is prescribed [3]. However, the exact future power demand is usually unknown and the computational burden is prohibitive. DP solutions are often realized offline and deployed as benchmarks [2]. ECMS is an instantaneous optimization for HEV energy management [4], [5]. By defining an equivalent fuel cost for battery energy, ECMS solves the optimal power split at each time instant rather than over a time horizon. It has demonstrated that given an appropriate equivalence factor, ECMS is comparable to DP [6]. Nevertheless, tuning the equivalence factor is nontrivial. ECMS variants, such as telemetric-ECMS and adaptive-ECMS, are proposed to adjust the equivalence factor based on information from telemetry equipment or on-board sensors [7], [8]. Due to the uncertainty of future driving profiles, endowing the controller with an appropriate prediction ability can achieve better performance. The forecast ability in the adaptive-ECMS is different from the model predictive control approach used in this paper.

MPC has attracted increasing attention in the HEV energy management research community. Given a finite moving horizon, an MPC controller can maintain computational load within a practical range. An MPC controller solves a short-term energy management problem at each time step, via nonlinear programming [9], quadratic programming [10], Pontryagin's minimum principle [11] or DP [12]. The performance of MPC strongly depends on the power reference provided in each prediction horizon. The precision of future power prediction is instrumental for the overall vehicle fuel economy. Considering that road grade information is static, we focus on the horizon velocity prediction. No telemetry devices or environment detecting sensors are assumed, and the future driving information is completely unknown.

We investigated three velocity predicting approaches. In reference [9], the authors assumed power demand decreases exponentially over the prediction horizon. The aim of this simple method was to provide an intuitive understanding of how velocity prediction affects fuel economy. To systematically investigate this method, a generalized exponentially varying velocity predictor is considered in this paper.

Markov-chain models are widely used for vehicular velocity modeling [13], [14]. An MPC controller with a stochastic Markov-chain velocity predictor is usually called stochastic MPC (SMPC) [12], [15]. The 1-stage Markov-chain process has proven to be effective in generating fixed-route driving patterns. However, when comprehensive driving tasks are considered, the accuracy of 1-stage Markov-chain may decline. On the other hand, the predicted power demand relies not only on the present vehicle states, but also on the historical values [16]. Typically, the more historical data used, the more accurate the prediction. For SMPC, multi-stage Markov-chain processes can hence be formulated to enhance the velocity prediction accuracy.

Artificial neural networks (NNs) are a successful method for time series forecasting [17]. Applications of NNs to predicting city power load [18], driving handling behaviors [19], and traffic flows [20] have verified its strong capability in predicting nonlinear dynamic behaviors. In reference [21], the authors also employed NN to predict the road type and traffic congestion in order to improve the HEV fuel efficiency. Although tuning the NN parameters may produce different pre-

C. Sun is a Ph.D. student of National Engineering Laboratory for Electric Vehicles, Beijing Institute of Technology, Beijing 100081, China. Now he is a visiting sutudent researcher of Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA (chaosun.email@gmail.com).

X. Hu is with the Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA (xiaosong@chalmers.se).

S. Moura is with the Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA (smoura@berkeley.edu).

F. Sun is with the National Engineering Laboratory for Electric Vehicles, Beijing Institute of Technology, Beijing 100081, China (sunfch@bit.edu.cn).

diction performances, it's easy to design a NN-based velocity predictor with reasonable precision. To our best knowledge, NN-based velocity predictor for predictive energy management of HEVs has not been investigated.

In this paper, we present a comparative study of three classes of velocity predictors for MPC-based energy management of a power-split HEV. This work adds two original contributions to the related literature. First, a NN-based velocity predictor is implemented, for the first time, in MPC-based HEV energy management. Three different NN structures are examined in terms of prediction accuracy and consequent fuel economy. Second, the generalized exponentially varying and Markov-chain velocity predictors are investigated to fully explore their potential. The above three classes of velocity predictors are systematically compared in terms of prediction precision, computation time, and consequent fuel consumption. Although the foregoing contributions are made specifically for MPC-based energy management of a power-split HEV, the proposed approach extends to other HEV configurations.

The remainder of the paper is arranged as follows. In Section II the problem formulation is introduced, including the control-oriented vehicle model and nonlinear MPC optimization strategy. Section III details the three different types of horizon velocity predictors. Comparison results are illustrated in Section IV, followed by key conclusions in Section V.

II. Problem Formulation

A. Control-oriented Power-split HEV Model

Power-split HEVs are dominant in the current HEV market [14]. In a planetary gear set functioned power-split HEV, the engine and MG1 (motor/generator 1) are connected to the planet carrier and the sun gear, respectively. A torque coupler is used to combine the ring gear with MG2 to power the final drive. The kinematic constraint on the ring, carrier, and sun gear angular velocities is given mathematically by

$$\omega_s S + \omega_r R = \omega_c (S + R), \tag{1}$$

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, respectively. By neglecting the inertia of the pinion gears and assuming that all the powertrain shafts are rigid, inertial dynamics of the powertrain is derived as

$$\begin{aligned} J_{MG1}\frac{d\omega_{MG1}}{dt} &= T_{MG1} + FS, &(2)\\ J_{eng}\frac{d\omega_{eng}}{dt} &= T_{eng} - F(S+R), &(3)\\ J_{MG2}\frac{d\omega_{MG2}}{dt} &= T_{MG2} - (T_{axle}/g_f) + FR, &(4)\end{aligned}$$

where J_{MG1}, J_{eng} and J_{MG2} are inertias of MG1, engine and MG2, respectively; $T_{eng} = T_c$ is the engine torque; $T_{MG1} = T_s$ and $T_{MG2} = T_r$ are MG1 and MG2 torques (positive in motoring mode), respectively; F represents the internal force on pinion gears; g_f is the gear ratio of the final drive; T_{axle} is the torque produced from powertrain on the drive axle. To reduce the control-oriented model's complexity, we disregard the inertial dynamics, and use the steady-state values of (2)-(4). MG2 torque and vehicle velocity are given by

$$\begin{aligned} \omega_{MG2} &= \frac{g_f}{R_{wheel}}V, &(5)\\ m\frac{dV}{dt} &= \frac{T_{axle}+T_{brake}}{R_{wheel}} + mg\sin(\theta), &\\ &\quad -\frac{1}{2}\rho A C_d V^2 - C_r mg\cos(\theta), &(6)\end{aligned}$$

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

At each time instant, the controller computes an optimal split between the engine, MG1, and MG2 to minimize fuel consumption. Fuel flow rate of the engine ($\dot{m}_{fuel}$) and power transfer efficiencies for MG1 and MG2 (η_{MG1} and η_{MG2}) are extracted from empirical maps,

$$\begin{aligned} \dot{m}_{fuel} &= \psi_1(\omega_{eng}, T_{eng}), &(7)\\ \eta_{MG1} &= \psi_3(\omega_{MG1}, T_{MG1}), &(8)\\ \eta_{MG2} &= \psi_2(\omega_{MG2}, T_{MG2}), &(9)\end{aligned}$$

where ψ_1, ψ_2 and ψ_3 are corresponding empirical maps.

Upper and lower boundaries of the battery state of charge (SOC) need to be specified to maintain the battery within a safe operating region and extend its life [22]. For HEV energy management, the SOC is modeled as a single state. Although more sophisticated battery models have been developed to describe battery dynamics [23], [24], the internal resistance model is still the most prevailing model for HEV supervisory control due to its simplicity, described as

$$\begin{aligned} \dot{\text{SOC}} &= -(I_{batt}/Q_{max}), &(10)\\ P_{batt} &= V_{oc}I_{batt} - I_{batt}^2 R_{batt}, &(11)\end{aligned}$$

where I_{batt} and Q_{max} are battery current and maximum capacity, respectively; P_{batt} and R_{batt} are batter power and internal resistance; V_{oc} represents the open circuit voltage. Positive P_{batt} denotes discharge. The battery in a power-split HEV is connected to a bi-directional converter to supply power or recuperate energy from the electrical machines. Terminal battery power is described by

$$P_{batt} = P_{MG1}/(\eta_{MG1}\eta_{inv})^{k_{MG1}} + P_{MG2}/(\eta_{MG2}\eta_{inv})^{k_{MG2}}, \tag{12}$$

where P_{MG1} and P_{MG2} are MG1 and MG2 shaft powers, respectively (positive in motoring mode); η_{inv} is the inverter efficiency;

$$k_i = \left\{ \begin{array}{ll} 1 & \text{if } P_i > 0 \\ -1 & \text{if } P_i \le 0 \end{array} \right., \quad \text{for } i = \{MG1, MG2\}. \tag{13}$$

A complete description of the battery SOC dynamics can be obtained from (9)-(13). Equations (1)-(13) summarize the control model used for MPC. Throughout this study, MPC is applied to a detailed plant model furnished by the QSS-toolbox developed at ETH Zürich [25].

B. Nonlinear Model Predictive Control

A hierarchical MPC [26] is employed for fuel consumption minimization, formulated as a constrained nonlinear optimization problem and solved by DP at each time step. The engine torque and speed are selected as control 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

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

with $x = \text{SOC}$, $u = [T_{eng}, \omega_{eng}]^T$, $d = V_{predict}$, $y = [\dot{m}_{fuel}, P_{batt}, T_{MG1}, \omega_{MG1}, T_{MG2}]^T$, where $V_{predict}$ is the future velocity sequence provided by a velocity predictor in each control horizon.

Considering a one second time step ($\Delta t = 1$), at time step k, the cost function J_k is formulated as

$$J_k = \int_{k\Delta t}^{(k+H_p)\Delta t} \left([\dot{m}_{fuel}(u(t))]^2 + \lambda O(t)\right) dt, \tag{14}$$

where H_p is the prediction horizon length, which is herein equal to the control horizon length [9]; $O(t)$ is the engine on/off switching time; λ is the penalty for engine on/off switching. Additionally, the following physical constraints must be enforced:

$$\begin{array}{l} \text{SOC}^{\min} \leq \text{SOC} \leq \text{SOC}^{\max}, \\ T_{eng}^{\min} \leq T_{eng} \leq T_{eng}^{\max},\ \omega_{eng}^{\min} \leq \omega_{eng} \leq \omega_{eng}^{\max}, \\ T_{MG1}^{\min} \leq T_{MG1} \leq T_{MG1}^{\max},\ \omega_{MG1}^{\min} \leq \omega_{MG1} \leq \omega_{MG1}^{\max}, \\ T_{MG2}^{\min} \leq T_{MG2} \leq T_{MG2}^{\max},\ \omega_{MG2}^{\min} \leq \omega_{MG2} \leq \omega_{MG2}^{\max}, \\ I_{batt}^{\min} \leq I_{batt} \leq I_{batt}^{\max},\ P_{batt}^{\min} \leq P_{batt} \leq P_{batt}^{\max}. \end{array} \tag{15}$$

For HEV structures, the final battery SOC is requested to be the same as the initial SOC,

$$\text{SOC}(T) = \text{SOC}(0). \tag{16}$$

In our study, the driving information of an assigned trip task is completely unknown, which complicates the task of satisfying (16) over the global time horizon via MPC. This problem can be addressed by approximating a cost-to-go function, or by regulating the terminal SOC reference in each control horizon. The second approach is used in this paper. Terminal SOC references in all control horizons are specified to be SOC(0) to guarantee the battery SOC converges to the initial value at the end of the trip. This approach is conservative in utilizing battery energy, but effective in enforcing constraint (16),

$$\overline{\text{SOC}}(k + H_p) = \text{SOC}(0), \tag{17}$$

where $\overline{\text{SOC}}(k + H_p)$ is the terminal SOC reference in control horizon k.

The MPC controller is applied in the supervisory level of the control architecture. The optimization task is solved using DP when predicted vehicle velocities are provided in each receding horizon. For compactness of notation, denote $V_k = V(k\Delta t)$. The control procedure is described as follows [27]:

- A horizon velocity predictor is used to estimate the control horizon driving profile, based on current velocity request V_k and historical velocities. Assume f_p is the velocity prediction function,

$$\begin{aligned} V_{predict} &= f_p\left(\ldots, V_{k-3}, V_{k-2}, V_{k-1}, V_k\right) \\ &= \left\{V_{k+1}, V_{k+2}, \ldots, V_{k+H_p-1}\right\}; \end{aligned}$$

- Given $V_{predict}$, calculate the optimal control policy minimizing the objective function (14);
- Apply the first element of the control policy. Feedback states, and repeat the control procedure.

In this paper, we assume the target vehicle is equipped without any radar, GPS or similar devices. The road grade is zero and future driving profiles are completely unknown.

III. Horizon Velocity Predictors

Three horizon velocity predictors are investigated in this paper: generalized exponentially varying predictor, Markov-chain based predictor and artificial neural network based predictor. The NN-based velocity predictor is implemented for the first time in the HEV energy management problem, and is compared with the other two approaches.

A. Exponentially Varying Velocity Predictors

The relationship between predicted future velocities and total fuel consumption is complex. To obtain an intuitive understanding of this relationship, exponentially varying future velocities were considered in the related literature [28].

In each receding horizon, the exponentially varying horizon velocities are formulated as

$$V_{k+n} = V_k \times (1 + \varepsilon)^n, \text{ for } n = 1, 2, ..., H_p, \tag{18}$$

where V_k is the initial velocity at time step k, ε is the exponential coefficient. Different ε values are considered to examine the sensitivity of fuel economy to the predicted future velocities.

B. Markov-chain Velocity Predictors

Markov-chain is an important methodology used in modeling driving velocities or power demands in HEVs [12]-[15]. In this study, the Markov-chain states and emissions are defined on discrete-valued domains given by vehicle velocity $\overline{V}$ (0 to 36 m/s) and vehicle acceleration $\overline{\alpha}$ in (-1.6 to 1.6 m/s^2), respectively.

Note that the driving behavior studied in this paper is comprehensive in an average sense. Thus, the Markov emission probability distribution is computed from a comprehensive dataset. Eight different driving cycles are included in this dataset, considering both highway and urban driving scenarios. Six of the sample cycles are standard driving cycles and the other two are real collected driving data. Suppose the vehicle velocity and acceleration are discretized into p and q intervals and indexed by i and j, respectively. Velocity at time step k is V_{k+n-1}, and next step acceleration is α_{k+n}, where n indexes time in the receding horizon. The Markov-chain process is defined by an emission probability matrix $T \in \mathbb{R}^{p \times q}$ with

$$[T]_{ij} = \Pr[\alpha_{k+n} = \overline{\alpha}_j | V_{k+n-1} = \overline{V}_i], \tag{19}$$

for $i \in \{1, \ldots, p\}$, $j \in \{1, \ldots, q\}$ and $n \in \{1, \ldots, H_p\}$ [29]. Suppose $p = 60$ and $q = 60$, the probability matrix extracted from the sample dataset is shown in Fig. 1.

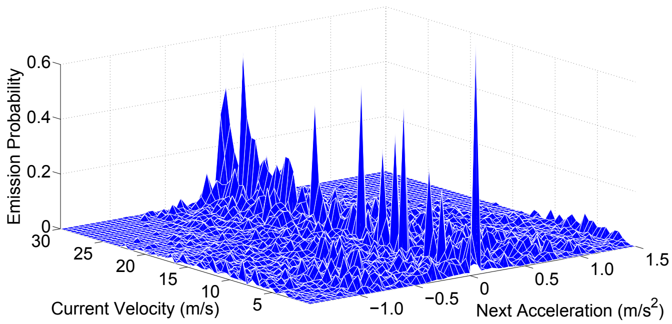

Fig. 1. One-stage Markov emission probability matrix (60×60).

A random Markov-chain model can update its emission probability matrix online by utilizing historical driving data [29]. To reduce complexity and ensure fairness in the comparison, we do not consider such adaptive Markov chain models in this paper. The complexity of the Markov-chain model will also increase if more conditions, such as temperature, position and road grade, are considered.

The precision of Markov-chain velocity predictors relies heavily on the emission probability matrix. To increase the prediction accuracy, 1-stage Markov-chain process can be extended to multi-stage [30], using the following probabilities

$$\begin{aligned}&\Pr[\alpha_{k+n}=\overline{\alpha}_j|V_{k+n-1}=\overline{V}_{i_1},\\&V_{k+n-2}=\overline{V}_{i_2},\ldots,V_{k+n-s}=\overline{V}_{i_s}]\end{aligned}\tag{20}$$

where s is the Markov-chain stage number. When the stages increase, more historical speeds are needed for computing the probability matrix, and the number of Markov states increases exponentially. Thus, the resolution of the emission probability matrix can be improved, leading to higher velocity prediction precision. In the meantime, the sample velocity dataset for generating multi-stage Markov-chain processes needs to cover all possible input states. This requirement is inevitable and may cause both the size of the emission probability matrix and the online computation to grow substantially.

C. NN Velocity Predictors

NNs can be trained to learn a highly nonlinear input/output relationship by adjusting weights to minimize the error between the actual and predicted output patterns of a training set [31]. Three different types of network structures are examined in this paper: back propagation (BP-NN), layer recurrent (LR-NN) and radial basis function neural network (RBF-NN).

A three-layer BP-NN has a hierarchical feed forward network structure, which is also the basis of other network architectures. 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. Output layer yields the desired output patterns. For a BP-NN, the activation function is hyperbolic tangent sigmoid function. The basic formula of BP-algorithm is

$$\begin{aligned}a^1&=\tan sig(n)=\frac{e^n-e^{-n}}{e^n+e^{-n}},\\n&=Wa^0+b,\end{aligned}\tag{21}$$

where a^1 and a^0 are neural outputs of the current layer and prior layer, respectively; n is accumulator output; W is weight and b is bias. The literature shows that a three-layer BP-NN with a sigmoid function as the activation function can approximate any nonlinear systems with arbitrary precision. The main disadvantage of BP-NN is that it has a slow learning convergence and is at risk of getting trapped into local minima.

With a self-connected hidden layer, the LR-NN has an internal state, which allows the network to exhibit temporal dynamic behavior [32]. However, this may increase the training convergence time. The basic formula of the recurrent layer in a LR-NN is

$$\begin{aligned}a^1(t)&=\tan sig(n(t))=\frac{e^n-e^{-n}}{e^n+e^{-n}},\\n(t)&=Wa^0(t)+W'a^1(t-\Delta t)+b,\end{aligned}\tag{22}$$

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

RBF-NN is a widely used feed forward network for time series forecasting. 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 is determined. A RBF-NN usually has better convergence speed and performance compared with BP-NN. In our simulation, the Gaussian function is used as the radial basis function in the hidden layer to activate the neurons, formulated as

$$\begin{aligned}a^1&=\exp\left(-\frac{\parallel n-c\parallel^2}{2b^2}\right),\\n&=Wa^0+b,\end{aligned}\tag{23}$$

where c is the neural net center and b is the spread width. Both c and b can be fit using a gradient descent method.

For velocity prediction purpose, the inputs of neural networks are historical velocity sequences, and the outputs are predicted horizon velocity sequences. Each input-output pattern is composed of a moving window of fixed length, which can be expressed as

$$[V_{k+1},V_{k+2},\ldots,V_{k+H_p}]=f_{\text{NN}}(V_{k-H_h+1},\ldots,V_k),\tag{24}$$

where H_h is the dimension of the input velocity sequence; f_{NN} represents the nonlinear map function of an NN-based predictor. In an MPC framework, the prediction horizon length is H_p, so that the NN velocity predicting process is H_p-step ahead.

The size of the neural network depends on the number of input nodes and the number of hidden nodes. Here we only focus on evaluating the speed prediction performance of neural networks with different numbers of hidden layer nodes. The same driving dataset which is used for Markov emission probability computation is used for networking training. About 85% of the data is employed as training sample to establish the network and the rest 15% is used for performance validation.

IV. Simulation Results and Comparison

In this section we provide a comprehensive comparative analysis of the three velocity forecasting methods. Our discourse begins with an explanation of the training and validation data. Then a definition of performance metrics are defined.

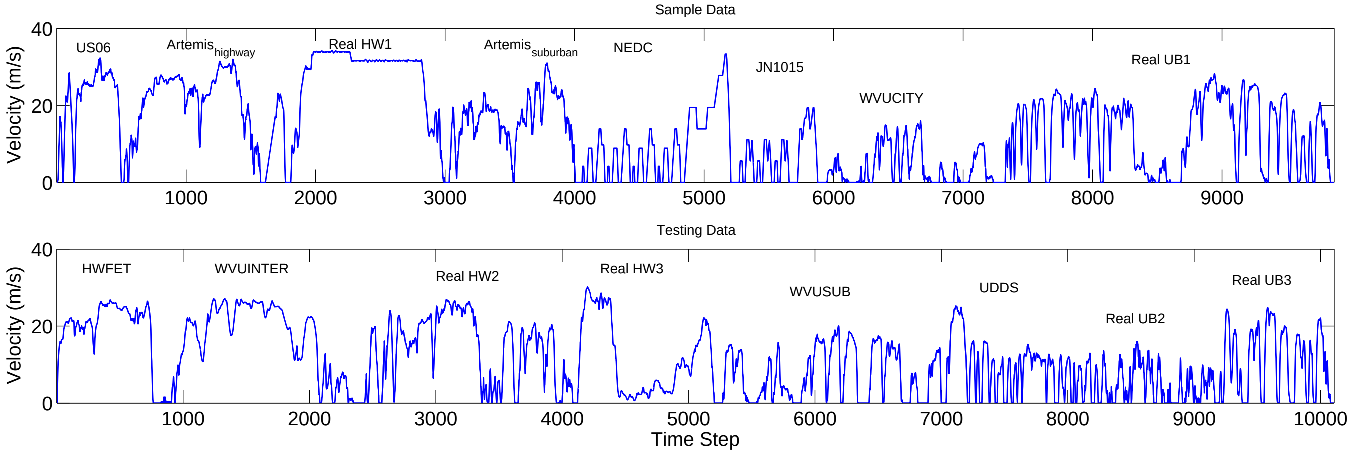

Fig. 2. Sample (training) and testing (validation) dataset. Real HW means data real life highway driving trip; Real UB means real life urban driving trip.

The eight driving cycle samples used for training include four highway types and four urban types. The same data is used both for probability computation in Markov chain approach and network training in NN approach. A different set of eight driving cycles are used for testing and comparing performance of the velocity predictors, which are standard driving cycles HWFET, WVUINTER, UDDS, WVUSUB and real collected driving data Real HW2, Real HW3, Real UB2 and Real UB3. The real collected cycles in sample and testing data are from [33], [34]. All sample and test cycles are concatenated for ease of presentation in Fig. 2.

The vehicle parameters and efficiency maps are obtained from the Toyota Prius HEV model in Autonomie [35]. Simulation was performed on a personal computer with an Intel CoreI i7-3630QM CPU @2.4GHz. The prediction and control horizon length H_p is specified to be 10, and the step computation is completed within 0.75 seconds. The initial SOC and final SOC in all the simulations are set as 0.60. Upper and lower SOC bounds are 0.8 and 0.3, respectively. Simulations for eight testing trips are conducted. Detailed results for the UDDS testing cycle are provided in this section, along with the performance distributions across all eight validation cycles.

The following four metrics are used to assess the tested velocity predictors:

1) Average root mean squared error (RMSE, in m/s) of the predicted velocities in all of the control horizons;
2) Online computation time T_{cal} (in microseconds) of the velocity prediction process at each time step;
3) Violating frequency e of the velocity and acceleration constraints from the predicted velocity sequences;
4) Compensated fuel consumption (in grams), which is the most important metric to evaluate vehicle fuel economy.

Due to the property of prediction, MPC can't guarantee an exact final SOC constraint at the end of a global time horizon. For fair comparison in the following analysis, the produced fuel consumption results are compensated by converting the final battery SOC deviation into equivalent engine fuel. Simulation results for the UDDS cycle are illustrated from subsection A to D. The same simulation procedure is conducted for the other seven testing cycles. A comprehensive fuel consumption performance comparison is demonstrated in subsection E based on all simulation results.

TABLE I
SIMULATION RESULTS OF EXPONENTIALLY VARYING VELOCITY PREDICTOR FOR UDDS

No.	ε	SOC(T)	RMSE	Fuel	T_{cal}
1	-0.03	0.5851	3.6261	259.13	0.023
2	-0.02	0.5830	3.4514	257.26	0.026
3	-0.01	0.5981	3.3790	253.18	0.032
4	0	0.5989	3.4439	256.99	0.034
5	+0.01	0.6154	3.6707	260.75	0.029
6	+0.02	0.6288	4.0667	268.33	0.023

(Fuel consumption is the value after compensation.)

A. Results of Exponentially Varying Predictor for UDDS

As shown in Table I, improved fuel efficiency can be achieved when the predicted velocity decreases slightly, as opposed to an increase or a large decrease. Terminal SOC of MPC simulation becomes larger as ε grows. This is because when the predicted velocity changes aggressively, the engine tends to provide more power. Thus, more engine power will be absorbed by the battery through the generator. The average RMSE reaches its minimum when $\varepsilon = -0.01$, yielding the best fuel economy.

B. Results of Markov-chain Predictor for UDDS

Different types of driving cycles are included in the Markov probability generation dataset: four highway types and four urban types. In this case, the driving behavior we studied is comprehensive and blended, which is the main reason that the 1-stage Markov-chain model has the worst performance in characterizing future horizon velocities, as shown in Table II. The 1-stage model is, however, the most computationally efficient.

For multi-stage Markov-chain models, one requires an often prohibitively large set of driving data to identify the transition

TABLE II
SIMULATION RESULTS OF MARKOV-CHAIN PREDICTOR FOR UDDS

Stage	SOC(T)	RMSE	e (%)	Fuel	T_{cal}
1	0.5755	3.6678	0	273.30	1.647
2	0.5840	3.2328	0	252.10	2.821
3	0.6219	3.2144	0	247.10	2.919
4	0.6132	3.1436	0	248.86	2.831
5	0.6181	3.1133	0	245.39	2.797

matrix. This phenomenon exemplifies the poor applicability of multi-stage Markov-chain velocity predictors. To address this problem in realistic implementation, a sufficiently rich sample database is required. In our work, the emission probabilities for unavailable high-stage Markov states are replaced by the ones derived from lower-stage states to overcome the lack of a large dataset. The maximum Markov stage in this simulation is five, since when the stage number is larger than five, the extracted probability matrix changes negligibly.

Simulation results of Markov-chain velocity predictor for UDDS testing are shown in Table II, from which we can see that more stages results in smaller RMSE. Hence, increased fuel efficiency can be achieved via multi-stage Markov-chain velocity predictors. However, in real-world implementation, it is usually difficult to guarantee that all possible Markov states be contained in the probability generation dataset. Moreover, the magnitude 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. Note that adaptive or self-learning improving approaches are not considered here for fair comparison of the three velocity predictors.

C. Results of NN-based Predictor for UDDS

For NN-based predictors, the dimension of the input, H_h, is specified to be 10. Although more neural nodes result in higher training precision, excessive complexity can lead to over-fitting. Table III shows the comparison results between different types of neural networks with different numbers of nodes. Note that the error caused by velocity/acceleration constraint violations (e) of the predicted horizon speed is included in the average RMSE computation.

The RBF-NN structure achieves better fuel economy than the LR-NN or BP-NN velocity predictors, despite a similar RMSE, since there are substantially fewer constraint violations (e). Besides, RBF network structure needs much less training time and tuning effort.

TABLE III
SIMULATION RESULTS OF NN-BASED PREDICTOR FOR UDDS

Net-Node	SOC(T)	RMSE	e (%)	Fuel	T_{cal}
BP-20	0.6101	2.3221	914 (6.67%)	233.92	0.218
BP-50	0.6074	2.2958	528 (3.85%)	230.46	0.214
LR-20	0.6193	2.2842	819 (5.98%)	233.49	0.214
LR-50	0.5990	2.2851	933 (6.81%)	229.16	0.216
RBF-50	0.6158	2.2811	26 (0.19%)	229.81	0.219
RBF-100	0.6037	2.2747	24 (0.18%)	228.51	0.208

TABLE IV
VELOCITY PREDICTOR COMPARISON FOR UDDS

Methods	SOC(T)	RMSE	e (%)	Fuel	T_{cal}
DP	0.6000	–	–	216.39	–
ECMS	0.6877	–	–	242.40	–
OECMS	0.6004	–	–	217.05	–
DMPC	0.6014	0	0	226.19	–
-0.01 EV	0.5981	3.3790	0	253.18	0.032
1-stage MC	0.5755	3.6678	0	273.30	1.647
5-stage MC	0.6181	3.1133	0	245.39	2.919
RBF-100	0.6037	2.2747	24 (0.18%)	228.51	0.208

D. Velocity Predictor Assessment

DP, deterministic MPC (DMPC) and ECMS are benchmarks. In DMPC, the controller has full knowledge of the horizon velocity profile at each time step. In ECMS, the equivalence factor is tuned from the sampling cycles and implemented in the testing trips as a constant (-357.5). To fully exploit the potential of ECMS, we also tuned the optimal equivalence factor for each testing trip through trial-and-error. This well-tuned ECMS is noted as OECMS and demonstrated for comparison.

The 1-stage and 5-stage Markov-chain velocity predictors are selected to compare with the best velocity predictors from the exponentially varying and NN types, denoted as 1-stage MC, 5-stage MC, -0.01 EV and RBF-100, respectively. These four simulations together with benchmark results are shown in Table IV.

1) Prediction precision: As can be seen from Table IV, the average RMSE of the 1-stage MC velocity predictor is larger than that of the -0.01 EV velocity predictor. This means that the 1-stage MC process behaves poorly in modeling comprehensive and blended driving cycles. The prediction precision from 5-stage MC predictor is improved by 16% compared with 1-stage MC, proving that employing more states can increase the prediction precision. Consequently, the average RMSE of RBF-100 is the minimum one as opposed to the other approaches, although the constraints are violated slightly by 0.18%. This indicates that the RBF-100 predictor is preferable in modeling comprehensive driving behaviors.

The nature of each prediction method can be seen visually in Fig. 3. This figure elucidates how a 1-stage MC is too short-sited to capture velocity profiles longer than a few time steps. One can also observe how the 5-stage MC roughly predicts constant acceleration. In contrast, the RBF-100 captures micro-trip-like behaviors better than the other three.

2) Fuel consumption: From Table IV, we can see that the RBF-100 predictor is the most energy-saving predictor, which is reasonable as it achieves the minimum prediction precision. Fig. 4 shows the SOC trajectories for DP, DMPC and the selected MPC controllers given in Table IV. The difference between the DP and MPC-based trajectories is due to DP being a global optimization; whereas the MPC controller yields a

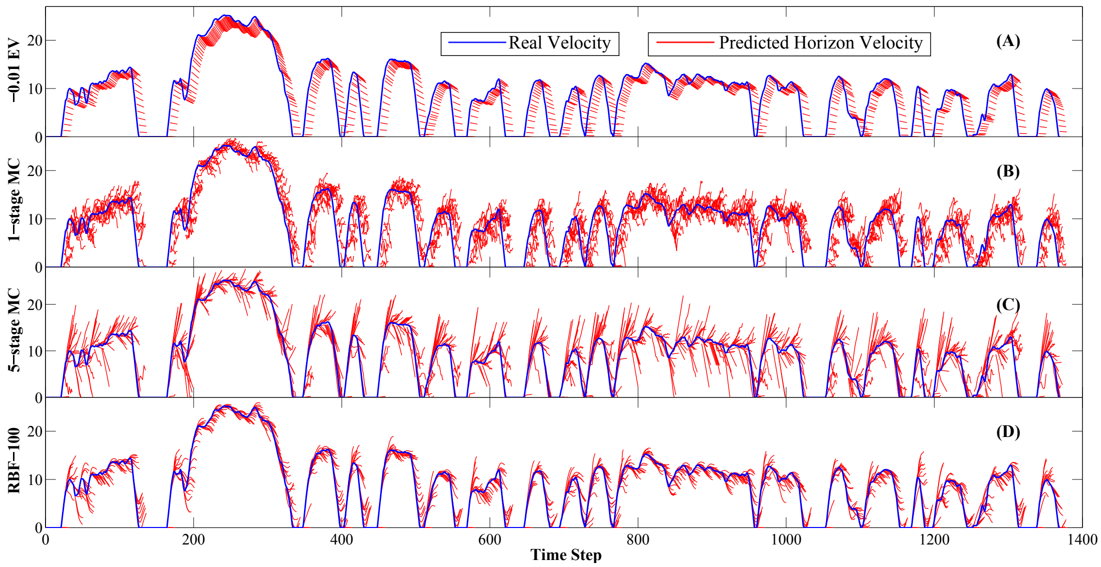

Fig. 3. Prediction result of the four predictors in Table IV for UDDS. The predicted velocity [m/s] is shown across the prediction horizon at each time step.

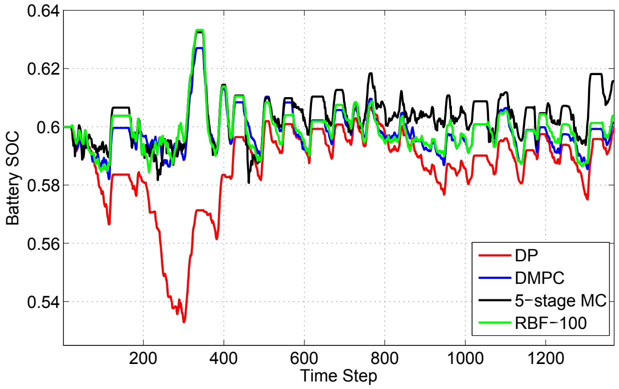

Fig. 4. SOC trajectories of MPC approaches compared with DP result for UDDS testing.

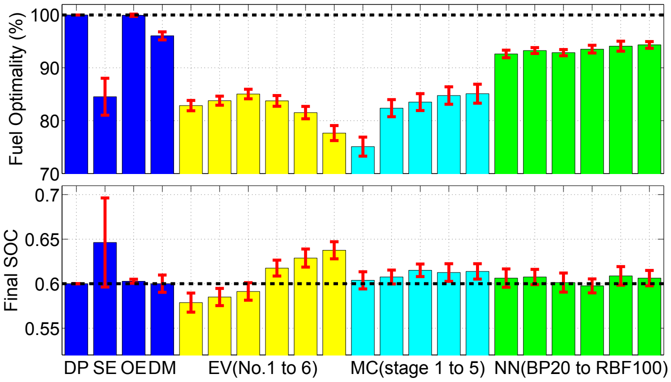

Fig. 5. Normalized fuel consumption and final SOC comparison. SE is simple ECMS; OE is well-tuned OECMS; DM is DMPC.

locally optimal solution for each control horizon. With the same terminal SOC constraint in each receding horizon, the SOC trajectories of RBF-100 MPC and DMPC are overlaid. The SOC trajectory for the 5-stage MC approach differs noticeably from the DMPC benchmark, due to poor velocity prediction.

3) Computation time: From Table IV, it is obvious that the Markov-chain method is computationally heavier than the NN-based and exponentially varying methods. This is because the process of generating a stochastic Markov emission needs additional calculations to form the probability distributions.

4) Constraint violation: The exponentially-varying and Markov-chain approaches do not violate the velocity and acceleration constraints. In the RBF-100 case, the constraints are violated 24 times (0.18%) in the UDDS cycle. The constraint violation pertaining to the RBF-100 predictor is thus negligible.

E. Summarized Simulation Results

Similar outcomes are achieved for the remaining seven tests. Fig. 5 shows the average values and standard deviations of normalized fuel consumption and final SOC. DP and OECMS produce the optimal/near-optimal results if the driving profiles are known a prior. NN-based predictors generally maintain the average fuel optimality over 92%, which is considerably better than the other two predictors and fairly close to DMPC, both in terms of average and standard deviation. Compared with simple ECMS, MPC with NN-based velocity predictors saves more than 7% fuel, and guarantees the final SOC constrained.

Furthermore, the 4% gap between DP and DMPC fuel economy is because DP is a global approach, whereas DMPC conservatively optimizes the fuel consumption locally. Compared with DMPC, the extra fuel consumption caused from velocity predicting errors varies from 2% to 21% based on different methods. Note these results are evaluated across both emission certification cycles and real-world drive cycle data. Consequently, we conclude the NNs provide a promising blend of prediction capability and computational efficiency for MPC energy management in HEVs.

V. Conclusions

This paper presents a comparative study of three classes of velocity predictors for MPC-based energy management in a power-split HEV: generalized exponentially varying, Markov chains, and artificial neural networks. The NN-based horizon velocity predictor is proposed, and the sensitivity of its prediction precision on different NN structures is examined to elucidate a successful template. Generalized exponentially varying and Markov-chain velocity predictors are systematically described and compared with the NN-based predictor in terms of the prediction precision, computational cost, and fuel economy. Results demonstrate that NN-based velocity predictors provide the best overall performance across a range of certification and real-world drive cycles. The study provides better understanding for predictive HEV energy management.

Acknowledgment

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

References

[1] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 352–363, 2004.

[2] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles," *IEEE Control systems*, vol. 27, no. 2, pp. 60–70, 2007.

[3] C.-C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 839–849, 2003.

[4] L. Serrao, S. Onori, and G. Rizzoni, "ECMS as a realization of Pontryagin's minimum principle for hev control," in *American Control Conference (ACC)*, 2009, pp. 3964–3969.

[5] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1279–1287, 2011.

[6] P. Pisu and G. Rizzoni, "A comparative study of supervisory control strategies for hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 506–518, 2007.

[7] A. Sciarretta, L. Guzzella, and M. Back, "A real-time optimal control strategy for parallel hybrid vehicles with on-board estimation of the control parameters," in *Proc. IFAC Symp. Adv. Automotive Contr*, 2004, pp. 19–23.

[8] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, "A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management," *European Journal of Control*, vol. 11, no. 4, pp. 509–524, 2005.

[9] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-based energy management of a power-split hybrid electric vehicle," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 593–603, 2012.

[10] D. Rotenberg, A. Vahidi, and I. Kolmanovsky, "Ultracapacitor assisted powertrains: modeling, control, sizing, and the impact on fuel economy," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 576–589, 2011.

[11] V. Ngo, T. Hofman, M. Steinbuch, and A. Serrarens, "Predictive gear shift control for a parallel hybrid electric vehicle," in *IEEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2011, pp. 1–6.

[12] L. Johannesson, M. Asbogard, and B. Egardt, "Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 71–83, 2007.

[13] S. J. Moura, H. K. Fathy, D. S. Callaway, and J. L. Stein, "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 545–555, 2011.

[14] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1242–1251, 2008.

[15] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *American Control Conference (ACC)*, 2010, pp. 5844–5849.

[16] R. Langari and J.-S. Won, "Intelligent energy management agent for a parallel hybrid vehicle-part I: system architecture and design of the driving situation identification process," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 925–934, 2005.

[17] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[18] P. Santos, A. Martins, and A. Pires, "Designing the input vector to ann-based models for short-term load forecast in electricity distribution systems," *International Journal of Electrical Power & Energy Systems*, vol. 29, no. 4, pp. 338–347, 2007.

[19] Y. Lin, P. Tang, W. Zhang, and Q. Yu, "Artificial neural network modelling of driver handling behaviour in a driver-vehicle-environment system," *International Journal of Vehicle Design*, vol. 37, no. 1, pp. 24–45, 2005.

[20] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport reviews*, vol. 24, no. 5, pp. 533–557, 2004.

[21] J. Park, Z. Chen, L. Kiliaris, M. L. Kuang, M. A. Masrur, A. M. Phillips, and Y. L. Murphey, "Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4741–4756, 2009.

[22] S. Piller, M. Perrin, and A. Jossen, "Methods for state-of-charge determination and their applications," *Journal of power sources*, vol. 96, no. 1, pp. 113–120, 2001.

[23] X. Hu, S. Li, and H. Peng, "A comparative study of equivalent circuit models for li-ion batteries," *Journal of Power Sources*, vol. 198, pp. 359–367, 2012.

[24] X. Hu, S. Li, H. Peng, and F. Sun, "Robustness analysis of state-of-charge estimation methods for two types of Li-ion batteries," *Journal of Power Sources*, vol. 217, pp. 209–219, 2012.

[25] L. Guzzella and A. Amstutz, "CAE tools for quasi-static modeling and optimization of hybrid powertrains," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1762–1769, 1999.

[26] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *American Control Conference (ACC)*, 2008, pp. 3719–3724.

[27] E. F. Camacho, C. Bordons, E. F. Camacho, and C. Bordons, *Model predictive control*. Springer London, 2004, vol. 2.

[28] H. A. Borhan and A. Vahidi, "Model predictive control of a power-split hybrid electric vehicle with combined battery and ultracapacitor energy storage," in *American Control Conference (ACC)*, 2010, pp. 5031–5036.

[29] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6077–6082.

[30] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006.

[31] M. T. Hagan, H. B. Demuth, M. H. Beale *et al.*, *Neural network design*. Pws Pub. Boston, 1996.

[32] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[33] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[34] D. LeBlanc, J. Sayer, C. Winkler, R. Ervin, S. Bogard, J. Devonshire, M. Mefford, M. Hagan, Z. Bareket, R. Goodsell *et al.*, "Road departure crash warning system field operational test: Methodology and results," *University of Michigan Transportation Research Institute, Tech. Rep. UMTRI-2006-9-1, June*, 2006.

[35] R. V. Gopal and A. Rousseau, "System analysis using multiple expert tools," *System*, vol. 1, p. 0754, 2011.