49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Stochastic Model Predictive Control with Driver Behavior Learning for Improved Powertrain Control

M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, I.V. Kolmanovsky

***Abstract*— In this paper we advocate the use of stochastic model predictive control (SMPC) for improving the performance of powertrain control algorithms, by optimally controlling the complex system composed of driver and vehicle. While the powertrain is modeled as the deterministic component of the dynamics, the driver behavior is represented as a stochastic system which affects the vehicle dynamics. Since stochastic MPC is based on online numerical optimization, the driver model can be learned online, hence allowing the control algorithm to adapt to different drivers and drivers' behaviors. The proposed technique is evaluated in two applications: adaptive cruise control, where the driver behavioral model is used to predict the leading vehicle dynamics, and series hybrid electric vehicle (SHEV) energy management, where the driver model is used to predict the future power requests.**

I. INTRODUCTION

Modern automotive vehicles are complex systems where the mechanical components interact with the control electronics and with the human driver. With the constant increase in powertrain complexity, tightening of emissions standards, increase in gas price, and the increased number of vehicle functionalities, advanced control algorithms are needed to meet the specification requirements while keeping sensor and actuator costs limited.

Model predictive control (MPC) [1] is an appealing candidate for control of complex automotive systems, due to its capability of coordinating multiple constrained actuators and optimizing the system behavior with respect to a performance objective. However, MPC requires a model of the dynamics, and, in general, the more precisely the model represents the real dynamics, the better the closed-loop performance is. While suitable models are available for the dynamics of most vehicle components [2], the overall automotive vehicle behavior strongly depends on what the driver is doing. As such, including a prediction model of possible future driver's actions may increase the closed-loop performance of MPC.

In this paper we propose to model the driver as a stochastic process whose output affects a deterministic model of the vehicle. Thus, the obtained model of the vehicle and the driver dynamics is a stochastic dynamical system that requires appropriate stochastic control algorithms. Stochastic control algorithms have been already proposed in automotive applications to tackle the uncertainty arising from the vehicle environment. Concrete examples are stochastic dynamic programming applications to hybrid electric vehicles (HEV) energy management [3], [4] and emission control [5], and linear stochastic optimal control for chassis control [6]. However, stochastic linear control is often inadequate to capture the variety of driver behaviors, while stochastic dynamic programming is numerically intensive and cannot be easily updated if the underlying statistical model changes.

In recent years various stochastic model predictive control algorithms (SMPC) have been proposed, based on different prediction models and stochastic optimal control problems, see [7]–[9] and the references therein. In this paper, we model the deterministic dynamics by a linear system and the stochastic driver model by a Markov chain. The choice of Markov chains to represent driver behaviors is motivated by previous literature [3], [5], and by the approximation properties of Markov chains [10]. The overall model is used in a finite horizon stochastic optimal control problem, where the expected performance objective is optimized, subject to the constraints on states and inputs. Concurrently, the Markov chain modeling the driver is updated online by applying a simple learning algorithm, based on linear filtering of the transition frequencies. This allows the SMPC controller to adapt to changes in the driver behavior.

The paper is structured as follows. In Section II we discuss the driver model based on Markov chains and the corresponding online learning algorithm, and in Section III we introduce the stochastic model predictive control algorithm used in this paper. In Sections IV and V we present the applications of the proposed approach to the adaptive cruise control (ACC) and to energy management in series hybrid electric vehicle (SHEV), respectively. The conclusions are summarized in Section VI.

Notation: $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Z}_{0+}$ denote the set of real, integer, and nonnegative integer numbers, respectively. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes the cardinality. For a vector a, $[a]_i$ is the i^{th} component, and for a matrix A, $[A]_{ij}$ is the ij^{th} element and $[A]_i$ is the i^{th} row. We denote a square matrix of size $s \times s$ entirely composed of zeros by 0_s, and the identity by I_s. Subscripts are dropped when clear from the context.

II. STOCHASTIC DRIVER MODEL AND LEARNING ALGORITHM

We model the actions of the driver on the vehicle by a stochastic process $w(\cdot)$ where $w(k) \in \mathcal{W}$, for all $k \in \mathbb{Z}_{0+}$. We assume that at time k, the value $w(k)$ can be measured, and, with a little abuse of notation, we denote

M. Bichi, G. Ripaccioli, and D. Bernardini are with Dept. Information Engineering, University of Siena, Italy, ripaccioli, bernardini@dii.unisi.it, mbichi@gmail.com

S. Di Cairano is with Powertrain Control R&A, Ford Research and Adv. Engineering, dicairano@ieee.org

A. Bemporad is with Dept. Mechanical and Structural Engineering, University of Trento, Italy bemporad@ing.unitn.it

I.V. Kolmanovsky is with the Dept. Aerospace Engineering, the University of Michigan, Ann Arbor, MI, ilya@umich.edu

by $w(k)$ the measured realization of the disturbance at $k \in \mathbb{Z}_{0+}$. Depending on the specific application, vector $w(k)$ may represent different quantities, such as power request in an HEV, acceleration, velocity, angular rate applied to the steering wheel, or any combination of the above.

For prediction purposes, the random process generating w is modelled as a Markov chain with states $W = \{w_1,\ w_2,\ \dots,\ w_s\}$, where obviously $w_i \in \mathcal{W}$, for all $i \in \{1, \dots, s\}$. The cardinality $|W|$ defines the trade off between stochastic model complexity and its precision. The Markov chain is defined by a transition probability matrix $T \in \mathbb{R}^{s\times s}$, such that

$$[T]_{ij} = \Pr[w(k+1) = w_j | w(k) = w_i], \qquad (1)$$

for all $i, j \in \{1, \dots, s\}$, where $w(k)$ is the state of the Markov chain at time k. By using the Markov chain model, given $w(k) = w_i$, the probability distribution of $w(k+\ell)$ is computed as

$$\Pr[w(k+\ell) = w_j | w(k) = w_i] = \left[\left(T^{\ell}\right)' \epsilon_i\right]_j. \qquad (2)$$

where the ϵ_i is the i^{th} unitary vector, i.e., $[\epsilon]_i = 1$, $[\epsilon]_j = 0$, for all $j \neq i$.

A straightforward extension of this model is to model the stochastic process $w(\cdot)$ by a set of Markov chains T_m, $m = 1, \dots, \mu$, where, in automotive applications, the currently active Markov chain is chosen at every instant depending on current conditions such as velocity, temperature, or road surface.

The Markov chain transition matrix T can be updated online by different learning algorithms, with varying complexity [10], [11]. From a batch of measurement $\{w(k)\}_{k=0}^{L}$, the Markov chain transition matrix T is estimated by

$$[T]_{ij} = \frac{n_{ij}}{n_i},\ i, j \in \{1, \dots, s\}, \qquad (3)$$

where $n_{ij} = |\mathcal{K}_{ij}|$,

$$\mathcal{K}_{ij} = \{k \mid \operatorname*{argmin}_{h\in\{1,\dots,s\}} |w(k) - w_h| = i,\ \operatorname*{argmin}_{h\in\{1,\dots,s\}} |w(k+1) - w_h| = j\},$$

and[1] $n_i = \sum_{j=1}^{s} n_{ij}$, for all $i \in \{1, \dots, s\}$. When new values of $w(\cdot)$ are collected, the Markov chain is updated. The learning procedure used here is detailed in Algorithm 1, where $N \in \mathbb{Z}_{0+}^{s\times s}$ is used to store the measured data.

Algorithm 1 is a linear filtering algorithm that estimates T and, if $w(\cdot)$ is generated by a Markov chain, it can be shown to converge to the correct value, see, e.g., [10], [11]. The parameter $\lambda > 0$ acts as the filter constant, where a lower value increases the convergence speed at the price of higher sensitivity to noise.

[1]Note that minimizer in (4) may not be unique. For those cases, omitted here for the sake of simplicity, a selection rule can be included. For instance, the smallest minimizer can be chosen.

Algorithm 1 On-line driver's model learning procedure

Given T, **at** $k = 0$ **set:**
$N = 0_s$, $\tau = 0$;
$i = \operatorname*{argmin}_{h\in\{1,\dots,s\}} |w(0) - w_h|$;
for all $k \geq 1$ **do**
  $\tau = \tau + 1$;
  $j = \operatorname*{argmin}_{h\in\{1,\dots,s\}} |w(k) - w_h|$;
  $[N]_{ij} = [N]_{ij} + 1$;
  if $\tau = \tau_{max}$ **then**
    for all $h \in \{1, 2, \dots, s\}$ **do**
      $[T]_h = ([N]_h + \lambda[T]_h)(\lambda + \sum_{l=1}^{s}[N]_{hl})^{-1}$;
    end for
    $N = 0$, $\tau = 0$;
  end if
  $i = j$;
end for

III. Stochastic model predictive control

In this paper we apply the SMPC formulation based on scenario enumeration and multi-stage stochastic optimization introduced in [12]. Consider a process whose discrete-time dynamics are modelled by the linear system

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k) \qquad (4a)$$
$$y(k) = Cx(k) + D_1u(k) + D_2w(k), \qquad (4b)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the input, $w(k) \in \mathcal{W}$ is an additive stochastic disturbance. The state, input, and output vectors may be subject to the following constraints

$$x(k) \in \mathbf{X},\ u(k) \in \mathbf{U},\ y(k) \in \mathbf{Y},\ \forall k \in \mathbb{Z}_{0+}. \qquad (5)$$

To simplify the exposition, we consider hereafter a scalar disturbance w. However, the approach described below is easily extended to multi-dimensional disturbances. For predicting the evolution of the disturbance w, a time-varying probability vector $p(k) = [p_1(k),\ p_2(k),\ \dots,\ p_s(k)]'$ is introduced, which defines the probability of disturbance realization at time k,

$$p_j(k) = \Pr[w(k) = w_j],\ j = 1, 2, \dots, s, \qquad (6)$$

with $\sum_{j=1}^{s} p_j(k) = 1$, for all $k \in \mathbb{Z}_{0+}$. We model the evolution of $p(k)$ by the Markov chain (1). As a consequence the complete model of plant and disturbance is

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k) \qquad (7a)$$
$$y(k) = Cx(k) + D_1u(k) + D_2w(k) \qquad (7b)$$
$$\Pr[w(k+1) = w_j | w(k) = w_i] = [T]_{ij}. \qquad (7c)$$

The adopted SMPC problem formulation is based on a maximum likelihood approach, where at every time-step an optimization tree (scenario tree) is built using the updated information on the system state and on the stochastic disturbance. Each node of the tree represents a predicted disturbance scenario which is taken into account in the optimization problem. Starting from the root node, which is associated with the current measurement of $w(k)$, a list of candidate nodes is generated by considering all the possible future values of the disturbance, together with their realization probability. Then, the node with maximum probability is added to the tree. The procedure is repeated iteratively, by generating at every step new candidates as children of the last node added to the tree, until the tree contains a n_{max} nodes. Note that every sequence of connected nodes (i.e., a path) in the tree represents a disturbance sequence realization, where the number of nodes is the length of the path.

The constructed tree produces a “multiple-horizon” approach, where different paths may have different prediction horizons. The tree generation algorithm expands the tree in the most likely direction, so that the paths with higher probability are extended longer in the future, since they will have more impact in the performance optimization. Causality of the resulting control law is enforced by allowing only one control move for every node, except leaf nodes (i.e., nodes with no successor) that have no associated control move. The reader is referred to [12] for further details.

Let us introduce the following quantities to formally define the proposed control problem:

- $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$: the set of tree nodes. Nodes are indexed progressively as they are added to the tree (i.e., $\mathcal{T}_1$ is the root node and $\mathcal{T}_n$ is the last node added).
- $x_{\mathcal{N}}$, $u_{\mathcal{N}}$: the state and the input, respectively, associated with node $\mathcal{N}$.
- $\pi_{\mathcal{N}}$: the probability of reaching node $\mathcal{N}$ from $\mathcal{T}_1$.
- $pre(\mathcal{N})$: the predecessor of node $\mathcal{N}$.
- $succ(\mathcal{N}, j)$: the successor of node $\mathcal{N}$ with mode j.
- $\mathcal{S} \subset \mathcal{T}$: the set of leaf nodes, defined as $\mathcal{S} \triangleq \{\mathcal{T}_i \in \mathcal{T},\ i = 1, 2, \dots, n : succ(\mathcal{T}_i, j) \notin \mathcal{T},\ j = 1, 2, \dots, s\}$.

After the tree has been built, the following stochastic MPC optimization problem is solved

$$\min_{\mathbf{u}} \sum_{i \in \mathcal{T} \setminus \{\mathcal{T}_1\}} \pi_i (x_i - x_r) Q (x_i - x_r) + \sum_{j \in \mathcal{T} \setminus \mathcal{S}} \pi_j (u_j' R u_j + y_j' S y_j) \tag{8a}$$

$$\text{s.t.} \quad x_1 = x(k), \quad w_1 = w(k), \tag{8b}$$

$$x_i = A x_{pre(i)} + B_1 u_{pre(i)} + B_2 w_{pre(i)}, \quad i \in \mathcal{T} \setminus \{\mathcal{T}_1\}, \tag{8c}$$

$$y_i = C x_i + D_1 u_i + D_2 w_i,\ i \in \mathcal{T} \setminus \mathcal{S}, \tag{8d}$$

$$x_i \in \mathbf{X},\ i \in \mathcal{T} \setminus \{\mathcal{T}_1\}, \tag{8e}$$

$$u_i \in \mathbf{U},\ y_i \in \mathbf{Y},\ i \in \mathcal{T} \setminus \mathcal{S}, \tag{8f}$$

where $\mathbf{u} = \{u_i\}_{i=1}^{m_u}$ is the decision vector and $m_u = |\mathcal{T}| - |\mathcal{S}|$ is the number of optimization variables. Then, the decision vector element u_1 associated to the root node $\mathcal{T}_1$ of the tree is used as the control input $u(k)$.

Summarizing, the stochastic model predictive control algorithm is composed of three steps. At any control cycle $k \in \mathbb{Z}_{0+}$, (i) the optimization tree is built from the current value of the state and measured disturbance, (ii) problem (8) is solved for the obtained optimization tree, (iii) the input vector $u(k) = u_1$ is applied to the system.

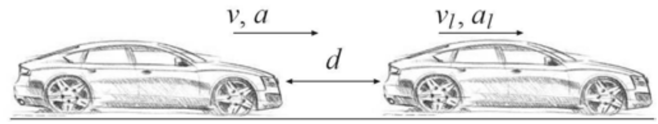


Fig. 1. Adaptive cruise control operation.

In the next sections we demonstrate two applications of SMPC with learning in the automotive domain.

IV. Application to adaptive cruise control

Adaptive cruise control (ACC) [13] extends the functionalities of conventional cruise control. In conventional cruise control, the reference velocity set by the driver is tracked by controlling the throttle, rejecting disturbances such as road slope and air drag. Adaptive cruise control, in addition, enforces a separation distance from the leading traffic to increase driving comfort and safety. In what follows, we consider two vehicles, the follower or host, which is the vehicle equipped with an ACC system, and the leader or target vehicle. The ACC controls the follower acceleration in order to track the desired velocity as close as possible without violating a minimum separation distance between leader and follower, in spite the fact that the acceleration of the leader is time-varying and not controllable.

As a prediction model for the MPC control problem, we consider the scheme represented in Figure 1, where $v(k)$ and $a(k)$ are the speed and the acceleration of the follower, respectively, and $v_l(k)$ is the velocity of the leader. The acceleration $a(k)$ is modeled as the integrator

$$a(k+1) = a(k) + T_s u(k), \tag{9}$$

where $T_s = 1$s is the sampling period, and the control input $u(k)$ is the rate of change of acceleration (jerk). The leader and follower velocities are

$$v(k+1) = v(k) + T_s a(k), \tag{10}$$

$$v_l(k+1) = v_l(k) + T_s a_l(k), \tag{11}$$

where $a_l(k)$ is the leader acceleration, that is assumed to be a stochastic disturbance. Thus, the distance $d(k)$ between the leader and the follower evolves as

$$d(k+1) = d(k) + T_s (v_l(k) - v(k)). \tag{12}$$

The system dynamics are modelled by (4a) where $x(k) = [d(k)\ v(k)\ a(k)\ v_l(k)]'$, $w(k) = a_l(k)$, and

$$A = \begin{bmatrix} 1 & -T_s & 0 & T_s \\ 0 & 1 & T_s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ T_s \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ T_s \end{bmatrix}. \tag{13}$$

In order to guarantee comfort and safety, the state and manipulated input are subject to the constraint (5), where

$$\mathbf{X} \triangleq \{x \in \mathbb{R}^4 : [x]_1 \geq d_{\min}(k),\ 0 \leq [x]_2 \leq v_{ref}\}, \tag{14a}$$

$$\mathbf{U} \triangleq \{u \in \mathbb{R} : u_{\min} \leq u \leq u_{\max}\}. \tag{14b}$$

The minimum separation distance varies with the velocity $v(k)$,

$$d_{min}(k) = \delta + \gamma v(k), \tag{15}$$

where $\delta = 3$m is a constant value that models the minimum distance for the case where $v(k) = 0$, and $\gamma = 2$s is the headaway time-gap, the time needed to reach the current position of the leader. The bound v_{ref} is desired speed set by the driver, which shall not be exceeded, and the jerk is bounded for guaranteeing comfort, $u_{\max} = -u_{\min} = 3$m/s^3. In addition, the follower velocity must be non-negative.

For the ACC control problem, in model (4) the disturbance $w(k)$ is the acceleration $a_l(k)$ of the leading vehicle. This is modelled by a Markov chain that can take $s = 9$ different values. The Markov chain is first trained offline by (3) with acceleration profiles from several standard driving cycles, then the Markov chain is adapted online by Algorithm 1.

In the optimization problem (8) we set

$$x_{ref} = \begin{bmatrix} d_{ref} \\ v_{ref} \\ 0 \\ 0 \end{bmatrix}, Q = \begin{bmatrix} Q_d & 0 & 0 & 0 \\ 0 & Q_v & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ R = Q_u,$$

where Q_d, Q_v, Q_u are the weights on the tracking error of relative distance, on the tracking error of desired velocity, and on jerk, respectively. Also the reference distance is time-varying

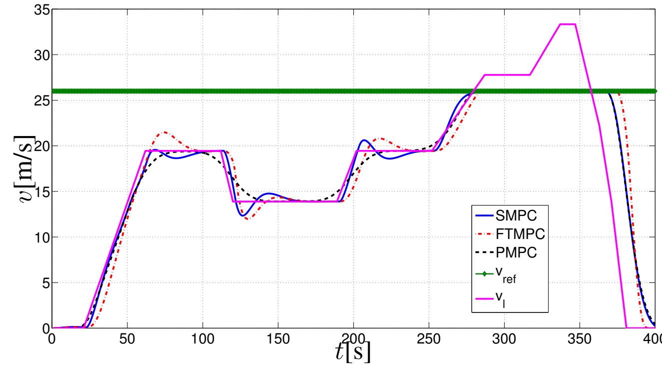
$$d_{ref}(k) = \delta_{ref} + \gamma_{ref} v(k) \tag{16}$$

where $\delta = 4$m and $\gamma = 3$s. By the term that weighs u in (8a), the controller tries to minimize the jerk to provide smooth accelerations that improve comfort and reduce fuel consumption.

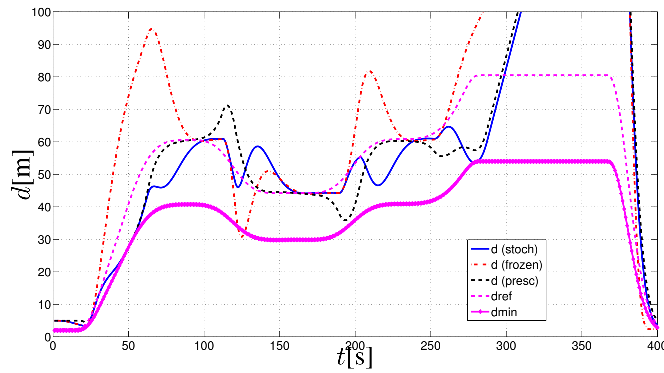
We have simulated the closed-loop system, where the European Urban Driving Cycle (EUDC) is used to model v_l, the speed reference is $v_{ref} = 26$m/s, the weights are $Q_d = 0.1$, $Q_v = 5$, $Q_u = 10^4$, and the optimization tree which defines the optimal control problem is built with $n_{max} = 50$ nodes. We have compared the SMPC controller with a deterministic "frozen-time" MPC (FTMPC), where w is assumed constant in prediction, and with a "prescient" MPC (PMPC), where the disturbance w is exactly known in advance on the entire prediction horizon. Figure 2(a) shows a comparison, in terms of vehicle speed, between the controllers. The SMPC has an intermediate behavior between the FTMPC and the PMPC controller, where the latter knows the variations of v_l in advance. Indeed, SMPC provides better distance and velocity reference tracking than the simpler FTMPC. One can also notice how the SMPC anticipates driver's action with respect to FTPMC, especially when the leader stops accelerating at 60s and in the long braking action at 370s.

V. Application to Energy Management in Hybrid Electric Vehicles

As a second application of SMPC, we consider the energy management problem of a series HEV (hybrid electric vehicle) [14], where only the electric motor is directly connected to the traction driveline. The power supplied to the motor is the combination of the power provided by a battery, and of



(a) Velocity and reference velocity



(b) Distance, reference distance, and distance constraint

Fig. 2. Comparison between the SMPC, FTMPC and PMPC for adaptive cruise control

the power produced by a generator connected to the internal combustion engine (ICE). A preliminary study of SMPC application to this problem was proposed by the authors in [15]. Here, we extend the approach with online learning of the Markov chain, and by considering a refined model with both regenerative breaking (use of the motor to recharge the battery) and non-regenerative braking capabilities.

The purpose of the energy management controller is to minimize the fuel consumption by optimally delivering the power requested by the motor $P_{\rm req}$, which is a function of the driver commands on the gas and brake pedals and of the current vehicle conditions. The energy management system selects how much power $P_{\rm mec}$ must be provided by the ICE through the generator, and how much power $P_{\rm el}$ by the battery. The power balance equation

$$P_{\rm req}(k) = P_{\rm el}(k) + P_{\rm mec}(k) - P_{\rm br}(k), \ \forall k \in \mathbb{Z}_{0+} \tag{17}$$

is enforced, where $P_{\rm br} \geq 0$ is the power drained by conventional friction brakes, in case regenerative braking is not sufficient to provide the desired vehicle braking power.

Since the rotational dynamics of ICE, motor, and generator are much faster than the battery charging dynamics, the main relevant dynamics are the ones of the (normalized) battery state of charge

$$SoC(k+1) = SoC(k) - KT_s P_{\rm el}(k), \tag{18}$$

where $SoC \in [0\ 1]$, $SoC = 1$ corresponds to a fully charged battery, $T_s = 1$s is the sampling period, $K > 0$ is a scalar parameter identified for a generic HEV battery. Note that a positive value of $P_{\rm el}$ indicates that power is provided by the battery to the motor.

The SMPC-based energy management system controls the conventional braking power $P_{\rm br}$ and the variation of the mechanical power provided by the ICE , ΔP, where

$$\Delta P(k) = P_{\rm mec}(k) - P_{\rm mec}(k-1). \quad (19)$$

By combining (17)–(19), we get a model as in (4) with

$$\begin{aligned}
&x(k) = \begin{bmatrix} SoC(k) \\ P_{mec}(k-1) \end{bmatrix},\ u(k) = \begin{bmatrix} \Delta P(k) \\ P_{\rm br}(k) \end{bmatrix}, \\
&y(k) = P_{\rm el}(k),\ w(k) = P_{\rm req}(k), \quad (20)\\
&A = \begin{bmatrix} 1 & KT_s \\ 0 & 1 \end{bmatrix},\ B_1 = \begin{bmatrix} KT_s & -KT_s \\ 1 & 0 \end{bmatrix},\ B_2 = \begin{bmatrix} -KT_s \\ 0 \end{bmatrix}, \\
&C = \begin{bmatrix} 0 & -1 \end{bmatrix},\ D_1 = \begin{bmatrix} -1 & 1 \end{bmatrix},\ D_2 = 1.
\end{aligned}$$

Power request $P_{\rm req}(k)$ is the stochastic disturbance $w(k)$, and battery power $P_{\rm el}(k)$ is the output, since it is uniquely assigned by (17), given $x(k)$ and $u(k)$.

A. Stochastic model predictive controller design

In order to design the stochastic MPC controller, power request, which is decided by the driver, is modeled as a Markov chain (1), accordingly to Section II. In [15] the authors have discussed the case where the Markov chain model is trained offline based on multiple standard driving cycles data, hence representing an average driver behavior, which is an approach similar to the one used in stochastic dynamic programming [3]. In this paper we illustrate the case where the transition probabilities are learned online by Algorithm 1.

SMPC has a clear advantage over stochastic dynamic programming with respect to adaptation to changes in driver's behavior. Stochastic dynamic programming does not easily allow for updates in the control law in reaction to observed changes in the stochastic model, because of the numerically intensive computations required to solve the dynamic program.

With respect to using a fixed Markov chain learned offline as in [15], we show adaptation provides improvements in fuel economy. By learning transition probabilities on line, the Markov chain better represents a specific drive cycle/driver, and the prediction capabilities of SMPC improve consequently. This approach is also more meaningful in "every day driving", which does not exactly match the standard drive cycles, but it has indeed a specific pattern that reflects the commonly travelled road, the local traffic flows, and the specific driving style of the driver.

For predicting $P_{\rm req}$, a Markov chain with $s = 16$ states is used, that is initialized by $T = I$. We run the controller multiple times through the drive cycle used for the simulations in Section V-B in order to learn the transition probabilities.

The stochastic MPC problem (8) is formulated based on model (4) where the dynamic parameters are defined by (20). By solving (8), the SMPC algorithm selects the optimal mechanical power variation and conventional braking power. The desired mechanical power is converted into the optimal engine operating point by the static relation

$$[\tau(k), \omega(k)] = f(P_{\rm mec}(k)),$$

based on the stationary optimal engine power curve. A low-level controller provides tracking of the desired engine operating point. In order to operate the engine always around the optimal power curve, the mechanical power transients should be short, which requires reduced mechanical power variations [15]. This also reduces power losses due to inertia.

Thus, the cost at each node is

$$\begin{aligned} J^*(k) = r_{\Delta P}\Delta P(k)^2 + q_P(P_{\rm mec}(k) - P^*_{\rm mec})^2 + \\ q_{soc}(SoC(k) - SoC_{ref})^2 + r_{br}P_{br}(k)^2, \quad (21) \end{aligned}$$

where $P^*_{\rm mec}$ is the ICE maximum efficiency power, SoC_{ref} is the reference state of charge, the weight $r_{\Delta P}$ enforces smooth mechanical power variations, the weight q_P pushes the system to operate closer to the maximum efficiency power, and q_{soc} and r_{br} penalizes deviations from battery setpoint and the use of the friction brakes, respectively. From (21), the cost function (8a) is defined by setting

$$x_{ref} = \begin{bmatrix} SoC_{ref} \\ P^*_{mec} \end{bmatrix},\ Q = \begin{bmatrix} q_{soc} & 0 \\ 0 & q_P \end{bmatrix},\ R = \begin{bmatrix} r_{\Delta P} & 0 \\ 0 & r_{br} \end{bmatrix},\ S = 0.$$

The constraints in SMPC problem (8) are used to enforce desired operating ranges for the variables, which result in increased battery lifetime and enforced electric and mechanical limits. In details, we impose (14) where

$$\begin{aligned}
\mathbf{X} =&\ \{x \in \mathbb{R}^2 : SoC_{\rm min} \le [x]_1 \le SoC_{\rm max}, \\
&\ 0 \le [x]_2 \le P_{\rm mec,max}\}, \\
\mathbf{U} =&\ \{u \in \mathbb{R}^2 : \Delta P_{\rm min} \le [u]_1 \le \Delta P_{\rm max}, [u]_2 \ge 0\}, \\
\mathbf{Y} =&\ \{y \in \mathbb{R} : P_{\rm el,min} \le [y]_2 \le P_{\rm el,max}\}.
\end{aligned}$$

To guarantee that (17) is enforced at each sample time, the constraint on $\Delta P(k)$ has been implemented as a soft constraint. For additional details on controller tuning see [15].

B. Simulations of the SMPC energy management controller

The quasi-static nonlinear simulation model of a light hybrid vehicle derived from the QSS toolbox [14] is used as for generating simulation data. The controller parameters in the simulations are $SoC_{\rm min} = 0.4$, $SoC_{\rm max} = 0.6$, $P_{\rm mec,max} = 20$ kW, $\Delta P_{\rm max} = -\Delta P_{\rm min} = 5$ kW, $P_{\rm el,max} = -P_{\rm el,min} = 40$ kW. The constraints on SoC are set tight around 50% of battery charge to preserve battery lifetime. The weights in (8a) are $q_{SoC} = 500$, $q_P = 0.2$, $r_{\Delta P} = 0.4$, $r_{br} = 1000$, and the state references are $SoC_{ref} = 0.5$ and $P_{mec,ref} = P^*_{mec} = 15.87$ kW. The optimization tree defining the optimal control problem has $n_{max} = 100$ nodes.

The SMPC controller is again compared with a frozen-time MPC controller (FTMPC) that assumes $P_{\rm req}$ to be constant in prediction, and with a prescient MPC controller (PMPC), that has perfect knowledge of the future power request along the entire prediction horizon. For the simulations, we have used the New European Driving Cycle (NEDC). Figures 3 and 4 show the mechanical power variation and the

TABLE I
SIMULATION RESULTS ON THE NEDC CYCLE

	fuel cons. [kg]	% Fuel Improv.
FTMPC	0.281	–
SMPC (static)	0.243	13.5%
SMPC (adaptive)	0.199	29.2%
PMPC	0.197	29.9%

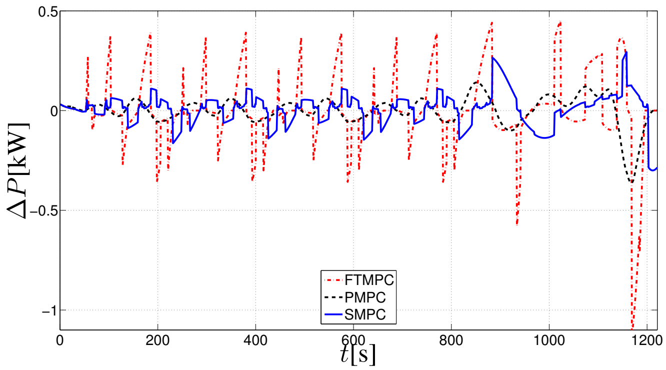


Fig. 3. Mechanical power variation ΔP for SMPC (solid line), PMPC (dashed line), FTMPC (dash-dotted line)

battery state of charge for the three controllers, respectively.

A comparison of the resulting fuel economy on the NEDC cycle (lasting 1220s) is reported in Table I, together with the improvement with respect to the base FTMPC controller. In Table I the fuel consumption of the SMPC controller discussed in [15], where no adaptation was used, is also shown. Even if the fuel consumption is not explicitly minimized, SMPC provides better fuel economy than FTMPC, since SMPC performs better prediction of the future power requests, and the minimization of cost function (21) forces an efficient operation of the powertrain. In fact, the battery power is used to mitigate the ICE power transients, that are in general inefficient, and the controller operates the ICE on the optimal curve, possibly close to the maximum efficiency point $P^*_{\rm mec}$.

VI. CONCLUSIONS

We have developed an approach for control of complex powertrain systems based on modeling the vehicle as a deterministic dynamical system, and the driver as a stochastic process, whose dynamics is updated on line. Stochastic model predictive control is applied to optimize expected performance over a tree of scenarios, while enforcing constraints on states, inputs, and outputs. From a computational viewpoint, by assuming a linear system model we solve the SMPC problem via standard quadratic programming. Compared to stochastic dynamic programming, the proposed controller is easily reconfigurable to changing stochastic parameters and it can more easily handle high order models. The overall procedure has been exemplified in two automotive applications, the adaptive cruise control and the energy management of hybrid electric vehicles. In both cases we have been able to demonstrate improved performance with

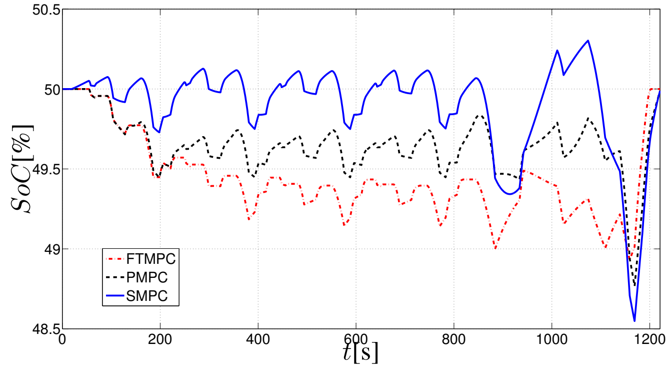


Fig. 4. State of charge SoC for SMPC (solid line), PMPC (dashed line), FTMPC (dash-dotted line)

respect to deterministic MPC schemes, due to the ability of SMPC of providing better predictions without violating causality, as it happens in anticipative MPC schemes.

REFERENCES

[1] C. Garcia, D. Prett, and M. Morari, "Model predictive control: theory and practice–a survey," *Automatica(Oxford)*, vol. 25, no. 3, pp. 335–348, 1989.

[2] L. Guzzella and C. Onder, *Introduction to modeling and control of internal combustion engine systems*. Springer, 2004.

[3] C. Lin, H. Peng, and J. Grizzle, "A stochastic control strategy for hybrid electric vehicles," in *Proc. American Contr. Conf.*, vol. 5, no. 30, 2004, pp. 4710–4715.

[4] 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, p. 71, 2007.

[5] I. Kolmanovsky, I. Siverguina, and B. Lygoe, "Optimization of powertrain operating policy for feasibility assessment and calibration: stochastic dynamic programming approach," in *Proc. American Contr. Conf.*, vol. 2, Anchorage, AK, 2002.

[6] D. Wilson, R. Sharp, and S. Hassan, "The application of linear optimal control theory to the design of active automotive suspensions." *Vehicle Syst. Dyn.*, vol. 15, no. 2, pp. 105–118, 1986.

[7] A. Bemporad and S. Di Cairano, "Optimal control of discrete hybrid stochastic automata," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Morari and L. Thiele, Eds., no. 3414. Springer-Verlag, 2005, pp. 151–167.

[8] P. Couchman, M. Cannon, and B. Kouvaritakis, "Stochastic MPC with inequality stability constraints," *Automatica*, vol. 42, pp. 2169–2174, 2006.

[9] J. Primbs, "Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise," in *Proc. American Contr. Conf.*, New York, NY, 2007, pp. 4470–4475.

[10] B. D. O. Anderson, "From Wiener to Hidden Markov Models," *Control Systems Magazine*, vol. 19, no. 3, pp. 41–51, 1999.

[11] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[12] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Proc. 48th IEEE Conf. on Decision and Control*, Shanghai, China, 2009, pp. 6333–6338.

[13] C. Liang and H. Peng, "Optimal adaptive cruise control with guaranteed string stability," *Vehicle system dynamics*, vol. 32, no. 4-5, pp. 313–330, 1999.

[14] L. Guzzella and A. Amstutz, "QSS-Toolbox Manual," *Institut für Mess-und Regeltechnik, Eidgenössische Technische Hochschule Zürich. Zürich*, 2005.

[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 *Proc. American Contr. Conf.*, Baltimore, MD, 2010, pp. 5844–5849.