

Analytic Solutions to the Dynamic Programming sub-problem in Hybrid Vehicle Energy Management

Viktor Larsson, Lars Johannesson and Bo Egardt

Abstract—The computationally demanding Dynamic Programming (DP) algorithm is frequently used in academic research to solve the energy management problem of an Hybrid Electric Vehicle (HEV). This paper is focused exclusively on how the computational demand of such a computation can be reduced. The main idea is to use a local approximation of the gridded cost-to-go and derive an analytic solution for the optimal torque split decision at each point in the time and state grid. Thereby it is not necessary to quantize the torque split and identify the optimal decision by interpolating in the cost-to-go. Two different approximations of the cost-to-go are considered in the paper: i) a local linear approximation, and ii) a quadratic spline approximation. The results indicate that computation time can be reduced by orders of magnitude with only a slight degradation in simulated fuel economy. Furthermore, with a spline approximated cost-to-go it is also possible to significantly reduce the memory storage requirements. A parallel Plug-in HEV is considered in the paper but the method is also applicable to an HEV.

I. INTRODUCTION

During the last fifteen years significant attention has been given to the topic of optimal energy management for Hybrid Electric Vehicles (HEVs) and Plug-in HEVs (PHEVs). The task is non-trivial to solve since a priori information regarding the future driving conditions is needed. Furthermore, the plant model is generally non-linear with both continuous and integer decisions, i.e. torque split between engine/motor, choice of gear and engine on/off. Many different methods have been proposed for solving the energy management problem. Some examples are: rule based methods, Dynamic Programming (DP), convex optimization, and the Equivalent Consumption Minimization Strategy (ECMS) that is derived from the Pontryagin maximum principle. Refer to [1]–[3] for a review of different methods. This paper will focus exclusively on DP, which has been used in numerous studies [4]–[15]. The main advantage with DP is that it is a very versatile algorithm that can handle a wide range of problem formulations. It provides the global optimal solution, which cannot be guaranteed by a rule based method. In contrast to a convex optimization formulation, DP can handle integer decision variables without any need for approximations or iterative methods. Moreover, an important advantage compared to an ECMS strategy is that state constraints can be treated in a more formal way.

However, DP requires the problem to be gridded in time and states, meaning that the *cost-to-go* is defined over a time and state grid. As a consequence the memory and computational demand will increase exponentially with the number of gridded variables, an effect known as the *curse of dimensionality* [16]. Due to the high computational demand, DP is generally perceived as a method to obtain the optimal fuel economy for a known drive cycle [5], [10], [14], i.e. as a benchmark rather than a method that can be implemented in a commercial system. There are, nonetheless, energy management problems where DP is tractable; it can, for example, be used to precompute an optimal strategy for a frequently driven route, such as a city bus route or a commuter route. Previous studies [9], [12] have shown that a near optimal fuel economy can be obtained by optimizing the energy management based on historical driving data logged along the route. The idea is then to solve the DP problem offline and use the resulting cost-to-go as feedforward information when the vehicle is driven along the route. The computation can for example be performed onboard the vehicle during the start of the trip, or alternatively, it can be solved on a server and the solution can then be transmitted to the vehicle. Regardless if the DP problem is solved as a benchmark, or for use as precomputed feedforward information, it is desirable to keep the computational demand at a reasonable level. Nevertheless, even with battery State of Charge (SoC) as the sole dynamic state, computation time might still be several minutes. The computational bottleneck is that a high number of DP *sub-problem's* must be solved¹, i.e. the optimal control signal(s) must be determined at each point in the time and state grid. The typical methodology used in the literature is to quantize the continuous control signal (the torque split) and then evaluate the gridded cost-to-go through time consuming interpolation [7]; a step that must be repeated for each feasible integer decision. Moreover, the DP memory requirements can easily be several megabytes if the time and state grid is dense.

This paper investigates how the computational demand and the memory requirements of the DP algorithm can be reduced in (P)HEV energy management problems. The key concept is to derive an analytical solution for the continuous control signal in each sub-problem, thereby avoiding the need to quantize the control signal and interpolate in the cost-to-go. However, the idea is not to solve the non-linear Hamilton-Jacobi-Bellman (HJB) partial differential equation, which would be non-trivial even with a simple powertrain model. The proposed method is to approximate the gridded cost-to-go, locally, with a low order polynomial. The local approximation is then only used to

¹Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org."

V. Larsson and B. Egardt are with the Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden. email: {viktor,egardt}@chalmers.se. L. Johannesson is with Viktoria Swedish ICT, Göteborg, Sweden. email: lars.johannesson@viktoria.se.

¹2000 time steps and 2000 state grid points gives 4 million sub-problems.

compute the continuous control signal at each point in the time and state grid. Two different approximations are considered in the paper: i) a local linear approximation, and ii) a quadratic spline approximation. The latter is also beneficial from a memory point of view as the cost-to-go can be stored as a small number of spline parameters at each time step, rather than a vector defined over the gridded state.

Paper Outline: After the introduction a brief review of previous work in the field is presented and the contribution of the paper is put into context. The following two sections presents the vehicle model and formulates the energy management problem. Thereafter the conventional DP algorithm is introduced and two analytic solutions to the DP sub-problem are derived. The succeeding section investigates the solutions obtained with the different DP algorithms. The paper is ended with simulations, discussion and conclusion.

II. PREVIOUS WORK

Several studies have investigated how to reduce the computational demand of DP in an energy management context. The most straightforward approach is to use a sparse grid for the SoC state, although that will degrade the quality of the solution to some extent. For an HEV a very sparse grid can be used without a significant degradation [8], [15]. However, a PHEV can drive longer distances electrically and will therefore have two distinct modes of operation, electric and hybrid vehicle mode. The PHEV cost-to-go will consequently contain two different regions and be less smooth as compared to the cost-to-go of an HEV. The solution will thus tend to degrade more quickly as the grid size is reduced [15]. A more sophisticated approach is to use iterative DP [13], where the idea is to solve the problem recursively using a sparse grid; after each iteration the position of the grid points are then updated and focused around the optimal state trajectory obtained at the previous iteration.

The most promising technique to reduce computational demand, for a specific grid size, is to minimize the use of interpolation when solving the DP sub-problem. In [14], [17] the idea is to avoid all interpolation and only consider the set of discrete control signals that brings the plant model to grid points where the cost-to-go is defined, meaning that it can be evaluated directly. The corresponding disadvantage is that the relevant set of discrete control signals must be computed for every sub-problem. An alternative approach is to derive an analytic solution for the continuous control signal in each sub-problem. Such a solution can for example be obtained if a local linear approximation of the cost-to-go is used. The main advantage with this method is that there is no need to grid the continuous control signal and consequently only one interpolation in the cost-to-go is required for every feasible integer decision. This methodology was investigated by the authors in [8], where a parallel HEV was considered and the battery current was assumed to be quadratic in electric motor torque at a given motor speed. The results showed that the DP problem, with a sparse grid, could be solved in a less than one second for half an hour of driving.

The previously described approaches reduces computation time but not necessarily the memory requirements of the

cost-to-go; if a dense grid is used several megabytes can be required. Nonetheless, within the energy management field not much attention has been given to this topic. The typical approach to address DP memory requirements in other fields is to use approximate or neuro DP [18]–[20], where the key concept is to approximate the cost-to-go, using for example neural networks, splines or other basis functions. The idea is then to only save the function parameters at each time step, rather than all the values in the cost-to-go grid. In [15] the authors proposed a method to approximate the cost-to-go with cubic splines, mainly as a way to reduce the memory storage requirements (the spline approximation was not used to compute the continuous control signal).

The main contribution of this paper is to combine and extend the previous work by the authors [8], [15]. The idea is to further explore the use of a local cost-to-go approximation and the possibility to derive an analytic solution for the continuous control signal in the DP sub-problem. The paper will investigate if a local linear approximation can cause problems with the numerical stability of the DP algorithm, something that was not considered in [8]. Furthermore, the paper will also explore the use of a quadratic spline approximation and the corresponding analytical solution for the continuous control signal.

III. VEHICLE MODELLING

A post transmission parallel PHEV is considered in the paper, meaning that the electric traction motor is mounted directly at the final drive of the front axis. The engine is coupled to the front wheels through a clutch, a five stepped automatic transmission and a final drive. The configuration is shown in Fig. 1 and the key powertrain data is shown in Table I. The main modelling assumptions are presented next.

A simple quasi-static chassis model is considered, as the main focus is on computational aspects rather than vehicle modelling. The longitudinal forces acting on the vehicle chassis are determined using a non-causal and inverse approach, see [7], meaning that the torque demanded at the wheels, T_d , to follow a velocity and road slope trajectory is given by

$$T_d = r_w(1/2\rho_a C_d A v^2 + m_e a + mg(f_r \cos \theta + \sin \theta)), \quad (1)$$

where r_w represents wheel radius; ρ_a density of air; C_d air drag resistance; A vehicle frontal area; f_r rolling resistance; g acceleration of gravity; v velocity; a acceleration; θ road slope; m vehicle mass and m_e equivalent vehicle mass, including moments of inertia of the rotating parts.

The combined electrical power demand of the motor and its inverter is modelled jointly. Two different model complexities are considered in the paper, a quadratic model

$$P_m^{qp} = d_0(\omega_m)T_m^2 + d_1(\omega_m)T_m + d_2(\omega_m), \quad (2)$$

where T_m and ω_m represents motor torque and speed, and a piecewise linear model

$$P_m^{lin} = \max \{d_1^-(\omega_m)T_m + d_2(\omega_m), d_1^+(\omega_m)T_m + d_2(\omega_m)\}, \quad (3)$$

where d_1^- defines the linear slope when $T_m < 0$ and d_1^+ when $T_m \geq 0$. The d coefficients for both models are speed

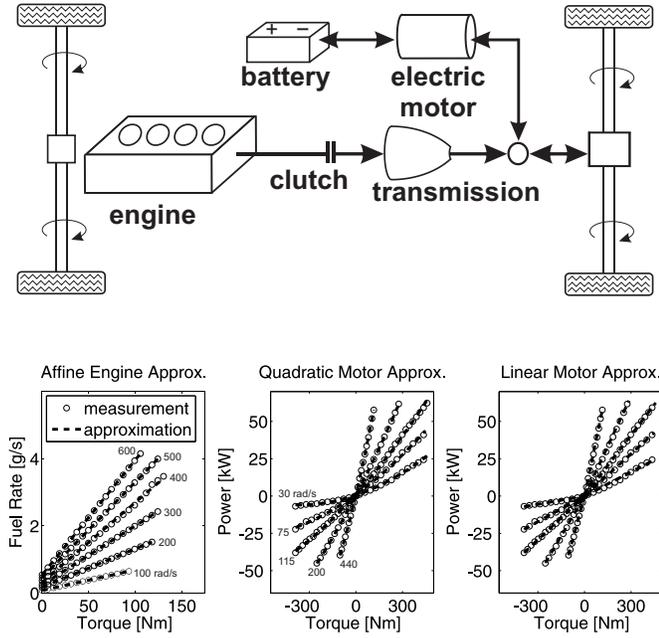


Fig. 1: The PHEV configuration considered and the linear least squares approximations of the engine and the motor. Each line represents different engine/motor speeds.

TABLE I: Powertrain data.

Chassis		Parameters	
Mass	m	1720 kg	
Air drag resistance	$C_d A$	0.68	
Gearbox ratios	r_{gb}	2.6, 1.6, 1.0, 0.7, 0.5	
Final gear ratio	r_f	4.4	
Aux. load.	P_a	200 W	
Battery		Li-Ion	
Battery pack capacity	E_{tot}	8 kWh	
Low/High SoC constraints	$[x_{min}, x_{max}]$	[0.25, 0.90]	
Initial SoC	x_0	0.85	
Final SoC target	x_f	0.30	
Engine		4 Cyl. Spark Ignited	
Max power	$P_{e,max}$	65 kW	
Electric Motor		Permanent Magnet	
Max power	$P_{m,max}$	60 kW	

dependent, non-negative and determined by linear least squares from a power loss map. The instantaneous mass fuel rate of the engine is approximated to be affine in engine torque T_e at given engine speed ω_e . Consequently the instantaneous fuel cost is given by

$$g = c_f (c_0(\omega_e)T_e + c_1(\omega_e))e_{on}, \quad (4)$$

where the fuel price is represented by c_f and the engine state by $e_{on} \in \{0, 1\}$. The speed dependent coefficients, c , are

non-negative and determined by linear least squares from a brake specific fuel consumption map. Fig. 1 illustrates the approximations of the motor and the engine. The gear ratio and efficiency of the final drive at the front axis are represented by r_f and η_f respectively. Note that the efficiency depends on the sign of the torque demand at the wheels, if the torque demand is positive $\eta_f = \eta_{f_0}$, otherwise $\eta_f = \eta_{f_0}^{-1}$. Furthermore, the gears of the automatic transmission, $k \in \{1, 2, \dots, 5\}$, are represented by a drive ratio $r_{gb}(k)$ and a mechanical efficiency $\eta_{gb}(k)$. A Li-Ion battery is considered and it is modelled as an equivalent circuit with a constant internal resistance R_{in} in series with an open circuit voltage V_{oc} that is affine in SoC [21]. Letting the state x be SoC, the resulting state equation becomes

$$\dot{x} = f(x, P_b) = -\frac{V_{oc}(x) - \sqrt{V_{oc}^2(x) - 4R_{in}P_b}}{2R_{in}Q}, \quad (5)$$

where Q denotes cell capacity. With P_a representing auxiliary power loads the requested net battery power is

$$P_b = P_m + P_a. \quad (6)$$

The net torque of the powertrain at the wheels is

$$T_p = \eta_f r_f (T_m + r_e(k)T_e) + T_b, \quad (7)$$

where $r_e(k) = \eta_{gb}(k)r_{gb}(k)$ represents the effective gear ratio. The torque of the friction brakes T_b is non-zero only if the electric motor is torque saturated during regeneration; hence, it is not treated as a free decision variable.

IV. THE ENERGY MANAGEMENT PROBLEM

The problem is to find the control signal trajectory that minimizes the total energy cost for a given drive cycle. Let u_I represent the integer control signal,

$$u_I = e_{on} \cdot k \in \underbrace{\{0, 1, \dots, 5\}}_{U_I}, \quad (8)$$

i.e. choice of engine state e_{on} and gear number k . In terms of the torque split there is only one degree of freedom since the net torque of the powertrain T_p should meet the torque demand T_d . The continuous control signal is thus defined by the motor torque

$$u_c = T_m, \quad (9)$$

meaning that the engine torque is given implicitly by Eq. (7).

Assuming that the drive cycle is known a priori, the resulting energy management problem can be formulated as

$$\begin{aligned} J^* = \min_{u(\cdot)} \{ & G(x(t_f)) + \int_{t_0}^{t_f} g(\omega_e(u_I(t)), T_e(u_c(t))) dt \}, \\ \text{s.t. } & \dot{x}(t) = f(x(t), u_c(t)) \\ & T_p(u_c(t), u_I(t), t) = T_d(v(t), \theta(t)) \\ & u_c(t) \in U_c(\omega_m(t), \omega_e(u_I(t)), x(t)) \\ & u_I(t) \in U_I(v(t), \theta(t)) \\ & x(t) \in [x_{min}, x_{max}] \\ & x(t_0) = x_0 \end{aligned} \quad (10)$$

where $x = \text{SoC}$. The instantaneous fuel cost of the engine is given by $g(\omega_e(u_I), T_e(u_c))$ and $f(x, u_c)$ represents the non-linear state equation. In the above formulation there is no explicit constraint on the final state, instead it is given as a soft constraint that penalizes values below the desired final state, x_f . The final cost function G will therefore represent both the soft final constraint and the cost to recharge the battery up to the initial state x_0 . The motivation for using a soft constraint is that there is no clearly defined lower SoC-limit in terms of battery degradation. Furthermore, when computing the spline approximation it is computationally convenient if the DP cost-to-go is defined numerically between x_{\min} and x_{\max} at all time instances. It would, however, also be possible to specify an explicit final constraint and use level-set functions in the DP recursion to keep track of the backward reachable set [22]. The feasible set for the continuous control signal U_c enforces the speed dependent torque constraints of both the motor and the engine, as well as the state dependent battery power limits. The feasible set for integer decision U_I is defined implicitly by the drive cycle speed and torque demand.

V. THE DYNAMIC PROGRAMMING ALGORITHM

Dynamic Programming is a well know optimal control algorithm based on Bellman's principle of optimality [16]. The main idea in the algorithm is to grid the problem (in time, states and control signals) and divide it into a sequence of smaller problems that are solved recursively, typically backwards in time from the final time step to the first. Each point in the time and state grid defines a DP sub-problem, in which the sum of a stage cost and the cost-to-go (at the next time step and state) is minimized. The stage cost is the cost associated with a control decision at a given time step and state; the cost-to-go represents the cost required to reach the end of the problem along the optimal state trajectory, from a specific time step and state.

A. Solution with the conventional DP algorithm

To solve the energy management problem with the conventional DP algorithm, the problem must first be time discretized² into n time steps and the SoC state gridded into m discrete points, x_1, x_2, \dots, x_m , thus forming a grid of size $n \times m$ over time and state. The cost-to-go matrix $J \in \mathbb{R}^{n \times m}$ is then initialized at time step n with a final cost at each of the discrete points of the state. The problem is thereafter solved recursively backwards in time, over the grid, until the first time step is reached and the cost-to-go matrix is defined at all grid points. To simplify the subsequent presentation consider the following notation:

Definition 1. Let a DP sub-problem be defined as the problem of finding the optimal control signals, $\{u_c^*, u_I^*\} \in \{U_c, U_I\}$, at a specific grid point $[i, j]$, i.e. at time step i and state x_j .

Definition 2. Let $J_i[j]$ denote the value in the cost-to-go matrix at grid point $[i, j]$.

²The discrete time state dynamics are determined using the Euler method with a sample time δt of one second, i.e. $x(t+1) = x(t) + f(x(t), u_c(t))$.

Definition 3. Let $J_i(x)$ denote a value of the cost-to-go at time step i , at a point x between the grid points where the cost-to-go matrix is defined.

Each DP sub-problem, in the time and state grid, is then defined by

$$J_{i-1}[j] \triangleq \min_{\{u_c, u_I\} \in \{U_c, U_I\}} \left\{ \underbrace{g(u_c, u_I)}_{\text{stage cost}} + \underbrace{J_i(x_j + f(x_j, u_c))}_{\text{cost-to-go}} \right\}, \quad (11)$$

where $i = n, n-1, \dots, 2$, $j = 1, \dots, m$; and the initialization of the cost-to-go is defined by $J_n[j] = G(x_j)$. The computational demand of Eq. (11) is high since the cost-to-go is not an analytic function that can be evaluated or differentiated, instead it is a matrix defined only at a finite number of grid points. The cost-to-go is therefore typically evaluated by linear interpolation between the grid points where the cost-to-go matrix is defined [7]. Hence, in order to solve Eq. (11) it is necessary to quantize the continuous control signal u_c into p points (for each feasible gear in U_I) and interpolate in the cost-to-go. The optimal control signals are then found by minimizing over the quantized values of u_c and the feasible integer decisions in U_I . Algorithm 1 summarizes the pseudocode for the conventional DP algorithm.

Algorithm 1 Conventional DP with quantized control signal

```

Initialize cost-to-go matrix at final time sample
for Time steps do
  for Gridded state values do
    for Integer control signal do
      for Quantized continuous control signal do
        Compute stage cost and interpolate in cost-to-go
      end for
      Select continuous control that gives the lowest cost
    end for
    Select integer control that gives the lowest cost
    Update cost-to-go matrix with cost of optimal control
  end for
end for

```

B. Using an analytic solution for the continuous control signal

The key to reducing the computational demand of the DP algorithm is to formulate the right hand side of Eq. (11) as an expression that can be minimized algebraically with respect to the continuous control signal u_c , for a fixed integer decision³ \bar{u}_I . Therefore define

$$h(u_c, \bar{u}_I) \triangleq g(u_c, \bar{u}_I) + \tilde{J}_i(x_j + f(x_j, u_c)), \quad (12)$$

where \tilde{J}_i is some local approximation of the gridded cost-to-go valid near x_j . The minimizing continuous control signal is then obtained by differentiating h with respect to u_c and solving for the case when the derivative is equal to zero,

$$\hat{u}_c(\bar{u}_I) = \arg \min_{u_c} h(u_c, \bar{u}_I), \quad (13)$$

³It is here assumed that $\bar{u}_I \geq 1$, the problem is trivial if the engine is off.

provided that h is strictly convex in u_c . Note that the minimizing control signal is the unconstrained optimum, which might be outside of the torque constraints. The constrained optimum is therefore given by

$$u_c^*(\bar{u}_I) = \begin{cases} \min U_c, & \text{if } \hat{u}_c(\bar{u}_I) < \min U_c(\omega_m, \omega_e(\bar{u}_I), x) \\ \hat{u}_c(\bar{u}_I), & \text{if } \hat{u}_c(\bar{u}_I) \in U_c(\omega_m, \omega_e(\bar{u}_I), x) \\ \max U_c, & \text{if } \hat{u}_c(\bar{u}_I) > \max U_c(\omega_m, \omega_e(\bar{u}_I), x) \end{cases}. \quad (14)$$

The solution to the DP sub-problem is now implicitly parameterized by the integer decision variable u_I . Hence it is possible to redefine Eq. (11) as

$$J_{i-1}[j] \triangleq \min_{u_I \in U_I} \{g(u_c^*(u_I), u_I) + J_i(x_j + f(x_j, u_c^*(u_I)))\}. \quad (15)$$

The difference between Eq. (11) and Eq. (15) might seem subtle, but in terms of computational demand it is significant. In Eq. (11) the continuous control signal u_c must be quantized into p points and the cost-to-go must thus be evaluated through interpolation p times for each feasible gear decision in U_I . Using Eq. (15) only one interpolation is required for every integer decision.

VI. DERIVING AN ANALYTIC SOLUTION FOR THE CONTINUOUS CONTROL SIGNAL

This section will investigate two different approximations that can be used to derive an analytic solution for the continuous control signal. The idea is to exploit that the state x has slow time dynamics compared to the sample time of the system, a valid assumption if the sample time is about one second and the energy buffer is sufficiently large (i.e. a battery is considered and not a flywheel or a super capacitor). A large buffer implies that the state will change only slightly over one time sample and it should therefore be sufficient to use a local approximation of the cost-to-go in each DP sub-problem.

A. Local linear approximation of the cost-to-go

The first approach is to consider a local linear approximation of the cost-to-go and the quadratic motor model given by Eq. (2). The cost-to-go is then described by a first order Taylor expansion around each grid point $[i, j]$,

$$\tilde{J}_i(x_j + f(x_j, u_c)) = J_i[j] + \left. \frac{\partial J_i}{\partial x} \right|_{x_j} \cdot f(x_j, u_c). \quad (16)$$

The partial derivative with respect to the state is for simplicity defined by the forward difference (numerical derivative) of the cost-to-go,

$$s_i[j] = \left. \frac{\partial J_i}{\partial x} \right|_{x_j} = \frac{J_i[j+1] - J_i[j]}{x_{j+1} - x_j}. \quad (17)$$

Substituting Eq. (16) into Eq. (12) yields

$$h_a(u_c, \bar{u}_I) = g(u_c, \bar{u}_I) + J_i[j] + s_i[j] \cdot f(x_j, u_c). \quad (18)$$

The solution to the DP sub-problem can now be determined by minimizing Eq. (18) with respect to u_c . In order to do so rewrite Eq. (18) using Eq. (2) and Eq. (4)-(9), to obtain an

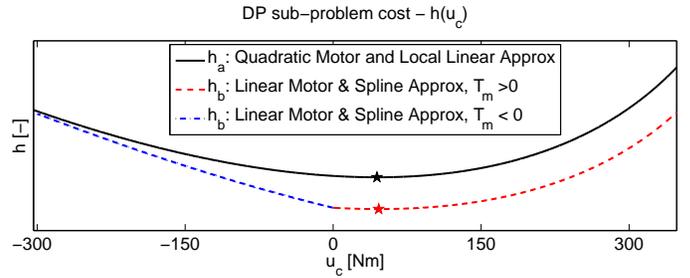


Fig. 2: The cost of the DP sub-problem h plotted vs. the control signal u_c . The star symbol indicates the minimum.

algebraic expression dependent only on the continuous control signal u_c ,

$$h_a(u_c, \bar{u}_I) = c_f \cdot \left(c_0 \frac{\frac{T_d}{\eta_f r_f} - u_c}{r_e(\bar{u}_I)} + c_1 \right) + J_i[j] - s_i[j] \frac{V_{oc}(x_j) - \sqrt{V_{oc}^2(x_j) - 4R_{in}(d_0 u_c^2 + d_1 u_c + d_2 + P_a)}}{2R_{in}Q}, \quad (19)$$

assuming a fixed $\bar{u}_I \neq 0$. Provided that the cost-to-go is decreasing with respect to the state, i.e. $s < 0$, it is straightforward to show that h_a is convex in u_c using composition rules for convex functions [23]. Fig. 2 illustrates the shape of h_a . The minimizing control signal \hat{u}_c is thus given by

$$\begin{aligned} \hat{u}_c(\bar{u}_I) &= \arg \min_{u_c} h_a(u_c, \bar{u}_I) \\ &= -\frac{d_1}{2d_0} + \frac{c_0 \beta_0}{2d_0} \sqrt{\frac{d_0 V_{oc}^2(x_j) + \beta_2 R_{in}}{\beta_1 c_o^2 + d_0 s_i^2[j] r_e^2(\bar{u}_I)}}, \end{aligned} \quad (20)$$

where $\beta_0 = c_f Q$, $\beta_1 = R_{in} \beta_0^2$, $\beta_2 = d_1^2 - 4d_0(d_2 + P_a)$. The optimal control signal u_c^* considering the constraints U_c , is then defined by Eq. (14). The DP algorithm with a local linear approximation of the cost-to-go is summarized in Algorithm 2.

Algorithm 2 DP with local linear approximation

```

Initialize cost-to-go matrix at final time sample
for Time steps do
  for Gridded state values do
    Compute local linear cost-to-go approximation
    for Integer control signal do
      Determine continuous control with analytic solution
      Compute stage cost and interpolate in cost-to-go
    end for
    Select integer control that gives the lowest cost
    Update cost-to-go matrix with cost of optimal control
  end for
end for

```

B. Quadratic spline approximation of the cost-to-go

The second approach is to approximate the cost-to-go with a quadratic spline function. Thereby numerical differentia-

tion can be avoided and the derivative information will be more smooth. Furthermore, with a spline approximation the analytic solution can also take second derivative information into account when solving the DP sub-problem. Appendix A gives a brief introduction to splines and outlines how a spline approximation of the cost-to-go can be obtained by solving a constrained linear least squares problem; a more detailed description of the method is given in [15]. The disadvantage of using a spline is that it is not possible to find an analytic solution to the DP sub-problem with a quadratic model for the electric motor. Therefore the piecewise linear motor model, given by Eq. (3), is considered when the analytic solution is computed. However, to preserve the system dynamics the update of the cost-to-go, as defined by Eq. (15), is still determined by the quadratic motor model, given by Eq. (2).

Suppose that a spline approximation has been computed, the cost-to-go is then locally described by a quadratic function in the neighbourhood of a grid point $[i, j]$,

$$\begin{aligned} \tilde{J}_i(x_j + \tilde{f}(x_j, u_c)) = & s_0 \cdot (x_j + \tilde{f}(x_j, u_c))^2 \\ & + s_1 \cdot (x_j + \tilde{f}(x_j, u_c)) + s_2, \end{aligned} \quad (21)$$

where $\tilde{f}(x, u_c)$ represents the state equation given by Eq. (5) with the piecewise linear motor model defined by Eq. (3). Furthermore, assume that every sub-problem will belong to a specific spline segment, i.e. the possibility that a sub-problem might involve two or more spline segments is neglected. To solve the DP sub-problem rewrite Eq. (12), using Eq. (3)-(9) and Eq. (21), to obtain an expression dependent only on u_c ,

$$\begin{aligned} h_b(u_c, \bar{u}_I) = & c_f \cdot \left(c_0 \frac{\frac{T_d}{\eta_j r_f} - u_c}{r_e(\bar{u}_I)} + c_1 \right) \\ & + s_0 \cdot (x_j + \tilde{f}(x_j, u_c))^2 + s_1 \cdot (x_j + \tilde{f}(x_j, u_c)) + s_2. \end{aligned} \quad (22)$$

If \tilde{J}_i is convex and nonincreasing on a spline segment it is straightforward to show that h_b is convex in u_c using composition rules for convex functions [23]. The shape of h_b is illustrated in Fig. 2 and the minimizing control signal is given by

$$\begin{aligned} \hat{u}_c^\dagger(\bar{u}_I) = & \arg \min_{u_c} h_b(u_c, \bar{u}_I) \\ = & - \frac{d_1^\dagger r_e^2(\bar{u}_I) (R_{in} Q (2s_0 x_j + s_1) - s_0 V_{oc}(x_j))^2}{4R_{in} (c_f c_0 R_{in} Q^2 + r_e(\bar{u}_I) s_0 d_1^\dagger)^2} \\ & + \frac{V_{oc}^2(x_j) - 4R_{in} (P_a + d_2)}{4R_{in} d_1^\dagger}, \end{aligned} \quad (23)$$

where d_1^\dagger represents d_1^+ or d_1^- and $\hat{u}_c^\dagger(\bar{u}_I)$ represents the two solutions, \hat{u}_c^+ and \hat{u}_c^- . The optimal unconstrained control signal is thus given by

$$\hat{u}_c(\bar{u}_I) = \begin{cases} \hat{u}_c^+(\bar{u}_I), & \text{if } u_c^+ > 0 \wedge u_c^- > 0 \\ \hat{u}_c^-(\bar{u}_I), & \text{if } u_c^+ < 0 \wedge u_c^- < 0 \\ 0, & \text{if } u_c^+ < 0 \wedge u_c^- > 0. \end{cases} \quad (24)$$

The optimal control signal with respect to the constraints u_c^* is then defined by Eq. (14). The DP algorithm with a spline approximation of the cost-to-go is summarized in Algorithm

3. Finally, note that an additional advantage with a spline approximation is that the storage requirements are reduced as the cost-to-go, at each time step, is represented by a small number of spline parameters, rather than a vector defined over all the gridded values of the state.

Algorithm 3 DP with spline approximation

```

Initialize cost-to-go vector at final time sample
for Time steps do
  Compute spline approximation of cost-to-go vector
  for Gridded state values do
    for Integer control signal do
      Determine continuous control with analytic solution
      Compute stage cost and interpolate in cost-to-go
    end for
    Select integer control that gives the lowest cost
    Update cost-to-go vector with cost of optimal control
  end for
  Store spline parameters as cost-to-go representation
end for

```

VII. THE BEHAVIOUR OF THE COST-TO-GO

This section will present the cost-to-go obtained at a few different time steps for the three different DP algorithms, i.e. the conventional algorithm with a gridded control signal and the two approaches with an analytic solution to the DP sub-problem. To facilitate a fair comparison the same initialization of the cost-to-go and the same drive cycle, shown in Fig. 6, is used for all algorithms. A PHEV is considered and the resulting cost-to-go is therefore representative for an energy management strategy where the battery is net discharged.

A. Conventional algorithm with gridded control signal

The shape of the cost-to-go for the conventional DP algorithm, with 2000 grid points for the state, is shown to the left in Fig. 3. The initialization of the cost-to-go at the final sample is shown as the solid black line, which is defined by two separate slopes. The steeper slope enforces the soft final constraint, i.e. it penalizes low final states, and the more gentle slope represents the cost to recharge the battery up to x_0 at the end of the drive cycle. Note that the region where the cost-to-go has a constant and gentle slope can be interpreted as the region in the state space from which it is possible to reach the end of drive cycle using mainly electric energy. Consequently, as the DP iterations progress backwards the constant and gentle slope will gradually vanish from the solution. This effect is clearly seen by investigating the numerical derivative of the cost-to-go with respect to the state, shown in the right plot of Fig. 3. It is also clear that the overall shape of the cost-to-go remains convex throughout the backward iterations, i.e. the derivative is in general monotonically increasing with respect to the state.

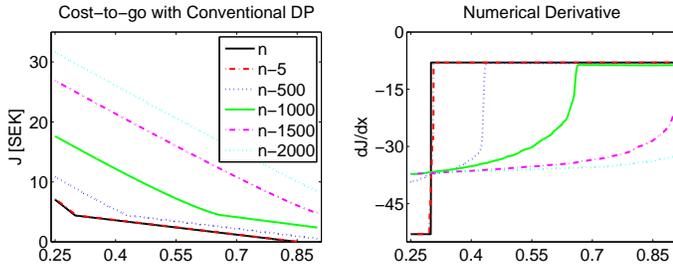


Fig. 3: Conventional DP with 2000 grid points for the state. The cost-to-go J and its (numerical) derivative $\frac{\partial J}{\partial x}$ at a few different time steps, n represents the final time step of the drive cycle.

B. Analytic solution with local linear approximation

The results obtained with an analytic solution to the DP sub-problem and the local linear cost-to-go approximation are illustrated in Fig. 4. It is clear that the overall outcome is very similar to what was obtained with the conventional DP algorithm. Nevertheless, with a densely gridded state there can be numerical problems with the cost-to-go, as exemplified by the oscillatory behaviour in the lower left plot of Fig. 4 where 2000 state grid points are used. In contrast, if the number of grid points is reduced to 1000 the oscillatory behaviour is no longer present, as seen in the plot to the right.

The numerical problems can occur if it is possible for the state trajectories of two adjacent grid points to cross during a time step, thereby violating Bellman's principle of optimality. A (very) conservative bound on the maximum number of state grid points m_{\max} is thus defined by the battery power limits,

$$\frac{x_{\max} - x_{\min}}{m_{\max}} > 2\delta t \cdot \max_{u_c} |f(x, u_c)|, \quad (25)$$

where δt represents the sample time and x_{\min}/x_{\max} the minimum/maximum values of the state in the grid; for the modelling assumptions in this paper $m_{\max} \approx 133$. However, practical experience has shown that it is possible to use many more grid points without numerical problems. If a dense grid is used, i.e. $m \gg m_{\max}$, the principle of optimality is typically compromised in regions where the cost-to-go derivative is not monotonically increasing with respect to the state, something that might occur locally. In such a case a local non-monotonicity tends to be amplified due to the local linear approximation of the cost-to-go. Nevertheless, the numerical problems are typically of transient character and tends to vanish from the solution after a few iterations. The rationale is that it is only the control decision that is determined based on the local linear approximation, interpolation is still used to update the cost-to-go in Eq. (15). If a dense grid is used, a simple but effective approach to suppress this type of behaviour is to smooth the numerical cost-to-go derivative before the optimal continuous control signal is computed.

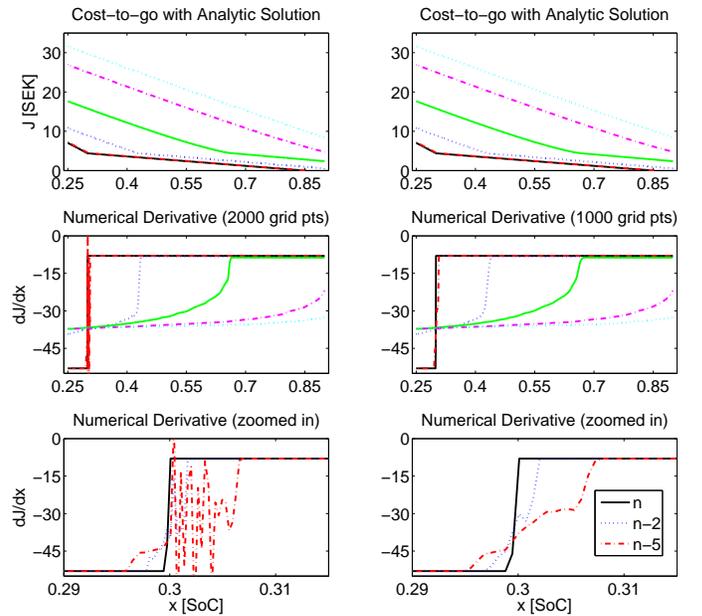


Fig. 4: Local linear approximation. The cost-to-go J and its (numerical) derivative $\frac{\partial J}{\partial x}$ at a few different time steps. The state is gridded with 2000 points in the plots to the left and 1000 points in the plots to the right.

C. Analytic solution with spline approximation

The spline approximated cost-to-go and its derivative with respect to the state at a few different time steps are depicted in Fig. 5. Four quadratic splines are used to represent the cost-to-go and 2000 state grid points were used during the DP recursion. Furthermore, to the right in Fig. 5 the spline approximated cost-to-go is shown overlaid with the cost-to-go obtained with the conventional DP algorithm. The overall shape and behaviour of the two is clearly very similar. The only substantial difference is seen during the first few backward iterations, as illustrated in the lower right plot of Fig. 5. The discrepancy is explained mainly by the initialization of the cost-to-go, which is defined by two affine functions, meaning that the derivative is non-smooth. A quadratic spline approximation cannot accurately describe the associated initial behaviour of the cost-to-go. However, in contrast to the local linear approximation, there are no numerical problems visible in the derivative of the cost-to-go. This is an intrinsic characteristic of the spline approximation, which is constrained to be convex and thus has a derivative that is monotonically increasing with respect to the state.

VIII. SIMULATION RESULTS

The simulation study is performed to assess how the optimal state trajectory and fuel consumption are affected by the cost-to-go of the three DP algorithms. However, the analytic expression for the continuous control signal u_c are not completely equivalent for the local linear cost-to-go approximation, Eq. (20), and the spline approximation, Eq. (23)-(24), since

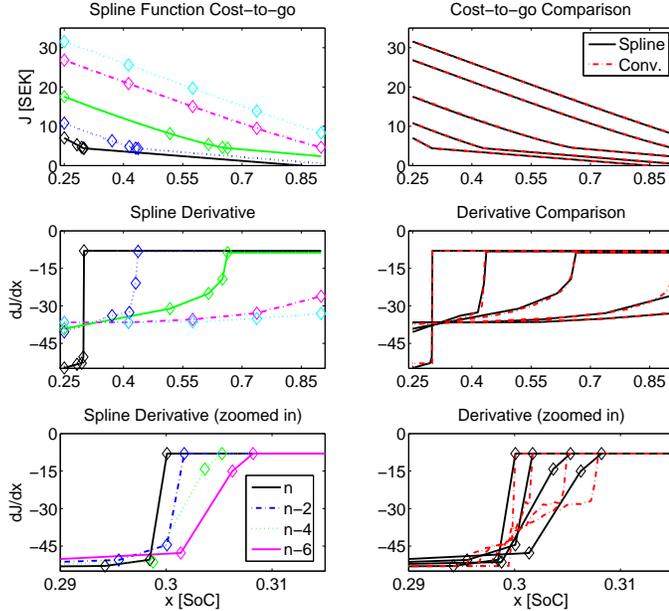


Fig. 5: The spline approximated cost-to-go \tilde{J} and its derivative $\frac{\partial \tilde{J}}{\partial x}$ at a few different time steps. The diamond symbols indicate the spline knot points. The plots to the right depict the spline approximated cost-to-go together with cost-to-go obtained with conventional DP.

different model orders are considered for the electric traction motor. The continuous control signal is therefore quantized to ensure that the only difference between the simulations is solely due to differences in the cost-to-go and not on the expressions that are used to decide the control signal.

The vehicle model presented in Section III, with the quadratic motor model, is simulated along the drive cycle shown in Fig. 6. The optimal control decision at time sample i is determined by

$$\{u_c^*, u_I^*\}_i = \arg \min_{\{u_c, u_I\} \in \{U_c, U_I\}} \{g(u_c, u_I) + J_{i+1}(x_{i+1}(u_c, u_I))\}, \quad (26)$$

where u_c is quantized and the cost-to-go is evaluated through linear interpolation, except for the spline approximated cost-to-go which can be evaluated directly.

The simulated state trajectories are shown in Fig. 6 and the resulting fuel consumption is summarized in Table II. It is clear that the results are almost identical for all three algorithms. The numerical problems encountered in the cost-to-go of the local linear approximation with 2000 state grid points, shown in Fig. 4, does not affect the overall result very much since the problems are only present at a few time steps near the end of the drive cycle. If the number of state grid points is reduced to 1000, numerical problems are no longer an issue as seen in Fig. 4; the consumption is nevertheless slightly higher as the accuracy of the solution is degraded when the number of state grid points is reduced. The fuel consumption for the spline approximated cost-to-go is somewhat higher than for

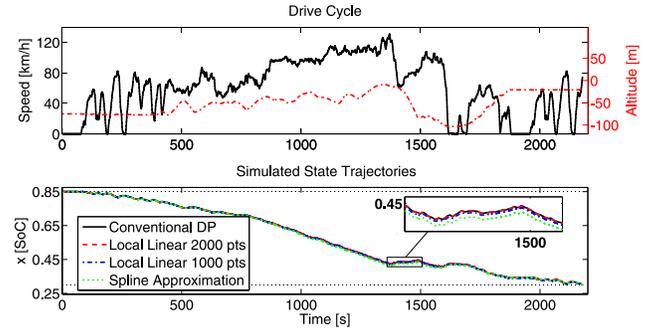


Fig. 6: The drive cycle and the simulated SoC trajectories.

TABLE II: Simulation results and computation time.

DP Algorithm	Computation Time	Cost-to-go Repres.	Final SoC	Relative Fuel Cons.
Conventional	100% (490s)	2000 grid pts.	0.3001	100.00%
Local linear	1.4% (6.9s)	2000 grid pts.	0.3001	100.04%
	1.0% (5.1s)	1000 grid pts.	0.3001	100.07%
Spline	2.5% (11.9s)	4 splines	0.3002	100.19%

the local linear approximation, something that is reasonable considering that the cost-to-go is represented by four quadratic splines rather than thousands of grid points. The difference in fuel consumption will decrease if the number of splines is increased.

IX. COMPUTATION TIME

The computation time⁴ for the three different DP algorithms is illustrated in Table II. All computations are performed using MATLAB^{®5}, except for the spline approximation which is computed using a dedicated C-routine generated using CVX Gen [24]. The results indicate that the computation time is decreased by a factor of seventy using the local linear approximation and a factor of forty with the spline approximation, both compared to the conventional algorithm with an identical grid size. The significant reduction compared to the conventional algorithm is explained by the fact that there is no need to quantize the continuous control signal when the cost-to-go is approximated. Consequently, there is a drastic reduction in the use of interpolation. The difference in computation time between local linear and the spline approximation DP algorithms is explained by the complexity of the cost-to-go approximation. The local linear approximation is computed by numerical differentiation. It is more computationally demanding to compute a spline approximation, i.e. to solve a constrained linear least squares problem.

⁴Using a desktop PC with Intel i5 2320 Processor and 8GB DDR3 RAM.

⁵The Matlab implementations are vectorized in order to be computationally efficient, the only for loops used is with respect to time and integer decision.

X. DISCUSSION

It is clear that the computational burden of the DP algorithm can be reduced significantly if the continuous control signal in the DP sub-problem is obtained analytically. The downside is that it is not possible to develop a generic method as a specific analytic solution must be derived for each particular powertrain configuration. To find a solution it is necessary to make a trade-off between powertrain model accuracy and the quality of the cost-to-go approximation; an oversimplified powertrain model will result in a degraded solution and a crude cost-to-go approximation can cause problems with numerical stability. The main disadvantage with a local linear cost-to-go approximation is that it can lead to a violation of the principle of optimality, at least if the change in the state over a time step is comparable to the distance between the state grid points. Consequently the method is not very suitable for small energy buffers and longer sample times.

The significant reduction in computation time and memory demand means that it is possible to implement the proposed method in a vehicle ECU. However, as the proposed method is essentially an off-line computation it would also be possible to perform the computation at a higher level. For example, with an app on an external device such as smartphone or on a server. The solution can then be transmitted to the vehicle over the cellular network.

XI. CONCLUSION

The paper has investigated the possibility to use an analytic solution for the continuous control signal when solving the DP sub-problem in a hybrid electric vehicle energy management problem. To derive an analytic solution two different approximations of the cost-to-go is considered; i) a local linear approximation, and ii) a quadratic spline approximation. The results indicate that the computation time can be decreased by almost two orders of magnitude with only a slight degradation in simulated fuel economy. Furthermore, with a spline approximated cost-to-go the memory storage requirements are also reduced with about two orders of magnitude. A post-transmission parallel PHEV was considered in the paper, but the method will work equally well for an HEV or another powertrain configuration, as long as an analytic solution can be found.

REFERENCES

- [1] F. R. Salmasi, "Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2393–2404, Sep. 2007.
- [2] A. Sciarretta and L. Guzzella, "Control of Hybrid Electric Vehicles," *IEEE Control Systems Magazine*, vol. 27, no. 2, 2007.
- [3] S. G. Wirasingha and A. Emadi, "Classification and Review of Control Strategies for Plug-In Hybrid Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, 2011.
- [4] C.-c. Lin, H. Peng, and J. W. Grizzle, "A Stochastic Control Strategy for Hybrid Electric Vehicles," in *Proceedings of the American Control Conference*, 2004.
- [5] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, "A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management," in *Proceedings of the 44th IEEE Conference on Decision and Control*, vol. 11, no. 4-5, Oct. 2005.
- [6] Q. Gong, Y. Li, and Z.-R. Peng, "Optimal Power Management of Plug-in HEV with Intelligent Transportation System," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2007.
- [7] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems*, 2nd ed. Springer Verlag, 2007.
- [8] L. Johannesson and B. Egardt, "Approximate Dynamic Programming Applied to Parallel Hybrid Powertrains," in *Proceedings of the 17th IFAC World Congress*, no. 978, 2008.
- [9] L. Johannesson, S. Pettersson, and B. Egardt, "Predictive energy management of a 4QT series-parallel hybrid electric bus," *Control Engineering Practice*, vol. 17, no. 12, Dec. 2009.
- [10] D. Ambühl and L. Guzzella, "Predictive Reference Signal Generator for Hybrid Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, 2009.
- [11] V. Ngo, T. Hofman, M. Steinbuch, and A. Serrarens, "Optimal Control of the Gearshift Command for Hybrid Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, Oct. 2012.
- [12] V. Larsson, L. Johannesson, B. Egardt, and S. Karlsson, "Commuter Route Optimized Energy Management of Hybrid Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, 2014.
- [13] H.-G. Wahl and F. Gauterin, "An Iterative Dynamic Programming Approach for the Global Optimal Control of Hybrid Electric Vehicles under Real-time Constraints," in *Proceedings of the IEEE Intelligent Vehicle Symposium*, 2013.
- [14] R. M. Patil, Z. Filipi, and H. K. Fathy, "Comparison of Supervisory Control Strategies for Series Plug-In Hybrid Electric Vehicle Powertrains Through Dynamic Programming," *IEEE Transactions on Control Systems Technology*, 2013.
- [15] V. Larsson, L. Johannesson, and B. Egardt, "Cubic Spline Approximations of the Dynamic Programming Cost-to-go in HEV Energy Management Problems," in *Proceedings of the 13th European Control Conference*, 2014.
- [16] R. Bellman and S. Dreyfus, *Applied Dynamic Programming*. Princeton University Press, 1962.
- [17] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, no. 2, pp. 245–254, Feb. 2009.
- [18] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont Massachusetts: Athena Scientific, 1996.
- [19] J. A. Tejada-Guibert, S. a. Johnson, and J. R. Stedinger, "The Value of Hydrologic Information in Stochastic Dynamic Programming Models of a Multireservoir System," *Water Resources Research*, vol. 31, no. 10, pp. 2571–2579, Oct. 1995.
- [20] Y. Cai and K. L. Judd, "Dynamic programming with shape-preserving rational spline Hermite interpolation," *Economics Letters*, vol. 117, no. 1, pp. 161–164, Oct. 2012.
- [21] X. Hu, S. Li, and H. Peng, "A comparative study of equivalent circuit models for Li-ion batteries," *Journal of Power Sources*, vol. 198, Jan. 2012.
- [22] P. Elbert, S. Ebbesen, and L. Guzzella, "Implementation of Dynamic Programming for n-Dimensional Optimal Control Problems With Final State Constraints," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, 2013.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [24] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, Nov. 2011.

APPENDIX A
COMPUTING THE SPLINE APPROXIMATION

The idea is to compute a spline approximation, \tilde{J}_i , of the gridded cost-to-go, J_i , at time step i . The spline function is here defined as a piecewise quadratic function, $\tilde{J}_i : [x_0, x_k] \rightarrow \mathbb{R}$, defined over a set of k disjoint subintervals called knot spans, $[x_q, x_{q+1})$, $q = 0, \dots, k-1$,

$$\tilde{J}_i = \begin{cases} P_1(x) = s_0^1 x^2 + s_1^1 x + s_2^1 & , x_0 \leq x < x_1 \\ \dots & , \dots \\ P_k(x) = s_0^k x^2 + s_1^k x + s_2^k & , x_{k-1} \leq x < x_k, \end{cases} \quad (27)$$

where the points x_0, \dots, x_k are the spline knot points. The spline function is required to have C1 continuity, i.e. continuous in the function and in the first derivative. Furthermore, the spline function is required to be convex, meaning that first derivative of the spline function should be monotonically increasing. Hence, the quadratic polynomials must satisfy

$$P_q(x_q) = P_{q+1}(x_q), \quad q = 1, \dots, k-1 \quad (28)$$

$$P'_q(x_q) = P'_{q+1}(x_q), \quad q = 1, \dots, k-1 \quad (29)$$

$$-P''_q \leq 0, \quad q = 1, \dots, k. \quad (30)$$

The spline approximation of the cost-to-go vector J_i is then defined by a constrained linear least squares problem

$$\begin{aligned} \min_S \quad & \|C(x_{1:m})S - J_i(x_{1:m})\|_2^2 \\ \text{s.t.} \quad & A_z(x_{1:k-1})S = 0, \quad z = 1, 2 \\ & BS \leq 0, \end{aligned} \quad (31)$$

where S is a vector containing the spline coefficients and the A_z matrices enforces the continuity constraints, i.e. Eq. (28)-(29). The B matrix enforces the convexity constraint of Eq. (30) and C is a block matrix, describing the state values in the spline at the gridded values of the state where the cost-to-go is defined. The methodology and the selection of spline knot points are described more in depth in [15].



Viktor Larsson received the M.Sc. degree from Luleå University of Technology, Sweden, in 2008. During 2009-2014 he was a Ph.D.-student in the automatic control group at the Department of Signals and Systems at Chalmers University of Technology, Göteborg, where he received the Ph.D. degree in 2014. His research interest includes optimal control of hybrid and plug-in hybrid electric vehicles.



Lars Johannesson received the M.Sc. degree in Automation and Mechatronics and the Ph.D. degree in automatic control from Chalmers University of Technology, Göteborg, Sweden in 2004 and 2009, respectively. He has been with the Electromobility group at Viktoria Swedish ICT since 2011, working with research on powertrain control within the Chalmers Energy Initiative. His main research interests are optimal control of hybrid electric vehicles, control of auxiliary systems, active cell balancing, and system studies. He is a member of the IEEE.



Bo Egardt (SM'90-F'03) received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from Lund Institute of Technology, Lund, Sweden in 1974 and 1979, respectively. During 1980, he was a Research Associate at the Information Systems Laboratory, Stanford, CA. From 1981 to 1989, he was with Asea Brown Boveri, where he was heavily involved in the introduction of adaptive control in the process industry. In 1989, he was appointed Professor of automatic control at Chalmers University of Technology, Göteborg, Sweden. His main areas of interest include adaptive and hybrid control and applications of control in the automotive area. Dr. Egardt has been an Associate Editor of IEEE Transactions on Control Systems Technology and of the *European Journal of Control*. He is a member of the editorial board for the *International Journal of Adaptive Control and Signal Processing*. He is a Fellow of the IEEE.