Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Approximate Dynamic Programming Applied to Parallel Hybrid Powertrains ⋆

Lars Johannesson * **Bo Egardt** **

* *Chalmers University of Technology, Göteborg, Sweden (Tel: 0046-31-7723724; e-mail: larsjo@chalmers.se).*
** *Chalmers University of Technology, Göteborg, Sweden (e-mail: egardt@chalmers.se)*

Abstract: The extra degree of freedom offered in hybrid electric vehicles have inspired many researchers to formulate and solve optimal control problems of various kinds. This paper presents an Approximate Dynamic Programming scheme that efficiently solves the optimal power split between the internal combustion engine and the electric machine in parallel hybrid powertrains. Gear switches and switches between hybrid and pure electric mode are formally treated. The scheme combines two ideas to reduce the computational time of the iterations performed in the dynamic programming. First, the value function is approximated using piecewise linear functions on a sparse grid. Secondly, by using model approximation the iterations performed in the dynamic programming are reduced to solving scalar quadratic problems. In the simulations the approximation scheme is able to find a good approximation of the optimal control trajectory.

Keywords: Dynamic Programming, Predictive Control, Hybrid Vehicles, Hybrid Powertrains, Powertrain Control

1. INTRODUCTION

The extra degree of freedom offered in a hybrid electric vehicle (HEV) — i.e. the power split between internal combustion engine (ICE) and electric machine (EM) — has inspired many researchers to formulate and solve optimal control problems of various kinds, see Sciarretta and Guzzella [2007] and references therein. The energy management strategy (EMS) strongly affects the performance of the HEV, e.g. its fuel efficiency. It is not trivial, however, to device an EMS which is efficient across a variety of operating conditions determined by traffic conditions, topography and driver characteristics. With the availability of traffic information from GPS, mobile phones and digital maps (GIS), predictions of the vehicle propulsion load can be made. This enables predictive control of the hybrid powertrain. Previous work within this field includes Sciarretta et al. [2004], Beck et al. [2006], and Deguchi et al. [2003].

Previous work by the authors include Johannesson et al. [2006], where optimal control of the power split of a mild parallel hybrid was investigated for known routes with a stochastic model of the speed profile along the route. One of the conclusions of this study was that if the route is known to the EMS, the optimal planning will be done on a long horizon with the topography playing a crucial role. The results indicated that with a navigation system that works in conjunction with a traffic information system, it should be possible to design a controller that achieves close to the minimal attainable fuel consumption.

Dynamic programming (DP), which was the tool used in the work mentioned, is a versatile tool, allowing to take into consideration e.g. non-linear models and state and control constraints; it can also be applied to both deterministic and stochastic models of the driving mission. However, a fundamental drawback of dynamic programming is the rapidly growing computational demands as the problem size increases.

In order to fight this *curse of dimensionality*, the results in Johannesson et al. [2006] formed the inspiration for a novel algorithm for predictive control on known routes; the basic idea, pursued in Johannesson and Egardt [2007], is to move as much as possible of the computations *outside* the real-time control loop. The algorithm was motivated by studying the value function for a large number of speed trajectories along a specific route. For a given position and SoC, it turned out that the slope of the value function varies only slightly between the studied speed trajectories. Hence, a valid approximation for the value function can be obtained by applying DP to a previously measured drive cycle, or even to a simulated/predicted one, in an *off-line* computation. When actually driving along the route, the previously stored value function approximation (as a function of position along the route) is used as terminal cost in a simple receding horizon controller. The proposed algorithm resulted in fuel consumption almost identical to the minimal attainable consumption on the studied route.

The objective of the present contribution is to develop this idea one step further. The motivation is that even though we have used the term *off-line* above, in practice we would like to recalculate the DP part periodically (although at a fairly low rate) in order to adjust to model uncertainties, disturbances and changes in traffic. Hence, it is of interest to minimize computations in the off-line step. This is particularly so when we would like to extend the number

⋆ This work was supported by the Swedish Energy Agency.

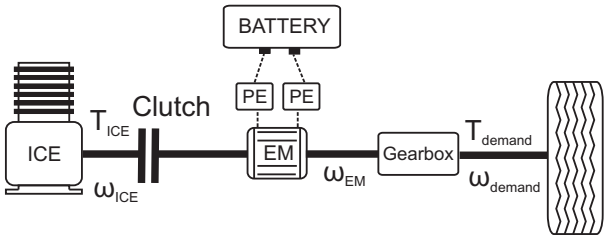

Fig. 1. Layout of the system architecture. The acronym PE is short for Power Electronics.

Table 1. Vehicle Specifications

Vehicle parameters	
Parameter	*Value*
Total vehicle Mass	1600kg
ICE type and max power	Atkinson, 43 kW
ICE efficiency	Prius 1 data
Maximal EM power	30 kW
Maximum EM torque	300 Nm
Battery type and max power	NiMH Battery, 32 kW
Battery capacity	7.5 Ah
Nominal voltage	270 V
Power electronics efficiency	0.95
Gearbox	5 stepped automatic

of states in order to handle gear switching dynamics and switching between hybrid and pure electric propulsion (combustion engine stop/start) – which is indeed the case here.

In order to arrive at the sought computational simplifications, two ideas will be pursued:

- Approximation of the value function using piecewise linear functions on a (possibly sparse) grid.
- Model approximation leading to simplification of the iterations performed in the dynamic programming.

The first idea is very much in line with the rationale behind the algorithm described above. The second idea has previously been used in several works including Beck et al. [2006]. Both approximations will be dealt with in detail later.

The paper is organized as follows. The vehicle and powertrain models used are described in the following section. The subsequent two sections describe the main contribution of the paper. The paper is ended with simulation results and conclusions.

2. VEHICLE MODEL

The system architecture for the studied parallel hybrid is shown in Fig. 1. The vehicle specifications are given in Table 1. The HEV is powered by an Internal Combustion Engine (ICE) with a modern Nickel Metal Hydride (NiMH) battery as a buffer. The mechanical power produced by the ICE is transferred to the driving wheels via a 5 stepped gearbox and a differential gear. The Electric Machine (EM) is located between the ICE flywheel and the gearbox and can add or absorb torque on the shaft. In order to enable efficient pure electric propulsion, the ICE can be disconnected from the rest of the powertrain by opening a clutch.

2.1 Quasi Static Vehicle Model

The objective of the chassis and powertrain model is to model the efficiency of the power flows in a parallel HEV. The model is a backwards (inverse) simulation model.

Chassis model The force needed to give the vehicle with mass m, the acceleration a, at a certain velocity v, and road grade θ, is modeled by the sum of the inertia force, the air drag force, the projected normal force and the rolling resistance force:

$$F = ma + 0.5\rho v^2 C_d A_{front} + mg\sin(\theta) + f_r mg\cos(\theta), \quad (1)$$

where ρ is the density of air, C_d the vehicle's air drag coefficient, A_{front} the vehicle's front area, g the gravitational acceleration and f_r the roll resistance coefficient.

Powertrain model The propulsion force results in a torque demand T_{dem}, defined here at the final drive. Note that T_{dem} is defined as negative when power can be regenerated. The shaft speed at the final drive is denoted ω_{dem}. The speed of the electric machine, ω_{EM}, is related to ω_{dem} through the gear ratio, r_g, as

$$\omega_{EM} = r_g \omega_{dem}, \quad (2)$$

where index g, denotes the gear number. The torque demand must be fulfilled by a sum of torque from the internal combustion engine, T_{ICE}, and torque from the electric machine, T_{EM}:

$$T_{ICE} - T_{EM} = \frac{\eta_g T_{dem}}{r_g}, \quad T_{dem} \leq 0 \quad (3)$$

$$T_{ICE} - T_{EM} = \frac{T_{dem}}{\eta_g r_g}, \quad T_{dem} > 0, \quad (4)$$

where η_g, denotes the mechanical efficiency at the selected gear. Note that T_{EM}, is defined as positive when charging the buffer. The ICE torque, T_{ICE} and the gear, g, are determined by the EMS. There are no *a priori* restrictions on gear changes and the inertia of the engine is not modeled.

Finally, the logical signal ICE_{on}, also decided by the EMS, determines if the ICE clutch is closed or opened (the latter being an option during pure electric propulsion or during braking):

$$\omega_{ICE} = \omega_{EM}, \quad ICE_{on} = 1, \quad (5)$$

$$\omega_{ICE} = 0, \quad ICE_{on} = 0. \quad (6)$$

The starting of the ICE is simplified in the way that we do not distinguish between if the ICE is started from a starter or if the clutch is closed.

The fuel mass rate, $c(\omega_{ICE}, T_{ICE})$, is given by linear interpolation in a quasi static, fuel mass rate map, see Fig. 2. When the clutch is open $T_{ICE} = 0$ and $c = 0$.

The electric machine losses are given by linear interpolation in a loss map and the electric converter efficiency is modeled as constant. The battery is modeled by a resistive circuit, shown in Fig. 3. The open circuit voltage $V_{oc}(SoC)$ and the charge/discharge resistances $R_c(SoC)$ and $R_d(SoC)$ depend on the State-of-Charge (SoC).

The time derivative of the SoC is determined by:

$$\frac{dSoC}{dt} = \frac{i}{Q_{max}}, \quad (7)$$

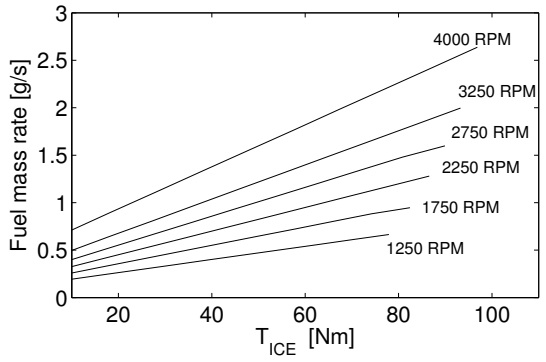

Fig. 2. The fuel mass rate map plotted at six different ICE speeds.

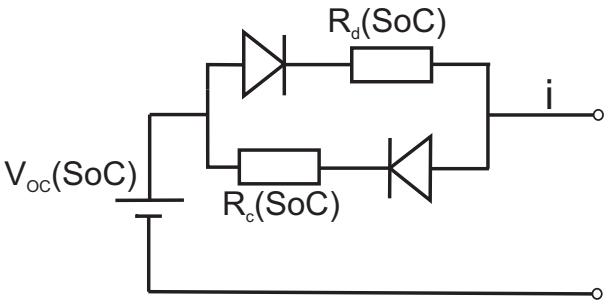

Fig. 3. Equivalent battery circuit.

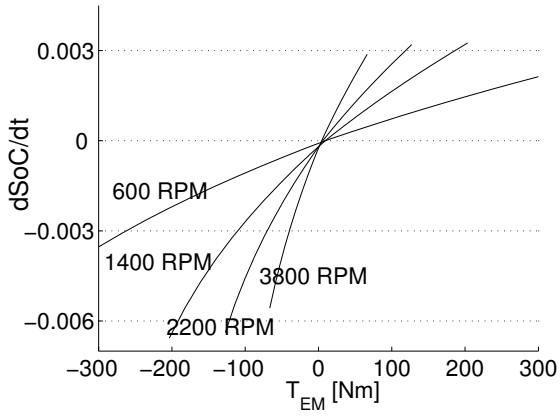

Fig. 4. Relationship between T_{EM} and $dSoC/dt$ at $SoC = 0.5$.

where Q_{max} is the total capacity. In Fig. 4 the relationship between T_{EM} and $dSoC/dt$ is shown with four EM speeds for $SoC = 0.5$.

A number of restrictions must be considered when simulating and optimizing the powertrain energy flow. The battery current and the SoC are limited according to

$$i \in [i_{EB,\min}, i_{EB,\max}], \quad (8)$$

$$SoC \in [0, 1]. \quad (9)$$

It should be noted that to avoid excessive wear of the battery, the SoC interval used in the optimization is only 15% of the total battery capacity and placed symmetric around 50% of the total buffer capacity. The ICE and EM torques, T_{ICE} and T_{EM}, are limited by rotational speed dependent constraints

$$T_{ICE}(\omega_{ICE}) \le T_{ICE,\max}(\omega_{ICE}), \quad (10)$$

$$T_{EM}(\omega_{ICE}) \in [T_{EM,\min}(\omega_{EM}), T_{EM,\max}(\omega_{EM})]. \quad (11)$$

3. DYNAMIC PROGRAMMING

From a given velocity trajectory and topographic profile, the torque demand trajectory can be calculated using the chassis model. The trajectories are discretized with an appropriate sampling time (1 second is used here). The shaft speed, $\omega_{dem,k}$, and the torque demand, $T_{dem,k}$, are then given for all time steps, k, in the interval $1 \le k \le N$. With the shaft speed and torque demand trajectories determined, the dynamic states in the vehicle model are the continuous state, SoC, and the two discrete states, ICE_{on} and the gear number, g. The control signal is the triple, $\mathbf{u}_k = (T_{ICE,k}, g_{k+1}, ICE_{on,k+1})$. With a slight abuse of notation, the next discrete state is directly determined by the control signal.

Based on a quantization of the continuous state SoC, a Dynamic Programming problem is formulated. In the well known Dynamic Programming algorithm the optimal control signal is found by backwards iterations of a collection of value functions from the final time sample N to the first:

$$J_k^i(SoC_k) = \min_{T_{ICE},j} \{c_{i,k}(T_{ICE}) + d_{ij} + J_{k+1}^j(SoC_{k+1})\}. \quad (12)$$

For the present formulation, we have a collection of value functions $J_k^i(SoC_k), i = 1, \ldots, n$, where n is the number of admissible discrete states (gears, engine on/off) at the sample k. Analogously, the value functions at the next sample, $k+1$, are denoted $J_{k+1}^j(SoC_{k+1}), j = 1, \ldots, m$, where m is the number of admissible discrete states (gears, engine stop/start). The current value functions, $J_k^i(SoC_k)$, are calculated over a grid in SoC. In (12), linear interpolation is used to calculate $J_{k+1}^j(SoC_{k+1})$.

Gear switches and transitions from engine off to on, are penalized by the instantaneous cost term d_{ij}. The cost, d_{ij}, is a fuel equivalent for changing gear and turning the ICE on. The fuel equivalent used here is a simplification; generally the cost would be dependent on the vehicle speed and torque demand. It would then include the fuel consumption and the fuel equivalent for the change of SoC that will result when changing the engine speed during an ICE start or gear change. Moreover, it would also include the fuel equivalent for the energy loss that occurs when changing the gear during brake regeneration and possibly also a comfort penalty for too many shifts.

The other instantaneous cost term, $c_{i,k}(T_{ICE})$, or perhaps clearer $c(\omega_{ICE}, T_{ICE})$, is the fuel consumption. The value function $J_k^i(SoC_k)$ is the sum of instantaneous costs that remain before reaching the final time sample from the current SoC. The next state of charge SoC_{k+1} is a function of $\omega_{dem,k}$, $T_{dem,k}$, SoC_k and $\mathbf{u}_k$.

The control signal $\mathbf{u}$ must not violate the constraints (3)-(6) and (8)-(11).

The value functions J_N^i, at the final sample are initiated with a penalty function. The detail of the grid in SoC, which J_k^i are calculated over determines the accuracy of the solution.

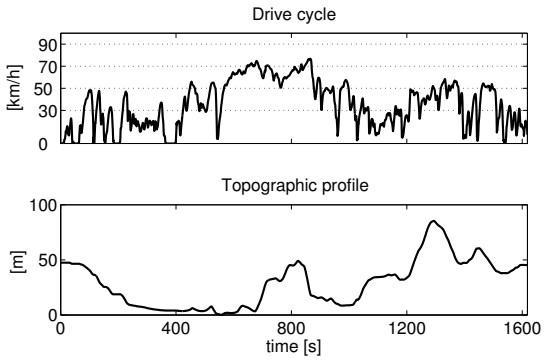

Fig. 5. The drive cycle and the topographic profile that is used in the simulations

3.1 Simulation

After that the DP-iterations (12) have terminated, the optimal control and state trajectory are simulated from the initial state, using the value functions J_k^i. The optimal control signal trajectory is thus given by

$$\mathbf{u}_k^* = \arg\min_{T_{ICE},j} \{c(\omega_{ICE,k}, T_{ICE}) + d_{ij} + J_{k+1}^j(SoC_{k+1})\}, \quad (13)$$

where linear interpolation is used to calculate $J_{k+1}^j(SoC_{k+1})$.

4. APPROXIMATE DYNAMIC PROGRAMMING

As mentioned already in the Introduction, the main contribution here is to improve the computational efficiency of the DP outlined in the previous section. We will deal with the two main ideas mentioned in the Introduction separately below. The drive cycle and the topographic profile that will be used in the simulations are shown in Fig. 5

4.1 Local linear approximation of the value function

We will now study the numerical sensitivity of the DP-solution with respect to the detail of the grid in SoC. The study will be done by comparing two simulations, one based on a sparse grid and the other on a detailed grid.

Due to lack of measurement data from a real powertrain, the transition penalties d_{ij}, are here simply manually tuned to produce realistic simulation results, realistic in the sense that there are not too many gear changes and ICE turn-on's. The cost for turning on the ICE is tuned with the guideline that the ICE should not be turned on for shorter periods than 10 seconds. A simplified torque reserve constraint is imposed on the ICE, requiring the ICE to be turned on when the velocity exceeds 40 km/h.

Simulated SoC-trajectories are shown in Fig. 6 for two SoC-grids, one with 40 grid points and the other with 2000 grid points. The two grids result in almost identical fuel consumption, 4.12 $l/100km$.

In the dynamic programming iteration (12), it is obvious that, for a specific dynamic state, the local properties of the value function will determine the optimal control signal. An example of a value function is shown in Fig. 7.

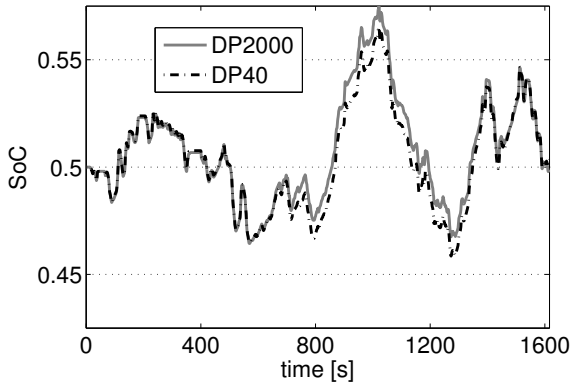

Fig. 6. Simulated SoC-trajectories based on DP-calculation with 2000 respectively 40 equally spaced grid points

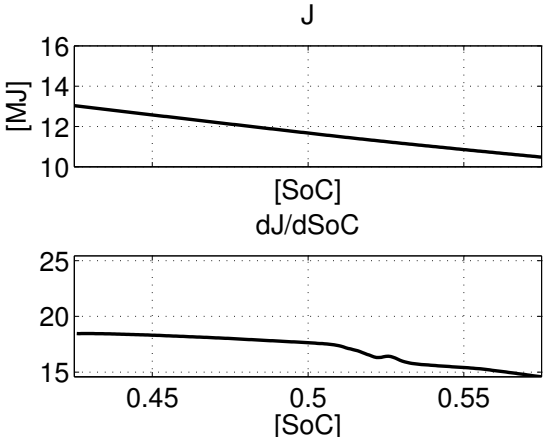

Fig. 7. The value function at 800 s, corresponding to gear 5 and ICE on.

The shown value function is locally well approximated by a linear function of SoC. This is true for all examined value functions, except when getting close to the hard constraints on SoC. What determines the required degree of accuracy of the approximation is the effect on the computed control signal, obtained by linear interpolation. When using equally spaced grid with 40 or less grid points, due to the constraints (8), (11) the linear interpolation in (12) is actually almost always an interpolation on a function with only two linear pieces, one for charging and one for discharging. Applying this piecewise linear approximation to the j^{th} value function in a neighborhood of $SoC = SoC_0$ gives

$$J_{k+1}^j(SoC_0 + \Delta SoC) \approx J_{k+1}^j(SoC_0) + \lambda_s(j, SoC_0)\Delta SoC. \quad (14)$$

The index s is used to denote that the slope λ is dependent on the sign of ΔSoC. The similarity between (14) and the ECMS, see Sciarretta and Guzzella [2007], is due to the relation between the partial derivative of the value function in dynamic programming and the costate from Pontryagin Maximum Principle see Naidu [2003].

4.2 Quadratic programming in the DP iteration

So far, the powertrain model has been treated as a general, nonlinear, quasi-static model underlying the optimal

control problem. We will now exploit the particular form of the nonlinearities, and show how this can be used to our advantage in the computations.

The first observation is that, for a given engine speed, ω_{ICE}, (or, equivalently, a given vehicle speed and gear) the fuel consumption is close to linear in the engine torque, i.e., with a slight abuse of notation,

$$c(\omega_{ICE}, T_{ICE}) \approx c_0(\omega_{ICE}) + c_1(\omega_{ICE})T_{ICE} \qquad (15)$$

The second observation is that the electric losses are well described by a piecewise quadratic function of the electric machine torque T_{EM}, so that the change of SoC over a sampling interval Δt can be written as

$$SoC_{k+1} \approx SoC_k + \Delta t \cdot [b_{s0}(\omega_{EM}, SoC) + b_{s1}(\omega_{EM}, SoC)T_{EM} + b_{s2}(\omega_{EM}, SoC)T_{EM}^2]. \qquad (16)$$

The index s is used to denote that the parameters are dependent on the sign of ΔSoC.

The DP iteration of the i^{th} value function, at time index k, according to (12) can be rewritten in two steps by using the approximations (14), (15) and (16). The first step calculates for each fixed transition $i \to j$ (and with $\omega_{ICE}, \omega_{EM}$ determined by j)

$$J_k^{ij}(SoC_k) = \min_{T_{ICE}} \{c_0(\omega_{ICE}) + c_1(\omega_{ICE})T_{ICE} + J_{k+1}^j(SoC_k) + \lambda_s(j, SoC_k)\Delta t[b_{s0}(\omega_{EM}, SoC_k) + b_{s1}(\omega_{EM}, SoC_k)T_{EM} + b_{s2}(\omega_{EM}, SoC_k)T_{EM}^2]\}. \qquad (17)$$

From (3),(4) we can conclude that the r.h.s. of (17) is piecewise quadratic in T_{ICE} and thus the minimization can be carried out explicitly by solving two scalar quadratic problems. For the discrete state corresponding to $ICE_{on} = 0$, i.e. the engine is switched off, there is no optimization involved, since T_{EM} is then directly given by the torque demand.

It remains to take care of the discrete decision variable in the DP iteration. Once the optimization (17) has been carried out for every admissible j, the choice of discrete control is simply determined by

$$J_k^i(SoC_k) = \min_j \{J_k^{ij}(SoC_k) + d_{ij}\}. \qquad (18)$$

5. RESULTS

The suggested Approximate Dynamic Programming (ADP) scheme will now be evaluated by comparing with the DP-solution based on 2000 equally spaced grid points. Note that ADP scheme is here only used to calculate the value functions. In the simulations and when calculating the optimal control signal (13), it is the nonlinear, quasi-static model that is used for both the ADP-solution and the nonlinear DP-solution.

The value functions from the ADP scheme are calculated using 20 equally spaced grid points. The simulated SoC-trajectories are shown in Fig. 8. The corresponding control trajectories are shown in Fig. 9 and Fig. 10. From the figures we see that ADP-scheme is able to produce a close approximation to the optimal state and control trajectory. At most time samples the gear choice and the logical signal ICE_{on} are identical to the DP-solution. In Fig. 10 the ICE torque is shown for the part of the drive cycle with the largest differences between the ADP and the DP-solution. Nevertheless, the trajectories are very similar. By studying the derivative of the value functions with respect to SoC along the drive cycle an estimate of the fuel equivalent of the difference in SoC is provided. Using this estimate, the difference in fuel consumption between the DP and the ADP-solution is assessed to 0.8%.

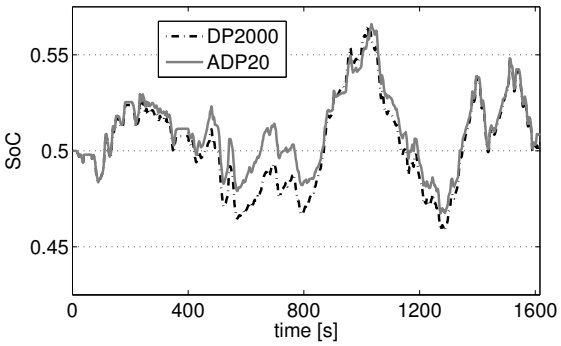

Fig. 8. Simulated SoC-trajectories based on a DP-calculation with 2000 and an ADP calculation with 20 equally spaced grid points.

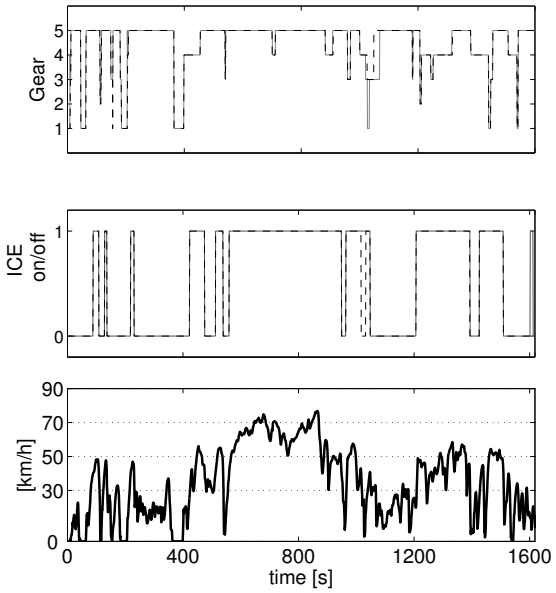

Fig. 9. Simulated gear and ICE-state trajectories based on a DP-calculation with 2000 and an ADP calculation with 20 equally spaced grid points. The black dashed curves correspond to the accurate DP-solution.

6. CONCLUSIONS

The presented ADP scheme gives the possibility to efficiently calculate the value functions. By introducing heuristics many of the discrete transitions could be ruled out immediately. It is plausible that further improvements can be made by optimizing the spacing of the grid or by using some smarter functional approximation of the value functions.

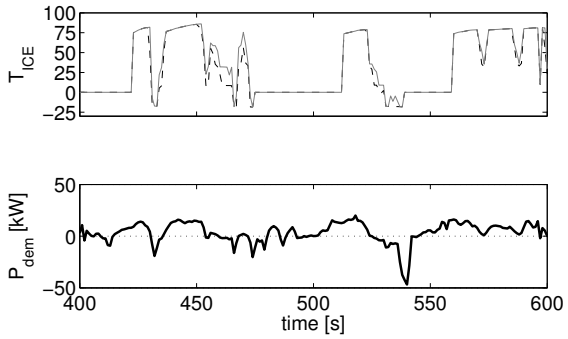

Fig. 10. Simulated ICE torque trajectories based on a DP-calculation with 2000 and an ADP calculation with 20 equally spaced grid points. The black dashed curves correspond to the accurate DP-solution.

REFERENCES

Ralf Beck, Alexander Bollig, and Dirk Abel. Comparison of two real-time predictive strategies for the optimal energy management of a hybrid electric vehicle. In *ECOSM Rencontres Scientifiques de l'IFP*, Paris, France, 2006.

Yoshitaka Deguchi, Kouichi Kuroda, Makoto Shouji, and Taketoshi Kawabe. HEV charge/discharge control system based on car navigation information. In *Proc. JSAE Spring Conference*, Yokohama, Japan, 2003.

Lars Johannesson, Mattias Åsbogård, and Bo Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *Transactions on Intelligent Transportation Systems.Special Issue: ITSC 05*, 2006.

Lars Johannesson and Bo Egardt. A novel algorithm for predictive control of parallel hybrid powertrains based on dynamic programming. In *Proc. Fifth IFAC Symposium on Advances in Automotive Control*, Monterey, USA, 2007.

Desineni Naidu. *Optimal Control Systems*. Electrical Engineering textbook series. CRC Press, 2003.

Antonio Sciarretta and Lino Guzzella. Control of hybrid electric vehicles. *IEEE Control Systems Magazine*, 27 (2):60–70, 2007.

Antonio Sciarretta, Lino Guzzella, and Michael Back. Real-time optimal control strategy for parallel hybrid vehicles with on-board estimation of the control parameters. In *Proc. of the IFAC Symposium on Advances in Automotive Control*, Salerno, Italy, 2004.