An efficient optimum energy management strategy using Parallel dynamic programming for a hybrid train powered by Fuel-Cells and batteries

Hujun Peng
Institute of Electrical Machines (IEM)
RWTH Aachen University
Aachen, Germany
hujun.peng@iem.rwth-aachen.de

Jianxiang Li
Institute of Electrical Machines (IEM)
RWTH Aachen University
Aachen, Germany
jianxiang.li@rwth-aachen.de

Kai Deng
Institute of Electrical Machines (IEM)
RWTH Aachen University
Aachen, Germany
kai.deng@iem.rwth-aachen.de

Andreas Thul
Institute of Electrical Machines (IEM)
RWTH Aachen University
Aachen, Germany
andreas.thul@iem.rwth-aachen.de

Weihan Li
Institute for Power Electronics and Electrical Drives
(ISEA ESS) RWTH Aachen University
Aachen, Germany
weihan.li@isea.rwth-aachen.de

Lars Löwenstein
Siemens Mobility GmbH
Vienna, Austria
lars.loewenstein@siemens.com

Dirk Uwe Sauer
Institute for Power Electronics and Electrical Drives
(ISEA ESS) RWTH Aachen University
Aachen, Germany

Kay Hameyer
Institute of Electrical Machines (IEM)
RWTH Aachen University
Aachen, Germany

***Abstract*—A parallel Dynamic Programming algorithm, basing on the matrix calculation, is used to develop the optimum energy management strategy for a fuel cell and lithium-ion battery hybrid train. In this paper, besides the state of charge of the battery, the power from the fuel cell is defined as the other state variable. Then, the control variable is the power change rate in the fuel cell system. With the help of this problem formulation, an efficient parallel Dynamic Programming is easy to implement. The parallel calculation requires only one loop over the time stages. To make the parallel Dynamic Programming basing on the matriculated calculation successful, a semi-physical soft constraints mechanism is developed to initialize the cost function at the end time stage properly. With this parallel Dynamic Programming, the effect of a weighting factor between maximizing the fuel economy and avoiding high dynamic power change of fuel cell, on the total hydrogen consumption is investigated time efficiently.**

***Index Terms*—fuel cell hybrid train, optimum energy management, parallel dynamic programming, soft constraints, fuel economy, weighting factor**

I. Introduction

The electrification of railway vehicles is an effective way to save energy and reduce emissions. However, the cost of complete electrification of the railway network, particularly for lines with low traffic, is not cost-effective. For these railway sections, the fuel cell hybrid train is an efficient alternative. In a fuel cell hybrid train, a lithium-ion battery system is used in addition to fuel cells, to provide and absorb high transient power during acceleration and recuperative braking operation. Moreover, the hybrid train operates in charge-sustaining mode. Therefore, the fuel cell system meets the average power requirement, and the battery system fulfills the transient peak power demand. Fig. 1 shows the whole system configuration of the hybrid train driveline. As a boundary condition, the final state of charge (*soc*) of the battery is equal to the initial value.

The power distribution between the two energy storage systems, the energy management strategy (EMS), provides a degree of freedom to optimize the performance of the hybrid train, including the driveability, the hydrogen consumption and the fuel cell lifetime. Developing an energy management strategy under many static and dynamic constraints, defined globally or locally, is a challenging task. In the literature, there are three methods of developing energy management strategies: rule-based, real-time optimization, and global optimization methods. The rule-based method is based on the engineer's experience and implemented using conditional rules. Because of the low computational effort, the rule-based strategy is real-time capable. However, it does not guarantee optimal performance [1]. The real-time optimization method is also real-time capable because the power distribution is determined basing on a predefined transient cost function. The most famous of them is the equivalent consumption minimization strategy (ECMS), which treats the battery power in a sense as fuel consumption in the future using an equivalent factor [2]. This method does not guarantee an optimal power distribution as well [3]. However, the optimal distribution is accessible if the drive cycle of the vehicle is known in advance, which is a reasonable assumption for railway transport. For this purpose, the global optimization method is used. The most famous one is based on Dynamic Programming (DP). The DP is based on Bellman's principle of optimality [4] and implemented off-line. In [5], parallel dynamic programming basing on the matrix calculation was used to optimally distribute the power among three storage systems: a fuel cell system, a battery pack, and an ultracapacitor. The power from the fuel cell system and the capacitor are chosen as the state variables, with dynamic power constraints in the fuel cell system ignored. In most of the literature about energy management strategy for a hybrid vehicle with dual storage system, DP is implemented using multiple embedded loops [6], [7], including the loop over time-stages, state-space, and control space. Therefore, the exemplary implementation of Dynamic Programming is very time-consuming. As a result, to reduce the computation time, the SoC is defined as the single state variable. Then, the control variable is the power from one storage system, and power from the other storage system is passively determined according to the energy conservation [8]. However, under this one-dimensional framework, it is complicated to implement parallel dynamic programming conveniently under consideration of dynamic constraints on control variables. For example, the change rate of fuel cell power cannot exceed some certain limits [9], which is related to the fuel cell lifetime [10]. Therefore, in works of literature on EMS using DP, the dynamic constraints on the components are not considered.

In this work, to implement parallel dynamic programming accessible, the power from the fuel cell is defined as the other state variable besides the SoC, instead of as the control variable. Then, the control variable is the rate of change in fuel cell power. This transformed formulation by increasing the dimension, that seems to make the problem more complicated to solve, can make parallel Dynamic Programming easy to implement, because the dynamic constraints on the control variable (fuel cell power) under the one-dimensional framework are replaced with static constraints on control variables (power change rate) under the two-dimensional framework.

The paper is organized as follows: first, the basic principle of DP and its application in EMS is introduced; second, the parallelization of DP by increasing the dimension is detailedly described; subsequently, the results of optimum energy management strategy using the parallel dynamic programming are displayed and based on that follows an investigation of the effect of a penalty factor. Last, the conclusion and the possible outlook are given.

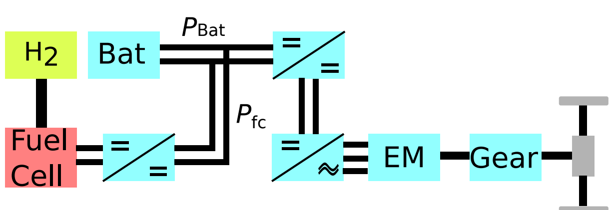


Fig. 1: System configuration of hybrid train driveline.

II. One-dimensional DP based energy management strategy

First, the general formulation of the optimal control problem is introduced. Then, "Bellman's principle of optimality" is described. Finally, this formulation is adapted to the case of a fuel cell hybrid train. Notably, the static and dynamic constraints related to the components of the hybrid system are discussed.

A. General formulation of optimum control

The profile of the optimal control problem considered are as follows: fixed time interval, multi-state multi-control, fixed initial and desired final state, fixed definition interval of control and state variables, a predetermined disturbance signal [11]. These characteristics are summarized as follows:

$$\text{cost functional} \quad J = h\left(\boldsymbol{x}\left(t_{\mathrm{f}}\right)\right) + \int_{t_0}^{t_{\mathrm{f}}} g\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) dt \tag{1}$$

$$\text{optimal control} \quad \boldsymbol{u}^*(t) = \arg\min_{\boldsymbol{u}(t)} J(\boldsymbol{x}_0, \boldsymbol{u}(t)), \tag{2}$$

$$\text{system dynamics} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right), \tag{3}$$

$$\text{boundary conditions} \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \quad \boldsymbol{x}(t_{\mathrm{f}}) = \boldsymbol{x}_{\mathrm{f}}, \tag{4}$$

$$\text{admissible controls} \quad \boldsymbol{u}(t) \in \mathscr{U}(t) \subset \mathbb{R}^n, \tag{5}$$

$$\text{admissible states} \quad \boldsymbol{x}(t) \in \mathscr{X}(t) \subset \mathbb{R}^m, \tag{6}$$

where $h\left(\boldsymbol{x}\left(t_{\mathrm{f}}\right)\right)$ is a penalty function in terms of the final states; t_0 and t_{f} the start time and the end time respectively; $g\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right)$ the transition cost in the time-variant function of states and control variables; n and m are the numbers of control inputs and state variables, respectively. Moreover, the admissible control set in (5) is limited by both time-variant static and dynamic constraints.

B. Principle of Dynamic Programming

Since DP is applied to solve a continuous-time problem, the continuous model in (3) must be discretized in time. Then the time-discrete model can be rewritten as:

$$\boldsymbol{x}\left[k+1\right] = \boldsymbol{F}\left(\boldsymbol{x}\left[k\right], \boldsymbol{u}\left[k\right], k\right), \quad k = 0, 1, 2, \ldots, N-1 \tag{7}$$

where N is the number of discrete time stages and k the index of the k-th time instant. Let $\pi = \{\boldsymbol{u}\left[0\right], \boldsymbol{u}\left[1\right], \ldots, \boldsymbol{u}\left[N-1\right]\}$ be a control policy. The discretized cost functional using the policy π with an initial state x_0 is calculated as follows [11]:

$$J_\pi\left(x_0\right) = h_N(\boldsymbol{x}\left[N\right]) + \sum_{k=0}^{N-1} g(\boldsymbol{x}\left[k\right], \boldsymbol{u}\left[k\right], k)\Delta t, \tag{8}$$

where $h_N(\boldsymbol{x}\left[N\right])$ and $g(\boldsymbol{x}\left[k\right], \boldsymbol{u}\left[k\right], k)\Delta t$ are the final cost and transition cost, corresponding to $h\left(\boldsymbol{x}\left(t_{\mathrm{f}}\right)\right)$ and $g\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right)$ in equation (1) respectively. The optimal control policy π^* is the policy minimizing the cost J in terms of π under a given initial state x_0:

$$J^*(x_0) = \min_{\pi \in \mathit{\Pi}} J_\pi(x_0), \tag{9}$$

where $\mathit{\Pi}$ is the set of all feasible policies.

Based on Bellman's principle of optimality [4], the DP algorithm calculates the optimal cost-to-go function $J^*(\boldsymbol{x}[k])$ at each node $\boldsymbol{x}[k]$ in the discretized state-time space by backward calculation:

- cost for end state:

$$J^*(\boldsymbol{x}[N]) = h_N(\boldsymbol{x}[N]), \tag{10}$$

- iterative calculation for $k = N-1$ to 0:

$$J^*(\boldsymbol{x}[k]) = \min_{\boldsymbol{u}[k]\in\mathscr{U}[k]} (g(\boldsymbol{x}[k], \boldsymbol{u}[k], k)\Delta t + J^*(\boldsymbol{x}[k+1])), \tag{11}$$

where $\boldsymbol{x}[k+1]$ is calculated using (7). After backward recursion from $k = N-1$ to $k = 0$, the optimal policy is determined.

The exemplary implementation of DP uses embedded loops over time-stages, state variables, and control inputs. The computational complexity corresponding to that is in the order of

$$\mathcal{O}(N \cdot p^n \cdot q^m), \tag{12}$$

where N is the number of time steps; p and q are the numbers of discretization for state and control input; n and m is the number of states and control inputs, respectively [7].

C. Formulation of one-dimensional DP to solve EMS

To keep the computational time affordable, in most of the literature using DP to implement the optimal power distribution between two storage systems, the state of charge (SoC) of the battery is defined as the single state variable. Then the fuel cell power is the control input, and the corresponding hydrogen consumption is the cost. Generally, the cost functional in EMS problem for a fuel cell hybrid vehicle is defined like following [9]:

$$\begin{aligned} J = & \underbrace{w_{\text{bt}}\left(x(t_{\text{f}}) - x(t_0)\right)^2}_{\text{charge sustaining Penalty}} + \underbrace{\Delta t \sum_{k=0}^{N-1} \dot{m}_{\text{H}_2}\left(P_{\text{fc}}[k]\right)}_{\text{fuel cell consumption}} + \\ & \underbrace{\lambda\left(\sum_{k=0}^{N-1} \left|\text{Sign}(P_{\text{fc}}[k]) - \text{Sign}(P_{\text{fc}}[k+1])\right|\right)}_{\text{fuel cell on/off penalty}}, \end{aligned} \tag{13}$$

where the weighting factor w_{bt} enforces the SoC to return to the expected final value and the λ provides a healthy tunnel for the fuel cell to avoid frequent on/off operations. Moreover, the constraints considered are summarized as follows:

- model dynamics:

$$\dot{x} = -\frac{1}{Q_{\text{batt}}}\left(\frac{V_{\text{ocv}}}{2R_0} - \sqrt{\left(\frac{V_{\text{ocv}}}{2R_0}\right)^2 - \frac{P_{\text{batt}}}{R_0}}\right), \tag{14}$$

- energy conservation:

$$P_{\text{BT}}[k] = P_{\text{load}}[k] - P_{\text{fc}}[k], \tag{15}$$

- fuel cell power static limits:

$$P_{\text{fcmin}}(t) \leqq P_{\text{fc}}[k] \leqq P_{\text{fcmax}}(t), \tag{16}$$

- fuel cell power dynamic limits:

$$R_{\text{down}-\text{fc}}\Delta t \leqq P_{\text{fc}}[k] - P_{\text{fc}}[k-1] \leqq R_{\text{up}-\text{fc}}\Delta t, \tag{17}$$

- battery power limits:

$$P_{\text{BTmin}} \leqq P_{\text{BT}}[k] \leqq P_{\text{BTmax}}, \tag{18}$$

- battery current limits:

$$I_{\text{BTmin}} \leqq I_{\text{BT}}[k] \leqq I_{\text{BTmax}}, \tag{19}$$

- battery SoC limits:

$$soc_{\text{min}} \leqq x[k] \leqq soc_{\text{max}}. \tag{20}$$

Among them, $P_{\text{load}}[k]$ is derived from the predetermined drive cycle; according to (15), the battery power is calculated passively, depending on the control input $P_{\text{fc}}[k]$. To protect the battery, the SoC should be between the low and high boundary as (20). (16)-(19) are related to constraints on the control inputs. Notably, (17) is a dynamic constraint, and the others are static constraints.

In the implementation of one-dimensional DP, as can be found in most of the literature about fuel cell vehicle, the dynamic constraint on fuel cells (17) is not considered. As a result, the implementation is straightforward. However, it is necessary to keep the power change rate of fuel cells under certain limits to make fuel cells operates healthily [9]. With that considered, the DP is implemented using embedded loops over state and control space, besides the loop over the time-stages. The computational time is then $\mathcal{O}(N \cdot p \cdot q)$ according to (12) with $n = m = 1$. Correspondingly, this simulation takes enormous time, which is in the praxis not desired. Another drawback to using one-dimensional DP to solve EMS, with die dynamic constraints on the control variable considered, is that the precondition of no aftereffect for using DP is not fulfilled since the actual control input determines the range of control variable in the next time interval according to the equation (17).

III. Parallel Dynamic Programming

In this part, firstly, the underlying thoughts behind implementing parallel dynamic programming under MATLAB is introduced. Then, a semi-physical mechanism for designing soft constraints is discussed, which is crucial for the success of parallel dynamic programming. Finally, the two dimensional DP to solve EMS using the parallel dynamic programming is displayed.

A. Parallel Dynamic Programming

MATLAB software has optimal routines for matrix-based algorithms, which significantly improve the performance by eliminating iterative loops. Unfortunately, there are not integrated works about this field, only [5], [12] noted to implement the DP basing on matrix calculation. However, both do not consider the dynamic constraints on system components. Because of the matrix calculation, the loops over the state and control space can be eliminated. As a result, the computational performance is significantly improved, but need higher memory storage. In fact, during matrix calculation, all operations on elements performed in one step. Correspondingly, the computational time is in the order of $\mathcal{O}(N)$. Fig. 2 shows the

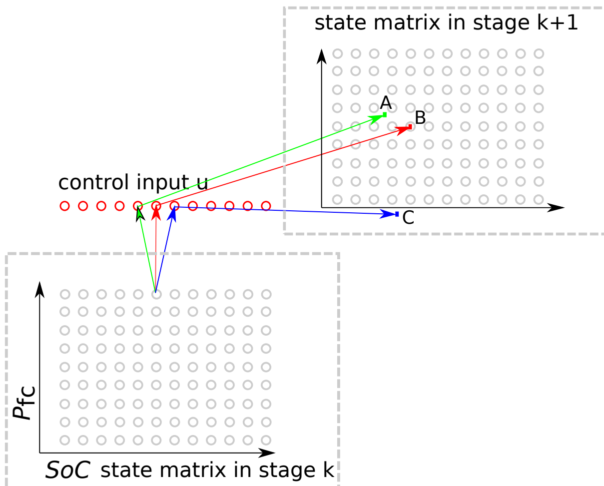


Fig. 2: The parallel dynamic programming transition pattern in each time stage.

transition from the discretized states in the time stage k to state space in the time stage $k+1$ under different discretized values of the control input. After the transition, there are three different situations:

1) The new state locates between discretized states as the Point A shows.
2) The new state locates just in discretized states as the Point B shows.
3) The new state locates outside of the feasible discretized states as the Point C shows.

For situation A, the cost-to-go $J^*(\boldsymbol{x}[k+1])$ of the resulted state is interpolated. For situation C, the cost-to-go $J^*(\boldsymbol{x}[k+1])$ is allocated a vast number. According to (11), the cost-to-go $J^*(\boldsymbol{x}[k])$ can be determined. If the new state cannot end in a feasible state under all of the discretized control values, the corresponding cost-to-go value $J^*(\boldsymbol{x}[k])$ is equal to the same huge number, to exclude those control policy, that leads the state trajectory to pass through this node in state-time space.

B. Soft Constraints

As seen in (10), before the iterative loop over the time stages for $k = N-1$ to $k = 0$, the cost-to-go function should be initialized. Because of the transition, as shown in Fig. 2, is not the type of point-to-point, the soft constraints must be used. Moreover, the design of soft constraints is crucial for the success of DP. If the penalty is small for the states violating the boundary condition as in (4), the soc trajectory does not result in the target soc. However, if the penalty is too substantial, the cost-to-go function loses the physical meaning, because the cost-to-go is defined as the sum of the transition cost and the cost-to-go of the new state, which start from the final time stage.

In this study, the soft constraint is designed by using a semi-physical approach. The state-space in the final time stage is divided into two parts. The one covers a narrow range of soc in the neighboring of the target soc (part ”narrow”), and the rest is the other one (part ”rest”). The initial cost is determined to base on the fuel cell specific consumption curve and the position, in which the end states locate. The end states with soc, that is larger than the target value of soc and locates in the part ”narrow”, are allocated the negative value of the hydrogen consumption, required to charge the battery from the target soc to the current soc using the maximum efficiency of the fuel cell. For the states outside of the part ”narrow”, the hydrogen consumption amount calculated using the method before is multiplied with a gradually decreasing penalty factor smaller than one, as the state is further away over the target value. On the contrary, the end states with soc, that is smaller than the desired value of soc and locates in the part ”narrow”, is allocated the value of the hydrogen consumption, required to charge the battery from the current soc to the target soc using low efficiency of the fuel cell. For the state outside of the part ”narrow”, the hydrogen consumption amount calculated is multiplied with a gradually increasing penalty factor higher than one, as the final state is further away under the target value. In this way, the target state inside the part ”narrow” is favored, and the soc trajectory ends in this part ”narrow” indeed, if it is possible under given drive cycle. Moreover, if the load demand is too high or small, so that the target value to reach is impossible, this initialization mechanisms can make the final state as close as possible to the target. As additional benefits, this soft constraints can determine the reachable final states under a given load demand profile and a given initial soc. Then the target soc can be adjusted if necessary.

Summarily, the try-and-error tunning is avoided, and the soc is enforced to end in the neighboring of the target value if possible under given load profile and initial conditions.

C. Two-dimensional DP to solve EMS

In solving EMS for fuel cell hybrid vehicle, using two-dimensional DP, the fuel cell power is chosen to be the other state variable, besides the soc. Then, the power change rate of fuel cells is defined as the control variable. The EMS problem is reformulated as follows:

- state vector: $\boldsymbol{x} = [x_1, x_2] = [soc, P_{\text{fc}}]$,
- control input: $u = \dfrac{dP_{\text{fc}}}{dt}$,
- system dynamics:

$$\begin{aligned} \dot{x_1} &= -\frac{1}{Q_{\text{batt}}}\left(\frac{V_{\text{ocv}}}{2R_0} - \sqrt{\left(\frac{V_{\text{ocv}}}{2R_0}\right)^2 - \frac{P_{\text{load}} - x_2}{R_0}}\right), \\ \dot{x_2} &= u, \end{aligned} \tag{21}$$

- state constraints:

$$\begin{aligned} soc_{\min} &\leqq x_1 \leqq soc_{\max}, \\ P_{\text{fcmin}} &\leqq x_2 \leqq P_{\text{fcmax}}, \end{aligned} \tag{22}$$

- control constraint:

$$R_{\text{down}-\text{fc}} \leqq u \leqq R_{\text{up}-\text{fc}}. \tag{23}$$

The constraints on battery power and current remain the same as before.

The cost functional from (13) can be rewritten and expanded like the following:

$$J = \underbrace{w_{\text{bt}} \left(x_1(t_{\text{f}}) - x_1(t_0)\right)^2}_{\text{charge sustaining Penalty}} + \underbrace{\Delta t \sum_{k=0}^{N-1} \dot{m}_{\text{H}_2} \left(x_2[k]\right)}_{\text{fuel cell consumption}} + \underbrace{\lambda \left(\sum_{k=0}^{N-1} |\text{Sign}(x_2[k]) - \text{Sign}(x_2[k+1])| \right)}_{\text{fuel cell on/off penalty}} + \underbrace{\Delta t \sum_{k=0}^{N-1} \sigma |u[k]|}_{\text{fuel cell power change penalty}} , \tag{24}$$

where the introduced factor σ in the last term penalizes high dynamic power change from the fuel cell system, to keep the fuel cell operating with harmony between fuel economy and fuel cell lifetime. For $\sigma = 0$, the cost function returns to the form defined in (13).

With the dimension increased, the dynamic constraint on the control input under the one-dimensional framework is replaced by a static constraint on the new control input under the two-dimensional framework, that makes parallel dynamic programming convenient to implement. Besides that, the aftereffect under the one-dimensional framework is eliminated, because the control variable under the two-dimensional framework in the stage before does not influence the range of control input in the next time interval.

IV. Results

Firstly, the optimum energy management solution corresponding to minimum hydrogen consumption is displayed. Then, the effect of the weighting factor σ in (24) on the balance between fuel economy and fuel cell working condition is investigated and based on that follows a discussion.

A. Time Analysis of the optimum power distribution

The setup parameters of two dimensional DP are summarized in Tab. I. The penalty factor λ is allocated an enormous number to enforce that the fuel cell is not cut off during the whole drive cycle. Moreover, at first, the penalty factor σ in (24) is set to be zero. The drive cycle is shown in Fig. 4a and the whole range is $198.8\,\text{km}$. From the drive cycle, the power demand is calculated. Alongside with that, the time history of relevant variables is displayed in Fig. 4. The power plots in Fig. 4b correspond to optimal power distribution history. The battery power and the fuel cell power are limited in the range of ($-1000\,\text{kW}$ – $1000\,\text{kW}$) and ($0\,\text{kW}$ – $200\,\text{kW}$) respectively. The soc plot in Fig. 4c shows that the end value of soc reaches the initial value of 0.8 and the charge sustenance is satisfied. From the control input plot in Fig. 4d, a highly frequent change of fuel cell operating point can be observed, and the maximal permitted change rate is $\pm 20\,\text{kW/s}$. Because the penalty factor σ here is zero, the substantial dynamic change in the fuel cell power is allowed, as long as the minimization of hydrogen consumption benefits from that. The total hydrogen consumption is $24.22\,\text{kg}$ using the efficiency map and the fuel cell power history. The total simulation time is $37\,\text{min}$ under the discretization shown in Tab.I. Compared with $1354\,\text{h}$ for the exemplary implementation of DP using embedded loops over time stages, states, and control variables, the computational time is reduced to only 0.05% of that. From this, the computational efficiency of the parallel algorithm is proved, which is suitable to apply for the parameter analysis in the next part.

TABLE I: Setup parameters of the two-dimensional parallel dynamic programming algorithm

Parameter	Min.	Max.	Δ	Num. of stages
Time (s)	0	9520	1	9521
P_{fc} (kW)	0	200	5	41
soc	0.2	0.9	0.0001	7001
$\frac{dP_{\text{fc}}}{dt}$ (kW/s)	-20	20	5	9

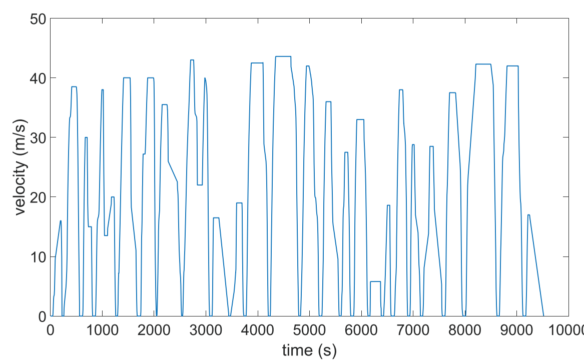
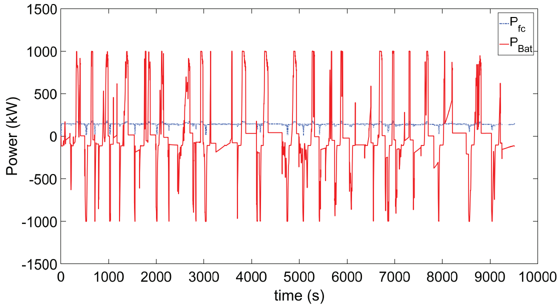


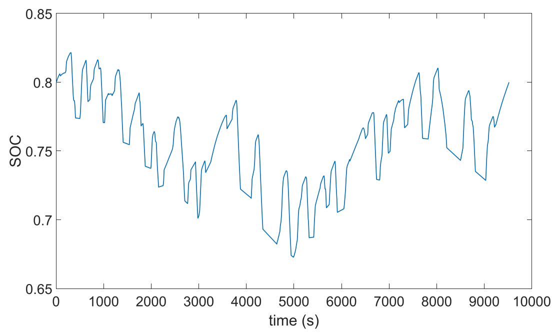
Fig. 3: Drive cycle.

B. Investigation of the effect of penalty factor

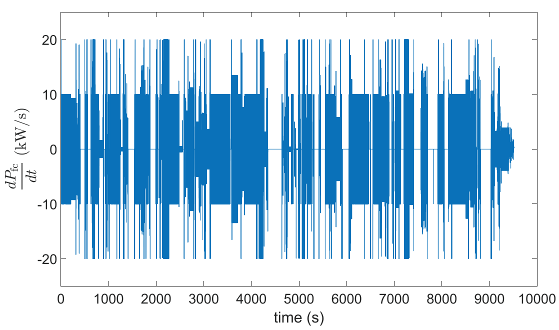
To investigate the effect of the penalty factor σ on the fuel economy, the above displayed 2d DP is executed for the different penalty factors σ. The corresponding results and comparisons are shown as follows. The soc trajectories for different penalty factors in Fig. 6 show a negligible difference. The fuel cell power history shown in Fig. 7, with the penalty factor increasing, clearly show less significant dynamic change. At some instants, high fuel cell power peaks occur to meet the high power demand during acceleration or recuperative braking operation. Moreover, Fig. 8 compares the distribution of the power change rate of the fuel cell under different penalty factors. There is much less frequent power change in the fuel cell system under higher penalty factor, from



(a) Battery and fuel cell power history.



(b) soc trajectory.



(c) Control trajectory.

Fig. 4: Variable histories under $\sigma = 0$.

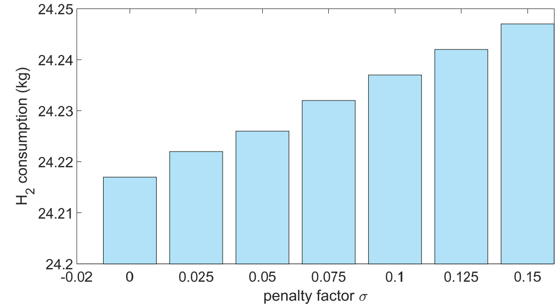
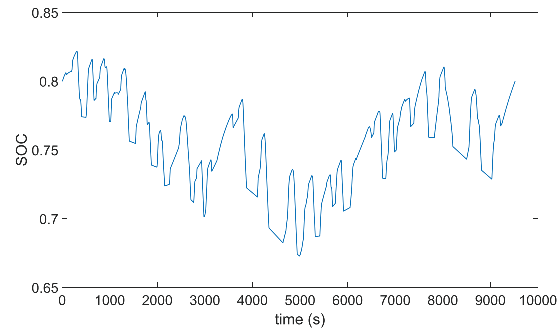
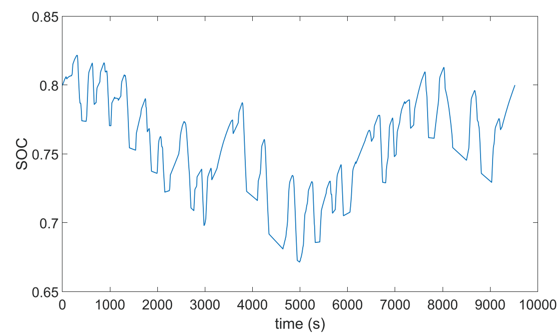


Fig. 5: Hydrogen consumption under different penalty factors.



(a) $\sigma = 0.025$



(b) $\sigma = 0.05$

Fig. 6: soc trajectories under different penalty factors.

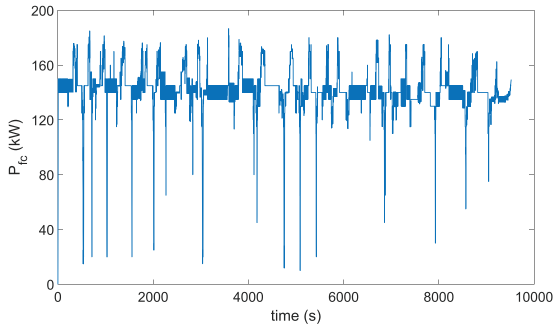
that the lifetime of fuel cell benefits [10]. As the cost for that, more hydrogen is consumed as shown in Fig. 5. To realize the optimal compromises between the hydrogen consumption and fuel cell lifetime, a precise aging model of the fuel cell needs to be developed firstly, which is beyond the scope of this work here.

V. Conclusions

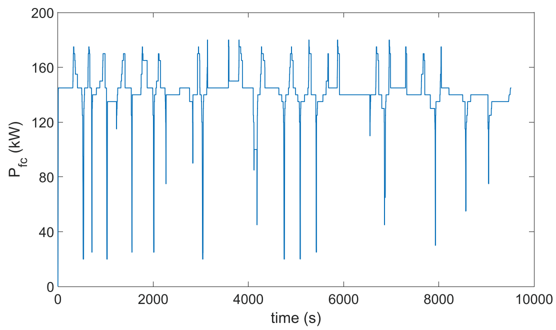
In this paper, a two-dimensional parallel dynamic programming algorithm is implemented for the energy management problem in the fuel cell hybrid train. Under the two-dimensional framework, the dynamic constraint of the control input is converted into the static constraint of the state variable, which makes the precondition of no aftereffect for using DP satisfied. Besides that, a semi-physical soft constraint mechanism is designed to enforce the soc sustenance. Moreover, both static and dynamic limits in fuel cell system are considered. With the parallel computation, the simulation time is reduced to only 0.05% of that using embedded loops. Furthermore, the effect of the weighting factor σ between minimizing hydrogen consumption and avoiding frequent changes in fuel cell operating points is investigated using this parallel algorithm. Under a proper penalty factor, the fuel cell system works in much less dynamic condition under the cost of consumption increase. This factor should be considered in developing a real-time strategy in future work.

Acknowledgments

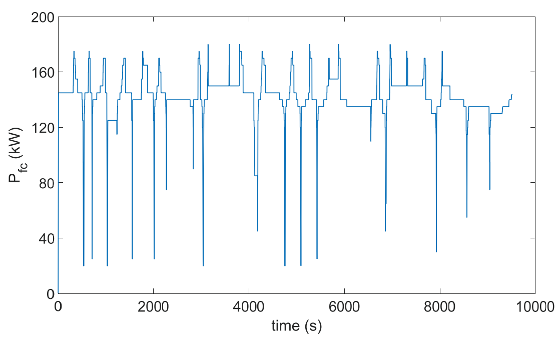
This work is funded by the BMVi under the National Innovation Programme Hydrogen and Fuel Cell Technology



(a) $\sigma = 0$

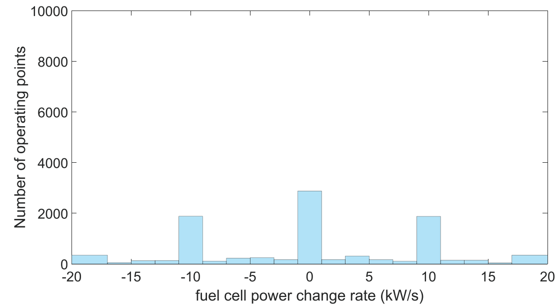


(b) $\sigma = 0.025$

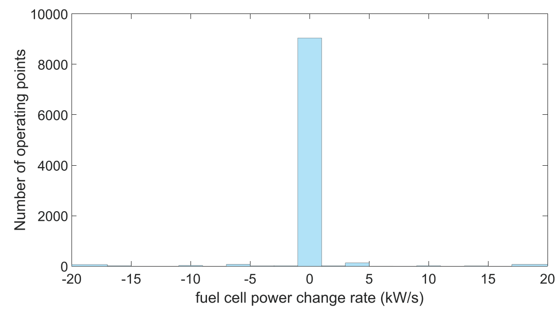


(c) $\sigma = 0.05$

Fig. 7: Fuel cell power trajectories under different penalty factors.



(a) $\sigma = 0$



(b) $\sigma = 0.025$

Fig. 8: The distribution of fuel cell power change rate under different penalty factors.

(NIP). The authors gratefully acknowledge the support by Siemens AG and NIP.

References

[1] H. Li, A. Ravey, A. N'diaye, and A. Djerdir, "A review of energy management strategy for fuel cell hybrid electric vehicle," 12 2017, pp. 1–6.

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

[3] A. M. Ali and D. Söffker, "Towards optimal power management of hybrid electric vehicles in real-time: A review on methods, challenges, and state-of-the-art solutions," *Energies*, vol. 11, no. 3, 2018.

[4] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 11 1954.

[5] M. Ansarey, M. S. Panahi, H. Ziarati, and M. Mahjoob, "Optimal energy management in a dual-storage fuel-cell hybrid vehicle using multi-dimensional dynamic programming," *Journal of Power Sources*, vol. 250, pp. 359 – 371, 2014.

[6] H. Lian, C. Zeng, and Z. Cai, "Dynamic programming based optimal control strategy of the hybrid vehicular power system," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2017, pp. 7123–7127.

[7] D. Kirk, *Optimal Control Theory: An Introduction*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2004.

[8] Y. Zou, T. Liu, S. Fengchun, and H. Peng, "Comparative study of dynamic programming and pontryagin's minimum principle on energy management for a parallel hybrid electric vehicle," *Energies*, vol. 6, pp. 2305–2318, 04 2013.

[9] D. Fares, R. Chedid, F. Panik, S. Karaki, and R. Jabr, "Dynamic programming technique for optimizing fuel cell hybrid vehicles," *International Journal of Hydrogen Energy*, vol. 40, no. 24, pp. 7777 – 7790, 2015, hydrogen and Fuel Cell Systems for Clean Energy Applications.

[10] P. Thounthong and P. Sethakul, "Analysis of a fuel starvation phenomenon of a pem fuel cell," in *2007 Power Conversion Conference - Nagoya*, April 2007, pp. 731–738.

[11] O. Sundstrom and L. Guzzella, "A generic dynamic programming matlab function," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, July 2009, pp. 1625–1630.

[12] D. Kum, H. Peng, and N. K. Bucknor, "Optimal energy and catalyst temperature management of plug-in hybrid electric vehicles for minimum fuel consumption and tail-pipe emissions," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 14–26, Jan 2013.