
Proceedings Paper:
Sanusi, I. orcid.org/0000-0002-3198-9048, Mills, A., Konstantopoulos, G. orcid.org/0000-0003-3339-6921 et al. (1 more author) (2019) Power management optimisation for hybrid electric systems using reinforcement learning and adaptive dynamic programming. In: 2019 American Control Conference (ACC). 2019 American Control Conference (ACC), 10-12 Jul 2019, Philadelphia, PA, USA. IEEE , pp. 2608-2613. ISBN 9781538679012

Power management optimisation for hybrid electric systems using reinforcement learning and adaptive dynamic programming

Ibrahim Sanusi[1], Andrew Mills[2], George Konstantopoulos[3], Tony Dodd[4]

***Abstract*— This paper presents an online learning scheme based on reinforcement learning and adaptive dynamic programming for the power management of hybrid electric systems. Current methods for power management are conservative and unable to fully account for variations in the system due to changes in the health and operational conditions. These conservative schemes result in less efficient use of available power sources, increasing the overall system costs and heightening the risk of failure due to the variations. The proposed scheme is able to compensate for modelling uncertainties and the gradual system variations by adapting its performance function using the observed system measurements as reinforcement signals. The reinforcement signals are nonlinear and consequently neural networks are employed in the implementation of the scheme. Simulation results for the power management of an autonomous hybrid system show improved system performance using the proposed scheme as compared with a conventional offline dynamic programming approach.**

I. INTRODUCTION

Hybrid electric systems such as those deployed on unmanned aerial vehicles (UAV) often have architectures which support two or more power sources [1]. The power sources typically consist of joint propulsion and electrical generation systems such as the gas turbine engines (GTE), and one or more energy storage devices e.g fuel cells, supercapacitors and batteries [2]. With limited energy resources on-board the hybrid systems, power management strategies have been identified as key enabling technologies to support enhanced capabilities of the systems such as longer operational times and increased endurance [1], [3]. The enhanced capabilities are envisaged to be associated with increased power requirements, mission risks and overall system costs. It is therefore the aim of the power management strategies to reduce the risks and overall system costs whilst providing an effective way to support the system power requirements.

The operation of an autonomous vehicle can be divided into phases, for example a car or aircraft may have pre-planned routes or missions (e.g hill climbing or aircraft radar sweeps) associated with varying power demands [1]. There is an energy interdependency between the operation phases as the power drawn from a source for a duration of a phase may become unavailable for the remaining phases. This is the case for the energy storage devices where the available power for a phase is dependent on previous charge/discharge energy cycles at the other phases. Current industry-standard approaches for the power management are therefore based on pre-defined rule based power schedules between the multiple power sources [4]. These approaches follow a series of *if-then* rules designed for the worst-case peak power requirements. As such, they are usually conservative and unable to adapt to dynamic changes in the systems. Over the years, research trends have favoured optimisation based power management approaches to optimise the desired power requirements and constraints of the hybrid systems [5], [6].

In [7], the hybrid system power management was formulated as a mixed-integer nonlinear multi-objective optimisation problem and solved using a differential evolutionary fuzzy scheme. The proposed solution is however non-deterministic and does not provide any solution guarantees to be suited for real-time implementation. Consequently, an intelligent power management system (PMS) that guarantees a feasible solution was proposed in [3] using a three level optimisation strategy. Both approaches are, however, unable to account for unmodelled variations in the system resulting from degradation or changes in the system operating conditions. Furthermore, the energy interdependency between the sources is considered in a heuristic rule based manner that is suboptimal in both schemes.

Other approaches have considered the dynamic programming (DP) technique which is well suited to handle the energy interdependency by solving the optimisation problem as a sequence of operations [8]. The DP technique is based on Bellman's optimality principle and limits the optimisation search to the potentially optimal trajectories. In [9], DP was used to develop a hydroelectric scheduling technique between thermal and hydro power sources to minimise the system generation cost while satisfying the system load requirements. Likewise, [10] proposed an optimal dispatch of direct load control using DP to minimise the system production cost. Related works on power management optimisation using DP include [11], [12] for optimal charge/discharge of energy storage devices; [2], [5] and [13] for optimal energy management for hybrid electric vehicles. All of these works depend on accurate system models and are therefore limited in their ability to account for system variations and modelling uncertainties.

The authors are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, United Kingdom.
[1]iesanusi1@sheffield.ac.uk
[2]a.r.mills@sheffield.ac.uk
[3]g.konstantopoulos@sheffield.ac.uk
[4]t.j.dodd@sheffield.ac.uk

Extension of the DP techniques to provide adaptation and self-learning capabilities are enabled using frameworks based on reinforcement learning (RL) and adaptive dynamic programming (ADP) [8], [14], [15], [16]. Using ADP, an adaptive power management scheme was developed for residential load management in both [17] and [18]. Both of these approaches applied a heuristic approach in the online management scheme by limiting the control inputs to one of three choices as *charge, discharge* and *idle*, greatly reducing the optimality of the solutions. In [19], a dual Q-learning scheme was proposed as an extension to the residential load management optimisation. This scheme is however restricted to problems involving repeated known cycles for the load and system costs.

In contrast to the above approaches, this paper proposes a new online learning scheme based on reinforcement learning and adaptive dynamic programming (RL-ADP) that is able to compensate for both modelling uncertainties and gradual variability due to changes in the system health or operating conditions. The system learns by using reinforcement signals in the form of the system measurements to adapt the system performance function, which is then used to determine the best power control strategy online. The rest of the paper is organised as follows. Section II provides the mathematical formulation for the power management problem while Section III provides a dynamic programming solution. Section IV extends the RL-ADP theory to the formulated problem and introduces the proposed algorithm. Simulation results are presented in Section V and conclusion in Section VI.

II. PROBLEM DEFINITION

An autonomous hybrid electric system consisting of a GTE propulsion system and an energy storage device in form of a battery is considered. The propulsion system provides the necessary thrust needed by the system whilst also providing electrical power to the on-board system loads. Electrical power is generated from the propulsion system through two sets of generators coupled to the rotating shafts as shown in Fig. 1. This additional load on the propulsion system results in higher fuel burn at peak load requirements. A hybrid battery integration therefore promotes feasibility of power scheduling for efficient system operation and increased system capability.

The governing power equation for the system is given by:

$$P_{eng} = P_{FN} + P_{prop} + P_{core} \tag{1}$$

where P_{eng} is the total useful power from the GTE, P_{FN} is the propulsive power needed for thrust while P_{prop} and P_{core} are respectively the electrical power from the propeller and core shafts. For the load demand side, the power balance equation is given by:

$$P_{prop} + P_{core} = P_{load} - P_{bat} \tag{2}$$

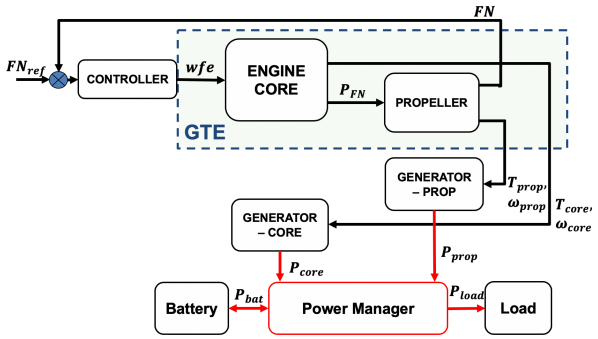

Fig. 1. Block diagram of a hybrid electric system consisting of a GTE with battery integration. The GTE produces thrust (FN) for a given amount of fuel flow (wfe) whilst also providing electric power via two sets of generators coupled to both the propeller and core shafts.

where P_{load} is the required load power and P_{bat} is the battery power output. $P_{bat} > 0$ indicates that the battery is discharging, and charging when $P_{bat} < 0$. It is assumed that the thrust requirement is always satisfied by the thrust control loop, thus combining (1) and (2) gives:

$$'P_{eng} = P_{load} - P_{bat} \tag{3}$$

where $'P_{eng} = P_{eng} - P_{FN}$. Fig. 2 shows a sample power demand profile for a hybrid electric system and the discrete time steps k considered for optimisation. The change in energy between the time steps k is defined as:

$$\Delta E_{k+1} := {}'P_{eng,k}\Delta t = (P_{load,k} - P_{bat,k})\Delta t \tag{4}$$

The dynamics for the battery state of charge (SOC) consistent with [18] and [19] is given as:

$$SOC_{k+1} = SOC_k - sign(P_{bat,k}) \cdot \eta(P_{bat,k})\Delta t \tag{5}$$

where $sign(P_{bat})$ indicates discharging (+) or charging (-) of the battery while $\eta(P_{bat})$ gives the battery efficiency. The power management optimisation problem therefore aims to find the control strategy for P_{bat} that will optimise a desired performance cost for a given load profile P_{load}. The state equations are thus defined as follows:

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, u_k) = \begin{bmatrix} (P_{load,k} - u_k)\Delta t \\ x_{2,k} - sign(u_k) \cdot \eta(u_k)\Delta t \end{bmatrix}$$
$$\textit{subject to: } \boldsymbol{x} \in \mathbb{X}, \quad u \in \mathbb{U} \tag{6}$$

where $\boldsymbol{x}_k = \begin{bmatrix} \Delta E_k & SOC_k \end{bmatrix}^\top$, $u_k = P_{bat,k}$ and $\mathbb{X}, \mathbb{U}$ are

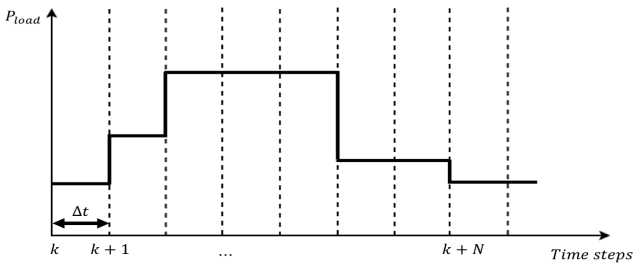

Fig. 2. Sample operational phases and power requirements for the autonomous hybrid electric system in time steps $k, k+1, \cdots, k+N$.

sets of constraints on the state and input respectively. The desired cost to be optimised at the discrete time steps k is:

$$Q(\boldsymbol{x}_k, u_k) = \sum_{n=k}^{N} \lambda^{n-k} R(\boldsymbol{x}_n, u_n) \tag{7}$$

where N is the length of the load profile, $\lambda \in [0, 1]$ is a forgetting factor and $R(\boldsymbol{x}, u)$ is a scalar reward signal assumed to be directly measurable from the system. The solution to the formulated optimisation problem will require knowledge of the system models and result in the nonlinear Hamilton-Jacobi Bellman (HJB) equations which are known to be difficult and often impossible to solve analytically [20]. An approach that provides a recursive solution to the optimisation problem will now be presented.

III. DYNAMIC PROGRAMMING SOLUTION

DP considers the recursive form for the cost function of (7) as:

$$Q(\boldsymbol{x}_k, u_k) = R(\boldsymbol{x}_k, u_k) + \lambda Q(\boldsymbol{x}_{k+1}, u_{k+1}) \tag{8}$$

Equation (8) is called the Bellman equation and serves as a fixed-point equation for the Bellman's principle of optimality [21]. DP assumes that the system model is known, and discretises the system states into levels with associated cost Q. DP therefore uses the Bellman equation to limit the optimisation search to only the optimal trajectories by solving the following recursion:
Solve backwards from terminal state $Q(\boldsymbol{x}_N, u_N)$ *for* $n = N : -1 : k$

$$Q(\boldsymbol{x}_k, u_k) \leftarrow \min_{u_k} \left\{ R(\boldsymbol{x}_k, u_k) + \lambda Q(\boldsymbol{x}_{k+1}, u_{k+1}) \right\}$$

$$\text{subject to: } \boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, u_k) = \begin{bmatrix} (P_{load,k} - u_k)\Delta t \\ x_{2,k} - sign(u_k) \cdot \eta(u_k)\Delta t \end{bmatrix}$$

$$\boldsymbol{x} \in \mathbb{X}, \quad u \in \mathbb{U} \tag{9}$$

Remarks

- The problem space for DP is known to increase with increased number of states and actions. This is known as the DP curse of dimensionality. Although, known to limit its practicality, DP has been shown to scale well with problems involving hundreds of states and actions [8].
- A major drawback of DP is its dependence on accurate system models (i.e. $F(\boldsymbol{x}, u)$ and $R(\boldsymbol{x}, u)$). For this problem, the state equations, i.e. $F(\boldsymbol{x}, u)$, are given by the system energy requirements and are known. However, analytical models to accurately describe the changes in the system health or operational conditions are typically unknown. These changes are assumed to reflect in the measured reward signals, i.e. gradual changes in the measured GTE and battery efficiencies. Consequently, the standard DP framework assumes a fixed $R(\boldsymbol{x}, u)$ and is unable to cope with varying system conditions. An online framework based on RL-ADP is therefore proposed to compensate for both modelling uncertainties and gradual variations in the system by recursively solving the sequence of operations using dynamic programming and function approximations.

IV. RL-ADP SOLUTION

Motivated by the Bellman optimality equations, RL-ADP algorithms make use of iterative fixed-point equations that are known to successively lead to improved policies [22]. The iterative fixed-point equations involve both value and policy update steps respectively given as:

$$Q_{k+1}(\boldsymbol{x}_k, u_k) = R(\boldsymbol{x}_k, u_k) + \lambda Q_k(\boldsymbol{x}_{k+1}, u_{k+1}) \tag{10}$$

$$u_{k+1} = \arg\min_{u_k} \left(R(\boldsymbol{x}_k, u_k) + \lambda Q_{k+1}(\boldsymbol{x}_{k+1}, u_{k+1}) \right) \tag{11}$$

These are implemented *forward-in-time* without requiring models of the system. Convergence of the iterative updates has been proven by showing that interleaving (10) and (11) leads to a contraction map under certain conditions [20]. Learning is achieved by making use of function approximations and temporal difference (TD) error as follows:

$$Q(\boldsymbol{x}, u) \approx \beta^\top \Phi(\boldsymbol{x}, u) \tag{12}$$

$$\therefore e_k = R(\boldsymbol{x}_k, u_k) + \lambda \beta_k^\top \Phi(\boldsymbol{x}_{k+1}, u_{k+1}) - \beta_k^\top \Phi(\boldsymbol{x}_k, u_k) \tag{13}$$

where $\Phi(\boldsymbol{x}, u)$ is a set of basis function and β are the function weights. Equation (13) is solved for $e_k = 0$ at each time step to yield the least squares approximation to the TD error equation. This way, only the measured data (i.e $R(\boldsymbol{x}_k, u_k)$, $\boldsymbol{x}_{k+1}$ and u_k) are used to compute the optimal control inputs without knowledge of the system models.

Given a load profile $P_{load,k}|k = 0, 1, \cdots, N$, we wish to solve online the best control strategy (i.e. control sequence $\boldsymbol{U}_N = [u_0, u_1, \cdots, u_N]$) that minimises the desired cost. Mathematically,

$$\boldsymbol{U}_N = \min Q^*(\boldsymbol{x}_k, u_k)$$

$$= \min_{u_k} \left\{ R(\boldsymbol{x}_k, u_k) + \lambda \min_{u_{k+1}} \left\{ R(\boldsymbol{x}_{k+1}, u_{k+1}) + \cdots \right.\right.$$

$$\left.\left. + \lambda \min_{u_{k+j-1}} \left\{ R(\boldsymbol{x}_{k+j-1}, u_{k+j-1}) + \lambda \min_{u_{k+j}} Q^*(\boldsymbol{x}_{k+j}, u_{k+j}) \right\} \right\} \right\} \tag{14}$$

$$for\ j = 1, 2, \cdots, N$$

Conventional RL-ADP algorithms require that the optimal Q-function strictly follows the one-step Bellman optimality equation:

$$Q^*(\boldsymbol{x}_{N-1}, u_{N-1}) = R(\boldsymbol{x}_{N-1}, u_{N-1}) + \lambda \min_{u_N} Q^*(\boldsymbol{x}_N, u_N) \tag{15}$$

Clearly, the power management optimisation problem (14) involves varying Q-functions due to the dependence of x on the varying load requirements, P_{load} and does not conform with (15). A novel approach is therefore to consider the optimisation problem as being composed of:

- A planning/scheduling phase to determine the control sequence $\boldsymbol{U}_N$ using algorithms such as DP.

- Iterative adaptation of the Q-function from the system measurements to compensate for modelling uncertainties and system variation in the reward measurements.

Remarks

- Obtaining a Q-function approximation that spans the entire state space in (14) may be infeasible with increased number of discrete stages for optimisation. This negates the use of traditional Q-learning algorithms but favours the iterative adaptation of the varying Q-functions at each stage:

$$\therefore Q(\boldsymbol{x}_k, u_k) \approx \beta_k^\top \Phi(\boldsymbol{x}_k, u_k) = \sum_{n=k}^{k} \lambda^{n-k} R(\boldsymbol{x}_n, u_n) = R(\boldsymbol{x}_k, u_k) \tag{16}$$

- Consequently, the adapted function gives the instantaneous reward signals while convergence to the optimal Q-function ($Q^*(\boldsymbol{x}, u)$) is obtained using an online DP algorithm.

Adaptation of the Q-function is achieved by defining a cost E_k based on the TD error (13) as follows:

$$E_k = \frac{1}{2} e_k^2 \tag{17}$$

$$\beta_{k+1} = \beta_k - \gamma \frac{\partial E_k}{\partial \beta_k} = \beta_k - \gamma \left[\frac{\partial E_k}{\partial Q(\boldsymbol{x}_k, u_k)} \frac{\partial Q(\boldsymbol{x}_k, u_k)}{\partial \beta_k} \right] \tag{18}$$

where $\gamma > 0$ is the learning rate. The adapted Q-function is then used to generate reward signals and used in an online planning/scheduling scheme to determine the control sequence $\boldsymbol{U}_N$. Following the computed control sequence, only the first control input is applied to the system online, and the process is repeated. Algorithm 1 gives the template for the proposed procedure.

Algorithm 1 Online RL-ADP framework for power management optimisation

1: Initialise $Q(\boldsymbol{x}, u) \approx \beta_0^\top \Phi(\boldsymbol{x}, u)$ and obtain the control sequence $\boldsymbol{U}_N = [u_0, u_1, \cdots, u_N]$ from offline dynamic programming of (9) with $R(\boldsymbol{x}_n, u_n) = \beta_0^\top \Phi(\boldsymbol{x}_n, u_n) \mid_{n=N:-1:k}$

Online computation: for $k = 0 : N$

2: Apply the first control input u_k.

Q-function update step

3: Obtain real-time measurements for the reward signal $R(\boldsymbol{x}_k, u_k)$, the states $\boldsymbol{x}_{k+1}$ and the control input u_k.

4: Compute the TD error from (13), and adapt the Q-function using (17) and (18).

Online planning/scheduling step

5: Perform online dynamic programming using the updated Q-function with $R(\boldsymbol{x}_n, u_n) = \beta_{k+1}^\top \Phi(\boldsymbol{x}_n, u_n) \mid_{n=N:-1:k+1}$ and determine the control sequence $\boldsymbol{U}_{k \to N} = [u_{k+1}, u_{k+2}, \cdots, u_N]$.

6: Repeat steps 2 to 5 till $k = N$.

V. SIMULATION STUDIES

The proposed RL-ADP framework for power management optimisation is demonstrated on a representative autonomous hybrid electric system model to compensate for both modelling uncertainties and variations in the system efficiency. The electrical power from the GTE and battery are constrained between $30KW \leq {}'P_{eng} \leq 150KW$ and $-60KW \leq P_{bat} \leq 60KW$ respectively i.e. the sets $\mathbb{X}, \mathbb{U}$, while the battery SOC is expressed as a percentage between $0 - 100\%$. The reward signal is assumed given by the GTE efficiency, η_{GTE} which is the measured pounds of fuel flow per hour per unit thrust. The intervals between the discrete time steps k, i.e Δt for the optimisation are considered to be fixed and determined by changes in the load demand as shown in Fig. 2.

Given a load profile $P_{load,k}$, the aim of the power management optimisation framework is then to determine the best power control strategy that optimises the cost function of (7) subject to variations in the systems.

Algorithm implementation

Preliminary test was first carried out to determine suitable basis function that can model the search space complexities of the power management optimisation problem involving the different load demands and the system energy constraints. The test data consists of randomly sampled ${}'P_{eng}$, P_{bat} and SOC levels with the reward signals as the measured η_{GTE} from the system, penalised with large values for violations of the system energy constraints. Approximation of the Q-function using the test data with some choice of basis function is then carried out and the results shown below:

TABLE I
CROSS-VALIDATED MEAN-SQUARED ERROR (MSE)

Model	**Polynomial**	**2-layer neural network**		
Complexity	2^{nd} order	5 hidden	20 hidden	50 hidden
MSE	206.46	0.44	0.26	0.18

Results from Table I indicate that the approximation is more complex than a second order and use of higher order polynomials may lead to over-fitting. Neural networks however offer better approximation to cope with the nonlinearities with considerations for the trade-off between model complexity and the cross-validated MSE. Consequently, a 2-layer neural network for the Q-function is trained as follows:

$$Q(\boldsymbol{x}, u) \approx {\beta^{(2)}}^\top \Phi(\boldsymbol{x}, u) \tag{19}$$

where

$$\Phi(\boldsymbol{x}, u) = \Phi(\underline{\boldsymbol{x}}) = \begin{bmatrix} 1 & \frac{e^{{\beta^{(1)}}^\top \underline{\boldsymbol{x}}} - e^{-{\beta^{(1)}}^\top \underline{\boldsymbol{x}}}}{e^{{\beta^{(1)}}^\top \underline{\boldsymbol{x}}} + e^{-{\beta^{(1)}}^\top \underline{\boldsymbol{x}}}} \end{bmatrix} = \begin{bmatrix} 1 & \frac{e^{\boldsymbol{z}} - e^{-\boldsymbol{z}}}{e^{\boldsymbol{z}} + e^{-\boldsymbol{z}}} \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{a} \end{bmatrix} \tag{20}$$

$\underline{\boldsymbol{x}} = \begin{bmatrix}1 & x_1 & x_2 & u\end{bmatrix}^\top \in \mathbb{R}^{1\times 4}$, $\boldsymbol{z} = \beta^{(1)^\top}\underline{\boldsymbol{x}} \in \mathbb{R}^{n_h\times 1}$, $\boldsymbol{a} = tanh(\boldsymbol{z}) = \frac{e^{\boldsymbol{z}}-e^{-\boldsymbol{z}}}{e^{\boldsymbol{z}}+e^{-\boldsymbol{z}}} \in \mathbb{R}^{n_h\times 1}$, n_h is the number of hidden nodes, and $\beta^{(1)} \in \mathbb{R}^{4\times n_h}$, $\beta^{(2)} \in \mathbb{R}^{n_{h+1}\times 1}$ are respectively the inner and outer layer weights. The update sequence for the function weights follows from (17) and (18):

Outer layer

$$\beta^{(2)}_{k+1} = \beta^{(2)}_k - \gamma\left[\frac{\partial E_k}{\partial Q(\boldsymbol{x}_k,u_k)}\frac{\partial Q(\boldsymbol{x}_k,u_k)}{\partial \beta^{(2)}_k}\right] \tag{21}$$

where $\frac{\partial E_k}{\partial Q(\boldsymbol{x}_k,u_k)} = \lambda e_k$ and $\frac{\partial Q(\boldsymbol{x}_k,u_k)}{\partial \beta^{(2)}_k} = \Phi(\boldsymbol{x}_k,u_k)$

Inner layer

$$\beta^{(1)}_{k+1} = \beta^{(1)}_k - \gamma\left[\frac{\partial E_k}{\partial Q(\boldsymbol{x}_k,u_k)}\frac{\partial Q(\boldsymbol{x}_k,u_k)}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \beta^{(1)}_k}\right] \tag{22}$$

where $\frac{\partial Q(\boldsymbol{x}_k,u_k)}{\partial \boldsymbol{a}} = \sum_{i=2}^{n_h+1}\beta^{(2)}_{(i)}$, $\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{z}} = 1 - tanh(\boldsymbol{z})^2$ and $\frac{\partial \boldsymbol{z}}{\partial \beta^{(1)}_k} = \underline{\boldsymbol{x}}$. The parameters for the neural network implementation are selected as follows: $\lambda = 1$, $n_h = 20$ and $\gamma = 0.3e^{-4}$. There are no stability guarantees for this choice of weight update, but strategies to limit divergence such as the use of target networks discussed in [23] proved successful in the provided simulations. Two scenarios are considered to demonstrate the effectiveness of the proposed approach:

A. Performance of offline power schedule vs Algorithm 1

Algorithms such as DP can be used to construct offline power schedules for the power management optimisation problem. Typically, these are designed for fixed nominal system models for the worst-case peak power requirements and are usually suboptimal by being unable to adapt to the actual system conditions. A DP algorithm as described in Section III was used to compute feasible offline power schedules for the hybrid system and serves as the baseline.

Given the system mismatch and other uncertainties at design time between the nominal and actual (but unknown) GTE efficiency, the computed offline power schedules will be suboptimal and result in reduced system performance. Fig. 3 and Fig. 4 show the given load profile and the results from using Algorithm 1 compared with the baseline. Whilst both power management strategies were able to satisfy the system load requirements, Algorithm 1 was able to compensate for the system mismatch using the actual system measurements as reward signals to deliver improved performance as shown by the reduced average fuel consumed during the simulation.

B. Variation in system objectives and load requirements

The use of the offline (pre-defined) power schedules heightens the risk of failure due to system variations. Variations can occur from changes in system operation objectives which may result in a change in the load demand profile [3]. Consider a load demand change at time steps 19 to 20 in Fig. 5. The offline power schedule is infeasible as it is unable to adapt to the event change and satisfy the system load requirements at all times. Algorithm 1 was however able to satisfy the load requirements by fully delivering the required load power, given the information about the load change online. The RL-ADP scheme is therefore able to determine the best power strategy by computing the best charging/discharging cycles for the battery SOC in order to cope the load change as shown in Fig. 4 and Fig. 6.

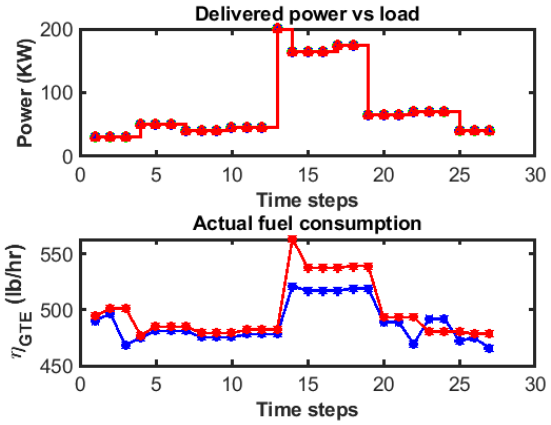

Fig. 3. **TOP**: Offline DP power scheduling (red) and Algorithm 1 (blue) vs the load demand profile (green). The load demand profile is overlaid as both algorithms satisfied the requirements. **BOTTOM**: Fuel consumption using offline DP power scheduling with average fuel: 498.04 lb/hr (red) vs Algorithm 1 with average fuel: 488.54 lb/hr (blue).

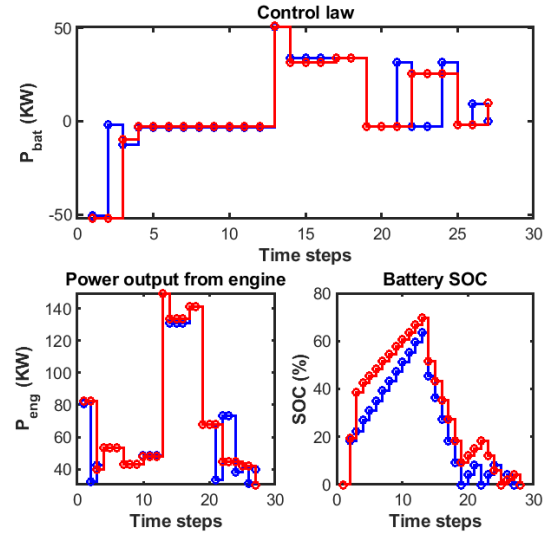

Fig. 4. **TOP**: Control law from applying offline DP power scheduling (red) vs Algorithm 1 (blue). **BOTTOM**: GTE power output and battery SOC from implementation of both power management strategies.

VI. CONCLUSIONS

This paper has proposed and demonstrated an online power management optimisation scheme based on reinforce-

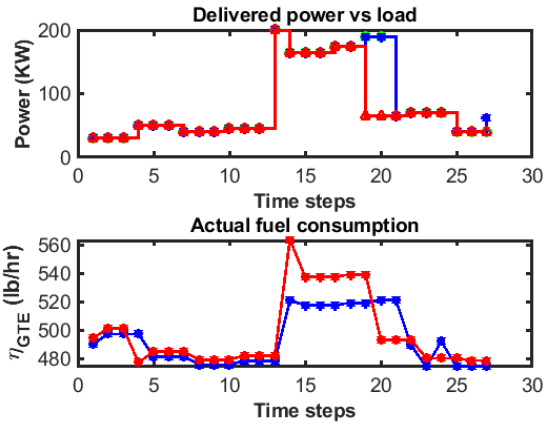

Fig. 5. **TOP**: Offline DP power scheduling (red) and Algorithm 1 (blue) vs the load demand profile (green). The load demand profile is overlaid by the output of Algorithm 1 indicating that the requirements are fully satisfied but not with the Offline DP. **BOTTOM**: Fuel consumption using offline DP power scheduling with average fuel: 498.04 lb/hr (red) vs Algorithm 1 with average fuel: 493.47 lb/hr (blue).

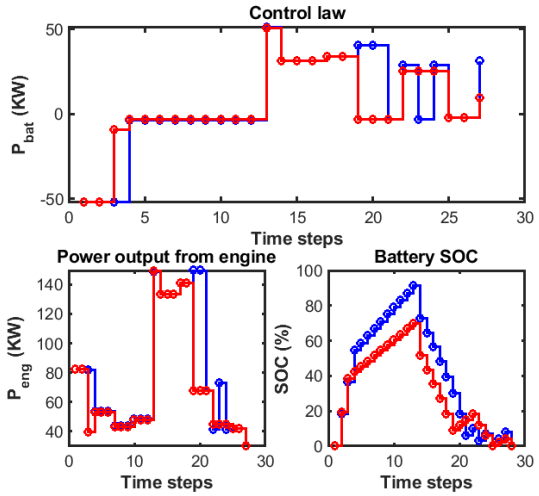

Fig. 6. **TOP**: Control law from applying offline DP power scheduling (red) vs Algorithm 1 (blue). **BOTTOM**: GTE power output and battery SOC from implementation of both power management strategies.

ment learning and adaptive dynamic programming. Current power management strategies are heuristic and thus suboptimal, and are unable to compensate for modelling uncertainties and variation in system conditions. The proposed scheme computes online the optimal control strategies by using system measurements as reinforcement signals to adapt the system performance function. Future work will extend the proposed strategy to multiple power sources with increased number of states.

ACKNOWLEDGMENT

The authors are with the Rolls-Royce Control and Monitoring Systems Technology Centre, University of Sheffield, UK, and kindly acknowledge the contributions of Rolls-Royce PLC in this work.

REFERENCES

[1] L. Karunarathne, J. T. Economou, K. Knowles, Power and energy management system for fuel cell unmanned aerial vehicle, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 226 (4) (2012) 437–454.

[2] L. V. Pérez, G. R. Bossio, D. Moitre, G. O. García, Optimization of power management in an hybrid electric vehicle using dynamic programming, Mathematics and Computers in Simulation 73 (1-4) (2006) 244–254.

[3] M. M. Mansor, I. Giagkiozis, D. Wall, A. R. Mills, R. C. Purshouse, P. J. Fleming, Real-time improved power management for autonomous systems, IFAC Proceedings Volumes 47 (3) (2014) 2634–2639.

[4] B. Belvedere, M. Bianchi, A. Borghetti, M. Paolone, A microcontroller-based automatic scheduling system for residential microgrids, in: PowerTech, 2009 IEEE Bucharest, IEEE, 2009, pp. 1–6.

[5] H. Lee, J. Jeong, Y.-i. Park, S. W. Cha, Energy management strategy of hybrid electric vehicle using battery state of charge trajectory information, International Journal of Precision Engineering and Manufacturing-Green Technology 4 (1) (2017) 79–86.

[6] J. Hoelzen, Y. Liu, B. Bensmann, C. Winnefeld, A. Elham, J. Friedrichs, R. Hanke-Rauschenbach, Conceptual design of operation strategies for hybrid electric aircraft, Energies 11 (1) (2018) 217.

[7] S. Abedi, A. Alimardani, G. Gharehpetian, G. Riahy, S. Hosseinian, A comprehensive method for optimal power management and design of hybrid res-based autonomous energy systems, Renewable and Sustainable Energy Reviews 16 (3) (2012) 1577–1587.

[8] W. B. Powell, Approximate Dynamic Programming: Solving the curses of dimensionality, Vol. 703, John Wiley & Sons, 2007.

[9] S.-C. Chang, C.-H. Chen, I.-K. Fong, P. B. Luh, Hydroelectric generation scheduling with an effective differential dynamic programming algorithm, IEEE Transactions on Power Systems 5 (3) (1990) 737–743.

[10] Y.-Y. Hsu, C.-C. Su, Dispatch of direct load control using dynamic programming, IEEE Transactions on Power Systems 6 (3) (1991) 1056–1061.

[11] X. Dong, G. Bao, Z. Lu, Z. Yuan, C. Lu, Optimal battery energy storage system charge scheduling for peak shaving application considering battery lifetime, in: Informatics in Control, Automation and Robotics, Springer, 2011, pp. 211–218.

[12] Y. Riffonneau, S. Bacha, F. Barruel, S. Ploix, Optimal power flow management for grid connected pv systems with batteries, IEEE Transactions on Sustainable Energy 2 (3) (2011) 309–320.

[13] C.-C. Lin, H. Peng, J. W. Grizzle, J.-M. Kang, Power management strategy for a parallel hybrid electric truck, IEEE Transactions on Control Systems Technology 11 (6) (2003) 839–849.

[14] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, Vol. 1, MIT press Cambridge, 1998.

[15] D. P. Bertsekas, J. N. Tsitsiklis, Neuro-dynamic programming: an overview, in: Decision and Control, 1995, Proceedings of the 34th IEEE Conference on, Vol. 1, IEEE, 1995, pp. 560–564.

[16] P. J. Werbos, Approximate dynamic programming for real time control and neural modeling, Handbook of intelligent control (1992) 493–526.

[17] M. Boaro, D. Fuselli, F. De Angelis, D. Liu, Q. Wei, F. Piazza, Adaptive dynamic programming algorithm for renewable energy scheduling and battery management, Cognitive Computation 5 (2) (2013) 264–277.

[18] T. Huang, D. Liu, A self-learning scheme for residential energy system control and management, Neural Computing and Applications 22 (2) (2013) 259–269.

[19] Q. Wei, D. Liu, G. Shi, A novel dual iterative q learning method for optimal battery management in smart residential environments, IEEE Transactions on Industrial Electronics 62 (4) (2015) 2509–2518.

[20] F. L. Lewis, D. Vrabie, K. G. Vamvoudakis, Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers, IEEE Control Systems 32 (6) (2012) 76–105.

[21] R. Bellman, A markovian decision process, Journal of Mathematics and Mechanics (1957) 679–684.

[22] D. P. Bertsekas, Dynamic programming and optimal control, Vol. 1, Athena Scientific Belmont, MA, 1995.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529.