A ROBUST FUEL OPTIMIZATION STRATEGY FOR HYBRID ELECTRIC VEHICLES: A DEEP REINFORCEMENT LEARNING BASED CONTINUOUS TIME DESIGN APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper deals with the fuel optimization problem for hybrid electric vehicles in reinforcement learning framework. Firstly, considering the hybrid electric vehicle as a completely observable non-linear system with unknown dynamics in continuous time frame, we solve an open-loop deterministic trajectory optimization problem without explicitly considering the system dynamics. This is followed by the design of a deep reinforcement learning based controller for the non-linear system such that the actual states and the control policy remain close to the optimal trajectory and optimal policy even in the presence of external disturbances, modeling errors, uncertainties and noise. The control strategy will autonomously learn the optimal policy and adapt itself to the different conditions which is in sharp contrast to the conventional methods like PID or Model Predictive Control (MPC) that mostly depend on a set of pre-defined rules and provide sub-optimal solutions to the control problem. The controller thus designed is compared with the traditional fuel optimization strategies for hybrid electric vehicles to illustate the efficacy of the proposed method.

1 INTRODUCTION

Hybrid electric vehicles powered by fuel cells and batteries have attracted great enthusiasm in modern days as they have the potential to eliminate emissions from the transport sector. Now, both the fuel cells and batteries have got several operational challenges which makes the separate use of each of them in automotive systems quite impractical. HEVs and PHEVs powered by conventional diesel engines and batteries merely reduce the emissions, but cannot eliminate completely. Some of the drawbacks include carbon emission causing environmental pollution from fuel cells and long charging times, limited driving distance per charge, non-availability of charging stations along the driving distance for the batteries. Fuel Cell powered Hybrid Electric Vehicles (FCHEVs) powered by fuel cells and batteries offer emission-free operation while overcoming the limitations of driving distance per charge and long charging times. So, FCHEVs have gained significant attention in recent years. As we find, most of the existing research which studied and developed several types of Fuel and Energy Management Systems (FEMS) for transport applications include Sulaiman et al. (2018) who has presented a critical review of different energy and fuel management strategies for FCHEVs. Li et al. (2017) has presented an extensive review of FMS objectives and strategies for FCHEVs. These strategies, however can be divided into two groups, i.e., model-based and model-free. The model-based methods mostly depend on the discretization of the state space and therefore suffers from the inherent curse of dimensionality. The coumputational complexity increases in an exponential fashion with the increase in the dimension of the state space. This is quite evident in the methods like state-based EMS (Jan et al., 2014; Zadeh et al., 2014; 2016), rule-based fuzzy logic strategy (Motapon et al., 2014), charge depleting and charge sustaining (CDCS) strategy (Moura et al., 2010), wavelet transform based strategy (Zhang et al., 2008), variable frequency control techniques (Segura et al., 2012), classical PI and PID strategies (Segura et al., 2012), equivalent consumption minimization strategy (ECMS) (Paganelli et al., 2002; Garcia et al., 2012; 2014; Yan et al., 2012; 2014), Potryagin's minimum principle (PMP) (Zheng et al., 2013; 2014), model predictive control (MPC) (Kim et al., 2007; Torreglosa et al., 2014), differential dynamic programming (DDP) (Kim et al.,

2007) and adaptive optimal control (Lin et al., 2011). Out of all these methods, differential dynamic programming is considered to be computationally quite efficient which rely on the linearization of the non-linear system equations about a nominal state trajectory followed by a policy iteration to improve the policy. Yu et al. (2018) have performed a linearization of the system dynamics and the cost function about the nominal trajectory and then designed a LQG controller using dynamic programming (DP). In this approach, the control policy for fuel optimization is used to compute the optimal trajectory and the policy is updated until the convergence is achieved.

The model-free methods mostly deal with the Adaptive Dynamic Programming (Bithmead et al., 1991; Zhong et al., 2014) and Reinforcement Learning (RL) paradigms (Mitrovic et al., 2010; Khan et al., 2012). Here, they tend to compute the control policy by continous engagement with the environment and measuring the system response. These continuous trials help the algoithm to learn a function that will enable it to achieve at a solution of the DP equation recursively in an online fashion. In deep reinforcement learning, multi-layer neural networks are used to represent the learning function using a non-linear parameterized approximation form. Although a compact paremeterized form do exist for the learning function, the inability to know it apriori renders the method suffer from the curse of dimensionality.

Several reserachers in the past have used the DDP (Jacobsen et al., 1970; Theoddorou et al., 2010; Levine et al., 2014) such as the RL techniques (Todorov et al., 2009; Akrour et al., 2016) where they have preffered to design a LQR feedback policy around an optimal trajectory followed by a recursive optimization of the control law using multiple simulations. However, the strategy still suffers from the problem of compuatational complexity which is very large as given by $O(d^2)$ where d is the dimension of the state space for a partially observed system. Morever, the problem of sub-optimal convergence of the solution also leads to ineffectiveness of this method.

The problem of computational complexity of the traditional RL methods like policy iteration (PI) and value iteration (VI) (Bellman et al., 1954; 2003; Barto et al., 1983; Bartsekas, 2007) can be overcome by a simulation based approach (Sutton et al., 1998). Here, the crux of the idea is that if the policy or the value function can be parameterized with sufficient accuracy using a small number of parameters, we will be able to transform the optimal control problem to an approximation problem in the parameter space (Bartesekas et al., 1996; Tsitsiklis et al., 2003; Konda et al., 2004) side stepping the need for model knowledge and excessive computations. Furthermore, this formulation renders itself to an online solution approach using RL where the parameters are adjusted on the-fly using input-output data. However, the convergence requires sufficient exploration of the state-action space and the optimality of the obtained policy depends primarily on the accuracy of the parameterization scheme. Thus, the formulation of the problem in the RL paradigm requires some insight into the dynamics of the system.

As a result, a good approximation of the value function is of utmost importance to the stability of the closed-loop system and it requires convergence of the unknown parameters to their optimal values. Hence, this sufficient exploration condition manifests itself as a persistence of excitation (PE) condition when RL is implemented online (Mehta et al., 2009; Bhasin et al., 2013; Vrabie, 2010) which is impossible to be guaranteed a priori. Therefore, many researchers have proposed a model-based RL scheme to relax the PE condition which is implemented using a concurrent learning (CL)-based system identifier to simulate experience by evaluating the Bellman error (BE) over unexplored areas of the state space (Chowdhary et al., 2011; 2013; Kamalapurkar et al., 2014).

The fuel optimization problem for the hybrid electric vehicle therefore have been formulated as a fully observed stochastic Markov Decision Process (MDP). Instead of using Trajectory-optimized LQG (T-LQG) or Model Predictive Control (MPC) to provide a sub-optimal solution in the presence of disturbances and noice, we propose a deep reinforcement learning-based optimization strategy using concurrent learning (CL) that uses the state-derivative-action-reward tuples to present a robust optimal solution. A deep NN is used to parameterize the policy and value function. The convergence of the weight estimates of the policy and the value function to their optimal values justifies our claim. The continuous time representation of the problem and its solution that is in sharp distinction to the conventional techniques which mostly use a discrete time formulation of the state and the action spaces is another major issue that has been addressed by the current work. The actor-critic-identifier identifier used for the optimization purpose has been shown in Figure 1. The major contribution of the proposed approch can be therefore be summarized as follows:

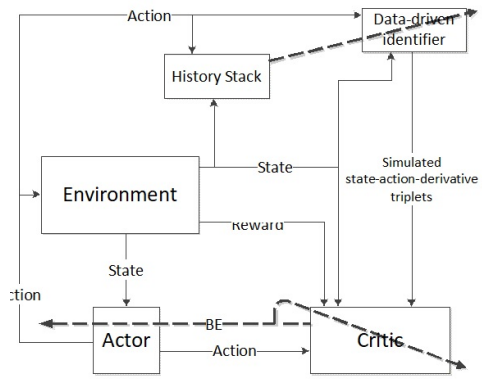

Figure 1: Reinforcement Learning-based Optimization Architecture

1) Compared to the traditional methods of solving the dynamic programming problem which requires simulataneous optimization of the control law as well as the nominal trajectory using an underlying discrete time formulation of the state space, this strategy aims to solve the fuel optimization problem in continuous time frame using a seperated approach where we first solve an open-loop trajectory optimization problem in a model-free manner in continous time and then design a reinforcement learning based control algorithm around the nominal trajectory. This "divide and conquer" strategy is quite useful as computational time for the entire optimization problem is substancially reduced in comparison to the conventional methods. This novel continuous time formulation will also enable the algorithm to be used in most of the practical non-linear stochastic systems, thus paving the way for a much wider range range of applications of RL and at the same time, reducing the computational burden to a significant extent.

2) The popular methods in RL literature including policy iteration and value iteration which suffers from the curse of dimensionality is replaced by a technique where the value function has been parameterized by sufficient number of parameters (PE condition). Therefore, the proposed model-based RL scheme aims to relax the PE condition by using a concurrent learning (CL)-based system identifier to reduce the computational complexity. Concurrent learning (CL)-based system identifiers used for optimal tracking of the uncertain systems often requires the knowledge of the actual controller. An estimate of the true controller designed using the CL-based method generally introduces an approximate estimation error which makes the stability analysis of the system quite intractable. The proposed method will also be able to investigate the stability of the closed-loop system by introducing the estimation error and analyzing the augmented system trajectory obtained under the influnece of the control signal.

3) The proposed control algorithm implemented in fuel management system in hybrid electric vehicles will nullify the limitations of the conventional PID or other control strategies (Model Predictive Control) which suffers from the problems of sub-optimal behaviour in the presence of external disturbances, model-uncertainties, frequent charging and discharging, change of enviroment and other noises. The H-infinity (H_∞) performance index defined as the ratio of the disturbance to the control energy has been established for the RL based optimization technique and compared with the traditional strategies to address the robustness issue of the proposed design scheme.

The rest of the paper is organised as follows: Section 2 presents the problem formulation including the open-loop optimization and reinforcement learning-based controller design in subsections 2.1 and 2.2 respectively. Section 3 provides the simulation results and discussion followed by the conclusion in Section 4.

2 Problem Formulation

Considering the fuel management system of a hybrid electric vehicle as a continous time affine non-linear dynamical system:

$$\begin{aligned} \dot{x} &= f(x, w) + g(x)u, \\ y &= h(x, v) \end{aligned} \tag{1}$$

where, $x \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$, $u \in \mathbb{R}^{n_u}$ are the state, output and the control vectors respectively, *f(.)* denotes the drift dynamics and *g(.)* denotes the control effectivenss matrix. The functions f and h are assumed to be locally Lipschitz continuous functions such that *f(0)* = 0 and $\nabla f(x)$ is continous for every bounded $x \in \mathbb{R}^{n_x}$. The process noise *w* and measurement noise *v* are assumed to be zero-mean, uncorrelated Gausssian white noise with covariances *W* and *V*, respectively.

Assumption 1: We consider the system to be fully observed:

$$y = h(x, v) = x \tag{2}$$

Remark 1: This assumption is considered to provide a tractable formulation of the fuel management problem to side step the need for a complex treatment which is required when a stochastic control problem is treated as partially observed MDP (POMDP).

Optimal Control Problem: For a continous time system with unknown nonlinear dynamics *f(.)*, we need to find an optimal control policy π_t in a finite time horizon $[0, t]$ where π_t is the control policy at time *t* such that π_t = *u(t)* to minimize the cost function given by *J* = $\int_0^t (x^T Q x + u R u^T) dt + x^T F x$ where, *Q,F* > 0 and *R* ≥ 0.

We now solve the open loop optimization problem using a general non-linear programming solver without actually knowing the exact form of the underlying dynamics (Yu et al., 2017). Then,we design a deep reinforcement learning based control strategy using the CL-based system identification technique to obtain the state trajectories that will be able to track the nominal optimal trajectory under the influence of modeling errors and uncertainties (Kamalapurkar et al., 2014).

2.1 Open Loop Optimization

Considering a noise-free non-linear stochastic dynamical system with unknown dynamics:

$$\begin{aligned} \dot{x} &= f(x, 0) + g(x)u, \\ y &= h(x, v) = x \end{aligned} \tag{3}$$

where, $x_0 \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^{n_y}$, $u \in \mathbb{R}^{n_u}$ are the initial state, output and the control vectors respectively, *f(.)* have their usual meanings and the corresponding cost function is given by $J_d\ (x_0,\ u_t)$ = $\int_0^t (x^T Q x + u R u^T) dt + x^T F x$.

The open loop optimization problem is to find the control sequence u_t such that for a given initial state x_0,

$$\begin{aligned} \bar{u}_t &= arg\ min\ J_d(x_0, u_t), \\ subject\ to\ \dot{x} &= f(x, 0) + g(x)u, \\ y &= h(x, v) = x. \end{aligned} \tag{4}$$

The problem is solved using the gradient descent approach (Bryson et al., 1962; Gosavi et al., 2003), and the procedure is illustrated as follows:
Starting from a random initial value of the control sequence $U^{(0)} = [u_t^{(0)}]$ the control policy is updated iteratively as

$$U^{(n+1)} = U^{(n)} - \alpha \nabla_{\boldsymbol{U}} J_d(x_0, U^{(n)}), \tag{5}$$

until the convergence is achieved upto a certain degree of accuracy where $U^{(n)}$ denotes the control value at the nth iteration and α is the step size parameter. The gradient vector is given by:

$$\nabla_{\boldsymbol{U}} J_d(x_0, U^{(n)}) = (\frac{\partial J_d}{\partial u_0}, \frac{\partial J_d}{\partial u_1}, \frac{\partial J_d}{\partial u_2}, ..., \frac{\partial J_d}{\partial u_t})|_{(x_0, u_t)} \tag{6}$$

The ***Gradient Descent Algorithm*** showing the approach has been detailed in the **Appendix A.1**.

Remark 2: The open loop optimization problem is thus solved using the gradient descent approach without considering the exact dynamics of the system. This method proves to be a very simple and useful strategy for implementation in case of complex dynamical systems with complicated cost-to-go functions.

2.2 REINFORCEMENT LEARNING BASED CONTROLLER DESIGN

Considering the affine non-linear dynamical system given by equation (1), our objective is to design a control law to track the optimal time-varying trajectory $\bar{x}(t) \in \mathbb{R}^{n_x}$. A novel cost function is formulated in terms of the tracking error defined by $e = x(t) - \bar{x}(t)$ and the control error defined by the difference between the actual control signal and the desired optimal control signal to overcome the challenge of the infinte cost posed by the cost function when it is defined in terms of the tarcking error *e(t)* and the actual control signal signal *u(t)* only (Zhang et al., 2011; Kamalapurkar et al., 2015).

Assumption 2: (Kamalapurkar et al., 2015) The function *g(x)* in equation (1) is bounded, the matrix *g(x)* has full column rank for all $x(t) \in \mathbb{R}^{n_x}$ and the function $g^+ : \mathbb{R}^n \rightarrow \mathbb{R}^{mXn}$ which is defined as $g^+ = (g^T g)^{-1}$ is bounded and locally Lipschitz.

Assumption 3: (Kamalapurkar et al., 2015) The optimal trajectory is bounded by a known positive constant $b \epsilon \mathbb{R}$ such that $\|\bar{x}\| \leq b$ and there exists a locally Lipschitz function h_d such that $\dot{\bar{x}} = h_d(\bar{x})$ and $g(\bar{x})\ g^+(\bar{x})(h_d(\bar{x}) - f(\bar{x})) = h_d(\bar{x}) - f(\bar{x})$.

Using the *Assumption 2* and *Assumption 3*, the control signal u_d required to track the desired trajectory $\bar{x}(t)$ is given as $u_d(\bar{x}) = g_d^+(h_d(\bar{x}) - f_d)$ where $f_d = f(\bar{x})$ and $g_d^+ = g^+(\bar{x})$. The control error is given by $\mu = u(t) - u_d(\bar{x})$. The system dynamics can now be expressed as

$$\dot{\zeta} = F(\zeta) + G(\zeta)\mu \tag{7}$$

where, the merged state $\zeta(t) \in \mathbb{R}^{2n}$ is given by $\zeta(t) = [e^T, \bar{x}^T]^T$ and the functions $F(\zeta)$ and $G(\zeta)$ are defined as $F(\zeta) = [f^T(e+\bar{x}) - h_d^T + u_d^T(\bar{x})g^T(e+\bar{x}), h_d^T]^T$ and $G(\zeta) = [g^T(e+\bar{x}), \mathbf{0}_{mXn}]^T$ where, $\mathbf{0}_{mXn}$ denotes a matrix of zeroes. The control objective is to solve a finite-horizon optimal tracking problem online, i.e., to design a control signal μ that will minimize the cost-to-go function, while tracking the desired trajectory, is given by $J(\zeta, \mu) = \int_0^t r(\zeta(\tau), \mu(\tau))d\tau$ where, the local cost $r : \mathbb{R}^{2n} X R^m \rightarrow \mathbb{R}$ is given as $r(\zeta, \tau) = Q(e) + \mu^T R\mu$, $R \epsilon \mathbb{R}^{mXm}$ is a positive definite symmetric matrix and $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continous positive definite function.

Based on the assumption of the existence of an optimal policy, it can be characterized in terms of the value function $V^* : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ which is defined as $V^*(\zeta) = min_{\mu(\tau)\epsilon U|\tau\epsilon\mathbb{R}_{t\geq 0}} \int_0^t r(\phi^u(\pi, t, \zeta), \mu(\tau))d\tau$, where $U \in \mathbb{R}^m$ is the action space and $\phi^u(t; t_0, \zeta_0)$ is the trajectory of the system defined by equation (10) with the control effort $\mu : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ with the initial condition $\zeta_0 \in \mathbb{R}^{2n}$ and the initial time $t_0 \in \mathbb{R}_{\geq 0}$. Taking into consideration that an optimal policy exists and that V^* is continously differentiable everywhere, the closed-form solution (Kirk, 2004) is given as $\mu^*(\zeta) = -1/2\ R^{-1}G^T(\zeta)(\nabla_\zeta V^*(\zeta))^T$ where, $\nabla_\zeta(.) = \frac{\partial(.)}{\partial x}$. This satisfies the Hamilton-Jacobi-Bellman (HJB) equation (Kirk, 2004) given as

$$\nabla_\zeta V^*(\zeta)(F(\zeta) + G(\zeta)\mu^*(\zeta)) + \bar{Q}(\zeta) + \mu^{*T}(\zeta)R\mu^*(\zeta) = 0 \tag{8}$$

where, the initial condition $V^* = 0$, and the funtion $\bar{Q} : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ is defined as $\bar{Q}([e^T, \hat{x}^T]^T) = Q(e)$ where, $(e(t), \hat{x}(t)) \in \mathbb{R}^n$. Now, we use an actor-critic based method to obtain the parametric estimates of the optimal value function and the optimal policy given as $\hat{V}(\zeta, \hat{W}_c)$ and $\hat{\mu}(\zeta, \hat{W}_a)$ where, $\hat{W}_c \in \mathbb{R}^L$ and $\hat{W}_a \in \mathbb{R}^L$ define the vector paramater estimates as the analytical solution of the HJB equation is not feasible. The task of the actor and critic is to learn the corresponding parameters. Replacing the estimates $\hat{V}$ and $\hat{\mu}$ for V^* and $\hat{\mu}^*$ in the HJB equation, we obtain the residual error, also known as the Bell Error (BE) as $\delta(\zeta, \hat{W}_c, \hat{W}_a) = \bar{Q}(\zeta) + \hat{\mu}^T(\zeta, \hat{W}_a)R\hat{\mu}(\zeta, \hat{W}_a) + \nabla_\zeta \hat{V}(\zeta, \hat{W}_c)(F(\zeta) + G(\zeta)\hat{\mu}(\zeta, \hat{W}_a))$ where, $\delta : \mathbb{R}^{2n}$ X $\mathbb{R}^L$ X $\mathbb{R}^L \rightarrow \mathbb{R}$. The solution of the problem requires the actor and the critic to find a set of parameters $\hat{W}_a$ and $\hat{W}_c$ respectively such that $\delta(\zeta, \hat{W}_c, \hat{W}_a) = 0$ and $\hat{\mu}^T(\zeta, \hat{W}_a) = -1/2\ R^{-1}G^T(\zeta)(\nabla_\zeta V^*(\zeta))^T$ where, $\forall\zeta \in \mathbb{R}^n$. As the exact basis fucntion for the approximation is not known apriori, we seek to find a set of approximate parameters that minimizes the BE. However, an uniform approximation of the value function and the optimal control policy over the entire operating domain requires to find parameters that will able to minimize the error $E_s : \mathbb{R}^L$ X $\mathbb{R}^L \rightarrow \mathbb{R}$ defined as $E_s(\hat{W}_c, \hat{W}_a) = sup_\zeta(|\delta, \hat{W}_c, \hat{W}_a|)$ thus, making it necessary

to have an exact knowledge of the system model. Two of the most popular methods used to render the design of the control strategy robust to system uncertainties in this context are integral RL (Lewis et al., 2012; Modares et al., 2014) and state derivative estimation (Bhasin et al., 2013; Kamalapurkar et al., 2014). Both of these methods suffer from the persistence of exitation(PE) condition that requires the state trajectory $\phi^{\hat{u}}(t; t_0, \zeta_0)$ to cover the entire operating domain for the convergence of the parameters to their optimal values. This condition is slightly relaxed in (Modares et al., 2014) where the integral technique is used in augmentation with the replay of the experience where every evaluation of the BE is intuitively formalized as a gained experience, and these experiences are kept in a history stack so that they can be iteratively used by the learning algorithm to improve data efficiency. Kamalapurkar et al. (2014) has designed a dynamic system identifier which is used to model the parametric estimate $\hat{F}(\zeta, \hat{\theta})$ of the system drift dynamics where, $\hat{\theta}$ is used to indicate the estimate of the matrix of unknown parameters. Here, the system identifier is used to simulate the experience by extrapolating the Bell Error (BE) over the unexplored territory in the operating domain.Thus, to relax the PE condition, the CL-based system identifier (Chowdhary et al., 2011; Kamalapurkar et al., 2014) prompts an exponential convergence of the parameters to their optimal values making it possible for the learning laws of the optimization technique to use the simulated experience in concurrence with the experience gained and kept along the actual system trajectory.

2.2.1 Parametric System Identification

Defined by any compact set $C \subset \mathbb{R}$, the function *f* can be defined using a neural network (NN) as *f(x)* = $\theta^T \sigma_f(Y^T x_1) + \epsilon_0(x)$ where, $x_1 = [1, x^T]^T \in \mathbb{R}^{n+1}$, $\theta \in \mathbb{R}^{n+1Xp}$ and $Y \in \mathbb{R}^{n+1Xp}$ indicates the constant unknown output-layer and hidden-layer NN weight, $\sigma_f : \mathbb{R}^p \to \mathbb{R}^{p+1}$ denotes a bounded NN activation function, ϵ_θ: $\mathbb{R}^n \to \mathbb{R}^n$is the function reconstruction error, $p \in \mathbb{N}$ denotes the number of NN neurons. Using the universal functionional approximation property of single layer NNs, given a constant matrix Y such that the rows of $\sigma_f(Y^T x_1)$ form a proper basis, there exist constant ideal weights θ and known constants $\bar{\theta}$, $\bar{\epsilon}_\theta$, $\bar{\epsilon}'_\theta \in \mathbb{R}$ such that $||\boldsymbol{\theta}|| \leq \bar{\theta} < \infty$, $sup_{x \epsilon C}$ $||\epsilon_\theta(x)|| \leq \bar{\epsilon}_\theta$, $sup_{x \epsilon C}$ $||\nabla_{\boldsymbol{x}} \epsilon_\theta(x)|| \leq \bar{\epsilon}_\theta$ where, $||.||$ denotes the Euclidean norm for vectors and the Frobenius norm for matrix (Lewis et al., 1998).
Taking into consideration an estimate $\hat{\theta} \in \mathbb{R}^{p+1Xn}$ of the weight matrix θ , the function *f* can be approximated by the function $\hat{f} : \mathbb{R}^{2n}$ X $\mathbb{R}^{p+1Xn} \to \mathbb{R}^n$ which is defined as $\hat{f}(\zeta, \hat{\theta})$ = $\hat{\theta}^T \sigma_\theta(\zeta)$, where $\sigma_\theta : \mathbb{R}^{2n} \to \mathbb{R}^{p+1}$ can be defined as $\sigma_\theta(\zeta)$ = $\sigma_f(Y^T[1, e^T + \bar{x}^T]^T)$. An estimator for online identification of the drift dynamics is developed

$$\dot{\hat{x}} = \hat{\theta}^T \sigma_\theta(\zeta) + g(x)u + k\tilde{x} \tag{9}$$

where, $\tilde{x} = x - \hat{x}$ and $k \in \mathbb{R}$ R is a positive constant learning gain.

The BE is now approximated as

$$\begin{aligned} \hat{\delta}\left(\zeta, \hat{\theta}, \hat{W}_c, \hat{W}_a\right) = \bar{Q}(\zeta) + \hat{\mu}^T\left(\zeta, \hat{W}_a\right) R\hat{\mu}\left(\zeta, \hat{W}_a\right) \\ +\nabla_\zeta \hat{V}\left(\zeta, \hat{W}_a\right)\left(F_\theta(\zeta, \hat{\theta}) + F_1(\zeta) + G(\zeta)\hat{\mu}\left(\zeta, \hat{W}_a\right)\right) \end{aligned} \tag{10}$$

In equation (10), $F_\theta(\zeta, \hat{\theta}) = \begin{bmatrix} \hat{\theta}^T \sigma_\theta(\zeta) - g(x) g^+(x_d) \hat{\theta}^T \sigma_\theta\left(\begin{bmatrix} \mathbf{0}_{n\times 1} \\ x_d \end{bmatrix}\right) \\ \mathbf{0}_{n\times 1} \end{bmatrix}$, and $F_1(\zeta) = \left[\left(-h_d + g(e + x_d) g^+(x_d) h_d\right)^T, h_d^T\right]^T$. Consult **Appendix A.2** for more details.

2.2.2 Value Function Approximation

As V^* and μ^* are functions of the state ζ, the optimization problem as defined in Section 2.2 is quite an intractable one, so the optimal value function is now represented as $\mathcal{C} \subset \mathbb{R}^{2n}$ using a NN as $V^*(\zeta) = W^T \sigma(\zeta) + \epsilon(\zeta)$, where $W \in \mathbb{R}^L$ denotes a vector of unknown NN weights, $\sigma : \mathbb{R}^{2n} \to \mathbb{R}^L$ indicates a bounded NN activation function, $\epsilon : \mathbb{R}^{2n} \to \mathbb{R}$ defines the function reconstruction error, and $L \in \mathbb{N}$ denotes the number of NN neurons. Considering the universal function approximation property of single layer NNs, for any compact set $\mathcal{C} \subset \mathbb{R}^{2n}$, there exist constant ideal weights W and known positive constants $\bar{W}, \bar{\epsilon}$, and $\bar{\epsilon}' \in \mathbb{R}$ such that $\|W\| \leq \bar{W} < \infty$ $\sup_{\zeta \in \mathcal{C}} \|\epsilon(\zeta)\| \leq \bar{\epsilon}$, and $\sup_{\zeta \in \mathcal{C}} \|\nabla_\zeta \epsilon(\zeta)\| \leq \bar{\epsilon}'$ (Lewis et al., 1998). A NN representation of the optimal policy is

obtained as $\mu^*(\zeta) = -\frac{1}{2}R^{-1}G^T(\zeta)\left(\nabla_\zeta\sigma^T(\zeta)W + \nabla_\zeta\epsilon^T(\zeta)\right)$. Taking the estimates $\hat{W}_c$ and $\hat{W}_a$ for the ideal weights W, the optimal value function and the optimal policy are approximated as $\hat{V}\left(\zeta, \hat{W}_c\right) = \hat{W}_c^T\sigma(\zeta)$, $\hat{\mu}\left(\zeta, \hat{W}_a\right) = -\frac{1}{2}R^{-1}G^T(\zeta)\nabla_\zeta\sigma^T(\zeta)\hat{W}_a$. The optimal control problem is therefore recast as to find a set of weights $\hat{W}_c$ and $\hat{W}_a$ online to minimize the error $\hat{E}_{\hat{\theta}}\left(\hat{W}_c, \hat{W}_a\right) = \sup_{\zeta\in\chi}\left|\hat{\delta}\left(\zeta, \hat{\theta}, \hat{W}_c, \hat{W}_a\right)\right|$ for a given $\hat{\theta}$, while simultaneously improving $\hat{\theta}$ using the CL-based update law and ensuring stability of the system using the control law $u = \hat{\mu}\left(\zeta, \hat{W}_a\right) + \hat{u}_d(\zeta, \hat{\theta})$ where, $\hat{u}_d(\zeta, \hat{\theta}) = g_d^+\left(h_d - \hat{\theta}^T\sigma_{\theta d}\right)$, and $\sigma_{\theta d} = \sigma_\theta\left(\begin{bmatrix} \mathbf{0}_{1\times n} & x_d^T \end{bmatrix}^T\right)$.

2.2.3 Experience Simulation

The simulation of experience is implemented by minimizing a squared sum of BEs over finitely many points in the state space domain as the calculation of the extremum (supremum) in $\hat{E}_{\hat{\theta}}$ is not tractable. See assumption 5 in **Appendix A.3** for the detailed explaination of the approximation.

2.2.4 Stability Analysis

To perform the stability analysis, we take the non-autonomous form of the value function (Kamalapurkar et al., 2015) defined by $V_t^* : \mathbb{R}^n \mathrm{X} \mathbb{R} \rightarrow \mathbb{R}$ which is defined as $V_t^*(e, t) = V^*\left(\left[e^T, x_d^T(t)\right]^T\right), \forall e \in \mathbb{R}^n, t \in \mathbb{R}$, is positive definite and decrescent. Now, $V_t^*(0, t) = 0, \forall t \in \mathbb{R}$ and there exist class $\mathcal{K}$ functions $\underline{v} : \mathbb{R} \rightarrow \mathbb{R}$ and $\bar{v} : \mathbb{R} \rightarrow \mathbb{R}$ such that $\underline{v}(\|e\|) \leq V_t^*(e, t) \leq \bar{v}(\|e\|)$, for all $e \in \mathbb{R}^n$ and for all $t \in \mathbb{R}$. We take an augemented state given as $Z \in \mathbb{R}^{2n+2L+n(p+1)}$ is defined as $Z = \left[\ e^T,\ \tilde{W}_c^T,\ \tilde{W}_a^T,\ \tilde{x}^T,\ (\mathrm{vec}(\tilde{\theta}))^T\right]^T$ and a candidate Lyapunov function is defined as $V_L(Z, t) = V_t^*(e, t) + \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T\tilde{W}_a\frac{1}{2}\tilde{x}^T\tilde{x} + \frac{1}{2}\operatorname{tr}\left(\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}\right)$ where, vec $(\cdot)$ denotes the vectorization operator. From the weight update for the simulation of experience in Section 2.2.3 we get positive constants $\underline{\gamma}, \bar{\gamma} \in \mathbb{R}$ such that $\underline{\gamma} \leq \left\|\Gamma^{-1}(t)\right\| \leq \bar{\gamma}, \forall t \in \mathbb{R}$. Taking the bounds on Γ and V_t^* and the fact that $\operatorname{tr}\left(\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}\right) = (\mathrm{vec}(\tilde{\theta}))^T\left(\Gamma_\theta^{-1} \otimes \mathbb{I}_{p+1}\right)(\mathrm{vec}(\tilde{\theta}))$ the candidate Lyapunov function be bounded as $\underline{v}_l(\|Z\|) \leq V_L(Z, t) \leq \bar{v}_l(\|Z\|)$ for all $Z \in \mathbb{R}^{2n+2L+n(p+1)}$ and for all $t \in \mathbb{R}$, where $v_l : \mathbb{R} \rightarrow \mathbb{R}$ and $\overline{v_l} : \mathbb{R} \rightarrow \mathbb{R}$ are class $\mathcal{K}$ functions. The corresponding theorem and the proof has been explained in detail in **Appendix A.4**.

3 Simulation Results and Discussion

Here, we are going to present the simulation results to demonstrate the performance of the proposed method with the fuel management system of the hybrid electric vehicle. We consider a two dimensional non-linear model given by

$$f = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1 & x_2(1-(cos(2x_1+2)^2)) \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix},\ g = \begin{bmatrix} 0 \\ cos(2x_1+2) \end{bmatrix} \tag{11}$$

where $a, b, c, d \in \mathbb{R}$ are unknown positive parameters whose values are selected as $a = -1$, $b = 1$, $c = -0.5$, $d = -0.5$ and x_1 and x_2 are the two states of the hybrid electric vehicle given by the charge present in the battery and the amount of fuel in the car, respectively. The control objective is to minimize the cost function given by $J(\zeta, \mu) = \int_0^t r(\zeta(\tau), \mu(\tau))d\tau$ where, the local cost $r : \mathbb{R}^{2n}XR^m \rightarrow \mathbb{R}$ is given as $r(\zeta, \tau) = Q(e) + \mu^T R\mu$, $R\,\epsilon\,\mathbb{R}^{mXm}$ is a positive definite symmetric matrix and $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continous positive definite function, while following the desired trajectory $\bar{x}$ We chhose $Q = I_{2x2}$ and $R = 1$. The optimal value function and optimal control for the system (15) are $V^*(x) = \dfrac{1}{2}x_1^2 + \dfrac{1}{2}x_2^2$ and $u^*(x) = -cos(2(x_1)+2)x_2$. The basis function $\sigma : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ for value function approximation is $\sigma = [x_1^2,\ x_1^2x_2^2,\ x_2^2]$. The ideal weights are

$W = [0.5, 0, 1]$. The initial value of the policy and the value function weight estimates are $\hat{W}_c$ = $\hat{W}_a$ = $[1, 1, 1]^T$, least square gain is $\Gamma(0)$ = $100 I_{3X3}$ and that of the system states are $x(0)$ = $[-1, -1]^T$. The state estimates $\hat{x}$ and $\hat{\theta}$ are initialized to 0 and 1 respectively while the history stack for the CL is updated online. Here, Figure 2 and Figure 3 shows the state trajectories obtained by

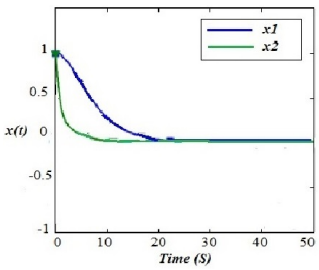

Figure 2: State Trajectories

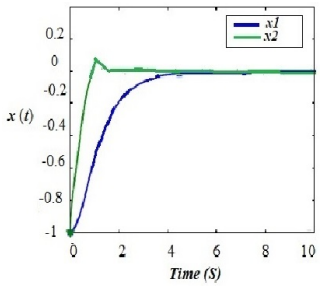

Figure 3: State Trajectories

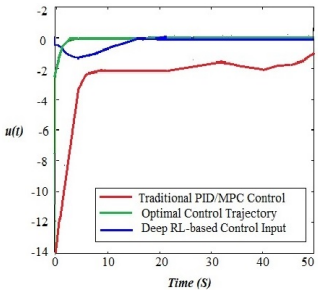

Figure 4: Control Input

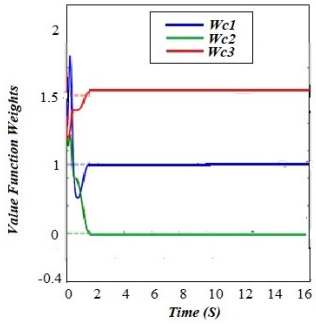

Figure 5: Value Function

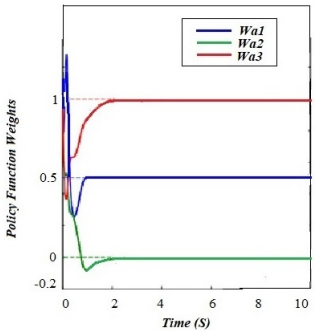

Figure 6: Policy Function

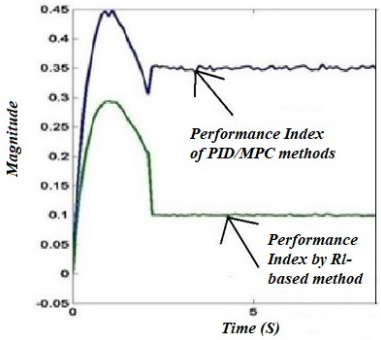

Figure 7: Performance Index

the traditional methods and that obtained by the RL-based optimization technique respectively in the presence of disturbances. It can be stated that settling time of trajectories obtained by the proposed method is significantly less as compared with that of the conventional PID strategies thus justifying the uniqueness of the method. Figure 4 shows the corresponding control inputs whereas Figure 5 and Figure 6 indicates the convergence of the NN weight functions to their optimal values. The H_∞ performance index in Figure 7 shows a value of 0.3 for the RL-based method in comparison to 0.45 for the traditional control design which clearly establishes the robustness of our proposed design.

4 Conclusion

In this paper, we have proposed a robust deep Reinforcement Learning-based optimization strategy for hybrid electric vehicles. We have solved an open-loop deterministic trajectory optimization problem for a fully observed non-linear system with unknown dynamics, followed by the design of a deep RL-based robust optimal controller. The simulation results validate the efficacy of the method over the traditional PID control-based optimization techniques. The offline learning procedure is quite simple and its online execution is fast. Future work will generalize the approach for large-scale partially observed uncertain systems.

References

R. Akrour, A. Abdolmaleki, H. Abdulsamad, and G. Neumann. Model Free Trajectory Optimization for Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

A. Barto, R. Sutton, and C. Anderson. Neuron-like adaptive elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man, and Cybernetics*, 13: 834-846, 1983.

R. Bellman. The theory of dynamic programming. *DTIC Document, Technical Representations*. 1954.

D. Bertsekas and J. Tsitsiklis. Neuro-Dynamic Programming. *Athena Scientific*, 1996.

D. Bertsekas. Dynamic Programming and Optimal Control. *Athena Scientific*, 2007.

S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. L. Lewis, and W. Dixon. A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. *Automatica*, 49: 89-92, 2013.

R. P. Bithmead, V. Wertz, and M. Gerers. Adaptive Optimal Control: The Thinking Man's G.P.C. *Prentice Hall Professional Technical Reference*, 1991.

A. Bryson and H. Y.-C. Applied Optimal Control: Optimization, Estimation and Control. *Washington: Hemisphere Publication Corporation*, 1975.

G. V. Chowdhary and E. N. Johnson. Theory and flight-test validation of a concurrent-learning adaptive controller. *Journal of Guidance Control and Dynamics*,34:, 592-607, 2011.

G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *International Journal of Adaptive Control and Signal Processing*, 27: 280-301, 2013

P. García, J. P. Torreglosa, L. M. Fernández, and F. Jurado. Viability study of a FC-batterySC tramway controlled by equivalent consumption minimization strategy. *International Journal of Hydrogen Energy*, 37: 9368-9382, 2012.

A. Gosavi. Simulation-based optimization: Parametric optimization techniques and reinforcement learning. *Kluwer Academic Publishers*, Norwell , MA, USA, 2003.

J. Han, and Y. Park. A novel updating method of equivalent factor in ECMS for prolonging the lifetime of battery in fuel cell hybrid electric vehicle. In *IFAC Proceedings*, volume 45(4), pp. 227-232, 2012.

J. Han, J.F. Charpentier, and T. Tang. An Energy Management System of a Fuel Cell/Battery Hybrid Boat. *Energies*, 7: 2799-2820, 2014.

J. Han, Y. Park, and D. Kum. Optimal adaptation of equivalent factor of equivalent consumption minimization strategy for fuel cell hybrid electric vehicles under active state inequality constraints. *Journal of Power Sources*, 267: 491–502, 2014.

D. Jacobsen and D. Mayne. Differential Dynamic Programming. *Elsevier*, 1970.

R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon. Model-based reinforcement learning for infinite-horizon approximate optimal tracking. *In Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 5083-5088, 2014.

R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon. Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica*, 51: 40-48, 2015.

S. G. Khan et al. Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*, 36: 42-59, 2012.

M.J. Kim and H. Peng. Power management and design optimization of fuel cell/battery hybrid vehicles. *Journal of Power Sources*, 165: 819-832, 2007.

D. Kirk. Optimal Control Theory: An Introduction. *Mineola*, NY, Dover, 2004.

V. Konda and J. Tsitsiklis. On actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42: 1143-1166, 2004.

S. Levine and P. Abbeel. Learning Neural Network Policies with Guided Search under Unknown Dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

F. L. Lewis, S. Jagannathan, and A. Yesildirak. Neural network control of robot manipulators and nonlinear systems. *CRC Press*, Philadelphia, PA, 1998.

F. L. Lewis, D. Vrabie, and V. L. Syrmos. Optimal Control, 3rd edition. *Wiley*, NJ, 2012.

H. Li, A. Ravey, A. N'Diaye, and A. Djerdir. A Review of Energy Management Strategy for Fuel Cell Hybrid Electric Vehicle. In *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1-6, 2017.

W.S. Lin and C.H. Zheng. Energy management of a fuel cell/ultracapacitor hybrid power system using an adaptive optimal control method. *Journal of Power Sources*, 196(6): 3280- 3289, 2011.

P. Mehta and S. Meyn, "Q-learning and pontryagin's minimum principle. *In Proceedings of IEEE Conference on Decision and Control*, pp. 3598-3605, 2009.

D. Mitrovic, S. Klanke, and S. Vijayakumar. Adaptive Optimal Feedback Control with Learned Internal Dynamics Models. *Springer*, pp. 65-84, Berlin, 2010.

H. Modares and F. L. Lewis. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica*, 50(7): pp. 1780-1792, 2014.

S. J. Moura, D. S. Callaway, H. K. Fathy, and J. L. Stein. Tradeoffs between battery energy capacity and stochastic optimal power management in plug-in hybrid electric vehicles. *Journal of Power Sources*, 1959(9): pp. 2979-2988, 2010.

S. N. Motapon, L. Dessaint, and K. Al-Haddad. A Comparative Study of Energy Management Schemes for a Fuel-Cell Hybrid Emergency Power System of More-Electric Aircraft. *IEEE Transactions on Industrial Electronics*, 61: pp. 1320-1334, 2014.

G. Paganelli, S. Delprat, T. M. Guerra, J. Rimaux, and J. J. Santin. Equivalent consumption minimization strategy for parallel hybrid powertrains. In *IEEE 55th Vehicular Technology Conference, VTC Spring 2002 (Cat. No.02CH37367)*, pp. 2076-2081, 2002.

F. Segura and J. M. Andújar. Power management based on sliding control applied to fuel cell systems: A further step towards the hybrid control concept. *Applied Energy*, 99: 213-225, 2012.

R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. *MIT Press*, Cambridge, MA, USA, 1998.

E. Theoddorou, Y. Tassa, and E. Todorov. Stochastic Differential Dynamic Programming. In *Proceedings of American Control Conference*, 2010.

E. Todorov and Y. Tassa. Iterative Local Dynamic Programming. In *Proceedings of the IEEE International Symposium on ADP and RL*, 2009.

J. P. Torreglosa, P. García, L. M. Fernández, and F. Jurado. Predictive Control for the Energy Management of a Fuel-Cell-Battery-Supercapacitor Tramway. *IEEE Transactions on Industrial Informatics*, 10(1): 276-285, 2014.

D. Vrabie. Online adaptive optimal control for continuous-time systems. *Ph.D. dissertation*, University of Texas at Arlington, 2010.

Dan Yu, Mohammadhussein Rafieisakhaei, and Suman Chakravorty. Stochastic Feedback Control of Systems with Unknown Nonlinear Dynamics. In *IEEE Conference on Decision and Control*, 2017.

M. K. Zadeh. Stability Analysis Methods and Tools for Power Electronics-Based DC Distribution Systems: Applicable to On-Board Electric Power Systems and Smart Microgrids. *NTNU*, 2016.

M. K. Zadeh, L. Saublet, R. Gavagsaz-Ghoachani, B. Nahid Mobarakeh, S. Pierfederici, and M. Molinas. Energy management and stabilization of a hybrid DC microgrid for transportation applications. In *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 3397-3402, 2016.

X. Zhang, C. C. Mi, A. Masrur, and D. Daniszewski. Wavelet transform-based power management of hybrid vehicles with multiple on-board energy sources including fuel cell, battery and ultracapacitor. *Journal of Power Sources*, 185(2): 1533-1543, 2008.

C. Zheng, S. W. Cha, Y. Park, W. S. Lim, and G. Xu. PMP-based power management strategy of fuel cell hybrid vehicles considering multi-objective optimization. *International Journal Precision Engineering and Manufacturing*, 14(5): 845-853, 2013.

C. H. Zheng, G. Q. Xu, Y. I. Park, W. S. Lim, and S. W. Cha. Prolonging fuel cell stack lifetime based on Pontryagin's Minimum Principle in fuel cell hybrid vehicles and its economic influence evaluation. *Journal Power Sources*, 248: 533-544, 2014.

X. Zhong, H. He, H. Zhang, and Z. Wang. Optimal Control for Unknown Diiscrete-Time Nonlinear Markov Jump Systems Using Adaptive Dynamic Programming. *IEEE Transactions on Neural networks and learning systems*, 25(12): pp. 2141-2155, 2014.

A APPENDIX

A.1 THE GRADIENT DESCENT ALGORITHM

The Gradient Descent Algorithm has been explained as follows:

Algorithm; Gradient Descent

Input : Design Parameters $U^{(0)} = u_t^0,\ \alpha, h\ \epsilon\ \mathbb{R}$

Output : Optimal control sequence $\{\bar{u}_t\}$

1. $n \leftarrow 0, \nabla_U J_d\left(x_0, U^{(0)}\right) \leftarrow \epsilon$

2. **while** $\nabla_U J_d\left(x_0, U^{(n)}\right) \geq \epsilon$ **do**

3. Evaluate the cost function with control $U^{(n)}$

4. Perturb each control variable $u_i^{(n)}$ by h, $i = 0, \cdots, t$, and calculate the gradient vector $\nabla_U J_d\left(x_0, U^{(n)}\right)$ using (7) and (8)

5. Update the control policy: $U^{(n+1)} \leftarrow U^{(n)} - \alpha \nabla_U J_d\left(x_0, U^{(n)}\right)$

6. $n \leftarrow n+1$

7. **end**

8. $\{\bar{u}_t\} \leftarrow U^{(n)}$

A.2 ANALYSIS OF PARAMETRIC SYSTEM IDENTIFICATION

Assumption 4: (Chowdhary et al., 2013) A history stack containing recorded state-action pairs $\{x_j, u_j\}_{j=1}^M$ 1 along with numerically computed state derivatives $\{\dot{\bar{x}}_j\}_{j=1}^M$ 1 that satisfies $\lambda_{\min}\left(\sum_{j=1}^M \sigma_{fj}\sigma_{fj}^T\right) = \underline{\sigma_\theta} \leq 0, \quad \|\dot{\bar{x}}_j - \dot{x}_j\| < \bar{d},\ \forall j$ is available a priori, where $\sigma_{fj} = \sigma_f\left(Y^T \left[\begin{array}{cc} 1, & x_j^T \end{array}\right]^T\right)$, $\bar{d} \in \mathbb{R}$ is a known positive constant, $\dot{x}_j = f\left(x_j\right) + g\left(x_j\right) u_j$, and $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue. The weight estimates $\hat{\theta}$ are updated according to the following CL-based update law: $\dot{\hat{\theta}} = \Gamma_\theta \sigma_f\left(Y^T x_1\right)\tilde{x}^T + k_\theta \Gamma_\theta \sum_{j=1}^M \sigma_{fj}\left(\dot{\bar{x}}_j - g_j u_j - \hat{\theta}^T \sigma_{fj}\right)^T$ where $k_\theta \in \mathbb{R}$ is a constant positive CL gain, and $\Gamma_\theta \in \mathbb{R}^{p+1\times p+1}$ is a constant, diagonal, and positive definite adaptation gain matrix.

A.3 ANALYSIS OF EXPERIENCE SIMULATION

Assumption 5: (Kamalapurkar et al., 2014) There exists a finite set of points $\{\zeta_i \in \mathcal{C} \mid i = 1, \cdots, N\}$ and a constant $\underline{c} \in \mathbb{R}$ such that $0 < \underline{c} = \frac{1}{N}\left(\inf_{t\in\mathbb{R}_{\geq t_0}}\left(\lambda_{\min}\left\{\sum_{i=1}^N \frac{\omega_i \omega_i^T}{\rho_i}\right\}\right)\right)$ where ρ_i $=1 + \nu\omega_i^T \Gamma \omega_i \in \mathbb{R}$, and $\omega_i = \nabla_\zeta \sigma\left(\zeta_i\right)\left(F_\theta\left(\zeta_i, \hat{\theta}\right) + F_1\left(\zeta_i\right) + G\left(\zeta_i\right)\hat{\mu}\left(\zeta_i, \hat{W}_a\right)\right)$.
Using *Assumption 5*, simulation of experience is implemented by the weight update laws given by $\dot{\hat{W}}_c = -\eta_{c1}\Gamma\frac{\omega}{\rho}\hat{\delta}_t - \frac{\eta_{c2}}{N}\Gamma\sum_{i=1}^N \frac{\omega_i}{\rho_i}\hat{\delta}_{ti}$ and $\dot{\Gamma} = \left(\beta\Gamma - \eta_{c1}\Gamma\frac{\omega\omega^T}{\rho^2}\Gamma\right)\mathbf{1}_{\{\|\Gamma\|\leq\bar{\Gamma}\}}, \|\Gamma\left(t_0\right)\| \leq$

$\bar{\Gamma}$, $\dot{\hat{W}}_a = -\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a + \left(\frac{\eta_{c1} G_\sigma^T \hat{W}_a \omega^T}{4\rho} + \sum_{i=1}^{N} \frac{\eta_{c2} G_{\sigma i}^T \hat{W}_a \omega_i^T}{4N\rho_i}\right)\hat{W}_c$ where, $\omega = \nabla_\zeta \sigma(\zeta)\left(F_\theta(\zeta,\hat{\theta}) + F_1(\zeta) + G(\zeta)\hat{\mu}\left(\zeta,\hat{W}_a\right)\right)$, $\Gamma \in \mathbb{R}^{L\times L}$ is the least-squares gain matrix, $\bar{\Gamma} \in \mathbb{R}$ denotes a positive saturation constant, $\beta \in \mathbb{R}$ indicates a constant forgetting factor, $\eta_{c1}, \eta_{c2}, \eta_{a1}, \eta_{a2} \in \mathbb{R}$ defines constant positive adaptation gains, $1_{\{\cdot\}}$ denotes the indicator function of the set $\{\cdot\}$, $G_\sigma = \nabla_\zeta \sigma(\zeta) G(\zeta) R^{-1} G^T(\zeta) \nabla_\zeta \sigma^T(\zeta)$, and $\rho = 1 + \nu\omega^T\Gamma\omega$, where $\nu \in \mathbb{R}$ is a positive normalization constant. In the above weight update laws, for any function $\xi(\zeta,\cdot)$, the notation ξ_i, is defined as $\xi_i = \xi(\zeta_i,\cdot)$, and the instantaneous BEs $\hat{\delta}_t$ and $\hat{\delta}_{ti}$ are given as $\hat{\delta}_t = \hat{\delta}\left(\zeta,\hat{W}_c,\hat{W}_a,\hat{\theta}\right)$ and $\hat{\delta}_{ti} = \hat{\delta}\left(\zeta_i,\hat{W}_c,\hat{W}_a,\hat{\theta}\right)$.

A.4 Stability Analysis

Theorem

Given the Assumptions $3-5$ hold, and the number of NN neurons, and the minimum singular values $\underline{c}$ and $\underline{\sigma}_\theta$ are large enough [44], the proposed controller u along with the weight update laws $\hat{W}_c$, $\dot{\Gamma}$ and $\dot{\hat{W}}_a$ and the identifier (10) along with the weight update laws (11) ensure that the system states remain bounded, the tracking error is ultimately bounded, and that the control policy $\hat{\mu}$ converges to a neighborhood around the optimal control policy μ^*.

Proof: Using (1) and the fact that $V_t^*(e(t),t) = \dot{V}^*(\zeta(t)), \forall t \in \mathbb{R}$, the time-derivative of the candidate Lyapunov function is given by

$$\begin{aligned}\dot{V}_L = \nabla_\zeta V^*\left(F + G\mu^*\right) - \tilde{W}_c^T \Gamma^{-1}\dot{\hat{W}}_c - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c \\ -\tilde{W}_a^T\dot{\hat{W}}_a + \dot{V}_0 + \nabla_\zeta V^* G\mu - \nabla_\zeta V^* G\mu^*\end{aligned} \tag{12}$$

Under sufficient gain conditions [45], using (9), (10)-(11), and the update laws given $\hat{W}_c$, $\dot{\Gamma}$ and $\hat{W}_a$ the time-derivative of the candidate Lyapunov function can be bounded as

$$\dot{V}_L \leq -v_l(\|Z\|), \forall \|Z\| \geq v_l^{-1}(\iota), \forall Z \in \chi \tag{13}$$

where ι is a positive constant, and $\chi \subset \mathbb{R}^{2n+2L+n(p+1)}$ is a compact set. Considering (13) and (15), the theorem 4.18 in [46] can be used to establish that every trajectory $Z(t)$ satisfying $\|Z(t_0)\| \leq \overline{v_l}^{-1}\left(\underline{v}_l(\rho)\right)$, where ρ is a positive constant, is bounded for all $t \in \mathbb{R}$ and satisfies $\limsup_{t\to\infty}\|Z(t)\| \leq \underline{v_l}^{-1}\left(\overline{v_l}\left(v_l^{-1}(\iota)\right)\right)$ [45].