OPTIMAL ENERGY MANAGEMENT FOR A HYBRID VEHICLE USING NEURO-DYNAMIC PROGRAMMING TO CONSIDER TRANSIENT ENGINE OPERATION

Rajit Johri
Mechanical Engineering,
University of Michigan
Ann Arbor, Michigan, USA 48109
rajit@umich.edu

Ashwin Salvi
Mechanical Engineering,
University of Michigan
Ann Arbor, Michigan, USA 48109
asalvi@umich.edu

Zoran Filipi*
Mechanical Engineering,
University of Michigan
Ann Arbor, Michigan, USA 48109
filipi@umich.edu
* Corresponding Author

ABSTRACT

This paper proposes a self-learning approach to develop optimal power management with multiple objectives, e.g. to minimize fuel consumption and transient engine-out NOx and particulate matter emission for a series hydraulic hybrid vehicle. Addressing multiple objectives is particularly relevant in the case of a diesel powered hydraulic hybrid since it has been shown that managing engine transients can significantly reduce real-world emissions. The problem is formulated as an infinite time horizon stochastic sequential decision making/markovian problem. The problem is computationally intractable by conventional Dynamic programming due to large number of states and complex modeling issues. Therefore, the paper proposes an online self-learning neural controller based on the fundamental principles of Neuro-Dynamic Programming (NDP) and reinforcement learning. The controller learns from its interactions with the environment and improves its performance over time. The controller tries to minimize multiple objectives and continues to evolve until a global solution is achieved. The control law is a stationary full state feedback based on 5 states and can be directly implemented. The controller performance is then evaluated in the Engine-in-the-Loop (EIL) facility.

Keywords: Neuro dynamic programming (NDP), reinforcement learning, series hydraulic hybrid, power management, engine-in-the-loop (EIL), transient diesel emissions, online learning, optimal control, numerical optimization.

INTRODUCTION

There is a strong impetus for more efficient and cleaner vehicles because of growing environmental concerns and dwindling oil reserves. Advanced powertrains and exhaust after-treatment systems are required to meet the next generation emission regulations. Hybrid powertrains offer better fuel economy through optimization of engine operation, regeneration of braking energy, and by providing the opportunity for engine downsizing and engine shut-downs. Combined with efficient diesel engines, hybrids can provide significant leap in fuel economy. However, diesel engine exhaust after-treatment systems are complex and costly. Some components need regeneration e.g. lean NOx trap and that comes with a fuel economy penalty. The additional flexibility in controlling engine in a hybrid provides an opportunity for minimizing engine-out emissions and reducing the burden on aftertreatment. An intelligent supervisory controller; designed with multiple objectives such as the fuel economy and low vehicle exhaust emission, is essential for realizing both efficient and clean vehicles. This provides the impetus for the work presented here.

The supervisory power management controller's main task is to orchestrate the engine and secondary power to meet the driver power demand. The supervisory controller has profound impact on system operation and the ultimate benefits of hybridization. Numerous strategies have been proposed for design of supervisory controller. These approaches can be categorized as heuristic, optimal and suboptimal. Heuristic strategies [1], [2] often rely on researchers' knowledge about individual system efficiencies. These strategies are easy to implement but cannot capture complex system level effects. Optimal strategies aim to minimize an objective function, typically fuel consumption, over a given time horizon. Dynamic programming (DP) has been applied to numerically solve the optimization problem [3], [4]. Optimal controllers, however, are inherently non-causal i.e. they require knowledge about the future driving conditions. This limits their practical applicability and requires rule extraction which in turn sacrifices some of the fuel economy [5]. A suboptimal controller based on stochastic dynamic programming (SDP) eliminates the rule extraction step and gives a closed form controller which can be implemented in vehicle. SDP is not dependent on a particular driving cycle but the statistical characteristics of multiple driving cycles. SDP has been

successfully applied to many hybrid architectures [6], [7], [8]. The key objective, in the previous studies, has been fuel economy. Our goal is to develop a technique capable of developing a strategy for minimizing both the fuel consumption and emissions, e.g. NOx and soot.

Previous work done by Lin et al. [6], Tate et al. [9] and Johnson et al. [10] for including emissions while designing supervisory controller for hybrids used a steady state lookup table for predicting emissions. However, quasi-steady state map based models cannot accurately predict real emissions when the engine is operated transiently. This is due to complex nature of diesel combustion. Previous work done by Hagena et al. [11] showed that transient soot emissions accounts for as much as half of the total soot when engine is operated dynamically over an urban driving schedule. Hence, designing the supervisory controller for a hybrid with emissions objective requires transient emission predictions. In this paper we use hierarchical neuro-fuzzy model to predict transient NOx and soot emissions [12]. The model is composed of many local models valid for a certain input subspace. The idea is to divide the input space into smaller regions and train local models. The models have been shown to be computationally fast and accurate [12].

An inherent problem with including transient emission models in policy optimization is resulting increase of number of states. This is an obstacle in applying DP due to the well-known curse of dimensionality. The computational and memory cost to solve problems grow exponentially with increase in the states. This makes practical applicability of DP to real-life problems somewhat limited as most of these problems have large state space. Researchers have tried to circumvent this by using reduced models for design of optimal controllers. Policy optimization using SDP is confined to 2 and 3 states with a maximum state-action cardinality of 10^5. Authors presented an alternative algorithm [13], neuro-dynamic programming (NDP), to solve problems with large state space. In this paper we develop the NDP approach further and applies it to design a supervisory controller for series hydraulic hybrid which actively minimizes multiple objectives.

A self-learning neural controller based on principles of NDP and reinforcement learning is designed in this paper for series hydraulic hybrid vehicle (S-HHV). The controller learns to solve the energy management problem by interacting with the vehicle and powertrain and observing the consequences of its actions. To the best of our knowledge this approach is the first direct application of NDP techniques to solve power management problems for hybrid powertrains. The supervisory controller objectives are to minimize both fuel consumption and transient engine-out emissions. The self-learning controller comprises three neural networks, namely two actor and one critic networks. The critic network predicts the optimal cost-to-go value and the actor calculates the optimal engine speed and engine torque commands, based on current system states, to minimize the given objective function over infinite horizon. The problem considered in this paper has a state-action cardinality of 10^9. This is a considerable breakthrough in design of optimal power management controller for hybrids.

The paper is organized in three major sections. First, we describe the vehicle powertrain configuration, and modeling followed by the experimental setup for Engine-in-the-Loop (EIL) studies. In the next section, we formulate the energy management as a sequential decision problem. The concept of NDP is introduced and applied to the problem. Finally, results from EIL testing with self-learning controller based on NDP policy are presented. The paper ends with conclusions.

SERIES HYDRAULIC HYBRID

A series hydraulic hybrid configuration with two drive motors, one at each axle, is used for this study. Figure 1 gives a schematic of the vehicle configuration. The series configuration provides a full flexibility in operating engine as there is no mechanical coupling between engine and wheels. The additional degree of freedom in operating engine requires a methodical approach to supervisory control development. The supervisory controller acts as an intermediary between driver and propulsion system. The driver signal is sent to supervisory controller which then makes an informed decision and sends appropriate signals to engine and hydraulic pump/motors. The conventional wisdom suggesting the engine operation at the "sweet spot" has already been challenged by Filipi et al. [14], as this may not be best for the system-level efficiency. Adding the emissions objective will clearly pose a new challenge, and this motivates the development of an advanced algorithm.

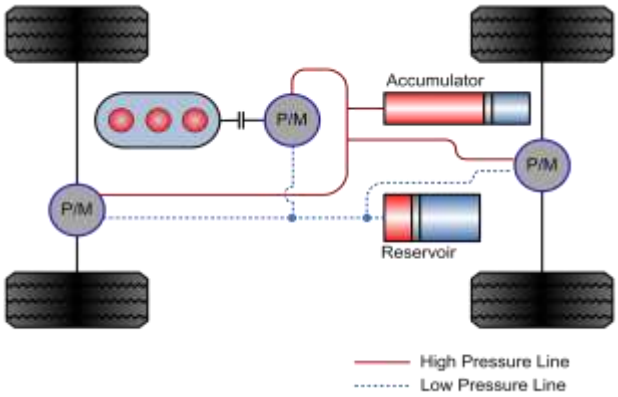

FIGURE 1: SERIES HYDRAULIC HYBRID

The energy is stored in accumulator by compressing nitrogen gas. The hydraulic accumulator is capable of high rate of charging or discharging with very high efficiency but has low energy density. The former is an advantage for series hybrid application while latter adds unique control challenges.

The simulation models for hydraulic hybrid powertrain and components were developed in the Automotive Research Center at the University of Michigan and previously used for optimization of design Filipi et al. [5] and supervisory control with fuel economy objective, e.g. Kim et al. [15].

TABLE 1: SERIES HYDRAULIC HYBRID SPECIFICATIONS

Engine	Description	6.4L International
	Max. Power	261kW @ 3000 RPM
	Max. Torque	881Nm @ 2000 RPM
Pump	Design	Axial Piston Variable Displacement
	Size	300 cc/rev
	Max Power	700 kW @ 350 bar @ 4000 RPM
Motor	Design	Axial Piston Variable Displacement
	Size	180 X 2 cc/rev
	Max Power	420 kW @ 350 bar @ 4000 RPM
Accumulator	Capacity	98 Liter (Gas Volume)
	Max Pressure	350 bar
	Min Pressure	120 bar
Vehicle	Type	HMMWV
	Weight	5112 kg
	Coeff. of Drag	0.7
	Frontal Area	3.58 m^2
	Tire Radius	0.4412 m
	Final Drive Ratio	4.086
Transmission	Design	2 speed automatic
	Gear Ratios	3, 1

The **engine** is modeled as a lookup table with speed and mass of fuel injected as inputs and brake torque as output. A diesel engine fuel injector controller provides the mass of fuel injected to the lookup table based on throttle command and engine speed. Turbo-lag is simulated by including a time delay in injection with time constant calibrated based on data obtained from engine testing [5].

The **hydraulic pump/motor** is modeled using updated Wilson's theory [16]. The pump/motor is a variable displacement axial piston type. The displacement command controls the torque and flow. Details of the model are provided in [5], [15]. The theoretical flow and torque is corrected with physics based expressions for losses which encompass laminar, compressibility and turbulent leakage for flow, and viscous, hydrodynamic and mechanical for torque.

A **hydraulic accumulator** stores energy in hydraulic hybrids. A full thermodynamic model is used for modeling the accumulator dynamic performance and efficiency. The equations are derived from energy conservation principles [16] and include the effects of the heat transfer. The real gas properties are captured using Benedict-Webb-Rubin equation. The formulas are omitted in this work for brevity and are available in [5]. The accumulator is modeled with elastomeric foam in the gas side in order to increase the thermal time constant and elevate the thermal efficiency [17].

The **vehicle** is modeled as a point mass system and pitch plane dynamics are ignored. This is deemed sufficient for system efficiency studies. The resistive force acting on the vehicle is sum of rolling friction and aerodynamic drag.

EMISSION MODEL

An emission model is required to quantify the engine-out emissions and to optimize the supervisory controller. The models need to capture transients accurately while being computationally efficient so that they can be used within DP framework. However emission formation in diesel engine is very complex phenomenon making it challenging to design a single model that can accurately capture all the nonlinearities. To circumvent this problem, the paper utilizes a neuro-fuzzy model tree framework similar to one used by Johri et al. [12] for design of virtual sensors for diesel engine emission. The model combines various local neural network based models with fuzzy framework and trains them on a large set of experimental data. Each model is locally valid and the contribution of each model is weighted according to their validity function. The output of model, y is the weighted sum of all local sub models $f_{NN}(.)$ with validity functions, ϕ determine the regions of input space where that particular model is active.

$$y = \sum_{i=1}^{M} f_{NN}(\bar{w}, \bar{u}) \cdot \Phi_i(\bar{u}) \quad (1)$$

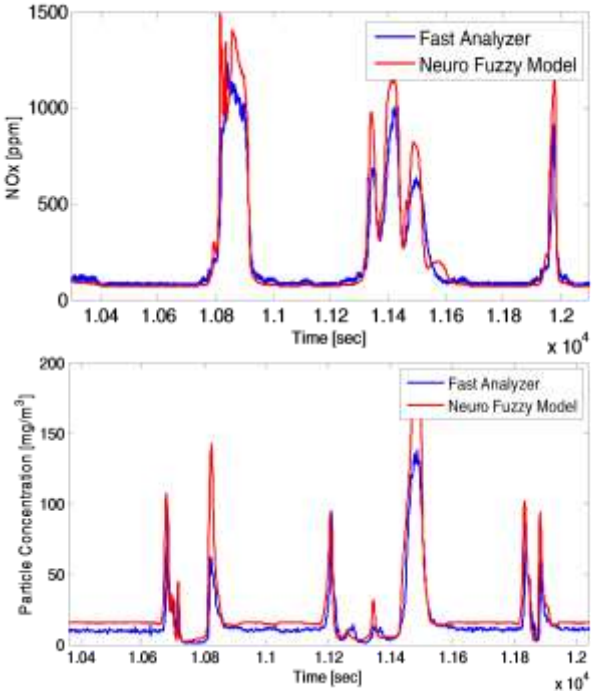

FIGURE 2: MODEL PREDICTION VS. MEASURED DATA

The local models in this paper are custom recurrent neural networks. The input space is divided based on engine operating speed. This choice of input space division is based on the experiment carried out to characterize the engine. Details about

the experiment and perturbation signal are available in [12]. The fuzzy framework of neuro-fuzzy model tree uses triangle membership functions. Each local neural network, in this paper, has 6 inputs. Both NOx and soot models have 4 inputs in common, namely, current engine speed, current engine torque, current steady state boost and current mass of fuel injected. The NOx model, in addition, uses steady state NOx and previous predicted NOx whereas soot model uses rate of change of fuel injection and previous predicted soot. The steady state boost, fuel injection and steady state NOx are calculated using lookup table based models. The feedback from output is also considered as different input. Each input is preprocessed and normalized to -1 and 1. Figure 2 shows the model prediction along with the test cell measurement of NOx and soot using fast emission analyzers. It can be seen that models provide a very good estimate of transient emissions.

ENGINE EXPERIMENTAL SETUP

The engine used for this work is a 6.4L V8 medium duty diesel engine manufactured by the Navistar Ltd. The engine incorporates modern technologies to provide high power density while meeting emission regulations. A common rail direct injection system permits precise control of fuel injection timing, pressure and quantity. The engine is equipped with dual stage variable geometry turbocharger (VGT). An exhaust gas recirculation circuit (EGR) allows for introducing cooled exhaust gases into intake manifold and reduce NOx emissions. EGR is modulated using EGR valve and VGT vane geometry.

The engine is coupled to a 330 kW AVL ELIN series 100 APA Asynchronous Dynamometer. The dynamometer is suited for transient testing with 5ms response time and 10ms torque reversal time (+100% to -100%). The engine is fully instrumented with time based measurements like temperature, manifold pressure and flow rates as well as crank based measurements like in-cylinder and fuel injection pressures.

Engine out temporal measurement of NOx and particulate matter emissions are carried out using fast analyzers from Cambustion Ltd. The CLD 500 Fast NOx analyzer consists of chemiluminescent detector with a 90%-10% response time of 3ms for NO and 10ms for NOx. The particulate matter is analyzed using differential mobility spectrometer (DMS) 500. The instrument measures the number of particles and their spectral weighting between 5nm to 1000nm with a time response of 200ms. The particle size-number distribution is then converted to mass and the masses per bin are summed to get the total particulate matter in the exhaust. More details about the experimental setup are given in [18].

SUPERVISORY CONTROLLER

Problem Formulation

Given the vehicle configuration, the paper examines the following power management problem over an infinite horizon:

$$\begin{aligned} &minimize: J = \lim_{N\to\infty} E_{w_k}\left\{\sum_{k=0}^{N-1} \alpha^k g\left(x_k, u_k, w_k\right)\right\} \\ &subject\,to: x_{k+1} = f(x_k, u_k, w_k) \\ &\qquad x \in X \\ &\qquad u \in U \\ &where,\ \ x_k = \left[SOC^k, \omega_{wh}^k, \omega_e^{k-1}, m_f^{k-1}\right]^T, \\ &\qquad u_k = \left[\omega_e^k, T_e^k\right]^T, \\ &\qquad w_k = P_{dem}^k \end{aligned} \tag{2}$$

In the above discrete time stochastic optimal control problem, *k* is the time index. The *g* represents the instantaneous cost, which in this paper, is weighted sum of fuel consumption and transient emissions. Discount factor, $0<\alpha<1$, implies that the future costs are less important than the cost incurred at the present time. Also it ensures that the cumulative optimization cost remains finite over infinite horizon. The system has 4 states x_k : State of Charge SOC^k , present vehicle speed ω_{wh}^k , previous engine speed ω_e^{k-1} and previous fuel injected m_f^{k-1}, 2 control inputs u_k : present engine speed ω_e^k and present engine torque T_e^k and 1 disturbance input w_k : present driver power demand P_{dem}^k .

The above optimization is subject to constraints imposed by deterministic dynamic equations for vehicle along with admissible set of states, *X* and control inputs, *U*.

$$U = \begin{cases} \omega_{e,\min} \le \omega_e^k \le \omega_{e,\max} \\ T_{e,\min}(\omega_e^k) \le T_e^k(\omega_e^k) \le T_{e,\max}(\omega_e^k) \\ T_{m,\min}(\omega_m^k, SOC^k) \le T_m^k \le T_{m,\max}(\omega_m^k, SOC^k) \\ T_{p,\min}(\omega_p^k, SOC^k) \le T_p^k \le T_{p,\max}(\omega_p^k, SOC^k) \end{cases} \tag{3}$$

$$X = \begin{cases} SOC_{\min} \le SOC^k \le SOC_{\max} \\ 0 \le m_f^k(T_e^k, \omega_e^k) \le m_{f,\max}(T_{e,\max}, \omega_e^k) \\ \omega_{wh}^k = \omega_{wh,req} \end{cases}$$

where, script ω denotes speed, *T* denotes torque and subscripts *e*, *m* and *p* denotes engine, motor and pump respectively.

The driver power demand is the stochastic component of the model, w_k and is modeled using discrete time Markov chain. Given the present power demand, P_{dem} and vehicle velocity, ω_{wh} the model gives transition probability to next power demand.

$$\begin{aligned} &p_{ij,m} = \Pr\left\{w = P_{dem}^j \mid P_{dem} = P_{dem}^i, \omega_{wh} = \omega_{wh}^m\right\} \\ &i, j = 1, 2, \ldots, N_p \qquad m = 1, 2, \ldots, N_\omega \end{aligned} \tag{4}$$

where, *i* and *j* index denotes present and future power demand

respectively and *m* indexes present vehicle speed. The transition probability matrix is estimated by statistically analyzing the different driving cycles [8].

The instantaneous cost is the cost incurred by system when it transitions from given system states to new system states with a given control input applied. The instantaneous cost g is

$$g = w_{FC} \cdot FC + w_{NOx} \cdot NO_x + w_{PM} \cdot PM + \mu \cdot (SOC - SOC_{ref})^2 \cdot (SOC < SOC_{ref}) \quad (5)$$

where *FC*, *NOx* and *PM* are the normalized fuel consumption, normalized transient NOx emission and normalized transient particulate matter emission respectively. The w_{FC}, w_{NOx} and w_{PM} are the weighting parameter and $\sum (w_{FC} + w_{NOx} + w_{PM}) = 1$. The *FC* takes into consideration system efficiency, i.e. both engine and pump efficiency and hence is a function of SOC also. The latter term penalizes the deviation of SOC below a threshold SOC, 0.2 in this paper. The penalty is imposed to maintain vehicle drivability at all conditions [8].

A hybrid policy iteration algorithm has been applied in the past by Lin et al. [6], Johri et al. [8] to solve stochastic control problems of the above nature. The policy iteration algorithm iterates between policy evaluation step and policy improvement step until the optimal cost-to-go function converges. The problem formulated above with each state and control input discretized with cardinality of 25 has approximately 10^9 state-action pairs. A state-action space of this size is computationally intractable with conventional policy iteration algorithm. Also, even if the problem could be solved, it would require vast amounts of memory to store every action value for every combination of states. NDP provides an approach to sub-optimally solve the above problem. The idea is to approximate the optimal cost-to-go function with a surrogate function. The value function *J(.)* in Bellman equation is replaced by suitable approximation, $\tilde{J}(\cdot, r)$ where *r* is the parameter vector. This can be considered as mapping of higher dimensional cost-to-go function with a lower dimensional function.

Neuro-Dynamic Programming (NDP)

NDP is a class of reinforcement learning methods that deal with the curse of dimensionality using neural network based approximations of the cost-to-go function. Reinforcement learning accomplishes a particular task by trial-and-error based on interactions with environment. The controller learns to perform a task solely on the outcome of its experience. Two critical ideas in NDP approach are (i) Compact functional representation of cost-to-go, and (ii) Recursive method for updating the functional approximation of cost-to-go upon each observation of state transition and associated cost. Sutton et al. [19] proposed temporal difference learning as a method for approximating long-term future cost as a function of present state. The algorithm improves the approximation of the long term cost as more and more state transitions are observed in an incremental fashion. The paper proposes a neural network functional approximation for cost-to-go combined with recursive update of network for solving policy optimization problem in hybrids.

The NDP approach presented in this paper is an on-line learning control scheme. Figure 3 shows the structure of NDP framework. The critic network is trained to predict optimal cost-to-go function. The action network is trained such that the control policy is optimal with respect to cost-to-go function. In contrast to usual supervised training of neural networks, there are no input-output training pairs for optimal cost-to-go value. Instead the critic network is updated using the reinforcement signal obtained by interacting with the environment. This signal is known as temporal difference (TD) d_k and defined by

$$d_k = g(i_k, u_k, i_{k+1}) + \alpha \tilde{J}(i_{k+1}, r_k) - \tilde{J}(i_k, r_k) \quad (6)$$

TD is the prediction error between predicted performance and observed performance in response to action u_k. Bellman's equation is a fixed point equation and by rearranging, we can obtain the TD formulation. For Bellman equation to hold, the TD error should be zero. Therefore, for a given control policy, π the equation $d_k = 0$ can be solved for in least square sense. TD methods are family of algorithms and detailed discussion is given by Bertsekas et al. [20] and Sutton et al. [19].

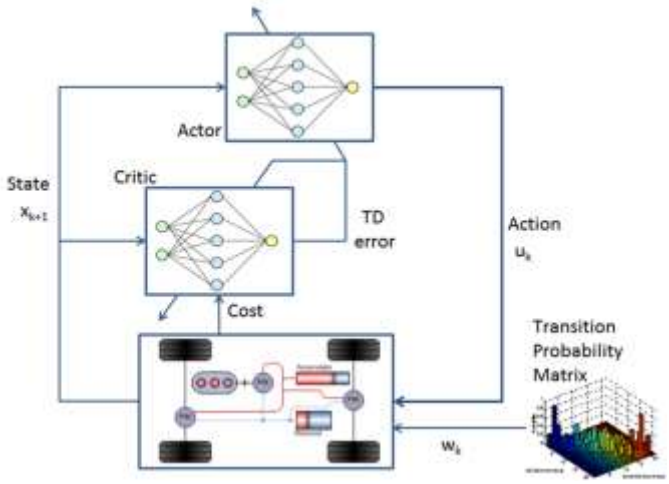

FIGURE 3: SCHEMATIC DIAGRAM FOR IMPLEMENTATION OF NEURO DYNAMIC PROGRAMMING

The critic neural network is trained incrementally using TD(λ) update. The parameter vector is then updated by running a TD(λ) update [19]

$$r_{k+1} = r_k + \gamma_k d_k \sum_{m=0}^{k} (\alpha\lambda)^{k-m} \nabla \tilde{J}(i_m, r_m) \quad (7)$$

where γ is the step size, λ is the TD parameter and gradient $\nabla \tilde{J}(i, r)$ is the vector of partial derivatives with respect to parameter vector *r*. The above equation is an incremental gradient update with steepest descent update.

Define eligibility vector, z_k

$$z_k = \sum_{m=0}^{k} (\alpha\lambda)^{k-m} \nabla \tilde{J}(i_m, r_m) \tag{8}$$

The TD update can be written as

$$r_{k+1} = r_k + \gamma_k d_k z_k \tag{9}$$

where z_k is updated by

$$z_{k+1} = \alpha\lambda z_k + \nabla \tilde{J}(i_{k+1}, r_{k+1}) \tag{10}$$

The critic network update can be accelerated by including non-linear learning rates. The idea is to use approximation of Hessian matrix and the update formula becomes

$$r_{k+1} = r_k + \gamma_k d_k \sum_{m=0}^{k} (\alpha\lambda)^{k-m} H_m^{-1} \nabla \tilde{J}(i_m, r_m) \tag{11}$$

where H_k is Hessian and is approximated by

$$H_k \approx \sum_m^k \nabla \tilde{J}(i_m, r_m) \cdot \nabla^T \tilde{J}(i_m, r_m) \tag{12}$$

Direct computing of the matrix H^{-1} is very computationally costly. Kalman theory can be efficiently applied for calculating the inverse of Hessian. Define Kalman matrix $K_{k+1} = H_{k+1}^{-1}$ and applying Sherman-Morrison matrix identity

$$K_{k+1} = K_k - \frac{\left(K_k \nabla \tilde{J}(i_{k+1}, r_{k+1})\right) \cdot \left(K_k \nabla \tilde{J}(i_{k+1}, r_{k+1})\right)^T}{1 + \nabla^T \tilde{J}(i_{k+1}, r_{k+1}) K_k \nabla \tilde{J}(i_{k+1}, r_{k+1})} \tag{13}$$

The step size in Eq. (11) plays a very important role in convergence of functional approximation to optimal cost-to-go value. In TD(λ) algorithm, the cost-to-go, J of being in a particular state is estimated from $\tilde{J}(\cdot, r)$ which itself is non-stationary and steadily changing. The step size needs to strike a balance between minimizing error (small step size) and responding to non-stationary data (large step size).

Powell [21] derived an optimal step size rule, called ***Bias Adjusted Kalman Filter*** (BAKF), for estimating parameter from sequence of independent observations $\hat{\theta}^n$ with unknown mean θ^n and variance σ^2. The step size solution is given explicitly by formula

$$\alpha_{n-1} = 1 - \frac{\sigma^2}{(1 + \lambda^{n-1})\sigma^2 + (\beta^{n-1})^2} \tag{14}$$

where λ is computed recursively using

$$\lambda^n = \begin{cases} (\alpha_{n-1})^2 & n = 1 \\ (1 - \alpha_{n-1})^2 \lambda^{n-1} + (\alpha_{n-1})^2 & n > 1 \end{cases} \tag{15}$$

and $\beta^{n-1} = \theta^n - \mathrm{E}\left[\bar{\theta}^{n-1}\right]$, i.e. bias in smoothed estimate from previous iteration. The bias itself is computed recursively as it is also unknown. This paper employs BAKF algorithm for calculating optimal stepsize.

In a standard actor critic method, the policy μ is kept constant till the critic's computations converge to J^μ. This new converged value of J^μ is then used by critic to calculate a new policy. This may not be suitable for problems with large state space as evaluating over all state combinations would mean long computational time between policy updates. In this paper, the new policy is calculated subsequent to every state transition. The actor carries out new policy after every simulated transition and is known as ***optimistic policy iteration***. The convergence behavior of algorithm is quite complex and not fully understood [20]. However, optimistic policy iteration with TD(λ) update is one of the most effective NDP methods.

It is computationally intractable to visit every state-action pair and evaluate cost-to-go function. To circumvent this problem, an exploitation policy uses the present knowledge of cost-to-go function and chooses policies which are greedy or opportunistic with respect to present cost-to-go function approximation. However, a pure exploitation policy is susceptible to getting stuck at local optimum because of poor estimate of cost-to-go function at certain states. To avoid this, we use a modified exploitation policy known as ***ε-greedy*** policy [19]. The algorithm chooses a greedy policy based on the present knowledge of cost-to-go function most of the times but reverts to exploration strategy with small probability ε. On limit, the algorithm is converges to optimal policy [19].

NDP BASED SUPERVISORY CONTROL

The self-learning controller has three neural networks, one critic network and two action networks which are trained using TD(λ) approach, Figure 3. The networks are initialized with random weights i.e. they start naïve. The networks are trained incrementally using TD signal and learn to control the hybrid powertrain as the algorithm progresses. The algorithm performs Monte Carlo simulations to generate sample trajectories. At any given state, the action network is evaluated and the control input is applied to the system. The algorithm calculates the TD and updates the critic network using Eq. (11). The system moves to newer states based on the applied input. The actor network is then updated to produce control actions which are *ε-greedy* with respect to latest cost-to-go function approximation. Since the algorithm calculates newer policies and next states based on the present states visited, it can get stuck in a confined state-action space. To overcome this problem, the algorithm is restarted frequently from random states.

Critic and action neural networks are multilayer feedforward perceptron networks with one hidden layer. The input and hidden layers have *tan-sigmoid* activation function whereas output layer has linear activation function. The critic network is trained incrementally by backpropogating the temporal difference and the weights are updated using Eq. (11). The output of the action networks are optimal engine speed and engine torque for given system states. The training of action network is also carried out sequentially using update method similar to critic network, given by Eq. (11). This generates an implementable S-HHV power management controller.

RESULTS

A self-learning neural network controller is generated using NDP algorithm described in earlier with weights, w_{FC} = 0.7, w_{NOx} = 0.1 and w_{PM} = 0.2. The selection of weights are random and are to show the ability of algorithm to not only learn to manage two power sources but to do by minimizing weighted sum of fuel consumption and transient emissions. The controller is then evaluated over the FTP75 city driving schedule using EIL. Figure 4 shows the engine operating points over BSFC map for NDP and SDP based controllers. The color scale indicates the amount of fuel consumed in each region during FTP75 driving schedule. The plot also shows the best BSFC line. It can be seen that the engine operation for NDP controller deviates from it to reduce NOx. The engine would have operated near the best BSFC line if the engine efficiency is the sole objective, as is the case with SDP based controller.

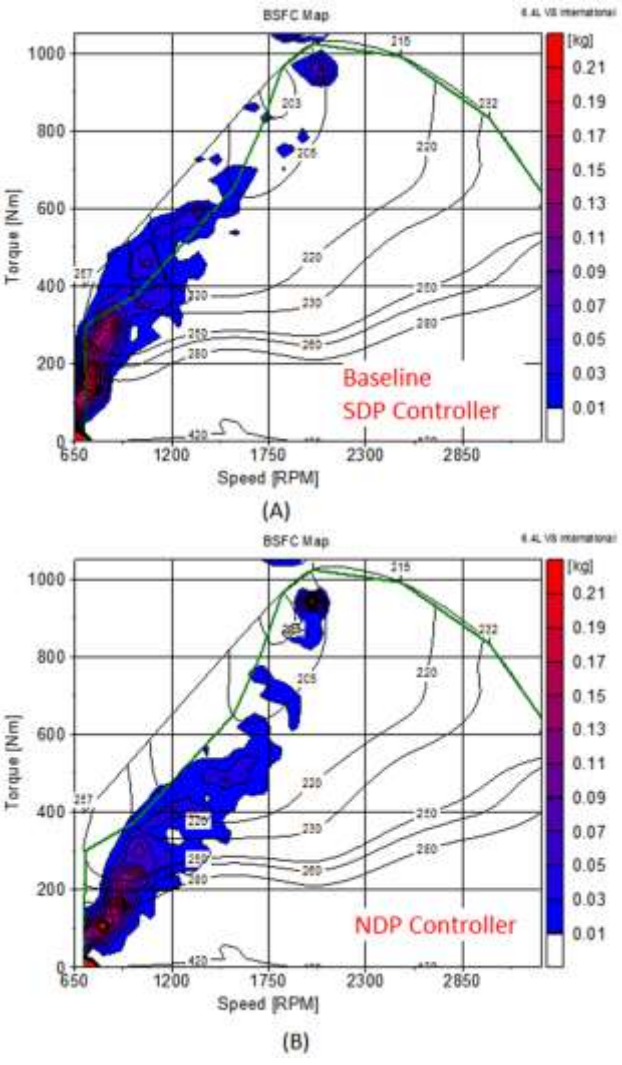

FIGURE 4: ENGINE VISITATION POINT ON BSFC MAP. (A) SDP CONTROLLER, (B) NDP CONTROLLER

Figure 5 shows the time trace of fuel injected per stroke for SDP and NDP based controllers. It can be seen that the fuel injection ramps up and down slowly for NDP based controller to reduce transient spikes of emission. This is in agreement with finding by Hagena et al. [11] that a step change in fueling results in large spike in transient emissions and can be significantly reduced if fueling change occurs gradually. The effect is particularly strong when a step change is initiated from idle [11]. The NDP strategy successfully avoids this, e.g. the engine is not brought down to idle at ~35 sec, and when the ramp-up increased the initial rate is mild, followed by a steeper slope only beyond 10mg/stroke. Therefore, the NDP based controller results in 29.3% improvement of NOx over SDP based controller in EIL over FTP schedule.

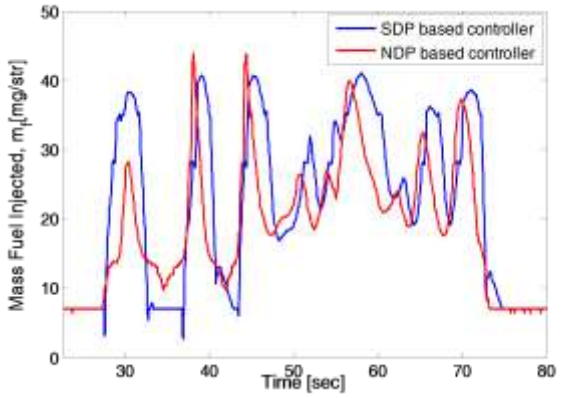

FIGURE 5: TEMPORAL PLOT OF MASS FUEL INJECTED OVER SECTION OF FTP SCHEDULE

By systematically evaluating the system behavior NDP controller manages significant emission reduction with minimal fuel economy penalty. The S-HHV fuel economy with NDP based controller is 16.2 mpg, only 3% lower than the result obtained using SDP with fuel economy being sole objective. The improvement of fuel economy over the conventional baseline is 52%.

CONCLUSION

In this paper, a power management strategy for series hydraulic hybrid is developed using neuro-dynamic programming and reinforcement learning. The power management of hydraulic hybrid is setup as a sequential decision making problem under uncertainty (stochastic control). The controller objective is to minimize multiple objectives, fuel consumption and transient emissions, over an infinite horizon. Two key aspects of the NDP approach are:

1. Approximation of cost-to-go function with neural networks: This reduces the memory required as only network weights need to be stored and cost-to-go can be approximated using neural network compared to traditional dynamic programming approach which requires storing of cost-to-go value at every state.
2. Incremental Learning: The learning of the cost-to-go function is performed in an incremental fashion using Temporal Difference algorithm.

The self-learning controller is implemented on a dSpace real-time system and simulated along with real diesel engine and virtual powertrain/vehicle. EIL results show that NDP based supervisory controller is able to successfully orchestrate the power management in a series hydraulic hybrid to meet performance objectives while significantly reducing NOx emissions and preserving most of the fuel economy gain attainable with optimized policy.

ACKNOWLEDGMENT

The financial support for this research has been provided by the State of Michigan 21st Century Jobs Fund in partnership with the Bosch-Rexroth Corporation. In addition, the authors wish to gratefully acknowledge technical interactions with Simon Baseley and Ed Greif, both at Bosch-Rexroth.

REFERENCES

[1] Jalil, N. , Kheir, N.A. , and Salman, M. , 1997, "A rule-based energy management strategy for a series hybrid vehicle," *Proceedings of the American Control Conference*, **1**, pp. 689-93.

[2] Baumann, B. M., Washington, G. , Glenn, B. C., and Rizzoni, G. , 2000, "Mechatronic design and control of hybrid electric vehicles," IEEE/ASME Transactions on Mechatronics, **5**(1), pp. 58-72.

[3] Lin, C. C., Filipi, Z. , Louca, L. , Peng, H. , Assanis, D. , and Stein, J. , 2004, "Modelling and control of a medium-duty hybrid electric truck," International Journal of Heavy Vehicle Systems, **11**(3-4), pp. 349-371.

[4] Sciarretta, A. and Guzzella, L. , 2007, "Control of hybrid electric vehicles," IEEE Control Systems Magazine, **27**(2), pp. 60-70.

[5] Filipi, Z. , Louca, L. , Daran, B. , Lin, C.-C. , Yildir, U. , Wu, B. , Kokkolaras, M. , Assanis, D. , Peng, H. , Papalambros, P. , Stein, J. , Szkubiel, D. , and Chapp, R. , 2004, "Combined optimisation of design and power management of the hydraulic hybrid propulsion system for the 6 X 6 medium truck," International Journal of Heavy Vehicle Systems, **11**(3-4), pp. 372-402.

[6] Lin, C. C., Peng, H. , and Grizzle, J.W. , 2004, "A stochastic control strategy for hybrid electric vehicles," *Proc. of American Control Conference*, **5**, pp. 4710-4715.

[7] Liu, J. , Hagena, J. , Peng, H. , and Filipi, Z.S. , 2008, "Engine-in-the-loop study of the stochastic dynamic programming optimal control design for a hybrid electric HMMWV," International Journal of Heavy Vehicle Systems, **15**(2-4), pp. 309-26.

[8] Johri, R. and Filipi, Z. , 2010, "Low-Cost Pathway to Ultra Efficient City Car: Series Hydraulic Hybrid System with Optimized Supervisory Control," SAE International Journal of Engines, **2**(2), pp. 505-520.

[9] Tate, E.D. , Grizzle, J.W. , and Peng, H. , 2010, "SP-SDP for Fuel Consumption and Tailpipe Emissions Minimization in an EVT Hybrid," IEEE Transactions on Control Systems Technology, **18**(3), pp. 673-87.

[10] Johnson, V. H., Wipke, K. B., and Rausen, D. J., 2000, "HEV Control Strategy for Real-Time Optimization of Fuel Economy and Emissions," *SAE Technical Paper 2000-01-1543*.

[11] Hagena, J. R., Filipi, Z. S., and Assanis, D. N., 2006, "Transient Diesel Emissions: Analysis of Engine Operation During a Tip-In," *SAE Technical Paper 2006-01-1151*.

[12] Johri, R. , Salvi, A. , and Filipi, Z. , 2011, "Real-Time Transient Soot and NOx Virtual Sensors for Diesel Engine using Neuro-Fuzzy Model Tree and Orthogonal Least Squares," *Proceedings of the ASME 2011 Internal Combustion Engine Division Fall Technical Conference*.

[13] Johri, R. and Filipi, Z. , 2011, "Self-Learning Neural controller for Hybrid Power Management using Neuro-Dynamic Programming," *SAE Technical Paper 2011-24-0081*.

[14] Filipi, Z and Kim, Y J, 2010, "Hydraulic Hybrid Propulsion for Heavy Vehicles: Combining the Simulation and Engine-In-the-Loop Techniques to Maximize the Fuel Economy and Emission Benefits," Oil & Gas Science & Technology, **65**(1), pp. 155-178.

[15] Kim, Y. and Filipi, Z. , 2007, "Simulation Study of a Series Hydraulic Hybrid Propulsion System for a Light Truck," SAE Transactions, Journal of Commercial Vehicles, **116**(2007-01-4151), pp. 147-161.

[16] Pourmovahed, A. , Beachley, N.H. , and Fronczak, F.J. , 1992, "Modeling of a hydraulic energy regeneration system. Part I. Analytical treatment," Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, **114**(1), pp. 155-159.

[17] Pourmovahed, A. , Baum, S.A. , Fronczak, F.J. , and Beachley, N.H. , 1988, "Experimental Evaluation Of Hydraulic Accumulator Efficiency With And Without Elastomeric Foam.," Journal of Propulsion and Power, **4**(2), pp. 185-192.

[18] Filipi, Z. , Fathy, H. , Hagena, J. , Knafl, A. , Ahlawat, R. , Liu, J. , Jung, D. , Assanis, D. , Peng, H. , and Stein, J. , 2006, "Engine-in-the-Loop Testing for Evaluating Hybrid Propulsion Concepts and Transient Emissions – HMMWV Case Study," SAE Transactions, Journal of Commercial Vehicles, **115**(2006-01-0443).

[19] Sutton, R. S. and Barto, A. G., 1998, *Reinforcement learning an introduction*, MIT Press.

[20] Bertsekas, D. P. and Tsitsiklis, J. N., 1996, *Neuro-dynamic programming*, Athena Scientific.

[21] Powell, W. B., 2007, *Approximate dynamic programming: solving the curses of dimensionality*, Wiley-Interscience.