52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

An Approximate Dynamic Programming Approach for Model-free Control of Switched Systems

Wenjie Lu and Silvia Ferrari

Abstract— **Several approximate dynamic programming (ADP) algorithms have been developed and demonstrated for the model-free control of continuous and discrete dynamical systems. However, their applicability to hybrid systems that involve both discrete and continuous state and control variables has yet to be demonstrated in the literature. This paper presents an ADP approach for hybrid systems (hybrid-ADP) that obtains the optimal control law and discrete action sequence via online learning. New recursive relationships for hybrid-ADP are presented for switched hybrid systems that are possibly nonlinear. In order to demonstrate the ability of the proposed ADP algorithm to converge to the optimal solution, the approach is demonstrated on a switched, linear hybrid system with a quadratic cost function, for which there exists an analytical solution. The results show that the ADP algorithm is capable of converging to the optimal switched control law, by minimizing the cost-to-go online, based on an observable state vector.**

I. INTRODUCTION

Many complex systems can be described as hybrid dynamical systems that are characterized by both continuous and discrete state and control variables. A common example of hybrid system that has been used, among other applications, to describe systems of collaborative agents, is a switched system in which multiple modes of motion are switched according to a finite set of discrete actions or events [1], [2]. A switched system can coordinate a variety of subsystems (modes) with their unique structures, allowing more flexibility in dynamic models. The hybrid nature of multi-agent networks has been recognized by several authors [3], [4]. A hybrid modeling approach for a mobile multi-agent network was recently developed in [5], and shown highly effective at maintaining a desired formation, and connectivity among the agents. A hybrid modeling framework for robust maneuver-based motion planning in nonlinear systems with symmetries was proposed in [6]. The reader is referred to [7] for a more comprehensive review of hybrid systems with autonomous or controlled events.

The optimal control of a switched system seeks to determine multiple optimal continuous controllers, and a corresponding optimal discrete switching sequence, such that a scalar objective function of the hybrid system state and control is minimized over a period of time [7]. Dynamic programming has been proposed for the constrained optimal control of discrete-time linear hybrid systems [8], [9]. Because of the high dimensionality of the state and control spaces, however, the optimal control of switched systems is often challenging or even computationally intractable. Approximate dynamic programming (ADP) is an effective approach for overcoming the curse of dimensionality of dynamic programming algorithms, by approximating the optimal control law and value function recursively over time [10], [11]. Furthermore, by using recursive relationships that adapt the control law and value function forward in time, ADP algorithms have the ability to solve an optimal control problem online, subject to an observed state, and without an explicit or accurate representation of the system dynamics [12], [13].

Several approximate dynamic programming (ADP) algorithms have been developed and demonstrated for the model-free control of continuous and discrete dynamical systems [14], [15], [16]. However, the applicability of ADP to hybrid systems that involve both discrete and continuous state and control variables has yet to be demonstrated in the literature. This paper presents an ADP approach for hybrid systems (hybrid-ADP) that obtains the optimal control law and discrete action sequence via online learning. The hybrid-ADP approach presented in this paper is not to be confused with hybrid ADP algorithms that, despite a similar name, referred to a class of ADP methods that combine direct and indirect optimization of the control law and value function approximations.

This paper presents new ADP recursive relationships for the optimal control of switched hybrid systems that are possibly nonlinear, and model free. In order to demonstrate the ability of the proposed ADP relationships to converge to the optimal solution, the algorithm is demonstrated on a switched, linear hybrid system with a quadratic cost function, for which there exists an analytical solution. The analytical solution of this linear, quadratic switched optimal control problem was first obtained by Riedinger in [17]. Other approaches to the same linear, quadratic switched optimal control problem are reviewed comprehensively in [18]. Also,

This work was supported by grant NSF ECS CAREER 0448906.

W. Lu and S. Ferrari are with Department of Mechanical Engineering and Material Science, Duke University, Durham, NC {wenjiie.lu, silvia.ferrari}@duke.edu

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

an approach for iterating between the optimization of the switching sequence, and the optimization of the switching instants was developed for switched affine systems in [2]. The method in [2], however, cannot be used to optimize the continuous control laws. Another parametric-optimization method was proposed in [19] to optimize the continuous control laws, for a given (predesigned), fixed switching sequence.

Existing iterative approaches seek to overcome the curse of dimensionality by fixing either the switching sequence or the continuous control law. The hybrid-ADP approach developed in this paper exploits the ADP recursive approximation approach and Bellman's equation in [20], [21], [22] to adapt the continuous control law, the mode switching sequence, the switching instants, and the corresponding value function, iteratively over time. The results show that the proposed hybrid-ADP algorithm is capable of converging to the optimal switched control law of a linear, quadratic switched hybrid system online, subject to actual system dynamics.

The paper is organized as follows. Section II describes the switched optimal control problem formulation and assumptions. The background on ADP is reviewed in Section III. Section IV presents new ADP recursive relationships and transversality conditions, and learning rules for ADP critic and control networks. The numerical simulations and results are presented in Section V.

II. Optimal Control of Switched Systems

The optimal control of switched hybrid systems arises in a wide variety of fields, such as mobile manipulator systems, autonomous robotic sensor planning, and autonomous assemble lines. In these applications, both the discrete actions and the continuous control are crucial to system performance. The switched system considered in this paper has E discrete modes, and the mode and continuous state at time t are denoted by $\xi(t) \in \mathcal{E} = \{1, \dots, E\}$ and $\mathbf{x}(t) \in \mathbb{R}^n$, respectively. The continuous control for the system under mode ξ is denoted by $\mathbf{u}_\xi(t) \in \mathcal{U}_\xi \subset \mathbb{R}^{m_\xi}$. The discrete action is denoted by $a(t) \in \mathcal{E}$, and is represented by a piecewise-constant function from the right, denoted by $t-$. Then, switched dynamical system is described by the set of equations,

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f_\xi[\mathbf{x}(t), \mathbf{u}_\xi(t)] \\ \xi(t) &= a(t) \end{aligned} \tag{1}$$

where f_ξ is the nonlinear dynamic equation of the switched system under mode $\xi \in \mathcal{E}$. Let $\{0,\ t_1,\ \dots, t_i, t_{i+1}, \dots, \infty\}$ denote the sequence of the switching instants when $\xi(t) \neq \xi(t-)$, and let $\{\xi_0,\ \xi_1, \dots, \xi_i, \dots, \xi_\infty\}_{\xi_i \in \mathcal{E}}$ denote the switching mode sequence.

The initial system state $\mathbf{x}_0$, and the goal state $\mathbf{x}_g$ are assumed known *a priori*. The problem considered in this paper is a switched, nonlinear, infinite-horizon, continuous-time, optimal control problem, with and objective function,

$$J \triangleq \sum_{i=0}^{\infty} \int_{t_i}^{t_{i+1}(-)} \mathscr{L}_{\xi(t_i)}[\mathbf{x}(\tau), \mathbf{u}_{\xi(t_i)}(\tau)] d\tau \tag{2}$$

to be minimized with respect to the continuous control $\mathbf{u}^*(\cdot)$ and the discrete control $a^*(\cdot)$, subject to (1), and with a known Lagrangian $\mathscr{L}_\xi : \mathbb{R}^n \times \mathcal{U}_\xi \to \mathbb{R},\ \xi \in \mathcal{E}$.

The above optimal control problem is approached using ADP, under the following assumptions.

Assumption 1: The switch between modes can occur at any time, and it is fully controlled by the discrete action $a(t)$. The cost of each switch is zero.

Assumption 2: The dynamic equations $f_\xi(\mathbf{y}, \mathbf{w})$ and the cost function $\mathscr{L}_\xi(\mathbf{y}, \mathbf{w})$, can only be evaluated at $\mathbf{y} \in \mathcal{N}[\mathbf{x}(t)]$, $\forall\ \xi \in \mathcal{E}$ and $\forall\ \mathbf{w} \in \mathcal{U}_\xi$, where $\mathcal{N}[\mathbf{x}(t)] = \{\mathbf{y} \mid \|\mathbf{x}(t) - \mathbf{y}\| < r\}$ is the neighborhood set of the system's current continuous state $\mathbf{x}(t)$. The operator $\|\cdot\|$ is the L-2 norm and r is a positive number.

Assumption 3: The system state $\mathbf{x}$ is fully observable, and error free.

The ADP approach is reviewed in the next section, and then used in Section IV to obtain new ADP relationships for the switched optimal control problem presented in this section.

III. Background on Approximate Dynamic Programming

Approximate dynamic programming (ADP) is an effective approach for overcoming the curse of dimensionality associated with dynamic programming (DP) algorithms for optimal control problems. [20], [23]. ADP has been successfully demonstrated for model-free, online control of continuous dynamical systems [12], [13], [10], [24], and discrete Markov decision process (MDP) [25]. For a non-hybrid optimal control problem, a discrete-time value function can be defined,

$$V[\mathbf{x}(\kappa)] \triangleq \sum_{j=\kappa}^{\infty} \gamma^{j-\kappa} \mathscr{L}[\mathbf{x}(j), \mathbf{u}(j)] dt \tag{3}$$

where dt is the size of time grids, j and κ are the indices of the time grids, and the Lagrangian $\mathscr{L}$ and discount factor γ are semi-definite positive functions. Let $V^*(\kappa)$ denote the optimal value function, and $\mathbf{u}^*(\cdot)$ denote the optimal control law. Then, from Bellman's equation, the ADP recursive relationship,

$$\begin{aligned} V^*[\mathbf{x}(\kappa)] &= \mathscr{L}[\mathbf{x}(\kappa), \mathbf{u}(\kappa)] dt \\ &\quad + \sum_{j=\kappa+1}^{\infty} \gamma^{j-\kappa} \mathscr{L}[\mathbf{x}(j), \mathbf{u}(j)] dt \\ &= \mathscr{L}[\mathbf{x}(\kappa), \mathbf{u}(\kappa)] dt + V^*[\mathbf{x}(\kappa+1)] \end{aligned} \tag{4}$$

can be obtained, where $\mathscr{L}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)]dt$ is the instantaneous reward or cost function, as shown in [20], [23].

Classical DP algorithms iterate backwards in time, starting from a known final time and state, and using the principle of optimality to eliminate sub-optimal costs and control laws. As a result, they cannot be applied to optimize the value function and control law online. The ADP approach, on the other hand, iterates forward in time, by using the recursive relationships in (4) to improve its approximation of the optimal value function $V^*(\kappa)$ (or its gradient), and optimal control law $\mathbf{u}^*(\cdot)$, through aggregation functions [26], [27], such as supporting vector machines [28], and neural networks [29]. The value function approximation is commonly referred to as the critic network, and the control law approximation is referred to as control network, and they are both optimized based on the difference between the reward (or cost) expected and the reward (or cost) obtained by actuating the controller.

An example of ADP algorithm based on Q-learning [10] is shown in Algorithm 1. Let $V^i(\mathcal{X})$ denote the values of all states $\mathbf{x}\in\mathcal{X}$ with the assumption that $\mathcal{X}$ is countable. The index i denotes the serial number of ith iteration. The algorithm starts with an initial guess of the value function, possibly generated by a potential function [30]. At the ith iteration, an initial state is randomly generated, and then a state trajectory is calculated by solving the Bellman equation given $V^{i-1}(\mathcal{X})$. After that, with the rewards obtained along the trajectory, the value of $V^i(\mathbf{x}(t))$ for each visited state can be calculated. At last, the value of $V^i(\mathbf{x}(t))$ is updated by the Q-learning method, as shown in Algorithm 1, where α_i is the learning rate, which is a function of i. Algorithm 1 can be extended to a continuous state space $\mathcal{X}$, by adopting a neural network [29] to approximate the value function, and can be used to solve an online optimal problem by replacing "Randomly choose initial state $\mathbf{x}_0^i$" with "The current state $\mathbf{x}$" following the Gauss-Seidel variation [10].

Algorithm 1 ADP algorithm based on Q-learning

```
Require: Initialize $V^i(\mathcal{X})$ and set $i=1$
  while $i \le N_{max}$ do
      Randomly choose initial state $\mathbf{x}_0^i$.
      Solve: $V^i[\mathbf{x}(\kappa)] = \max_{\mathbf{u}_\kappa}\{\mathscr{L}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)] + V^{i-1}[\mathbf{x}(\kappa+1)]\}$
      Record visited $\mathbf{x}^i(\kappa)$
      Update $V^i(\mathcal{X})$ as $V^i(\mathbf{x}) = \begin{cases} (1-\alpha_i)V^{i-1}(\mathbf{x})+\alpha_i V^i(\mathbf{x}) & \text{if } \mathbf{x}=\mathbf{x}^i(\kappa) \\ V^{i-1}(\mathbf{x}) & \text{otherwise} \end{cases}$
      i=i+1;
  end while
```

IV. Hybrid ADP Approach

This section presents new ADP optimality conditions, recursive relations, and transversality conditions for the optimal control of switched systems, formulated in Section II. The objective function (2) is minimized with respect to the continuous control law $\mathbf{u}(\cdot)$ *and* the discrete switching action $a(\cdot)$, over an infinite-time horizon $t\in(0\ \infty)$. Let the continuous optimal control law be denoted by $\mathbf{u}^*(\mathbf{x})$, and the optimal discrete switching action function be denoted by $a^*(\mathbf{x})$. For an initial state $\mathbf{x}_0$, the optimal sequence of switching instants is $\{0,\ t_1^*,\ \dots,t_i^*,t_{i+1}^*,\dots,\infty\}$, where $\xi^*(t)\neq\xi^*(t-)$, and the optimal switching mode sequence is $\{\xi_0^*,\ \xi_1^*,\dots,\xi_i^*,\dots,\xi_\infty^*\}$. At $t=0$, the optimal value function is denoted by $V^*[\mathbf{x}_0,\xi_0^*]$. At any $t>0$, the optimal continuous state is denoted by $\mathbf{x}^*(t)$, the optimal switching mode is denoted by $\xi^*(t)=\xi_i^*,\ t\in[t_i^*\ t_{i+1}^*)$, and, thus, the optimal value function is denoted by $V^*[\mathbf{x}^*(t),\xi_i^*]$. Then, the Bellman equation for the hybrid objective function in (2) can be written as

$$V^*[\mathbf{x}^*(t),\xi_i^*] = V^*[\mathbf{x}^*(t_{i+1}^*),\xi_{i+1}^*] + \int_t^{t_{i+1}^*}\mathscr{L}_{\xi_i^*}[\mathbf{x}^*(\tau),\mathbf{u}^*(\tau,\xi_i^*)]d\tau \quad (5)$$

When $t\in[t_i^*\ t_{i+1}^*)$ (no switch occurs during $[t_i^*\ t_{i+1}^*)$), the optimality conditions can be derived according the Pontryagin's minimum principle [31]. Let $[t_0^*\ t_{i+1}^*)$ be divided into N equal segments, and let $\kappa\in\{0,1,\dots,N\}$ represent the instant $(t_{i+1}^*-t_0^*)/N\times\kappa+t_0^*$. Equation (5) therefore can be approximated as

$$V^*[\mathbf{x}^*(\kappa),\xi_i^*] \approx V^*[\mathbf{x}^*(\kappa+1),\xi_i^*] + \frac{t_{i+1}^*-t_0^*}{N}\mathscr{L}_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa,\xi_i^*)] \quad (6)$$

After denoting $\frac{t_{i+1}^*-t_i^*}{N}\mathscr{L}_{\xi_i^*}[\cdot]$ as $\mathscr{L}_{\xi_i^*}^N[\cdot]$, (6) is rewritten as

$$V^*[\mathbf{x}^*(\kappa),\xi_i^*] = V^*[\mathbf{x}^*(\kappa+1),\xi_i^*] + \mathscr{L}_{\xi_i^*}^N[\mathbf{x}^*(\kappa),\mathbf{u}^*(\tau,\xi_i^*)]. \quad (7)$$

The optimality condition for the optimal controller $\mathbf{u}^*(\kappa),\ \forall\kappa\in\{0,1,\dots,N\}$ can be obtained by setting the derivative of the value function (7) regarding $\mathbf{u}^*$ as 0, such that,

$$\frac{\partial V^*[\mathbf{x}^*(\kappa+1),\xi_i^*]}{\partial\mathbf{x}^*(\kappa+1)}\frac{\partial\mathbf{x}^*(\kappa+1)}{\partial\mathbf{u}^*(\kappa)} + \frac{\partial\mathscr{L}_{\xi_i^*}^N[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial\mathbf{u}^*(\kappa)} = 0. \quad (8)$$

In order to solve (8), the value function gradient $\partial_{\mathbf{x}}V^*[\mathbf{x}^*(\kappa+1),\xi_i^*]$, computed by the critic network, is required, and its recursive relationship is obtained by taking derivative on both sides of (7). Let

$\boldsymbol{\lambda}^*[\mathbf{x}^*(\kappa),\xi_i^*] \triangleq \partial_{\mathbf{x}} V^*[\mathbf{x}^*(\kappa),\xi_i^*]$ for the remainder of the paper. Then, the critic recursive relationship can be derived as follows,

$$
\begin{aligned}
&\boldsymbol{\lambda}^*[\mathbf{x}^*(\kappa),\xi_i^*] = \frac{\partial V^*[\mathbf{x}^*(\kappa),\xi_i^*]}{\partial \mathbf{x}^*(\kappa)} = \\
&\frac{\partial \mathscr{L}^N_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial \mathbf{x}^*(\kappa)} + \frac{\partial V^*[\mathbf{x}^*(\kappa+1),\xi_i^*]}{\partial \mathbf{x}^*(\kappa)} \\
&= \frac{\partial \mathscr{L}^N_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial \mathbf{x}^*(\kappa)} + \frac{\partial \mathscr{L}^N_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial \mathbf{u}^*(\kappa)} \frac{\partial \mathbf{u}^*(\kappa)}{\partial \mathbf{x}^*(\kappa)} \\
&+ \frac{\partial V^*[\mathbf{x}^*(\kappa+1),\xi_i^*]}{\partial \mathbf{x}^*(\kappa+1)} \frac{\partial \mathbf{x}^*(\kappa+1)}{\partial \mathbf{x}^*(\kappa)} \\
&+ \frac{\partial V^*[\mathbf{x}^*(\kappa+1),\xi_i^*]}{\partial \mathbf{x}^*(\kappa+1)} \frac{\partial \mathbf{x}^*(\kappa+1)}{\partial \mathbf{u}^*(\kappa)} \frac{\partial \mathbf{u}^*(\kappa)}{\partial \mathbf{x}^*(\kappa)} \\
&= \frac{\partial \mathscr{L}^N_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial \mathbf{x}^*(\kappa)} + \frac{\partial \mathscr{L}^N_{\xi_i^*}[\mathbf{x}^*(\kappa),\mathbf{u}^*(\kappa)]}{\partial \mathbf{u}^*(\kappa)} \frac{\partial \mathbf{u}^*(\kappa)}{\partial \mathbf{x}^*(\kappa)} \\
&\quad + \boldsymbol{\lambda}^*[\mathbf{x}^*(\kappa+1),\xi_i^*] \frac{\partial \mathbf{x}^*(\kappa+1)}{\partial \mathbf{x}^*(\kappa)} \\
&\quad + \boldsymbol{\lambda}^*[\mathbf{x}^*(\kappa+1),\xi_i^*] \frac{\partial \mathbf{x}^*(\kappa+1)}{\partial \mathbf{u}^*(\kappa)} \frac{\partial \mathbf{u}^*(\kappa)}{\partial \mathbf{x}^*(\kappa)}. \qquad (9)
\end{aligned}
$$

According to optimality conditions for hybrid optimal control problems [17], the optimal discrete action of the switched system in Section II obeys

$$
\begin{aligned}
a^*(t) = \underset{\xi}{\operatorname{argmin}} \{&\boldsymbol{\lambda}^*[\mathbf{x}^*(t),\xi] f_\xi[\mathbf{x}^*(t),\mathbf{u}^*(t)] \\
&+ \mathscr{L}_\xi[\mathbf{x}^*(t),\mathbf{u}^*(t)]\} \qquad (10)
\end{aligned}
$$

Thus, when $t = t_i^* \in \{t_1^*,\ldots,t_\infty^*\}$ (a switch occurs during at t_i^*), the optimal value function must satisfy the following transversality condition

$$
V^*[\mathbf{x}^*(t_{i+1}^*),\xi_i] = V^*[\mathbf{x}^*(t_{i+1}^*),\xi_{i+1}^*]. \qquad (11)
$$

By differentiating both sides of (11), the transversality condition for the critic network is obtained from (11), i.e.:

$$
\begin{aligned}
\boldsymbol{\lambda}^*[\mathbf{x}^*(t_{i+1}^*),\xi_i^*] &= \frac{\partial V^*[\mathbf{x}^*(t_{i+1}^*),\xi_i^*]}{\partial \mathbf{x}^*(t_{i+1}^*)} \\
&= \frac{\partial V^*[\mathbf{x}^*(t_{i+1}^*),\xi_{i+1}^*]}{\partial \mathbf{x}^*(t_{i+1}^*)} \\
&= \boldsymbol{\lambda}^*[\mathbf{x}^*(t_{i+1}^*),\xi_{i+1}^*] \qquad (12)
\end{aligned}
$$

Furthermore, the optimal switching time t_i^* can be determined from the recursive relationship

$$
\begin{aligned}
&\boldsymbol{\lambda}^*[\mathbf{x}^*(t_i^*),\xi_i^*] f_{\xi_i^*}[\mathbf{x}^*(t_i^*),\mathbf{u}^*(t_i^*)] + \mathscr{L}_{\xi_i^*}[\mathbf{x}^*(t_i^*),\mathbf{u}^*(t_i^*)] \\
&= \boldsymbol{\lambda}^*[\mathbf{x}^*(t_i^*-),\xi_{i-1}^*] f_{\xi_{i-1}^*}[\mathbf{x}^*(t_i^*-),\mathbf{u}^*(t_i^*-)] \\
&\quad + \mathscr{L}_{\xi_{i-1}^*}[\mathbf{x}^*(t_i^*-),\mathbf{u}^*(t_i^*-)] \qquad (13)
\end{aligned}
$$

From above analysis, the optimality conditions in (8), (9), (10), (12), and (13) are to be solved simultaneously to obtain the optimal continuous control $\mathbf{u}^*(\cdot)$, discrete action $a^*(\cdot)$ and switching times $\{0,\ t_1^*,\ \ldots,t_i^*,t_{i+1}^*,\ldots,\infty\}$, and switching sequence $\{\xi_0^*,\ \xi_1^*,\ldots,\xi_i^*,\ldots,\xi_\infty^*\}$. In order to reduce the computational complexity associated with the numerical solution of these optimality conditions, learning rules for the hybrid-ADP critic and control networks are derived in the remainder of this section. Since the critic and control networks consist of continuous and discrete variables, a separate neural network is used to approximate the control or critic function for each mode ξ, such that $2E$ neural networks are implemented for the actor and the critic. Let NN_λ^ξ denote the *critic network* used to approximate $\boldsymbol{\lambda}^*(\mathbf{x},\xi)$, and NN_u^ξ denote the *control* (or actor) network used to approximate $\mathbf{u}^*(\mathbf{x},\xi)$.

When $\xi(\kappa) = \xi(\kappa+1)$, the control neural network under the mode $\xi(\kappa)$, NN_u^ξ, is updated by the actor recurrence relationship

$$
\begin{aligned}
\Delta w_u = &-\eta\{\frac{\partial \mathscr{L}^N_{\xi_i}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)]}{\partial \mathbf{x}(\kappa)} \\
&- \frac{\partial \mathbf{x}(\kappa+1)}{\partial \mathbf{x}(\kappa)} \boldsymbol{\lambda}[\mathbf{x}(\kappa+1),\xi_i]\} \frac{\mathbf{u}[\mathbf{x}(\kappa),\xi_i]}{\partial w_u} \qquad (14)
\end{aligned}
$$

While holding the control network fixed, the critic neural network under the mode $\xi(\kappa)$, NN_λ^ξ, is updated by the critic recurrence relationship,

$$
\begin{aligned}
\Delta w_u = &-\epsilon\{\frac{\partial \mathbf{x}(\kappa+1)}{\partial \mathbf{u}(\kappa)} \boldsymbol{\lambda}[\mathbf{x}(\kappa+1),\xi_i] \\
&- \frac{\partial \mathscr{L}^N_{\xi_i}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)]}{\partial \mathbf{u}(\kappa)}\} \frac{\mathbf{u}[\mathbf{x}(\kappa),\xi_i]}{\partial w_\lambda} \qquad (15)
\end{aligned}
$$

where, the learning rates η and ϵ are user-defined parameters.

When $\xi(\kappa) \neq \xi(\kappa+1)$, according to equation (12), the control neural network of the mode $\xi(\kappa)$, NN_u^ξ, is updated by the actor recurrence relationship,

$$
\begin{aligned}
\Delta w_u = &-\eta\{\frac{\partial \mathscr{L}^N_{\xi_i}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)]}{\partial \mathbf{x}(\kappa)} \\
&- \frac{\partial \mathbf{x}(\kappa+1)}{\partial \mathbf{x}(\kappa)} \boldsymbol{\lambda}[\mathbf{x}(\kappa+1),\xi_{i+1}]\} \frac{\mathbf{u}[\mathbf{x}(\kappa),\xi_i]}{\partial w_u} \\
&\qquad (16)
\end{aligned}
$$

While holding the control network fixed, the critic neural network of the mode $\xi(\kappa)$, NN_λ^ξ, is updated by critic recurrence relationship,

$$
\begin{aligned}
\Delta w_\lambda = &-\epsilon\{\frac{\partial \mathbf{x}(\kappa+1)}{\partial \mathbf{u}(\kappa)} \boldsymbol{\lambda}[\mathbf{x}(\kappa+1),\xi_{i+1}] \\
&- \frac{\partial \mathscr{L}^N_{\xi_i}[\mathbf{x}(\kappa),\mathbf{u}(\kappa)]}{\partial \mathbf{u}(\kappa)}\} \frac{\boldsymbol{\lambda}[\mathbf{x}(\kappa),\xi_i]}{\partial w_\lambda} \qquad (17)
\end{aligned}
$$

and the discrete action $a(t)$ is updated by the recurrence relationship,

$$a(t) = \underset{\xi}{\operatorname{argmin}}\, \boldsymbol{\lambda}[\mathbf{x}(t), \xi] f_\xi[\mathbf{x}(t), \mathbf{u}(t)] + \mathscr{L}_\xi[\mathbf{x}(t), \mathbf{u}(t)] \tag{18}$$

where $\mathbf{u}$ and $\boldsymbol{\lambda}$ are evaluated from the (fixed) control and critic neural networks. The learning rules (14), (15), (17), (16), and (18) only need to evaluate f_ξ and $\mathscr{L}_\xi$ in $\mathcal{N}(\mathbf{x}(t))$, which is consistent with the Assumption (2).

All of the hybrid-ADP recurrence relationships derived in this section are implemented iteratively over time, such that the optimal continuous control law, mode switching sequence, switching instants, and value function are determined from observations of the switched system state. In the next section, the proposed hybrid-ADP approach is demonstrated through a linear, quadratic switched optimal control problem for which there exists an analytical solution to be compared to the hybrid-ADP solution presented in this section.

V. NUMERICAL SIMULATIONS

The hybrid-ADP approach presented in the previous section can be applied to nonlinear switched systems in the form described in Section II, for which linear quadratic regulator (LQR) or analytical solutions may not be available. However, in order to demonstrate the effectiveness of the hybrid-AD solution, this section considers the optimal control of a hybrid dynamical system with linear continuous dynamics, and quadratic (hybrid) objective function, for which an analytical solution can be obtained via Riedinger's method [17].

The autonomous hybrid system consists of two power systems, one gasoline-driven, and one electric-driven, that each live in a one-dimensional workspace $\mathcal{W} \subset \mathbb{R}$, and to be represented by a continuous state $\mathbf{x} = [x\, \dot{x}]^T \in \mathbb{R}^2$, where $x \in \mathcal{W}$, and $\mathbf{x}$ is fully observable and error free. It is assumed that the system mode can switch to any of the two power systems, at any time, and that the two power systems are independent and supplied with sufficient fuel. The agent starts at a predefined state $\mathbf{x}_0$, and seeks to move to another predefined goal state $\mathbf{x}_g$.

When the gasoline-driven power system is chosen ($\xi = 1$), its dynamics are modeled by the system of equations,

$$\dot{\mathbf{x}}(t) = \mathbf{A}_{\xi(t)}\mathbf{x}(t) + \mathbf{B}_{\xi(t)}\mathbf{u}(t) \tag{19}$$

$$\xi(t) = 1, \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{20}$$

where $\mathbf{u} \in \mathbb{R}^2$ is the agent continuous control input, $\mathbf{x}_0$ is the agent initial state, $\mathbf{A}_1 = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}$, and $\mathbf{B}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. When the electric-driven power system is chosen ($\xi = 2$), its dynamics are modeled by the system of equations,

$$\dot{\mathbf{x}}(t) = \mathbf{A}_{\xi(t)}\mathbf{x}(t) + \mathbf{B}_{\xi(t)}\mathbf{u}(t) \tag{21}$$

$$\xi(t) = 2, \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{22}$$

where $\mathbf{A}_2 = \begin{pmatrix} 0 & 1 \\ -1 & -0.5 \end{pmatrix}$, $\mathbf{B}_2 = \begin{pmatrix} 0 \\ 0.8 \end{pmatrix}$. The mode $\xi(t)$ was fully controlled by a switching signal $a(t)$.

The overall system performance depends on the switching sequence, and on the continuous control laws, and is defined as,

$$J = \int_0^\infty \mathbf{x}^T \mathbf{Q}_{\xi(t)}\mathbf{x} + \mathbf{u}^T \mathbf{R}_{\xi(t)}\mathbf{u} dt, \quad \xi(t) = 1, 2 \tag{23}$$

where $\mathbf{Q}_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix}$, $\mathbf{Q}_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.4 \end{pmatrix}$, $\mathbf{R}_1 = 1$, and $\mathbf{R}_2 = 1$

Adopting Riedinger's approach [17], when the discrete time step used to simulate the system is 0.05 (s), the exact analytical solution to the optimal control problem of this hybrid mobile agent has a cyclic switching sequence such that

1) An optimal switch from mode 1 to mode 2 occurs when $\dot{x} = 0.85x$, and the following optimal continuous control is given by

$$\mathbf{P}_2 = \begin{pmatrix} 0.88 & 0.25 \\ 0.25 & 0.66 \end{pmatrix} \tag{24}$$

$$\mathbf{u} = -(\mathbf{R}_2 + \mathbf{B}_2^T \mathbf{P}_2 \mathbf{B}2)^{-1} \mathbf{B}_2^T \mathbf{P}_2 \mathbf{A}_2 \mathbf{x} \tag{25}$$

2) An optimal switch from mode 2 to mode 1 takes place when $\dot{x} = -1.25x$, and the following optimal continuous control is given by

$$\mathbf{P}_1 = \begin{pmatrix} 0.95 & 0.24 \\ 0.24 & 0.60 \end{pmatrix} \tag{26}$$

$$\mathbf{u} = -(\mathbf{R}_1 + \mathbf{B}_1^T \mathbf{P}_1 \mathbf{B}1)^{-1} \mathbf{B}_1^T \mathbf{P}_1 \mathbf{A}_1 \mathbf{x} \tag{27}$$

At time $t = 0$, the critic network is trained to satisfy the following initial guess of $\boldsymbol{\lambda}$ for both power modes,

$$\boldsymbol{\lambda} = (\mathbf{x} - \mathbf{x}_g) \tag{28}$$

which leads the hybrid system to $\mathbf{x}_g$. At the same time, the control network for each mode is trained according to

$$\mathbf{u}_\xi = -(R_\xi dt + dt^2 B_\xi^T B_\xi)^{-1}[dt B_\xi^T((I + A_\xi dt)\mathbf{x} - \mathbf{x}_g)] \tag{29}$$

to satisfy (8) given (28). Subsequently, the hybrid-ADP recursive relationships presented in Section IV are used to adapt the critic and control networks online, while the same networks are used to control the power system. When the system state arrives at the goal state $\mathbf{x}_g$, with a tolerance of 0.01 (m), the task of bringing the system from $\mathbf{x}_0$ to $\mathbf{x}_g$ is repeated, and the critic and control network are trained to learn the optimal solution online, without knowledge of the system models in (19)-(22). Each of the learning tasks is referred to as a trial, and learning is conducted over several trials, until the recurrence relationships are satisfied within a desired tolerance.

The learning rate η and ϵ were chosen equal to 5×10^{-6}. Both the critic and control neural networks had two hidden layers with 20 neurons in each layer, and their transfer functions were hyperbolic tangent sigmoid functions. In this simulation, the critic and control neural networks were initialized using (28) and (29), and the initial system state was $\mathbf{x}_0 = [1.0\ -0.6]^T$ (m), while the goal state was $\mathbf{x}_g = [0\ 0]^T$ (m). The simulation results are summarized in Figs. 1-2.

As shown in Fig. 1, the value of the objective function, J, declined after each trial; it decreased by 7.7% after 5 trials, by 13.7% after 50 trials, and by 21.7% after 385 trials. After 385 trials, the difference between J and the value of the objective function corresponding to the analytical solution, J^*, is less than 0.02. As shown in Fig. 1, from the 175th trial to the 182th trial, the total reduction of J was $1.2e-4$, while the reduction was $2.5e-3$ at the 183th trial. Such a relatively high reduction was brought by changing the switching instant and switching mode sequence. The changes of switching sequence and instant were caused by the accumulated learning of the critic and control neural networks during previous trials. The learning and accuracy of these networks were crucial to obtaining the correct switching sequence and instants.

As a comparison, the state trajectories obtained from the analytical solution are also plotted in Fig. 2, using a dashed line. The state trajectories obtained during each trial by the hybrid-ADP are shown in Fig. 2, using a solid line. The trajectories obtained while the system is in gasoline-driven mode are shown in red, and those obtained while the system is in electric-driven mode are shown in blue. The switching mode and instants can be identified by the change in color, along each trajectory. As can be seen from the ‘Initial’ hybrid-ADP trajectory in Fig. 2, the initial critic and control neural networks gave an incorrect sequence of the switching mode and instants, and incorrect control laws, thereby yielding the high initial cost in Fig. 1. By applying the hybrid-ADP approach, the system was capable of updated the critic and control networks to minimize the cost along each of its trajectories, in an online fashion. As a result of hybrid-ADP learning, after the 5th trial, the system changed its switching mode sequence to one that starts out by operating under the second (electric) mode, and then switches to gasoline. As shown in Fig. 2, at the 50th trial, the system switched its mode three times instead of only one time during the 5th trial, and at the 385th trial the system has learned the optimal switching sequence.

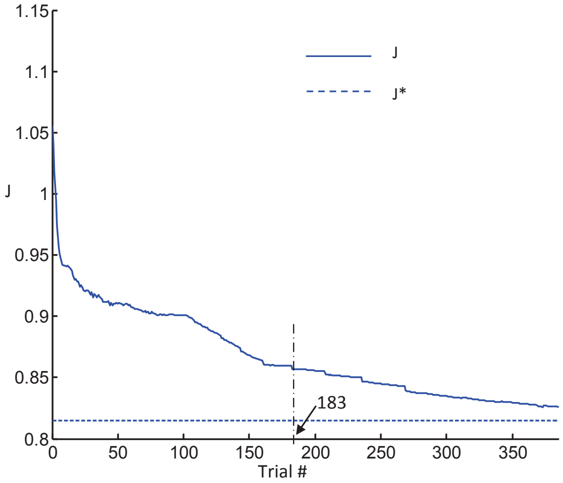

Fig. 1. Objective function optimization

Different from the previous example, the switched system schematized in Fig. 3 consists of three subsystems and its continuous controller is a 2 dimensional vector function. The matrices of defining dynamic equations and the cost term in the objective function are given as follows.

$$\mathbf{A}_1 = \begin{pmatrix} -1 & 4 \\ -3 & 2 \end{pmatrix}, \ \mathbf{A}_2 = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, \ \mathbf{A}_3 = \begin{pmatrix} -3 & 1 \\ -3 & -1 \end{pmatrix}$$
$$\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$
$$\mathbf{Q}_1 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \ \mathbf{Q}_2 = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \ \mathbf{Q}_3 = \begin{pmatrix} 2 & 0 \\ 0 & 5 \end{pmatrix}$$
$$\mathbf{R}_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.25 \end{pmatrix}, \ \mathbf{R}_2 = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{R}_3 = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix} \tag{30}$$

In this simulation the learning rate η and ϵ were chosen equal to 5×10^{-6}. Both the critic and control neural networks had two hidden layers with 20 neurons in each layer, and their transfer functions were hyperbolic tangent sigmoid functions. The critic and control neural networks were initialized using (28) and (29), and the initial system state was $\mathbf{x}_0 = [-0.2\ -1]^T$ (m), while the goal state was $\mathbf{x}_g = [0\ 0]^T$ (m). The simulation results are summarized in Fig. 4.

By applying the hybrid-ADP approach, the system was capable of updated the critic and control networks to minimize the cost along each of its trajectories, in an online fashion. As shown in Fig. 4, from the first trial to the 168th trial, the value of the objective function oscillated and did not decrease because that during this period the critic networks and control networks were updated based the accumulated learning of the switched system, and that the accumulated learning was

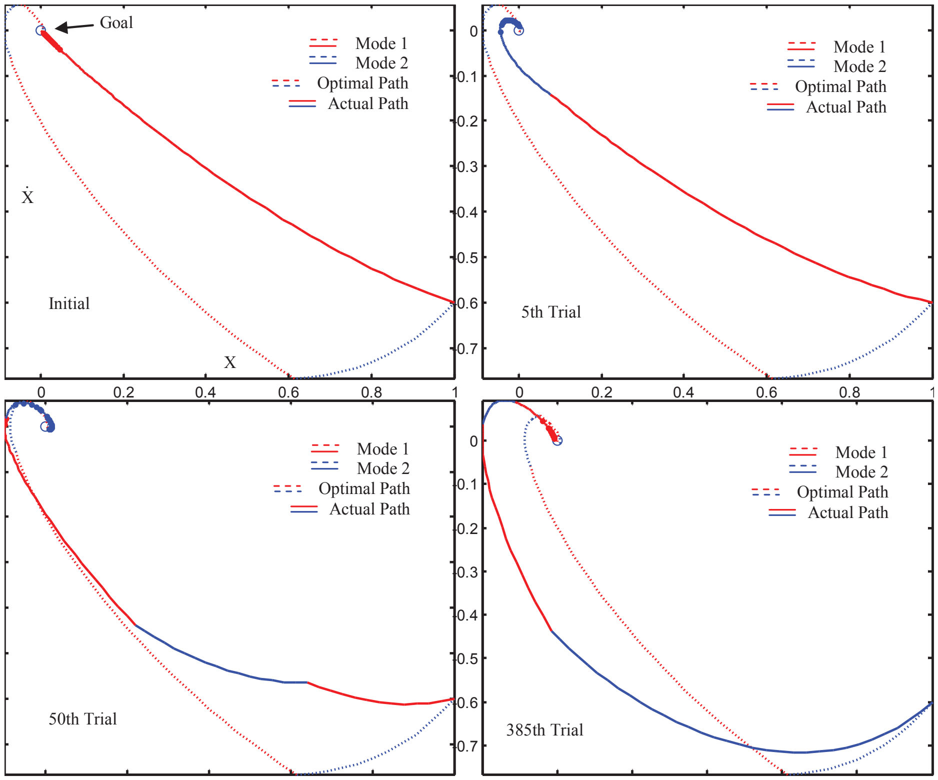

Fig. 2. State trajectory optimization for four trials.

not sufficient enough to have a correct switch. Then, a dramatic decrease of the value function occurred at the 169th trial, and thereafter converged to 0.7376 which is close to the optimal value.

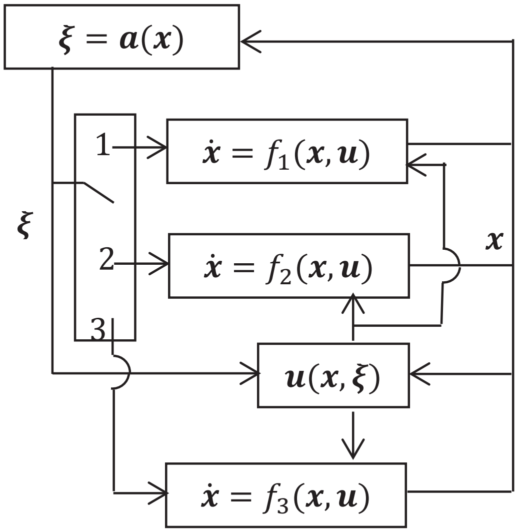

Fig. 3. Switched System with Three Subsystems

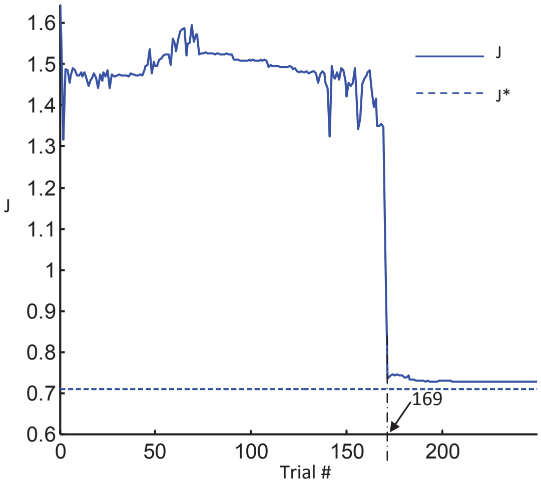

Fig. 4. Objective function optimization

VI. Conclusions and Future Work

The advantages of the switched system allow hybrid models to characterize the modern autonomous systems with discrete actions and continuous control. The hybrid-ADP presented in this paper can learn the optimal continuous control, and switching mode sequence and instants online. Due to the online nature of ADP, the actions and controls can adapt to the uncertainties of the environment and the hybrid system. The control and critic neural networks learn environment online while retaining their baseline performance. The switching sequence and instants were calculated based on the updated critic and control neural networks. The proposed hybrid-ADP focuses on exploiting the current knowledge of the critic and control networks, and it is myopic in terms of exploring the state-control space. Future work will focus on accelerating the hybrid-ADP convergence by investigating the balance between exploration and exploitation over repeated trials.

References

[1] Z. Sun and S. Ge, *Switched Linear Systems: Control and Design*, ser. Communications and Control Engineering. Springer, 2005.

[2] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, "Optimal control of continuous-time switched affine systems," *Automatic Control, IEEE Transactions on*, vol. 51, no. 5, pp. 726 – 741, may 2006.

[3] S. Ferrari, R. Fierro, and D. Tolic, "A geometric optimization approach to tracking maneuvering targets using a heterogeneous mobile sensor network," in *Proc. of the 2009 Conference on Decision and Control*, Cancun, MX, 2009.

[4] R. Fierro and F. Lewis, "A framework for hybrid control design," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 27, no. 6, pp. 765 –773, nov 1997.

[5] M. Zavlanos and G. Pappas, "Distributed hybrid control for multiple-pursuer multiple-evader games," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds., 2007, vol. 4416, pp. 787–789.

[6] R. Sanfelice and E. Frazzoli, "A hybrid control framework for robust maneuver-based motion planning," in *American Control Conference, 2008*, june 2008, pp. 2254 –2259.

[7] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: model and optimal control theory," *Automatic Control, IEEE Transactions on*, vol. 43, no. 1, pp. 31 –45, jan 1998.

[8] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, p. 17091721, 2005.

[9] S. Hedlund and A. Rantzer, "Convex dynamic programming for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 9, p. 1536, 2002.

[10] W. B. Powell, *Approximate Dynamic Programming : Solving the Curses of Dimensionality*. Wiley-Interscience, 2007.

[11] J. Murray, C. Cox, G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 32, no. 2, pp. 140 – 153, may 2002.

[12] S. Ferrari and R. Stengel, "On-line adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 777–786, 2004.

[13] S. Ferrari and R. Stengel, "Model-based adaptive critic designs," in *Learning and Approximate Dynamic Programming*, J. Si, A. Barto, and W. Powell, Eds. John Wiley and Sons, 2004.

[14] G. Lai, F. Margot, and N. Secomandi, "An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation," *Oper. Res.*, vol. 58, no. 3, pp. 564–582, May 2010.

[15] L. Gao and Z. Zhang, "Approximate dynamic programming approach to network-level budget planning and allocation for pavement infrastructure," in *Transportation Research Board 88th Annual Meeting*, 2009.

[16] W. P. Hugo Simao, "Approximate dynamic programming for management of high-value spare parts," *Journal of Manufacturing Technology Management*, vol. 20, pp. 147 – 160, May 2009.

[17] P.Riedinger, F.Kratz, C. Iung, and C. Zane, "Linear quadratic optimization for hybrid systems," in *Proceedings of the 38th CDC*, 1999.

[18] L. Christos, Cassandras John, *Stochastic Hybrid Systems*. CRC Press, 2006.

[19] X. Xu and P. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 2 – 16, jan. 2004.

[20] R. A. Howard, *Dynamic Programming and Markov Processes*. The M.I.T. Press, 1960.

[21] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ: Princeton University Press, 1962.

[22] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear hjb solution using approximate dynamic programming: Convergence proof," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 4, pp. 943 – 949, aug. 2008.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vols. I and II*. Belmont, MA: Athena Scientific, 1995.

[24] J. SI, *Handbook of learning and approximate dynamic programming*. Hoboken, NJ, IEEE Press, 2007.

[25] M. Fox, M. Ghallab, G. Infantes, and D. Long, "Robot introspection through learned hidden markov models," *Artificial Intelligence*, vol. 170, pp. 59–113, 2006.

[26] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions (Encyclopedia of Mathematics and its Applications)*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.

[27] T. Kollar and N. Roy, "Trajectory optimization using reinforcement learning for map exploration," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008.

[28] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

[29] D. F. Specht, "Applications of probabilistic neural networks," *Proceedings of SPIE*, vol. 1294, pp. 344–353, 1990.

[30] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[31] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mischenko, *The Mathematical Theory of Optimal Processes*. New York: Interscience, 1962.