Theoretical and Numerical Analysis of Approximate Dynamic Programming with Approximation Errors

Ali Heydari*

South Dakota School of Mines and Technology, Rapid City, South Dakota 57701

DOI: 10.2514/1.G001154

This study is aimed at answering the question of how the approximation errors at each iteration of approximate dynamic programming affect the quality of the final results, considering the fact that errors at each iteration affect the next iteration. To this goal, convergence of value iteration scheme of approximate dynamic programming for deterministic nonlinear optimal control problems with discrete-time known dynamics subject to an undiscounted known cost functions is investigated while considering the errors existing in approximating respective functions. The boundedness of the results of approximate solution is obtained based on that quantities are known in a general optimal control problem and assumptions that are verifiable. Moreover, because the presence of the approximation errors lead to the deviation of the results from optimality, sufficient conditions for stability of the system operated by the result obtained after a finite number of value iterations, along with an estimation of its region of attraction, are derived in terms of a calculable upper bound of the control approximation error. Finally, the process of implementation of the method on an orbital maneuver problem is investigated, through which the assumptions made in the theoretical developments are verified and the sufficient conditions are applied for guaranteeing stability and near optimality.

Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_r$</td>
<td>set ${x \in \mathbb{R}^n: \tilde{h}(x) \leq r}$ for some $r \in \mathbb{R}_+$</td>
</tr>
<tr>
<td>$c$</td>
<td>positive constant real number</td>
</tr>
<tr>
<td>$f(\ldots)$</td>
<td>dynamics of the system subject to control</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>feedback control policy</td>
</tr>
<tr>
<td>$h^{*}(\cdot)$</td>
<td>optimal feedback control policy for $J$</td>
</tr>
<tr>
<td>$\tilde{h}(\cdot)$</td>
<td>feedback control policy calculated based on $\tilde{h}$</td>
</tr>
<tr>
<td>$\tilde{h}^{*}(\cdot)$</td>
<td>approximation of $h^{*}(\cdot)$</td>
</tr>
<tr>
<td>$i$</td>
<td>iteration index</td>
</tr>
<tr>
<td>$J$</td>
<td>cost function with running cost $U(x, u)$</td>
</tr>
<tr>
<td>$J^i$</td>
<td>cost function with running cost $U(x, u) + cU(x, 0)$</td>
</tr>
<tr>
<td>$k$</td>
<td>time index</td>
</tr>
<tr>
<td>$L_V$</td>
<td>Lipschitz constant of $\tilde{h}$</td>
</tr>
<tr>
<td>$m$</td>
<td>number of control elements</td>
</tr>
<tr>
<td>$N$</td>
<td>final time</td>
</tr>
<tr>
<td>$n$</td>
<td>number of state elements</td>
</tr>
<tr>
<td>$n_c$</td>
<td>number of basis functions in the critic</td>
</tr>
<tr>
<td>$n_a$</td>
<td>number of basis functions in the actor</td>
</tr>
<tr>
<td>$Q(\cdot)$</td>
<td>state penalizing term</td>
</tr>
<tr>
<td>$R$</td>
<td>control penalizing weight</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>set of real numbers</td>
</tr>
<tr>
<td>$\mathbb{R}_+$</td>
<td>set of nonnegative real numbers</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>reference length for nondimensionalization</td>
</tr>
<tr>
<td>$T$</td>
<td>reference time for nondimensionalization</td>
</tr>
<tr>
<td>$U(x, u)$</td>
<td>running cost given state $x$ and control $u$</td>
</tr>
<tr>
<td>$u_k$</td>
<td>control vector at time $k$</td>
</tr>
<tr>
<td>$u_X$</td>
<td>$x$ component of the force on the spacecraft</td>
</tr>
<tr>
<td>$u_Y$</td>
<td>$y$ component of the force on the spacecraft</td>
</tr>
<tr>
<td>$V_h(x)$</td>
<td>value function of control policy $h(\cdot)$, given current state $x$</td>
</tr>
<tr>
<td>$V^{*}(x)$</td>
<td>optimal value function for $J$ given current state $x$</td>
</tr>
<tr>
<td>$V^i(x)$</td>
<td>exact value function at the $i$th iteration of value iteration</td>
</tr>
<tr>
<td>$\nabla V^i(x)$</td>
<td>exact value function at the $i$th iteration of value iteration subject to $J$</td>
</tr>
<tr>
<td>$\nabla V^{*}(x)$</td>
<td>optimal value function for $J$</td>
</tr>
<tr>
<td>$\nabla V^{*}(x)$</td>
<td>approximate value function at the $i$th iteration of approximate value iteration</td>
</tr>
<tr>
<td>$W_c$</td>
<td>weight matrix for the critic</td>
</tr>
<tr>
<td>$W_a$</td>
<td>weight matrix for the actor</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$ component of the position of spacecraft in the orbital frame</td>
</tr>
<tr>
<td>$y$</td>
<td>$y$ component of the position of spacecraft in the orbital frame</td>
</tr>
<tr>
<td>$\delta(\cdot)$</td>
<td>tolerance for evaluation of convergence of approximation value iteration</td>
</tr>
<tr>
<td>$\epsilon(\cdot)$</td>
<td>approximation error between $\nabla V^{*}(x)$ and $\nabla V_{h}(x)$</td>
</tr>
<tr>
<td>$\mu(x)$</td>
<td>approximation error between $h^{*}(\cdot)$ and $h(\cdot)$</td>
</tr>
<tr>
<td>$\mu_E$</td>
<td>gravitational parameter for the Earth</td>
</tr>
<tr>
<td>$\sigma(\cdot)$</td>
<td>basis functions for the actor</td>
</tr>
<tr>
<td>$\phi(\cdot)$</td>
<td>basis functions for the critic</td>
</tr>
</tbody>
</table>

I. Introduction

Approximate (or adaptive) dynamic programming (ADP) or reinforcement learning (RL) has been investigated extensively by different researchers as a powerful tool for approximating solutions to mathematically intractable problems seeking optimum [1–13]. ADP has shown its great potential in aerospace applications as well [5,14–18], from control of agile missiles to spacecraft rendezvous. A popular algorithm for ADP is value iteration (VI) [3,19,20]. Considering optimal control problems with continuous state and action spaces, which are the subject of this work, the convergence proof of VI for linear systems was analyzed in [21,22].
As for nonlinear systems, the convergence was established by different researchers including [9] (whose idea was adapted in [10,11,23] by other authors), [34], and [25] through different approaches. All these convergence analyses are based on the assumption of perfect function reconstruction (i.e., no error in the function approximation). Although this assumption helps in deriving the results, it restricts their validity severely because the approximation errors exist almost in every application when the system is nonlinear or when the cost function terms are nonquadratic and nonlinear. What makes their presence potentially problematic is the fact that the errors propagate throughout the iterations; hence, regardless of how small the errors are, a phenomenon similar to resonance might happen, which could lead to the complete unreliability of the results.

Analyzing VI under the presence of approximation errors, i.e., approximate VI (AVI), is an open research problem with a few results, including [4,26–31], to the best of the knowledge of the author. Bertsekas and Tsitsiklis [4], Singh and Yee [26], Farahmand et al. [27], and Munos and Szepesvári [28] investigate problems with discounted cost functions, and the results are solely valid for such problems, prevalent in computer science. As a matter of fact, any “forgetting” nature of discounted problems helps in the development of the error bounds, and if the discount factor approaches 1, as in typical infinite-horizon optimal control problems, the bounds go to infinity; hence, the results fail.

On the other hand, the results in [29], adapted in [30,31], provide error analyses but with some assumptions whose verification is not straightforward. For example, the approximation error between the exact and the approximate quantities should be possible to be written in a multiplicative form, instead of an additive form. Moreover, the boundedness results are conditional upon a constant being upper-bounded by a term that depends on the optimal value function. Because the optimal value function is not known, finding the desired approximation accuracy is not straightforward. As for non-VI based approaches in which the approximation errors are not neglected, interested readers are referred to [32–34]. These studies provide stability of the closed-loop system in an online learning scheme considering the approximation errors. This is conducted through deriving a Lyapunov function involving the weights of the function approximators. An advantage of these works is not requiring perfect knowledge of the internal dynamics of the system, with some disadvantages including requiring persistency of excitation in online operation as well as a stabilizing initial control policy. Finally, interested readers are referred to [35] for a convergence analysis from a different perspective, where the convergence of gradient descent is used along with establishing an analogy between ADP and back-propagation through time to show the convergence to a local minimum. Also, Fairbank and Alonso [36] might be of interest to the interested readers, in which an example on the divergence of VI with function approximators is presented, when no constraint/condition is enforced on the size of the approximation errors.

Moreover, the important concern that a neurocontroller is valid only when the state trajectory remains within the domain for which the controller is trained is addressed through finding an estimation of the region of attraction (ROA) for the result obtained through the AVI. It should be noted that, in the general case, if a neurocontroller is trained for a given domain, it is not guaranteed that any state trajectory initiated from the domain remains inside the domain. If it exits the domain, then the neurocontroller becomes invalid. An estimation of the ROA is found in this work, so that as long as the system’s initial condition is within the region, it is guaranteed that the entire trajectory remains in the region; hence, the neurocontroller will remain valid for control.

The rest of this study is organized as follows. The optimal control problem is presented in Sec. II, and the exact value iteration scheme is revisited in Sec. III. Section IV presents the approximate value iteration, followed by the theoretical analyses in Sec. V. Afterward, a famous aerospace example is numerically investigated in Sec. VI. Finally, concluding remarks are given in Sec. VII.

II. Optimal Control Problem

Let the system subject to control be given by the discrete-time nonlinear dynamics:

\[ x_{k+1} = f(x_k, u_k), \quad k \in \mathbb{N} \]  

where \( f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) is a smooth function versus both of its inputs (i.e., the state and control vectors, \( x \) and \( u \), respectively), with \( f(0,0) = 0 \). The set of nonnegative integers is denoted with \( \mathbb{N} \), and positive integers \( n \) and \( m \) denote the dimensions of the continuous state and control spaces. Moreover, subscript \( k \) represents the discrete time index. The performance index is given by

\[ J = \sum_{k=0}^{\infty} U(x_k, u_k) \]  

where \( U(x_k, u_k) := Q(x_k) + u_k^T R u_k \) for a convex and smooth positive-definite function \( Q : \mathbb{R}^n \to \mathbb{R}_+ \) and a positive-definite \( m \times m \) real matrix \( R \). The set of nonnegative real numbers is denoted with \( \mathbb{R}_+ \). Denoting a feedback control policy with \( h : \mathbb{R}^n \to \mathbb{R}^m \) for control calculation, i.e., \( u_k = h(x_k) \), the problem is finding the control policy that minimizes cost function (2) subject to dynamics (1). The control policy which leads to such a characteristic is called optimal control policy, denoted with \( h^*(\cdot) \).

**Definition 1:** A control policy \( h(\cdot) \) is defined to be admissible within a compact set if it asymptotically stabilizes the system for any initial state within the set [37], and the respective “cost-to-go” or “value function” starting from any state \( x_0 \) in the set, denoted with \( V_h : \mathbb{R}^n \to \mathbb{R}_+ \) and defined by

\[ V_h(x_0) = \sum_{k=0}^{\infty} U(x_k^h, h(x_k^h)) \]  

is upper-bounded, where \( x_k^h := f(x_{k-1}^h, h(x_{k-1}^h)), \forall k \in \mathbb{N} - \{0\} \) and \( x_0^h := x_0 \). In other words, \( x_k^h \) denotes the \( h \)th element on the state trajectory/history initiated from \( x_0 \) and propagated using control policy \( h(\cdot) \).

III. Exact Value Iteration

Based on Eq. (3), it can be seen that a value function satisfies the recursive relation given by

\[ V_h(x) = U(x, h(x)) + V_h(f(x, h(x))), \quad \forall x \in \mathbb{R}^n \]  

Defining the optimal value function as the value function associated with the optimal control policy and denoting it with \( V^{*}(\cdot) \), the Bellman equation [38], given next, provides the solution to the problem:
h^*(x) \in \arg \min_{u \in \mathcal{U}} \left( U(x, u) + V^i(f(x, u)) \right) \quad (5)

V^i(x) = \min_{u \in \mathcal{U}} \left( U(x, u) + V^i(f(x, u)) \right) \quad (6)

Note that the minimizing \( u \) in Eq. (5) may not be unique, and the notation \( \in \) used here (motivated by [39]) allows selecting any of the minimizers. Interested readers are referred to [25], in which sufficient conditions for uniqueness of the minimizer is established for control affine systems. Because of the so-called curse of dimensionality [38], however, the proposed solution is mathematically impracticable for general nonlinear systems. ADP uses the idea of approximating the optimal value function, either using lookup tables or function approximators (e.g., neural networks) for remedying the problem. The optimal value function approximator is called the critic in the ADP/RL literature. The approximation is performed over a compact and connected set containing the origin, called the domain of interest. This domain, denoted with \( \Omega \), has to be selected based on the specific problem at hand, and it should be noted that the ADP based results are valid only if the entire state trajectory initiated from the initial state vector remains within the domain for which the value function is approximated. The following assumption plays a major role in the ability to solve the optimal control problem.

Assumption 1: There exists at least one (possibly unknown) admissible control policy for the given system within the compact set \( \Omega \subset \mathbb{R}^n \) containing the origin.

This assumption guarantees that there is no state vector in \( \Omega \) for which the value function associated with the optimal control policy is infinite because, otherwise, the optimal control policy will not be “optimal” compared with the admissible control policy whose existence is guaranteed by the assumption.

Value iteration (VI) is one of the learning schemes for finding the optimal value function. The VI process starts with an initial guess \( V^0(\cdot) \) and iterates through

\[
V^{i+1}(x) = \min_u \left( U(x, u) + V^i(f(x, u)) \right) \quad \forall x \in \Omega \quad (7)
\]

for \( i = 0, 1, \ldots \) until the iterations converge. As one of the available convergence proofs, it is shown in [25] that, if the initial guess on \( V^0(\cdot) \) is smooth and \( 0 \leq V^0(x) \leq U(x, 0) \), \( \forall x \in \Omega \), then the VI converges monotonically to the optimal solution. Using the converged value function, denoted with \( V^\star(\cdot) \), the optimal control policy can be obtained using Eq. (5).

IV. Approximate Value Iteration

The problem with the exact VI is the issue that exact reconstruction of the right-hand side of Eq. (7) is not generally possible except for very simple problems. In general, parametric function approximators are used for this purpose, which hence give rise to function approximation errors. This is done through reconstructing \( V^{i+1}(x) \) by evaluating the right-hand side of Eq. (7) at numerous sample states distributed throughout \( \Omega \) (e.g., in a batch training form). More details are given in Sec. VI. When the approximation errors are considered, Eq. (7) reads

\[
\hat{V}^{i+1}(x) = \min_u \left( U(x, u) + \hat{V}^i(f(x, u)) \right) + \epsilon^i(x) \quad \forall x \in \Omega \quad (8)
\]

where the approximate value function at the \( i \)-th iteration is denoted with \( \hat{V}^i(\cdot) \), and the approximation error at this iteration is denoted with \( \epsilon^i(\cdot) \). Note that the value function in the right-hand side of Eq. (8) is also an approximate quantity, generated from the previous iteration.

A critical point is the fact that \( \epsilon^i(x) \) does not represent the error between the exact and the approximate value functions, denoted with \( V^{i+1}(x) \) and \( \hat{V}^{i+1}(x) \), respectively. The exact value function \( V^{i+1}(x) \) is based on using the exact \( V^i(x) \) in the right-hand side of Eq. (7), whereas \( \hat{V}^{i+1}(x) \) is being calculated based on \( \hat{V}^i(x) \), per Eq. (8). The difference between \( V^{i+1}(x) \) and \( \hat{V}^{i+1}(x) \) is an approximation error which is the cumulative effect of \( \epsilon^i(\cdot) \)'s in the previous iterations. The “per iteration” error, denoted with \( \epsilon^i(x) \), however, is simply the error of approximating/REPLACING \( \min_u (U(x, u) + V^i(f(x, u))) \) with \( \hat{V}^{i+1}(x) \).

Also, note that when \( \epsilon^i(\cdot) \neq 0 \), the convergence of the approximate VI (AVI) does not follow from the cited previous investigations, as mentioned in the introduction.

Finally, before proceeding to the convergence analysis, it is worth mentioning that one typically trains a control approximator (called the actor) to approximate the solution to the minimization problem given by Eq. (5) based on the value function resulting from the AVI. The control approximator will hence lead to another approximation error term in the process, regardless of whether the value function reconstruction is exact or approximate. However, the effect of the actor’s approximation error can be removed from the convergence analysis of AVI because the actor training can be postponed after the conclusion of the value function learning through Eq. (7) or Eq. (8) in offline learning. In other words, one can learn the optimal value function and then use the result for training the actor. The detailed algorithm is presented in [25]. However, one might be interested in online learning because it leads to the advantage of not requiring perfect knowledge of the internal dynamics of the system [9,40]. Even in case of online learning, the effect of the actor’s approximation error can be removed from the convergence analysis because the control will be directly calculated from the minimization of the right-hand side of Eq. (8) and applied on the system. In other words, even though the actor will be updated simultaneously along with the critic in online learning, the critic training is independent of the actor’s approximation accuracy. Of course, once the learning is concluded (and if it is concluded), the operation of the system will be based on the control resulting from the trained actor; hence, the actor’s approximation error can affect the stability of the system. That effect will be analyzed separately in this study, after the convergence analysis.

V. Theoretical Analyses

A. Continuity Analysis

Smooth function approximators are shown to uniformly approximate a function if the function is continuous [41,42]. Otherwise, the approximation accuracy is not guaranteed to be suitable on new states that were not used in the training. On the other hand, the minimization operation in Eq. (8) may lead to discontinuity of the right-hand side versus \( u \) because the \( u \) that minimizes the term is given by

\[
u \in \arg \min_u \left( U(x, u) + \hat{V}^i(f(x, u)) \right) \quad (9)
\]

and hence may change discontinuously versus \( x \), because \( \arg \min \cdot \) does not necessarily change continuously. Therefore, an important step is analyzing the continuity of the function subject to approximation, that is,

\[
\hat{V}^{i+1}(x) = \min_u \left( U(x, u) + \hat{V}^i(f(x, u)) \right) \quad \forall x \in \Omega \quad (10)
\]

Note that the the difference between \( \hat{V}^{i+1}(\cdot) \) and \( V^{i+1}(\cdot) \) is the fact that the latter is the approximation of the former, i.e., \( \hat{V}^{i+1} = V^{i+1} + \epsilon^i(\cdot) \), as mentioned before.

Let \( C(\Omega) \) denote the set of continuous functions at point \( x \) (respectively, \( C(f) \)) denote the set of continuous functions at point \( x \) (respectively, within \( \Omega \)). The following theorem establishes the desired continuity.

Theorem 1: If the approximate value iteration scheme, implemented using a continuous function approximator, is initiated using a continuous initial guess, then the function subject to approximation by the critic will be continuous at any finite iteration.

Proof: Based on the continuity of the function approximator, one has \( \hat{V}^i(\cdot) \in C(\Omega) \), \( \forall i \). The theorem can be proved by showing
that, if \( \hat{V}^i(x) \in \mathcal{C}(\Omega) \), then \( V^{i+1}(x) \in \mathcal{C}(\Omega) \). Let \( W(x, u) := U(x, u) + \hat{V}^i(f(x, u)) \) and \( h(x) \in \text{argmin}_{h(x) \in \mathcal{C}(\Omega)} W(x, u) \). Note that functions \( f(\cdot) \) and \( U(\cdot, \cdot) \) are smooth and, hence, continuous. Given the continuity of \( W(x, u) \) versus \( u \) at any given \( x \), the boundedness of \( \lim_{\epsilon \to 0} W(x, u) \), and the unboundedness of \( \lim_{\epsilon \to 0} W(x, u) \), which follows from the inclusion of \( U(x, u) = Q(x) + u^T R u \) in the definition of \( W(x, u) \), one has \( h(x) < \infty \). Because, \( W(x, h(x)) = V^{i+1}(x) \) the proof of continuity of \( W(., h(.)) \) suffices. The proof is done by showing that the directional limit of \( W(., h(.)) \) at any selected point is equal to its evaluation at the point, and hence, it is continuous at that point (motivated by [43]).

Let \( \bar{x} \) be an arbitrary point in \( \Omega \). Set

\[
\bar{u} := h(\bar{x})
\]

Select an open set \( \alpha \subset \mathbb{R}^n \) such that \( \bar{x} \) belongs to the boundary of \( \alpha \) and limit

\[
\bar{u} := \lim_{x \to \bar{x}, x \in \alpha} h(x)
\]

exists. If \( \bar{u} = \bar{u} \), for every such \( \alpha \), then \( h(\cdot) \in \mathcal{C}(\bar{x}) \). In this case, the continuity of \( W(., h(.)) \) at \( \bar{x} \) follows from the continuity of its forming functions [44].

Now assume \( \bar{u} \neq \bar{u} \), for some \( \alpha \) denoted with \( \alpha_0 \). From \( W(., \bar{u}) \in \mathcal{C}(\Omega) \) for the given \( \bar{u} \), one has

\[
W(\bar{x}, \bar{u}) = \lim_{x \to \bar{x}, x \in \alpha_0} W(x, \bar{u})
\]

If it can be shown that, for every selected \( \alpha_0 \), one has

\[
W(\bar{x}, \bar{u}) = W(\bar{x}, \bar{u})
\]

then the continuity of \( W(., h(.)) \) at \( \bar{x} \) follows because, from Eqs. (13) and (14), one has

\[
W(\bar{x}, \bar{u}) = \lim_{x \to \bar{x}} W(x, \bar{u})
\]

and Eq. (15) leads to the continuity by definition [44].

The proof that Eq. (14) holds is done by contradiction. Assume that, for some \( \bar{x} \) and some \( \alpha_0 \), one has

\[
W(\bar{x}, \bar{u}) > W(\bar{x}, \bar{u})
\]

Inequality (16) leads to \( h(\bar{x}) \neq \bar{u} \) given the definition of \( h(\cdot) \). But this is against Eq. (11); hence, Eq. (16) cannot hold. Now, assume

\[
W(\bar{x}, \bar{u}) < W(\bar{x}, \bar{u})
\]

hence, there exists some \( \epsilon_1 > 0 \) such that

\[
W(\bar{x}, \bar{u}) + \epsilon_1 = W(\bar{x}, \bar{u})
\]

then, because of the continuity of both sides of Eq. (18) at \( \bar{x} \) for the fixed \( \bar{u} \) and \( \bar{u} \), there exists an open set \( \gamma \) containing \( \bar{x} \) (see Fig. 1) and some \( \epsilon_2 > 0 \), such that

\[
W(x, \bar{u}) + \epsilon_2 < W(x, \bar{u}). \quad \forall x \in \gamma
\]

Given the definition of \( h(\cdot) \) which leads to \( W(x, h(x)) \leq W(x, \bar{u}) \), inequality (19) implies that, at points that are close enough to \( \bar{x} \), function \( W(x, h(x)) \) is away from \( W(x, \bar{u}) \) at least by a margin of \( \epsilon_2 \). But this contradicts Eq. (12), which implies that \( h(x) \) can be made arbitrarily close to \( \bar{u} \) as \( x \) gets close to \( \bar{x} \) within \( \alpha_0 \). The reason is that, given the continuity of \( W(x, u) \) versus both \( x \) and \( u \), the ability of making \( h(x) \) arbitrarily close to \( \bar{u} \) leads to the conclusion that function \( W(x, h(x)) \) can be made arbitrarily close to \( W(x, \bar{u}) \) if \( x \) approaches \( \bar{x} \) from a certain direction. Note that sets \( \gamma \) and \( \alpha_0 \) are not disjoint because \( \bar{x} \) is within \( \gamma \) and on the boundary of \( \alpha_0 \), as shown in Fig. 1. Hence, inequality (17) also cannot hold. Therefore, Eq. (14) holds, and hence, \( W(., h(.)) \in \mathcal{C}(\bar{x}) \). Finally, the continuity of the function subject to investigation at any arbitrary \( \bar{x} \in \Omega \) leads to the continuity of the function in \( \Omega \).

It is worth mentioning that the proof of continuity of the value function given by theorem 1 is valid for problems with infinite-horizon cost functions, which are the problems subject to investigation in this paper. However, one might be interested in extending the continuity results to finite-horizon problems. In this case, the result will directly extend to problems with fixed final times given the analogy discussed in [25], but they may not be valid for finite-horizon problems with free final times.

B. Convergence Analysis

Analysis of boundedness and convergence of sequence \( \{\hat{V}(x)\}_{x=0}^{\infty} \) resulting from the AVI given by Eq. (8) and its relation versus the optimal value function is presented in this subsection. The idea is establishing the desired results through enforcing a bound on the per iteration approximation errors given by \( e(\cdot) \). The selected bound is given by \( |e(x)| \leq cU(x, 0) \forall i \in \mathbb{N} \) for some \( c \in [0, 1] \). Considering this bound, parameter \( c \) corresponds to the accuracy of the function approximator. One has indirect control over \( c \) (e.g., selecting a richer function approximator and/or a smaller domain of interest both lead to a smaller \( c \)). However, \( c \) is not a design parameter that one selects directly.

Let \( \{\hat{V}(x)\}_{x=0}^{\infty} \) and \( \{\hat{V}(x)\}_{x=0}^{\infty} \), where \( \hat{V}: \mathbb{R}^n \to \mathbb{R}_+ \) and \( \hat{V}: \mathbb{R}^n \to \mathbb{R}_+ \), be defined as sequences of functions initiated from some \( \hat{V}(\cdot) \) and \( \hat{V}(\cdot) \) and generated by

\[
\hat{V}(x) = \min_u \left( U(x, u) + cU(x, 0) + \hat{V}(f(x, u)) \right). \quad \forall x \in \Omega
\]

Now, assuming an upper bound for the approximation error \( e(x) \), the following results can be obtained, which may resemble a boundedness idea in [24]. However, the concept, the proof, and the applications of the boundedness result, presented in this study, are different.

Lemma 1: Let \( |e(x)| \leq cU(x, 0), \forall i \in \mathbb{N} \) for some \( c \in [0, 1] \). If the recursive relations given by Eqs. (8), (20), and (21) are initialized such that \( \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega \), then one has \( \hat{V}(x) \leq \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega, \forall i \in \mathbb{N} \). Moreover, if \( \hat{V}(x) = \hat{V}(x) \), then \( \hat{V}(x) \) and \( \hat{V}(x) \) are, respectively, the greatest lower bound and the least upper bound of \( \hat{V}(x) \) as \( e(x) \)'s change within interval \([-cU(x, 0), cU(x, 0)] \), \( \forall i \).

Proof: The lemma can be proved using mathematical induction. Initially \( \hat{V}(x) \leq \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega \) by assumption. Let \( \hat{V}(x) \leq \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega \) hold for some \( i \). Comparing Eq. (20) with Eq. (8), it follows that \( \hat{V}(x) \leq \hat{V}(x) \) because \( e(x) \leq cU(x, 0) \) and \( \hat{V}(x) \leq \hat{V}(x) \). Therefore, one has \( \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega \). The proof of \( \hat{V}(x) \leq \hat{V}(x), \forall x \in \Omega \) is similar through comparing Eq. (21) with Eq. (8) and using mathematical induction. Proof of the last part of the lemma follows from assuming \( e(x) \leq cU(x, 0), \forall i \) (respectively, \( e(x) = -cU(x, 0), \forall i \)), which leads to \( \hat{V}(x) \leq \hat{V}(x) \) (respectively, \( \hat{V}(x) = \hat{V}(x) \)). Therefore, there are no other “tighter” bounds for \( \hat{V}(x) \)
Considering recursive relations (20) and (21), it is seen that \( \nabla \) and \( \nabla' \) are, respectively, the value functions at the \( i \)th iteration of exact VI for cost functions

\[
J = \sum_{k=0}^{\infty} \left( U(x_k, u_k) + cU(x_k, 0) \right)
\]

\[
J = \sum_{k=0}^{\infty} \left( U(x_k, u_k) - cU(x_k, 0) \right)
\]

subject to dynamics (1). The following lemma provides sufficient conditions for their convergence to the respective optimal value functions.

**Lemma 2:** The exact value iterations given by Eqs. (20) and (21) converge to the optimal value functions of cost functions (22) and (23), respectively. If they are initialized by smooth functions \( \nabla^{\infty} \) and \( \nabla'^{\infty} \) such that \( 0 \leq \nabla^{\infty}(x) \leq (1 - c)U(x, 0) \), \( \forall x \in \Omega \) and \( 0 \leq \nabla'(x) \leq (1 + c)U(x, 0) \), \( \forall x \in \Omega \), where \( c \in [0, 1] \).

**Proof:** The proof follows from [25] because iterations given by Eqs. (20) and (21) are exact VIs.

Considering Lemmas 1 and 2, the following theorem proves the boundedness of the elements of \( \{\nabla(x)^{\infty}\} \) resulting from the approximate VI.

**Theorem 2:** Let \( |e(x)| \leq cU(x, 0) \), \( \forall x \in \Omega \), \( \forall i \in \mathbb{N} \) for some \( c \in [0, 1) \). If the approximate value iteration given by Eq. (8) is initialized such that \( 0 \leq \nabla^{\infty}(x) \leq (1 - c)U(x, 0) \), \( \forall x \in \Omega \), then the elements of sequence \( \{\nabla^{\infty}(x)^{\infty}\} \) as \( i \to \infty \) are bounded by the optimal value functions of cost functions (22) and (23) denoted with \( \nabla^{\infty}(x) \) and \( \nabla'(x) \), respectively, in the sense that the greatest lower bound of \( \nabla^{\infty}(x) \) converges to \( \nabla^{\infty}(x) \) and the least upper bound of \( \nabla'(x) \) converges to \( \nabla'(x) \) as \( i \to \infty \).

**Proof:** The proof follows from the boundedness of \( \{\nabla^{\infty}(x)^{\infty}\} \) given in Lemma 1 and the convergence of the bounds for smooth \( \nabla^{\infty}(x) \) and \( \nabla'(x) \), which satisfy \( 0 \leq \nabla^{\infty}(x) = \nabla'(x) = \nabla'(x) \leq (1 - c)U(x, 0) \), \( \forall x \in \Omega \), based on Lemma 2.

Moreover, the following result can be achieved, with the uniformness feature that will be used in stability analysis.

**Theorem 3:** Let \( |e(x)| \leq cU(x, 0) \), \( \forall x \in \Omega \), \( \forall i \in \mathbb{N} \) for some \( c \in [0, 1) \). Also, let the approximate value iteration given by Eq. (8) be initialized such that \( 0 \leq \nabla^{\infty}(x) \leq (1 - c)U(x, 0) \), \( \forall x \in \Omega \). As \( i \to \infty \), for example by selecting a richer function approximator, the results from the approximate value iteration [Eq. (8)] converge uniformly to the results from the exact value iterations given by Eq. (2) corresponding to cost function (2) in compact set \( \Omega \). More specifically, the least upper bound and the greatest lower bound of \( \nabla^{\infty}(x) \) for \( i \to \infty \) converge uniformly to the optimal value function associated with cost function (2) as \( c \to 0 \).

**Proof:** The proof is given in the Appendix.

Theorem 2 proves that sequence \( \{\nabla^{\infty}(x)^{\infty}\} \) is upper- and lower-bounded. Then, theorem 3 proves the uniform convergence of these bounds to the desired optimal solution if \( c \to 0 \). However, when the approximation error does not vanish, the mere fact that the sequence is upper-bounded does not prove its convergence (the elements of a sequence can be upper-bounded but oscillatory). The established boundedness results resemble the "convergence to a neighborhood" or interval presented in [29]; however, besides the idea behind the analysis which is different in here, the assumptions are also different and less restrictive in this study, as discussed in Sec. I. The challenge in proof of actual convergence of the AVI to a specific limit function is the point that the monotonicity feature used for the convergence proof of exact VI (e.g., in [25]) is no longer guaranteed when approximation errors exist.

### C. Stability Analysis

Even though it is proved that the AVI result remains bounded (Theorem 2), it is not necessarily optimal, due to the presence of the approximation error. Once the solution is not optimal with respect to the selected cost function, it may not even stabilize the system. Therefore, stability analysis of the control resulting from the AVI is nontrivial. This subsection is aimed at this pursuit.

Let the AVI be terminated at the \( i \)th iteration, once a convergence tolerance, denoted with positive (semi-)definite function \( \delta(x) \), is achieved, i.e., when

\[
|\nabla^{i+1}(x) - \nabla^{i}(x)| \leq \delta(x), \quad \forall x \in \Omega
\]  

(24)

Note that if approximation errors do not exist, the convergence of VI to a finite limit function [9,24,25] guarantees the satisfaction of the convergence criterion [Eq. (24)] for a large enough, but finite \( i \) for any given arbitrary positive-definite \( \delta(x) \). However, if the errors exist, an arbitrarily selected \( \delta(x) \) can be achieved only when the approximation errors are small enough, per the uniform convergence result of theorem 3.

Once the convergence criteria is achieved, the resulting value function \( \nabla^{i}(x) \) can be used for calculating the feedback control, denoted with \( h'(x) \) through solving the minimization problem given by

\[
h'(x) \in \arg\min_u \left( U(x, u) + \nabla^i(f(x, u)) \right)
\]

(25)

in online operation (i.e., on the fly) based on the instantaneous state of the system, denoted with \( x \). This approach, however, leads to a considerable computational load during the online operation of the system. Another approach, widely used by ADP practitioners, is training another function approximator, called the actuator, for approximating the solution to the minimization problem given by Eq. (25), for different states within the domain of operation. Denoting the approximation of \( h'(x) \) with \( \hat{h}'(x) \), the approximation error of the actuator, denoted with \( \mu(x) \), will be introduced into the process:

\[
\hat{h}'(x) = h'(x) + \mu(x), \quad \forall x \in \Omega
\]

(26)

The next theorem provides a sufficient condition for asymptotic stability of \( \hat{h}'(x) \) in a subset of \( \Omega \), which is an estimation of its region of attraction [37].

**Theorem 4:** Let the value function be approximated using a smooth function approximator with an approximation error upper-bounded by \( |e(x)| \leq cU(x, 0) \), \( \forall x \in \Omega \), \( \forall i \in \mathbb{N} \), for some \( c \in [0, 1) \). Also, let the Lipschitz constant of function \( \nabla^i(f(x, \cdot)) \), whose existence follows from the smoothness of the functions, be given by \( L_V \). If the approximation error of the actuator is upper-bounded by

\[
|\mu(x)| \leq \frac{(1 - c)U(x, 0) - \delta(x)}{L_V}, \quad \forall x \in \Omega
\]

(27)

with the equality holding only at the origin, then the control policy \( \hat{h}'(x) \) resulting from the approximate value iteration, terminated with the tolerance of \( \delta(x) \), asymptotically stabilizes the system for any initial state selected in compact domain \( \mathcal{B}_r \subset \Omega \) containing the origin, where \( \mathcal{B}_r := \{ x \in \mathbb{R}^n : \| x \| \leq r \} \). \( r \) is the greatest \( r \) using which \( \mathcal{B}_r \subset \Omega \) holds, and \( \| \cdot \| \) denotes vector norm.

**Proof:** The proof is given in the Appendix.

Inequality (27) provides an upper bound for the norm of the actor’s approximation error. However, it is important to note that the upper bound has to be positive-definite, otherwise no nonzero approximation error can satisfy it. In other words, one needs the numerator of the right-hand side of inequality (27) to be positive for \( x \neq 0 \). Therefore, it is required to have

\[
\delta(x) < (1 - c)U(x, 0), \quad \forall x \in \Omega - \{0\}
\]

(28)

To be more precise, satisfaction of Eq. (24) after a finite \( i \) for an arbitrary \( \delta(x) \) needs uniform convergence of exact VI. Although the cited proofs provide its pointwise convergence, uniform convergence also can be proved, for example using some boundedness of \( \nabla^i(x)^{\infty} \), the result given in [24] leads to the desired uniform convergence, as shown in [45].
On the other hand, considering (24) and the boundedness result in Theorem 2, one has
\[ \delta(x) \leq c(V^*(x) + V^t(x)) \] (29)
if the number of iterations of AVI is large enough, where upper-bounded positive-definite functions \( V^*(x) \) and \( V^t(x) \) were defined in the proof of theorem 3. The reason is the least upper bound and the upper boundedness of the terms in the right hand side, cf. δ(29) and the upper boundedness of the terms in the right hand side, cf. the proof of theorem 3, if the critic’s approximation error is small enough, leading to a small c, inequality (28) can always be achieved. This will then lead to a positive-definite right-hand side in inequality (27), which determines the upper bound of the actor’s approximation error.

Note that \( \delta(x) \) can be explicitly obtained from the results of the concluded AVI (e.g., \( \delta(x) = |V^*(x) - V^{t+1}(x)| \)). Therefore, in practice, one can check the validity of inequality (28) before training the actor and, if not satisfied, will need to increase the approximation capability/accuracy of the critic (e.g., by increasing the number of neurons). It is an interesting feature of the upper bound of the actor approximation error given by theorem 4 that it can be calculated for any general nonlinear system because all the parameters are either known or calculable for a given system. For example, besides checking the validity of inequality (28), which was discussed, the Lipschitz constant \( L_\Gamma \) can be calculated analytically or numerically through examining the trained critic and actor. Note that, to find the Lipschitz constant \( L_\Gamma \), one needs \( \Gamma \) defined in the proof of theorem 4, unless the functions are globally Lipschitz. Set \( \Gamma \) can be obtained using the pointwise values obtained for Eq. (25) and the trained actor. The former corresponds to \( h(\cdot) \), and the latter is \( \tilde{h}(\cdot) \). Using these data, set \( \Gamma \), i.e., the union of the images of \( \Omega \) under \( h(\cdot) \) and \( \tilde{h}(\cdot) \), can be found (see the next section for an example).

Another interesting feature of the given stability result is the point that it admits termination of the learning process after a finite number of iterations, through admitting the convergence criterion given by inequality (24). This holds for both cases of having or not having the approximation errors. Finally, providing an estimate of the region of attraction is an important result. The reason is, it guarantees that for any initial condition inside the region, the entire trajectory remains within the region, hence, the tuned controller remains valid for application. More details are given in the proof of the theorem.

VI. Numerical Example: Orbital Maneuver Problem

A. Problem Setup

The orbital maneuver problem with continuous thrust simulated in [18, 46] is selected for numerical analyses in this study. A rigid spacecraft is orbiting around the Earth. It needs to perform a maneuver to move to a given circular orbit. The regulation of the states corresponds to positioning the spacecraft in the destination orbit, with the desired velocity, to stay in the orbit after the maneuver. Assuming planar motion, the nondimensionalized discretization vector of the center of mass of the spacecraft from the center of the orbital plane (i.e., a frame positioned at the destination orbit and rotating with the orbital velocity of the destination orbit) is denoted by \([X, Y]^T\), where real numbers \(X\) and \(Y\) are the components of the vector in the orbital plane. The equations of motion of the spacecraft in the gravity field given in the orbital frame are [46]
\[
\begin{align*}
\ddot{X} - 2\dot{Y} + (1 + X)(1/r^3 - 1) &= u_x \\
\ddot{Y} + 2\dot{X} + Y(1/r^3 - 1) &= u_y
\end{align*}
\]
where \(u_x\) and \(u_y\) denote the components of the nondimensionalized total force (per unit of mass) applied on the spacecraft, and \(r = \sqrt{(1 + X)^2 + Y^2}\). For nondimensionalizing, a reference length \(R\) and a reference time \(T\) are selected. The radius of the destination orbit is selected for \(R\), and the inverse of the angular velocity of the spacecraft orbiting in the destination orbit, i.e., \(\sqrt{(R^3/\mu_E)}\), is selected for \(T\), where \(\mu_E\) denotes the gravitational parameter for the Earth [46].

Selecting the state vector as \(x = [X, Y, \dot{X}, \dot{Y}]^T\) and the control vector as \(u = [u_x, u_y]^T\), the state equation of the orbital maneuver problem can be written as
\[
\dot{x} = \begin{bmatrix} x(3) & x(4) \\ 2x(4) - (1 + x(1))(1/r^3 - 1) & -2x(3) - x(2)(1/r^3 - 1) \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} u \] (30)

Note that the elements of vector \(x\) are denoted with \(x(i), i = 1, 2, 3, 4\), as opposed to the customary notation of \(x_i\), to avoid mistaking them with the discrete time steps (i.e., in \(x_k\) used through the paper).

Minimizing the cost function
\[
J = \int_0^\infty (100x^2 + u^2) dt
\]
leads to both positioning the spacecraft in the destination orbit and having it orbit with the desired orbital velocity because both the relative position and the relative velocity will be forced to converge to zero.

B. Implementation of the Solution

The dynamics of the problem given by Eq. (30) is in the continuous-time form. Using the (nondimensionalized) sampling time of \(\Delta t = 0.01\), the continuous-time problem is discretized to
\[
x_{k+1} = F(x_k) + g(u_k)
\]
with cost function terms \(Q(x) = 100x^2 x\Delta t, \quad R = \text{diag}(1, 1) \Delta t\).

Because the system is control affine and the utility function is quadratic in \(u\), the minimum of the term in the right-hand side of Eq. (8) can be simply found by setting its gradient to zero, which leads to
\[
u = -\frac{1}{2} R^{-1} g^T \nabla \tilde{V}(f(x, u)) \] (31)
where \(\nabla \tilde{V}(x) = (\partial \tilde{V}(x)/\partial x)^T\) [47]. Note that Eq. (31) is implicit because \(u\) exists on the right-hand side as well. It is proved in [25] that selecting any finite \(u^0\) and conducting the successive approximation given by
\[
u^{i+1} = -\frac{1}{2} R^{-1} g^T \nabla \tilde{V}(f(x, u^i)) \] (32)
parameter \(u^i\) converges to the solution to Eq. (31), if the sampling time \(\Delta t\) is small enough. Note that a complete set of iterations on Eq. (32), called inner loop in [25], needs to be done at each single iteration of Eq. (8), called outer loop. However, selecting a small enough sampling time, the inner-loop iterations are observed to converge very quickly [25]. This approach is used for finding the minimum of Eq. (8) during the critic training as well as for training the actor using Eq. (25), which because of inevitable approximation errors leads to Eq. (26).

The linear-in-parameter structures \(\tilde{V}(\cdot) = W_i^T \phi(\cdot)\) and \(\hat{h}(\cdot) = W_q^T \sigma(\cdot)\) are selected for function approximation, where \(\phi: \mathbb{R}^n \rightarrow \mathbb{R}^q\) and \(\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^{n_q}\) are the nonlinear smooth basis functions to be selected, \(W_i \in \mathbb{R}^{q \times n}\), \(\forall i\), and \(W_q \in \mathbb{R}^{n_q \times n}\) are the unknown parameters or weights to be found. Positive integers \(n_i\) and \(n_q\) denote the number of neurons or basis functions in the critic and the actor,
converged). The approximate value is evaluated at each iteration of A VI leads to a new set of weights for the critic (i.e., the weights of the critic evolve with the iterations). Therefore, they are denoted with superscript $i$ to relate them to their respective iterations. However, only one actor will be trained to learn the resulting control policy, denoted with $\hat{h}(\cdot)$. Thus, the actor’s weight matrix $W_a$ is not iteration dependent.

Denoting the vector whose elements are all the nonrepeating polynomials made up through multiplying the polynomial elements of vector $A$ by those of vector $B$ with $A \otimes B$, the following basis functions are selected for the function approximators:

$$
\phi(x) = [(x \otimes x)^T, (x \otimes x \otimes x)^T]^T
$$

(33)

$$
\sigma(x) = [x^T, (x \otimes x)^T]^T
$$

(34)

Five hundred random state vectors, denoted with $x[p], p \in \{1, 2, \ldots, 500\}$, were selected from $\Omega_1 = \{x \in \mathbb{R}^4 : -0.3 \leq x(i) \leq 0.3, i = 1, 2, 3, 4\}$, for learning the value function using Eq. (8). Selecting a constant convergence tolerance of 0.01, the convergence was evaluated in a fashion similar to Eq. (24). Starting with $V^0(x) = 0$ as the initial guess, the AVI converged after 330 iterations of Eq. (8), each involving an inner loop over Eq. (32), which was observed to converge in less than four iterations. Each iteration of the process involves finding $W_{c}^{i+1}$ given $W_{c}$ using

$$
W_{c}^{i+1T} \phi(x[p]) \approx U(x[p], u[p]) + W_{c}^{iT} \phi(f(x[p], u[p])), \quad \forall p \in \{1, 2, \ldots, 500\}
$$

(35)

where each $u^{[p]}$ is the converged value of Eq. (32) in which, the $x$ is substituted with the respective sample state $x^{[p]}$ and $W_{c}^{iT} \phi(\cdot)$ is used for $V^i(\cdot)$. The method of least squares, as detailed in [43], was used for finding $W_{c}^{i+1}$. Figure 2 shows the evolution of the elements of the critic’s weight, versus the iteration index. In terms of the elapsed time, the critic training took around 80 s on a desktop computer with an Intel Core i7-3770 3.40 GHz processor and 8 GB of memory, running Windows 7 and MATLAB 2013 (single threading).

Once the critic training is concluded, the actor training is done in one shot over the selected random states, that is, $W_a$ is found using

$$
W_{a}^{i} \sigma(x[p]) \approx u^{[p]}, \quad \forall p \in \{1, 2, \ldots, 500\}
$$

(36)

evaluated at $i = 330$ (i.e., the iteration at which the critic training converged).

C. Analysis of the Results

For evaluation of the function approximation accuracy (i.e., to quantify the approximation error $e^i(x)$, $\forall i$), another set of sample states were selected. The point is that the approximation error at the sample states used in the training, $x^{[p]}$‘s, may be very low, but it is important to evaluate the error at other states, to evaluate the generalization accuracy of the function approximators. To this goal, 20,000 equidistant states were selected by gridding $\Omega_1$, denoted with $y^{[p]}, p \in \{1, 2, \ldots, 20,000\}$. Function $e^i(x)$ is then given by

$$
e^i(y[p]) := W_{c}^{i+1T} \phi(y[p]) - (U(y[p], u^{[p]}) + W_{c}^{iT} \phi(f(y[p], u^{[p]}))), \quad \forall p \in \{1, 2, \ldots, 20,000\}, \quad \forall i
$$

(37)

where $u^{[p]}$ is the converged value of Eq. (32) evaluated at the respective sample state $y^{[p]}$ using critic $W_{c}^{iT} \phi(\cdot)$.

Having the pointwise values of function $e^i(y[p]), \forall p, \forall i$, constant $c$ used in $\|e^i(x)\| \leq cU(x(\cdot)), \forall i, \forall x$, can be found using

$$
c \approx \max_{p \in \{1, 2, \ldots, 20,000\}} \frac{|e^i(y[p])|}{\max_{i \in \{1, 2, \ldots, 330\}} |e^i(y[p])|}
$$

which led to $c = 0.15$. If $c$ was obtained using $y^{[p]}$‘s, that is, at the states used in the training, the result would be $c \approx 0.10$, which is close to what was achieved using new set of states. This demonstrates the good generalization capability of the function approximator. Note that a value less than 1 is desired per the theory presented in this work (e.g., for proof of boundedness as in theorem 2). However, the iteration has already converged; therefore, the concern of divergence does not exist in here. But the existing concern is the quality of the result compared with the optimal solution and the reliability of the controller.

If the assumptions of theorem 4 hold, asymptotic stability of the controller can be concluded. The main issue is verification of inequality (27), for which function $\mu(\cdot)$ is needed. The process of evaluation of the approximation accuracy of the critic at the new state vectors $y[p]$ is done for quantifying $\mu(\cdot)$ as well. As for the upper bound of this error given by inequality (27), functions $U(y[p], 0)$ and $\delta(y[p]) = |W_{c}^{iT} \phi(y[p]) - W_{c}^{i+1T} \phi(y[p])|$ evaluated at $i = 330$ (i.e., the iteration at which the learning converged) are used. But the Lipschitz constant $L_V$ is also required; cf. inequality (A11) given in the Appendix. Note that the respective function is smooth and, hence, differentiable. Thus, finding the maximum of its gradient with respect to $u$ leads to its Lipschitz constant [48]:

$$
\frac{\partial V^i(f(x, u), u)}{\partial u} \bigg|_{x=f(x, u)} = \frac{\partial V^i(x)}{\partial x} \bigg|_{x=f(x, u)} \frac{\partial f(x, u)}{\partial u} = W_{c}^{iT} \nabla f(x, u) g
$$

therefore

$$
L_V \approx \max_{p \in \{1,2,20,000\}} W_{c}^{330T} \nabla \phi(f(y[p], w^{330}[p])) g = 0.186
$$

To be more accurate, $L_V$ is the maximum number between the result of the foregoing equation and

$$
L_V \approx \max_{p \in \{1,2,20,000\}} W_{c}^{330T} \nabla \phi(f(y[p], W_{c}^{i} \sigma(y[p])) g
$$

where the difference is that one is evaluated at $w^{330}[p]$‘s and the other one at $W_{c}^{i} \sigma(y[p])$‘s. Note that the former is $\hat{h}(y[p])$ and the latter is $h'(y[p]), \forall i = 330$. But, considering the maximum norm of the actor approximation error given by $\mu_{\max} := \max_{p \in \{1,2,20,000\}} \|\mu(y[p])\| = 0.02$ compared with the maximum

---

Fig. 2 Evolution of the weights of the critic during the AVI.
norm of the control, which was observed to be around 10, the difference between the two evaluations of $L_V$ turned out to be negligible.

Evaluating $\mu(x)$ and its upper bound given by inequality (27), it turned out that $[\mu(x)]$ never exceeds the bound. As a matter of fact, it remains smaller than 11% of the upper bound. Therefore, the asymptotic stability of the controller about the origin follows.

Selecting the initial condition of $x_0 := \{0.05, 0.05, 0.3, 0.3\}^T$, the system is operated using the trained neurocontroller, and the resulting state trajectories are presented in Fig. 3. For comparison purposes, the (open-loop) optimal solution to the problem is calculated numerically, using the direct method of optimization, and superimposed with the results. It can be seen from these results that the controller has been very accurate in approximating the optimal solution. Besides comparing the resulting state trajectories, the cost-to-go also can be compared. The cost-to-go for the numerical open-loop solution (the optimal cost-to-go) turned out to be 2.4802, which is slightly less than the cost-to-go resulting from the closed-loop controller, 2.4806. Note that the latter is the actual resulting cost-to-go using the trained neurocontroller, not the one approximated by $\widehat{\mu}(x)$. This approximation, however, is supposed to be upper- and lower-bounded by the optimal cost-to-go corresponding to cost functions (22) and (23), per theorem 2. The upper and lower-cost-to-go were numerically found to be 2.7047 and 2.2456, respectively, which confirm the analytical result given by the theorem and provide an idea of the near-optimality of the AVI results.

Considering the previous simulated initial conditions, it is seen that the state trajectory did not exit $\Omega_1$, because no state element ever exited the interval of $[0.03, 0.03]$. Therefore, the control calculated by the neurocontroller was valid. However, this was not guaranteed or obvious from the given initial condition. But, per theorem 4, one can find an estimation of the ROA for the trained neurocontroller to guarantee such a desired behavior.

As for finding the estimation of the ROA, numerically analyzing $B_{r'}$, defined in Theorem 4, it was observed that $\bar{r} = 1.05$ for the selected $\Omega_1$, where $\bar{r}$ is the greatest $r$ using which $B_{r} \subseteq \Omega_1$. But, evaluating the converged critical at the selected initial condition, one has $V_{330}(x_0) = 2.4598$, which means $x_0 \notin B_2$. Therefore, it was not guaranteed that $x_0 \in \Omega_1, \forall k$. If interested to use the trained neurocontroller with guaranteed stability, one needs to select smaller initial conditions, such that they belong to $B_{r'}$. Note that $V_{330}(\cdot)$ is continuous and vanishes at the origin. Therefore, $B_{r'}$ is a compact set with the origin as an interior point [44]. Details of this conclusion are given in the proof of theorem 4 in the Appendix. However, if controlling larger initial conditions, like the selected $x_0$, is of interest, one needs to retrain the neurocontroller using a larger domain of interest. To this purpose, $\Omega_2 := \{x \in R^4 : -0.5 \leq x(i) \leq 0.5, i = 1, 2, 3, 4\}$ was selected, and the neurocontroller was retrained. Note that as the training domain is expanded, it is advisable to pick more random sample states as well. For this training, 2000 random states were selected from $\Omega_2$, instead of 500 used earlier. Once the training is concluded, evaluating the critic upper bound constant $c$ using the discussed method, it was observed to be around 0.76, which is close to the critical value of 1. Such a large critic approximation error led to the norm of the system under the resulting controller, given the fact that these numbers are calculated through sampling the space. In other words, when the error is so large, it is possible that at some other samples in the state space, the bound, given by inequality (27) is even violated. The problem can be resolved by improving the approximation capability of the function approximators. An option is using multilayer neural networks. Another option is using richer basis functions. For example, instead of the basis function [Eqs. (33) and (34)], one may select the richer sets of basis functions given by

$$\phi(x) = [(x \otimes x)^T, (x \otimes x \otimes x)^T, (x \otimes x \otimes x \otimes x)^T]$$

(38)

$$\sigma(x) = [x^T, (x \otimes x)^T, (x \otimes x \otimes x)^T] \otimes [x \otimes x \otimes x \otimes x]$$. 

(39)

Selecting this new set of basis functions the training was redone over $\Omega_2$, and the critic upper bound constant $c$ turned out to be 0.29, with $[\mu(x)]$ never exceeding 12% of its upper bound. This new neurocontroller led to $\bar{r} = 3.12$; therefore, $x_0 \in B_2$, and one can be assured that the trajectory will not exit the domain on which the neurocontroller is trained. Using this new neurocontroller for the given initial conditions, it was observed that the results are extremely similar to what is presented in Fig. 3. This similarity may mean that the developed sufficient conditions for guaranteed stability and ROA are still conservative, and milder conditions for the approximation bounds can probably be obtained.

VII. Conclusions

Analytical investigations of the effects of the approximation errors on the quality of the result of approximate dynamic programming were conducted. It was observed through verifiable assumptions and conditions that the learning results remain bounded. Once the learning is terminated after a finite number of iterations, it was shown that the stability of the result can be verified, and an estimation of the domain of attraction can be obtained. The comprehensive numerical analysis of the theoretical results through a nontrivial fourth-order aerospace problem demonstrated the process of using the theory in practice. Given the theoretical analyses without ignoring the approximation errors and considering the termination of learning after a finite number of iterations, these results contribute to the mathematical rigor of ADP.

Acknowledgments

This research was partially supported by a grant from the National Science Foundation. The author would like to thank anonymous reviewers whose comments improved the paper.

Appendix: Proofs of Theorems 3 and 4

The proofs of Theorems 3 and 4 are given in this Appendix.

Proof of theorem 3: Let the optimal value function associated with cost function (2) be given by $V' (x)$. Let $V^\dagger (x)$ be defined as...
\[
\hat{V}^*(x_0) := \sum_{k=0}^{\infty} U(x_k^*, 0), \quad \forall \ x_0 \in \mathbb{R}^n \tag{A1}
\]

where \(x_k^* := f(x_k, h^*(x_k^*))\), \(\forall k \in \mathbb{N} \setminus \{0\}\) and \(x_0^* := x_0\). In other words, the summation in Eq. (A1) is evaluated along the optimal trajectory with respect to Eq. (2). One has

\[
V^*(x) \leq \hat{V}^*(x), \quad \forall \ x \in \mathbb{R}^n \tag{A2}
\]

where \(\hat{V}^*(x)\) is the optimal value function associate with cost function (22); otherwise, the control resulting from \(\hat{V}^*(x)\) will be the optimal control for cost function (2). Moreover,

\[
\hat{V}^*(x) \leq V^*(x) + c\hat{V}^*(x), \quad \forall \ x \in \mathbb{R}^n \tag{A3}
\]

otherwise \(\hat{V}^*(x)\) will not be the optimal value function for cost function (22). Note that both sides of inequality (A3) include infinite sums of \(U(x_k, u_k) + cU(x_k, 0)\) terms, but they are evaluated along different trajectories (i.e., the applied controls are different). The summation in the left-hand side is based on the control that minimizes cost function (22), and the summation in the right-hand side is based on the control that minimizes cost function (2).

By inequalities (A2) and (A3), one has

\[
|V^*(x) - \hat{V}^*(x)| \leq c\hat{V}^*(x), \quad \forall \ x \in \mathbb{R}^n \tag{A4}
\]

Let \(\hat{V}^*_{\text{max}} := \sup_{x_0 \in \Omega} \hat{V}^*(x)\). Note that \(\hat{V}^*_{\text{max}}\) is a finite constant, by the upper-boundedness of \(V^*(x)\) in \(\Omega\) which follows from assumption 1. Therefore, the foregoing inequality leads to

\[
|V^*(x) - \hat{V}^*(x)| \leq c\hat{V}^*_{\text{max}}, \quad \forall \ x \in \mathbb{R}^n \tag{A5}
\]

Inequality (44) proves the convergence of \(\hat{V}^*(x)\) to the optimal value function associated with cost function (2) as \(c \to 0\). Moreover, because the right-hand side of inequality (A5) is independent of \(x\), this convergence is uniform [44]. Let \(\hat{V}^*(x)\) be defined as

\[
\hat{V}^*(x_0) := \sum_{k=0}^{\infty} U(x_k^*, h^*(x_k^*)), \quad \forall \ x_0 \in \mathbb{R}^n \tag{A6}
\]

where \(h^*(\cdot)\) is the optimal control policy for cost function \(J\), i.e., the summation in the right-hand side of Eq. (A6) is evaluated along the trajectory that is optimal with respect to \(J\) given by Eq. (23). Through a similar argument, it can be seen that \(\hat{V}^*(x) \leq V^*(x)\) and \(V^*(x) \leq \hat{V}^*(x) + c\hat{V}^*(x)\), which leads to

\[
|V^*(x) - \hat{V}^*(x)| \leq c\hat{V}^*(x), \quad \forall \ x \in \mathbb{R}^n \tag{A7}
\]

Defining \(\hat{V}^*_{\text{max}} := \sup_{x_0 \in \Omega} \hat{V}^*(x)\), a similar uniform convergence can be concluded because the right-hand side of inequality (A7) will be upper-bounded by the \(x\)-independent term \(c\hat{V}^*_{\text{max}}\). It should be noted that \(\hat{V}^*_{\text{max}}\) will be a finite constant as long as \(c \in [0, 1]\). The reason is the upper-boundedness of \(V^*(x)\), \(\forall x \in \Omega\), which leads to an upper-bounded \(V^*(x)\), because \(\hat{V}^*(x) \leq V^*(x), \forall x \in \Omega\). One has \(U(x, u) = Q(x) + u^TRu\); hence,

\[
V^*(x_0) = \sum_{k=0}^{\infty} (1 - c)Q(x_k^*) + h^T(x_k^*)R \hat{h}^*(x_k^*) \tag{A8}
\]

being bounded leads to a bounded \(\sum_{k=0}^{\infty} (1 - c)Q(x_k^*)\), and the boundedness of the latter leads to a bounded \(\hat{V}^*(x_0) = \sum_{k=0}^{\infty} Q(x_k^*)\) when \(0 \leq c < 1\). Finally, these uniform convergence results along with theorem 2 prove this theorem.

Proof of theorem 4: The idea for the proof is using continuous function \(\hat{V}^*(\cdot)\) as a Lyapunov function for the system [37]. From the boundedness of \(\hat{V}^*(x)\) per lemma 1 and the positive-definiteness of the bounds (they are value functions of the respective finite-horizon cost functions as shown in (23)) for \(j \geq 1\), it follows that \(\hat{V}^*(x)\) is a positive-definite function.

Considering inequality (24), one has

\[
\hat{V}^*(x) + \delta(x) \geq \hat{V}^{i+1}(x), \quad \forall \ x \in \Omega \tag{A9}
\]

Using inequality (A9) in Eq. (9), considering Eq. (25), leads to

\[
\hat{V}^*(x) \geq U(x, h'(x)) + \hat{V} \left(f(x, h'(x))\right) + e'(x) - \delta(x), \quad \forall \ x \in \Omega \tag{A10}
\]

Note that inequality (A10) is based on \(h'(\cdot)\) (i.e., it is independent of the actor’s approximation error); see remarks at the end of Sec. IV. Thus, the next step is replacing \(h'(\cdot)\) with \(\hat{h}'(\cdot)\) because the system will be operated using \(\hat{h}'(\cdot)\). From the Lipschitz continuity of \(f(\cdot, \cdot)\) and \(\hat{V}^*(\cdot)\) within compact sets \(\Omega\) and \(\Gamma\), which follows from their smoothness in the respective compact domains [48], one has

\[
\|\hat{V}^i(f(x, u)) - \hat{V}(f(x, v))\| \leq L_V\|u - v\|, \quad \forall \ x \in \Omega, \quad \forall \ u, v \in \Gamma \tag{A11}
\]

where \(\Gamma\) is a compact subset of \(\mathbb{R}^n\) such that \(\hat{h}'(x) \in \Gamma\) and \(h'(x) \in \Gamma\), \(\forall x \in \Omega\). In other words, \(\Gamma\) is the union of the images of \(\Omega\) under \(h'(\cdot)\) and \(\hat{h}'(\cdot)\). From inequality (A11), one has

\[
\hat{V}^i(f(x, h'(x))) \geq \hat{V}(f(x, h'(x)) + \mu(x))) - L_V\|\mu(x)\|, \quad \forall \ x \in \Omega \tag{A12}
\]

After replacing \(\hat{V}^i(f(x, h'(x)))\) in the right-hand side of inequality (A10) using inequality (A12), one has

\[
\hat{V}^i(x) \geq U(x, h(x)) + \hat{V} \left(f(x, \hat{h}'(x))\right) - L_V\|\mu(x)\| + e'(x) - \delta(x), \quad \forall \ x \in \Omega \tag{A13}
\]

because \(\hat{h}'(x) = h'(x) + \mu(x)\).

The asymptotic stability follows if

\[
\Delta \hat{V}^i(x) := \hat{V}^i(f(x, \hat{h}'(x))) - \hat{V}^i(x) \leq 0, \quad \forall \ x \in \Omega \tag{A14}
\]

with the equality holding only at \(x = 0\). Considering inequality (A13), condition (A14) holds if

\[
\|\mu(x)\| \leq \frac{U(x, h(x)) + e'(x) - \delta(x)}{L_V}, \quad \forall \ x \in \Omega \tag{A15}
\]

with the possible equality only at the origin. Using \(U(x, 0) \leq U(x, u), \forall x \in \Gamma\) and \(|e'(x)| \leq cU(x, 0)\), which leads to \(-cU(x, 0) \leq e'(x)\), one has

\[
(1 - c)U(x, 0) \leq U(x, h'(x)) + e'(x), \quad \forall \ x \in \Omega \tag{A16}
\]

Considering inequality (A16), if inequality (27) holds, then inequality (A15) will hold, which leads to \(\Delta \hat{V}^i(x) \leq 0\). In the foregoing inequality, the two sides are equal only at the origin, due to the positive-definiteness of \(U(\cdot, \cdot)\). Hence, the value function \(\hat{V}^i(x)\) serves as a Lyapunov function, and the asymptotic stability of the system under the approximate control policy \(\hat{h}'(\cdot)\), within \(\Omega\) follows, as long as the entire state trajectory remains inside \(\Omega\) because, if it
leaves $\Omega$, the control policy $\hat{B}(\cdot)$ will no longer be valid, i.e., relation (A1), which is the backbone of the stability result, will no longer hold. This concern can be resolved by considering the fact that $\hat{B}_1$ will be an estimation of ROA for the system [37], per the definition of boundedness established in lemma 1.

Finally, because $\hat{B}_1$ is contained in $\Omega$, it is bounded. Also, the set is closed because it is the inverse image of a closed set, namely $[0, \hat{r}]$ under a continuous function (due to the continuity of the function approximator) [44]. Hence, $\hat{B}_1$ is compact. It also contains the origin because $\hat{V}(0) = 0$, which is the consequence of its lower- and upper-boundness established in lemma 1.

References


