2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Constrained and Stabilizing Stacked Adaptive Dynamic Programming and a Comparison with Model Predictive Control

Lukas Beckenbach, Pavel Osinenko, Thomas Göhrt and Stefan Streif

Abstract— **Model predictive control (MPC) is in many applications the de facto approach to optimal control. It typically provides an optimal input (sequence) for a finite-horizon of given running costs. Another approach, called dynamic programming (DP), is based on the Hamilton-Jacobi-Bellman formalism and usually seeks optimal inputs over an infinite horizon of running costs. Unlike MPC, DP is much less computationally tractable and typically requires state space discretization which leads to the so-called curse of dimensionality. Adaptive dynamic programming (ADP), an approach based on reinforcement learning, seeks to address the difficulties of DP by introducing approximation models for the optimal cost function and control policies. In a variant of ADP called stacked ADP (sADP), control policies are optimized over a finite stack of value function approximants, thus making it somewhat similar to MPC. First, similarities and differences between a variant of ADP and MPC are discussed. Second, MPC stability results are transferred to ADP and state and input constraints are considered. The work is concluded by a case study.**

I. Introduction

Dynamic programming (DP) dates back to the mid 20th century. In [4], Bellman described optimality principles and the structure of DP with paticular application to stochastic descision processes. The core of DP is the Hamilton-Jacobi-Bellman (HJB) equation which is a partial differential equation describing the behavior of the optimal cost function, sometimes called *cost-to-go* due to the infinite horizon over which an optimal control policy is sought. In general, when applying DP, one has to face the problem of discretizing the state space and computing the cost function and optimal policies at the discrete points. Only in some very special cases, like the linear dynamics case, analytic solutions can be found. In the linear case with a quadratic running cost, the corresponding optimal controller is the linear quadratic regulator (LQR) which can be found by solving the corresponding algebraic Riccati equation [16].

To avoid the difficulty of computing a control input for the infinite horizon optimization problem, as in DP, finite horizon solutions may be sought instead. This is the common setup of MPC, although infinite-horizon variants of it were suggested. In particular, e. g., [29] suggested the so-called quasi-infinite-horizon MPC interpreting a certain terminal set constraint, where a local controller is applied, as an approximation to the infinite "tail".

Stability properties of MPC have been thoroughly studied. The common approaches are based on terminal constraints [29], terminal costs [28], dual-mode MPC [24], contractive MPC [10] etc.

The authors are with the Automatic Control and System Dynamics Laboratory, Technische Universität Chemnitz, 09107 Chemnitz, Germany, {lukas.beckenbach,pavel.osinenko,thomas.goehrt, stefan.streif}@etit.tu-chemnitz.de

TABLE I
Brief overview of nonlinear discrete-time MPC and ADP

	MPC	ADP
Cost function	Finite sum of running costs	Approximation to the infinite sum of running costs
Stability analysis	Terminal constraint and/or cost, contraction, dual-mode	"DP-like" analysis [2], [21], contraction [15], best approximation [30]
Optimality	w.r.t. initial point & finite horizon	Global & infinite horizon (idealized case), w.r.t. initial point (practically)
Constraints	Incorporated	Usually unconstrained
Major design parameters	Terminal constraint and/or cost	Actor/critic model structure

Somewhat independently from MPC, approximate approaches to DP started to emerge in the 80s (see, e. g., [8]), leading to the so-called *reinforcement learning* and ADP. In particular, Sutton et al. [32] addressed reinforcement learning by the so-called *temporal differences* which describe the discrepancy between the iteration steps of the value function approximation. In [34], Werbos suggested to use neural networks as the approximation models which consequently led to the term "neural dynamic programming" (NDP), as also studied in, e. g., [6]. The approximant for the value function was also termed "critic", and the one for the optimal control policy – "actor".

The classical stability analyses of DP [7], were adopted for ADP (refer, e. g., to [2], [21]). Unfortunately, the latter analyses require certain strong assumptions to be satisfied, e. g., the optimal value function and control policy need to be solved exactly by subsequent iterations. Such a setup can be denoted as "idealized" ADP. Another approach sought to show convergence of the actor and critic parameters to their optima [30] thus yielding best approximation.

However, it cannot be in general proven that the optimal parameters guarantee that the critic recovers the value function exactly, i. e., there is always some discrepancy, even though it is minimized. Consequently, the computed control policies can be at best locally optimal in general. Furthermore, a choice of the critic and actor approximation model structure is in general a subtle design matter. Particular difficulties of addressing ADP convergence were discussed, e. g., in [14].

A summary of selected comparison aspects for the case of a general nonlinear discrete-time system with known system model is shown in Table I. In case of unknown system parameters, MPC and ADP may be coupled with parameter identification (refer, e. g., to adaptive MPC [1], [12] and, respectively, [35]). It should be noted that the statements given in this table may deviate for some variants of MPC and ADP and represents rather common cases.

A comparative elaboration on the relationship between MPC and ADP can be found in, e. g., [5] and [13]. In [13], the author discussed multiparametric linear model predictive control and neural network based actor-critc approximation and points out several synergies between MPC and reinforcement learning. In [5], the ADP policy iteration algorithm including desirable properties (stability, etc.) was studied and a connection to specific receding horizon control optimization problems was discussed. In [25], a general MPC strategy for Markov decision processes was extended with learning-based techniques including a value function. For another comparison, please refer also to [11].

In this work, the stacked ADP (sADP) of [26] is revisited for the sake of a comparative study with MPC. In this approach, the critic is modified so as to include not a single value function approximant, but rather a finite stack of them. Such a setup has immediate similarities to MPC. In addition to the comparison of sADP and MPC, the contribution of this paper is to provide a stability analysis of the sADP control algorithm through inclusion of a terminal constraint, under the consideration of state and input constraints.

The next section describes the basic setup of ADP using the so-called Q-learning which is a particular variant of ADP with certain benefits such as removing the necessity of using the state-to-input gradients. Section III describes sADP followed by a stability analysis. The work is concluded by a preliminary case study, which shows performance improvement of sADP compared to MPC.

II. Basic setup of ADP

Consider a discrete-time nonlinear system

$$x(k+1) = f(x(k), u(k)), \quad k \in \mathbb{Z}_{\geq 0} \tag{1}$$

where k is the time step index, $x \in \mathbb{X} \subseteq \mathbb{R}^n$ and $u \in \mathbb{U} \subseteq \mathbb{R}^m$ denote the state and control variable, respectively. The state and input constraints, $\mathbb{X}$ and $\mathbb{U}$, are assumed to contain the origin. The infinite-horizon optimal control problem is stated as follows:

$$\begin{aligned} &\min_{u} \;\; J(x(k)) = \sum_{i=k}^{\infty} r(x(i), u(i)), \\ &\text{s. t.} \;\; x(i+1) = f(x(i), u(i)), \;\; u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ... \end{aligned} \tag{2}$$

where r is the running cost which is assumed positive-definite (p. d.). The optimal value function, found by optimization of the whole policy, reads:

$$J^*(x(k)) = \min_{u(\bullet)} \sum_{i=k}^{\infty} r(x(i), u(i)),$$
$$u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ...$$

Further, the equation

$$\begin{aligned} &J^*(x(k)) = \min_{u(\bullet)} (r(x(k), u(k)) + J^*(x(k+1))), \\ &u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ... \end{aligned} \tag{3}$$

is known as the Bellman's principle of optimality [4]. The corresponding optimal policy is, respectively:

$$\begin{aligned} &u^*(x(k)) = \arg\min_{u(\bullet)} (r(x(k), u(k)) + J^*(x(k+1))), \\ &u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ... \end{aligned} \tag{4}$$

Equations (3) and (4) are in general notoriously hard to solve. ADP suggests to seek for approximate solutions instead. To this end, approximation models are used in place of J^* and u^*. Concrete variants of ADP implementation include value iteration (e.g. in [27]), policy iteration (e.g. in [21]), dual algorithms (e.g. in [20]) etc. A particular variant, called Q-learning, has the advantage that it does not require state-to-input gradients. In Q-learning, introduced by Watkins [33] and subsequently employed in LQR [19], one uses the so-called quality function, or Q-function defined as follows:

$$Q(x(k), u(k)) = r(x(k), u(k)) + J(x(k+1)). \tag{5}$$

Consequently, the optimal Q-function has the property that

$$Q^*(x(k), u(k)) = r(x(k), u(k)) + J^*(x(k+1)).$$

The Bellman's optimality principle now amounts to

$$\begin{aligned} &J^*(x(k)) = \min_{u(\bullet)} Q^*(x(k), u(k)), \\ &u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ... \end{aligned} \tag{6}$$

The corresponding optimal control policy is

$$\begin{aligned} &u^*(x(k)) = \arg\min_{u(\bullet)} Q^*(x(k), u(k)), \\ &u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, \;\; i = k, ... \end{aligned} \tag{7}$$

In Q-learning, the critic is associated with the Q-function and is usually implemented as some parametric model. In the current work, a model of the following form is used:

$$\hat{Q}(x, u) = w^\top \varphi(x, u), \tag{8}$$

where w is the parameter vector and φ is the regressor. A general recommendation for the structure of φ is not possible since nothing is a priori known about the optimal Q-function (except some specific cases, such as linear, where the problem amounts to LQR [19]). However, as will be shown further, a quadratic approximator is a possible option. To find the optimal parameters, the squared Bellman error

$$\begin{aligned} e(k)^2 = &\big(w(k)^\top \varphi(x(k), u(k)) - \\ &w(k-1)^\top \varphi(x(k+1), u(k)) - r(x(k), u(k))\big)^2 \end{aligned} \tag{9}$$

is minimized with respect to $w(k)$ at each time step k. The Bellman error, or temporal difference, describes how the approximate Q-function deviates from satisfying the Bellman equation [20], [31]. The actor in "standard" Q-learning, associated with the control input, is implemented via solving

the following optimization problem:

$$\begin{aligned} \min_{u(k)\in\mathbb{U}} \quad & \hat{Q}(x(k+1),u(k)), \\ \text{s. t.} \quad & x(k+1)\in\mathbb{X}. \end{aligned} \tag{10}$$

sADP suggests a stacked version of this optimization problem and its details are given in the next section.

III. Constrained and stabilizing Stacked ADP

This section discusses the stacked ADP approach and points out the difference to nonlinear MPC. A stabilizing scheme through additional constraints for the critic and actor is proposed.

A. Relations between stacked ADP and MPC

In sADP (see [26]), (10) is modified into the following optimization problem:

$$\begin{aligned} \min_{u(\bullet|k)} \quad & \sum_{i=1}^{N} \hat{Q}(x(i|k),u(i-1|k)), \\ \text{s. t.} \quad & x(i|k)=f(x(i-1|k),u(i-1|k)),\ \forall i=1,\dots,N \\ & u(i-1|k)\in\mathbb{U}, \\ & x(i|k)\in\mathbb{X}, \end{aligned} \tag{11}$$

where the following notation is used: $u(\bullet|k)$ and $x(\bullet|k)$ are control, respectively, state sequences starting at the time step k. For example, $x(1|k)=x(k+1)$ and $x(N|k)=x(k+N)$. In contrast to [26], state and input constraint are directly included into the optimization problem. One can immediately see the analogy to MPC, whereas the main difference is that the "running costs" are not fixed, but adjusted by the critic. In addition, each $\hat{Q}$ approximates the infinite horizon cost whereas in MPC the running cost represents the current stage cost. Like in MPC, sADP suggests to employ the first control action of the sequence $u(\bullet|k)$ at each time step k.

In the following, stability analysis of sADP is addressed. The main goal is to prove the concept of transferring some basic stability analysis ideas from MPC to ADP. For the sake of simplicity, it is suggested to use a terminal constraint and modify the actor (11) as follows:

$$\begin{aligned} \min_{u(\bullet|k)} \quad & \sum_{i=1}^{N} \hat{Q}(x(i|k),u(i-1|k)), \\ \text{s. t.} \quad & x(i|k)=f(x(i-1|k),u(i-1|k)),\ \forall i=1,\dots,N \\ & u(i-1|k)\in\mathbb{U},\ x(i|k)\in\mathbb{X},\ x(N|k)=0. \end{aligned} \tag{12}$$

Such a terminal constraint is considered classical in MPC (e.g. in [29]). Further research is required to inherit different stability analysis ideas in sADP (see also Remark 4). Notice that, similar to MPC, the control sequence $u(\bullet|k)$ is calculated under state and input constraints. The following assumption is necessary for the upcoming analysis:

Assumption 1: There exists a feasible solution $u^*(\bullet|0)$ to the optimization problem (12) at $k=0$.

This assumption is usual in MPC setups and is used to achieve recursive feasibility [22] (for the detailed analysis, please refer to Section III-C).

B. Necessary Modification of the Critic Optimization

Now, the critic is discussed. First, the resulting Q-function approximant needs to be p. d. (as does the original running cost r) which is motivated by the fact that the optimal Q-function is p. d. by definition [33]. The second constraint implies that, for fixed sequences of states and inputs, the Q-function approximant stack is non-increasing in $w(\bullet|k)$. Therefore, (9) in the current sADP is modified into:

$$\begin{aligned} \min_{w(\bullet|k)} \quad & \sum_{i=1}^{N} e(i|k)^2 = \sum_{i=1}^{N} \big(w(i|k)^\top \varphi(x(i|k),u(i-1|k)) \\ & - w(i|k-1)^\top \varphi(x(i|k+1),u(i-1|k)) \\ & - r(x(i-1|k),u(i-1|k))\big)^2, \end{aligned} \tag{13}$$

$$\text{s. t.} \quad w^\top\varphi(x,u) \text{ is p. d. in } x,u, \tag{14}$$

$$\begin{aligned} & \sum_{i=1}^{N} w(i|k)^\top \varphi(x(i|k),u(i-1|k)) \le \\ & \sum_{i=1}^{N} w(i|k-1)^\top \varphi(x(i|k),u(i-1|k)). \end{aligned} \tag{15}$$

Here, a sequence of critic parameters $w(\bullet|k)$ is calculated by minimization over a stack of squared temporal differences. The critic parameters characterize the approximation of the optimal value function.

Remark 1: The condition on positive definiteness of the Q-function approximant in general leads to a p. d. optimization problem. However, the Q-function approximation structure is a design matter and a suitable choice may allow formulating simple linear constraints on the critic parameters. In particular, in case of a quadratic approximation, all the parameters may be forced positive. Otherwise, one might employ the technique called cylindrical algebraic decomposition to transform a constraint that contains quantifiers over x and u into a quantifier-free one which depends on w only (for a description, refer, e. g., to [3]).

Remark 2: The second condition (15) implies that, for fixed sequences of states and inputs, the Q-function approximant stack is non-increasing in $w(\bullet|k)$. As mentioned above, this assumption is necessary to perform stability analysis. The case study in Section IV shows that it is not too restrictive to hinder functioning of sADP, but future analysis is required as to whether this assumption could be relaxed. Notice that this condition is stated only for the parameters w and does not describe a decay condition of the Q-function approximant in the sense of Lyapunov. The actual decay property will be derived later in Section III-C.

To conclude, the following assumption is made:

Assumption 2: For each $k\in\mathbb{Z}_{\ge 0}$, there exists a feasible solution $w^*(\bullet|k)$ to the optimization problem (13).

Remark 3: This assumption is trivially satisfied if, e. g., the approximation model in the critic is linear in parameters w, and if the first constraint (14) is linear (true for a quadratic approximator).

C. Stability of Stacked ADP Algorithm

The sADP algorithm first solves the optimization problem for the critic (13), and then solves the actor optimization problem (12). The latter uses the stack of Q-function approximants with the parameters obtained in the first step of the sADP strategy as the cost function. Using the introduced constraints, stability of the closed-loop dynamics can be proven via the following theorem.

Theorem 1: Under the sADP control strategy (13), (12), and Assumptions 1, 2, the resulting optimal control renders the origin of the closed-loop uniformly asymptotically stable in the sense of Lyapunov (see e.g. [17]).

Proof: The proof is done in two steps. First, it is shown that shifting the optimal control sequence $u^*(\bullet|k)$ by one and adding a leading zero is feasible. Second, a Lyapunov function for the closed-loop system is constructed.

Step 1. By assumption, $u^*(\bullet|k)$ is the optimal feasible sequence starting at the time step k. Consider a candidate control sequence $u(i-1|k+1)$, $i-1=0,\dots,N-1$ defined as follows:

$$u(\bullet|k+1) := \{u^*(1|k),\dots,u^*(N|k),0\}.$$

Clearly, $u(\bullet|k+1)$ is feasible at time step $k+1$ with the corresponding state sequence

$$x(\bullet|k+1) = \{x^*(2|k),\dots,x^*(N|k),0\},$$

where $x^*(\bullet|k)$ is the optimal state sequence under the optimal control sequence $u^*(\bullet|k)$.

Step 2. Consider the following function:

$$\bar{Q}(w^*(\bullet|k),x(\bullet|k+1),u(\bullet|k+1)) := \sum_{i=1}^{N} w^*(i|k)^\top \varphi(x(i|k+1),u(i-1|k+1))$$

where the critic parameters $w^*(\bullet|k)$ of the time step k are used. Split the sum into:

$$\bar{Q}(w^*(\bullet|k),x(\bullet|k+1),u(\bullet|k+1)) = \sum_{i=1}^{N-1} w^*(i|k)^\top \varphi(x(i|k+1),u(i-1|k+1)) + w^*(N|k)^\top \varphi(x(N|k+1),u(N-1|k+1)),$$

where the last term, under the policy $u(\bullet|k+1)$ and due to the positive definiteness of the Q-function approximation model, equals zero. Hence

$$\bar{Q}(w^*(\bullet|k),x(\bullet|k+1),u(\bullet|k+1)) = \sum_{i=1}^{N-1} w^*(i|k)^\top \varphi(x(i|k+1),u(i-1|k+1)),$$

which in turn equals

$$\begin{aligned} & \sum_{i=1}^{N} w^*(i|k)^\top \varphi(x^*(i|k),u^*(i-1|k)) - \\ & w^*(1|k)^\top \varphi(x^*(1|k),u^*(0|k)) \\ = \; & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k),u^*(\bullet|k)) - \\ & w^*(1|k)^\top \varphi(x^*(1|k),u^*(0|k)). \end{aligned}$$

By optimality,

$$\begin{aligned} & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k+1),u^*(\bullet|k+1)) \\ \leq \; & \bar{Q}(w^*(\bullet|k),x(\bullet|k+1),u(\bullet|k+1)) \end{aligned}$$

where $x^*(\bullet|k+1)$ and $u^*(\bullet|k+1)$ are the optimal state and control sequences, respectively, at the step $k+1$. Therefore, under the parameter stack $w^*(\bullet|k)$, the following decay conditions holds:

$$\begin{aligned} & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k+1),u^*(\bullet|k+1)) \\ \leq \; & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k),u^*(\bullet|k)) - \\ & w^*(1|k)^\top \varphi(x^*(1|k),u^*(0|k)). \end{aligned}$$

By the constraint in (13), for the optimal state and control sequence, it holds that

$$\begin{aligned} & \bar{Q}(w^*(\bullet|k+1),x^*(\bullet|k+1),u^*(\bullet|k+1)) \\ \leq \; & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k+1),u^*(\bullet|k+1)) \end{aligned}$$

and, therefore, it can be concluded that the following inequalities hold:

$$\begin{aligned} & \bar{Q}(w^*(\bullet|k+1),x^*(\bullet|k+1),u^*(\bullet|k+1)) \\ \leq \; & \bar{Q}(w^*(\bullet|k),x(\bullet|k+1),u(\bullet|k+1)) \\ \leq \; & \bar{Q}(w^*(\bullet|k),x^*(\bullet|k),u^*(\bullet|k)) - \\ & w^*(1|k)^\top \varphi(x^*(1|k),u^*(0|k)). \end{aligned}$$

But since $w^*(1|k)^\top \varphi(x(1|k),u(0|k))$ is p. d., it follows that $\bar{Q}(w^*(\bullet|k),x^*(\bullet|k),u^*(\bullet|k))$ is monotone decreasing with k with at least the rate $w^*(1|k)^\top \varphi(x^*(1|k),u^*(0|k)) > 0$. Since $\bar{Q}$ is p. d. and zero at the origin, it is a Lyapunov function for the closed-loop system. By the LaSalle-Yoshizawa theorem [18, Theorem 2.1], the origin of the closed-loop system is uniformly asymptotically stable. ■

Remark 4: In practice, the zero terminal constraint is hard to satisfy exactly. Instead, terminal set constraints can be used (refer, e. g., to [9]). Another option is to introduce a sufficiently high terminal cost, e. g., of the form:

$$x(k+N|k)^\top S\, x(k+N|k)$$

where $S = S^\top \gg I$, or, e. g., in the form

$$x(k+N|k)^\top x(k+N|k) < \varepsilon,$$

with ε small. For further discussion, please refer to MPC surveys, e. g., [23], [29].

Theorem 1 demonstrates principal applicability of "MPC-like" stability analysis ideas to sADP. Certain additional constraints related to the critic need to hold. To satisfy them, a suitable model of the critic can be used (see Remark 1). This can be seen as a certain disadvantage of ADP since the number of the design parameters is larger than in MPC. On the other hand, in the following case study, it is shown that sADP may deliver better performance than MPC. In general, a particular choice of the control structure depends on the application and computational resources available. Table II summarizes the results and technical details of the two control strategies.

TABLE II

OVERVIEW OF THE MPC AND SADP SCHEME

MPC	sADP
$\min_{u(\bullet\|k)} \quad \sum_{i=1}^{N} r(x(i\|k), u(i-1\|k))$ $x(i\|k) = f(x(i-1\|k), u(i-1\|k))$ $\forall i = 1, \ldots, N$ $u(i-1\|k) \in \mathbb{U},\ x(i\|k) \in \mathbb{X},\ x(N\|k) = 0.$	1.) $\min_{w(\bullet\|k)} \quad \sum_{i=1}^{N} e(\bullet\|k)^2$ $w(k)^\top \varphi(x,u)$ is p. d. $\sum_{i=1}^{N} w(i\|k)^\top \varphi(x(i\|k), u(i-1\|k)) \leq$ $\sum_{i=1}^{N} w(i\|k-1)^\top \varphi(x(i\|k), u(i-1\|k)).$ 2.) $\min_{u(\bullet\|k)} \quad \sum_{i=1}^{N} \hat{Q}(x(i\|k), u(i-1\|k))$ $x(i\|k) = f(x(i-1\|k), u(i-1\|k)),\ \forall i = 1, \ldots, N$ $u(i-1\|k) \in \mathbb{U},\ x(i\|k) \in \mathbb{X},\ x(N\|k) = 0.$

IV. CASE STUDY USING MPC AND STACKED ADP

Consider the following particular time-discrete variant (with the sample time 0.04) of the van der Pol oscillator:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) + 0.04(x_2(k)) \\ x_2(k) + 0.04(-x_1(k) + 0.5(1 - x_1^2(k))x_2(k) + u(k)) \end{bmatrix},$$

with the initial state $[-5 \quad 8]^\top$. Let the running cost be quadratic:

$$r(x(k), u(k)) = \frac{1}{2} x^\top(k) \tilde{Q} x(k) + \frac{1}{2} u^\top(k) R u(k)$$

where $\tilde{Q} = I$, the identity matrix, and $R = 0.5$. sADP is implemented with a quadratic actor of the form:

$$\bar{Q}(x(k), u(k)) = w(k)^\top \varphi(x(k), u(k)),$$

where $\varphi_1 = [x_1^2(k) \quad x_2^2(k) \quad u^2(k)]^\top$. At each time step k, the optimization problem (13) is solved to find the critic parameters $w^*(\bullet|k)$. Then, the actor is updated via (12) and the first element of the control sequence is applied. In the current case study, $\mathbb{X} = \mathbb{R}^2$ and $\mathbb{U} = \mathbb{R}$.

The stack length N is set to 4. A particular variant of MPC and sADP (see Table II) are applied to the system. Both algorithms are compared using the following performance mark:

$$\sum_{k=1}^{N_{\text{sim}}} r(x(k), u(k)).$$

where N_{sim} is the total number of time steps in simulation.

In Fig. 1, the state under both schemes is shown. It can be seen that the state x_1 under sADP converges somewhat slower, but has a smaller overshoot during the transient time.

The policies of MPC and sADP can be seen in Fig. 2. Fig. 3 illustrates the behavior of the first element of the stack $w^*(i|k), i = 1, \ldots, N$ for each time step k. It can be seen that this first element, i. e., $w^*(1|k)$ increases at some times of the simulation, which might make one to suspect that the second constraint in (13) is violated. This is, however, not so and the constraint is satisfied at all times. In fact, it constraints the whole stack $w^*(i|k), i = 1, \ldots, N$ whereas the single elements thereof might increase or decrease.

In the simulation, sADP yielded a better performance mark of 7279 compared to the MPC with 9345 which might

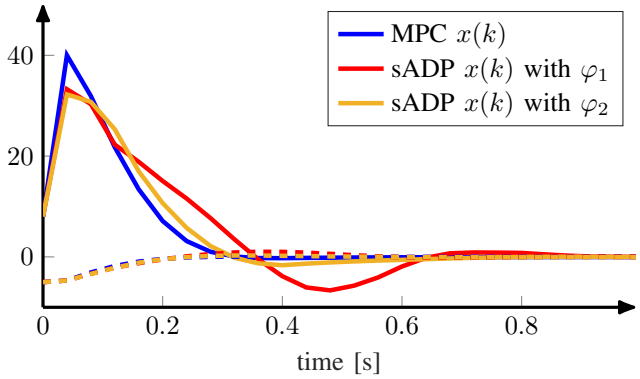

Fig. 1. State trajectory under sADP and MPC.

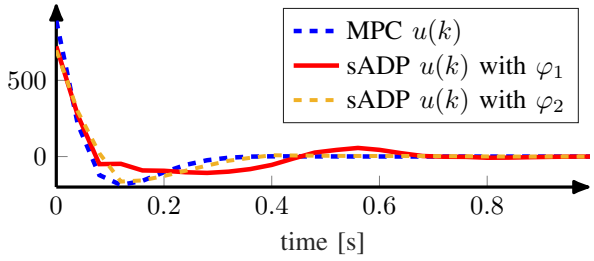

Fig. 2. The control of the MPC and sADP scheme with different φ.

be due to a better balance between the state convergence speed and control effort. However, it should be noted that, in general, performance depends on the concrete application and a choice of the control strategy is determined by a number of factors. In particular, computational complexity should be taken into account. In the current study, sADP with the regressor φ_1 required approximately 16 % more time to compute than MPC. The sADP algorithm with a regressor $\varphi_2 = [x_1^4(k) \ x_2^2(k) \ u^4(k)]^\top$ yielded even better performance mark of 6607, however, at the cost of 40 % higher computation time than that of MPC. The corresponding state trajectory is shown in Fig. 1. Furthermore, sADP introduces additional design factors which are up to the

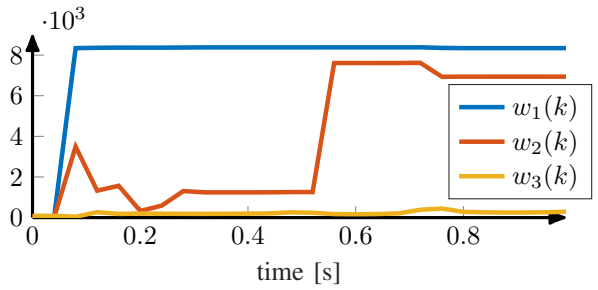

Fig. 3. The $w^*(1|\bullet)$-parameters of the critic of the sADP scheme under $\varphi_1 = [x_1^2(k) \quad x_2^2(k) \quad u^2(k)]^\top$.

user. These factors are related to the approximation model in the critic (refer to Remark 1). In particular, a bad choice of the approximation model might lead to unsatisfying results and difficulties with fulfilling the additional critic constraints, not to mention additional computational effort. A particular choice of the control strategy depends on the application and computational resources, whereas the benefits and disadvantages of ADP and MPC should be taken into account.

V. Conclusion

The current work was dedicated to a brief comparative study of MPC and ADP with a particular implementation, stacked ADP. As the major advantages of MPC, solid stability analysis results, natural incorporation of constraints and simplicity of optimization problem formulation are recognized. In ADP, the major advantage is the ability of the critic to capture information on the infinite horizon of the running costs. However, new design factors come into place, such as the critic model structure. The main focus of this work was to transfer some basic stability analysis ideas from MPC to stacked ADP. Certain additional constraints concerning the critic needed to be satisfied. Nevertheless, a case study demonstrated usability of the suggested approach. Further analyses on interconnections and mutual benefits of the two optimal control strategies are required since the current study was preliminary.

References

[1] V. Adetola, D. DeHaan, and M. Guay. Adaptive model predictive control for constrained nonlinear systems. *Syst Control Lett*, 58:320–326, 2009.

[2] A. Al Tamimi, F. L. Lewis, and M. Abu Khalaf. Discrete-Time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof. *IEEE T Syst Man Cy B*, 38(4):943–949, 2008.

[3] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical Algebraic Decomposition I: The Basic Algorithm. *SIAM J Comput*, 13(4):865–877, 1984.

[4] R. Bellman. *Dynamic Programming*. Princeton University Press, 1st edition, 1957.

[5] D. P. Bertsekas. Dynamic Programming and Suboptimal Control: A survey from ADP to MPC. *Eur J Control*, 11(4):310–334, 2005.

[6] D. P Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. 3. Athena Scientific, 1996.

[7] D. Blackwell. Discounted Dynamic Programming. *Ann Math Stat*, 36(1):226–235, 1965.

[8] W. A. Cebuhar and V. Costanza. Approximation procedures for the optimal control of bilinear and nonlinear systems. *J Optimiz Theory App*, 43(4):615–627, 1984.

[9] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing Receding-Horizon Control of Nonlinear Time-Varying Systems. *IEEE T Automat Contr*, 43(7):1030–1036, 1998.

[10] S. L. de Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE T Automat Contr*, 45(6):1053–1071, 2000.

[11] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel. Model predictive control and reinforcement learning as two complementary frameworks. *International Journal of Tomography & Statistics*, 6, 2007.

[12] H. Fukushima, T.-H. Kim, and T. Sugie. Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica*, 43(2):301–308, 2007.

[13] D. Görges. Relations between Model Predictive Control and Reinforcement Learning. *Proceedings of the IFAC World Congress 2017*, pages 5071–5079, July 2017. Toulouse, France.

[14] A. Heydari. Theoretical and Numerical Analysis of Approximate Dynamic Programming with Approximation Errors. *J Guid Control Dynam*, 39(2):301–311, 2015.

[15] A. Heydari and S. N. Balakrishnan. Global optimality of approximate dynamic programming and its use in non-convex function minimization. *Appl Soft Comput*, 24:291–303, 2014.

[16] R. E. Kalman. Contributions to the theory of optimal control. *Boletin Sociedad Matematica Mexicana*, 5:102–119, 1960.

[17] R. E. Kalman and J. E. Bertram. Control System Analysis and Design Via the Second Method of Lyapunov: IiDiscrete-Time Systems. *J Basic Eng-T ASME*, 82(2):394–400, 1960.

[18] M. Krstić, I. Kanellakopoulos, and Kokotović. *Nonlinear and Adaptive Control Design*. Wiley-Interscience, 1st edition, 1995.

[19] T. Landelius and H. Knutsson. Greedy Adaptive Critics for LQR Problems: Convergence proofs. Technical report, Linköping University, Sweden, 1996. LiTH-ISY-R-1896.

[20] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circ Syst Mag*, 9(3):32–50, 2009.

[21] D. Liu and Q. Wei. Policy Itereation Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems. *IEEE T Neur Net Lear*, 25(3):621–634, 2014.

[22] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[24] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE T Automat Contr*, 38(11):1623–1633, 1993.

[25] R. R. Negenborn, B. De Schutter, M. A. Wierung, and H. Hellendoorn. Learning-based Model Predictive Control For Markov Decision Processes. *IFAC Proceedings Volumes*, 38(1):354–359, 2005.

[26] P. Osinenko, T. Göhrt, G. Devadze, and S. Streif. Stacked adaptive dynamic programming with unknown system model. *Proceedings of the IFAC World Congress 2017*, pages 4218–4223, July 2017. Toulouse, France.

[27] W. B. Powell. *Approximate Dynamic Programming - Solving the Curses of Dimensionality*. John Wiley & Sons, 2007.

[28] S. V. Raković and M. Lazar. Minkowski terminal cost functions for MPC. *Automatica*, 48(10):2721–2725, 2012.

[29] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE T Automat Contr*, 44(3):1999, 1999.

[30] Y. Sokolov, R. Kozma, L. D. Werbos, and P. J. Werbos. Complete stability analysis of a heuristic approximate dynamic programming control design. *Automatica*, 59:9–18, 2015.

[31] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[32] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT Press Cambridge, 1998.

[33] C. Watkins. *Learning from delayed rewards*. Ph.d. dissertation, Dept. Comput. Sci., Cambridge Univ., May 1989.

[34] P. J. Werbos. A Menu of Designs for Reinforcement Learning Over Time. In W. T. Miller III, R. S. Sutton, and P. J. Werbos, editors, *Neural Networks for Control*, chapter 3. The MIT Press, 1995.

[35] H. Zhang, D. Liu, Y. Luo, and D. Wang. *Adaptive Dynamic Programming for Control*. Communications and Control Engineering. Springer-Verlag London, 2013.