Approximately optimal nonlinear stabilization with preservation of the Lyapunov function property

Lars Grüne and Oliver Junge
Submitted to the CDC 2007, March 2007

Abstract— We present an approximate optimization approach to the computation of stabilizing feedback laws using a partitioning of the state space and a corresponding approximation of the optimal value function of the problem. By including the discretization errors into the optimal control formulation we are able to compute approximate optimal value functions which preserve the Lyapunov function property and corresponding optimally stabilizing feedback laws which are constant on each partition element. The actual computation uses efficient graph theoretic algorithms.

I. INTRODUCTION

Optimization based stabilization schemes, either based on infinite horizon or on receding finite time horizon optimal control formulations, can be implemented using online or offline optimization techniques. In this paper we consider the latter technique, in which the resulting optimal value function and the optimal feedback law is computed offline and stored. This approach has received considerable attention during the last years, often under the name of *explicit* optimization schemes (see, e.g., [6] for a survey), in particular for hybrid systems or robust control problems where the resulting optimization problems are typically difficult to be solved online. Common feature of many of these approaches is that the optimal value function can be used as a Lyapunov function in order to ensure asymptotic stability of the resulting closed loop system.

While under suitable structural assumptions on the underlying system the optimization problem can in principle be solved exactly, e.g., in the case of constrained linear or piecewise affine systems [1], [14], [3], in the general nonlinear setting this is not feasible and approximations have to be used, typically involving some kind of partitioning of the state space. In order to guarantee the stabilizing property of the resulting approximately optimal feedback law, the computed approximately optimal value function should still be a Lyapunov function, i.e., it should decay along the approximately optimal solution. This property can be ensured in various ways, e.g., by using adaptive state partitioning strategies based on numerical error estimation techniques in order to ensure that the local error smaller than the decay of the exact Lyapunov function [7], [10], [15]. Although attractive due to its relation to approximation theoretic considerations, this approach may lead to rather fine (and thus memory consuming) partitions because the

L. Grüne is with the Mathematical Institute, University of Bayreuth, 95440 Bayreuth, Germany, lars.gruene@uni-bayreuth.de
O. Junge is with the Faculty of Mathematics, Munich University of Technology, 85748 Garching, Germany, junge@ma.tum.de

Lyapunov function property can only be established via sufficiently small local errors. Another approach is to design the partitioning in such a way that the approximated dynamics on the partition are in a suitable sense consistent with the exact dynamics, cf. e.g. [4], [5]. Here systems theoretic arguments can be used in order to establish stability on rather coarse partitions, however, the approach may lead to highly complicated partition structures.

In this paper we propose a different approach in order to preserve the Lyapunov function property in the approximation process. We consider simple state space partitionings and ensure consistency of the approximate dynamics by interpreting the discretization errors introduced by the partitioning as perturbations. These perturbations are then explicitly taken into account in the optimization which consequently becomes a dynamic game, i.e., a min–max problem. Our approach is developed within the graph theoretic setting introduced in [11] and applied to stabilization problems in [7], which relies on using Dijkstra's Algorithm for the resulting optimal control problem on the graph. Cornerstone of our approach is the observation that the key idea of Dijkstra's Algorithm, i.e., the reformulation of the minimization as a computationally efficiently solvable sorting problem, can be carried over to the min–max setting, cf. [9]. We prove that this approach indeed yields the desired Lyapunov function property of the approximated optimal value function. Furthermore, we show that the corresponding stabilizing feedback law is actually constant on each partition element. In particular, while in [11], [7], [9] the partitioning was interpreted as a *discretization* procedure in the spirit of the numerical analysis literature, this allows for a re-interpretation of the partitioning as a *quantization* of the continuous state space which opens the way for a variety of potential new applications for our method which we discuss at the end of the paper.

The paper is organized as follows: In Section II we pose the problem and summarize the main results from [11], [7], [9]. In Section III we develop our approach by interpreting the discretization error as a perturbation which acts as an opponent in the "stabilization game" and prove that the resulting optimal value function indeed has the Lyapunov function property and that the stabilizing feedback law can be chosen piecewise constant on each partition element. In Section IV we illustrate our approach by a numerical example. Finally, in Section V we give some conclusions and discuss further applications of our approach which are subject of ongoing research.

II. SETTING AND PRELIMINARIES

Our goal is to optimally stabilize the origin of the discrete-time control system

$$x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \ldots, \tag{1}$$

subject to the continuous instantaneous cost $g(x_k, u_k) \geq 0$.

To this end, we would like to construct an approximate *optimal feedback law* $u(x_k)$ such that the origin is an asymptotically stable equilibrium for the resulting closed loop system

$$x_{k+1} = f(x_k, u(x_k)), \quad k = 0, 1, \ldots \tag{2}$$

For simplicity of exposition, in this paper we consider a compact state space $X \subset \mathbb{R}^d$ but we remark that this setting is easily extended to more general compact metric spaces X which may in particular contain discrete state components, i.e., (1) may be a hybrid system, cf. [8] for details on how to treat hybrid systems within our approach. Likewise, while a typical system class we consider are sampled-data continuous time systems, we do not explicitly elaborate the sampled-data nature of the system but rather state our problem in discrete time, keeping in mind that the instantaneous cost $g(x_k, u_k)$ may contain integral expressions along the continuous time solution over a sampling interval. Finally, we note that our approach is easily extended to the stabilization of arbitrary compact target sets T, see [8].

A. The optimality principle

In what follows, we assume the map $f : X \times U \to \mathbb{R}^d$ to be continuous on some compact subset $X \times U \subset \mathbb{R}^d \times \mathbb{R}^m$ which contains the origin, $f(0,0) = 0$ and $\inf_{u \in U} g(x,u) > 0$ for all $x \neq 0$.

For a given control sequence $\mathbf{u} \in U^{\mathbb{N}}$ there is a unique associated trajectory $(x_k(x, \mathbf{u}))_{k \in \mathbb{N}}$ of (1). Let $\mathcal{U}(x) = \{\mathbf{u} \in U^{\mathbb{N}} : x_k(x, \mathbf{u}) \to 0\}$ denote the set of asymptotically controlling sequences for $x \in X$ and $S = \{x \in X : \mathcal{U}(x) \neq \emptyset\}$ the *stabilizable subset* $S \subset X$. The total cost along a controlled trajectory is given by

$$J(x, \mathbf{u}) = \sum_{k=0}^{\infty} g(x_k(x, \mathbf{u}), u_k) \in [0, \infty]. \tag{3}$$

Our feedback construction will be based on an approximation of the *(optimal) value function* $V : S \to [0, \infty]$, $V(x) = \inf_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} J(x, \mathbf{u})$, using the fact that V satisfies the *optimality principle*

$$V(x) = \inf_{u \in U} \{g(x,u) + V(f(x,u))\} =: L[V](x). \tag{4}$$

The operator L acting on real valued functions on X is called the *dynamic programming operator* and an alternative characterization of V is that it is the unique fixed point of the equation $V = L[V]$ which satisfied the boundary condition $V(0) = 0$.

Using the optimality principle, an optimal stabilizing feedback is given by

$$u(x) = \operatorname*{argmin}_{u \in U} \{g(x,u) + V(f(x,u))\}, \tag{5}$$

whenever this minimum exists. The key property in order to prove asymptotic stability is the fact that by the (obvious) inequality

$$V(x) \geq g(x, u(x)) + V(f(x, u(x))) \tag{6}$$

the optimal value function is a Lyapunov function of the closed loop system, provided V is finite, positive definite and proper.[1]

B. Direct discretization

In this section we sketch the discretization procedure proposed in [11]. We are going to approximate V by functions which are piecewise constant. Let $\mathcal{P}$ be a partition of X, i.e. a collection of pairwise disjoint subsets P of X which covers the state space X. For a state $x \in X$ we let $\rho(x) \in \mathcal{P}$ denote the element in the partition which contains x. In order to simplify notation, in the sequel we identify any subset $\{P_1, \ldots, P_k\} \subseteq \mathcal{P}$ with the corresponding subset $\bigcup_{i=1,\ldots k} P_i \subseteq X$, and vice versa.

Let $\mathbb{R}^{\mathcal{P}}$ be the subspace of the space $\mathbb{R}^X$ of all real valued functions on X which are piecewise constant on the elements of the partition $\mathcal{P}$. The map

$$\varphi[v](x) = \inf_{x' \in \rho(x)} v(x')$$

is a projection from $\mathbb{R}^X$ onto $\mathbb{R}^{\mathcal{P}}$. Using this projection, we can define the *discretized dynamic programming operator*

$$L_{\mathcal{P}} = \varphi \circ L : \mathbb{R}^{\mathcal{P}} \to \mathbb{R}^{\mathcal{P}}.$$

Under the boundary condition $V_{\mathcal{P}}(x) = 0$ for all $x \in \rho(0)$ this operator has a unique fixed point $V_{\mathcal{P}}$ — the approximate (optimal) value function.

Since $V_{\mathcal{P}}$ is constant on each partition element $P \in \mathcal{P}$, we write $V_{\mathcal{P}}(P)$ for the value of $V_{\mathcal{P}}$ on P. Using this notation, one can show (see [9]) that the fixed point equation $V_{\mathcal{P}} = L_{\mathcal{P}}[V_{\mathcal{P}}]$ is equivalent to the *discrete optimality principle*

$$V_{\mathcal{P}}(P) = \min_{P' \in \mathcal{F}(P)} \{\mathcal{G}(P, P') + V_{\mathcal{P}}(P')\}, \tag{7}$$

where the map $\mathcal{F}$ is given by

$$\mathcal{F}(P) = \rho(f(P, U)) = \{P' \in \mathcal{P} : P' \cap f(P, U) \neq \emptyset\} \tag{8}$$

and the cost function $\mathcal{G}$ by

$$\mathcal{G}(P, P') = \inf\{g(x,u) \mid x \in P, f(x,u) \in P', u \in U\}. \tag{9}$$

Note that the approximate value function $V_{\mathcal{P}}(P)$ is the length of the shortest path from P to $\rho(0)$ in the weighted directed graph $(\mathcal{P}, E)$, where the set of edges is defined by $E = \{(P, P') : P' \in \mathcal{F}(P)\}$ and the edge (P, P') is weighted by $\mathcal{G}(P, P')$. As such, it can be computed by, e.g., Dijkstra's algorithm, cf. [11].

---

[1]These properties of V can be ensured by suitable asymptotic controllability properties and bounds on g whose details are not important for the considerations in this paper.

C. Dynamic games

As we will see later, in many cases the direct discretization described above is not very efficient. It will turn out that a more efficient discretization procedure can be obtained by employing ideas from dynamic game theory applied to perturbed system — even if our basic system (1) does not contain perturbations. In this section we briefly sketch how our approach can be extended to dynamic games, for details we refer to [9].

A *dynamic game* is a map $F : X \times U \times W \to X$, where $X \subset \mathbb{R}^d$ is a closed set, $U \subset \mathbb{R}^m$, $W \in \mathbb{R}^\ell$, together with a cost function $G : X \times U \times W \to [0, \infty)$. For a given initial state $x \in X$, a given control sequence $\mathbf{u} = (u_k)_{k \in \mathbb{N}} \in U^\mathbb{N}$ and a given perturbation sequence $\mathbf{w} = (w_k)_{k \in \mathbb{N}} \in W^\mathbb{N}$, the *trajectory* of the game is given by the sequence $(x_k(x, \mathbf{u}, \mathbf{w}))_{k \in \mathbb{N}}$ with

$$x_{k+1} = F(x_k, u_k, w_k), \quad k = 0, 1, 2, \dots.$$

Specifying a target set T, we define the *accumulated cost* as

$$J_{F,G}(x, \mathbf{u}, \mathbf{w}) = \sum_{k=0}^{k(T, x, \mathbf{uw})} G(x_k(x, \mathbf{u}, \mathbf{w}), u_k, w_k),$$

with $k(T, x, \mathbf{uw}) := \inf\{k \geq 0 \,|\, x_k(x, \mathbf{u}, \mathbf{w}) \in T\}$. This yields the *upper value function*

$$V_{F,G}(x) = \sup_{\beta \in \mathcal{B}} \inf_{\mathbf{u} \in U^\mathbb{N}} J_{F,G}(x, \mathbf{u}, \beta(\mathbf{u})), \quad x \in X, \quad (10)$$

of this game. Here, $\mathcal{B}$ denotes the set of all nonanticipating strategies $\beta : U^\mathbb{N} \to W^\mathbb{N}$, i.e. all strategies $\beta : U^\mathbb{N} \to W^\mathbb{N}$ satisfying

$$u_k = u'_k \quad \forall k \leq K \quad \Rightarrow \quad \beta(\mathbf{u})_k = \beta(\mathbf{u}')_k \quad \forall k \leq K$$

for any two control sequences $\mathbf{u} = (u_k)_k, \mathbf{u}' = (u'_k)_k \in U^\mathbb{N}$. By standard dynamic programming arguments [2] one sees that this function fulfills and is uniquely determined by the optimality principle

$$V_{F,G}(x) = \inf_{u \in U} \sup_{w \in W} \{G(x, u, w) + V_{F,G}(F(x, u, w))\}$$
(11)

for $x \notin T$ together with the boundary condition $V_{F,G}|_T \equiv 0$. If G does not depend on w, then this equation can be written as

$$V_{F,G}(x) = \inf_{u \in U} \left\{ G(x, u) + \sup_{x' \in F(x, u, W)} V_{F,G}(x') \right\}. \quad (12)$$

Note that in this equation the parameterization of F with w is not needed any more since it is sufficient to know the set valued image $F(x, u, W)$. The discretization described in the following section will be based on this observation.

D. Discretization of the dynamic game

We employ the same approach as in Section II-B in order to discretize (11). Note that the setting in Section II-B can be seen as a special case of the more general situation here using $W = \{0\}$ and $f(x, u, w) = f(x, u)$, which is why we keep the same notation. In our perturbed setup, one can

show [9] that the corresponding discrete optimality principle is given by

$$V_\mathcal{P}(P) = \inf_{\mathcal{N} \in \mathcal{F}(P)} \left\{ \mathcal{G}(P, \mathcal{N}) + \sup_{P' \in \mathcal{N}} V_\mathcal{P}(P') \right\} \quad (13)$$

for $P \cap T = \emptyset$ with boundary condition $V_\mathcal{P}(P) = 0$ for $P \cap T \neq \emptyset$, where

$$\mathcal{F}(P) = \{\rho(F(x, u, W)) : (x, u) \in P \times U\}. \quad (14)$$

and

$$\mathcal{G}(P, \mathcal{N}) = \inf \left\{ G(x, u) : \begin{array}{l} (x, u) \in P \times U, \\ \rho(F(x, u, W)) = \mathcal{N} \end{array} \right\}.$$

Note the difference of $\mathcal{F}(P)$ compared to (8): while in (8) $\mathcal{F}(P)$ was a subset of $\mathcal{P}$, in (14) $\mathcal{F}(P)$ now is a set of subsets $\mathcal{N}$ of $\mathcal{P}$. Thus, the map $\mathcal{F}$, together with the cost function $\mathcal{G}$ can be interpreted as a directed weighted hypergraph $(\mathcal{P}, E)$ with the set $E \subset \mathcal{P} \times 2^\mathcal{P}$ of hyperedges given by

$$E = \{(P, \mathcal{N}) : \mathcal{N} \in \mathcal{F}(P)\}.$$

Unlike other computational methods for optimal control problems, it was shown in [9] that the main trick in Dijkstra's method, i.e., the reformulation of the minimization as a computationally efficiently solvable sorting problem, can be carried over to the min–max setting without increasing the computational complexity, see [16] for details.

Denoting $\mathcal{T} := \{P \in \mathcal{P} : P \cap T \neq \emptyset\}$, the algorithm reads:

Algorithm 1: MIN–MAX DIJKSTRA$((\mathcal{P}, E), \mathcal{G}, \mathcal{T})$
1 for each $P \in \mathcal{P}$ set $V(P) := \infty$
2 for each $P \in \mathcal{T}$ set $V(P) := 0$
3 $\mathcal{Q} := \mathcal{P}$
4 while $\mathcal{Q} \neq \emptyset$
5      $P := \operatorname{argmin}_{P' \in \mathcal{Q}} V(P')$
6      $\mathcal{Q} := \mathcal{Q} \backslash \{P\}$
7   for each $(Q, \mathcal{N}) \in E$ with $P \in \mathcal{N}$
8   if $\mathcal{N} \subset \mathcal{P} \backslash \mathcal{Q}$ then
9   if $V(Q) > \mathcal{G}(Q, \mathcal{N}) + V(P)$ then
10                 $V(Q) := \mathcal{G}(Q, \mathcal{N}) + V(P)$

III. THE DISCRETIZATION AS A PERTURBATION

Consider the feedback that results from the optimality principle (5) using the directly discretized approximate value function $V_\mathcal{P}$ from (7), i.e.,

$$u_\mathcal{P}(x) = \operatorname{argmin}_{u \in U} \{g(x, u) + V_\mathcal{P}(f(x, u))\}.$$

It was shown in [7] that this feedback will practically stabilize the system under suitable asymptotic controllability conditions. Numerical experiments (e.g., the one in Section IV, below), however, reveal that typically a rather fine partition is needed in order to achieve stability of the closed loop system. Furthermore, even on this fine partition, the approximate optimal value function is not a Lyapunov function for the closed loop system, i.e. it does not decay monotonically along system trajectories. The reason is that the approximate optimal value function is rather heavily underestimated by the above discretization scheme. Whenever the trajectory

of the closed loop system enters a certain element of the partition, it may be impossible to reach another element with a lower value from the current state (but it is possible when starting from another state in the same element). Formally, this is reflected by the two inequalities

$$V_{\mathcal{P}}(x) \leq V(x) \qquad (15)$$

and

$$
\begin{aligned}
V_{\mathcal{P}}(x) \;\; &\leq \;\; \min_{u \in U}\{g(x,u) + V_{\mathcal{P}}(f(x,u))\} \qquad (16)\\
&= \;\; g(x, u_{\mathcal{P}}(x)) + V_{\mathcal{P}}(f(x, u_{\mathcal{P}}(x)))
\end{aligned}
$$

which were shown in [11], [7]. In terms of the Lyapunov function property (6), inequality (16) delivers exactly the *opposite* than what is needed in order to prove asymptotic stability.

In order to cope with this phenomenon — i.e., in order to construct an approximation which satisfies (6) — we are going to use the dynamic game formulation outlined above. The idea is to additionally incorporate the discretization error as a perturbation of the original control system. More precisely, in our context, instead of dealing with the single-valued system $x \mapsto f(x, u(x))$, we consider the multi-valued system $x \mapsto f(\rho(x), u(x))$. When computing the value of a given state x under the multi-valued dynamics, one assumes the "worst case" and sums the one step cost $g(x, u(x))$ with the *maximum* of V over the set $f(\rho(x), u(x))$.

A. The discretization is the opponent

Let us be more precise: we fix a partition $\mathcal{P}$, pick a target set $T \ni 0$ and consider a dynamic game with

$$F(x, u, W) = f(\rho(x), u) \qquad (17)$$

for every (x, u) and define

$$G(x, u) = \sup_{x' \in \rho(x)} g(x, u) \qquad (18)$$

where f and g are the control system and the instantaneous cost defined in Section II-A. Observe that we do not need to construct a parameterization $F(x, u, w)$ for $w \in W$ because, as noted above, in the discretization this parameterization is not needed anyway.

Using this specification of F and G, the discretization from Section II-D leads to the graph map

$$\mathcal{F}(P) = \{\rho(f(P, u)) : u \in U\} \qquad (19)$$

and the weights

$$\mathcal{G}(P, \mathcal{N}) = \inf\left\{\sup_{x \in P} g(x, u) : u \in U, f(P, u) = \mathcal{N}\right\} \qquad (20)$$

which can be used in order to define an approximate optimal value function $V_{\mathcal{P}}$ according to (13). By enlarging the target set T, if necessary, we can assume that T is a union of partition elements $P \in \mathcal{P}$, i.e., $T \subset \mathcal{P}$.

The following theorem shows the crucial properties of this approximate value function $V_{\mathcal{P}}$, in particular, that this function satisfies the opposite inequalities compared to (15),

(16), when the terminal set T is appropriately included in the formulation.

Theorem 1: Let V denote the optimal value function of the optimal control problem (1), (3) with cost function g and let $V_{\mathcal{P}}$ denote the approximate optimal value function of the game F, G given by (17), (18) on a given partition $\mathcal{P}$ with target set $T \subset \mathcal{P}$ with $0 \in T$. Then,

$$V(x) - \max_{y \in T} V(y) \leq V_{\mathcal{P}}(x) = V_{F,G}(x), \qquad (21)$$

i.e. $V_{\mathcal{P}}$ coincides with $V_{F,G}$ and is an upper bound for $V - \max V|_T$. Furthermore, $V_{\mathcal{P}}$ satisfies

$$V_{\mathcal{P}}(x) \geq \min_{u \in U}\{g(x,u) + V_{\mathcal{P}}(f(x,u))\} \qquad (22)$$

for all $x \in X \setminus T$.

Proof: We first note that $V_{F,G}$ is actually constant on the elements of the partition $\mathcal{P}$. In fact, on T this immediately holds since T is a union of partition elements, while outside T by definition of the game F, G we have

$$
V_{F,G}(x)
$$
$$
= \inf_{u \in U}\left\{\sup_{x' \in \rho(x)} g(x', u) + \sup_{x' \in f(\rho(x), u)} V_{F,G}(x')\right\},
$$

so that $\inf_{x' \in \rho(x)} V_{F,G}(x') = V_{F,G}(x)$. On the other hand, according to [9, Proposition 4],

$$V_{\mathcal{P}}(x) = \inf_{x' \in \rho(x)} V_{F,G}(x'),$$

so that $V_{\mathcal{P}} = V_{F,G}$ which shows the equality in (21). Now for $x \notin T$ the equation (12) yields

$$
\begin{aligned}
V_{F,G}(x) \;\; &= \;\; \inf_{u \in U}\sup_{x' \in \rho(x)}\{g(x', u) + V_{F,G}(f(x', u))\}\\
&\geq \;\; \min_{u \in U}\{g(x,u) + V_{F,G}(f(x,u))\} \qquad (23)
\end{aligned}
$$

which shows (22). It remains to show the inequality in (21). In order to prove this inequality we order the elements $P_1, P_2, \ldots \in \mathcal{P}$ such that $i \geq j$ implies $V_{\mathcal{P}}(P_i) \geq V_{\mathcal{P}}(P_j)$.

Since $\inf_{u \in U} g(x, u) > 0$, the equality $V_{\mathcal{P}}(P_i) = 0$ is equivalent to $P_i \subseteq T$. By the choice of our ordering this implies that there exists $i^* \geq 1$ such that $P_i \subseteq T \Leftrightarrow i \in \{1, \ldots, i^*\}$ and consequently the inequality in (21) holds for $x \in P_1, \ldots, P_{i^*}$.

Now we proceed by induction: fix some $i \in \mathbb{N}$, assume the inequality in (21) holds for $x \in P_1, \ldots, P_{i-1}$ and consider $x \in P_i$. If $V_{\mathcal{P}}(P_i) = \infty$ there is nothing to show. Otherwise, since V satisfies the dynamic programming principle, using expression (23) we obtain

$$
\begin{aligned}
V(x) - V_{F,G}(x) \;\; &\leq \;\; \inf_{u \in U}\{g(x,u) + V(f(x,u))\}\\
&\quad - \min_{u \in U}\{g(x,u) + V_{F,G}(f(x,u))\}\\
&\leq \;\; V(f(x, u^*)) - V_{F,G}(f(x, u^*))
\end{aligned}
$$

where $u^* \in U$ realizes the minimum in (23). Now, from $g(x, u^*) > 0$ we obtain $V_{F,G}(f(x, u^*)) < V_{F,G}(x)$ implying $f(x, u^*) \in P_j$ for some $j < i$. Since by the induction assumption the inequality in (21) holds on P_j, this implies that it also holds on P_i which finishes the induction step. ∎

B. The feedback is the shortest path

In principle, the feedback value for each x can now be obtained by taking the argmin in (22). However, the specific structure of our approximation allows for a choice of $u_\mathcal{P}$ which, just as $V_\mathcal{P}$, is actually constant on each partition element. To this end we define

$$u_\mathcal{P}(x) = \operatorname*{argmin}_{u \in U} \left\{ G(x, u) + \sup_{x' \in F(x, u, W)} V_\mathcal{P}(x') \right\}.$$

Due to the construction of the game F, G, this feedback is indeed constant in each partition element. Moreover, we can directly extract $u_\mathcal{P}$ from the min–max Dijkstra Algorithm 1 by proceeding as follows: if we associate the minimizing control value $\underline{u}(P, \mathcal{N})$ to each hyperedge $(P, \mathcal{N})$,

$$\underline{u}(P, \mathcal{N})$$
$$= \operatorname*{argmin}_{u \in U} \left\{ \sup_{x \in P} g(x, u) : u \in U, \rho(f(P, u)) = \mathcal{N} \right\},$$

then this feedback is given by

$$u_\mathcal{P}(x) = \underline{u}(\rho(x), \underline{\mathcal{N}}(\rho(x))), \tag{24}$$

where

$$\underline{\mathcal{N}}(P) = \operatorname*{argmin}_{\mathcal{N} \in \mathcal{F}(P)} \left\{ \mathcal{G}(P, \mathcal{N}) + \sup_{N \in \mathcal{N}} V_\mathcal{P}(N) \right\}$$

is defining the successor hypernode with minimal value for each node in the hypergraph defined by $\mathcal{F}$.

Note that $\underline{u}(P, \mathcal{N})$ as well as $\underline{\mathcal{N}}(P)$ can be computed offline. The computation of $\underline{\mathcal{N}}(P)$ can be inserted into the min–max Dijkstra Algorithm 1 just like the computation of the successor node on the shortest path in the standard Dijkstra algorithm. The augmented algorithm reads as follows:

Algorithm 2: MIN–MAX DIJKSTRA WITH FEEDBACK $((\mathcal{P}, E), \mathcal{G}, \mathcal{T})$

1 for each $P \in \mathcal{P}$ set $V(P) := \infty$ and $\underline{\mathcal{N}}(P) := \emptyset$
2 for each $P \in \mathcal{T}$ set $V(P) := 0$
3 $\mathcal{Q} := \mathcal{P}$
4 while $\mathcal{Q} \neq \emptyset$
5      $P := \operatorname*{argmin}_{P' \in \mathcal{Q}} V(P')$
6      $\mathcal{Q} := \mathcal{Q} \backslash \{P\}$
7   for each $(Q, \mathcal{N}) \in E$ with $P \in \mathcal{N}$
8   if $\mathcal{N} \subset \mathcal{P} \backslash \mathcal{Q}$ then
9   if $V(Q) > \mathcal{G}(Q, \mathcal{N}) + V(P)$ then
10                 $V(Q) := \mathcal{G}(Q, \mathcal{N}) + V(P)$
11                 $\underline{\mathcal{N}}(Q) := \mathcal{N}$

Once $\underline{u}(P, \underline{\mathcal{N}}(P))$ has been computed for every partition element, the only online computation that remains to do be done is the determination of the partition element for each state on the feedback trajectory. As described in [9] this can be done efficiently by storing the partition in a binary tree.

C. Behavior of the closed loop system

Finally, we can now state the following theorem on the properties of the feedback law $u_\mathcal{P}(x)$ constructed in the last section.

Theorem 2: Consider system (1), a target set $T \subset \mathcal{P}$ with $0 \in T$ and the approximate optimal value function

$V_\mathcal{P}$ defined via (13), (19), (20). Denote the trajectory of the closed loop system (2) with feedback $u = u_\mathcal{P}$ from (24) by x_k and assume $V_\mathcal{P}(x_0) < \infty$.

Then there exists $k^* \in \mathbb{N}$ such that

$$V_\mathcal{P}(x_k) \geq g(x_k, u_\mathcal{P}(x_k)) + V_\mathcal{P}(x_{k+1}), \ k \in \{0, \dots, k^* - 1\},$$

$$x_{k^*} \in T \quad \text{and} \quad x_k \in D, \ k > k^*$$

where $D := \{x \in X : V_\mathcal{P}(x) \leq \nu\}$ and

$$\nu := \max_{P \subseteq T} \sup_{N \in \underline{\mathcal{N}}(P)} V_\mathcal{P}(N),$$

i.e., the closed loop trajectory enters the target set T at time k^* and stays in the sublevel set D, afterwards.

Proof: From the construction of $u_\mathcal{P}$ we immediately obtain the inequality

$$V_\mathcal{P}(x_k) \geq g(x_k, u_\mathcal{P}(x_k)) + V_\mathcal{P}(x_{k+1}) \tag{25}$$

for all $k \in \mathbb{N}_0$ with $x_k \in X \setminus T$. This implies the existence of k^* such that the first two properties hold since $g(x_k, u_\mathcal{P}(x_k)) > 0$ for $x_k \notin T$, $V_\mathcal{P}$ is piecewise constant and equals zero only on T. For $x_k \in P \subseteq T$, the construction of $u_\mathcal{P}$ implies

$$x_{k+1} \in N$$

for some $N \in \underline{\mathcal{N}}(P)$ implying

$$V_\mathcal{P}(x_{k+1}) \leq \nu.$$

Thus (25) implies $V_\mathcal{P}(x_{k+1}) \leq \nu$ for all $k > k^*$, i.e., $x_k \in D$. ∎

Remark 1: If system (1) is asymptotically controllable to the origin and V is continuous, then we can use the same arguments as in [9] in order to show that on increasingly finer partitions $\mathcal{P}_l$ and for targets T_l shrinking down to $\{0\}$ we obtain $V_{\mathcal{P}_l} \to V$. Furthermore, the function V is a control Lyapunov function for the system. Using similar arguments as in [7], one can then show that the corresponding sets D_l shrink down to $\{0\}$, too, and consequently the family of feedback laws $u_{\mathcal{P}_l}$ renders the closed loop system semiglobally practically asymptotically stable in the sense of, e.g., [13], with respect to the accuracy parameter l.

Note, however, that all our results hold without imposing controllability assumptions on (1) or invariance assumptions on the target set T and that the set D in which the trajectories of the system will end up can be determined a posteriori from the numerically computed $V_\mathcal{P}$.

IV. AN EXAMPLE

In order to illustrate the benefit of the proposed new discretization method, we consider the classical inverted pendulum on a cart as already considered for the old method in [11], [7]. The system is given by the continuous time control system

$$\left(\frac{4}{3} - m_r \cos^2 \varphi\right) \ddot{\varphi} + \frac{1}{2} m_r \dot{\varphi}^2 \sin 2\varphi - \frac{g}{\ell} \sin \varphi$$
$$= -u \frac{m_r}{m\ell} \cos \varphi, \tag{26}$$

where we have used the parameters $m = 2$ for the pendulum mass, $m_r = m/(m + M)$ for the mass ratio with cart mass $M = 8$, $\ell = 0.5$ as the length of the pendulum and $g = 9.8$ for the gravitational constant. The instantaneous cost is

$$q(\varphi, \dot{\varphi}, u) = \frac{1}{2} \left(0.1\varphi^2 + 0.05\dot{\varphi}^2 + 0.01u^2 \right). \quad (27)$$

Denoting the evolution operator of the control system (26) for constant control functions u by $\Phi^t(x, u)$, we consider the discrete time system (1) with $f(x, u) = \Phi^T(x, u)$ for $T = 0.1$, i.e., the sampled continuous time system with sampling rate $T = 0.1$. The map Φ^T is approximated via the classical Runge-Kutta scheme of order 4 with step size 0.02. The discrete time cost function is obtained by numerically integrating the continuous time instantaneous cost according to

$$g(\varphi, \dot{\varphi}, u) = \int_0^T q(\Phi^t((\varphi, \dot{\varphi}), u), u) \, dt,$$

We choose $X = [-8, 8] \times [-10, 10]$ as the region of interest.

In [7], a feedback trajectory with initial value $(3.1, 0.1)$ was computed that was based on the directly discretized approximate optimal value function $V_{\mathcal{P}}$ from Section II-B on a partition of 2^{18} boxes (cf. Figure 1 (top)). In contrast to what one might expect, the approximate optimal value function does actually not decrease monotonically along this trajectory (cf. Figure 1 (bottom)). This effect is due to the fact that the approximate optimal value function $V_{\mathcal{P}}$ from Section II-B is only an approximation of the exact optimal value function but does not satisfy the Lyapunov inequality (6). In particular, on a coarser partition of 2^{14} boxes, the associated feedback is not stabilizing this initial condition any more.

In contrast to this, our new approach yields an approximation $V_{\mathcal{P}}$ which does satisfy the Lyapunov inequality (6) outside T and hence an approximation to V which is a Lyapunov function itself. Figure 2 shows the approximate upper value function on a partition of 2^{14} boxes with target set $T = [-0.1, 0.1]^2$ as well as the trajectory generated by the associated feedback for the initial value $(3.1, 0.1)$. As expected, the approximate value function is decreasing monotonically along this trajectory. Furthermore, despite the fact that we used considerably fewer boxes as for Figure 1, the resulting trajectory is obviously closer to the optimal one because it converges to the origin much faster.

Note that the behavior predicted by Theorem 2 is clearly visible in Fig. 2(bottom): $V_{\mathcal{P}}(x_k)$ is strictly decreasing until $k^* = 57$ while for $k > k^*$ the trajectory x_k occasionally leaves T but remains inside D, where $\nu \approx 0.25$ here.

It should be remarked that in order to construct the graph map $\mathcal{F}$ from (19) a numerical approximation of the set valued images $f(P, u)$ needs to be done, which is a serious problem at its own right. In this example an approximation of this image using a certain number of test points per partition element P yields sufficient accuracy.

V. Conclusions and outlook

In this paper we have presented an approximative solution technique for infinite horizon optimal control problems. Our
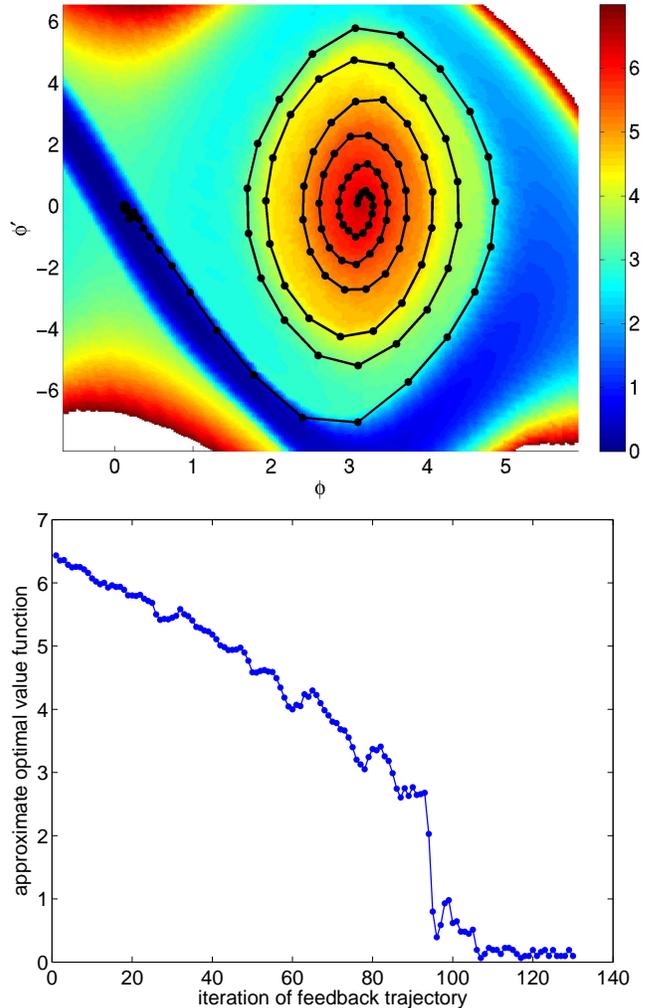

Fig. 1. Approximate optimal value function and feedback trajectory (top) and the approximate optimal value function along the feedback trajectory (bottom) for the inverted pendulum on a 2^{18} box partition using the direct discretization.

approach relies on a partitioning of the state space and yields an approximation to the optimal value function which preserves the Lyapunov function property of the original value function and can be computed by an efficient graph theoretic algorithm. The resulting stabilizing feedback law is constant on each partition element. This property allows to interpret the partitioning as a quantization of the systems state space, because in order to evaluate the feedback law only the partition element containing the current state must be known. In particular, if a given state space quantization allows for the construction of a stabilizing feedback law, then the algorithm will find it.

The interpretation of the partitioning as a quantization leads to several extensions of our method. For instance, the method is easily adapted to discrete event systems in which only the event of a transition from one partition element to another is communicated to the controller. Another extension is the design of more complex hierarchically organized controllers, e.g., our approach can be used in order to optimize
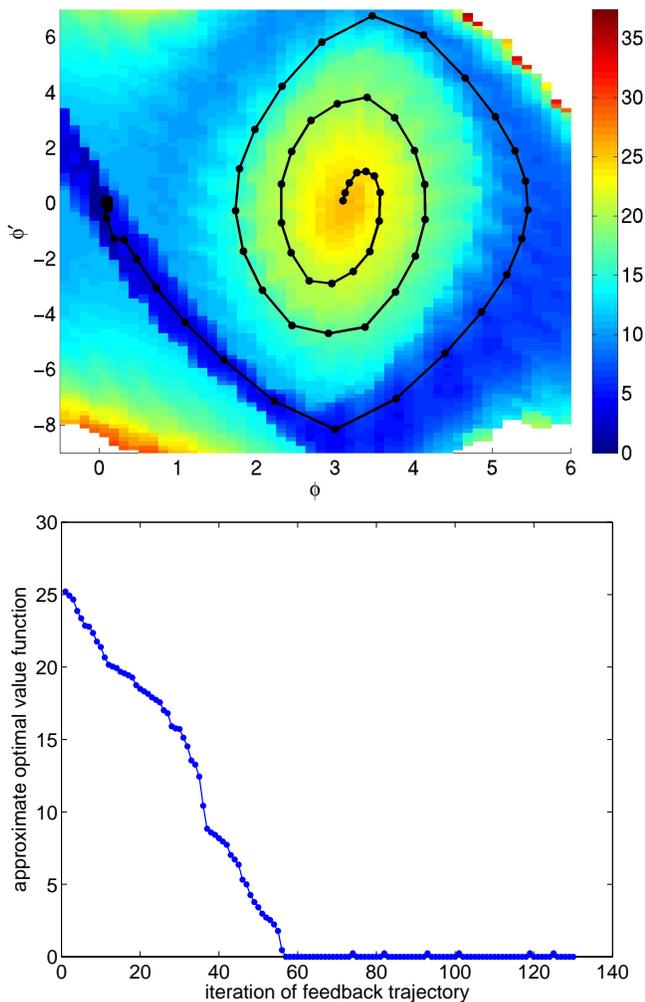
Fig. 2. Approximate upper value function and feedback trajectory (top) and the approximate upper value function along the feedback trajectory (bottom) for the inverted pendulum on a 2^{14} box partition using the dynamic game feedback construction.

the coefficients of a piecewise affine (with respect to the partition) controller. Finally, since the resulting graph can be interpreted as a finite automaton, ideas from supervisory control can be included into our scheme, like including information on the trajectory's past in order to gather more information on coarse partitions, as formalized in [12]. These extensions are topics of current research.

REFERENCES

[1] A. BEMPORAD AND C. FILIPPI, *Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming*, J. Optim. Theory Appl., 117 (2003), pp. 9–38.
[2] D. P. BERTSEKAS, *Dynamic Programming and Optimal Control. Vol. 1 and 2.*, Athena Scientific, Belmont, MA, 1995.
[3] J. BJÖRNBERG AND M. DIEHL, *Approximate robust dynamic programming and robustly stable MPC*, Automatica, 42 (2006), pp. 777–782.
[4] P. E. CAINES AND Y. J. WEI, *The hierarchical lattices of a finite machine*, Systems Control Lett., 25 (1995), pp. 257–263.
[5] P. E. CAINES AND Y.-J. WEI, *On dynamically consistent hybrid systems*, in Hybrid systems, II (Ithaca, NY, 1994), vol. 999 of Lecture Notes in Comput. Sci., Springer, Berlin, 1995, pp. 86–105.
[6] A. GRANCHAROVA AND T. A. JOHANSEN, *Survey of explicit approaches to constrained optimal control*, in Switching and learning in feedback systems. European summer school on multi-agent control, Maynooth, Ireland, September 8–10, 2003, R. Murray-Smith and R. Shorten, eds., vol. 3355 of LNCIS, Springer, Heidelberg, 2005, pp. 47–97.
[7] L. GRÜNE AND O. JUNGE, *A set oriented approach to optimal feedback stabilization*, Syst. Control Lett., 54 (2005), pp. 169–180.
[8] L. GRÜNE AND O. JUNGE, *Optimal stabilization of hybrid systems using a set oriented approach*, in Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems MTNS2006, Kyoto, Japan, 2006, pp. 2089–2093.
[9] L. GRÜNE AND O. JUNGE, *Global optimal control of perturbed systems*, J. Optim. Theory Appl., (2007). To appear.
[10] T. A. JOHANSEN, *Approximate explicit receding horizon control of constrained nonlinear systems*, Automatica, 40 (2004), pp. 293–300.
[11] O. JUNGE AND H. M. OSINGA, *A set oriented approach to global optimal control*, ESAIM Control Optim. Calc. Var., 10 (2004), pp. 259–270.
[12] T. MOOR, J. RAISCH, AND S. O'YOUNG, *Discrete supervisory control of hybrid systems based on l-complete approximations*, Discrete Event Dyn. Syst., 12 (2002), pp. 83–107. WODES '98 (Cagliari).
[13] D. NEŠIĆ AND A. R. TEEL, *A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models*, IEEE Trans. Automat. Control, 49 (2004), pp. 1103–1122.
[14] P. TØNDEL, T. A. JOHANSEN, AND A. BEMPORAD, *An algorithm for multi-parametric quadratic programming and explicit MPC solutions*, Automatica, 39 (2003), pp. 489–497.
[15] M. VON LOSSOW, *Mengenorientierte optimale Steuerung und Fehlerschätzung*. Diploma Thesis, Universität Bayreuth, 2005.
[16] M. VON LOSSOW, *A min-max version of Dijkstra's algorithm with application to perturbed optimal control problems*, in Proceedings of the GAMM Annual meeting, Zürich, Switzerland, 2007. To appear.