

Set Oriented Construction of Globally Optimal Controllers

Mengenorientierte Konstruktion global optimaler Regler

Lars Grüne, Universität Bayreuth,
Oliver Junge, Technische Universität München

Summary Recently, techniques for the design of optimal controllers have been developed which are based on a piecewise constant approximation of the value function of the underlying optimal control problem. They combine ideas from set oriented numerics with shortest path algorithms from graph theory. The approach is particularly well suited for problems with highly irregular value function, complicated state constraints and naturally handles hybrid systems. In this contribution, we give an overview of the approach and illustrate it by several numerical examples. ▶▶▶ **Zusammenfassung** In den letzten Jahren wurden Techniken zur Konstruktion opti-

maler Regler entwickelt, die auf einer stückweise konstanten Approximation der Wertefunktion des zugrunde liegenden Optimalsteuerungsproblems beruhen. Sie kombinieren Ideen der mengenorientierten Numerik mit Algorithmen zur Berechnung kürzester Wege aus der Graphentheorie. Der Ansatz ist insbesondere gut geeignet für Probleme mit hochgradig irregulärer Wertefunktion, komplexen Zustandsbeschränkungen und ermöglicht auf natürliche Weise die Behandlung von hybriden Systemen. In diesem Beitrag geben wir einen Überblick dieses Ansatzes und illustrieren ihn anhand einiger numerischer Beispiele.

Keywords Stabilizing controller, dynamic programming, set oriented numerics, shortest path algorithm, dynamic game, hybrid system ▶▶▶ **Schlagwörter** Stabilisierender Regler, dynamische Programmierung, mengenorientierte Numerik, Kürzeste-Wege-Algorithmus, dynamisches Spiel, hybrides System

1 Introduction

An elegant way to construct a globally stabilizing controller for a (nonlinear) control system is via Bellman's *optimality principle*. This is a fixed point equation which – together with a suitable boundary condition – characterizes the *value function* of the system.¹ From the value function, the associated optimally stabilizing controller can be computed by repeatedly solving a finite-dimensional optimization problem. The value function will then act as a Lyapunov function for the closed loop system.

In the case of linear dynamics and a quadratic (instantaneous) cost function, an explicit formula for V

as well as the controller can be derived by solving an associated *Riccati equation*. In the nonlinear case, one typically has to resort to a numerical approximation of V . In this case, one needs to solve the Bellman equation within a finite-dimensional approximation space. Typically, piecewise (multi)linear functions are employed for the approximation [1; 3]. Recently, however, piecewise constant functions have been proposed for this purpose [4–7; 11]. This approach leads to an (“all source, single destination”) shortest path problem on a finite graph and can be solved by corresponding fast algorithms from algorithmic graph theory (like Dijkstra's algorithm).

The approach is particularly well suited for problems with highly irregular value function, complicated state constraints and naturally handles hybrid systems. Its extension to perturbed systems conceptually enables the treatment of discrete event systems.

¹ The value function $V(x)$ of a state x of the system is the minimal cost accumulated along any controlled trajectory starting at x . The associated instantaneous cost may be given in the problem description or may have to be suitably defined.

2 Setting

Problem Formulation. Our goal is to globally and optimally stabilize a given subset T of the state space X of the perturbed discrete-time control system

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

$x_k \in X$, 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 $u(x_k)$ such that T is an asymptotically stable set for the resulting closed loop system

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

for any sequence (w_k) of perturbations and such that the accumulated cost $\sum_{k=0}^{\infty} g(x_k, u(x_k))$ along a feedback trajectory is minimal.

System Classes, Extensions. The class of systems which can be modelled by (1) is rather large, it includes

- sampled-data continuous time systems: $X \subset \mathbb{R}^n$ compact, f is the time- T -map of the control flow and $g(x_k, u_k)$ typically contains integral expressions along the continuous time solution over a sampling interval;
- certain types of hybrid systems: $X = Y \times D$, where $Y \subset \mathbb{R}^n$ compact and D is finite, cf. Sect. 6 for details;
- discrete event systems: here f is defined as a (generalized) Poincaré map.

The controls u_k and perturbations w_k have to be chosen from compact metric spaces U and W , respectively (which may in particular be discrete).

3 Basic Construction

The construction of the feedback controller will be based on the optimality principle. To this end we compute an approximation to the optimal value function of the problem which is piecewise constant on the elements of a partition of X . This type of discretization leads to a *discrete optimality principle*, i. e. a shortest path problem on a directed graph. In order to stress the basic ideas in this section we consider problems without perturbations,

$$x_{k+1} = f(x_k, u_k), \quad k = 0, 1, \dots, \quad (3)$$

and the simple target set $T = \{0\}$.

3.1 The Optimality Principle

We assume the map $f : X \times U \rightarrow \mathbb{R}^d$ 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 initial state $x \in X$ and a given control sequence $\mathbf{u} = (u_0, u_1, \dots) \in U^{\mathbb{N}}$ there is a unique associated trajectory $(x_k(x, \mathbf{u}))_{k \in \mathbb{N}}$ of (3). Let $\mathcal{U}(x) = \{\mathbf{u} \in U^{\mathbb{N}} : x_k(x, \mathbf{u}) \rightarrow 0 \text{ as } k \rightarrow \infty\}$ 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

$$J(x, \mathbf{u}) = \sum_{k=0}^{\infty} g(x_k(x, \mathbf{u}), u_k) \in [0, \infty]. \quad (4)$$

The (optimal) value function $V(x) = \inf_{\mathbf{u} \in \mathcal{U}(x)} J(x, \mathbf{u})$ satisfies the optimality principle

$$V(x) = \inf_{u \in U} \{g(x, u) + V(f(x, u))\} \quad (5)$$

in all states x for which $V(x)$ is finite. The operator

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

acting on real valued functions on X , is called the *dynamic programming operator* and V is the unique fixed point of the equation $v = L[v]$ satisfying the boundary condition $V(0) = 0$.

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

$$u(x) = \operatorname{argmin}_{u \in U} \{g(x, u) + V(f(x, u))\}, \quad (6)$$

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

$$V(x) \geq g(x, u(x)) + V(f(x, u(x))) \quad (7)$$

the optimal value function is a Lyapunov function of the closed loop system, provided V is finite, positive definite and proper.²

3.2 Discretization

We are going to approximate V by functions which are piecewise constant. This approach is motivated on the one hand by the fact that the resulting discrete problem can be solved by fast shortest path algorithms and on the other hand by the observation that – via a suitable generalization to perturbed systems, cf. Sect. 4 – the resulting feedback is piecewise constant, too, and can be computed offline.

Let \mathcal{P} be a partition of the state space X , i. e. a collection of pairwise disjoint subsets of X whose union covers X . For a state $x \in X$ we let $\rho(x) \in \mathcal{P}$ denote the partition element which contains x . In order to simplify notation, in the sequel we identify any subset $\{P_1, \dots, P_k\} \subseteq \mathcal{P}$ with the corresponding subset $\cup_{i=1, \dots, k} P_i \subseteq X$.

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

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

² These properties of V can be ensured by suitable asymptotic controllability properties and bounds on g .

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

$$L_{\mathcal{P}} := \psi \circ L.$$

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*.

The Discrete Optimality Principle. 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 [7] that the fixed point equation $V_{\mathcal{P}} = L_{\mathcal{P}}[V_{\mathcal{P}}]$ is equivalent to the *discrete optimality principle*

$$V_{\mathcal{P}}(P) = \min_{P' \in \mathcal{F}(P)} \{ \mathcal{G}(P, P') + V_{\mathcal{P}}(P') \}, \quad (8)$$

where the map \mathcal{F} is given by

$$\begin{aligned} \mathcal{F}(P) &= \rho(f(P, U)) \\ &= \{ P' \in \mathcal{P} : P' \cap f(P, U) \neq \emptyset \} \end{aligned} \quad (9)$$

and the cost function \mathcal{G} by

$$\mathcal{G}(P, P') = \inf_{u \in U} \{ g(x, u) \mid x \in P, f(x, u) \in P' \}. \quad (10)$$

Note that the value $V_{\mathcal{P}}(P)$ 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 $\mathcal{G}(P, P')$, cf. Fig. 1. As such, it can be computed by, e.g., Dijkstra's algorithm, cf. [11].

Properties of $V_{\mathcal{P}}$. From the definition of the projection ψ and by an induction argument on the elements of the partition one can conclude that

$$V_{\mathcal{P}}(x) \leq V(x) \quad (11)$$

for any $x \in S$ and every partition \mathcal{P} , i.e. the approximate value function $V_{\mathcal{P}}$ yields a *lower bound* on the true value function. As a result, the residual

$$e_{\mathcal{P}} = L_{\mathcal{P}}[V_{\mathcal{P}}] - V_{\mathcal{P}}$$

is an efficient pointwise a posteriori estimator for the error $e = V - V_{\mathcal{P}}$, i.e. one has $e_{\mathcal{P}} \leq e$ pointwise on S [4].

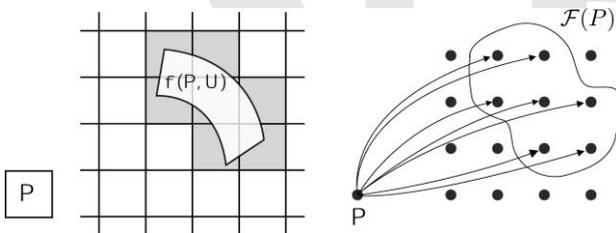


Figure 1 Partition of phase space, image of an element (left) and corresponding edges in the induced graph (right).

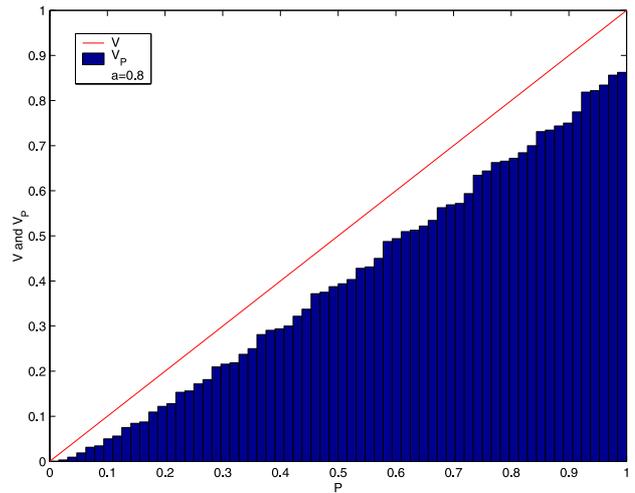


Figure 2 True and approximate optimal value function for the simple example, 64 subintervals, $a = 0.8$.

3.3 Example: A Simple 1D System

Consider the system

$$x_{k+1} = x_k + (1-a)u_k x_k, \quad k = 0, 1, \dots, \quad (12)$$

where $x_k \in X = [0, 1]$, $u_k \in U = [-1, 1]$ and $a \in (0, 1)$ is a fixed parameter. Let

$$g(x, u) = (1-a)x,$$

such that the optimal control policy is to steer to the origin as fast as possible, i.e. for every x , the optimal sequence of controls is $(-1, -1, \dots)$. This yields $V(x) = x$ as the optimal value function.

For the following computations we consider $a = 0.8$ and use a partition of 64 equally sized subintervals of $[0, 1]$. The weights (10) are approximated by minimizing over 100 equally spaced test points in each subinterval and 10 equally spaced points in U .

Figure 2 shows V and the approximate optimal value function $V_{\mathcal{P}}$.

4 Construction Including Perturbations

In many cases, the discretization described above is not very efficient in the sense that rather fine partitions are required in order to obtain a stabilizing controller. A more efficient discretization results by employing ideas from dynamic game theory. In this section we briefly sketch how our approach can be extended to dynamic games, for details we refer to [7], and in the following Sect. 5 we show how this extension can be used in order to obtain stabilizing controllers on coarser partitions.

4.1 Dynamic Games

A *dynamic game* is a map $F : X \times U \times W \rightarrow X$, where $X \subset \mathbb{R}^d$, $U \subset \mathbb{R}^m$, $W \subset \mathbb{R}^{\ell}$ are compact sets, together with a cost function $G : X \times U \times W \rightarrow [0, \infty)$. 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 associated 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 $T \subset X$, the total cost accumulated along a trajectory is

$$J(x, \mathbf{u}, \mathbf{w}) = \sum_{k=0}^{k^*} G(x_k(x, \mathbf{u}, \mathbf{w}), u_k, w_k),$$

with $k^* = k^*(T, x, \mathbf{u}, \mathbf{w}) := \inf\{k \geq 0 \mid x_k(x, \mathbf{u}, \mathbf{w}) \in T\}$. Here we are interested in what is called the *upper value function* of the game, which is defined by

$$V(x) = \sup_{\beta \in \mathcal{B}} \inf_{\mathbf{u} \in U^{\mathbb{N}}} J(x, \mathbf{u}, \beta(\mathbf{u})). \quad (13)$$

Here, \mathcal{B} denotes the set of all nonanticipating strategies³ $\beta: U^{\mathbb{N}} \rightarrow W^{\mathbb{N}}$. By standard dynamic programming arguments [2] one sees that this function is the unique solution to the optimality principle

$$V(x) = \inf_{u \in U} \sup_{w \in W} \{G(x, u, w) + V(F(x, u, w))\} \quad (14)$$

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

$$V(x) = \inf_{u \in U} \left\{ G(x, u) + \sup_{x' \in F(x, u, W)} V(x') \right\}. \quad (15)$$

Note that in this equation 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.

4.2 Discretization of the Dynamic Game

We employ the same approach as in Sect. 3.2 in order to discretize (14). Note that the setting in Sect. 3.2 can be seen as a special case of the more general situation here using $W = \{0\}$ and $F(x, u, w) = f(x, u)$. In our perturbed

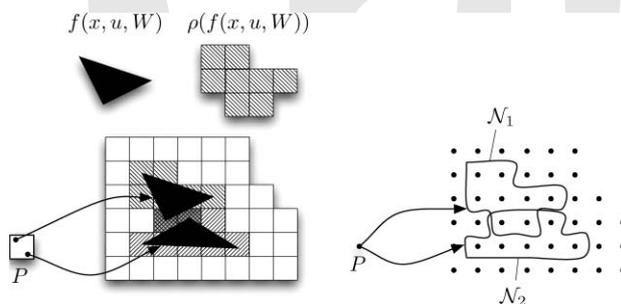


Figure 3 Partition of phase space, images of an element (left) and corresponding edges in the induced hypergraph (right).

³ A strategy β is *nonanticipating* if from $u_k = u'_k$ for all $k \leq K$ it follows that $\beta(\mathbf{u})_k = \beta(\mathbf{u}')_k$ for all $k \leq K$ for any two control sequences $\mathbf{u} = (u_k)_k, \mathbf{u}' = (u'_k)_k \in U^{\mathbb{N}}$.

setup, one can show [7] that the corresponding discrete optimality principle is given by

$$V_{\mathcal{P}}(P) = \inf_{\mathcal{N} \in \mathcal{F}(P)} \left\{ \mathcal{G}(P, \mathcal{N}) + \sup_{P' \in \mathcal{N}} V_{\mathcal{P}}(P') \right\} \quad (16)$$

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

$$\mathcal{F}(P) = \{\rho(F(x, u, W)) : (x, u) \in P \times U\} \quad (17)$$

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

Note the difference of $\mathcal{F}(P)$ compared to (9): while in (9) $\mathcal{F}(P)$ was a subset of \mathcal{P} , in (17) $\mathcal{F}(P)$ now is a set of subsets $\mathcal{N} \subset \mathcal{P}$. 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 $E = \{(P, \mathcal{N}) : \mathcal{N} \in \mathcal{F}(P)\}$, cf. Fig. 3.

4.3 A Min-Max Shortest Path Algorithm

It was shown in [7] 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 [13] for details.

For $\mathcal{T} := \{P \in \mathcal{P} : P \cap T \neq \emptyset\}$, 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 $\mathcal{Q} := \mathcal{Q} \setminus \{P\}$
- 7 for each $(Q, \mathcal{N}) \in E$ with $P \in \mathcal{N}$
- 8 if $\mathcal{N} \subset \mathcal{P} \setminus \mathcal{Q}$ then
- 9 if $V(Q) > \mathcal{G}(Q, \mathcal{N}) + V(P)$ then
- 10 $V(Q) := \mathcal{G}(Q, \mathcal{N}) + V(P)$

4.4 Convergence

It is natural to ask whether the approximate value function converges to the true one when the element diameter of the underlying partition goes to zero. This has been proven pointwise on the stabilizable set S in the unperturbed case [11], as well as in an L^1 -sense on S and an L^∞ sense on the domain of continuity in the perturbed case, assuming continuity of V on the boundary of the target set T [7].

5 The Discretization as a Perturbation

It was shown in [4] that the feedback (6) using the approximate value function $V_{\mathcal{P}}$ from (8) will practically stabilize the system under suitable conditions. Numerical experiments (cf. Sect. 5.5), 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 value function does not decay monotonically along system trajectories. The reason is that whenever the trajectory of the closed loop system enters a certain element, 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

$$V_{\mathcal{P}}(x) \leq V(x) \quad (18)$$

and

$$\begin{aligned} V_{\mathcal{P}}(x) &\leq \min_{u \in U} \{g(x, u) + V_{\mathcal{P}}(f(x, u))\} \\ &= g(x, u_{\mathcal{P}}(x)) + V_{\mathcal{P}}(f(x, u_{\mathcal{P}}(x))). \end{aligned} \quad (19)$$

A comparison with the Lyapunov function property (7) reveals that inequality (19) delivers exactly the *opposite* than what is needed in order to prove asymptotic stability.

5.1 Construction of the Dynamic Game

In order to cope with this phenomenon 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: 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 $f(\rho(x), u(x))$.

Let us be more precise: given a control system (3) and an associated cost function g , we consider the dynamic game

$$F(x, u, W) := f(\rho(x), u) \quad (20)$$

for $x \in X, u \in U$ and define

$$G(x, u) = \sup_{x' \in \rho(x)} g(x', u). \quad (21)$$

5.2 Properties of the Approximate Value Function

The following theorem shows the crucial properties of the associated approximate value function $V_{\mathcal{P}}$, in particular, that this function satisfies the opposite inequalities compared to (18), (19), when the terminal set T is appropriately included in the formulation.

Theorem 1 ([6]). *Let V denote the optimal value function of the optimal control problem described in Sect. 3.1 and let $V_{\mathcal{P}}$ denote the approximate optimal value function of the game F, G on a given partition \mathcal{P} with target set $T \subset \mathcal{P}$ with $0 \in T$. Then,*

$$V(x) - \max_{y \in T} V(y) \leq V_{\mathcal{P}}(x), \quad (22)$$

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

$$V_{\mathcal{P}}(x) \geq \min_{u \in U} \{g(x, u) + V_{\mathcal{P}}(f(x, u))\} \quad (23)$$

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

5.3 The Feedback is the Shortest Path

Define

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

Due to the construction of the game F, G , this feedback is actually constant on each partition element. Moreover, we can directly extract $u_{\mathcal{P}}$ from the min–max Dijkstra Algorithm 1, for details we refer to [6].

Once $u_{\mathcal{P}}$ has been computed for every partition element, the only online computation that remains to do is done is the determination of the partition element for each state on the feedback trajectory. In a sense, the fast *online* evaluation of the feedback law has been made possible by a comparatively large *offline* computation, namely the construction of the (hyper)graph. Note that, in comparison, the time for running the shortest path algorithms is negligible (on the order of a few seconds to minutes). In the examples in the following sections we summarize data on the size of the graph and the required computing time.

5.4 Behavior of the Closed Loop System

Finally, we can now state the properties of the feedback law $u_{\mathcal{P}}$ constructed in the last section.

Theorem 2 ([6]). *Under the assumptions of Theorem 1, if $(x_k)_k$ denotes the trajectory of the closed loop system with feedback $u_{\mathcal{P}}$ and if $V_{\mathcal{P}}(x_0) < \infty$, then there exists $k^* \in \mathbb{N}$ such that for $k = 0, \dots, k^* - 1$*

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

and $x_{k^} \in T$.*

If the system (3) is asymptotically controllable to the origin and V is continuous, then we can use the same arguments as in [7] in order to show that on increasingly finer partitions \mathcal{P}_ℓ and for targets T_ℓ shrinking down to $\{0\}$ we obtain $V_{\mathcal{P}_\ell} \rightarrow V$.

5.5 Example: An Inverted Pendulum

The Model. In order to illustrate the benefit of the approach proposed in this section, we consider the classical inverted pendulum on a cart, cf. [4; 11]. Let $\varphi \in [0, 2\pi]$ denote the angle between the pendulum and the upright vertical. Ignoring the dynamics of the cart, a model of the system is given by

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

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. As the cost function, we choose

$$q(\varphi, \dot{\varphi}, u) = \frac{1}{2} (0.1\varphi^2 + 0.05\dot{\varphi}^2 + 0.01u^2). \quad (24)$$

Denoting the system's evolution operator for constant control functions u by $\Phi^t(x, u)$, we consider the discrete time system (3) with $f(x, u) = \Phi^t(x, u)$ for $t = 0.1$, i.e., the sampled continuous time system with sampling period $t = 0.1$. The discrete time cost function is obtained by numerically integrating the continuous time instantaneous cost according to $g(\varphi, \dot{\varphi}, u) = \int_0^t q(\Phi^\tau((\varphi, \dot{\varphi}), u), u) d\tau$.

Implementation. We use the classical Runge-Kutta scheme of order 4 with step size 0.02 in order to approximate Φ^t . We choose $X = [-8, 8] \times [-10, 10]$ as the region of interest, $U = [-64, 64]$ and employ a box partition consisting of identically sized boxes for the discretization. In constructing the (hyper)graph (9) resp. (17) we approximate the graph map \mathcal{F} and the edge weights by integrating 9 equally spaced points on each partition box, choosing from 16 equally spaced values in U and minimizing over the associated values of g .

Results. Figure 4 shows the approximate optimal value function as well as the feedback trajectory for the initial value (3.1, 0.1) computed by the approach from Sect. 3.2 on a partition of 2^{18} boxes. The associated graph has $\sim 14 \times 10^6$ edges, the computation of the graph takes roughly 6 minutes on a 3 GHz CPU, while the determination of the approximate value function via Dijkstra's algorithm runs in less than half a second. As Fig. 5 clearly shows, the approximate optimal value function does not decrease monotonically along this trajectory. This is due to the fact that this $V_{\mathcal{P}}$ does not satisfy the Lyapunov inequality (7). 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, the approach from this section yields an approximation $V_{\mathcal{P}}$ which does satisfy the Lyapunov inequality (7) outside T and hence an approximation to V which is a Lyapunov function itself. Figure 6 shows the approximate upper value function on

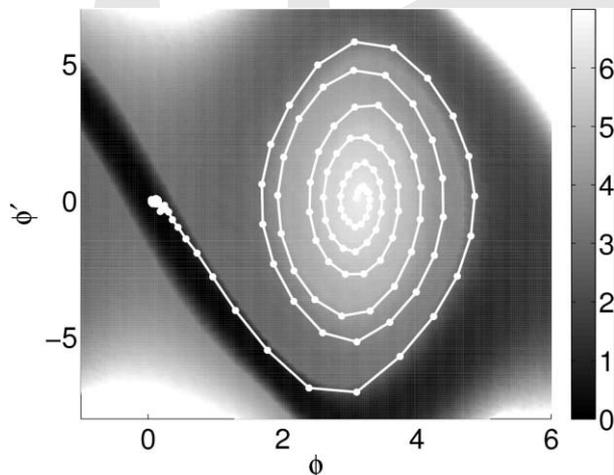


Figure 4 Approximate optimal value function and feedback trajectory for the inverted pendulum using the approach from Sect. 3.2.

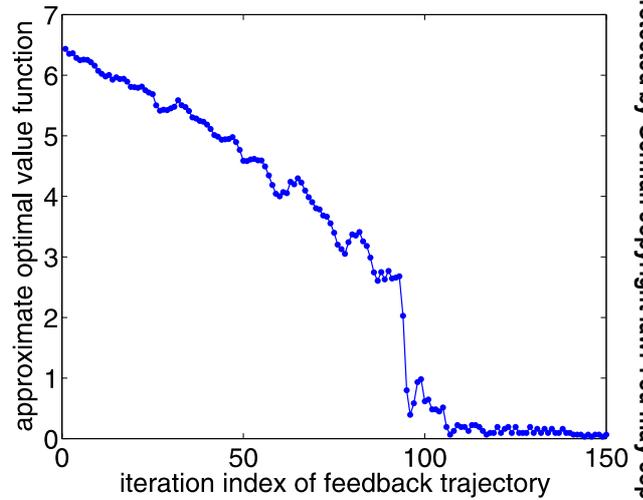


Figure 5 Approximate optimal value function along the feedback trajectory for the inverted pendulum using the approach from Sect. 3.2.

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). The associated hypergraph has $\sim 10^5$ hyperedges, its computation takes roughly 1 minute on a 3 GHz CPU, the min-max-Dijkstra algorithm again runs in less than half a second. As expected, the approximate value function is decreasing monotonically along this trajectory. Furthermore, despite the fact that we used considerably fewer boxes as for Fig. 4, the resulting trajectory is obviously closer to the optimal one because it converges to the origin much faster.

It should be remarked that in order to construct the graph map \mathcal{F} a numerical evaluation of the set valued images $f(P, u)$ needs to be done, which is a serious problem at its own right. Typically (as mentioned above) the set $f(P, u)$ is being approximated by mapping a certain number of sample points in P . Certainly, this introduces another source of error. However, in our example computations this error turned out to be negligible. Alternatively, in the case that one is interested in a rigorous

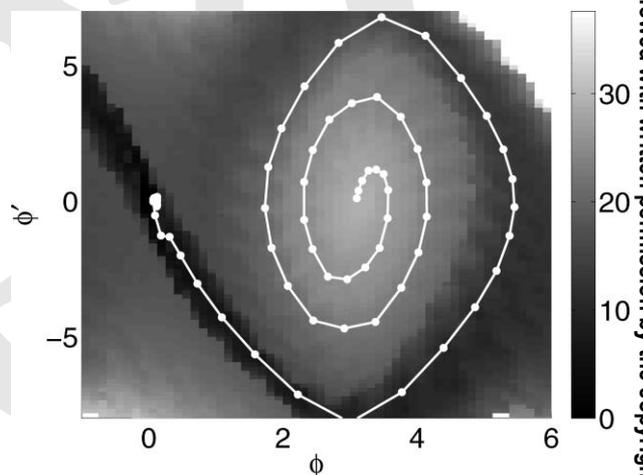


Figure 6 Approximate upper value function and feedback trajectory for the inverted pendulum using the dynamic game feedback construction.

article is protected by German copyright law. You may copy and distribute this article for your personal use only. Other use is only allowed with written permission by the copyright holder.

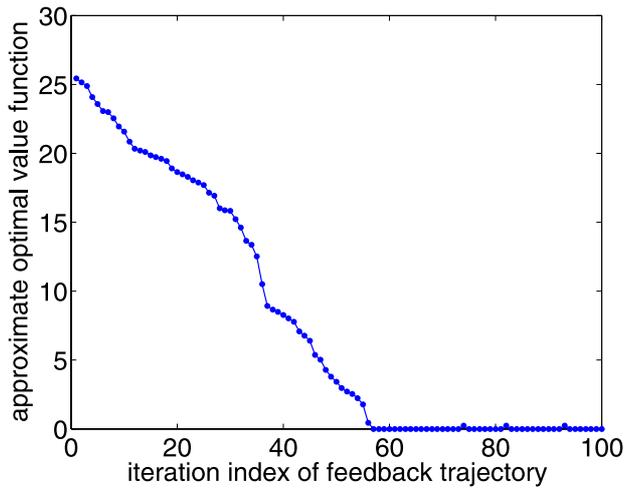


Figure 7 Approximate upper value function along the feedback trajectory for the inverted pendulum using the dynamic game feedback construction.

computation, either techniques based on Lipschitz estimates for f [10] or interval arithmetic can be employed.

6 Hybrid Systems

The discretization approach described in Sects. 3.2 and 4.2 is naturally suited for the treatment of *hybrid systems*. In order to exemplify this, we consider the problem of optimally stabilizing the continuous state component x of a discrete-time nonlinear hybrid control system given by

$$\begin{aligned} x_{k+1} &= f_c(x_k, y_k, u_k) \\ y_{k+1} &= f_d(x_k, y_k, u_k) \end{aligned} \quad k = 0, 1, \dots, \quad (25)$$

with continuous state dynamics $f_c: X \times Y \times U \rightarrow X \subset \mathbb{R}^n$ and discrete state dynamics $f_d: X \times Y \times U \rightarrow Y$. Here the set U of possible control inputs is finite⁴, the set $X \subset \mathbb{R}^n$ of continuous states is compact and the set Y of discrete states (or modes) is an arbitrary finite set. The solutions of (25) for initial values $x_0 = x$, $y_0 = y$ and control sequence $\mathbf{u} = (u_0, u_1, \dots) \in U^{\mathbb{N}}$ are denoted by $x_k(x, y, \mathbf{u})$ and $y_k(x, y, \mathbf{u})$, respectively, and we assume that for each $k \geq 0$ the map $x_k(\cdot, y, \mathbf{u})$ is continuous for each $y \in Y$ and each $\mathbf{u} \in U^{\mathbb{N}}$. Note that if f_d does not depend on x , then this is equivalent to $f_c(\cdot, y, u): X \times Y \rightarrow \mathbb{R}^d$ being continuous for each $y \in Y$, $u \in U$.

Problem Formulation. Similar to the previous sections, given a target set $T \subset X$, the goal of the optimization problem we want to solve is to find a control sequence $u_k, k = 0, 1, 2, \dots$, such that $x_k \rightarrow T$ as $k \rightarrow \infty$, while minimizing the accumulated cost $g: X \times Y \times U \rightarrow [0, \infty)$ with $g(x, y, u) > 0$ for all $x \notin T$, $y \in Y$ and $u \in U$.

To this end, we would like to construct an approximate *optimal feedback* $u: S \rightarrow U$ such a suitable approximate asymptotic stability property for the resulting closed loop system holds. Again, the construction will be based on an

⁴ If desired, continuous control values could also be included and treated with the discretization technique described in [4; 11].

approximation of the (optimal) value function which will act as a Lyapunov function. For an appropriate choice of g this function is continuous in x at least in a neighborhood of T [9].

6.1 Computational Approach

Let \mathcal{Q} be a *partition* of the continuous state set X , that is a finite collection of compact subsets $Q_i \subset X$, $i = 1, \dots, r$, with $\cup_{i=1}^r Q_i = X$, and $m(Q_i \cap Q_j) = 0$ for $i \neq j$ (where m denotes Lebesgue measure). Then the sets

$$\mathcal{P} := \{Q_i \times \{y\} \mid Q_i \in \mathcal{Q}, y \in Y\} \quad (26)$$

form a partition of the product state space $Z = X \times Y$. On \mathcal{P} the approaches from Sects. 3.2 and 4.2 can be applied literally.

6.2 Example: A Switched Voltage Controller

We reconsider an example from [12]: A switched power controller for DC to DC conversion. Within the controller, a semiconductor device is switching the polarity of a voltage source in order to keep the load voltage as constant as possible. The model is

$$\begin{aligned} \dot{x}_1 &= \frac{1}{C} (x_2 - I_{load}) \\ \dot{x}_2 &= -\frac{1}{L} x_1 - \frac{R}{L} x_2 + \frac{1}{L} u V_{in} \\ \dot{x}_3 &= V_{ref} - x_1, \end{aligned} \quad (27)$$

where $u \in \{-1, 1\}$ is the control input. In the following numerical experiment we use the same parameter values as given in [12].

The corresponding discrete time system is given by the time- h -map ϕ^h ($h = 0.1$ in our case) of (27), with the control input held constant during this sampling period. The cost function is

$$g(x, u) = q_p(\phi_1^h(x) - V_{ref}) + q_D(\phi_2^h(x) - I_{load}) + q_I\phi_3^h(x).$$

The third component in (27) is only being used in order to penalize a large L^1 -error of the output voltage. We slightly simplify the problem (over its original formulation in [12]) by using $x_3 = 0$ as initial value in each evaluation of the discrete map. Correspondingly, the map reduces to a two-dimensional one.

Confining our domain of interest to the rectangle $X = [0, 1] \times [-1, 1]$, our target set is given by $T = \{V_{ref}\} \times [-1, 1]$. For the construction of the finite graph, we employ a partition of X into 64×64 equally sized boxes. We use 4 test points in each box, namely their vertices, in order to construct the edges of the graph.

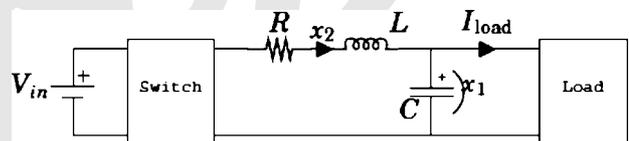


Figure 8 A switched DC/DC converter (taken from [12]).

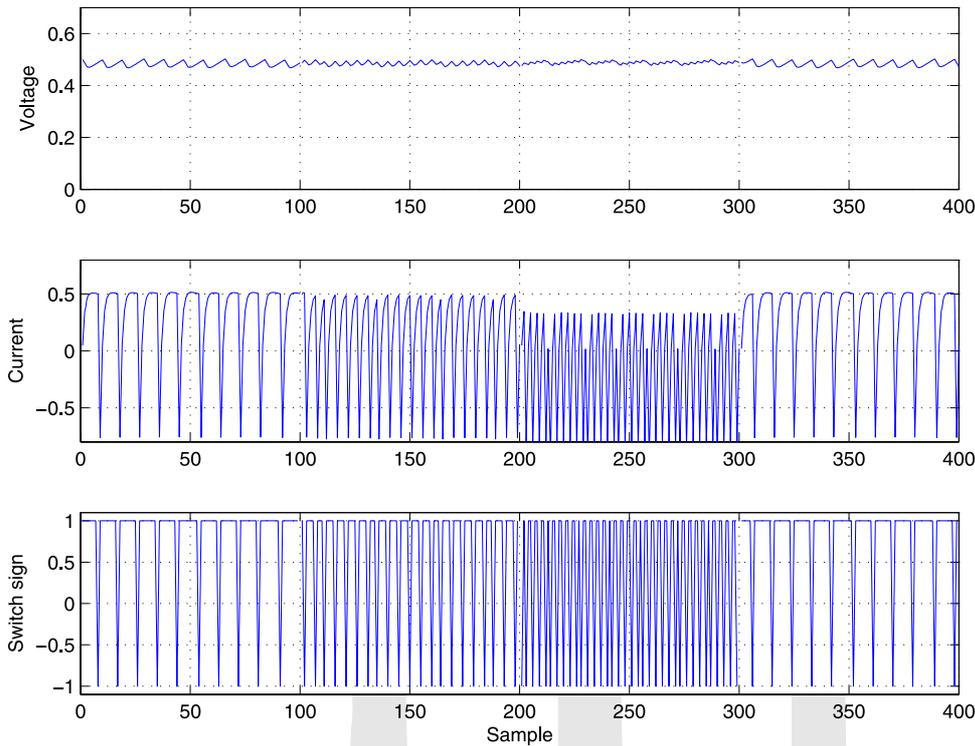


Figure 9 Simulation of the controlled switched power converter.

Using the resulting approximate optimal value function (associated to a nominal $I_{load} = 0.3$ A) and the associated feedback, we repeated the stabilization experiment from [12], where the load current is changed after every 100 iterations. Figure 9 shows the result of this simulation, proving that our controller stabilizes the system as requested.

7 Conclusion

We have developed a technique for the design of optimal controllers which is based on a piecewise constant approximation of the value function of the underlying optimal control problem. The technique combines ideas from set oriented numerics with shortest path algorithms from graph theory. The approach is particularly well suited for arbitrary nonlinear problems with highly irregular value function, complicated state constraints and naturally handles hybrid systems. Due to the *curse of dimensionality*, however, its application is currently restricted to systems with low dimensional state space. Current investigations aim at enabling the treatment of systems of moderate dimension (by, e.g., an adaptive construction of the underlying partition and by including past information, cf. [8]) as well as the treatment of discrete event systems.

References

- [1] M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA, 1997. With appendices by Maurizio Falcone and Pierpaolo Soravia.
- [2] D. P. Bertsekas. *Dynamic Programming and Optimal Control. Vol. 2*. Belmont, MA: Athena Scientific, 1995.
- [3] L. Grüne. An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation. *Numer. Math.*, 75(3):319–337, 1997.
- [4] L. Grüne and O. Junge. A set oriented approach to optimal feedback stabilization. *Sys. Ctrl. Let.*, 54(2):169–180, 2005.
- [5] 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, Kyoto, Japan*, pages 2089–2093, 2006.
- [6] L. Grüne and O. Junge. Approximately optimal nonlinear stabilization with preservation of the Lyapunov function property. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 702–707, 2007.
- [7] L. Grüne and O. Junge. Global optimal control of perturbed systems. *J. Optim. Theory Appl.*, 136(3):411–429, 2008.
- [8] L. Grüne and F. Müller. Set oriented optimal control using past information, Proceedings of the 18th International Symposium on Mathematical Theory of Networks and Systems MTNS2008, Blacksburg, Virginia, 2008, CD-Rom, Paper 125.
- [9] L. Grüne and D. Nešić. Optimization based stabilization of sampled-data nonlinear systems via their approximate discrete-time models. *SIAM J. Control Optim.*, 42:98–122, 2003.
- [10] O. Junge. Rigorous discretization of subdivision techniques. In B. Fiedler, K. Gröger, and J. Sprekels, editors, *EQUADIFF 99, Proceedings of the International Conference on Differential Equations*, volume 2, pages 916–918. World Sci. Publ., River Edge, NJ, 2000.
- [11] O. Junge and H. Osinga. A set oriented approach to global optimal control. *ESAIM Control Optim. Calc. Var.*, 10(2):259–270 (electronic), 2004.
- [12] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Trans. Autom. Control*, 51(8):1249–1260, 2006.
- [13] 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.

Received: October 8, 2008



Prof. Dr. Lars Grüne is professor for Applied Mathematics at the Mathematical Institute of the University of Bayreuth. Research areas: Mathematical systems and control theory, in particular numerical and optimization based methods for nonlinear systems.

Address: Mathematisches Institut, Universität Bayreuth, 95440 Bayreuth, Fax: +49-921-55-5361, e-mail: lars.gruene@uni-bayreuth.de



Prof. Dr. Oliver Junge is professor for Numerics of Complex Systems at the Faculty of Mathematics of the Technische Universität München. Research areas: Numerical analysis, dynamical systems, optimal control.

Address: Technische Universität München, Zentrum Mathematik, 85748 Garching, Fax: +49-89-289-17985, e-mail: junge@ma.tum.de

Vorschau auf Heft 7/2009

In unserem nächsten Heft finden Sie unter anderem folgende Themen:

- *L. Guzzella*: Gedanken zur Zukunft des Automobils
- *C. Wurmthaler, A. Kühnlein*: Modellgestützte Vorsteuerung für messbare Störungen
- *U. Katzke, B. Vogel-Heuser*: Vergleich der Anwendbarkeit von UML und UML-PA in der anlagennahen Softwareentwicklung der Automatisierungstechnik

Weitere Informationen über geplante Hefte, ausführliche Informationen über die in den letzten Heften der **at** erschienenen Beiträge sowie Hinweise für Autoren finden Sie im Internet unter <http://www.at-automatisierungstechnik.de>