

Global optimal feedbacks for stochastic quantized nonlinear event systems

Stefan Jerg, Oliver Junge, and Marcus Post[‡]

October 2013

Abstract

We consider nonlinear control systems for which only quantized and event-triggered state information is available and which are subject to random delays and losses in the transmission of the state to the controller. We present an optimization based approach for computing globally stabilizing controllers for such systems. Our method is based on recently developed set oriented techniques for transforming the problem into a shortest path problem on a weighted hypergraph. We show how to extend this approach to a system subject to a stochastic parameter and propose a corresponding model for dealing with transmission delays.

*Technische Universität München, Boltzmannstr. 3, 85747 Garching, Germany. jerg@ma.tum.de, oj@tum.de, mpost@ma.tum.de

[†]Research was partially supported by the priority program 1305 *Control Theory of Digitally Networked Dynamical Systems* of the German Research Foundation and the Bavarian State Ministry of Sciences, Research and Arts in the TopMath program.

1 Introduction

When the loop between a given control system (the *plant*) and an associated controller is closed via a *digital network* it is often necessary to include certain properties of the network into the model of the overall (closed loop) system, cf. [38]. For example, typically the bandwidth of the network imposes restrictions on how much data can be transmitted during a given time span. Also, in networks where data is transmitted in “packets” (which is the typical situation, e.g. in networks using TCP), the transmitted data might arrive delayed – or not at all (the packet gets *lost*).

There are essentially two ways in order to reduce the amount of information which is transmitted from the plant to the controller: One can reduce (1) the frequency by which data is transmitted and/or (2) the size of each data packet. A common approach for (1) is to transmit data not in regular time intervals (“sampled data approach”, cf. [2]), but only when necessary in order to guarantee stability, i.e. when a certain “event” requires to transmit new data. During the last years, this paradigm of *event-based control* is subject to much research effort, cf. [1, 17]. A method in order to reach goal (2) is to reduce the number of bits representing a state as much as possible. This is equivalent to using a *quantization* of the state space of the system (i.e. a partition). Whenever the current state is send to the controller, only the information on which cell of the partition the current state is contained in is transmitted, cf. [8, 14, 26, 31].

In addition to the requirement of reducing the amount of transmitted information as much as possible, typically any practical available network suffers from a further drawback, namely the fact that data is transmitted with varying *delays* – which may even be infinite in the sense that certain packets do not arrive at all at the controller, cf. [23]. This of course can noticeably deteriorate the behavior of the closed loop system, up to a complete loss of stabilizability for parts of the state space. There haven been various attempts to cope with this: For linear and non-quantized systems we refer to, e.g., [13], for non-linear systems with constant delay to [22], for modeling losses without delays to [27], and for a focus on network protocols to [28]. For a unified study of quantization and delay effects in nonlinear control systems see [21] where a quantized feedback control method is combined with the small-gain approach. The article [9] focuses on quantization and delay effects for linear systems using an LMI approach to find a controller with saturation. Recent investigations also deal with the design of encoders and decoders in order to stabilize a quantized time-delay nonlinear system and use a Liapunov-Krasowskii-functional approach and dynamic quantization to construct a stabilizing feedback [3, 6, 30], cf. also [24] for a small gain approach. In [32] an event based triggering scheme is presented in order to construct a real-time scheduler for stabilizing control tasks. We also refer to [29] for the construction of symbolic models for systems with time delays and to [35] for a study of distributed networked control systems with delays and losses, proposing a decentralized event-triggering scheme.

In this paper, we extend a recently developed approach for the construction of global optimal feedbacks for nonlinear quantized event systems which is based on a set oriented

discretization of the optimality principle (cf. [10–12, 16]) to the case where an additional external stochastic parameter is present in the system. We use this new construction together with an appropriately developed model for delays in order to design a global optimal feedback which to a certain extent is robust to delays. The underlying quantization is given by an arbitrary – but finite – partition of state space and yields a controller which practically stabilizes the system, i.e. it drives the system into a given target set. This is in contrast to, e.g., methods based on logarithmic quantization which are able to asymptotically stabilize to a point, cf. [36, 37]. We assume that the transmitted data is time stamped and that the plant and the controller possess synchronized clocks such that the controller can compute the actual delay of the received data. As with any method which is based on a space discretization of the optimality principle, our approach is subject to the curse of dimension, i.e. the number of nodes in the hypergraph scales exponentially in the dimension of the state space. Consequently, our method is restricted to systems of small dimension (up to four on current standard hardware, say). On the other hand, the class of systems to which it applies is rather general.

The paper is structured as follows: After recalling the basics of our construction from [11, 12] in Section 2 and giving an example which shows how badly delays can impact on the stabilizable set in Section 3, we develop the theoretical framework for dealing with such problems in Section 4 by discretizing a stochastic Bellman equation using a set oriented approach. There, we also prove a result on the stochastic stability of the associated feedback closed loop system. In Section 5, we then propose a corresponding model for incorporating delays and illustrate our concept by reconsidering the example from Section 3.

2 Optimal feedbacks for quantized event systems

The plant is modeled by a nonlinear discrete time control system (which may, e.g., be derived from a continuous time system by time sampling)

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, 1, 2, \dots, \quad (1)$$

where $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ is continuous, $\mathbf{x}_k \in \mathcal{X}$ is the state and $\mathbf{u}_k \in \mathcal{U}$ is the control input at time k , $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ compact. In addition to f , we are given a continuous running cost function $c : \mathcal{X} \times \mathcal{U} \rightarrow [0, \infty)$ as well as a closed *target set* $\mathcal{X}^* \subset \mathcal{X}$. We assume c to satisfy $c(\mathbf{x}, \mathbf{u}) = 0$ iff $\mathbf{x} \in \mathcal{X}^*$. Our goal is to compute a feedback law for this system which drives the system into the target set \mathcal{X}^* (where a different, locally acting controller takes over) while accumulating the least costs possible. However, the information which is transmitted from the plant to the controller is restricted in the following two ways:

1. *Event model:* The controller only receives information on the state whenever an *event* occurs. That is, even though the system (1) moves from \mathbf{x}_k to a new state

\mathbf{x}_{k+1} , this information possibly will not be transmitted to the controller. Instead, the plant “waits” until a certain condition on the new state is fulfilled (for example, when the new state exceeds a certain distance from the old one). We are going to check for this condition by introducing an *event function* $r : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{N} \cup \{\infty\}$. For example, $r(\mathbf{x}, \mathbf{u})$ might be defined as the smallest $r \in \mathbb{N}$ such that $\|f^r(\mathbf{x}, \mathbf{u}) - \mathbf{x}\| > \varepsilon$ for some prescribed tolerance $\varepsilon > 0$ (where the iterate f^r is defined by $f^0(\mathbf{x}, \mathbf{u}) = \mathbf{x}$, $f^r(\mathbf{x}, \mathbf{u}) = f(f^{r-1}(\mathbf{x}, \mathbf{u}), \mathbf{u})$). Note that we include the possibility $r(\mathbf{x}, \mathbf{u}) = \infty$, which handles the situation that for some $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$ there is no $r \in \mathbb{N}$ for which the encoded condition is fulfilled (for example, if $\mathbf{x} = f(\mathbf{x}, \mathbf{u})$). As in [12], based on the discrete time model (1) of the plant, we are now dealing with the discrete time system

$$\mathbf{x}_{\ell+1} = \tilde{f}(\mathbf{x}_\ell, \mathbf{u}_\ell), \quad \ell = 0, 1, \dots, \quad (2)$$

where $\tilde{f}(\mathbf{x}, \mathbf{u}) = f^{r(\mathbf{x}, \mathbf{u})}(\mathbf{x}, \mathbf{u})$ and we set $f^\infty(\mathbf{x}, \mathbf{u}) = \mathbf{x}$. Accordingly, we define an associated running cost $\tilde{c} : \mathcal{X} \times \mathcal{U} \rightarrow [0, \infty]$ by

$$\tilde{c}(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{r(\mathbf{x}, \mathbf{u})-1} c(f^k(\mathbf{x}, \mathbf{u}), \mathbf{u})$$

(with $\tilde{c}(\mathbf{x}, \mathbf{u}) = \infty$ if $r(\mathbf{x}, \mathbf{u}) = \infty$). The natural number ℓ enumerates the events and we can reconstruct the true time k from ℓ via the event function r by $k_{\ell+1} = k_\ell + r(\mathbf{x}_\ell, \mathbf{u}_\ell)$.

Remark: In practice, one will set $r(\mathbf{x}_\ell, \mathbf{u}_\ell) = \infty$ if $r(\mathbf{x}_\ell, \mathbf{u}_\ell) > R$ for some upper bound $R < \infty$.

2. *Quantization model:* The controller only receives *quantized* information on the state. Formally, we are given a (finite) partition $\mathcal{P} = \{P_1, \dots, P_d\}$ of *cells* $P_i \subset \mathcal{X}$ (in our implementation, we use boxes aligned with the coordinate axes). For each $\mathbf{x} \in \mathcal{X}$, we denote by $[\mathbf{x}] \in \mathcal{P}$ the cell containing \mathbf{x} . At event time ℓ , only $[\mathbf{x}_\ell]$ is transmitted from the plant to the controller. This fact is modeled by a *choice function* $\gamma : \mathcal{P} \rightarrow \mathcal{X}$ which chooses an arbitrary point from a given cell, i.e. γ fulfills $[\gamma(P)] = P$ for all $P \in \mathcal{P}$. We denote by Γ the set of all these choice functions. The quantized model of the plant is now given by the finite state system, cf. [11, 12],

$$P_{\ell+1} = F(P_\ell, \mathbf{u}_\ell, \gamma_\ell), \quad \ell = 0, 1, \dots, \quad (3)$$

defined by

$$F(P, \mathbf{u}, \gamma) = [\tilde{f}(\gamma(P), \mathbf{u})], \quad P \in \mathcal{P}, \mathbf{u} \in \mathcal{U}.$$

Computationally, an explicit construction of the choice function γ is not necessary. All we need to be able to compute (cf. the next section) is $F(P, \mathbf{u}, \Gamma) := \{F(P, \mathbf{u}, \gamma) \mid \gamma \in \Gamma\}$ which can be either approximated by mapping a finite set of

sample points from P or by using interval arithmetic in the case that the partition \mathcal{P} consists of rectangles. Both approaches can be made rigorous, cf. [15, 33]. As a result, in each step, in addition to \mathbf{u} , a choice function γ has to be chosen. Thus, we now have *two* control parameters, where γ should be viewed as having a perturbative effect on the dynamics. In fact, formally, together with a suitable cost function (cf. the next section) the system (3) constitutes a *dynamic game*.

Note that for any fixed \mathbf{u} , the function $\mathbf{x} \mapsto r(\mathbf{x}, \mathbf{u})$ is not necessarily constant on a cell. Accordingly, without further ado, it is not possible to recover the “true time k ” from the transition events in (3).

2.1 Computing the optimal feedback

In order to be compatible with our quantization, from now on we assume that \mathcal{X}^* is given by the closure of the union of the cells from a subset $X^* \subset \mathcal{P}$, i.e. $\mathcal{X}^* = \overline{\bigcup_{P \in X^*} P}$. For the quantized system (3) we define the cost function

$$C(P, \mathbf{u}) = \sup_{\mathbf{x} \in P} \tilde{c}(\mathbf{x}, \mathbf{u}).$$

For $P_0 \in \mathcal{P}$, $(\mathbf{u}_\ell)_\ell \in \mathcal{U}^{\mathbb{N}}$ and $(\gamma_\ell)_\ell \in \Gamma^{\mathbb{N}}$, the cost accumulated along a trajectory $(P_\ell)_\ell \in \mathcal{P}^{\mathbb{N}}$ of (3), starting in P_0 , is

$$J(P_0, (\mathbf{u}_\ell)_\ell, (\gamma_\ell)_\ell) = \sum_{\ell=0}^{L-1} C(P_\ell, \mathbf{u}_\ell),$$

where $L = L(P_0, (\mathbf{u}_\ell)_\ell, (\gamma_\ell)_\ell) := \inf\{\ell \geq 0 : P_\ell \in X^*\}$. Note that possibly $L = \infty$, in which case the series does not converge since there are finitely many partition elements and so $\min_{P \in \mathcal{P}} \inf_{\mathbf{u} \in \mathcal{U}} C(P, \mathbf{u}) > 0$ by the assumptions on c . The *optimal value function* is

$$V(P) = \sup_{\bar{\gamma}} \inf_{(\mathbf{u}_\ell)_\ell \in \mathcal{U}^{\mathbb{N}}} J(P, (\mathbf{u}_\ell)_\ell, \bar{\gamma}((\mathbf{u}_\ell)_\ell)),$$

where $\bar{\gamma} : \mathcal{U}^{\mathbb{N}} \rightarrow \Gamma^{\mathbb{N}}$ is a *strategy* of the form

$$\bar{\gamma}((\mathbf{u}_\ell)_\ell) = (\gamma_1(\mathbf{u}_1), \gamma_2(\mathbf{u}_1, \mathbf{u}_2), \gamma_3(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3), \dots)$$

and the sup in the definition of the optimal value function is over all strategies of this form. This construction models the fact that in each step of the dynamics, the choice function γ_ℓ is chosen after the control \mathbf{u}_ℓ , i.e. the “perturbing player” has the advantage of knowing the choice of \mathbf{u}_ℓ , cf. [7].

Typically, there will be cells $P \in \mathcal{P}$ with $V(P) = \infty$. For example, any cell P which contains a point $\mathbf{x} = \mathbf{x}_0 \in \mathcal{X}$ which is not stabilizable to \mathcal{X}^* , i.e. for which there is no control sequence $(\mathbf{u}_\ell)_\ell$ such that the associated trajectory $(\mathbf{x}_\ell)_\ell$ of (2) converges to \mathcal{X}^* will have $V(P) = \infty$. Another example is a cell P which contains a point \mathbf{x} with $r(\mathbf{x}, \mathbf{u}) = \infty$ for all $\mathbf{u} \in \mathcal{U}$.

We let $S = \{P \in \mathcal{P} \mid 0 < V(P) < \infty\}$ denote the *stabilizable* subset of \mathcal{P} and $\mathcal{S} = \cup_{P \in S} S \subset \mathcal{X}$. Note that we exclude the target region X^* from S here since we only want to control the system into \mathcal{X}^* . By standard arguments, cf. [4], the optimal value function V restricted to S is the unique solution to the *optimality principle*

$$V(P) = \inf_{\mathbf{u} \in \mathcal{U}} \left\{ C(P, \mathbf{u}) + \sup_{\gamma \in \Gamma} V(F(P, \mathbf{u}, \gamma)) \right\}, \quad P \in S,$$

with the boundary condition $V|_{X^*} = 0$. Given V , we can construct a feedback for (3) resp. (1) by setting

$$u(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} \left\{ C([\mathbf{x}], \mathbf{u}) + \sup_{\gamma \in \Gamma} V(F([\mathbf{x}], \mathbf{u}, \gamma)) \right\}, \quad \mathbf{x} \in \mathcal{S}.$$

Note that the minimum exists since V attains only finitely many values and f and c were assumed to be continuous and thus $\mathbf{u} \mapsto C([\mathbf{x}], \mathbf{u})$ is continuous on the compact set \mathcal{U} .

The optimal value function can be computed by an efficient shortest path algorithm applied to the hypergraph¹ $\mathcal{G} = (\mathcal{P}, \mathcal{E})$, whose nodes are the cells of the partition \mathcal{P} and whose edges are given by

$$\mathcal{E} = \{(P, F(P, \mathbf{u}, \Gamma)) \in \mathcal{P} \times 2^{\mathcal{P}} \mid \mathbf{u} \in \mathcal{U}\}$$

weighted by

$$w(P, \mathcal{N}) = \inf \{C(P, \mathbf{u}) : \mathbf{u} \in \mathcal{U}, F(P, \mathbf{u}, \Gamma) = \mathcal{N}\}, \quad (4)$$

cf. [11]. This hypergraph encodes local reachability information between the cells of \mathcal{P} . For a given control $\mathbf{u} \in \mathcal{U}$ and a given choice function $\gamma \in \Gamma$, $F(P, \mathbf{u}, \gamma)$ is a single cell from \mathcal{P} . Accordingly, $F(P, \mathbf{u}, \Gamma) = \{F(P, \mathbf{u}, \gamma) \in \mathcal{P} \mid \gamma \in \Gamma\}$ is the set of all cells which can be reached from P using this fixed $\mathbf{u} \in \mathcal{U}$. Since there are only finitely many cells in \mathcal{P} , there are only finitely many subsets $F(P, \mathbf{u}, \Gamma) \subset \mathcal{P}$ for varying $\mathbf{u} \in \mathcal{U}$ (even if \mathcal{U} is not finite). For each of these subsets, the hypergraph contains a corresponding edge (P, \mathcal{N}) , cf. Fig. 1. A shortest path in a such weighted hypergraph can be computed by an efficient Dijkstra-type algorithm, cf. [11, 34].

2.2 Construction of the hypergraph

In an implementation, the easiest way to construct this hypergraph is by mapping finitely many sampling points from each cell P (corresponding to choosing a finite set $\tilde{\Gamma}$ of choice functions from Γ), using finitely many sampling points $\tilde{\mathcal{U}} \subset \mathcal{U}$: For each cell $P \in \mathcal{P}$ and each $\tilde{u} \in \tilde{\mathcal{U}}$, compute

$$F(P, \tilde{u}, \tilde{\Gamma}) = [f(\tilde{x}, \tilde{u})] \in \mathcal{P}$$

¹For our purpose, a *hypergraph* is a pair $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ of a finite set \mathcal{P} of *nodes* and a set $\mathcal{E} \subset \mathcal{P} \times 2^{\mathcal{P}}$ of *edges* ($2^{\mathcal{P}}$ denotes the power set of \mathcal{P} , i.e. the set of all subsets of \mathcal{P}).

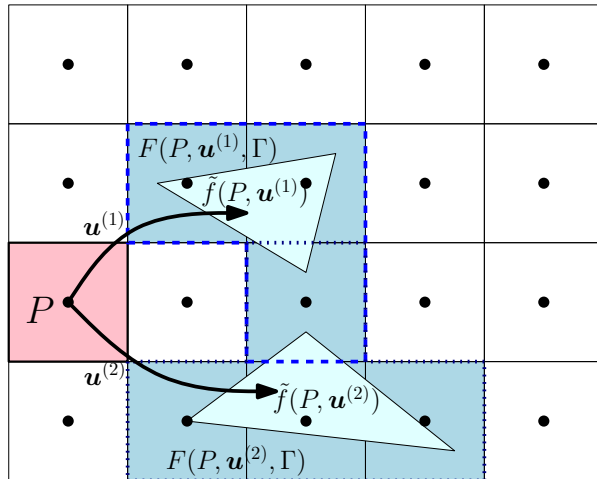


Figure 1: Edges of a hypergraph for two different controls $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$, starting in the same cell P . The two triangles correspond to $\hat{f}(P, \mathbf{u}^{(1)}) := \{\hat{f}(\mathbf{x}, \mathbf{u}^{(1)}) \mid \mathbf{x} \in P\}$ and $\tilde{f}(P, \mathbf{u}^{(2)})$, resp., while the dashed and dotted region correspond to $F(P, \mathbf{u}^{(1)}, \Gamma)$ and $F(P, \mathbf{u}^{(2)}, \Gamma)$, respectively.

where $\tilde{x} = \gamma(P, \tilde{u})$, $\gamma \in \tilde{\Gamma}$, is a sampling point from P .

The minimization in computing the weights (4) can then be performed discretely. This is the approach we have been using in the numerical experiments in the following sections. Typically, of course, depending on how the sampling points are chosen, this sampling approach will result in some edges being improperly constructed or even missing. In practice, this problem can largely be avoided by repeating the computation with an increasing number of sampling points until the results do not seem to change any more.

In principle, one could also construct the hypergraph in a rigorous way by using properly constructed Lipschitz estimates on the map f [15] or by using interval arithmetic [33]. We refer to [16] and [11] for further details on how to construct the hypergraph.

3 A controller subject to delays and losses

In addition to the two restrictions modeled in the previous section, namely (1) the event based information transmission and (2) the transmission of quantized information only, we now additionally assume that the transmission of the state information from the plant to the controller is realized via a digital network and that this transmission is subject to delays and even losses. More precisely, we assume that the state information P_ℓ generated at time k_ℓ reaches the controller at time $k_\ell + \delta_\ell$, where δ_ℓ is a random variable with a known distribution π on $\mathbb{N}_\infty := \{0, 1, 2, \dots, \infty\}$, cf. Figure 2.

In order to exemplify the effect of this additional restriction, we perform the following

experiment: we consider the classical inverted pendulum on a cart, cf. [16]. The dynamics of the pendulum is given by the continuous time control system

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

where $\varphi \in [0, 2\pi]$ denotes the angle between the pendulum and the upright position and $u \in \mathcal{U} := [-64, 64]$ is the force acting on the cart. 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

$$q(\varphi, \dot{\varphi}, u) = \frac{1}{2} (0.1\varphi^2 + 0.05\dot{\varphi}^2 + 0.01u^2). \quad (5)$$

Denoting the evolution operator of the system for constant control functions $u(t) \equiv \mathbf{u}$ by $\Phi^t(\mathbf{x}, \mathbf{u})$, $\mathbf{x} = (x_1, x_2) = (\varphi, \dot{\varphi})$, we consider the discrete time system $f(\mathbf{x}, \mathbf{u})$ given by approximating the evolution $\Phi^T(\mathbf{x}, \mathbf{u})$, $T = 0.01$, by the explicit Euler scheme with step size 0.0025 (and constant control $\mathbf{u} \in \mathcal{U}$). Likewise, the discrete time cost function $c(\mathbf{x}, \mathbf{u})$ is obtained by an associated numerical quadrature of the continuous time instantaneous cost. We choose $\mathcal{X} = [0, 2\pi] \times [-8, 8]$ as the state space and $\mathcal{X}^* = [-\frac{\pi}{8}, \frac{\pi}{8}] \times [-\frac{3}{4}, \frac{3}{4}]$ as the target region. For the partition \mathcal{P} of \mathcal{X} we use a uniform grid of $2^6 \times 2^6$ rectangles. By means of this grid we define the event function r as follows: Let $\mathbf{s}(\mathbf{x}) \in \mathbb{R}^n$ and $\rho(\mathbf{x}) \in \mathbb{R}^n$ denote the center and the radius of the rectangle containing \mathbf{x} , respectively. Then by means of the *event set*

$$\beta(\mathbf{x}) = \{\mathbf{y} \in \mathcal{X} : |y_i - s_i(\mathbf{x})| \leq e_r \cdot \rho_i(\mathbf{x}), i = 1, 2\} \quad (6)$$

with event radius $e_r = 9.4$ we define the event function

$$r(\mathbf{x}, \mathbf{u}) = \begin{cases} \min\{t \in \{0.01, 0.02, \dots, 10\} : \Phi^t(\mathbf{x}, \mathbf{u}) \notin \beta(\mathbf{x})\} & , \text{ if not empty} \\ \infty & , \text{ else} \end{cases} \quad (7)$$

In other words, the event function indicates when the corresponding event set is left. We note that an event set overlaps with other event sets, i.e. event sets do not form a partition of \mathcal{X} . In fact, using the given partition cells as event sets would not yield a stabilizing feedback in this example. A corresponding numerical experiment shows that in this case the image of a given cell near the target set stretches too far along the unstable direction of the origin. In contrast, the chosen event radius is rather arbitrary and could also be chosen such that the event sets are aligned with the given partition (e.g., $e_r = 9$) – which would enable the plant to emit events based on the given quantization of state space.

For the construction of the hypergraph, we use a grid of 5×5 equidistant sample points in each space cell P as well as 17 equally spaced points in the control set $\mathcal{U} = [-64, 64]$. The cost function C is computed by maximizing \tilde{c} over the grid points in each cell P .

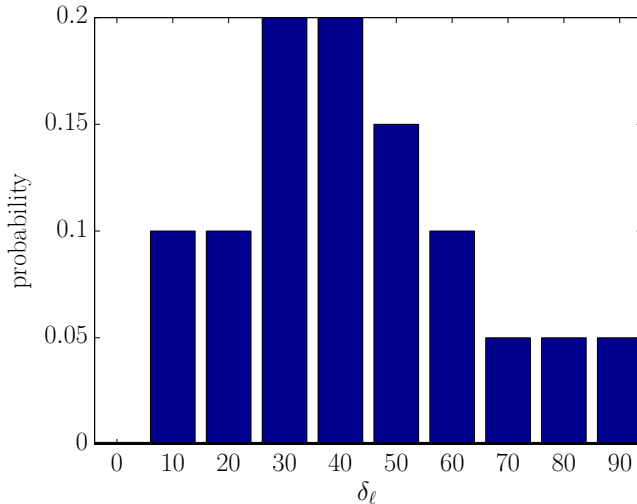


Figure 2: Probability distribution of the delays δ_ℓ in the inverted pendulum example. This discrete distribution is inspired by typical delay distributions as determined in [23].

In order to illustrate the effect of delays on the closed loop system we employ a feedback which was constructed for the model without delays as described above. We now simulate the closed loop system, first without delays and then with random delays up to 90 ms. The underlying distribution of the (independent) delays is depicted in Figure 2. The shape of this distribution is inspired by typical delay distributions as experimentally determined by [23]. Figure 3 shows the effect of the delays on the stabilizable set of the employed feedback.

4 Feedbacks for stochastic quantized event systems

In a digital network, packet delays and dropouts typically occur at random. In order to model this situation, we extend (2) by a third, random parameter δ (which will be used to model delays later on), i.e. we consider a *stochastic event system*

$$\mathbf{x}_{\ell+1} = g(\mathbf{x}_\ell, \mathbf{u}_\ell, \delta_\ell), \quad \ell = 0, 1, 2, \dots \quad (8)$$

where at each event instance ℓ the parameter $\delta_\ell \in \mathbb{N}_\infty$ is chosen independently from a given distribution $\pi : \mathbb{N}_\infty \rightarrow [0, 1]$. We assume the map $g : \mathcal{X} \times \mathcal{U} \times \mathbb{N}_\infty \rightarrow \mathcal{X}$ to be continuous in \mathbf{x} and \mathbf{u} . The running cost c and the target region \mathcal{X}^* are given as in Section 2, so we assume $c(\mathbf{x}, \mathbf{u}) = 0$ iff $\mathbf{x} \in \mathcal{X}^*$. In Section 5, we will develop a specific model g in which the parameter δ denotes the time delay by which the state information reaches the controller. In this section, we first abstractly extend the framework of the previous section to the case of a stochastic system (8).

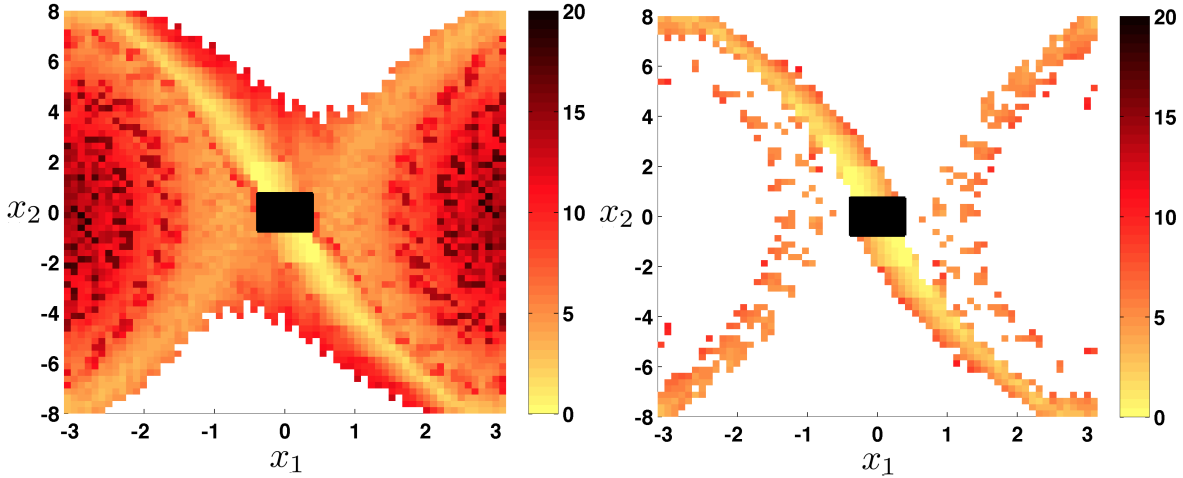


Figure 3: The inverted pendulum controlled by the feedback construction from Section 2 for the system without delay model: In color are the regions of state space which are stabilizable to a neighborhood \mathcal{X}^* of the origin (black centered rectangle) for a simulation without delay (left) and for one with stochastic delays up to 90 ms (right). The color of a cell indicates the average accumulated cost for initial states from that cell.

4.1 Quantization model

Again, the controller only receives quantized information on the state and, using the same construction as in Section 2, we model the plant by a stochastic finite state system

$$P_{\ell+1} = G(P_{\ell}, \mathbf{u}_{\ell}, \gamma_{\ell}, \delta_{\ell}), \quad \ell = 0, 1, \dots, \quad (9)$$

$P_{\ell} \in \mathcal{P}$, defined by

$$G(P, \mathbf{u}, \gamma, \delta) = [g(\gamma(P), \mathbf{u}, \delta)].$$

4.2 Computing the feedback

As in Section 2, the associated cost function is given by $C(P, \mathbf{u}) = \sup_{\mathbf{x} \in P} \tilde{c}(\mathbf{x}, \mathbf{u})$. By the assumptions on c we have $C(P, \mathbf{u}) = 0$ iff $P \in X^*$. Let

$$J(P_0, (\mathbf{u}_{\ell})_{\ell}, (\gamma_{\ell})_{\ell}) = \mathbb{E}_{\delta} \left\{ \sum_{\ell=0}^{L-1} C(P_{\ell}, \mathbf{u}_{\ell}) \right\} \in [0, \infty],$$

where $L = L(P_0, (\mathbf{u}_{\ell})_{\ell}, (\gamma_{\ell})_{\ell}, (\delta_{\ell})_{\ell}) := \inf\{\ell \geq 0 : P_{\ell} \in X^*\}$ and the random trajectory $P_{\ell} = P_{\ell}(P_0, (\mathbf{u}_{\ell})_{\ell}, (\gamma_{\ell})_{\ell}, (\delta_{\ell})_{\ell})$, $\ell = 0, 1, \dots$, is generated by (9) and the expectation is with respect to the product measure. Again, the optimal value function is

$$V(P) = \sup_{\bar{\gamma}} \inf_{(\mathbf{u}_{\ell})_{\ell} \in \mathcal{U}^{\mathbb{N}}} J(P, (\mathbf{u}_{\ell})_{\ell}, \bar{\gamma}((\mathbf{u}_{\ell})_{\ell})),$$

which on the stabilizable set $S = \{P \in \mathcal{P} \mid 0 < V(P) < \infty\}$ fulfills the optimality principle

$$V(P) = \inf_{\mathbf{u} \in \mathcal{U}} \left\{ C(P, \mathbf{u}) + \sup_{\gamma \in \Gamma} \mathbb{E}_{\delta} \{V(G(P, \mathbf{u}, \gamma, \delta))\} \right\} \quad (10)$$

together with the boundary condition $V|_{X^*} = 0$. Given V , we can construct a feedback for (8) by setting

$$u(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} \left\{ C([\mathbf{x}], \mathbf{u}) + \sup_{\gamma \in \Gamma} \mathbb{E} \{V(G([\mathbf{x}], \mathbf{u}, \gamma, \delta))\} \right\}, \quad (11)$$

(again, the minimum exists, cf. Section 2.A) for $\mathbf{x} \in \mathcal{S} := \cup_{P \in S} P$.

4.3 A stability theorem

Clearly, due to the randomness of the parameter δ , in general one cannot expect the feedback (11) to render the closed loop system (asymptotically) stable in a deterministic sense. However, one can prove stability with a certain probability. The key results here are from stochastic stability theory using stochastic Liapunov functions originally proved in [5, 18] and [19]. We are going to use a version from [20].

In summary, our result is as follows: Given a particular selection of $\lambda > 0$ bounding the attainable cost, one obtains the probability of actually achieving that cost (or lower). Furthermore, almost all of the trajectories achieving such a bound also converge to the target set. To be more precise, for $\lambda > 0$ and V from (10), let $\mathcal{S}_{\lambda} = \{\mathbf{x} \in \mathcal{X} : V([\mathbf{x}]) \leq \lambda\}$. Then using the optimal feedback (11), the closed loop system

$$\mathbf{x}_{\ell+1} = g(\mathbf{x}_{\ell}, u(\mathbf{x}_{\ell}), \delta_{\ell}), \quad \ell = 0, 1, 2, \dots, \quad (12)$$

is stochastically stable in the sense of the following theorem.

Theorem 4.1. *If $\mathbf{x}_0 \in \mathcal{S}_{\lambda}$ then with probability $1 - V([\mathbf{x}_0])/\lambda$, a (random) trajectory of (12) stays in \mathcal{S}_{λ} . Furthermore, for almost all trajectories $(\mathbf{x}_{\ell})_{\ell}$ which stay in \mathcal{S}_{λ} , we have that $\mathbf{x}_{\ell} \rightarrow \mathcal{X}^*$.*

Proof. We will show that $V(\mathbf{x}) := V([\mathbf{x}])$ is a stochastic Liapunov function for (12) in the sense of [20], Theorem 4.1 in Chapter 4. From (10) and (11) we have that

$$V(\mathbf{x}) = V([\mathbf{x}]) = C([\mathbf{x}], u(\mathbf{x})) + \sup_{\gamma \in \Gamma} \mathbb{E}_{\delta} \{V(G([\mathbf{x}], u(\mathbf{x}), \gamma, \delta))\},$$

i.e.

$$V(\mathbf{x}) - \sup_{\gamma \in \Gamma} \mathbb{E}_{\delta} \{V(G([\mathbf{x}], u(\mathbf{x}), \gamma, \delta))\} = C([\mathbf{x}], u(\mathbf{x})). \quad (13)$$

Now for any $\mathbf{u} \in \mathcal{U}$,

$$\begin{aligned} \mathbb{E}_{\delta} \{V(g(\mathbf{x}, \mathbf{u}, \delta))\} &\leq \sup_{\gamma \in \Gamma} \mathbb{E}_{\delta} \{V([g(\gamma([\mathbf{x}], \mathbf{u}), \mathbf{u}, \delta)])\} \\ &= \sup_{\gamma \in \Gamma} \mathbb{E}_{\delta} \{V(G([\mathbf{x}], \mathbf{u}, \gamma, \delta))\}, \end{aligned}$$

so that with $\mathbf{x} = \mathbf{x}_\ell$ and $u(\mathbf{x}) = u(\mathbf{x}_\ell)$ from (13) we get

$$V(\mathbf{x}_\ell) - \mathbb{E}_\delta \{V(\mathbf{x}_{\ell+1})\} \geq C([\mathbf{x}_\ell], u(\mathbf{x}_\ell)). \quad (14)$$

Since by (11) the feedback is constant on partition elements, i.e. $u(\mathbf{x}) = u([\mathbf{x}])$, the right hand side is a function constant on the finitely many $[\mathbf{x}_\ell]$ and greater than 0 for \mathbf{x}_ℓ not in the target set \mathcal{X}^* . Hence, there exists a nonnegative continuous function $\alpha(\mathbf{x}) \leq C([\mathbf{x}], u(\mathbf{x}))$ which is 0 exactly on the target set, and it immediately follows

$$V(\mathbf{x}_\ell) - \mathbb{E}_\delta \{V(\mathbf{x}_{\ell+1})\} \geq \alpha(\mathbf{x}_\ell) \quad (15)$$

which is the condition in [20], Theorem 4.1 (Chapter 4). \square

4.4 Implementation

In order to compute the optimal value function defined in the previous section we perform a standard value iteration (in Section 2 a Dijkstra type shortest path algorithm can be used due to the deterministic nature of the system). Based on (10), the value iteration reads

$$V_{j+1}(P) = \inf_{\mathbf{u} \in \mathcal{U}} \left\{ C(P, \mathbf{u}) + \sup_{\gamma \in \Gamma} \mathbb{E}_\delta \{V_j(G(P, \mathbf{u}, \gamma, \delta))\} \right\} \quad (16)$$

with $V_0(P) = 0$ if $P \in X^*$ and $V_0(P) = \infty$ else. As in similar situations (cf., e.g., [10]) the use of graph algorithms still proves helpful here. The main reason is that the dynamic game is represented by a graph for which an evaluation is much faster than its construction. Particularly, for the value iteration this means that the number of iterations does not influence the number of evaluations of G . For more details on how to construct the underlying hypergraph we refer to [10, 11]. However, for our stochastic

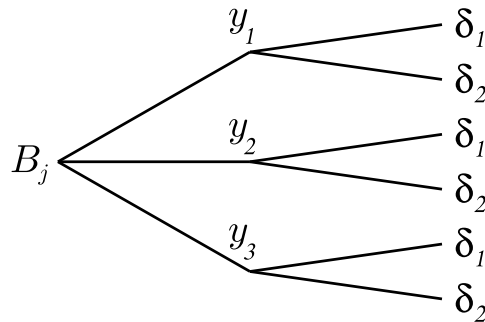


Figure 4: A hyperedge of order 2: the children at depth 1 correspond to the states in cell whereas the children at depth 2 correspond to the variation of δ .

framework, we need a slightly more general concept. We note that a classical hyperedge (which we call a hyperedge of order 1 here) is a tree of depth 1 with the root being the

start node and its children being sets of nodes reached by the one-step dynamics of the system under consideration. In order to be able to compute the optimal value function by (16) we introduce a new kind of hyperedge which we call hyperedge of order 2 (cf. Figure 4). A hyperedge of order 2 is a tree of depth 2, the children at depth 1 correspond to different states $y_i = \gamma_i(P)$ within the current cell P , whereas the children at depth 2 correspond to the different values of the stochastic parameter δ . Analogous to game trees (cf., e.g., [25]) it is now possible to first calculate the expectation over the values of nodes at depth 2, collect the result at depth 1 and then calculate the maximum over the y_i to obtain the new value $V_{j+1}(P)$ for cell P .

5 Numerical experiment

In this section, we first present an abstract model for an event system which incorporates delayed and lost transmissions of the state from the plant to the controller. We then apply the construction from the previous section in order to obtain a feedback which is robust to these delays and losses in a stochastic sense, i.e. the closed loop system will be stochastically stable in the sense of Theorem 4.1. We reconsider the example from Section 3 and experimentally demonstrate that our new feedback construction possesses almost the same stabilizable set as the original feedback for the system without delays and losses.

5.1 System and delay model

We consider a system modelled as described in Section 2, i.e. the plant is modelled by a nonlinear discrete time control system f and an event generator which implements an event function r . Whenever an event is generated in the plant, it is transmitted to the controller, but this transmission is subject to a delay $\delta \in \mathbb{N}_\infty$ (where $\delta = \infty$ corresponds to the possibility that the information does not reach the controller at all, i.e. a “packet loss”).

Since the transmission of the events from the plant to the controller is subject to delays and losses, the plant will still operate for some time with the old control input computed from the previous event, cf. Figure 5. Formally, we model this situation by the stochastic event system

$$\mathbf{z}_{\ell+1} = g(\mathbf{z}_\ell, \mathbf{u}_\ell, \delta_\ell), \quad \ell = 0, 1, 2, \dots, \quad (17)$$

where the time index ℓ enumerates the events as generated in the plant (cf. Section 2), the delays $\delta_\ell \in \mathbb{N}_\infty$ are chosen i.i.d. from a given distribution π , the vector $\mathbf{z}_\ell = (\mathbf{x}_\ell, \mathbf{w}_\ell) \in \mathcal{Z} := \mathcal{X} \times \mathcal{U}$ denotes the extended state ($\mathbf{x}_\ell \in \mathcal{X}$ the current state, $\mathbf{w}_\ell \in \mathcal{U}$ the old control input) and the mapping g is defined as follows:

$$g(\mathbf{z}, \mathbf{u}, \delta) = g((\mathbf{x}, \mathbf{w}), \mathbf{u}, \delta) = \begin{bmatrix} f^s(f^t(\mathbf{x}, \mathbf{w}), \mathbf{u}) \\ \mathbf{w}' \end{bmatrix},$$

where

$$\begin{aligned}
 t &= t(\delta, \mathbf{z}) = \min\{\delta, r(\mathbf{z})\}, \\
 s &= s(\delta, \mathbf{z}, \mathbf{u}, t) = s(\delta, (\mathbf{x}, \mathbf{w}), \mathbf{u}, t) = \begin{cases} r(f^t(\mathbf{x}, \mathbf{w}), \mathbf{u}) & \text{if } \delta < r(\mathbf{z}), \\ 0 & \text{if } \delta \geq r(\mathbf{z}), \end{cases} \\
 \mathbf{w}' &= \mathbf{w}'(\delta, \mathbf{z}, \mathbf{u}) = \mathbf{w}'(\delta, (\mathbf{x}, \mathbf{w}), \mathbf{u}) = \begin{cases} \mathbf{u} & \text{if } \delta < r(\mathbf{z}), \\ \mathbf{w} & \text{if } \delta \geq r(\mathbf{z}). \end{cases}
 \end{aligned}$$

In this model, any delay $\delta \geq r(\mathbf{z})$ is treated as $\delta = \infty$, i.e. as if the corresponding data would never reach the controller.

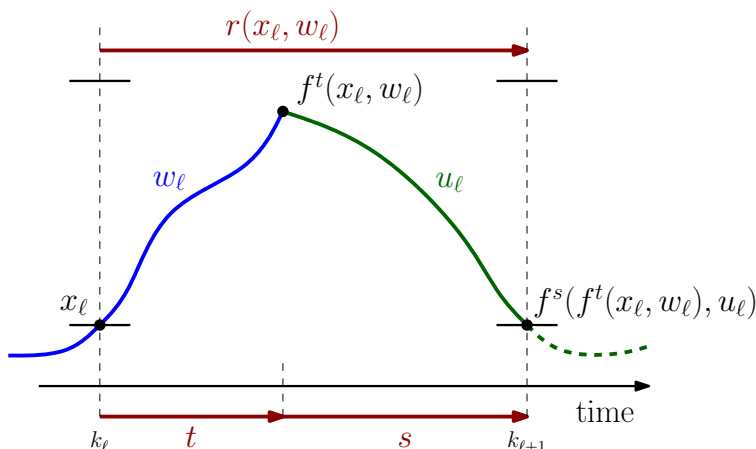


Figure 5: Delay model: at time k_ℓ the ℓ -th event is generated and the system is in state x_ℓ . The transmission of the state information from the plant to the controller is delayed by t time units, during which the old control input w_ℓ is still operational. At time $k_\ell+t$ (when the plant is already in state $f^t(x_\ell, w_\ell)$) the state information x_ℓ reaches the controller, changing the input to its new value $u_\ell = u(x_\ell)$.

5.2 The delayed inverted pendulum reconsidered

We reconsider our example from Section 3. However, we now compute a feedback law based on the framework of Section 4, i.e. the computation of the feedback already utilizes a model that includes the delay distributed according to Figure 2. In order to experimentally check for the stabilizable set in phase space, we randomly choose 100 finite sequences $(\delta_\ell^i)_{\ell=0}^{1000}$, $i = 1, \dots, 100$, by choosing each δ_ℓ^i i.i.d. according to the given distribution and 25 sample points in each partition cell. If the feedback trajectory of at least one sample point associated to some delay sequence $(\delta_\ell^i)_{\ell=0}^{1000}$ leaves the given phase space \mathcal{X} or does not reach the target set \mathcal{X}^* then this cell will be considered as being not stabilizable to the target region, cf. Figures 3 and 6. These figures illustrate that by incorporating the delay into the construction of the controller a much larger region of

the state space \mathcal{X} remains stabilizable. In fact, the stabilizable set for the delayed system with delay-based controller almost does not deteriorate in comparison to the undelayed system with the standard controller. It seems like at the boundary of the controllable region some discretization effects occur due to the finite number of test points in each box.

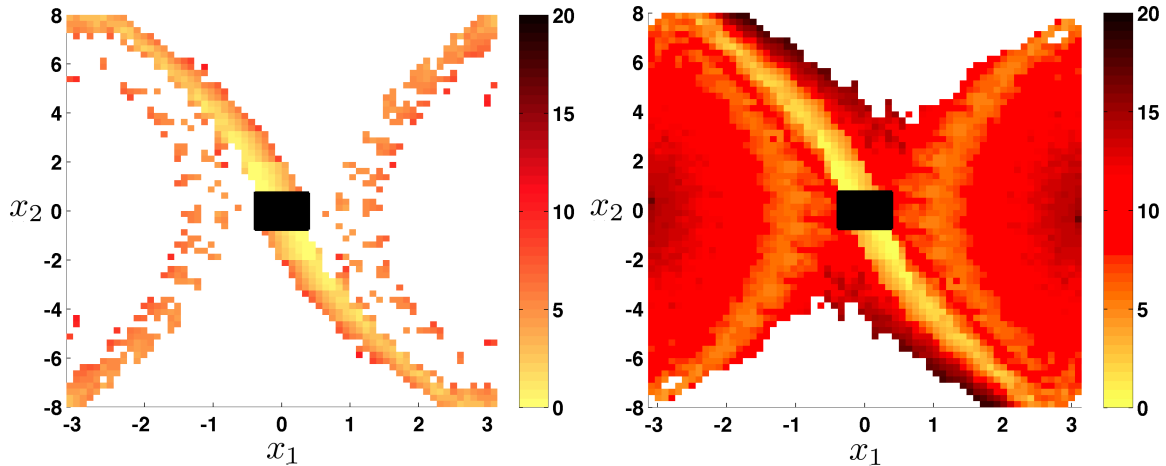


Figure 6: The inverted pendulum controlled by the feedback construction from Section 2 (left) and from Section 4 (right): In color are the regions of state space which are stabilizable to a neighborhood \mathcal{X}^* of the origin (black centered rectangle) for a simulation with stochastic delays. The color of a cell indicates the average accumulated cost for initial states from that cell.

References

- [1] K. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and Design of Nonlinear Control Systems*, pages 127–147. Springer-Verlag, 2008.
- [2] K. Åström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design, Third Edition*. Dover Books on Electrical Engineering Series. Dover Publications, 2011.
- [3] S.-I. Azuma and T. Sugie. Dynamic quantization of nonlinear control systems. *Automatic Control, IEEE Transactions on*, 57(4):875–888, 2012.
- [4] D. P. Bertsekas. *Dynamic Programming and Optimal Control. Vol. 2*. Belmont, MA: Athena Scientific, 1995.
- [5] R. S. Bucy. Stability and positive supermartingales. *J. Differential Equations*, 1:151–155, 1965.

- [6] C. De Persis and F. Mazenc. Stability of quantized time-delay nonlinear systems: A Lyapunov-Krasowskii-functional approach. In *Proc. of IEEE Conf. on Decision and Control*, pages 4093–4098, 2009.
- [7] W. Fleming. The convergence problem for differential games. *Journal of Mathematical Analysis and Applications*, 3:102–116, 1961.
- [8] D. Förstner, M. Jung, and J. Lunze. A discrete-event model of asynchronous quantised systems. *Automatica*, 38(8):1277–1286, 2002.
- [9] E. Fridman and M. Dambrine. Control under quantization, saturation and delay: An LMI approach. *Automatica*, 45(10):2258 – 2264, 2009.
- [10] L. Grüne and O. Junge. Approximately optimal nonlinear stabilization with preservation of the Lyapunov function property. In *Proc. 46th IEEE CDC*, pages 702–707, 2007.
- [11] L. Grüne and O. Junge. Global optimal control of perturbed systems. *JOTA*, 136(3):411–429, 2008.
- [12] L. Grüne and F. Müller. An algorithm for event-based optimal feedback control. In *Proc. 48th IEEE CDC*, pages 5311–5316, 2009.
- [13] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proc. IEEE*, 95(1):138–162, 2007.
- [14] H. Hsu. Global analysis by cell mapping. *Int. J. Bif. Chaos*, 2:727–771, 1992.
- [15] 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, 2000.
- [16] O. Junge and H. Osinga. A set oriented approach to global optimal control. *ESAIM Control Optim. Calc. Var.*, 10(2):259–270, 2004.
- [17] E. Kofman and J. Braslavsky. Level crossing sampling in feedback stabilization under data-rate constraints. In *Proc. IEEE CDC*, pages 4423–4428, San Diego, USA, 2006.
- [18] H. Kushner. On the stability of stochastic dynamical systems. *Proc. Nat. Acad. Sci. USA*, 53:8–12, 1965.
- [19] H. Kushner. *Stochastic stability and control*. Mathematics in Science and Engineering, Vol. 33. Academic Press, 1967.
- [20] H. Kushner and G. Yin. *Stochastic approximation algorithms and applications*. Applications of Mathematics, Vol. 35. Springer-Verlag, 1997.

- [21] D. Liberzon. Quantization, time delays, and nonlinear stabilization. *IEEE Transactions on Automatic Control*, 51(7):1190–1195, 2006.
- [22] D. Liberzon. Nonlinear control with limited information. *Commun. Inf. Syst.*, 9(1):41–58, 2009.
- [23] L. Litz, T. Gabriel, M. Groß, and O. Gabel. Networked Control Systems (NCS) – Stand und Ausblick. *at – Automatisierungstechnik*, 56(1):4–19, 2008.
- [24] T. Liu, Z.-P. Jiang, and D. J. Hill. Small-gain based output-feedback controller design for a class of nonlinear systems with actuator dynamic quantization. *Automatic Control, IEEE Transactions on*, 57(5):1326–1332, 2012.
- [25] U. Lorenz and B. Monien. Error analysis in minimax trees. *TCS*, 313(3):485–498, 2004.
- [26] J. Lunze. Qualitative modelling of linear dynamical systems with quantized state measurements. *Automatica*, 30(3):417–431, 1994.
- [27] S. Mastellone, C. Abdallah, and P. Dorato. Model-based networked control for nonlinear systems with stochastic packet dropout. In *Proc. American Control Conference.*, volume 4, pages 2365–2370, 2005.
- [28] D. Nesic and D. Liberzon. A unified framework for design and analysis of networked and quantized control systems. *IEEE Transactions on Automatic Control*, 54(4):732–747, 2009.
- [29] G. Pola, P. Pepe, M. D. D. Benedetto, and P. Tabuada. Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Systems and Control Letters*, 59(6):365–373, 2010.
- [30] R. Sailer and F. Wirth. Stabilization of nonlinear systems with delayed data-rate-limited feedback. In *Proc. European Control Conference*, pages 1734–1739. ECC2009, July 2009.
- [31] J. Schroeder. *Modeling, State Observation and Diagnosis of Quantized Systems*. Springer, Berlin, 2003.
- [32] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52:1680–1685, 2007.
- [33] W. Tucker. A rigorous ODE solver and Smale’s 14th problem. *Found. Comp. Math.*, 2(1):53–117, 2002.
- [34] M. von Lossow. A min-man version of Dijkstra’s algorithm with application to perturbed optimal control problems. In *In: Proceedings of the GAMM Annual Meeting, Zürich, Switzerland*, 2007.

- [35] X. Wang and M. Lemmon. Event-triggering in distributed networked systems with data dropouts and delays. In *Hybrid Systems: computation and control (HSCC)*, San Francisco, California, 2009.
- [36] C. Zhang, K. Chen, and G. Dullerud. Stabilization of markovian jump linear systems with limited information - a convex approach. In *Proc. ACC*, pages 4013–4019, 2009.
- [37] C. Zhang and G. Dullerud. Uniform stabilization of markovian jump linear systems with logarithmic quantization. In *Proc. IEEE CDC*, pages 2094 – 2099, 2009.
- [38] L. Zhang, H. Gao, and O. Kaynak. Network-induced constraints in networked control systems-a survey. *IEEE Transactions on Industrial Informatics*, 9(1):403–416, 2013.