PHYSCON 2015, Istanbul, Turkey, 19–22 August, 2015

TIME OPTIMAL SELF-TRIGGERED FEEDBACK CONTROL

Jorge Estrela da Silva
School of Engineering
Polytechnic of Porto
Portugal
jes@isep.ipp.pt

João Borges de Sousa
Faculty of Engineering
University of Porto
Portugal
jtasso@fe.up.pt

Fernando Lobo Pereira
Faculty of Engineering
University of Porto
Portugal
flp@fe.up.pt

Abstract
A dynamic programming based methodology is proposed for the design of aperiodic state feedback controllers for sampled data systems. The aperiodic sampling strategy follows a self-triggered control approach, i.e., the feedback controller computes both the control command and the next sampling instant. Three problem formulations are discussed for the self-triggered case: to reach a target in a given time with a minimum number of sampling instants; to reach a target no later than a given time with a minimum number of sampling instants; to reach the target in minimum time, with no more sampling instants than necessary. The control design is robust with respect to bounded disturbances.

Key words
aperiodic control, self-triggered control, dynamic programming.

1 Introduction

Periodic discrete-time control is based on the assumption that the cost of sampling the environment and changing the actuation value is negligible. However, there are several scenarios where such assumption is not verified. For instance, in mechanical systems, the simple act of changing the actuation value may incur in costs (e.g., wear and tear, energy losses due to static friction) which must be taken into account in the design of the feedback controller. In this case, it might still be adequate to keep a periodic sampling rate and simply design a feedback controller that changes the control command only when worthwhile. However, for systems designed with autonomy in mind, it may be desirable that the control system is partially switched off as frequently as possible. Additionally, when sensing involves data network utilization, as happens in networked systems [Hespanha et al., 2007], increasing the time between sampling instants leads to a decrease in network bandwidth usage, as also in the power consumption.

There are several approaches for aperiodic control [Mahmoud and Memon, 2015, Fiter et al., 2015]. One such approach is to make the state feedback controller responsible for the choice of the next sampling instant. This means that, at each sampling instant, the controller must compute the control command and also the time of next sampling instant. This approach has been coined as self-triggered control by some authors [Wang and Lemmon, 2009, Anta and Tabuada, 2010]. The typical objective of the self-triggered approach is to design a control law that meets a trade-off between the number sampling instants and some other given performance index. In [Anta and Tabuada, 2010], self-triggered approach is studied for the stabilization of state-dependent homogeneous systems and polynomial systems; in this work, the next sampling instant is chosen such that a desired decrease condition, with respect to a given Lyapunov function, is met, thus ensuring exponential input-to-state stability. That approach is extended to smooth control systems in [Anta and Tabuada, 2012]. In [Fiter et al., 2012], the authors claim to present the first online implementation of a self-triggered strategy, based on LMIs and a precomputed Lyapunov–Razumikhin function. In [Gommans et al., 2014], a self-triggered approach is developed for unconstrained discrete-time linear time-invariant systems subject to random noise with discounted quadratic cost functions.

In this paper, the problem of finite time target reachability using self-triggered control is discussed. To our best knowledge, this has not been discussed in the existing literature. We propose a self-triggered approach where the underlying storage function (in the Lyapunov sense) is the minimum number of sampling instants to reach a given target $\mathcal{T}$ within a given time parametrization. In order to accomplish that, the problem of reaching $\mathcal{T}$ in finite time with a minimal number of sampling instants from any given initial state must be solved at the design stage. Three types of time parametrizations, corresponding to three sub-problems, are considered: target reachability at exact time, target reachability within a time interval $[0, t_f]$ and target reachability in minimum time. The value function [Krasovskii and Subbotin, 1988] corresponding to each sub-problem is then used as a storage function for the control laws, in a dynamic programming fashion.

A dynamic programming based numerical algorithm, suitable for systems with Lipschitz continuous dynamics and bounded inputs, is proposed for the computation of the value function. Bounded disturbances and state-constraints are also considered. Robustness with respect to bounded disturbances is handled by formulating and solving the problem as two-person zero-sum deterministic differential games [Krasovskii and Subbotin, 1988]. For each considered sub-problem, the numerical algorithm computes an approximation of the value function (the minimal number of sampling instants) for a bounded region of the state space. Using the computed approximation of the value function, it is possible to synthesize a local state feedback control law that, at each sampling instant, will compute both the actuation and the time for the next sampling instant. The sampling instants are computed as multiples of a given minimum time interval between sampling instants. The performance of the control law depends essentially on the accuracy of the approximation of the value function. The results are illustrated by means of a problem involving a two-dimensional system.

Moreover, it must be remarked that the value function computed by the proposed algorithm can also be applied to the problem of minimizing the number of changes in the actuation for the periodic discrete-time setting. A controller meeting such objective can be implemented by using the value function computed by the proposed algorithm.

The paper is organized as follows. Section 2 presents the system model and assumptions. Section 3 presents the problem formulation and solution approach for undisturbed systems. In section 4, the approach is extended for systems with bounded disturbances. Section 5 presents a numerical algorithm for the solution of the problem. The methodology is illustrated with an application example in section 6. The paper ends with some concluding remarks in section 7.

2 System model

In what follows, the notation $\mathbb{N} = \{1, 2, 3, \ldots\}$ and $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$ is used. The model of the controlled system is of the form $\dot{x}(t) = f(x(t), u(t), v(t))$, where $x \in \mathbb{R}^n$ is the system state, $u(t) \in U_u$ is the control input, $v(t) \in U_v$ is a disturbance input and $f(x, u, v)$ is a Lipschitz continuous function in the region of interest $\mathcal{K} \subset \mathbb{R}^n$, where $\mathcal{K}$ is a closed set. The set U_v is compact, and U_u is assumed to be a discrete and finite set, with cardinality n_u, as typically happens for computer controlled systems. The existence of base clock with a period of Δ_t is assumed, with k defined as the number of clock cycles since the initial time $t = 0$. The input sequence $u(.)$ is the output of a sample-and-hold scheme with minimal hold time Δ_t (one clock cycle):

$$u(.) \in \mathcal{U}_u := \{u : \mathbb{R}_0^+ \to U_u : \\ \forall s \in [0, \Delta_t), k \in \mathbb{N}_0 : u(k\Delta_t + s) = u(k\Delta_t)\} \quad (1)$$

Therefore, the minimum time between sampling instants will be Δ_t. This also precludes the consideration of complex open-loop control sequences.

The disturbance input sequence is also assumed to be the output of a sample-and-hold scheme with minimal hold time Δ_t:

$$v(.) \in \mathcal{U}_v := \{v : \mathbb{R}_0^+ \to U_v : \\ \forall s \in [0, \Delta_t), k \in \mathbb{N}_0 : v(k\Delta_t + s) = v(k\Delta_t)\} \quad (2)$$

It is assumed that $f(x, u, v)$ and U_v are chosen in order to simulate the behaviour of the system when subjected to the optimal continuous-time variations of the disturbance input. This might not be trivial even for moderately complex systems but, on the other hand, it allows us to treat the system trajectories in the sense of π-trajectories (see [Krasovskii and Subbotin, 1988] and [Clarke et al., 1997]).

Moreover, the dynamics are bounded as follows:

$$\forall x \in \mathcal{K}, u \in U_u, v \in U_v : |f_i(x, u, v)| \leq f_{i,\max} \quad (3)$$

where $f_i(x, u, v)$, with $i \in \{1, \ldots, n\}$, is the i^{th} component of $f(x, u, v)$.

Define $y_\Delta(x, s, u, v)$ as the state of the system s units of time after departing from x with constant u and v. It is implied that, in order to compute $y_\Delta(x, s, u, v)$ it is enough that $v(t)$ is defined for $t \in [0, s)$. Similarly, define $y_u(x, s, u, v)$ for $u \in U_u$ and $v(.) \in \mathcal{U}_v$, and $y(x, s, u, v)$ for $u(.) \in \mathcal{U}_u$ and $v(.) \in \mathcal{U}_v$.

Consider also the set of augmented control sequences

$$\mathcal{U}_{u_a} := \{(u, \gamma) : u \in \mathcal{U}_u,\ \gamma : \mathbb{R}_0^+ \to \mathbb{N}\} \quad (4)$$

where $\gamma(t)$ is the number of clock cycles to be completed until the next sampling instant. The number of sampling instants up to $t = k\Delta_t$ is computed as follows:

$$n_s(k\Delta_t) = \begin{cases} 1, & k = 0 \\ n_s((k-1)\Delta_t), & \gamma((k-1)\Delta_t) > 1 \\ n_s((k-1)\Delta_t) + 1, & \gamma((k-1)\Delta_t) = 1 \end{cases}$$

Remark 1. *For many systems, the sampling action may be a little bit more complex than just reading the sensors at the desired sampling instant. Sensor measurements may be delivered asynchronously and, in general, some form of state estimation is used. Therefore, in what follows, it is assumed that the sensor measurements are queued and a suitable state estimator is executed only at the sampling instants, so that the purpose of minimizing the number of executions of the control law is not defeated.*

3 Undisturbed systems

3.1 Problem formulation

Consider the problem of ensuring that the system is at a given target set, $\mathcal{T} \subset \mathcal{K}$, k_{tt} clock cycles after departing from the initial state x_0, with a minimum number of control switches:

$$V(k_{tt}, x_0) = \min_{u_a \in \mathcal{U}_{u_a}} n_s((k_{tt}-1)\Delta_t) \tag{5}$$

subject to:

$$\dot{x}(t) = f(x(t), u(t), 0)$$
$$x(0) = x_0,\ x(k_{tt}\Delta_t) \in \mathcal{T},\ x(t) \in \mathcal{K}$$

Moreover, the following boundary condition is considered:

$$V(0, x) = \begin{cases} 0, & x \in \mathcal{T} \\ \infty, & x \notin \mathcal{T} \end{cases} \tag{6}$$

This formulation does not preclude the system from reaching the target before k_{tt} clock cycles are elapsed. The objective of reaching the target exactly after k_{tt} clock cycles, and not before, can be formulated by considering the following state constraint:

$$x(t) \notin \mathcal{T}, t < k_{tt}\Delta_t \tag{7}$$

Assuming the problem has a solution, define $u_a^*(.)$ as the minimizer of (5). If the problem does not have a solution, define $V(k_{tt}, x_0) = \infty$.

The main objective is to design a state feedback control law $f_c : \mathbb{N} \times \mathbb{R}^n \to U_a \times \mathbb{N}$ such that $f_c(k_{tt}, x) = u_a^*(0), \forall (k_{tt}, x) \in \mathbb{N} \times \mathbb{R}^n$.

Two additional problems are also considered: finding a control law $f_{c,\min N}(k_{tt}, x)$ such that the system reaches $\mathcal{T}$ in no more than k_{tt} clock cycles, with minimal number of sampling instants; finding a control law $f_{c,\min T}(x)$ such that the system reaches $\mathcal{T}$ in the minimum number of clock cycles.

3.2 Solution approach

Consider the following family of (backward in time) reachable sets (also known as *capture basins*), $R : U_u \times \mathbb{N}_0 \times (\mathbb{N} \times \mathbb{R}^n) \to (\mathbb{N} \times \mathbb{R}^n)$:

$$\begin{aligned} R(u, \gamma, S) :=& \{(k_{tt}, x_0) \in (\mathbb{N} \times \mathbb{R}^n) \setminus S : \\ & \exists s \in \{1, \ldots, \min(\gamma, k_{tt})\} \\ & (k_{tt} - s, y_\Delta(x_0, \gamma\Delta_t, u, 0)) \in S\} \end{aligned} \tag{8}$$

$$R(u, 0, S) := \emptyset \tag{9}$$

and

$$R(u, S) := \lim_{\gamma \to \infty} R(u, \gamma, S) \tag{10}$$

The set $R(u, \gamma, S)$ is composed of the pairs $(k_{tt}, x_0) \notin S$ such that the system is able to reach S from (k_{tt}, x_0) with constant control value u in no more than γ clock cycles. Note that the existential quantifier in (8) is evaluated only at the discrete time instants dictated by the base clock.

Define

$$\begin{aligned} R_i := \{(k_{tt}, x_0) \notin R_0^{i-1} : \exists u \in U_u : \\ (k_{tt}, x_0) \in R(u, R_{i-1})\} \end{aligned} \tag{11}$$

with $R_j^k := \bigcup_{i=j}^k R_i$ and $R_0 := \{0\} \times \mathcal{T}$. Then, the value function $V : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{N}_0$ is determined as follows:

$$V(k_{tt}, x) = \begin{cases} NI(k_{tt}, x), & NI(k_{tt}, x) \neq \emptyset \\ \infty, & NI(k_{tt}, x) = \emptyset \end{cases} \tag{12}$$

$$NI(k_{tt}, x) := \{i : (k_{tt}, x) \in R_i\} \tag{13}$$

either empty or a singleton.

Consider the feedback control law $f_c : \mathbb{N} \times \mathbb{R}^n \to U_a$. The first argument of the control law is the desired number of periods Δ_t to reach the target. The system may eventually reach the target before that time but it is required to be at the target at the desired time. Note that the desired time to reach the target must be at least Δ_t, the minimum time interval between samples. This is in accordance with the practical implementation: the minimum duration for the controller output will be the minimum duration of the sample and hold cycle. For some systems (e.g., a torpedo), it is indifferent whether the system reaches the target at the sampling instants or between them. However, the present formulation ensures that target reachability is detectable by the computer based control system at the sampling instants. This way, the control system will be able to confirm that the target was reached and, eventually, to commute to another operational mode.

The feedback control law is computed using the dynamic programming approach:

$$f_c(k_{tt}, x) \in \arg\min_{u \in U_u, s \in \{1,\ldots,k_{tt}\}} V(k_{tt} - s, y_\Delta(x, s\Delta_t, u, 0)) \tag{14}$$

Remark 2. *In order to implement the search procedure of (14), $V(k_{tt} - s, y_\Delta(x, s\Delta_t, u, 0))$ is evaluated for different values of u and s until $V(k_{tt} - s, y_\Delta(x, s\Delta_t, u, 0)) < V(k_{tt}, x)$ is met (terminating condition). A straightforward way of implementing this search is to incrementally compute the trajectories for every $u \in U_u$, with time step Δ_t, until one of them meets the terminating condition. At each time step, trajectories reaching a state with value higher than $V(k_{tt}, x)$ are pruned from the search procedure.*

3.3 Complementary problems

The minimum number of sampling instants to reach the target in no more than k_{tt} clock cycles is given by

$$V_{\min N}(k_{tt}, x) = \underset{s \in \{1,\dots,k_{tt}\}}{\arg\min} V(s, x) \tag{15}$$

To reach the target in no more than k_{tt} clock cycles with minimal number of sampling instants, the following control law should be used at each sampling instant:

$$(u(k\Delta_t), \gamma(k\Delta_t)) = f_c(k_{\min}(k_{tt} - k, x(k\Delta_t)), x(k\Delta_t)) \tag{16}$$

where

$$k_{\min}(k_{tt}, x) = \underset{s \in \{1,\dots,k_{tt}\}}{\arg\min} V(s, x) \tag{17}$$

Once again, reachability is considered only at the sampling instants.

The minimum number of clock cycles to reach the target is simply given by

$$V_{\min T}(x) = \arg\min\{s \in \mathbb{N} : V(s, x) \neq \infty\} \tag{18}$$

The corresponding feedback control law, to be computed only at the sampling instants, is given by

$$(u(k\Delta_t), \gamma(k\Delta_t)) = f_c(V_{\min T}(x(k\Delta_t)), x(k\Delta_t)). \tag{19}$$

4 Disturbed systems

4.1 Problem formulation

This problem is addressed in the framework of differential games (see, e.g., [Krasovskii and Subbotin, 1988, Ch. 10]). Consider the following finite horizon dynamic optimization problem:

$$V(k_{tt}, x_0) = \min_{(c,M):\mathbb{N}\times\mathbb{R}^n \to U_u \times \mathbb{N}} \max_{v \in \mathcal{U}_v} n_s((k_{tt} - 1)\Delta_t) \tag{20}$$

subject to:

$$\dot{x}(t) = f(x(t), u(k\Delta_t), v(k\Delta_t))$$

$$x(0) = x_0,\ x(k_{tt}\Delta_t) \in \mathcal{T},\ x(t) \in \mathcal{K}$$

$$\begin{pmatrix} u(k\Delta_t) \\ \gamma(k\Delta_t) \end{pmatrix} = \begin{cases} \begin{pmatrix} c(k_{tt}, x_0) \\ M(k_{tt}, x_0) \end{pmatrix}, & k = 0 \\ \begin{pmatrix} u((k-1)\Delta_t) \\ \gamma((k-1)\Delta_t) - 1 \end{pmatrix}, & \gamma((k-1)\Delta_t) > 1 \\ \begin{pmatrix} c(k_{tt} - k, x(k\Delta_t)) \\ M(k_{tt} - k, x(k\Delta_t)) \end{pmatrix}, & \gamma((k-1)\Delta_t) = 1 \end{cases}$$

In this formulation, it is implicit that the disturbance knows $u(k\Delta_t)$ when choosing $v(k\Delta_t)$. Future values of the sequence $u_a(.) := (u(.), \gamma(.))$ are not fixed *a priori* for each maximization, since $u_a(.)$ is defined by the feedback strategy $f_c(k_{tt}, x_0) := (c(k_{tt}, x_0), M(k_{tt}, x_0))$, thus depending on future values of $v(.)$. This corresponds to the upper value of the game, suitable to model system behaviour under worst case adversarial conditions.

4.2 Solution approach

The solution approach is similar to the one of the undisturbed case. However, in this case, $R(u, \gamma, S)$ must take into account the effect of the disturbance:

$$\begin{aligned} R(u, \gamma, S) = \{ &(k_{tt}, x_0) \in (\mathbb{N} \times \mathbb{R}^n) \setminus S : \\ &\exists s \in \{1, \dots, \min(\gamma, k_{tt})\} \\ &\forall v \in \mathcal{U}_v : \\ &(k_{tt} - s, y_u(x_0, s\Delta_t, u, v)) \in S\} \end{aligned} \tag{21}$$

The sets $R(u, S)$ and R_i and R_j^k are computed as for the undisturbed case and the value function $V(k_{tt}, x)$ is defined as in (12).

Let us define the auxiliary value function $V(u, \gamma, k_{tt}, x)$ as the minimum number of sampling instants to reach the target from x in k_{tt} sampling intervals, keeping a constant control u over the following $\gamma\Delta_t$ units of time:

$$V(u, \gamma, k_{tt}, x) = \begin{cases} NI(u, \gamma, k_{tt}, x), & NI(u, \gamma, k_{tt}, x) \neq \emptyset \\ \infty, & NI(u, \gamma, k_{tt}, x) = \emptyset \end{cases} \tag{22}$$

with

$$NI(u, \gamma, k_{tt}, x) := \{i : (k_{tt}, x) \in R(u, \gamma, R_0^{i-1})\} \tag{23}$$

either empty or a singleton. The auxiliary value function $V(u, \gamma, k_{tt}, x)$ can be computed by application of the dynamic programming principle:

$$V(u, \gamma, k_{tt}, x) = \max_{v \in U_v} V(u, \gamma - 1, k_{tt} - 1, y_\Delta(x, \Delta_t, u, v)) \tag{24}$$

$$V(u, 0, k_{tt}, x) = \begin{cases} V(k_{tt}, x) + 1, & V(k_{tt}, x) \neq \infty \\ \infty, & V(k, x) = \infty \end{cases} \tag{25}$$

$$V(k_{tt}, x) = \min_{u \in U_u, \gamma \in \{1,\dots,k_{tt}\}} V(u, \gamma, k_{tt}, x) \tag{26}$$

Also, define $\gamma_{max}(u, k_{tt})$ as the maximum γ for which there are states from which it is possible to reach the target in $k_{tt}\Delta_t$ units of time keeping a constant control u over the following $\gamma\Delta_t$ units of time. Armed with this, the search space for the optimization in the control law can be reduced:

$$f_c(k_{tt}, x) \in \underset{u\in U_u, \gamma\in\{1,\ldots,\gamma_{\max}(u,k_{tt})\}}{\arg\min} V(u, \gamma, k_{tt}, x) \qquad (27)$$

The complementary problems are computed as for the undisturbed case.

4.3 Reducing the problem dimensionality

It is possible to embed the information regarding γ in an auxiliary function of the form $V(u, k_{tt}, x)$, and therefore reduce the problem dimensionality, with obvious advantages for numerical implementation. Consider the following value function, defined as number of sampling instants to reach the target, from x, in k_{tt} clock cycles, keeping a constant control u until the next sampling instant (or reaching the target):

$$V(u, k_{tt}, x) = \min\{V_u(u, k_{tt}, x), V_{u'}(u, k_{tt}, x)\} \qquad (28)$$

$$V_u(u, k_{tt}, x) = \max_{v\in U_v} V(u, k_{tt} - 1, y_\Delta(x, \Delta_t, u, v)) \qquad (29)$$

$$V_{u'}(u, k_{tt}, x) = \min_{u'\in U_u\setminus\{u\}} \max_{v\in U_v} V(u', k_{tt} - 1, y_\Delta(x, \Delta_t, u', v)) + 1 \qquad (30)$$

$$V(u, 0, x) = \begin{cases} 1, & x \in \mathcal{T} \\ \infty, & x \notin \mathcal{T} \end{cases} \qquad (31)$$

The control law is implemented as follows. At each sampling instant, compute

$$u = c(k_{tt}, x) \in \underset{u\in U_u}{\arg\min} V(u, k_{tt}, x) \qquad (32)$$

and define $V(k_{tt}, x) := V(c(k_{tt}, x), k_{tt}, x)$. If $V(k_{tt}, x) = 1$, then $M(k_{tt}, x) = k_{tt}$; otherwise, it is necessary to compute the optimal number of clock cycles to reach $R_{V(k_{tt},x)-1}$ with constant control input u and disturbance input sequence $v(.)$ given by:

$$v(k\Delta_t) \in \underset{v\in U_v}{\arg\max} \min(V_v(v, k), V'_v(v, k)) \qquad (33)$$

$$V_v(v, k) = V(u, k_{tt} - k, y_\Delta(x(k\Delta_t), \Delta_t, u, v)) \qquad (34)$$

$$V'_v(v, k) = \min_{u'\in U_v\setminus\{u\}} V(u', k_{tt} - k, y_\Delta(x(k\Delta_t), \Delta_t, u, v) + 1 \qquad (35)$$

The system trajectory is then simulated, in temporal increments of Δ_t, until it reaches $R_{V(k_{tt},x)-1}$.

5 Numerical Computation of the value function

5.1 Undisturbed systems

The definition of the value function for the undisturbed case (12) depends on the definition of R_i. In order to compute R_i, it is necessary to compute $R(u, S)$, for every $u \in U_u$ and $S \in \{R_0, R_1, \ldots\}$, as defined in (11). This is made in sequence. First, $R(u, R_0)$ is computed for every $u \in U_u$; from that computation, R_1 - the set of points (k_{tt}, x_0) from which the target is reachable with a single sampling instant - is obtained. Then, $R(u, R_1)$ is computed for every $u \in U_u$, in order to produce R_2, and the computation proceeds up to the desired maximum number of sampling instants. The central procedure for this algorithm is the computation of $R(u, S)$.

In the case of nonlinear systems, it is, in general, impracticable to compute the exact composition of $R(u, S)$. In what follows, it is assumed that an approximation of $R(u, S)$, $\tilde{R}(u, S)$, is obtained using numerical methods. Moreover, the domain for the numerical computation of $\tilde{R}(u, S)$ is a bounded subset of $\mathbb{N}_0 \times \mathbb{R}^n$, defined according to the maximal time horizon $k_{\max}\Delta_t$ and spatial region of interest. The approximation of the reachable sets $R(u, R_i), u \in U_u, i \in \mathbb{N}$ can be obtained using any existing method (see, e.g., [Mitchell, 2008], [Bokanowski et al., 2010] and [Kurzhanski et al., 2006]) for the desired degree of accuracy. In what follows, a grid-based method is assumed. The grid covers the desired region of $\mathbb{N}_0 \times \mathbb{R}^n$, with inter-node spacing Δ_t for the temporal dimension and inter-node spacing Δx_j for each spatial dimension j. Define $K_j = \text{ceil}(\frac{f_{j,\max}\Delta_t}{\Delta x_j})$, where ceil$(x)$ is the smallest integer not smaller than x. Moreover, Δ_t is used as the time-step of the numerical solver.

In the undisturbed case, $\tilde{R}(u, S)$ can be computed by simple evaluation of the trajectories $(k_{tt} - s, y_\Delta(x_0, s\Delta_t, u, 0)), s \in \{1, \ldots, k_{tt}\}$ departing from each grid node (k_{tt}, x_0) in the computational domain. If the trajectory from (k_{tt}, x_0) reaches S in no more than $k_{tt}\Delta_t$ units of time without leaving the computational domain, then (k_{tt}, x_0) is marked as belonging to $\tilde{R}(u, S)$.

By careful choice of the order of evaluation of the nodes, it is possible to avoid evaluating the whole trajectory from each node. The key idea is to follow the reverse order of the optimal the trajectories, i.e., to start processing from the target set and to visit each node only once. Only the points that can possibly reach the target at a given time, according to the envelop (3) of the system dynamics, are evaluated. This set of points is denominated narrow-band. Define $\tilde{R}(u, \gamma, S)$ as the numerical approximation of $R(u, \gamma, S)$. Then, the narrow-band at iteration $\gamma \geq 1$ is defined as

$$NB(\gamma) = \{(k_{tt}, x) \notin \tilde{R}(u, \gamma - 1, S) \bigcup S : \exists(k_{tt} - 1, y) \in \tilde{R}(u, \gamma - 1, S) \bigcup S : \left\|\text{diag}(K_1\Delta x_1, \ldots, K_n\Delta x_n)^{-1}(x - y)\right\|_\infty \leq 1\} \qquad (36)$$

Thus, the narrow band is initially composed of the nodes in the direct neighbourhood of S; afterwards, $NB(\gamma)$ is updated based on the approximation of $R(u, \gamma - 1, S)$, i.e., it is composed of the nodes in the direct neighbourhood of $\tilde{R}(u, \gamma - 1, S)$.

In each iteration γ, the trajectory $(k_{tt} - s, y_\Delta(x_0, s\Delta_t, u, 0))$, $s \in \{1, \ldots, \min(k_{tt}, \gamma)\}$ emanating from each node (k_{tt}, x_0) in $NB(\gamma)$ is computed in temporal increments of Δ_t until one of the following conditions is verified: a) the trajectory leaves the computational space; b) the trajectory reaches a cell with no vertex in $\tilde{R}(u, \gamma - 1, S) \bigcup S$; c) the trajectory reaches a cell with all its vertices in $\tilde{R}(u, \gamma - 1, S) \bigcup S$. If the latter condition is verified, then the node is marked as belonging to $\tilde{R}(u, \gamma, S)$.

This may introduce both over-approximation and under-approximation errors. More specifically, if all vertices of a cell are marked as belonging to $\tilde{R}(u, \gamma - 1, S) \bigcup S$ then the algorithm assumes that all points of that cell also belong to $R(u, \gamma - 1, S) \bigcup S$; however, this may hide concavities of $R(u, \gamma - 1, S) \bigcup S$, thus leading to over-approximation. On the other hand, whenever the intersection of $R(u, \gamma - 1, S) \bigcup S$ with a cell does not include all the vertices of the cell, the algorithm will produce an under-approximation of $R(u, \gamma - 1, S) \bigcup S$.

The computation of $\tilde{R}(u, S)$ starts with $NB(1)$ and finishes after a maximum of $\gamma_{\max} \leq k_{\max}$ iterations. Notice that $\tilde{R}(u, \gamma, S)$ is defined incrementally from $\gamma = 1$ to $\gamma = \gamma_{\max}$, meaning that, at the end of the computation, only $\tilde{R}(u, \gamma_{\max}, S)$ must be stored in memory.

5.2 Disturbed systems

In the disturbed case, the objective is to compute $V(u, k_{tt}, x)$. Nevertheless, like for the undisturbed case, the synthesis procedure is centred in the sequential computation of $\tilde{R}(u, S)$ for every $u \in U_u$ and $S \in \{R_0, R_1, \ldots, R_{k_{\max}}\}$. This is done by computing the auxiliary value function $V(u, k_{tt}, x)$ at the grid nodes, applying (28)-(31) with the narrow-band approach. Initially, the nodes corresponding to the target are marked with 0 and all the remaining nodes are marked as infinity.

The narrow-band approach is applied taking into account that $R(u, 1, R(u, \gamma - 1, S)) = R(u, \gamma, S) \setminus R(u, \gamma - 1, S)$. However, in this case, a single pass is not, in general, sufficient to obtain convergence to the solution. The narrow-band approach is based on the principle that the control input always seeks the lower values of the value function (note that the running cost is null), thus allowing the computation of the value function in a single pass, from the target set to the level sets associated with increasing values of the value function. However, in the disturbed case, the computation of the worst case disturbance for a given state requires the knowledge of the value function in the whole neighbourhood of that state (i.e., not only in the lower level sets). Therefore, in the disturbed case, a single pass of the narrow-band approach does not provide the best approximation of the value function. Several iterations are required until a good approximation is achieved.

6 Application example

Consider the following nonlinear system:

$$\dot{x}(t) = \begin{pmatrix} \sin(x_2(t)) + v(t) \\ u(t) \end{pmatrix} \tag{37}$$

with state x constrained to $[-16, 16] \times [-\pi, \pi]$, control input $u \in U_u = \{-0.26, 0, 0.26\}$ and bounded disturbance $v \in U_v = [-0.25, 0.25]$. The minimum control period is $\Delta_t = 100$ ms. The computations are performed on a $301 \times 241 \times 241$ regular grid, where the first dimension is the time to reach the target. This implies that the longest optimization horizon is $300\Delta_t = 30$ s. The target is $\mathcal{T} = \{x : x_1^2 + (x_2/0.5)^2 \leq 2\}$.

The procedure described in section 5 was employed to compute R_i from $i = 1$ to $i = 56$, after which no more pairs (t, x) were added to the reachable sets. Figures 1-6 illustrate some of the computed reachable sets. The capture basin for this example, with no dependence on the number of sampling instants, is simply the reunion of every R_i. Figure 6 shows the level sets of the capture basin up to a time horizon of 30 s. Inspection of figures 1-5 can provide some more insight on the proposed approach. In figure 1, it is possible to see that there is no state from which the target can be reached in 6 s using a constant actuation, computed at the initial time. Recall that the first sampling instant and control action are assumed to happen at the initial time $t = 0$ s. In figure 2, it can be seen that the set of states from which the target can be reached in 2 s starts to vanish. This is in accordance with the underlying system dynamics and given target, as can be concluded from figure 6. Focusing on the case of $t_f = 6$ s, it is possible to see that figure 3 fills some of the empty regions of figure 2, i.e., the former displays the states from which the target can only be reached with 3 sampling instants. The same can be verified for $t_f \in \{12, 18, 24\}$ in figures 3, 4 and 5.

7 Conclusions

This paper describes a numerical approach for the optimal self-triggered control of systems. Optimality is considered in the sense of minimal number of sampling instants to reach a target in a given time frame.

The numerical algorithm is based on dynamic programming principle and suffers from the usual dimensionality problems. Nevertheless, the computed value function can be used to implement state feedback control laws, and also as a benchmark for other control designs. Moreover, the feedback control laws for the three considered problems can be obtained using the same value function.

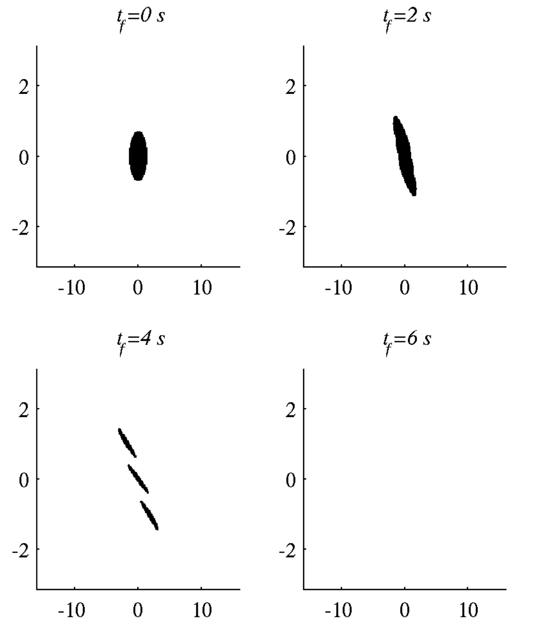

Figure 1. Projections of $R_0 \bigcup R_1$ for different time horizons.

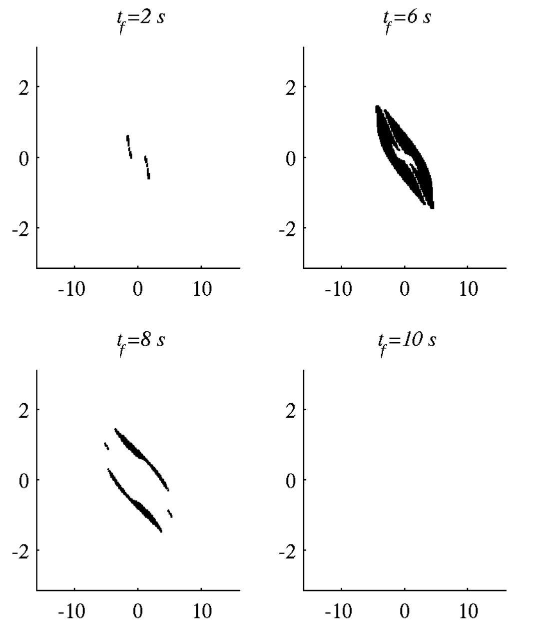

Figure 2. Projections of R_2 for different time horizons.

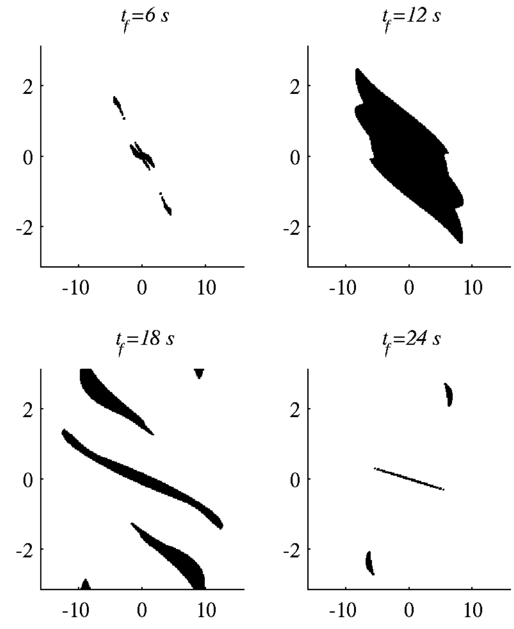

Figure 3. Projections of R_3 for different time horizons.

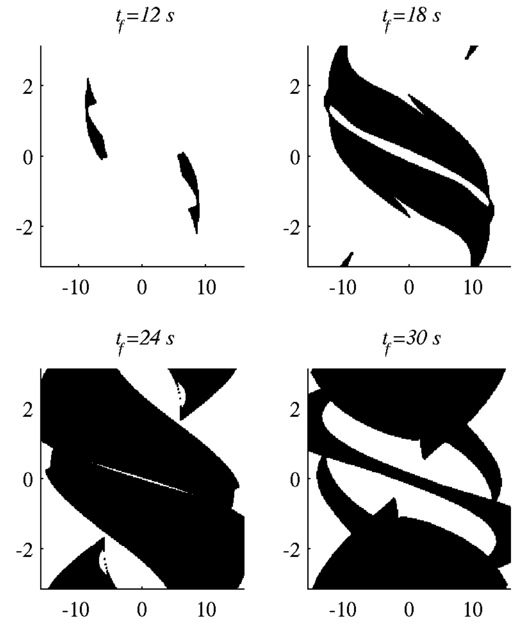

Figure 4. Projections of R_4 for different time horizons.

References

Anta, A. and Tabuada, P. (2010). To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030 –2042.

Anta, A. and Tabuada, P. (2012). Exploiting isochrony in self-triggered control. *Automatic Control, IEEE Transactions on*, 57(4):950–962.

Bokanowski, O., Cristiani, E., and Zidani, H. (2010). An efficient data structure and accurate scheme to solve front propagation problems. *J. Sci. Comput.*, 42:251–273.

Clarke, F. H., Ledyaev, Y. S., Sontag, E. D., and Subbotin, A. I. (1997). Asymptotic controllability implies feedback stabilization. *IEEE Transactions on Automatic Control*, 42(10):1394–1407.

Fiter, C., Hetel, L., Perruquetti, W., and Richard, J.-P. (2012). A state dependent sampling for linear state feedback. *Automatica*, 48(8):1860 – 1867.

Fiter, C., Hetel, L., Perruquetti, W., and Richard, J.-P. (2015). A robust stability framework for {LTI} systems with time-varying sampling. *Automatica*, 54(0):56 – 64.

Gommans, T., Antunes, D., Donkers, T., Tabuada,

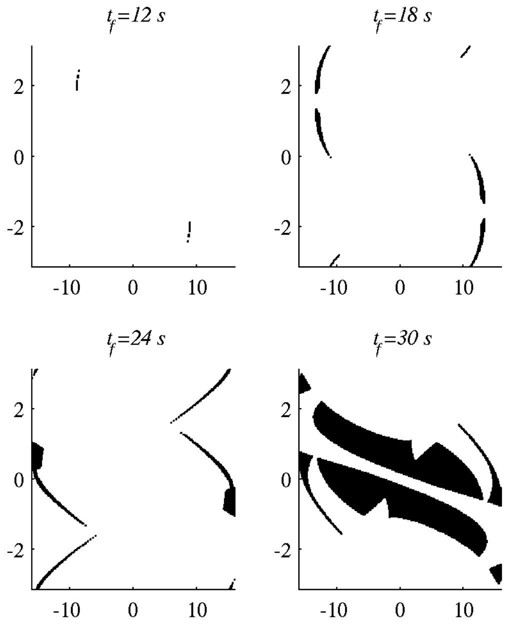

Figure 5. Projections of R_5 for different time horizons.

P., and Heemels, M. (2014). Self-triggered linear quadratic control. *Automatica*, 50(4):1279 – 1287.

Hespanha, J., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162.

Krasovskii, N. and Subbotin, A. (1988). *Game-theoretical control problems*. Springer-Verlag, New York.

Kurzhanski, A., Mitchell, I., and Varaiya, P. (2006). Ellipsoidal techniques for hybrid dynamics: the reachability problem. *Journal of Optimization Theory and Applications*, pages 499–521.

Mahmoud, M. S. and Memon, A. M. (2015). Aperiodic triggering mechanisms for networked control systems. *Information Sciences*, 296(0):282 – 306.

Mitchell, I. M. (2008). The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35(2-3):300–329.

Wang, X. and Lemmon, M. (2009). Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability. *Automatic Control, IEEE Transactions on*, 54(3):452–467.

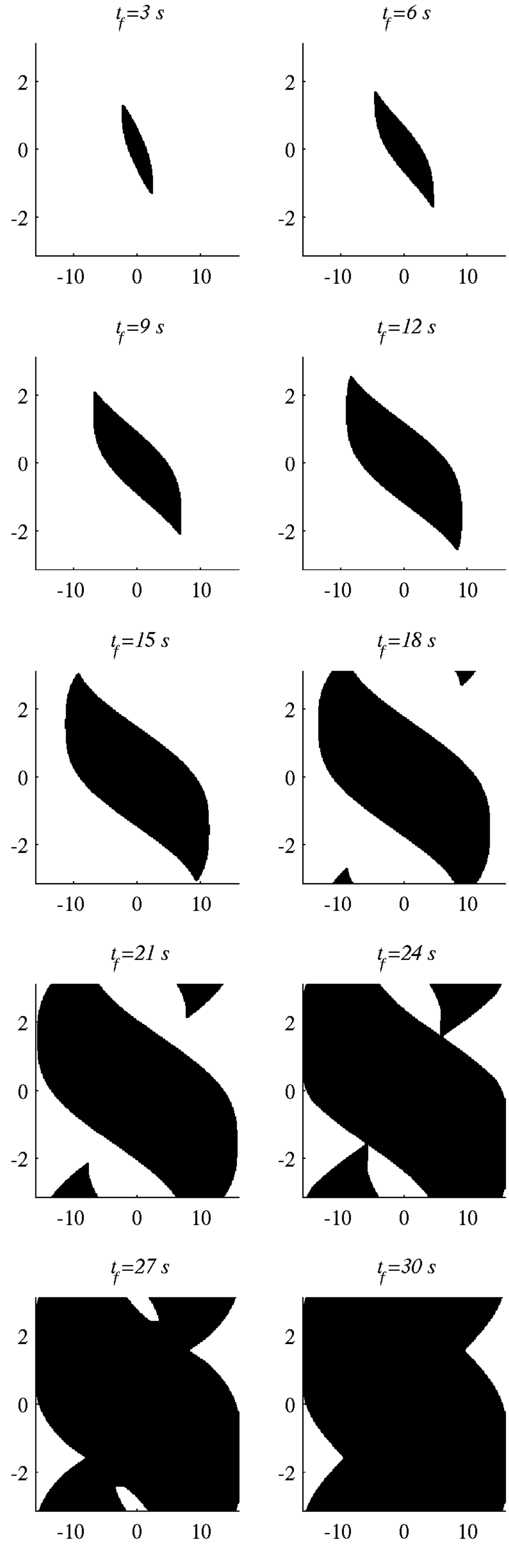

Figure 6. Projections of R_0^{56} for different time horizons.