Nonlinear Event-triggered Control and Self-triggered Control for Linear Systems

Fairouz Zobiri, Nacim Meslem, Brigitte Bidégaray-Fesquet

Univ. Grenoble Alpes, Laboratoire Jean Kuntzmann and GIPSA-lab, Grenoble, France

Recap

Recall that in the case of a linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$

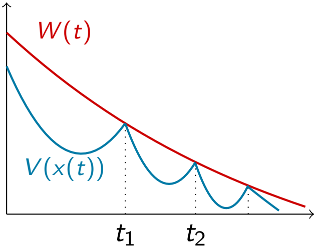
u is the event-triggered control law

$$u(t_k) = -Kx(t_k), \quad t = t_k, \; k \in \mathbb{N}$$
$$u(t) = u(t_k)), \quad t \in [t_k, t_{k+1}).$$

The time instants t_k are defined as

$$t_k = \{t > t_{k-1} \mid V(x(t)) \geq W(t)\}.$$

In the linear case: $V(x(t)) = x(t)^T P x(t)$, and $W(t) = W_k e^{-\alpha(t-t_k)}$, or a constant threshold.



Nonlinear Case

In the case of the nonlinear system of the form

$$\dot{x}(t) = f(x, u).$$

- No standard form for the Lyapunov function exists,
- finding a Lyapunov function can be an arduous task,
- event-triggered control algorithm cannot be described in details.

Nonlinear Case

In the case of the nonlinear system of the form

$$\dot{x}(t) = f(x, u).$$

- No standard form for the Lyapunov function exists,
- finding a Lyapunov function can be an arduous task,
- event-triggered control algorithm cannot be described in details.

Contraction Analysis!

Contraction Analysis Overview

Consider the system

$$\dot{x} = f(x). \tag{1}$$

Consider two neighboring trajectories. δx is the virtual displacement between them. A contraction region is defined as a region where

$$\frac{1}{2}\delta x^T \left(\frac{\partial f}{\partial x}^T + \frac{\partial f}{\partial x} \right) \delta x \leq -\beta \delta x^T \delta x.$$

A trajectory that starts in a contraction region remains there, and the system is exponentially stable.

Not always the case!

Contraction Analysis Overview

Operate the following coordinate change

$$\delta z = \Theta \delta x.$$

Where $\Theta(x)$ is a square matrix. In the new coordinate system

$$\frac{d}{dt}\delta z = \left(\dot{\Theta} + \Theta \frac{\partial f}{\partial x}\right) \Theta^{-1} \delta z = F \delta z$$

A contraction region is then defined as

$$\delta x^T \left(\frac{\partial f}{\partial x}^T M + \dot{M} + M \frac{\partial f}{\partial x} \right) \delta x \leq -\beta_m \delta x^T M \delta x.$$

where $\beta_m > 0$ and $M(x) = \Theta^T \Theta$ is a positive definite square matrix.

Nonlinear Event-Triggered Control

Consider the system

$$\dot{x} = f(x) + g(x)u,$$

where u is an event-triggered control law, and x_r is a reference trajectory that can be generated by

$$\dot{x}_r = f(x_r) + g(x_r)u_r, \tag{2}$$

where u_r is a continuous-time control law.
Let δx be the virtual displacement between the two trajectories

$$\delta x = x - x_r.$$

If the two trajectories are close enough, the virtual velocity is approximated by

$$\delta\dot{x} \approx A(x)\delta x + g(x)\delta u, \quad A(x) = \frac{\partial f}{\partial x}. \tag{3}$$

where $\delta u = -K(x)\delta x$.

Nonlinear Event-Triggered Control

The generalized Jacobian is $F = \left(\dot{\Theta} + \Theta(A(x) - Kg(x))\right)\Theta^{-1}$.
The control law is taken as

$$u = \int_{x_r}^{x} -K(\chi)\, d\chi.$$

The event-triggering conditions ensure that the trajectory remains inside a region of contraction, and the next sampling instant

$$t_k = \{t > t_{k-1} \mid \delta x^T \left((A - gK)^T M + \dot{M} + M(A - gK)\right)\delta x \geq 0\}.$$

Numerical Example

Consider the second order system

$$\dot{x}_1 = x_2 + \sin x_1,$$
$$\dot{x}_2 = x_1^2 + u,$$
$$x_0 = \begin{bmatrix} 1.2 & 1 \end{bmatrix}^T.$$

We take

$$F = \begin{bmatrix} -2 & 1 \\ 1 & -3 \end{bmatrix}, \Theta = \begin{bmatrix} 1 & 0 \\ \cos x_1 + 2 & 1 \end{bmatrix}, M = \begin{bmatrix} (\cos x_1 + 2)^2 + 1 & \cos x_1 + 2 \\ \cos x_1 + 2 & 1 \end{bmatrix}$$

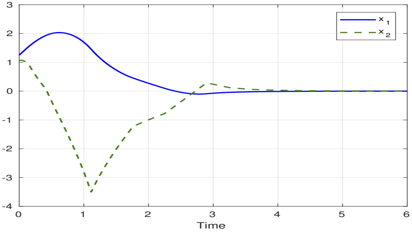
$$K = -\begin{bmatrix} -2x_1 + x_2 \sin x_1 - 2\cos^2 x_1 - 3\cos x_1, & -3 - \cos x_1 \end{bmatrix}.$$

And we select for the reference trajectory

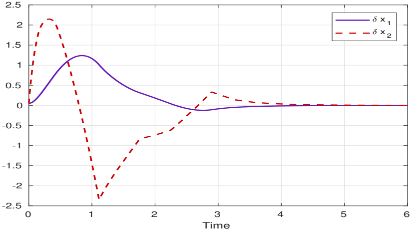
$$u_r = -x_1^2 - (x_2 + \sin x_1)\cos x_1 - k_1 x_1 - k_2(x_2 + \sin x_1),$$

where $k_1 = 6$, $k_2 = 5$ and $x_{r0} = [1.25\ 1.05]^T$.

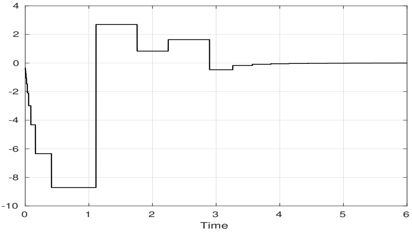
Numerical Example



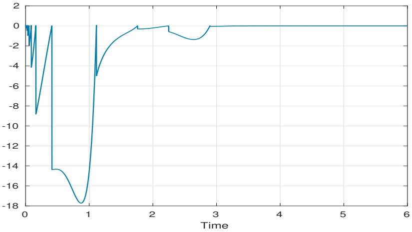
(a) States of the system



(b) Virtual displacement



(c) Control signal



(d) ETC

Self-Triggered Control

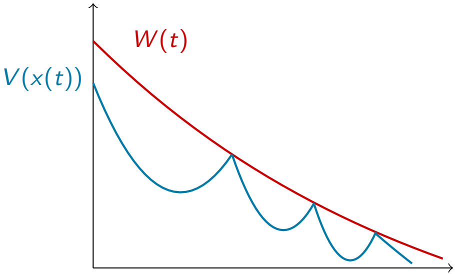
Self-Triggered Control

Recall that in the case of a linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$x(t_0) = x_0.$$

$$u(t_k) = -Kx(t_k), \quad t \in [t_k, t_{k+1}).$$

In self-triggered control, t_{k+1} is computed at t_k.



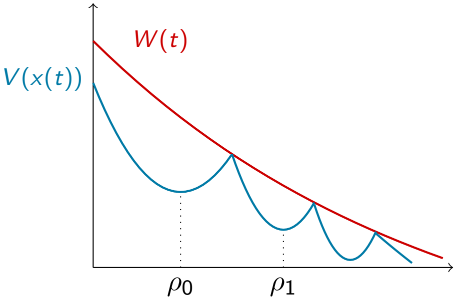
Self-Triggered Control

Recall that in the case of a linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$x(t_0) = x_0.$$

$$u(t_k) = -Kx(t_k), \quad t \in [t_k, t_{k+1}).$$

In self-triggered control, t_{k+1} is computed at t_k.



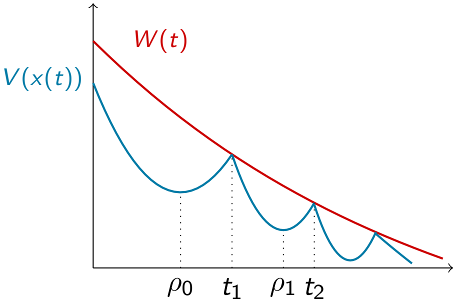
Self-Triggered Control

Recall that in the case of a linear system

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$x(t_0) = x_0.$$

$$u(t_k) = -Kx(t_k), \quad t \in [t_k, t_{k+1}).$$

In self-triggered control, t_{k+1} is computed at t_k.



Self-Triggered Control

The self-triggered algorithm is divided into two stages

- look for ρ_k, the minimizer of $V(x(t))$ on the interval $[t_k, t_{k+1})$.
- use ρ_k as a starting point to solve $W(t) - V(x(t)) = 0$.

We refer to the two stages as the minimization stage and the root-finding stage.

Finding ρ_k first offers many advantages

- it gives us an idea about the length of the interval $[t_k, t_{k+1})$,
- it is close to t_{k+1}, allows us to use a locally convergent root-finding algorithm for fast convergence

Minimization Stage

$$\dot{x}(t) = (A - BK)x(t) + BKe_k(t), \quad e_k(t) = x(t) - x_k(t).$$

Re-write the system's dynamics as

$$\begin{aligned} \dot{\xi}(t) &= \begin{bmatrix} A - BK & BK \\ A - BK & BK \end{bmatrix} \xi(t), \quad \xi(t) = \begin{bmatrix} x(t) & e_k(t) \end{bmatrix}^T, \\ \xi(t_0) &= \begin{bmatrix} x_0 & 0_n^T \end{bmatrix}^T =: \xi_0, \end{aligned} \tag{4}$$

We minimize the function $V(\xi(t)) = \xi(t)^T \mathcal{P} \xi(t)$, where

$$\mathcal{P} = \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix}$$

The minimization algorithm is a modified Newton algorithm, that takes t_k as the initial guess, modifies the search direction whenever it is not a direction of descent and performs a line search to scale the step size.

Root-finding Stage

The root-finding algorithm is a hybrid Newton-bisection algorithm.

The root is located within an interval of length $\kappa(\rho_k - t_k)$, $\kappa > 1$.

A Newton step is taken first, if the new iterate is outside of the interval or too close to the borders, the Newton iterate is rejected and a bisection step is taken instead.

Numerical Example

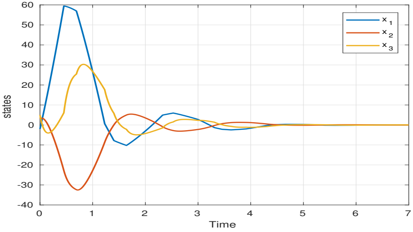
$$\dot{x}(t) = \begin{bmatrix} 1 & 1 & 0 \\ -2 & 0 & 4 \\ 5 & 4 & -7 \end{bmatrix} x(t) + \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} u(t),$$

with initial state $x_0 = [-2\ 3\ 5]^T$. Let $K = \begin{bmatrix} 8.38 & 26.36 & 10.38 \end{bmatrix}$, and

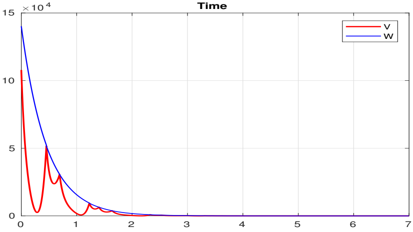
$$V(x(t)) = x(t)^T \begin{bmatrix} 275.7 & 1025.5 & 577.9 \\ 1025.5 & 3840.1 & 2173.5 \\ 577.9 & 2173.5 & 1234.1 \end{bmatrix} x(t), \qquad (5)$$

$$W(t_0) = 1.3\ V(x_0),\ \alpha = 2.18.$$

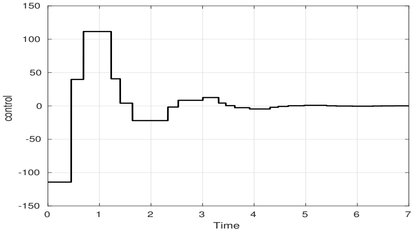
Numerical Example



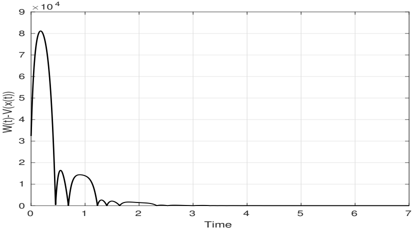
(a) States



(b) PLF and threshold



(c) Self-triggered control



(d) The difference $W(t) - V(x(t))$