

# Event-Based Set-point Tracking Based on a State Observer

Brigitte Bidegaray-Fesquet\*, Nacim Meslem<sup>†</sup>, and Fairouz Zobiri<sup>§¶</sup>

\*Univ. Grenoble Alpes, CNRS, Grenoble INP<sup>‡</sup>, LJK, 38000 Grenoble, France

<sup>†</sup>Univ. Grenoble Alpes, CNRS, Grenoble INP<sup>‡</sup>, GIPSA-lab, 38000 Grenoble, France

<sup>§</sup>KU Leuven, 10 Kasteelpark Arenberg, 3001, Heverlee, Belgium

<sup>¶</sup>Energyville, Thor Park, 3600, Genk, Belgium

Emails: Brigitte.Bidegaray@univ-grenoble-alpes.fr,

Nacim.Meslem@gipsa-lab.grenoble-inp.fr,

Fairouz.Zobiri@kuleuven.be

<sup>‡</sup> Institute of Engineering Univ. Grenoble Alpes

**Abstract**—We propose an event-based framework for set-point tracking involving three linear time invariant systems, a real, a reference and an observer system. To be actually event-based using only the knowledge of the output of the real system, we show that the two systems (reference and observer) involved in the event generator have to take into account the output observation error. The resulting setup is proved to be able to follow a Lipschitz predefined set-point trajectory with no Zeno phenomenon.

## I. INTRODUCTION

Event-based control is an alternative to continuous or periodic control, that enables to address the control of embedded systems with low resources [3], [7], [14], [15]. In this paradigm, the infrequent control updates are suitable for low power systems that spare the energy used in actuators, or to relax the communication requirements in embedded and networked systems.

The stabilization problem of continuous-time systems with event-triggered controllers has been widely studied in the literature. For example, the case of linear systems is considered in [6], [7], [10], [20], the case of nonlinear systems is addressed in [2], [9], [12], [13], [15], and the case of switched linear systems is studied in [8], [16], [17], [19]. Note that, the event-triggering conditions of all the aforementioned approaches are based on the knowledge of the whole state vector of the system at all times. Thus, to overcome this constraint, event-triggered stabilizing controllers based on output feedback are proposed in [1], [4] and on estimated state feedback are introduced in [6], [11]. In [6] the experiential stability problem of a specific closed-loop interconnection between a continuous-time linear system and a continuous-time linear controller via a digital channel is considered. The global asymptotic stability of the plant is guaranteed when an estimate of its state is provided by a continuous-time Luenberger observer. The control is updated when a Lyapunov function using an extended system involving both the estimated state and the control error is not sufficiently decreasing or when the error becomes large with respect to the estimated state. This type of triggering condition is also found in the pioneering work [15]. A non-zero dwelling time is also enforced with a timer. In [11] a combination of a stable scalar reference system and a classical state observer is used to design a new event-triggering condition. This condition is based on

a comparison between the Lyapunov function of the reference system and a quadratic function of the estimated state.

On the other hand, the set-point tracking problem with event-triggered policy is tackled in [18]. It is shown that, for continuous-time linear systems, it is possible for the system outputs to track reference signals even if the control signals are not updated continuously. Indeed the control values are updated only when the behavior of the system is no longer satisfactory. To assess the behavior a condition involving the full state of the system is used. This is not very realistic for practical applications and for two reasons: first the full state of a system is usually not measurable; second, the event generator is supposed to have access to the state for all times, which would induce exorbitant communication costs. However, in the case when the system is observable, it is possible to estimate the state of a system using a state observer. Thus, in this present work, we show how to extend the event-triggered algorithm presented in [18] to the case of estimated state feedback.

## II. A SIMPLE OBSERVER IMPLEMENTATION

The aim of this section is to show that a direct extension of our previous results, on the design of event-triggered controllers, to the case of estimated state feedback is not always possible. Thus, in the next section, we propose a new structure of the reference system to overcome this issue.

### A. Event-triggering based on measured state feedback

Consider the LTI system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \quad (1)$$

for which we want the output vector  $y(t) \in \mathbb{R}^m$  to follow a predefined reference vector  $r(t) \in \mathbb{R}^m$ . In this system, the state vector  $x(t)$  belongs to  $\mathbb{R}^n$  for all  $t \geq 0$ , and the control signal  $u(t) \in \mathbb{R}^p$ . Therefore the state matrix  $A$ , input matrix  $B$  and observation matrix  $C$  respectively belong to  $\mathcal{M}_{n \times n}(\mathbb{R})$ ,  $\mathcal{M}_{n \times p}(\mathbb{R})$ , and  $\mathcal{M}_{m \times n}(\mathbb{R})$ .

In [18] an event-based tracking strategy is investigated, by using a piecewise constant control signal  $\bar{u}(t)$  where for  $t \in$

$[t_k, t_{k+1})$

$$\bar{u}(t) = -Kx(t_k) + Gr(t_k) \equiv \bar{u}_k,$$

and the  $t_k$ ,  $k \in \mathbb{N}$ , are the control update times. They correspond to time instants when an event-triggering condition is fulfilled. This condition involves the comparison of the state  $x(t)$ , at each time instant, to the state of a reference system

$$\begin{cases} \dot{x}_r(t) = (A - BK)x_r(t) + BGr(t), \\ y_r(t) = Cx_r(t). \end{cases} \quad (2)$$

Under the assumption that the pair  $(A, B)$  is controllable and the pair  $(A, C)$  is observable, the feedback matrix  $K \in \mathcal{M}_{p \times n}(\mathbb{R})$  is constructed for both systems such that  $A - BK$  is Hurwitz with prescribed eigenvalues. Defining the gain matrix  $G \in \mathcal{M}_{p \times m}(\mathbb{R})$  by

$$G = -(C(A - BK)^{-1}B)^{-1},$$

we know that

$$\lim_{t \rightarrow \infty} \|y_r(t) - r(t)\| = 0.$$

The strategy defined in [18] is proved to ensure that the tracking error  $x(t) - x_r(t)$  remains small (i.e.  $\|x(t) - x_r(t)\| \leq \varepsilon$ ) so that  $y(t)$  is also close to the reference  $r(t)$  at order  $\varepsilon$  as time goes to infinity.

However, the full state vector is not always available for measurement, which makes unworkable an event-triggering condition based on the full state. A possible solution to this problem is to estimate the value of the state through a state observer.

### B. Construction of a state observer

We therefore first suggest to implement a state observer as

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + B\hat{u}(t) + L(y(t) - \hat{y}(t)), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \quad (3)$$

where the observer gain  $L \in \mathcal{M}_{n \times m}(\mathbb{R})$  is chosen such that  $A - LC$  is Hurwitz. In this system, the control is also piecewise constant and for  $t \in [t_k, t_{k+1})$ ,  $\hat{u}(t) = -K\hat{x}(t_k) + Gr(t_k)$  is computed using the observed state. This control is sent and used by the real system. The tracking error which is used to define the control update instants  $t_k$  is now taken as the difference between the observed state and the reference state

$$\hat{e}(t) = \hat{x}(t) - x_r(t).$$

The event-triggering condition is constructed from this error  $\hat{e}$  as it is done in [18] from  $x - x_r$ . Namely a pseudo-Lyapunov function is constructed

$$V(\hat{e}(t)) = \hat{e}(t)^T P \hat{e}(t),$$

where  $P$  is a positive definite matrix that satisfies the Lyapunov equation

$$(A - BK)^T P + P(A - BK) = -Q,$$

and  $Q$  is a positive definite matrix. A new control is computed and applied when the following event occurs  $V(\hat{e}(t)) = \delta$ . More precisely for a predefined  $\delta > 0$

$$t_k = \inf\{t \geq t_{k-1} \text{ such that } V(\hat{e}(t)) \geq \delta\}. \quad (4)$$

The construction of matrix  $P$  and functional  $V$  is quite classical. However we call  $V$  only a pseudo-Lyapunov function because the update condition in (4) only ensures that  $V$  remains below a threshold. This will be later seen in Figure 4 which shows a typical behavior of  $V$  for such an event-based update condition.

Figure 1 summarizes the communications between the systems (1)–(3), showing clearly that only the output of the real system is used by the two other systems. The dashed lines represent communications that only occur at control update times  $t_k$ .

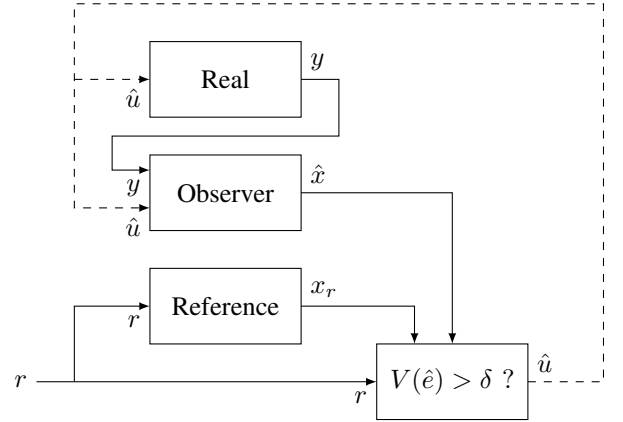


Fig. 1. Communications between the real, reference and observer systems (1)–(2)–(3). Dashed lines are event-triggered communications.

### C. Event-triggering property

The previous tracking implementation is valid if it fulfills the following conditions

- it manages to track the reference trajectory,
- there is no Zeno phenomenon, i.e. the control updates are really discrete in time.

To ensure this, it is necessary that when  $V(\hat{e})$  reaches the threshold  $\delta$  and the control is updated, the function  $V(\hat{e}(t))$  is pushed below the threshold, that is its time derivative is negative. Computing this derivative at time  $t_k$  we obtain

$$\left. \frac{dV(\hat{e}(t))}{dt} \right|_{t=t_k} = -\hat{e}(t_k)^T Q \hat{e}(t_k) + 2\hat{e}(t_k)^T PL(y(t_k) - \hat{y}(t_k)).$$

The output observation error  $y(t) - \hat{y}(t)$  is generally nonzero, and may be relatively large in transient time. Therefore we cannot ensure that the time derivative of the Lyapunov function is non positive at the update time  $t_k$ . Let us show on an example that such a situation can occur.

### D. Impact of the output observer error on the event-triggering strategy

To show that the above implementation is not valid as an event-triggered approach, we use a simple stabilization example from the literature (see e.g. [15]), i.e. we want to track the zero function. To this aim we consider the system

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u, \quad (5)$$

with  $x(0) = (-1 \ 3)^T$ . The feedback gain of the controller is  $K = (0 \ 6)$  and the observer gain  $L = (7.718 \ 27.789)^T$ . Matrices  $P$  and  $Q$  satisfying the Lyapunov equation are

$$P = \begin{pmatrix} 2.5 & 0.5 \\ 0.5 & 2.5 \end{pmatrix} \text{ and } Q = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix},$$

and we take the same initial condition for the observer and the reference system  $\hat{x}(0) = x_r(0) = (10 \cos(\frac{2\pi}{30}), 10 \sin(\frac{2\pi}{30}))^T$ .

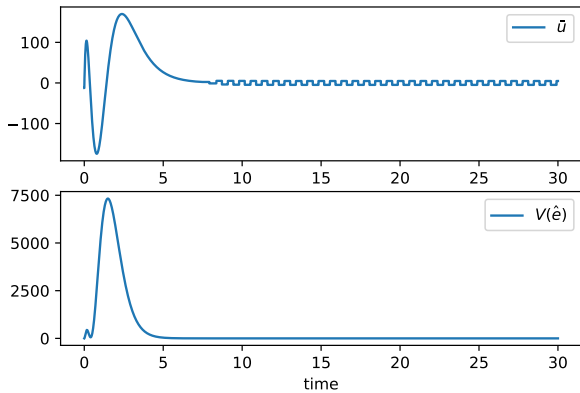


Fig. 2. The control signal (above) and the Lyapunov function for system (5) and control tracking implementation (1)–(3)

Though the control manages to stabilize the system and drive the states to zero, it is updated at every sampling instant in transient time, contrary to the requirements of an event-based scheme. Figure 2 shows the control signal and the Lyapunov function. We notice that there is first a transient part during which the Lyapunov function is much too large and therefore the update condition is always fulfilled, leading to a continuous control. This destroys the performance of the system even if it eventually shows event-based features at permanent regime.

### III. A NEW APPROACH

#### A. Redefining the reference system

To get around the problem, we would like to simply have

$$\left. \frac{dV(\hat{e}(t))}{dt} \right|_{t=t_k} = -\hat{e}(t_k)^T Q \hat{e}(t_k). \quad (6)$$

We can perform this without changing the event-triggering conditions, nor the observer system, but only the reference system. Indeed we will also feed the reference system with the output observation error, leading to the modified dynamics

$$\begin{cases} \dot{x}_r(t) = (A - BK)x_r(t) + BGr(t) + L(y(t) - \hat{y}(t)), \\ y_r(t) = Cx_r(t). \end{cases} \quad (7)$$

We now consider the systems (1), (3), and (7) to track the reference trajectory  $r$  and show that is qualified as an event-based approach. The modified communication scheme is pictured in Figure 3.

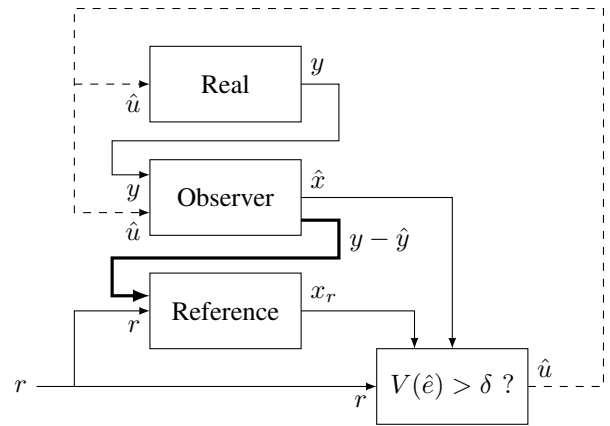


Fig. 3. Communications between the real, reference and observer systems (1)–(7)–(3). Dashed lines are event-triggered communications. The new communication channel is thicker.

#### B. Tracking property

The observation error  $e(t) = x(t) - \hat{x}(t)$  has the following dynamics

$$\dot{e}(t) = (A - LC)e(t),$$

and since  $A - LC$  is Hurwitz, the observation error tends to zero as time goes to infinity:

$$\lim_{t \rightarrow \infty} (x(t) - \hat{x}(t)) = 0. \quad (8)$$

Moreover, by construction of the control update events,  $V(\hat{e}(t)) \leq \delta$ . Indeed, when this condition is violated, the control is updated, and the derivative (6) is negative, pushing  $V(\hat{e}(t))$  below the threshold again. This yields an upper bound for  $\|\hat{e}\|$ : if we denote  $\lambda_{\min}(P)$  the lowest (positive) eigenvalue of  $P$ , for all times

$$\|\hat{x}(t) - x_r(t)\| \leq \sqrt{\frac{\delta}{\lambda_{\min}(P)}} \equiv \varepsilon.$$

Equation (8) also implies that  $y(t) - \hat{y}(t)$  tends to zero as  $t$  goes to infinity. Therefore the dynamics of equation (7) tends to that of  $\dot{x}_r(t) = (A - BK)x_r(t) + BGr(t)$  which is well known to ensure the tracking of  $r(t)$ .

These three limits ensure that the output of the real system tracks  $r(t)$  with a precision of  $\varepsilon$  in the limit  $t \rightarrow \infty$ . In the case of a stability problem, where  $r \equiv 0$ , this ensures that the real system is asymptotically stable with error  $\varepsilon$ .

#### C. Intersample delay

For the algorithm to be event-based, we have also to prove that there exists a minimum delay between two update instants, more precisely,

there exists  $\tau > 0$  such that  $t_{k+1} - t_k > \tau$  for all  $k \in \mathbb{N}$ .

To study this have to consider the derivative of  $\hat{e}$  for  $t \in [t_k, t_{k+1})$ :

$$\dot{\hat{e}}(t) = (A - BK)\hat{e}(t) + B(v(t) - \hat{v}(t_k)),$$

where  $v(t) = -K\hat{x}(t) + Gr(t)$ . This leads to

$$\frac{dV(\hat{e}(t))}{dt} = -\hat{e}(t)^T Q \hat{e}(t) + 2\hat{e}(t)^T P B (v(t) - v(t_k)).$$

The proof follows exactly the same steps as in [18]. A minimum delay  $\tau$  exists if  $r(t)$  is a Lipschitz function, which is the case in real systems for which a low pass filter is used on the reference input to avoid the saturation of actuators.

More precisely, we show that there exists a constant  $\beta$  that only depends on the threshold  $\delta$  and on the eigenvalues of  $P$  and  $Q$  such that  $dV(\hat{e}(t))/dt \leq -\beta$  over a time interval  $[t_k, t_k + \tau_1]$  and  $\tau_1$  depends on the model parameters and the Lipschitz constant of the reference vector. Besides the minimum delay  $\tau_1$  is smaller than the actual dwelling time, since do not ask the pseudo-Lyapunov to decrease but allow it to slightly grow before a new update of the control.

#### D. Test on a stabilization problem

To test this approach, we first use it for the previous stabilization problem. Figure 4 shows the control signal and the time evolution of the Lyapunov function. Here  $V(\hat{e}(t))$  remains below  $\delta = 0.3$  up to small numerical overshoots due to the fact that the numerical discretization does not catch exactly the event times. Although the controls of Figures 2 and 4 seem very similar, on Figure 4 there are only 585 control updates for 300,000 sampling instants (we have set the simulation time step to  $\Delta t = 10^{-4}$ ).

We have chosen  $\Delta t = 10^{-4}$  to satisfy a trade-off between the preservation of dynamics properties of the continuous-time system in the performed numerical simulations and a reasonable computational time. Table I compares both approaches for  $\Delta t = 10^{-4}$ ,  $\Delta t = 10^{-3}$ , and  $\Delta t = 10^{-2}$  in terms of the number of control updates and the maximum of the pseudo-Lyapunov function.

$\Delta t$	$t \in [0, 7.4]$	$t \in [7.4, 30]$	total	$V$ max
First approach (1)-(2)-(3)				
$10^{-4}$	73981	411	74392	7328
$10^{-3}$	7398	235	7633	7342
$10^{-2}$	739	184	923	7482
Second approach (1)-(7)-(3)				
$10^{-4}$	450	135	585	0.302
$10^{-3}$	422	204	626	0.33
$10^{-2}$	234	173	407	0.67

TABLE I. NUMBER OF CONTROL UPDATES AND MAXIMUM VALUES FOR  $V$  FOR BOTH APPROACHES.

We analyze separately two parts in the time evolution of the systems: a transient part (roughly for  $t \in [0, 7.4]$ ) where both strategies show different features, and a steady state part where they are essentially the same. In the transient part, the first strategy is very bad and the updates occur at each simulation time, which means that the control is continuous in time. As can be also seen in Figure 2 the pseudo-Lyapunov function  $V$  takes very large values during this part of the simulation. On the contrary, for the second approach using (7) for the reference system, the number of updates during the transient part is much lower and not directly connected to the number of simulation steps. Choosing a relatively small value of  $\Delta t$  allows to capture the events more precisely. The events are always captured after they occur, and  $\Delta t$  represents the maximum delay in the event detection. This explains that the

maximum value for  $V$  in Table I is always greater than  $\delta = 0.3$ . Detecting events in time leads to a better approximation of the dynamics and end up in a lower number of control updates in the steady state part.

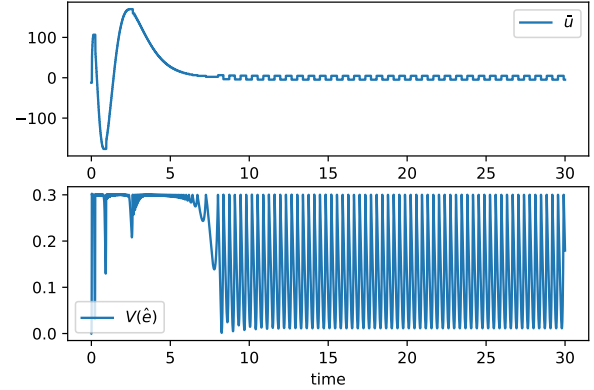


Fig. 4. The control signal (above) and the Lyapunov function for system (5) and control tracking implementation (1), (3), and (7)

Figure 5 shows the time evolution of the states of the real and observed systems. Both systems are stabilized to the precision induced by the choice of  $\delta$ .

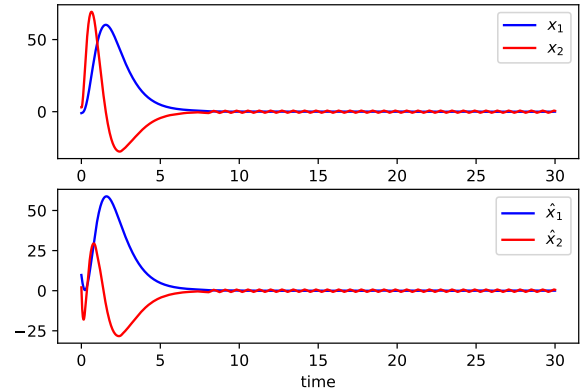


Fig. 5. Time evolution of the real (top) and observed (bottom) states

#### E. A tracking problem

To test tracking we use a control moment gyroscope [5] which is described by a 5-state system, with

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{J_D \Omega}{I_C + I_D} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-J_D \Omega}{I_D + K_A + K_B + K_C} & 0 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_C + I_D} \\ -\frac{1}{J_B + J_C} & 0 \\ 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

and  $K_A = 0.067$ ,  $I_B = 0.012$ ,  $J_B = 0.018$ ,  $K_B = 0.030$ ,  $I_C = 0.0092$ ,  $J_C = 0.023$ ,  $K_C = 0.022$ ,  $I_D = 0.015$ ,  $J_D = 0.027$ ,  $\Omega = 42$ . The gain matrix  $K$  is chosen in order to solve the linear quadratic regulator (LQR) problem which minimizes

$$\int_0^\infty (x^T Q_C x + u^T R u) dt$$

with  $Q_C = 100 C^T C$  and

$$R = \begin{pmatrix} 3 & 0 \\ 0 & 0.02 \end{pmatrix},$$

For this system, the second actuator  $u_2$  is more powerful than the first one  $u_1$ . For this reason we have chosen unbalanced entries in matrix  $R$  in order to have a relatively smooth response  $y$ . Taking for example  $R = I$  would lead to large ripples on  $y_1$ .

The observer gain matrix  $L$  is chosen allowing the online estimation of the state vector solving the problem

$$\text{find } \hat{x}(t) \text{ minimizing } \mathbb{E}[(x(t) - \hat{x}(t))^T (x(t) - \hat{x}(t))],$$

where  $\mathbb{E}(\cdot)$  stands for the mathematical expectation of the square of the estimation error. To do that, a linear quadratic minimization problem design method is applied for which  $W = B B^T$  and  $V = 0.1 I$  are the weighting matrices used to solve the associated Riccati equation.

Finally  $Q = 2 I$  is used to construct the Lyapunov matrix  $P$  and the control update threshold is  $\delta = 10^{-4}$ .

The target reference trajectory is a circle centered at the origin and we choose to let the real system depart from an initial point far off the target, namely from the origin. The time evolution of the two components of the output of the real system is plotted on Figure 6. There is a transient part during which the real system draws near the reference trajectory and then an almost periodic part during which the real system remains near the reference, as can be seen both on Figures 6 and 7. The latter Figure displays the trajectories in the output phase plane ( $y_1, y_2$ ).

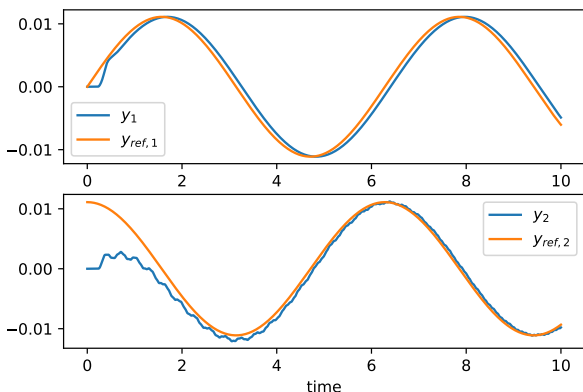


Fig. 6. Time evolution of the two outputs of the real and reference systems

The transient part itself can be split into two. Indeed, at the very beginning, the real system output remains constant. This is due to the fact that matrix  $A$  does not lead to an unstable

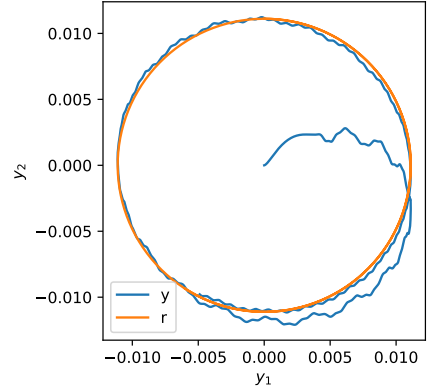


Fig. 7. Trajectory of the outputs in the phase plane

system. It has three zero and two conjugate pure imaginary eigenvalues, and therefore the free dynamics is a nutation. It takes time for the system to destabilize and for the pseudo Lyapunov function to reach  $\delta$ . This is one of the reasons to choose a very small  $\delta$ .

This computation, performed with a time step  $\Delta t = 10^{-3}$ , uses 325 control updates, which means updating the control 3.25% of the computation instants. This medium value for the time step fulfills our needs. A larger time step would mean missing the event times and therefore letting the system diverge too much from its prescribed trajectory. A smaller time step would render the numerical algorithm more stable and would lead to a longer first part of the transient behavior.

#### IV. CONCLUSION

We have been able to design an event-based tracking setup for linear time invariant systems able to follow a Lipschitz predefined reference trajectory. It also avoids the Zeno phenomenon. Its triggering mechanism only needs the knowledge of the measured output of the real system. In a further work we intend to extend this approach to the case when the real system is also subject to process and measurement noises. Our approach also makes use of the same matrices in the real system and the two other computed systems. An interesting question is the impact of model uncertainties on the control performances.

#### ACKNOWLEDGMENT

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-61 LABX-0025-01).

#### REFERENCES

- [1] Mahmoud Abdelrahim, Romain Postoyan, Jamal Daafouz, and Dragan Nešić. *Co-design of output feedback laws and event-triggering conditions for linear systems*, In 53rd IEEE Conf. on Decision and Control (CDC'14), pages 3560–3565, Los Angeles, U.S.A., 2014.
- [2] Adolfo Anta and Paulo Tabuada, *To sample or not to sample: self-triggered control for nonlinear systems*, IEEE Transactions on Automatic Control, 55:2030-2042, 2010.
- [3] Karl Johan Astrom, and Bo Bernhardsson, *Comparison of periodic and event based sampling for first-order stochastic systems*, IFAC Proceedings Volumes, Volume 32, Issue 2, July 1999, Pages 5006-5011.

- [4] M.C.F. Donkers and W.P.M.H. Heemels, *Output-based event-triggered control with guaranteed  $L_\infty$ -gain and improved and decentralized event-triggering*, IEEE Transactions on Automatic Control, 57(6):1362-1376, 2012.
- [5] Educational Control Products, *Control Moment Gyroscope (Model 750)*, available at [www.ecpsystems.com](http://www.ecpsystems.com).
- [6] Fulvio Forni, Sergio Galeani, Dragan Nešić, and Luca Zaccarian, *Event-triggered transmission for linear control over communication channels*, Automatica **50**, 490–498, 2014.
- [7] W.P.M.H. Heemels, M.C.F. Donkers, and Andrew R. Teel, *Periodic event-triggered control for linear systems*, IEEE Transactions on Automatic Control, 58(4):847–861, 2013.
- [8] Tai-Fang Li, Jun Fu, Fang Deng, and Tianyou Chai, *Stabilization of switched linear neutral systems: An event-triggered sampling control scheme*, IEEE Transactions on Automatic Control, vol. 63, no. 10, pp. 3537-3544, Oct. 2018.
- [9] Nicolas Marchand, Sylvain Durand, and Jose Fermi Guerrero-Castellanos, *A general formula for event-based stabilization of nonlinear systems*, IEEE Transactions on Automatic Control, 58(5):1332–1337, 2013.
- [10] Nacim Meslem and Christophe Prieur, *Event-triggered algorithms for continuous-time systems based on reachability analysis*, 52nd IEEE Conference on Decision and Control, pp. 2048-2053, 2013.
- [11] Nacim Meslem and Christophe Prieur, *Event-based stabilizing controller using a state observer*, First International Conference on Event-Based Control, Communications, and Signal Processing. Krakow, Poland: IEEE, Jun. 2015.
- [12] Nacim Meslem and Christophe Prieur, *Event-based controller synthesis by bounding methods*, European Journal of Control, 26, pp. 12-21, 2015.
- [13] Alexandre Seuret, Christophe Prieur, and Nicolas Marchand, *Stability of non-linear systems by means of event-triggered sampling algorithms*, IMA Journal of Mathematical Control and Information, 31(3):415-433, 2014.
- [14] Chen Peng and Fuqiang Li, *A survey on recent advances in event-triggered communication and control*, Information Sciences, vol. 457-458, pp.113–125, Aug. 2018.
- [15] Paulo Tabuada, *Event-triggered real-time scheduling of stabilizing control tasks*, IEEE Transactions on Automatic Control **52** 1680–1685, 2007.
- [16] Weiming Xiang and Taylor T. Johnson, *Event-triggered control for continuous-time switched linear systems*, IET Control Theory & Applications, vol. 11, no. 11, pp. 1694-1703, Jul. 2017.
- [17] Xiaoqing Xiao, Lei Zhou, Daniel W.C. Ho, and Guoping Lu, *Event-triggered control of continuous-time switched linear systems*, IEEE Transactions on Automatic Control, vol. 64, no. 4, pp. 1710-1717, Apr. 2019.
- [18] Fairouz Zobiri, Nacim Meslem, and Brigitte Bidégaray-Fesquet, *Event-based sampling algorithm for setpoint tracking using a state-feedback controller*, Second International Conference on Event-Based Control, Communications, and Signal Processing. Krakow, Poland: IEEE, Jun. 2016.
- [19] Fairouz Zobiri, Nacim Meslem, and Brigitte Bidégaray-Fesquet, *Event-triggered stabilizing controllers for switched linear systems*, Nonlinear Analysis: Hybrid Systems, 36, 100831, 2020.
- [20] Fairouz Zobiri, Nacim Meslem, and Brigitte Bidégaray-Fesquet, *Self-triggered stabilizing controllers for linear continuous-time systems*, International Journal of Robust and Nonlinear Control, 30 (16), pp. 6502-6517, 2020.