

University of Groningen

## Distributed rendez-vous algorithms for a class of cyberphysical systems

De Persis, Claudio; Postoyan, Romain

*Published in:*  
 Proceedings of the European Control Conference

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.**

*Document Version*  
 Early version, also known as pre-print

*Publication date:*  
 2015

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*  
 De Persis, C., & Postoyan, R. (2015). Distributed rendez-vous algorithms for a class of cyberphysical systems. In *Proceedings of the European Control Conference* (pp. 1457-1461). IEEEExplore.

### Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

# Distributed rendez-vous algorithms for a class of cyberphysical systems

Claudio De Persis and Romain Postoyan

**Abstract**— The objective is to design distributed coordination strategies for a network of agents in a cyber-physical environment. In particular, we concentrate on the rendez-vous of agents having double-integrator dynamics with the addition of a damping term in the velocity dynamics. We start with distributed controllers that solve the problem in continuous-time, and we then explain how to implement these using event-based sampling. The idea is to define a triggering rule per edge using a clock variable which only depends on the local variables. The triggering laws are designed to compensate for the perturbative term introduced by the sampling, a technique that reminds of Lyapunov-based control redesign. We first present an event-triggered solution which requires continuous measurement of the relative position and we then explain how to convert it to a self-triggered policy. The latter only requires the measurements of the relative position and velocity at the last transmission instants, which is useful to reduce both the communication and the computation costs. The strategies guarantee the existence of a uniform minimum amount of times between any two edge events. The analysis is carried out using an invariance principle for hybrid systems.

## I. INTRODUCTION

Most coordination algorithms ignore the fact that the agents have limited computation and communication capacities in practice. Nevertheless, these limitations may severely impact the desired convergence property. It is therefore essential to develop control strategies that take these constraints into account in their design. This problem can be addressed via the construction of event-based strategies, see *e.g.* [3], [6], [8], [7], [9], [16]. In that way, each agent updates its control input *only* at a sequence of time instants which depends on the local state variables, and not continuously. In [11], we presented a new type of triggering rules as well as novel proof concepts for the event-based rendez-vous of coupled agents, whose dynamics are modeled by double-integrators with the addition of a damping term in the velocity dynamics. The objective of this paper is to extend our preliminary results in [11] to networks of  $N$  agents, where  $N$  can be strictly bigger than 2. This generalization exhibits nontrivial challenges in terms of modelling, synthesis and analysis as we explain in the following.

Claudio De Persis is with the Faculty of Mathematics and Natural Sciences, University of Groningen, the Netherlands, [c.de.persis@rug.nl](mailto:c.de.persis@rug.nl). His work is partially supported by the Dutch Organization for Scientific Research (NWO) under the auspices of the project QUICK (QUantized Information Control for formation Keeping).

R. Postoyan is with the Université de Lorraine, CRAN, UMR 7039 and the CNRS, CRAN, UMR 7039, France [romain.postoyan@univ-lorraine.fr](mailto:romain.postoyan@univ-lorraine.fr). He is partially supported by the ANR under the grant COMPACS (ANR-13-BS03-0004-02).

We start from distributed controllers proposed in [1] which solve the coordination problem while ignoring the computation and communication constraints. We then design a triggering condition per edge which only relies on the knowledge of the relative position of the two agents under consideration. To do so, we define a clock variable for each edge of the network and the mechanism works as follows. Consider the edge  $\ell$ . After each update, the clock variable  $\phi_\ell$  is reset to a designed constant  $b_\ell$ , and its evolution until the next triggering time is given by the solution to an ordinary differential equation which depends on the relative position associated to the edge  $\ell$ . The next triggering instant occurs when  $\phi_\ell$  is equal to a designed constant  $a_\ell$ . To design the clock dynamics, we start from an energy-like Lyapunov function available for the continuous-time system and add an additional term that takes into account the ‘energy’ associated with the sampling error. We let this extra term depend on the clock variables. We then select the latter in such a way that the overall Lyapunov function computed along the trajectories of the system remains monotonically decreasing despite the sampling. Hence, the triggering rule is the result of a Lyapunov redesign. We stress that, although the vast majority of the results available in event-based control of multi-agent systems is based on Lyapunov analysis and design, to the best of our knowledge this is the first time in the context of event-based control of network systems that the candidate ‘physical’ Lyapunov function is extended to take into account the ‘cyber part’ of the system and gives rise to the triggering rule. Note that the idea to introduce clocks to define the triggering rule is inspired by the work on sampled-data systems in [2], which has been adapted to event-triggered control in [12].

We first assume that the relative position is continuously available, in which case we derive event-triggered control laws. These policies are relevant to limit the actuators wear and to reduce the actuators energy consumption (as the control input is less often updated). Afterwards, we explain how to derive self-triggering rules which only require the knowledge of the relative position and velocity at the last triggering instant. In that way, both the communication and the computation costs are reduced. While in [11] we could exactly recover the relative positions between two events because  $N$  was equal to 2, this is no longer possible for arbitrary values of  $N$ . We overcome this issue by synthesizing estimates of the relative position, which are then used in the design of the self-triggering laws.

The overall system is modelled as a hybrid system using the formalism of [5]. The case where  $N > 2$  induces some technical difficulties due to the need to carefully define

the jump map for the system to satisfy the hybrid basic conditions stated in Chapter 6 in [5] (which are needed in the analysis). We then analyse the stability of the system using an hybrid invariance principle of [5]. The technical developments are more involved than in the case where  $N = 2$ , due to the distributed nature of the problem and the fact that the hybrid systems will be shown to generate solutions which have an average dwell-time (as opposed to a dwell-time in [11]).

There are several contributions to the problem of distributed event-based control (see [3], [16], [6], [8], to name a few, and references therein). What differentiates our contribution from the vast majority of the existing results is that (i) it pursues a Lyapunov-based redesign of the triggering functions and (ii) it adopts a hybrid invariance principle from [5] to infer the results. These two features make the proposed approach general enough to be applicable to other classes of systems and other event-based coordination problems as we show in [10]. This paper focuses on a specific type of agents dynamics in order to facilitate the presentation of the proposed methodology only. Moreover, in this case, we can take advantage of the particular agents dynamics to design the self-triggered controllers. All the proofs are omitted for space reasons and can be found in [10] in a more general setting.

## II. PRELIMINARIES

Let  $\mathbb{R} := (-\infty, \infty)$ ,  $\mathbb{R}_{\geq 0} := [0, \infty)$ ,  $\mathbb{R}_{> 0} := (0, \infty)$ ,  $\mathbb{Z}_{\geq 0} := \{0, 1, 2, \dots\}$ , and  $\mathbb{Z}_{> 0} := \{1, 2, \dots\}$ . For  $(x, y) \in \mathbb{R}^{n+m}$ ,  $(x, y)$  stands for  $[x^T, y^T]^T$ . A function  $\gamma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is of class  $\mathcal{K}$  if it is continuous, zero at zero and strictly increasing and it is of class  $\mathcal{K}_\infty$  if in addition it is unbounded. A set-valued mapping  $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$  is outer semicontinuous if and only if its graph  $\{(x, y) : y \in M(x)\}$  is closed (see Lemma 5.10 in [5]). The notation  $\mathbb{I}$  denotes the identity matrix, and  $\mathbf{1}$  and  $\mathbf{0}$  are respectively the vector composed of 1 and 0 whose dimensions depend on the context. We use  $\text{diag}\{A_1, \dots, A_n\}$  to represent the block-diagonal matrix with square matrices  $A_1, \dots, A_n$  on the diagonal.

We will study hybrid systems of the form below using the formalism of [5]

$$\dot{x} \in F(x) \quad \text{for } x \in C, \quad x^+ \in G(x) \quad \text{for } x \in D, \quad (1)$$

where  $x \in \mathbb{R}^n$  is the state,  $F$  is the flow map,  $G$  is the jump map,  $C$  is the flow set and  $D$  is the jump set. A solution<sup>1</sup>  $\phi$  to (1) is: *maximal* if it cannot be extended; *complete* if its domain  $\text{dom } \phi$  is unbounded; *precompact* if it is complete and the closure of its range is compact, where the range of  $\phi$  is  $\text{rge } \phi := \{y \in \mathbb{R}^n : \exists (t, k) \in \text{dom } \phi \text{ such that } y = \phi(t, k)\}$ . We introduce the following definition to denote hybrid systems which generate solutions that have uniform average dwell-times.

*Definition 1:* The solutions to (1) have a *uniform semiglobal average dwell-time* if for any  $\Delta \geq 0$ , there exist

$\tau(\Delta) > 0$  and  $n_0(\Delta) \in \mathbb{Z}_{> 0}$  such that for any solution  $\phi$  to (1) with  $|\phi(0, 0)| \leq \Delta$

$$k - i \leq \frac{1}{\tau(\Delta)}(t - s) + n_0(\Delta), \quad (2)$$

for any  $(s, i), (t, k) \in \text{dom } \phi$  with  $s + i \leq t + k$ . We say that the solutions to (1) have a *uniform global average dwell-time* when  $\tau$  and  $n_0$  are independent of the ball of initial conditions.  $\square$

We say that a solution  $\phi$  *approaches the set*  $S \subset \mathbb{R}^{n_x}$  ([15]) if for any  $\epsilon > 0$  there exists  $(t^*, k^*) \in \text{dom } \phi$  such that for all  $(t, k) \in \text{dom } \phi$  with  $t + k \geq t^* + k^*$ ,  $\phi(t, k) \in S + \epsilon \mathbb{B}$ , where  $\mathbb{B}$  is the unit ball.

## III. PROBLEM STATEMENT

The objective is to construct distributed controllers to ensure the rendez-vous of networked systems with limited communication and/or computation capacities. In particular, we consider  $N$  agents which are interconnected over a connected<sup>2</sup> undirected graph  $\mathcal{G} = (\mathcal{I}, \mathcal{E})$  where  $\mathcal{I} := \{1, \dots, N\}$  is the set of nodes and  $\mathcal{E}$  is the set of pairs of nodes connected by edges. The dynamics of the nodes is given by

$$\dot{p}_i = v_i, \quad \dot{v}_i = -v_i + u_i, \quad (3)$$

where  $p_i \in \mathbb{R}$  is the position,  $v_i \in \mathbb{R}$  is the velocity,  $u_i \in \mathbb{R}$  is the control input,  $i \in \mathcal{I}$ . This dynamics corresponds to agents modeled as double integrators with the addition of a damping term possibly due to friction. For the sake of simplicity, the agents are supposed to evolve on a line (see Remark 3 below for a discussion on this point). In continuous-time, the control input  $u_i$  is defined as ([1])

$$u_i = \sum_{j \in \mathcal{N}_i} \psi_{ij}(z_{ij}) \quad (4)$$

where  $\mathcal{N}_i$  is the set of neighbours of the node  $i \in \mathcal{I}$ , *i.e.*  $\mathcal{N}_i := \{j \in \mathcal{I} : (i, j) \in \mathcal{E}\}$ , and

$$z_{ij} := p_j - p_i \quad (5)$$

is the relative position of agent  $i$  with respect to agent  $j$ . The functions  $\psi_{ij} : \mathbb{R} \rightarrow \mathbb{R}$ ,  $(i, j) \in \mathcal{E}$ , are designed and are required to be continuously differentiable, nondecreasing and odd, that implies that  $(x - y)(\psi_{ij}(x) - \psi_{ij}(y)) \geq 0$  and  $\psi_{ij}(-x) = -\psi_{ij}(x)$  for  $x, y \in \mathbb{R}$ . These functions are synthesized such that  $\psi_{ij} = \psi_{ji}$  for  $i \in \mathcal{I}$  and  $j \in \mathcal{N}_i$ . According to [1], the controllers in (4) guarantee that all the positions  $p_i$ ,  $i \in \mathcal{I}$ , asymptotically converge towards each other, which means that the rendez-vous is achieved.

In this paper, we take into account the resources limitations of the system in terms of communication and/or computation. Hence, we envision a setting where the agents only receive measurements from their neighbours and/or update their control inputs at some given time instants to be determined. The control input  $u_i$  in (4) becomes, for  $i \in \mathcal{I}$ ,

$$u_i = \sum_{j \in \mathcal{N}_i} \psi_{ij}(\hat{z}_{ij}) \quad (6)$$

<sup>2</sup>A graph is *connected* if, for each pair of nodes  $i, j$ , there exists a path which connects  $i$  and  $j$ , where a path is an ordered list of edges such that the head of each edge is equal to the tail of the following one.

<sup>1</sup>See [5] for the definition of solutions to (1).

where  $\hat{z}_{ij}$  is a sampled version of  $z_{ij}$ , which is locally maintained by agent  $i$ . This variable is held constant between two successive updates, *i.e.*  $\dot{\hat{z}}_{ij} = 0$  and is reset to the actual value of  $z_{ij}$  at the update time instant, which leads to the jump equation

$$\hat{z}_{ij}^+ = z_{ij}. \quad (7)$$

A sequence of update time instants will be generated and assigned to each pair  $(i, j) \in \mathcal{E}$ . These are time instants that are generated at agent  $i$  and that are triggered by measurements relative to neighbor  $j \in \mathcal{N}_i$ . Symmetrically, agent  $j$  will generate update time instants based on measurements relative to  $i$ . The triggering conditions will be such that the events generated by agent  $i$  relative to neighbor  $j$  and by agent  $j$  relative to neighbor  $i$  are the same. For this reason we term these instants as *edge events*. At each event of the edge  $(i, j) \in \mathcal{E}$ , the agents  $i$  and  $j$  communicate with each other and both of them update the sampled variables  $\hat{z}_{ij}$  and  $\hat{z}_{ji}$  according to (7), which leads to an update of the control inputs  $u_i$  and  $u_j$  in view of (6).

Our goal is to define the sequence of edge events in order to save resources while still ensuring the rendezvous. We first propose an event-triggered solution, which works as follows. For any  $i \in \mathcal{I}$ , agent  $i$  knows its relative position with any of its neighbours at any time instant and the corresponding part of the control input is only updated whenever a certain edge-dependent triggering condition is satisfied (see *e.g.* [13], [4], [3]). Afterwards, we explain how to construct self-triggered policies. In this case, for any  $i \in \mathcal{I}$ , agent  $i$  has access to the relative position and velocity of agent  $j \in \mathcal{N}_i$  (by relative velocity, we mean  $v_j - v_i$ ) and updates the corresponding sampled variables only at edge events (see [3], [8], [9]). The next edge event is determined by the values of the relative position and velocity of agents  $i$  and  $j$  at the last transmission. This scheme reduces both the usage of the CPU and of the agents sensors or of the communication channel. It typically generates more edge events compared to event-triggered control but it does not require the continuous measurement of the neighbours relative position.

The proposed strategies ensure the existence of a uniform strictly positive amount of time between two successive events of a given edge. This property is crucial as it prevents arbitrarily close-in-time edge events (and thus Zeno executions), which would exceed the hardware capacities and render the proposed hybrid controllers not realizable. We do tolerate the occurrence of a finite number of simultaneous edge events as in [3], [8], [9]. We assume that the hardware handles this situation by prioritizing the edge events, which typically leads to small-delays in the control input. We do not address the analysis of the effect of these delays in this paper.

#### IV. EVENT-TRIGGERED CONTROL

##### A. Triggering conditions and hybrid model

Consider the agent  $i \in \mathcal{I}$ . To define the events associated with the edge  $(i, j)$  where  $j \in \mathcal{N}_i$ , we introduce a variable

$\phi_{ij} \in \mathbb{R}$ , which we call a *clock*. The idea is to reset  $\phi_{ij}$  to a constant  $b_{ij} > 0$  after each event associated with  $(i, j)$  and to trigger the next one when  $\phi_{ij}$  becomes equal to  $a_{ij} \in [0, b_{ij}]$ . The constants  $a_{ij}$  and  $b_{ij}$  are designed parameters. Between two successive edge events,  $\phi_{ij}$  is given by the solution to the ordinary differential equation below

$$\dot{\phi}_{ij} = -\frac{1}{\sigma_{ij}} \left( 1 + \phi_{ij}^2 (\nabla \psi_{ij}(z_{ij}))^2 \right), \quad (8)$$

where  $\sigma_{ij}$  is a strictly positive constant which will be specified in the following,  $\nabla \psi_{ij}$  represents the gradient of the function  $\psi_{ij}$ , and we recall that  $z_{ij} = p_j - p_i$ . We notice that  $\phi_{ij}$  strictly decreases on flows in view of (8). The length of the inter-event times depends on the choice of the constants  $a_{ij}$  and  $b_{ij}$ . To take  $a_{ij}$  small and  $b_{ij}$  large typically helps enlarging the inter-event time, at the price of a degraded speed of convergence as the evolution of the velocities depends on the sampled control input, see for an illustration the simulation results in Section VI. The clock  $\phi_{ij}$  can be locally implemented on agent  $i$  provided that continuous measurements of  $z_{ij}$  are available, which is assumed to be the case in this section.

*Remark 1:* The clock dynamics (8) descends from the Lyapunov analysis we follow, which is provided in [10]. In [10], we first introduce an energy-like Lyapunov function which is commonly used in the stability study of the networked systems (3), see [1]. During the continuous evolution of (3) under the sampled-data control (6) (see (12) below for a formal description of the overall dynamical system under consideration), extra terms due to the sampling appear in the time derivative of this Lyapunov function along the solutions to the system. These terms disrupt the monotonic decrease of the energy-like Lyapunov function and hence the desired convergence property. To overcome this obstacle, we introduce an additional term in the Lyapunov function that takes into account the ‘energy’ associated with the sampling errors (and to which we refer to as the ‘cyber’ part of the Lyapunov function) and then design the update law, which regulates the sampling, in such a way that the combination of the ‘physical’ and the ‘cyber’ Lyapunov functions does not increase over time.  $\square$

The dynamics of the agent  $i \in \mathcal{I}$  can be described by the hybrid system below, where  $j \in \mathcal{N}_i$ ,

$$\left. \begin{aligned} \dot{p}_i &= v_i \\ \dot{v}_i &= -v_i + \sum_{j \in \mathcal{N}_i} \psi_{ij}(\hat{z}_{ij}) \\ \dot{\hat{z}}_{ij} &= 0 \\ \dot{\phi}_{ij} &= -\frac{1}{\sigma_{ij}} \left( 1 + \phi_{ij}^2 (\nabla \psi_{ij}(z_{ij}))^2 \right) \end{aligned} \right\} \begin{aligned} &\forall j \in \mathcal{N}_i \\ &\phi_{ij} \in [a_{ij}, b_{ij}] \end{aligned}$$

$$\left( \begin{array}{l} \hat{z}_{ij}^+ \\ \phi_{ij}^+ \end{array} \right) = \left\{ \begin{array}{l} \left( \begin{array}{l} z_{ij} \\ b_{ij} \end{array} \right) \phi_{ij} = a_{ij} \\ \left( \begin{array}{l} \hat{z}_{ij} \\ \phi_{ij} \end{array} \right) \phi_{ij} > a_{ij} \end{array} \right\} \exists j \in \mathcal{N}_i \phi_{ij} = a_{ij}. \quad (9)$$

The jump map in (9) means that only the pairs  $(\hat{z}_{ij}, \phi_{ij})$ ,  $j \in \mathcal{N}_i$ , for which  $\phi_{ij}$  is equal to  $a_{ij}$ , are reset to  $(z_{ij}, b_{ij})$ ;



the others remain unchanged. We see that the control input updates are edge-dependent and distributed as desired. In the analysis that follows, it is essential that each agent  $i$  maintains a local sampled version of the measurement  $z_{ij}$ ,  $j \in \mathcal{N}_i$ , which is consistent with the local sampled version of the corresponding quantity  $z_{ji}$  by the agent  $j$ . To be more specific, for  $(i, j) \in \mathcal{E}$ , it must be true that  $\hat{z}_{ij}(t, k) = -\hat{z}_{ji}(t, k)$  for all  $(t, k)$  in the domain of the solution. To guarantee this property, we make the following assumption.

*Assumption 1:* The following hold for any  $(i, j) \in \mathcal{E}$ .

- (i)  $a_{ij} = a_{ji}$ ,  $b_{ij} = b_{ji}$ ,  $\sigma_{ij} = \sigma_{ji}$ .
- (ii) The variables  $\hat{z}_{ij}$  and  $\phi_{ij}$  are respectively initialized at the same values as  $-\hat{z}_{ji}$  and  $\phi_{ji}$ .  $\square$

Item (i) of Assumption 1 introduces no conservatism as the constants  $a_{ij}$ ,  $a_{ji}$ ,  $b_{ij}$ ,  $b_{ji}$ ,  $\sigma_{ij}$ ,  $\sigma_{ji}$  are designed by the agents. Item (ii) of Assumption 1 is convenient for the analysis. When it is not verified, the clocks  $\phi_{ij}$  and  $\phi_{ji}$ ,  $(i, j) \in \mathcal{E}$ , will be different and this will imply that the updates for  $\hat{z}_{ij}$  and  $\hat{z}_{ji}$  will occur at different times and that the two measurements are different. This causes an asymmetry in the control laws of the neighboring agents  $i, j$  that may disrupt the convergence of the algorithms. Robustness of our algorithm to asymmetric initializations is an important open problem.

*Remark 2:* In different scenarios, item (ii) of Assumption 1 may be less critical. In fact, the scenario that was discussed above assumes that when the clock  $\phi_{ij}$  reaches  $a_{ij}$ , the agent updates  $\hat{z}_{ij}$  with the information collected by its sensor. A different scenario could be as follows. Assume that the two clock variables  $\phi_{ij}$  and  $\phi_{ji}$ ,  $(i, j) \in \mathcal{E}$ , are initially different until one of these, say  $\phi_{ij}$ , becomes equal to  $b_{ij}$  (recall that  $b_{ij} = b_{ji}$  in view of item (i) of Assumption 1). At this time instant, we can envision the case in which agent  $i$  (the one whose clock variable has become equal to  $b_{ij}$ ) notifies (without delay) agent  $j$  to update its own clock variable. Hence,  $(\hat{z}_{ij}, \phi_{ij})$  and  $(\hat{z}_{ji}, \phi_{ji})$  are updated respectively to  $(z_{ij}, b_{ij})$  and  $(z_{ji}, b_{ij})$ . In that way, the pairs  $(\phi_{ij}, \hat{z}_{ij})$  and  $(\phi_{ji}, -\hat{z}_{ji})$  are equal for all future times in view of<sup>3</sup> (9) and the convergence results presented hereafter do apply in this case.  $\square$

In view of Assumption 1, we see that we no longer need to distinguish  $\phi_{ij}$  from  $\phi_{ji}$ . We can therefore define a single clock  $\phi_\ell$  instead, where  $\ell$  is the index associated with the edge  $(i, j) \in \mathcal{E}$ . A similar remark applies for the sampled variables  $\hat{z}_{ij}$  and  $\hat{z}_{ji}$  as  $\hat{z}_{ij} = -\hat{z}_{ji}$ . For that purpose, we assign to each edge of  $\mathcal{E}$  an arbitrary direction and we denote by  $M$  the number of edges of the graph  $\mathcal{G}$  which we number. We define the incidence matrix  $D$  of  $\mathcal{G}$  as  $D = [d_{i\ell}]_{(i,\ell) \in \mathcal{I} \times \{1, \dots, M\}}$  with  $d_{i\ell} = 1$  if the node  $i$  is the positive end of the  $\ell^{\text{th}}$  edge,  $d_{i\ell} = -1$  if the node  $i$  is the negative end of the  $\ell^{\text{th}}$  edge, and  $d_{i\ell} = 0$  otherwise. In that

<sup>3</sup>Note that  $(\nabla \psi_{ij}(z_{ij}))^2 = (\nabla \psi_{ji}(z_{ji}))^2$  in (8) as  $z_{ij} = -z_{ji}$  from (5) and  $\psi$  is odd in view of Section III.

way, we define, for the  $\ell^{\text{th}}$  edge corresponding to  $(i, j) \in \mathcal{E}$ ,

$$z_\ell := \begin{cases} z_{ij} & \text{if } j \text{ is the positive end of the edge } \ell \\ z_{ji} & \text{if } i \text{ is the positive end of the edge } \ell, \end{cases}$$

and

$$\hat{z}_\ell := \begin{cases} \hat{z}_{ij} & \text{if } j \text{ is the positive end of the edge } \ell \\ \hat{z}_{ji} & \text{if } i \text{ is the positive end of the edge } \ell. \end{cases}$$

For the  $\ell^{\text{th}}$  edge corresponding to  $(i, j) \in \mathcal{E}$ , we rewrite the dynamics in (8) as

$$\dot{\phi}_\ell = -\frac{1}{\sigma_\ell} \left( 1 + \phi_\ell^2 (\nabla \psi_\ell(z_\ell))^2 \right), \quad (10)$$

where  $\sigma_\ell := \sigma_{ij} = \sigma_{ji}$ ,  $a_\ell := a_{ij} = a_{ji}$  and  $b_\ell := b_{ij} = b_{ji}$  (in view of Assumption 1).

We are not ready yet to present a model of the overall system. Indeed, it appears that the map which defines the jump equation in (9) and which becomes with the notation introduced above, with  $\mathcal{E}_i$  the set of edges connected to agent  $i$ ,

$$\begin{aligned} p_i^+ &= p_i \\ v_i^+ &= v_i \\ \begin{pmatrix} \hat{z}_\ell^+ \\ \phi_\ell^+ \end{pmatrix} &= \begin{cases} \begin{pmatrix} z_\ell \\ b_\ell \end{pmatrix} & \ell \in \mathcal{E}_i \text{ and } \phi_\ell = a_\ell \\ \begin{pmatrix} \hat{z}_\ell \\ \phi_\ell \end{pmatrix} & \ell \in \mathcal{E}_i \text{ and } \phi_\ell > a_\ell, \end{cases} \end{aligned} \quad (11)$$

is not outer semicontinuous. This is an issue because the outer semicontinuity of the jump map is a necessary condition for a hybrid system to be (nominally) well-posed (see Chapter 6 in [5]) which is required to apply the invariance principles of Chapter 8 in [5] we invoke to prove rendez-vous.

To overcome that issue, we redefine the jump map. We use the technique proposed in [14] for that purpose. Instead of doing it for the model of a single agent, we directly do it on the model of the overall system. Hence, we define the concatenated vectors  $p := (p_1, p_2, \dots, p_N) \in \mathbb{R}^N$ ,  $v := (v_1, v_2, \dots, v_N) \in \mathbb{R}^N$ ,  $\phi := (\phi_1, \dots, \phi_M) \in \mathbb{R}^M$ ,  $z := (z_1, \dots, z_M) \in \mathbb{R}^M$ , and  $\hat{z} := (\hat{z}_1, \dots, \hat{z}_M) \in \mathbb{R}^M$ . The system is modeled as follows

$$\left. \begin{aligned} \dot{p} &= v \\ \dot{v} &= -v - D\Psi(\hat{z}) \\ \dot{\hat{z}} &= \mathbf{0} \\ \dot{\phi} &= -\Sigma^{-1} \left( I + \left( \frac{\partial \Psi(z)}{\partial z} \Phi \right)^2 \right) \mathbf{1} \end{aligned} \right\} \forall \ell \phi_\ell \in [a_\ell, b_\ell]$$

$$\left. \begin{aligned} p^+ &= p \\ v^+ &= v \\ \begin{pmatrix} \hat{z}^+ \\ \phi^+ \end{pmatrix} &\in G(z, \hat{z}, \phi) \end{aligned} \right\} \exists \ell \phi_\ell = a_\ell, \quad (12)$$

where  $\Psi(z) := (\psi_1(z_1), \dots, \psi_M(z_M))$ ,  $\Psi(\hat{z}) := (\psi_1(\hat{z}_1), \dots, \psi_M(\hat{z}_M))$ ,  $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_M\}$  and  $\Phi = \text{diag}\{\phi_1, \dots, \phi_M\}$ . Inspired by [14], the set-valued jump map  $G$  is defined as, for  $z, \hat{z}, \phi \in \mathbb{R}^M$ ,

$$G(z, \hat{z}, \phi) := \{G_\ell(z, \hat{z}, \phi) : \ell \in \{1, \dots, M\} \text{ and } \phi_\ell = a_\ell\}, \quad (13)$$

with  $G_\ell(z, \hat{z}, \phi) := (\hat{z}_1, \dots, \hat{z}_{\ell-1}, z_\ell, \hat{z}_{\ell+1}, \dots, \hat{z}_M, \phi_1, \dots, \phi_{\ell-1}, b_\ell, \phi_{\ell+1}, \dots, \phi_M)$  for  $\ell \in \{1, \dots, M\}$ . In that way,

when the clock  $\phi_\ell$  is the only one which is equal to its lower bound  $a_\ell$ , the pair  $(\phi_\ell, \hat{z}_\ell)$  is reset to  $(b_\ell, z_\ell)$ , while the others remain unchanged. In contrast to (11), when several clocks have reached their lower bound, the jump map (13) only allows a single edge to reset its clock and its sampled variable. Consequently, a finite number of jumps successively occurs in this case, until all the concerned edge variables have been updated. A couple of remarks about system (12) need to be added. First, the mapping  $G$  in (13) is defined on  $\mathbb{R}^{3M}$ . When the states are in the jump set its definition is clear from (13), when these are not in the jump set, *i.e.* when  $\phi_\ell \neq a_\ell$  for any  $\ell \in \{1, \dots, M\}$ , it reduces to the empty set. Second,  $G$  is indeed outer semicontinuous as its graph is given by the union of the graphs of the mappings  $G_\ell$ ,  $\ell \in \{1, \dots, M\}$ , which are closed since these mappings are continuous.

### B. Main result

We are ready to state the main result of this section.

*Theorem 1:* Consider system (12) and select the constants  $\sigma_1, \dots, \sigma_M > 0$  such that

$$\max_{\ell \in \mathcal{E}_i} \sigma_\ell \leq \frac{1 - \varepsilon}{2 \deg_i} \quad (14)$$

where  $\deg_i$  is the degree<sup>4</sup> of agent  $i$ ,  $\varepsilon \in (0, 1)$ ,  $i \in \mathcal{I}$ . The solutions have a uniform semiglobal average dwell-time and the maximal solutions are precompact and approach the set  $\{(p, v, \hat{z}, \phi) : z = \mathbf{0}, v = \mathbf{0}, \phi_\ell \in [a_\ell, b_\ell] \text{ for } \ell \in \{1, \dots, M\}\}$ .  $\square$

We see that each agent only needs to know the degree of its neighbours to synthesize its constants  $\sigma_\ell$ . This can be achieved via an initial communication round during which the agents communicate their degrees to their neighbours for instance.

*Remark 3:* We concentrate in this paper on the case where  $p_i, v_i \in \mathbb{R}$  (see (3)). We have verified that the conclusions of Theorem 1 still hold when  $p_i, v_i \in \mathbb{R}^n$  with  $n \in \mathbb{Z}_{>0}$ , in which case we take  $\dot{\phi}_\ell = -\frac{1}{\sigma_\ell} (1 + \phi_\ell^2 \|\nabla \psi_\ell(z_\ell)\|^2)$  for  $\ell \in \{1, \dots, M\}$ , instead of (10), where  $\|\nabla \psi_\ell(z_\ell)\|$  is the induced matrix Euclidean norm of the matrix  $\nabla \psi_\ell(z_\ell)$ .  $\square$

## V. SELF-TRIGGERED CONTROL

The results of the previous section require continuous measurements of the relative positions, which may be difficult to achieve in practice. We explain in this section how to overcome this potential issue by implementing the event-triggering laws in a self-triggered fashion. In that way, measurement of the relative position (and velocity) are only collected at the edge events. The idea consists in replacing  $\nabla \psi_\ell(z)^2$  in (10) by a variable  $\lambda_\ell$  which is obtained based on the values of  $z_\ell$  and the corresponding relative velocity  $w_\ell$  (*i.e.*  $w_\ell = v_j - v_i$  if  $z_\ell = p_j - p_i$ ,  $(i, j) \in \mathcal{E}$ ) at the last event of edge  $\ell \in \{1, \dots, M\}$ .

<sup>4</sup>The degree of a node is the number of edges incident to the node, *i.e.*  $\deg_i$  is equal to the cardinality of  $\mathcal{E}_i$ .

### A. Construction of $\lambda_\ell$

Let  $\ell \in \{1, \dots, M\}$ . An additional property for the function  $\psi_\ell$  must be put in place, namely that there exists a positive number  $\bar{\psi}$  such that

$$|\psi_\ell(z)| \leq \bar{\psi} \quad \forall z \in \mathbb{R}, \quad (15)$$

which is reasonable as the functions  $\psi_\ell$  are designed by the agents.

Assume that  $\ell$  is the edge which links the agents  $i$  and  $j$  and let  $q$  be a solution to (12). Let  $(t_k^\ell, k) \in \text{dom } q$  be such that  $\phi_\ell(t_k^\ell, k) = b_\ell$  and assume that no other edge triggers an event until  $(t_{k+1}^\ell, k)$ . We make this assumption without loss of generality only to simplify the presentation. For almost all  $t \geq t_k^\ell$  with  $(t, k) \in \text{dom } q$ ,

$$\dot{z}_\ell(t, k) = w_\ell(t, k), \quad (16)$$

and, in view of (9) and (15),

$$\underline{w}_\ell(t, k) \leq w_\ell(t, k) \leq \overline{w}_\ell(t, k), \quad (17)$$

where

$$\begin{aligned} \overline{w}_\ell(t, k) &:= \exp(-(t - t_k^\ell)) w_\ell(t_k^\ell, k) + \overline{\Delta}_\ell(t, k) \\ \underline{w}_\ell(t, k) &:= \exp(-(t - t_k^\ell)) w_\ell(t_k^\ell, k) + \underline{\Delta}_\ell(t, k) \end{aligned} \quad (18)$$

with  $\overline{\Delta}_\ell(t, k) := (1 - \exp(-(t - t_k^\ell)))(\deg_i + \deg_j)\bar{\psi}$  and  $\underline{\Delta}_\ell(t, k) := -\overline{\Delta}_\ell(t, k)$ . Consequently, for all  $(t, k) \in \text{dom } q$  with  $t \geq t_k^\ell$ ,

$$\underline{z}_\ell(t, k) \leq z_\ell(t, k) \leq \overline{z}_\ell(t, k), \quad (19)$$

where  $\underline{z}_\ell(t, k) := z_\ell(t_k^\ell, k) + \int_{t_k^\ell}^t \underline{w}_\ell(s, k) ds$  and  $\overline{z}_\ell(t, k) := z_\ell(t_k^\ell, k) + \int_{t_k^\ell}^t \overline{w}_\ell(s, k) ds$ . The estimate of  $(\nabla \psi_\ell(z))^2$  we use to generate the events of edge  $\ell$  is defined as, for  $(t, k) \in \text{dom } q$  with  $t \geq t_k^\ell$ ,

$$\lambda_\ell(t, k) := \max_{\underline{z}_\ell(t, k) \leq z \leq \overline{z}_\ell(t, k)} (\nabla \psi_\ell(z))^2, \quad (20)$$

which is continuous with respect to  $t$ .

*Remark 4:* When we select the functions  $\psi_\ell$  such that  $\nabla \psi_\ell$  is nonincreasing on  $\mathbb{R}_{\geq 0}$  (as it is the case with sigmoid functions for instance), (20) becomes

$$\lambda_\ell(t, k) := \begin{cases} (\nabla \psi_\ell(\underline{z}_\ell(t, k)))^2 & \text{when } \underline{z}_\ell(t, k) > 0 \\ (\nabla \psi_\ell(\overline{z}_\ell(t, k)))^2 & \text{when } \overline{z}_\ell(t, k) < 0 \\ (\nabla \psi_\ell(0))^2 & \text{when } \underline{z}_\ell(t, k) \overline{z}_\ell(t, k) \leq 0. \end{cases} \quad (21)$$

$\square$

### B. Self-triggering rules

To define the events of edge  $\ell \in \{1, \dots, M\}$ , we simply implement the dynamics below instead of (10)

$$\dot{\phi}_\ell = -\frac{1}{\sigma_\ell} (1 + \phi_\ell^2 \lambda_\ell), \quad (22)$$

where  $\lambda_\ell$  is given by (20). This ordinary differential equation can be solved on-line based on the last received measurements. When a closed-form expression of the solution

to (22) can be obtained or when sufficient computational resources can be dedicated to the resolution of (22) when the measurements are received, the proposed self-triggering rules are also useful for scheduling purposes as the agent knows in advance the next instant when it will need to communicate with its neighbours and to compute a new control input.

### C. Hybrid model & analytical guarantees

The overall system is modelled as

$$\left. \begin{array}{l} \dot{p} = v \\ \dot{v} = -v - D\Psi(\hat{z}) \\ \dot{\hat{z}} = \mathbf{0} \\ \dot{\tau} = -\mathbf{1} \\ p^+ = p \\ v^+ = v \\ \left( \begin{array}{l} \hat{z}^+ \\ \tau^+ \end{array} \right) \in \Gamma(z, \hat{z}, \tau) \end{array} \right\} \begin{array}{l} \forall \ell \quad \tau_\ell \geq 0 \\ \\ \exists \ell \quad \tau_\ell = 0, \end{array} \quad (23)$$

where  $\tau := (\tau_1, \dots, \tau_M)$  and  $\tau_\ell, \ell \in \{1, \dots, M\}$ , is a clock used to trigger the events of edge  $\ell$ . The jump map  $\Gamma$  is defined similarly to (13)

$$\Gamma(z, \hat{z}, \tau) := \{\Gamma_\ell(z, \hat{z}, \tau) : \ell \in \{1, \dots, M\} \text{ and } \tau_\ell = 0\}, \quad (24)$$

with  $\Gamma_\ell(z, \hat{z}, \tau) := (\hat{z}_1, \dots, \hat{z}_{\ell-1}, z_\ell, \hat{z}_{\ell+1}, \dots, \hat{z}_M, \tau_1, \dots, \tau_{\ell-1}, T_\ell(a_\ell, b_\ell, z_\ell, w_\ell), \tau_{\ell+1}, \dots, \tau_M)$  for  $\ell \in \{1, \dots, M\}$  where  $T_\ell(a_\ell, b_\ell, z_\ell, w_\ell)$  is the time it takes for the solution to (22) to decrease from  $b_\ell$  to  $a_\ell$  given  $z_\ell$  and  $w_\ell$ .

The following result is obtained by following the same lines as for proving Theorem 1.

*Corollary 1:* Consider system (23) with  $\sigma_1, \dots, \sigma_M > 0$  such that (14) holds. The solutions have a uniform semiglobal average dwell-time and the maximal solutions are precompact and approach the set  $\{(p, v, \hat{z}, \tau) : z = \mathbf{0}, v = \mathbf{0}, \tau_\ell \in [0, \sigma_\ell(b_\ell - a_\ell)] \text{ for } \ell \in \{1, \dots, M\}\}$ .  $\square$

## VI. SIMULATIONS

We have run simulations for a line graph of  $N = 5$  nodes. The consensus protocol has been designed with  $\psi_\ell(z) = 10 \arctan z$  for  $z \in \mathbb{R}$  and  $\ell \in \{1, \dots, 4\}$ . Hence (15) holds with  $\bar{\psi} = 10$ . We have implemented the self-triggering rules developed in Section V with the parameters  $\sigma_\ell = \frac{1-\varepsilon}{2 \deg_i}$ ,  $i \in \{1, \dots, 5\}$ ,  $\ell \in \{1, \dots, 4\}$ ,  $\varepsilon = 1/4$ ,  $a_\ell = 0$ ,  $b_\ell = b$  and different values of  $b$  have been selected. We have simulated the system for 20 initial conditions of  $p$  randomly distributed in  $[1, 2]$ ,  $v(0, 0) = \mathbf{0}$ ,  $\hat{z}(0, 0) = D^T p(0, 0)$ ,  $\phi(0, 0) = b\mathbf{1}$ , with a simulation time of 20s in order to study the influence of  $b$ . Table I provides the obtained average of the total number of edge events, and the average of the time it takes for  $|z|$  to become less than 5% of its initial value, which we denote  $t_{5\%}$ . These results suggest that to increase the value of  $b$  reduces the number of edge events at the price of a longer convergence time. The parameters  $b_\ell$  (equivalently  $a_\ell$ ),  $\ell \in \{1, \dots, 4\}$ , may therefore be adjusted to reduce the communication and computation cost at the price of a slower convergence speed.

	$b = 1$	$b = 10$	$b = 100$
Average # edge events	2276	2101	2107
Average $t_{5\%}$	13.62	15.14	15.86

TABLE I

SIMULATION RESULTS (#: NUMBER OF).

## VII. CONCLUSION

We have extended our preliminary results in [11] to the event-based rendez-vous of a network composed of an arbitrary finite number of agents. We believe that this work demonstrates the potential of the proposed triggering rules and exemplify the interest of casting event-based coordination problems within the hybrid framework of [5].

The considered class of systems is a particular case of systems of the form  $\dot{p}_i = y_i$  and  $\dot{v}_i = f_i(v_i, u_i)$  (with  $p_i \in \mathbb{R}^{n_p}$  and  $v_i \in \mathbb{R}^{n_{v_i}}$ ), where the  $v_i$ -system satisfies a strict-passivity property, with  $i \in \mathcal{I}$ . We show in [10] that the proposed approach extends to such systems as well as to other coordination objectives.

## REFERENCES

- [1] M. Arcak. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, 2007.
- [2] D. Carnevale, A.R. Teel, and D. Nešić. A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems. *IEEE Trans. on Automatic Control*, 52(5):892–897, 2007.
- [3] D.V. Dimarogonas, E. Frazzoli, and K.H. Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, 2012.
- [4] M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels. Minimum attention control for linear systems. *Discrete Event Dynamic Systems*, pages 1–20, 2012.
- [5] R. Goebel, R.G. Sanfelice, and A.R. Teel. *Hybrid dynamical systems*. Princeton, U.S.A., 2012.
- [6] C. Nowzari and J. Cortés. Self-triggered coordination of robotic networks for optimal deployment. *Automatica*, 48(6):1077–1087, 2012.
- [7] C. De Persis. On self-triggered synchronization of linear systems. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems, Koblenz, Germany*, volume 4, pages 247–252, 2013.
- [8] C. De Persis and P. Frasca. Robust self-triggered coordination with ternary controllers. *IEEE Transactions on Automatic Control*, 58(12):3024–3038, 2013.
- [9] C. De Persis, P. Frasca, and J. Hendrickx. Self-triggered rendezvous of gossiping second-order agents. In *IEEE Conference on Decision and Control, Florence, Italy*, 2013.
- [10] C. De Persis and R. Postoyan. A Lyapunov redesign of coordination algorithms for cyberphysical systems. In *arXiv 1404.0576*, 2014.
- [11] C. De Persis and R. Postoyan. Lyapunov design of an event-based control for the rendez-vous of coupled systems. In *International Symposium on Mathematical Theory of Networks and Systems, Groningen, The Netherlands*, 2014.
- [12] R. Postoyan, P. Tabuada, D. Nešić, and A. Anta. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 2015.
- [13] M. Rabi, K.H. Johansson, and M. Johansson. Optimal stopping for event-triggered sensing and actuation. In *IEEE Conference on Decision and Control, Cancun: Mexico*, pages 3607–3612, 2008.
- [14] R.G. Sanfelice, J.J.B. Biemond, N. van de Wouw, and W.P.M.H. Heemels. An embedding approach for the design of state-feedback tracking controllers for references with jumps. *International Journal of Robust and Nonlinear Control*, 2013.
- [15] R.G. Sanfelice, R. Goebel, and A.R. Teel. Invariance principles for hybrid systems with connections to detectability and asymptotic stability. *IEEE Transactions on Automatic Control*, 52(12):2282–2297, 2007.
- [16] G.S. Seyboth, D.V. Dimarogonas, and K.H. Johansson. Event-based broadcasting for multi-agent average consensus. *Automatica*, 49:245–252, 2012.