



HAL
open science

Hybrid control of a three-agent network cluster

Carolina Albea-Sanchez, Alexandre Seuret, Luca Zaccarian

► **To cite this version:**

Carolina Albea-Sanchez, Alexandre Seuret, Luca Zaccarian. Hybrid control of a three-agent network cluster. 53th IEEE Conference on Decision and Control (CDC 2014), Dec 2014, Los Angeles, CA, United States. pp.6. hal-01065145

HAL Id: hal-01065145

<https://hal.science/hal-01065145>

Submitted on 18 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybrid control of a three-agent network cluster

Carolina Albea, Alexandre Seuret and Luca Zaccarian

Abstract—We consider the management problem for a cluster of three identical servers treating an external request. We assume that each agent has a constant computational speed and we build a deterministic hybrid dynamical model of the overall system. Then, inspired by consensus theory, we propose a centralized control law distributing the computational load among the agents and deciding when to turn on or off each agent, for energy efficiency, while ensuring that the queues of each agent asymptotically converge to a desirable level. We prove the properties of the proposed law and highlight some interesting peculiarities in terms of non-uniform convergence.

I. INTRODUCTION

Distribution of computational load across available resources is referred to as the “load balancing problem” in the literature. This problem refers to classical problems of distributing information over a server network as a common parallel computer architecture [1] or balancing energy delivery network architecture for the decentralized hierarchical integration of microgrids [2], for instance. Various taxonomies of load balancing algorithms exist. Assignment may be either deterministic and gradient methods, stochastic, or optimization based. A comparison of several deterministic methods is provided by Willebeek-LeMair and Reeves [3]. Approaches to modeling and static load balancing are given in [4], [5].

In addition, to distribute computational load, one may look at another control loop consisting in activity management of the agent. There are few studies related with this problem, which handles an on-off switch of the network components. Some algorithms have been already proposed to address this problem from the computer science point of view in the case of a sensor network (see for instance [6], [7]). However, these type of algorithms does not consider computational load, stability, robustness with respect to traffic loss, queueing delays and disturbance.

When wanting to suitably describe the dynamics behind the activity of servers that could be turned on or off, a natural approach that comes to mind is that

C. Albea and A. Seuret are with CNRS, LAAS, 7 avenue du Colonel Roche, 31077 Toulouse, France calbea,aseuret@laas.fr

L. Zaccarian is with both CNRS, LAAS, 7 avenue du Colonel Roche, 31077 Toulouse, France and Dipartimento di Ingegneria Industriale, University of Trento, Italy zaccarian@laas.fr

This work supported in part by the ANR project LimCoS contract number 12 BS03 005 01, and by HYCON2 Network of Excellence “Highly-Complex and Networked Control Systems,” grant agreement 257462.

of hybrid dynamical systems, where solutions may continuously flow according to some differential equation (thereby describing the continuous evolution of the network) and may discontinuously jump according to some re-initialization rule that would comprise, for example, the turning on and off of a server. Here we adopt this strategy and we develop our results using the notation and the framework proposed in the recent work [8], [9], wherein a hybrid dynamical systems is described as a set of constraints that a generic solution should satisfy when flowing and when jumping.

This paper proposes a novel strategy for management of the activity in a network. Our approach relies on a relevant dynamic model following deterministic control theory (differently, e.g., from [10] where probabilistic approaches like Markov chains are used), and to the theory of multi-agent systems [11], and the above mentioned theory of hybrid systems [8]. In addition to a load balancing control, this work aims at controlling the activity of agents in a network. A distributed activity management of the agents to adjust the number of them to the total demanded load appears as an innovative and relevant solution.

The paper is organized as follows. In Section II, problem formulation for three agents is stated. Then, a description of the control law based on a consensus algorithm and a hybrid modeling is proposed in Section III. The proof of the main theorem is provided in Section IV. Some simulations are showed in Section V to emphasize the control architecture presented in this article. Finally Section VI draws some conclusions.

Notation: The set of non-negative reals is denoted by $\mathbb{R}_{\geq 0}$. Given a vector w , $|w|$ denotes its Euclidean norm. The notation $\mathbf{1}$ represents the vector of appropriate dimension whose entries are all 1. Given a set \mathcal{D} , $\bar{\mathcal{D}}$ denotes its closure and \mathcal{D}^c denotes its complement.

II. PROBLEM FORMULATION

A. Source and agents

Consider a source \mathcal{S} that distributes the requests rate w among a set of three agents $\mathcal{N} := \{1, 2, 3\}$. The objective of the agents is to perform suitable processing of this processing load.

We represent the information flow through the network via the processing requests coming from the source and the processing service provided by the agents. The agents are assumed to have limited and fixed processing rate denoted as *computational capacity* y .

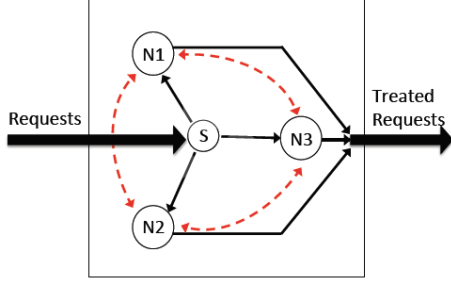


Fig. 1: The considered three-agent cluster.

The core contribution of this paper within the considered scenario is to propose a feedback control scheme for suitably turning on and off each agent, which adjusts the right number of agents to be activated in order to process the amount of requests from the source. For this reason, the following assumption is introduced

Assumption 1: The amount of load rate $\bar{w} \in \mathbb{R}_{>0}$ requested by the source is constant and the processing rate $y \in \mathbb{R}$ satisfies

$$\bar{w} = yn^* = 2y. \quad (1)$$

Assumption 1 means that there exists a number $n^* = 2$ of switched-on agents in the network that can treat exactly the number of requests provided by the sources.

B. Dynamics of the agents: a saturated integrator model

Associate with each agent $i \in \mathcal{N}$ a state variable $\alpha_i \in \{0, 1\}$, whose value indicates whether the agent should be in a quiescent state ($\alpha_i = 0$), typically associated with low consumption, or in an active state ($\alpha_i = 1$) that, as clarified later, corresponds to full information processing rate. We say that an agent $i \in \mathcal{N}$ is *active* if $\alpha_i = 1$ and it is *inactive* if $\alpha_i = 0$. For notational simplicity we introduce the following *network activity vector* $\alpha \in \{0, 1\}^3$:

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix},$$

which is a (discontinuously) time-varying quantity.

To perform suitable processing of the workload coming from source, we equip each agent with a queue which is able to locally store the requests arriving from the source and still not processed. To this aim, we introduce for each $i \in \mathcal{N}$ the state *queue length* $q_i \in \mathbb{R}_{\geq 0}$, whose variation \dot{q}_i corresponds to the difference between the requested rate coming from the source and the maximum processing rate y . Note that special care has to be taken to ensure that any solution $t \mapsto q_i(t)$ to the proposed dynamics satisfies $q_i(t) \in \mathbb{R}_{\geq 0}$ for all $t \geq 0$. Hence, to ensure existence of solutions of the corresponding mathematical model, we need to formulate the dynamics in terms of a hybrid system given below, where we use $x_i = (q_i, \beta_i, \alpha_i)$:

$$\begin{cases} \dot{q}_i = \beta_i \overbrace{(\alpha_i(u_i + \bar{w}/n) - y)}{:=\xi_i}, & (x_i, \xi_i) \in \mathcal{C}_i^0, \quad i \in \mathcal{N} \\ \dot{\beta}_i = 0, \\ q_i^+ = 0, \\ \beta_i^+ = g_\beta(\alpha_i, \xi_i), \end{cases} \quad (x_i, \xi_i) \in \mathcal{D}_i^0, \quad i \in \mathcal{N} \quad (2a)$$

where the flow and jump sets are given by:

$$\begin{aligned} \mathcal{C}_i^0 &= ([0, +\infty) \times \{0, 1\} \times \{0, 1\}) \setminus ([0, +\infty) \times \{0\} \times \{1\}), \\ \mathcal{D}_i^0 &= D_i^{\text{act}} \cup D_i^{\text{empty}}, \\ D_i^{\text{act}} &= \{(x_i, \xi_i) : q_i = 0 \text{ and } \alpha_i \neq \beta_i\} \\ D_i^{\text{empty}} &= \{(x_i, \xi_i) : q_i = 0, \beta_i = 1 \text{ and } \xi_i \leq 0\} \end{aligned} \quad (2b)$$

and the jump map for the activation states (α_i, β_i) is defined as:

$$g_\beta(\alpha_i, \xi_i) = \begin{cases} \alpha_i & \text{if } (x_i, \xi_i) \in D_i^{\text{act}} \\ 0, & \text{if } (x_i, \xi_i) \in D_i^{\text{empty}} \end{cases} \quad (2c)$$

In the selection of the input ξ_i to the i -th agent, \bar{w}/n represents the requests rate coming from the source and n is the number of active agents, which is defined by

$$n = \alpha^T \mathbf{1} = \alpha^T \alpha = \alpha^T \beta. \quad (3)$$

Note that for simplicity the total rate is split in equal parts among the active agents. Finally, $u_i \in \mathbb{R}$ is the control input that will be designed in this paper, as specified in the next section.

According to the hybrid system notation in [8], the continuous-time dynamics equation (2a) should be understood in the sense that solutions may satisfy $\frac{d}{dt}q_i(t, j) = \alpha_i(u_i + \bar{w}/n) - \beta_i y$ for all (t, j) in their domain. As a consequence, if one chooses the unreasonable initial conditions $q_i(0, 0) < 0$, namely $q_i(0, 0) \notin \mathbb{R}_{\geq 0}$ the dynamics constraint (2) implies that the solution cannot be continued because the flow map (that is, the set of allowed flow directions) is empty. Conversely, whenever one selects the reasonable initial condition $q_i(0, 0) \geq 0$, the following lemma ensures the well behavedness of dynamics (2) in terms of existence of complete non-negative solutions (in addition to regularity of the right hand side of (2a)).

Lemma 1: The system data in (2) satisfies the hybrid basic conditions of [8, Ch. 6]. Moreover, if $q_i(0, 0) \in \mathbb{R}_{\geq 0}$, then for any selection of the inputs ξ_i, α_i such that $\xi_i \leq 0$ and $q_i = 0$ only if $\alpha_i = 0$, dynamics (2) has at least one complete solution having unbounded domain in the ordinary time direction.

Proof: The hybrid basic conditions can be easily checked by inspection. Regarding existence of solutions, notice that for all $q_i > 0$ solutions can flow, while for $q_i = 0$, if the input ξ is negative (so that the flow map would point outside of the flow set), then the state belongs to the jump set and may jump to $\beta_i = 0$, from where flow is once again possible. ■

Remark 1: Dynamics (2) can be interpreted as a suitable description of the dynamics of a scalar saturated integrator whose output is q_i . In particular the integrator

is only saturated from below and its lower saturation value is zero. This dynamical description may be easily equipped with an upper saturation limit and corresponds to a desirable outer semicontinuous model of the saturated integrator ensuring existence of solutions from the well-posedness results in [8, Ch. 6]. \square

Despite the desirable result about existence of complete solutions (thereby providing a practically reasonable mathematical model), the hybrid basic conditions ensure that the agents dynamics in (2) is a well-posed hybrid dynamical system (see [8, Ch. 6]). As established in [8, Ch. 5 and 6], well-posedness ensures that no unpredicted solutions can be generated by arbitrarily small perturbations affecting the system's data.

A more realistic server structure dynamical model will have to take into account in future work.

C. Connection graph among agents

According to the activities of the agents $\alpha = [\alpha_1, \alpha_2, \alpha_3]^T$, we classically define the adjacency matrix as

$$A(\alpha) = [a_{ih}(\alpha)] = \begin{cases} \alpha_i \alpha_h & \text{if } i \neq h, \\ 0 & \text{if } i = h. \end{cases} \quad (4)$$

and the diagonal matrix $\Delta(\alpha)$ as

$$\Delta(\alpha) = \begin{cases} 0 & \text{if } i \neq h, \\ \sum_{k \in \mathcal{N}, k \neq i} a_{ik}(\alpha) & \text{if } i = h. \end{cases}$$

Therefore, the Laplacian matrix representing the undirected graph is given by

$$L(\alpha) = \Delta(\alpha) - A(\alpha). \quad (5)$$

The Laplacian of an undirected graph is a symmetric positive semi-definite matrix. It is worth mentioning that the resulting Laplacian $L(\alpha)$ represents the complete graph among the agents that are on (comprising the bold arrows in Figure 1 not insisting on inactive agents). Matrix $L(\alpha)$ satisfies the following useful properties.

Property 1: The following relations hold

$$L(\alpha)\mathbf{1} = L(\alpha)\alpha = 0, \quad \mathbf{1}^T L(\alpha) = \alpha^T L(\alpha) = 0. \quad (6)$$

Proof: The proof of the previous relation are straightforwardly derived from the property of a Laplacian matrix and from the construction of $L(\alpha)$. \blacksquare

D. Control objectives

The goal of this paper is to propose a suitable control law that both coordinates the activation/deactivation of the agents and assigns the inputs u_i to each agent to achieve the following goals along the solutions to the proposed closed-loop :

- The number of active agents satisfies,

$$\lim_{t \rightarrow +\infty} (n - n^*) = 0. \quad (7)$$

- Given a desired queue length $q^* > 0$ and a desired range $\varepsilon > 0$, the queue length q_i , for any $i = 1, 2, 3$, must satisfy

$$\lim_{t \rightarrow +\infty} \alpha_i = 1 \Rightarrow \lim_{t \rightarrow +\infty} q_i \in \left[q^* - \frac{\varepsilon}{2}, q^* + \frac{\varepsilon}{2} \right], \quad (8)$$

$$\lim_{t \rightarrow +\infty} \alpha_i = 0 \Rightarrow \lim_{t \rightarrow +\infty} q_i = 0. \quad (9)$$

- All the requested processing is treated during the transient:

$$\sum_{i \in \mathcal{N}} \dot{q}_i = \bar{w} - \sum_{i \in \mathcal{N}} \beta_i y, \quad \forall t \geq 0. \quad (10)$$

Remark 2: Note that the first two control objectives above essentially correspond to a practical load balancing requirement among the agents that will eventually remain active, together with the requirement that the agents eventually remaining inactive converge to having an empty queue (so that they can be shut off). Condition (7) can be enforced in the simplified setting of this paper because of Assumption 1. The third requirement is a flow constraint on the solutions comprising the fact the sum of the instantaneous difference between the total requested processing rate (\bar{w}) and the actual total processing rate of the agents ($\sum_{i \in \mathcal{N}} \beta_i y$) is losslessly stored within the queues of the agents ($\sum_{i \in \mathcal{N}} \dot{q}_i$). This requirement can be satisfied by a suitable selection of inputs u_i . \square

We may well represent the two first control objectives in terms of an error variable e , which is defined by

$$e = \begin{bmatrix} \alpha^T \tilde{q} \\ \tilde{n} \end{bmatrix}, \quad (11a)$$

$$\text{where } \tilde{n} = n - n^* \text{ and } \tilde{q} = \begin{bmatrix} q_1 - q^* \\ q_2 - q^* \\ q_3 - q^* \end{bmatrix}. \quad (11b)$$

In the next section, we will develop a control scheme based on a consensus algorithm and a hybrid modeling which fulfills these control objectives.

III. PROPOSED CONTROL LAW

A. Load balancing control law

Using model (2), and the interconnection (Laplacian) matrix (5), we may select a dynamic distributed controller, that represents the flux exchanged between agents, corresponding to

$$u_i = -k_p \sum_{h \in \mathcal{N}} a_{ih} (q_h - q_i), \quad \forall i = 1, 2, 3. \quad (12)$$

The coefficient k_p is positive parameter. This control refers to a classical consensus algorithm for simple integrator multi-agent systems [11], [12], which ensures that the length of the queues converge to an agreement. Gathering all the dynamics of the queues in (2), the following equations are derived

$$\dot{q} = u + \alpha \frac{\bar{w}}{n} - \beta y, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = -k_p L(\alpha) q, \quad (13)$$

where $L(\alpha)$ is the Laplacian matrix defined in (5). Equations (13) have been obtained from the entries of $L(\alpha)$ defined in (4), which ensure that the equality $\alpha_i u_i = u_i$ holds for any $i = 1, 2, 3$, and thanks to the fact that, according to dynamics (2), when the queue is empty, the agent will be turned off and we will obtain, in (13), $\dot{q}_i = 0$ until the agent is possibly activated again.

In the following lemmas, we will show two important properties of selection (12), which correspond to a few central features enabling us to meet the control objectives specified in Section II-D.

Lemma 2: The distributed control law (12) satisfies the complete processing condition (10).

Proof: Consider equations (13) and take the sum over all the agents, to get

$$\sum_{i \in \mathcal{N}} \dot{q}_i = \sum_{i \in \mathcal{N}} u_i + \underbrace{\alpha^T \mathbf{1} \frac{\bar{w}}{n}}_{=\bar{w}} - \sum_{i \in \mathcal{N}} \beta_i y,$$

therefore the result follows if:

$$\sum_{i \in \mathcal{N}} u_i = \alpha^T u = -k_p (\alpha^T L(\alpha)) q = 0, \quad (14)$$

which follows from the relation $\alpha^T L(\alpha) = 0$, established in Property 1. ■

Lemma 3: The quantity $\alpha^T q$ represents the sum of the active queues and satisfies

$$\alpha^T \dot{q} = -\tilde{n} y. \quad (15)$$

Proof: From equation (13), we get

$$\alpha^T \dot{q} = -k_p (\alpha^T L(\alpha)) q + (\alpha^T \alpha / n) \bar{w} - \alpha^T \beta y.$$

We may use again $\alpha^T L(\alpha) = 0$ together with the definitions of n in (3) and n^* in (1) to show that

$$\alpha^T \dot{q} = (n^* - n) y = -\tilde{n} y,$$

which ends the proof. ■

B. Hybrid control scheme

This section is dedicated to the introduction of an appropriate model to represent the continuous and discrete-time behavior of the system. Hence, we propose a hybrid modeling, which basically gathers the two dynamics in a single model. A possible solution is described below

$$\begin{aligned} \dot{x}_i &= f_i(x_i, u_i, \bar{w}), & (x_i, e) &\in \mathcal{C}_i, \\ x_i^+ &\in G_i(x_i), & (x_i, e) &\in \mathcal{D}_i, \end{aligned} \quad (16a)$$

for $i = 1, 2, 3$, where for each $i \in \{1, 2, 3\}$,

$$G_i(x_i, e) = \begin{cases} g_i^{\text{on}}(x_i), & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{on}}, \\ g_i^{\text{off}}(x_i), & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{off}} \setminus \mathcal{D}_i^0, \\ g_i^0(x_i), & \text{if } (x_i, e) \in \mathcal{D}_i^0 \setminus \mathcal{D}_i^{\text{off}}, \\ g_i^0(x_i) \cup g_i^{\text{off}}(x_i), & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{off}} \cup \mathcal{D}_i^0, \end{cases} \quad (16b)$$

$$x_i := \begin{bmatrix} q_i \\ \beta_i \\ \alpha_i \end{bmatrix}, \quad f_i(x_i, u_i, \bar{w}) := \begin{bmatrix} \alpha_i(u_i + \bar{w}/n) - \beta_i y \\ 0 \\ 0 \end{bmatrix} \quad (16c)$$

$$g_i^{\text{on}}(x_i) := \begin{bmatrix} q_i \\ 1 \\ 1 \end{bmatrix} \quad g_i^{\text{off}}(x_i) := \begin{bmatrix} q_i \\ 1 \\ 0 \end{bmatrix} \quad g_i^0(x_i) := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (16d)$$

and \mathcal{D}_i^0 is defined in (2b)

$$\begin{aligned} \mathcal{D}_i^{\text{on}} &= \{(x_i, e) : \alpha_i = 0, \phi_{\text{on}}(q_i, e) \geq 0, \alpha^T \tilde{q} + \tilde{n} q^* \geq \varepsilon/2\} \\ \mathcal{D}_i^{\text{off}} &= \{(x_i, e) : \alpha_i = 1, \phi_{\text{off}}(q_i, e) \leq 0, \alpha^T \tilde{q} + \tilde{n} q^* \leq -\varepsilon/2\} \\ \mathcal{C}_i &= \overline{\mathcal{D}_i^e}, \end{aligned} \quad (17)$$

where \mathcal{D}_i^e is the closure of the component of \mathcal{D}_i , $\mathcal{D}_i = \mathcal{D}_i^{\text{on}} \cup \mathcal{D}_i^{\text{off}} \cup \mathcal{D}_i^0$. The variable e represents the shared information, which has been introduced in (11a) and will be used to define the control law. The functions associated to the jump sets in (17) are selected as:

$$\phi_{\text{on}}(q_i, e) := \alpha^T \tilde{q} - \tilde{n} \tilde{q}_i - (\tilde{n} + 1)(\varepsilon + \tilde{n} q^*) \quad (18)$$

$$\phi_{\text{off}}(q_i, e) := \alpha^T \tilde{q} - \tilde{n} \tilde{q}_i - (\tilde{n} - 1)(\varepsilon - \tilde{n} q^*) \quad (19)$$

where we recall that $\tilde{q}_i = q_i - q^*$, $\tilde{q} = q - \mathbf{1}q^*$ and $\tilde{n} = n - n^*$ as defined in equation (11a).

Remark 3: Note that the state β_i multiplies the whole right hand side in the first equation of (2a) but it only multiplies the quantity y in the first equation in the definition of f_i in (16). This simplification is possible because f_i is only evaluated along flows, and no solutions to (2) can flow if $\beta_i = 0$ and $\alpha_i = 1$ (see the definition of \mathcal{C}_i^e in (2b)). □

We are now in the position to express the dynamics of the overall system as a hybrid system described by

$$\begin{aligned} \dot{x} &= f(x) & x &\in \mathcal{C} \\ x^+ &\in G(x) & x &\in \mathcal{D} \end{aligned} \quad (20)$$

where $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, and

$$\begin{aligned} f(x) &= \begin{bmatrix} f_1(x_1, u_1, \bar{w}) \\ f_2(x_2, u_2, \bar{w}) \\ f_3(x_3, u_3, \bar{w}) \end{bmatrix}, & \mathcal{C} &= \bigcap_{i=1,2,3} \mathcal{C}_i \\ G(x) &= \bigcup_{k=1,2,3: x_k \in \mathcal{D}_i} \bar{G}_k(x), & \mathcal{D} &= \bigcup_{i=1,2,3} \mathcal{D}_i \end{aligned}$$

where we recall that the input u_i 's are functions of the variable x as showed in equation (12) and

$$\begin{aligned} \bar{G}_1(x, e) &= \begin{bmatrix} G_1(x_1, e) \\ x_2 \\ x_3 \end{bmatrix}, & \bar{G}_2(x, e) &= \begin{bmatrix} x_1 \\ G_2(x_2, e) \\ x_3 \end{bmatrix}, \\ \bar{G}_3(x, e) &= \begin{bmatrix} x_1 \\ x_2 \\ G_3(x_3, e) \end{bmatrix}. \end{aligned}$$

The following theorem is the main result of this paper.

Theorem 1: Consider the overall network cluster whose dynamics is governed by equations (16)–(19), equivalently (20). Under Assumption 1, this network cluster meets the three design goals specified in Section II-D.

The proof of Theorem 1 is given in Section IV and requires suitable characterizations of the overall dynamics and using the peculiar selections of jump and flow sets in (17)–(19), in addition to the instrumental results of Lemmas 2, 3 and Property 1.

From an intuitive viewpoint, the scheme is characterized by a peculiar behavior where the activation and deactivation of the agents is autonomously decided in order to guarantee convergence to the set $[-\varepsilon/2, \varepsilon/2]$ of the quantity $\alpha^T \tilde{q}$. Indeed, as a byproduct of Lemma 3, this quantity evolves according to the following “quantized” behavior:

$$\alpha \dot{\tilde{q}} = -\tilde{n}y,$$

and enabling or disabling agents (namely, changing \tilde{n}) allows us to shrink it to zero.

A second part of the scheme, which is perhaps more classical, is the consensus law, operating when solutions flow, which can be rewritten in terms of the shifted variables \tilde{q}_i and \tilde{n} . From equation (13) and since q^* is constant, the dynamics of the shifted variable \tilde{q} can be written as

$$\dot{\tilde{q}} = -k_p L(\alpha) \tilde{q} - \frac{\bar{w}}{n} \alpha - \beta y, \quad (21)$$

which performs the desired load balancing among the queues of the active agents.

IV. PROOF OF THEOREM 1

To the aim of proving Theorem 1, we will use the following lemma, establishing useful properties of trajectories. Note that we cannot perform a Lyapunov proof of the convergence property established in Theorem 1 because the proven convergence result is non-uniform and it may take an arbitrarily long time to reach the desired attractor. Due to this fact, the attractor is unstable (even though globally attractive), thereby highlighting a somewhat interesting peculiarity of the problem under consideration.

Proof of Theorem 1.

The proof of Theorem 1 requires the use of the following lemma. Note that due to space limitations, the proof of this lemma is omitted.

Lemma 4: All solutions to (20) eventually evolve continuously in set $\mathcal{A} = \mathcal{F}_2 \cap \{|\alpha^T \tilde{q}| \leq \frac{\varepsilon}{2}\}$.

The processing constraint (10) is proven in Lemma 2. Next we prove the convergence properties in (7)–(9). To this aim, based on Lemma 4, we know that any solution to (20) is eventually continuous. After the solution evolves continuously, thereby not exhibiting any jumps and satisfying $n = n^* = 2$, we may use equation (21) to

obtain that function $V(\alpha, \tilde{q}) = \tilde{q}^T L(\alpha) \tilde{q}$ satisfies along flows:

$$\begin{aligned} \dot{V} &= -\tilde{q}^T L(\alpha) (-k_p L(\alpha) \tilde{q} + \alpha \frac{\bar{w}}{n} - \beta y) \\ &= -k_p \tilde{q}^T L(\alpha)^2 \tilde{q} + \tilde{q}^T L(\alpha) \underbrace{n(\bar{w}/n - y)}_{=0}, \end{aligned} \quad (22)$$

where the term on the right is derived using (3) and Property 1, and is equal to zero due to Assumption 1. Then, from the structure of $L(\alpha)$ and standard consensus results, we obtain that all active agents reach consensus. In particular, since from Lemma 4 the solution eventually evolves continuously in set \mathcal{F}_2 , then we have

$$\lim_{t \rightarrow \infty} n(t, j^*) = 2 = n^*,$$

where j^* is the maximum j in the eventually continuous domain of the solution. Moreover, notice that after a finite time, when the solution evolves continuously in \mathcal{F}_2 , we have that $\alpha^T \tilde{q}$ remains constant (from Lemma 3, Eq. (15)) and that $|\alpha^T \tilde{q}| \leq \varepsilon$ (from the definition of \mathcal{F}_2). This, together with the consensus established by (22), implies that if α_k converges to 1, then

$$\lim_{t \rightarrow \infty} |\tilde{q}_k(t, j^*)| = \frac{1}{2} \lim_{t \rightarrow \infty} |\alpha^T \tilde{q}/2| = \frac{\varepsilon}{2},$$

and if α_k converges to 0, then the corresponding agent remains inactive and its queue q_k converges to zero in finite time. ■

V. SIMULATIONS

Some simulations are performed in MATLAB/Simulink with the Hybrid Equations (HyEQ) Toolbox [13]. In these simulations, 3 agents are considered with a processing rate $y = 5$. The reference for the queue is $q^* = 20$ and the desired number of agents is $n^* = 2$. The parameter of the load balancing control law is $k_p = 2$.

Figure 2 shows three simulations in the situation when the request rate is constant and is chosen $\bar{w} = 2y$. Figures 2a,b,c show the evolution of the three queues and the components of α , for various initial conditions.

In Fig. 2a, all the queues are initially above the desired values q^* . All agents stay or turn on and the queues are decreasing because the processing rate of the network is larger than a threshold rate. Then agent 2 and 3 turn off when the system reaches the jump set, and the other agent remains on. System evolves in this situation until agent 1 reaches $2q^*$, at this moment, agent 3 turns on. In this last configuration, the processing rate in the network is going to become equal to the requests rate and the active agents stay in the desired neighborhood of $[q^* - \frac{\varepsilon}{2}, q^* + \frac{\varepsilon}{2}]$.

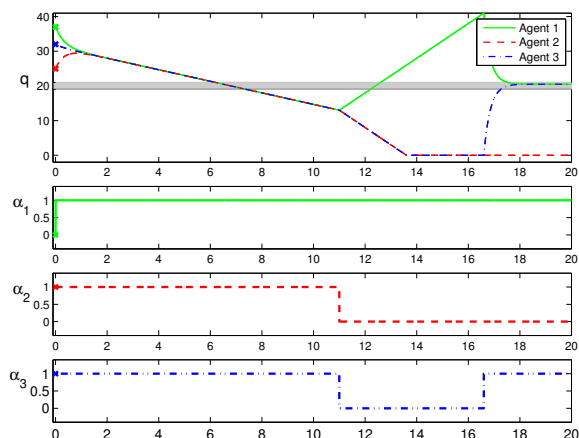
In Fig. 2b, all the queues are initially below the desired values q^* , and in Fig. 2c queues are initially above and below the desired values q^* . Note, that queues evolves with a similar behavior than in Fig. 2a.

VI. CONCLUSIONS

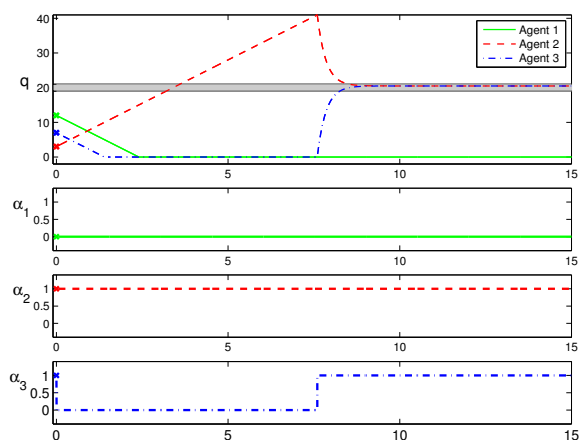
In this article, a novel control architecture for a three agents network is provided. The control law, which is based both on consensus algorithms and on an appropriate hybrid modeling, allows managing the activity of each agent in a distributed manner but with the knowledge of the state of the overall network. This preliminary work finally shows the potential of considering the hybrid representation for this system.

REFERENCES

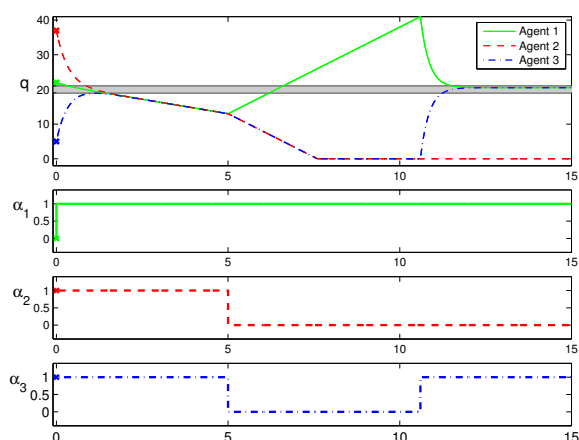
- [1] Z. Tang, J. D. Birdwell, J. Chiasson, C. T. Abdallah, and M. M. Hayat, "Closed loop control of a load balancing network with time delays and processor resource constraints," in *Advances in Communication Control Networks*. ser. Lecture Notes in Control and Information Sciences, S. Tarbouriech, C. Abdallah, and J. Chiasson, Eds. Springer, 2005, pp. 245–268.
- [2] B. Alagöz, A. Kaygusuz, and A. Karabiber, "A user-mode distributed energy management architecture for smart grid applications," *Energy*, vol. 44, no. 1, pp. 167–177, 2012.
- [3] M. H. Willebeek-LeMair and A. P. Reeves, "Strategies for dynamic load balancing on highly parallel computers," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, no. 9, pp. 979–993, 1993.
- [4] H. Kameda, E. Fathy, I. Ryu, and J. Li, "A performance comparison of dynamic vs. static load balancing policies in a mainframe-personal computer network model," in *Proc. of the 39th IEEE Conference on Decision and Control*, vol. 2, 2000, pp. 1415–1420.
- [5] E. Altman and H. Kameda, "Equilibria for multiclass routing in multi-agent networks," in *Proc. of the 40th IEEE Conference on Decision and Control*, vol. 1, 2001, pp. 604–609.
- [6] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation." in *NSDI*, vol. 8, 2008, pp. 323–336.
- [7] S. Wong, L. Valcarengi, S. Yen, D. Campelo, S. Yamashita, and L. Kazovsky, "Sleep mode for energy saving ponds: advantages and drawbacks," in *IEEE GLOBECOM Workshops*. IEEE, 2009, pp. 1–6.
- [8] R. Goebel, R. Sanfelice, and A. Teel, *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.
- [9] —, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 28–93, April 2009.
- [10] E. Gelenbe and C. Morfopoulou, "A framework for energy-aware routing in packet networks." *Computer Journal*, vol. 54, no. 6, 2011.
- [11] R. Olfati-Saber, J. A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [12] W. Ren and R. Beard, *Distributed Consensus Algorithm in Multi-Vehicle Cooperative Control*. Princeton University Press, 2012.
- [13] R. G. Sanfelice, D. Copp, and P. A. Nanez, "A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid equations (HyEQ) toolbox," in *Hybrid Systems: Computation and Control Conference*, 2013.



(a) Simulation with $q(0) = (37, 25, 32)^T$, $\alpha_0 = (0, 1, 1)^T$.



(b) Simulation with $q(0) = (12, 3, 7)^T$, $\alpha_0 = (0, 1, 1)^T$.



(c) Simulation with $q(0) = (22, 37, 5)^T$, $\alpha_0 = (0, 1, 1)^T$.

Fig. 2: Simulations a three nodes network representing the evolution of the queues q_i , $i = 1, 2, 3$ with $q^* = 20$, $n^* = 2$, $k_p = 2$, $\bar{w} = 10$, $y = 5$ and $\varepsilon = 2$ with different initial conditions.