Coordination Control in a Cyberphysical Environment *

Claudio De Persis
ITM, Faculty of Mathematics and Natural Sciences
University of Groningen, Netherlands – c.de.persis@rug.nl

Abstract

An important class of cyberphysical systems are flow networks. Motivated by the node load balancing problem for dynamical flow networks, we propose a self-triggered gossiping control algorithm. This is an edge-based control algorithm in which the inter-node flow is decided in a pair-wise fashion by the two nodes that exchange the flow. The times at which the flow is updated is designed on-line on the basis of the measurements relative to the quantity of goods stored at the nodes. To the best of our knowledge, this is the first time that a gossiping algorithm is devised for deterministic continuous-time systems without relying on an *a priori* schedule of the measurement and control times.

1 A motivating example: flow networks

Flow networks are graphs where flows are associated to edges and stored quantities to the nodes. Flow networks arise in many disciplines and model a variety of transportation phenomena ranging from traffic to supply chains and water networks.
In its simplest form, a flow network is an undirected graph $G = (I, E)$, where I is the set of n nodes and E the set of m edges. Associated to the graph G is the $n \times m$ incidence matrix B. For each node $i \in V$, let x_i denote the state variable representing the amount of stored material at the node. Similarly, for each edge k, let u_k be the flow of material through the edge. Then the system $\dot{x} = Bu$ models the evolution of the stored variables at the nodes as a function of the amount of flow exchanged among the nodes. In this system, the inter-nodes flow is viewed as a control variable at the edge. For each edge, the corresponding control variable has access to local information, namely to the difference in quantities stored at the two nodes connected by the edge. This information is compactly collected in the output vector $z = B^T x$.
Load balancing. The basic node load balancing problem for flow networks is as follows. Given the flow network $\dot{x} = Bu$ find an output feedback control law $u = Kz$ such that for any initial condition the solution of the closed-loop system converges asymptotically to the bisector $\{x \in \mathbb{R}^n : x_1 = x_2 = \ldots = x_n\}$.
It is well-known that if the graph G is connected, the matrix $K = -I_m$ provides a solution to the load balancing problem and the solution of the closed-loop system globally converges to a specific point of the bisector, namely to the average $\mathbf{1}_n^T x(0)/n$ of the initial conditions. Hence, the initial (possibly uneven) initial distribution of quantities stored ad the nodes is evenly distributed as time elapses. Although the setting is rather simplistic (no complex dynamics at the edges or at the nodes, no inflow and outflow, no capacity constraints, etc.) the example above highlights the importance of cooperative control algorithms in the control of dynamical flow networks.

*Joint work with Paolo Frasca, Department of Mathematical Sciences, Politecnico di Torino, Italy. The work is partially supported by the Dutch Organization for Scientific Resaerch under the auspices of the project QUICK.

In dynamical flow network, as well as in cooperative control algorithms, the problem of scheduling the information transmission among the different nodes is of fundamental importance. The algorithm reviewed before requires constant update on the status of the variables at the nodes. Our goal is to propose a different control algorithm in which each controller at the edge acquires information from the nodes only at times which are designed on-line and accordingly updates the flow through the edge. The control algorithm is actually an edge-based algorithm, despite the fact that it achieves an agreement on the variables at the nodes. In this regard, it is reminiscent of several pairwise "gossip" algorithms which have appeared in the literature. However, our approach appears to be the first one which has been devised for continuous-time deterministic systems and which does not require an *a priori* designed scheduling algorithm.

2 Self-triggered gossiping control of flow networks

We introduce the load balancing control algorithm and consider the state variables $(x, u, \theta) \in \mathbb{R}^n \times \{-1, 0, 1\}^m \times \mathbb{R}^m$. The variable x is the vector of stored material at the nodes, u is the ternary flow (goods move in one direction, in the opposite or they do not move at all). The variable θ_k represent a local clock at the edge k controller.
The equations which describe the continuous evolution of the system are

$$\begin{cases} \dot{x}_i = \sum_{k=1}^m b_{ik} u_k \\ \dot{u}_k = 0 \\ \dot{\theta}_k = -1 \end{cases} \tag{1}$$

where $i \in I$, $k \in E$ and b_{ik} are the entries of the incidence matrix B. The system satisfies the differential equation above for all t except for those values of the time at which the set $\mathcal{J}(\theta, t) = \{k \in V \ : \ \theta_k(t) = 0\}$ is non-empty. At these times a discrete transition occurs, which is governed by the following discrete update:

$$\begin{cases} x_i(t^+) = x_i(t) \quad \forall i \in I \\ u_k(t^+) = \begin{cases} \mathrm{sign}_\varepsilon\big(z_k(t)\big) & \text{if } k \in \mathcal{J}(\theta, t) \\ u_k(t) & \text{otherwise} \end{cases} \\ \theta_k(t^+) = \begin{cases} f_k(z(t)) & \text{if } k \in \mathcal{J}(\theta, t) \\ \theta_k(t) & \text{otherwise} \end{cases} \end{cases} \tag{2}$$

where for every $i \in I$ and $k \in E$, the map $f_k : \mathbb{R}^m \to \mathbb{R}_{>0}$ is defined by

$$f_k(z) = \begin{cases} \dfrac{|z_k|}{2(d_i + d_j)} & \text{if } |z_k| \geq \varepsilon \\ \dfrac{\varepsilon}{2(d_i + d_j)} & \text{otherwise,} \end{cases} \tag{3}$$

where d_i, d_j are the degrees of the nodes i, j connected via the edge k. We denote the ℓth time t at which $k \in \mathcal{J}(\theta, t)$ by t_ℓ^k.
The purpose of the talk is to analyze the hybrid control system (1), (2), (3) and show that the solutions converge to the desired balanced distribution of resources among the nodes.

Reference C. De Persis and P. Frasca. Self-triggered coordination with ternary controllers. Available at `http://arxiv.org/abs/1205.6917`.