Feedback scheduling of sensor network activity using a hybrid dynamical systems approach

Olesia Mokrenko, Carolina Albea-Sanchez, Luca Zaccarian, Suzanne Lesecq

Feedback scheduling of sensor network activity using a hybrid dynamical systems approach

Olesia Mokrenko, *Student Member, IEEE*, Carolina Albea, Luca Zaccarian, *Senior member, IEEE*, Suzanne Lesecq, *Member, IEEE*

***Abstract*— Wireless sensor nodes are now cheap and reliable enough to be deployed in different environments. However, their limited energy capacity limits their lifespan. In this paper, a power management strategy at network-level for a set of nodes is proposed. The control strategy makes use of a Hybrid Dynamical System approach, where solutions may continuously flow according to some differential equations and may discontinuously jump according to some rules. The goal of the proposed control strategy is to improve the scalability issues while the network lifespan is not decreased when compared to a Model Predictive Control (MPC) approach, leading to an improvement in the network power consumption. The strategy is evaluated in simulation on a realistic benchmark. Simulation results and their comparison to results obtained with an MPC strategy show the effectiveness of the proposed control strategy.**

I. Introduction

Recently, the interest in Wireless Sensor Networks (WSN) increases with significant results in areas such as structural health monitoring, building automation, civil and military surveillance, bio-medical health monitoring [1]. The wireless sensor nodes measure physical quantities, e.g. temperature, pressure, humidity, location of objects. The nodes can embed actuators or sensors or both of them. If the node is a sensor, it is usually powered by batteries, leading to very limited energy capacity. Each node can be placed in different functioning modes (active, standby, deep sleep, etc.) with their associated power consumption.

Usually, dozens of nodes are spread over a given area in order to sense the environment, for monitoring and/or control purposes. However, more nodes than mandatory for the application are typically deployed, leading to a huge amount of data to deal with, waste in the overall energy consumption, and a short network lifespan. The WSN lifespan increase has already been addressed in the literature, from sensor-level [2]–[4] to network-level. [5] provides an overview of these techniques. Model Predictive Control (MPC) [6] has been used in order to select the devices to be in *Active* mode both to limit the WSN overall energy consumption and to extend its lifespan while the WSN fulfills a given "mission". Even if the results are appealing, the control law requires a Mixed Integer Quadratic Programming (MIQP) problem to be solved, which is computationally demanding.

In the present paper, we propose another control approach in order to improve the scalability issues and decrease the number of switches while the network lifespan is not decreased when compared to the MPC approach, leading to an improvement in the network power consumption. This control makes use of a Hybrid Dynamical System (HDS) approach [7], where solutions may continuously flow according to some differential equations and may discontinuously jump according to some rules. Therefore, an innovative strategy that fulfills the control objectives, is proposed for the control of the functioning modes of the nodes.

The paper is organized as follows. Section II is first dedicated to the system description. Then the problem formulation is provided. The scheduling law design is developed in Section III. It is based on a Hybrid Dynamical System approach. Section IV evaluates in simulation the HDS control strategy on a realistic benchmark. Comparisons with an MPC approach [6] are also provided. Section V summarizes the main results and draws future work directions.

Notations: Throughout the paper, $\mathbb{N}^*$ denotes the set of natural numbers with $\mathbb{N}^* = \mathbb{N}\backslash\{0\}$. Vector e_k denotes column k of the identity matrix, or the k^{th} vector of the Euclidean basis.

II. Wireless sensor network description

A. Physical description

A multi-hop heterogeneous WSN architecture is considered all over the present work. Each node S_i, $i = 1, ..., n$, $n \in \mathbb{N}^*$, communicates only with a centralized controller that collects measurements and that is responsible for monitoring and controlling the overall WSN. The nodes are functionally equivalent: they are interchangeable but their hardware may differ. Each node S_i is powered by a battery, whose remaining energy level is denoted by x_i. Moreover, each node exhibits two functioning modes M_1 and M_2, associated with known power consumption:

- M_1 is the *Active* mode. S_i senses, computes and communicates in "duty cycling" (see Fig. 1), i.e. it is off by default and it wakes up periodically with a sampling period of $T_s = 1min$ to sense, process and exchange data with the controller;
- M_2 corresponds to the *Standby* mode, where S_i is in sleep state. In this state, only a small part of S_i is active (i.e. the Real Time Clock (RTC)), that will allow to

O. Mokrenko and S. Lesecq are with Univ. of Grenoble Alpes, 38000 Grenoble, France and CEA, LETI, MINATEC Campus, 38054 Grenoble, France {FirstName.LastName}@cea.fr

C. Albea is with CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France and Univ. of Toulouse, UPS, LAAS, 31400, Toulouse, France calbea@laas.fr

L. Zaccarian is with CNRS, LAAS, 7 avenue du Colonel Roche, 31400 Toulouse, France and Univ. of Toulouse, and Dipartimento di Ingegneria Industriale, Univ. of Trento, Italy. zaccaian@laas.fr

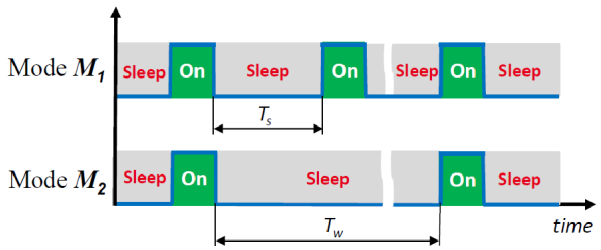

Fig. 1: Duty cycles of modes M_1 and M_2.

wake up S_i each $T_w = 1h$ to receive information from the controller and monitor the remaining capacity of the node battery.

Each node S_i is therefore characterized by:

- two states: $x_i(t,j) \in \mathbb{R}_{\geq 0}$ is the remaining energy in the node battery, where t is the continuous time and j is the total number of jumps of the solution. $x_i(0,0)$ denotes the initial battery capacity. $u_i \in \{0,1\}^2$ denotes the functioning mode M_1 or M_2. u_i has one component equal to 0 and one equal to 1. Thus, $u_i = e_k$ means S_i is in mode M_k. $u_i(0,0)$ denotes the initial functioning mode of node S_i;
- a power consumption vector $b_i \in \mathbb{R}^2_{>0}$. Each component k denotes the power consumed by S_i when operating in mode M_k (i.e., when $u_i = e_k$). From a practical viewpoint, the components in b_i are assumed unequal (i.e. the consumption is different in both modes);
- a switching power consumption between two functioning modes $\delta_i^{k\to l} \in \mathbb{R}_{>0}$, $k,l \in \{1,2\}$ that encompasses the fact that switching the node from mode M_k to mode M_l has an energy cost. Moreover, $\delta_i^{k\to k} = 0$.

Problem formulation The main goal of this paper is to extend the WSN lifespan by reducing the overall power consumption of the nodes via an appropriate management of the functioning mode of each node while providing a given functionality, hereafter named *mission*. The mission is expressed as a constraint on the number of nodes operating in mode M_1 at each time, in order to ensure the requested services at the application level (i.e. for the application/services built on top of the WSN). During the continuous-time evolution of solutions (x,u), we must have:

$$\sum_{i=1}^{n} u_i^T e_1 = d_1, \tag{1}$$

where $d_1 \in \mathbb{N}^*$ defines the exact number of active nodes, and can be seen as an external reference signal.

B. Hybrid representation and pairwise jump rules

Within the above setting, the flow dynamics of the n nodes are given by:

$$\begin{aligned} \dot{x}_i &= -b_i^T u_i, \\ \dot{u}_i &= 0, \end{aligned} \qquad i = 1,...,n, \qquad (x,u) \in \mathcal{C}, \tag{2}$$

where the flow set $\mathcal{C}$ has to be designed. The dynamics of state x can be conveniently written as $\dot{x} = -Bu$ where $x = \begin{bmatrix} x_1 \, x_2 \, ... \, x_n \end{bmatrix}^T \in \mathbb{R}^n$, $u = \begin{bmatrix} u_1 \, u_2 \, ... \, u_n \end{bmatrix}^T \in \{0,1\}^{2n}$, and $B = \text{diag}(b_1^T, b_2^T, ..., b_n^T) \in \mathbb{R}^{n\times 2n}$ is a block diagonal matrix having b_i^T on its (block) diagonal entries.

The jump dynamics of the n nodes comprise the possibility that the nodes autonomously decide to swap their respective roles within the network. Given (2), we readily understand that swapping roles means simply swapping the values of u_i. Then, we may define the sets $\mathcal{D}_{ih}$ to provide conditions under which two nodes are required to swap their roles (under the straightforward assumption that $\mathcal{D}_{ih} = \mathcal{D}_{hi}$, so that swapping is simultaneously enabled from both sides). The adopted paradigm intrinsically defines a distributed scheduling paradigm, as long as one restricts the sets $\mathcal{D}_{ih}$ to be non-empty (therefore relevant for the "potential swap" evaluation) only for pairs (i,h), belonging to the edges of a suitable undirected interconnection graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ characterizing the nodes that can communicate with each other.

In the general case, the following sets will be designed:

$$\mathcal{D}_{ih} = \mathcal{D}_{hi}, \forall i,h : \ (i,h) \in \mathcal{E}, \tag{3}$$

where $\mathcal{E}$ is the set of all edges of the interconnection graph. Based on the sets in (3), which will be designed in Section III, we may represent the swapping as an instantaneous update of the states of nodes S_i, S_h, corresponding to:

$$\begin{aligned} \begin{bmatrix} x_i^+ \\ x_h^+ \end{bmatrix} &= \begin{bmatrix} x_i - (u_i^+)^T \Delta_i u_i \\ x_h - (u_h^+)^T \Delta_h u_h \end{bmatrix}, \\ \begin{bmatrix} u_i^+ \\ u_h^+ \end{bmatrix} &= \begin{bmatrix} u_h \\ u_i \end{bmatrix}, \end{aligned} \qquad (x,u) \in \mathcal{D}_{ih}, \tag{4a}$$

where the switching consumption matrices are given by:

$$\Delta_i = \begin{bmatrix} 0 & \delta_i^{2\to 1} \\ \delta_i^{1\to 2} & 0 \end{bmatrix}, \quad i = 1,\dots,n. \tag{4b}$$

Equations (4) only indicate the instantaneous change for a pair of nodes, but for a complete description of the dynamics, we should also specify that across those jumps all other nodes S_i do not experience any change of their x_i and u_i components. In particular, relations (4) may be compactly represented as:

$$\begin{bmatrix} x^+ \\ u^+ \end{bmatrix} = \begin{bmatrix} g_x^{ih}(x,u) \\ g_u^{ih}(x,u) \end{bmatrix} =: g_{ih}(x,u), \quad (x,u) \in \mathcal{D}_{ih}, \tag{5}$$

where $g_{ih} : (\mathbb{R} \times \{0,1\}^2)^n \to (\mathbb{R} \times \{0,1\}^2)^n$ can be straightforwardly computed from (4). With the pairwise rules in (5), we may then define an overall jump rule gathering together all of the relations in (5), and associated to a jump set $\mathcal{D}$ corresponding to the set where at least one pair of nodes is ready for a swap operation (namely $x \in \mathcal{D}$ if $x \in \mathcal{D}_{ih}$ for at least one pair (i,h)):

$$\mathcal{D} := \bigcup_{(i,h)\in\mathcal{E}} \mathcal{D}_{ih}, \quad G(x,u) := \bigcup_{ih:\ (x,u)\in\mathcal{D}_{ih}} g_{ih}(x,u), \tag{6}$$

where by construction, G is a set-valued mapping (multiple pairs may be ready to swap at the same time) that possesses the useful property of having a closed graph because its

graph is the union of the (closed) graphs of g_{ih}. Discrete-time dynamics (6) together with (2) correspond to a hybrid description of the WSN system under the specific control action, using the notation in [7].

Here, we focus on solutions to (2), (6), insisting that they evolve in a specific set $\mathcal{O}$ where the remaining battery energy is non-negative for all nodes, and the input vector u has components equal to 0 or 1. We will also insist that, within this set, the flow set $\mathcal{C}$ is the closed complement of the flow set $\mathcal{D}$ relative to $\mathcal{O}$. More specifically:

$$\mathcal{O} = \mathbb{R}^n_{\geq 0} \times \{0,1\}^{2n}, \quad \mathcal{C} = \left(\overline{\mathcal{O} \setminus \mathcal{D}}\right) \bigcap \mathcal{O}. \tag{7}$$

Within the set $\mathcal{O}$, it is evident that, due to the positivity of the entries in vectors b_i, all solutions will be bounded and not complete (their domain is bounded). Then the objective is to design the jump sets $\mathcal{D}_{ih}$ in an intuitive way, the goal being to maximize the length of the solution domain in the ordinary time direction, i.e. the WSN lifespan until its batteries have been drained, named hereafter the *lifespan of the solution*.

III. Design of the scheduling law

A. Case with two nodes

In Section II-B, the design problem formulated in Section II-A has been reduced to the pairwise jump sets $\mathcal{D}_{ih}$ hybrid dynamics 2, 5, 6. The rest of the dynamics explain the evolution during flow (batteries are discharging) and what should happen at each reconfiguration of the network (basically, when the state (x,u) belongs to $\mathcal{D}_{ih}$, nodes S_i and S_h swap their roles). To suitably design $\mathcal{D}_{ih}$ for all $i \neq h$, we focus on a pairwise reconfiguration rule. This rule is described here for the simplified case of two nodes S_1, S_2, with $\mathcal{D}_{12} = \mathcal{D}_{21} = \mathcal{D}$. For this specific case, an optimal result can be proved, see below. Therefore, we focus on the two-nodes case, i.e. $n = 2$, where only $\mathcal{D}_{12}$ must be designed.

The rationale behind the choice of $\mathcal{D}_{12}$ is that we would like to design an algebraic condition on x and u that expresses when it is convenient to swap roles between the nodes, in such a way to maximize the *lifespan of the solution*. This quantity may be expressed as a cost function J to be maximized. When designing $\mathcal{D}_{12}$, the expected lifespan if the solution does not perform further jumps corresponds to:

$$J(x,u) = \min_{i=1,2} \frac{x_i}{b_i^T u_i}. \tag{8}$$

A first condition to encode in the flow set is that it is not convenient to jump (or swap roles) whenever $J(x^+,u^+) < J(x,u)$. Thus, one must ensure that:

$$\begin{aligned} J^+(x,u) := J(g_x(x,u), g_u(x,u)) < J(x,u) \\ \Rightarrow \ (x,u) \notin \mathcal{D}_{12}. \end{aligned} \tag{9}$$

Intuitively, condition (9) means that no swap is performed if it reduces the lifespan. Note that this condition is a function of both x and u.

Even though condition (9) is reasonable, it may still induce undesirable behaviours in some cases. For instance, consider two identical nodes, namely $\Delta_1 = \Delta_2$, $b_1 = b_2$. The *Active* (resp. *Standby* mode) is supposed to be associated with a large (resp. small) power consumption. Assume also that Δ_i are relatively small when compared to the power consumption of the *Active* mode. In this case, the best strategy is clearly to keep one node in *Active* mode until its battery is drained, and then swap once and only once along the solution. However, picking $\mathcal{D}_{12}$ as the closed complement of the left hand condition of (9) would force extra unnecessary jumps as soon as the energy level in the active node becomes sufficiently small, and then vice-versa, and so on.

One way to avoid this situation is to encode in $\mathcal{D}_{12}$ another condition: nodes will not swap holes if there is no "emergency" to do so. Indeed, when the left hand condition in (9) holds, solutions will still keep flowing if waiting any further (thus, no switch applied) does not cause any reduction in the expected lifespan after the potential switch. To characterize this reduction, denote by i^* the index (or the set of indexes) that minimizes J^+, namely:

$$i^* = \operatorname*{argmin}_{i \in 1,2} \frac{x_i^+}{b_i^T u_i^+} = \operatorname*{argmin}_{i \in 1,2} \frac{x_i - u_{3-i}^T \Delta_i u_i}{b_i^T u_{3-i}}. \tag{10}$$

Note that i^* may contain both indexes if the two expected lifespans coincide. Then, characterize the reduction as the derivative of J^+ during flow (if no jump was performed):

$$\delta J(x,u) = \min_{h \in i^*} \frac{\dot{x}_h^+}{b_h^T u_h^+} = \min_{h \in i^*} -\frac{b_h^T u_h}{b_h^T u_{3-h}}. \tag{11}$$

Note that the derivative of $u_{3-i}^T \Delta_i u_i$ is zero along flow. Function (11) captures the idea of how damageful it is to postpone the swap, i.e. how much smaller J^+ will become if the solution flows for some extra time, before jumping. Since the two components in b_i are supposed not equal, then $\delta J(x,u) \neq -1$.

Clearly, if the solution keeps flowing, the decrease rate of J will simply be 1 (as the lifespan decreases linearly as time flows). Therefore, one extra criterion for the selection of $\mathcal{C}$ may be:

$$\delta J(x,u) \geq -1 \quad \Rightarrow \quad (x,u) \notin \mathcal{D}_{12}, \tag{12}$$

which is well defined, as indicated above, because $\delta J(x,u) \neq -1$. In particular, what happens along solutions, as long as $J^+ \geq J$ is that $\delta J(x,u) > -1$ whenever there is no urge to jump, and then once the "argmin" in (10) changes, we start getting $\delta J(x,u) \leq -1$ and a jump occurs.

To summarize, we suggest the selection:

$$\mathcal{D}_{12} = \{(x,u) : \ J^+(x,u) \geq J(x,u) \text{ and } \delta J(x,u) \leq -1\}, \tag{13}$$

where it is emphasized that the definition is commutative (that is, there is no specific role of nodes S_1 and S_2 in the selection of $\mathcal{D}_{12}$). Selection (13) enables to prove the following optimality result.

Theorem 1: Assume that b_i, $i = 1,2$, has positive and distinct components and that matrices Δ_i, $i = 1,\ 2$ have strictly positive off diagonal elements. Consider any initial

condition with positive values of $x_i(0,0)$, $i=1,2$ and with $u_i(0,0)$, $i=1,2$ being two independent columns of the identity matrix. The solution of (2), (6), (13) is unique and it has a lifespan equal to the maximum lifespan that can be obtained by selecting arbitrary jump times for (2), (4) starting from the same initial conditions.

Due to space limitations, the proof of the theorem is omitted.

B. General case

The approach of Subsection III-A is now extended to the case with $n > 2$ nodes. The hybrid dynamics in (2), (6) essentially concentrate on pairwise relations between each possible pair of connected nodes. In particular, parallel quantities may be defined to those in (8), (9), (10), and (11) when focusing on nodes S_i, S_h satisfying $(i,h) \in \mathcal{E}$ (see (3)) as follows:

$$J_{ih}(x,u) := \min_{k=i,h} \frac{x_k}{b_k^T u_k} \tag{14}$$

$$J_{ih}^+(x,u) := J_{ih}(g_x^{ih}(x,u), g_u^{ih}(x,u)) \tag{15}$$

$$k_{ih}^* = \operatorname*{argmin}_{k\in\{i,h\}} \frac{x_k - u_{k^c}^T \Delta_k u_k}{b_k^T u_{k^c}} \tag{16}$$

$$\delta J_{ih}(x,u) = \min_{k\in k_{ih}^*} -\frac{b_k^T u_k}{b_k^T u_{k^c}}, \tag{17}$$

where the shortcut k^c represents the "second" node in a pair, namely $k^c = i$ if $k = h$ and *vice-versa*.

With the above definitions, the jump sets $\mathcal{D}_{ih}$ can now be defined by suitably generalizing the expression in (13):

$$\mathcal{D}_{ih} = \{(x,u) : J_{ih}^+(x,u) \geq J_{ih}(x,u) \text{ and } \delta J_{ih}(x,u) \leq -1\}, (i,h) \in \mathcal{E}. \tag{18}$$

With selection (18), we may now prove the following result, which is a straightforward consequence of the fact that the jumps in (4) have no effect on the mission because they merely correspond to swapping the states of nodes S_i and S_h. The proposition proof is omitted because it is a straightforward consequence of the fact that quantity $\sum_{i=1}^n u_i^T e_1$ is constant along flow and across jumps.

Proposition 1: Given any solution to (2), (6), (18), if the mission (1) is satisfied at the initial condition, then it is satisfied all along the solution.

IV. Application

A. System description

The effectiveness of the proposed approach is shown with $n = 6$ nodes, in simulation. Each node S_i may be placed in two functioning modes, i.e. M_1 *Active* or M_2 *Standby* mode (see Section II).

The nodes are supposed to be deployed over a given space (e.g. an office room) to control the environmental conditions (temperature, humidity, etc.). The mission is defined as in (1) with $d_1 = 3$, which ensures that enough information is provided to the room environmental controller. For each node S_i, the components of vector b_i are given in Table I. They are derived from the technical datasheet of the OpenPicus node [8] and from laboratory measurements. The first (respectively second) component of b_i corresponds to the power consumption in mode M_1 (respectively M_2). The initial state value $x(0,0)$ is also derived from Table I: for each S_i, it is equal to $\gamma_i \cdot X_i^{max}$, where γ_i is the energy coefficient and X_i^{max} represents the nominal battery capacity taken from the technical data-sheet of Li-polymer rechargeable batteries [9]. The initial state value $u(0,0)$ is chosen equal to $[1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0]^T$ so as to fulfil the mission (i.e. $d_1 = 3$ nodes in mode M_1).

The switching consumption matrices are supposed to be

TABLE I: Nodes power characteristics.

Node	Mode M_1 [mW]	Mode M_2 [mW]	Nominal battery capacity X_i^{max} [mWh]	Energy coef. γ_i [1]
S_1	34.854	5.846	3885	1
S_2	36.482	6.031	3885	0.8
S_3	36.593	6.105	3885	0.9
S_4	36.482	6.301	3515	0.7
S_5	36.556	6.105	3515	1
S_6	33.041	5.735	3515	1

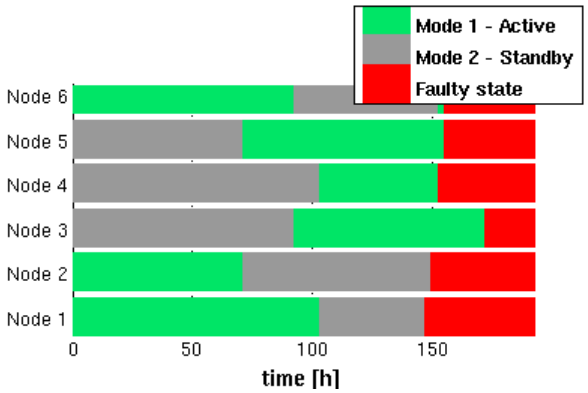

Fig. 2: Scenario 1. Evolution of the functioning modes of the nodes, without communication perturbations and with HDS strategy.

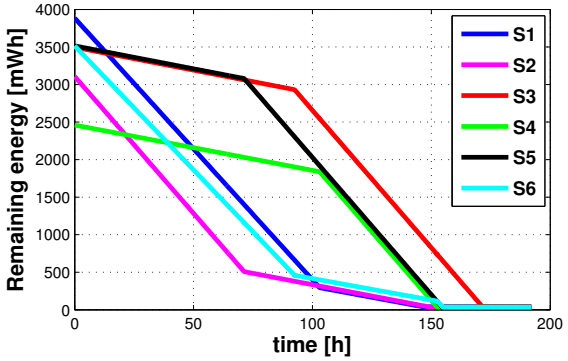

Fig. 3: Scenario 1. Remaining energy in each node battery, without communication perturbations and with HDS strategy.

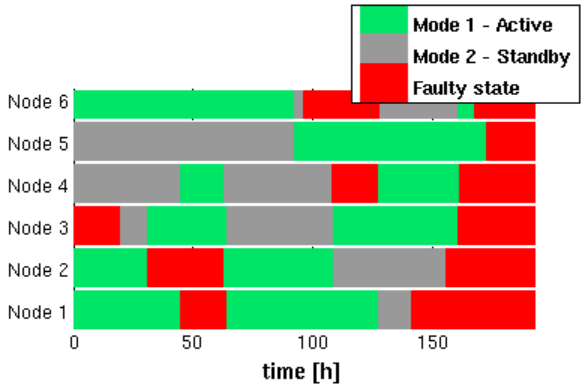

Fig. 4: Scenario 2. Evolution of the functioning modes of the nodes, with communication perturbations and HDS strategy.

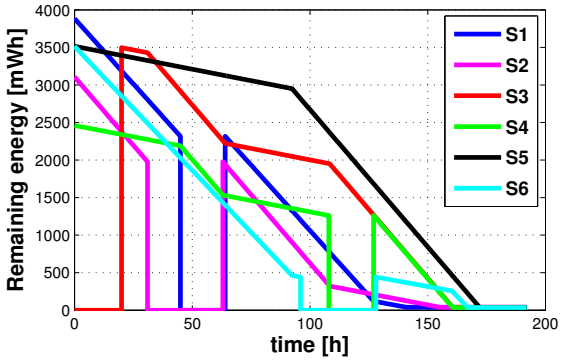

Fig. 5: Scenario 2. Remaining energy in each node battery, with communication perturbations and HDS strategy.

equal for all nodes, given by:

$$\Delta_i = \begin{bmatrix} 0 & 0.01 \\ 0.05 & 0 \end{bmatrix} mWh. \tag{19}$$

B. Simulation results

Simulations are performed in MATLAB/Simulink using the HyEQ Toolbox [10]. The first scenario corresponds to the ideal situation, i.e. without any perturbation in a WSN. Here, the remaining energy x_i of a node is equal to zero, i.e. when the battery of this node is drained, the node is considered to fall to a *faulty* condition. Fig. 2 shows the evolution of the functioning mode for each node when the HDS control is implemented. The energy in the battery of each node is given in Fig. 3. In these figures, we can see that, for instance, when node S_6 has no more energy ($x_6 = 0$) at time instant $151h$, it falls in the *faulty* condition (red color). Note that, the mission (3 nodes in mode M_1) is fulfilled until $\approx 154h$, namely, there are exactly three active nodes until $t = 154h$. This value represents the WSN lifespan. Note that, while the exception of node S_3, essentially all batteries expire at the same time time, while gives a rough idea of the level of sup-optimality of the proposed solution.

The second scenario is more realistic. In this scenario, we consider possible perturbations due to physical damages of some nodes and possible strong perturbations of the radio channel in a real WSN. Due to these perturbations, a node can unpredictably "disappear" and/or "appear" for the controller during the system evolution. The state of a "disappear" node is also named *faulty* condition, because from the controller viewpoint it is associated to the same non-visibility, as compared to the first scenario (when $x_i = 0$) of a faulty node. This more sophisticated effect can be modeled by introducing an additional time-varying exogenous input that changes the number of reachable (not faulty) nodes. Namely, when a sensor node falls in the *faulty* condition, it is removed from the "game", i.e. it is not considered in the problem to fulfil mission (1). This means that the dimension of the problem change each time a sensor node appears/disappears in the WSN. In this situation, the new problem will be considered (e.g. at time t_1 and at jump j_1) with the amended number of sensor nodes and with the corrected initial state values of remaining energy $x_i(t_1, j_1)$ in the battery and control state $u_i(t_1, j_1)$ of the system.

Fig. 4 shows the evolution of the functioning mode for each node with perturbations, when the proposed HDS control is applied. Fig. 5 shows the remaining energy in the battery of each node. In this scenario, the WSN lifespan is $\approx 161h$. This small increase in the lifespan roots in the fact that, in the simulation, the power consumption in the *faulty* condition is supposed to be equal to zero, leading to an underestimation of the energy consumption. However, this assumption is not always true in real-life. Indeed, during communication perturbations, the *faulty* node still consumes energy. When it is again reachable by the controller, it sends information about its remaining energy level, so that the HDS strategy will be able to consider this node as a potential node to be placed in *Active* or *Standby* mode.

C. Comparison between HDS and MPC strategies

A comparison between the hybrid control scheme and the MPC strategy proposed in [6] is performed here. The same WSN benchmark is considered, with the same profile of the perturbation as in scenario 2.

Recall that both control strategies have been designed with the objective of an increase of the WSN lifespan when compared to the traditional approach where all nodes communicate their data, without any control of their functioning mode. However, some small differences must be highlighted:

- the MPC problem penalizes the most consuming nodes; it fulfils the mission thanks to a set of constraints, and it does not take into account the remaining energy unlike the proposed HDS strategy;
- in [6], the energy consumed when switching from one mode to another $\delta_i^{k\to l}$ is supposed to be integrated in b_{il}.

Fig. 6 depicts the functioning mode of each node when the MPC strategy is implemented. With the MPC strategy the WSN lifespan is $139h$. The nodes remaining energy with MPC is given in Fig. 7.

The comparison between the HDS and MPC strategies is summarized in Table II. The WSN lifespan (that corresponds to the interval of time when the WSN fulfils the mission)

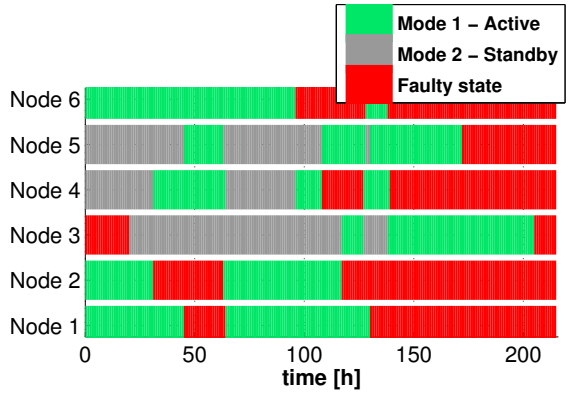

Fig. 6: Scenario 3. Evolution of the functioning modes of the nodes, with communication perturbations and MPC strategy.

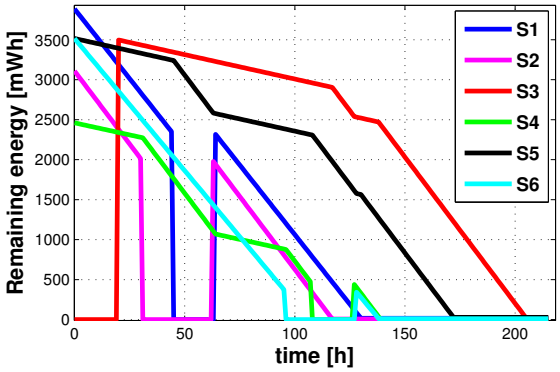

Fig. 7: Scenario 3. Remaining energy in each node battery, with communication perturbations and MPC strategy.

TABLE II: Comparison of two control strategies.

Strategy	WSN lifespan [h]	Simulation duration [sec]
HDS	161	≈ 3.3
MPC	139	≈ 60.5

using HDS is $22h$ longer than the one obtained using MPC. Moreover, the MPC approach requires solving MIQP problem, which is known to be NP-hard [11]. This usually implies a large computational cost. Indeed, in the Matlab environment, the whole simulation time for the MPC approach is $\approx 60.5sec$, thus $\approx 0.5sec$ for each MPC solution. The HDS algorithm seems less complex because the global simulation time is $\approx 3.3sec$. Moreover, the control algorithm is executed less frequently. These preliminary results regarding the complexity of the algorithms and their computational cost must be extended to obtain a fair comparison between the two approaches.

V. Conclusions and Future Work

In this paper, a hybrid control algorithm for energy management in a WSN has been proposed when the WSN has to provide a given functionality (named mission). The energy in the sensor nodes is modelled using a hybrid dynamical system representation. An optimality theorem has been presented in the case when two nodes are considered. In simulation, we considered two different scenarios, the first one corresponding to an ideal case and the second one to a more realistic scenario with perturbations. This second scenario takes into account the possibility that nodes may fall in a *faulty* condition due to physical damages or communication problem. These simulation results show that our control strategy is robust with respect to the presence of nodes in the *faulty* condition. The simulation on a realistic benchmark and comparison of the results with an MPC strategy show the effectiveness of the proposed control strategy.

Future work directions include the implementation of the proposed hybrid control approach on a real test-bench. The results of this implementation will be compared to the ones already achieved on the real test-bench for the MPC strategy. Note that both approaches (HDS and MPC) require the measurement of the remaining energy in the battery of the nodes. Even if techniques already exist to estimate the state-of-charge of a battery, a low computational-cost solution must be searched and implemented in the nodes. Lastly, an extension of the optimality theorem must be investigated to address the general case.

Acknowledgment

This work has been partly funded by the Artemis ARROWHEAD project under grant agreement nb. 332987, by the ANR project LimICoS nb. 12 BS03 005 01, by the iCODE institute, research project of the Idex Paris-Saclay, by grant OptHySYS funded by the University of Trento, and by the PEPS JCJC project HYBRISCON nb. 52 814.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 2002.

[2] N. P. Mandru. Optimal power management in wireless sensor networks for enhanced life time. *Journal of Global Research in Computer Science*, 3:73–78, 2012.

[3] W. Hailong, S. Yan, and W. Tuming. Dynamic power management of wireless sensor networks based on grey model. In *3rd International Conference on Advanced Computer Theory and Engineering*, 2010.

[4] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta. Optimal energy management policies for energy harvesting sensor nodes. *IEEE Transactions on Wireless Communications*, 2010.

[5] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.

[6] O. Mokrenko, S. Lesecq, W. Lombardi, D. Puschini, C. Albea, and O. Debicki. Dynamic power management in a wireless sensor network using predictive control. In *40th Annual Conference of the IEEE Industrial Electronics Society*, 2014.

[7] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[8] http://www.openpicus.com.

[9] http://www.farnell.com/datasheets/1666650.pdf and 1666648.pdf.

[10] R. Sanfelice, D. Copp, and P. Nanez. A toolbox for simulation of hybrid systems in matlab/simulink: Hybrid equations (hyeq) toolbox. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 101–106, 2013.

[11] A. Del Pia, S. S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in np. *arXiv preprint arXiv:1407.4798*, 2014.