

Energy-aware wireless networked control using radio-mode management

Nicolas Cardoso de Castro, Carlos Canudas de Wit, Federica Garin

► **To cite this version:**

Nicolas Cardoso de Castro, Carlos Canudas de Wit, Federica Garin. Energy-aware wireless networked control using radio-mode management. 2012 American Control Conference (ACC 2012), Jun 2012, Montréal, Canada. pp.2836-2841, 2012. <hal-00679683v2>

HAL Id: hal-00679683

<https://hal.inria.fr/hal-00679683v2>

Submitted on 28 Jun 2012

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.

Energy-aware wireless networked control using radio-mode management

Nicolas Cardoso de Castro, Carlos Canudas de Wit and Federica Garin

Abstract—Energy efficiency is one of the main issues in wireless Networked Control Systems. The control community has already shown large interest in the topics of intermittent control and event-based control, allowing to turn off the radio of the nodes, which is the main energy consumer, on longer time intervals than in the periodic case. While the existing literature only addresses policies using two radio-modes (**Tx** - Transmitting, and **Sleep**), this paper considers intermediate radio-modes, which consume more energy than the **Sleep** mode but have cheaper transition costs to the Transmitting mode. We propose an event-based radio-mode switching policy to perform a trade-off between energy saving and performance of the control application. To this end, we derive a switched model taking into account control and communication. We compute the optimal switching policy using Dynamic Programming and we illustrate the results in simulations.

Index Terms—Networked Control Systems, event-based control, radio-mode management.

I. INTRODUCTION

Wireless Networked Control Systems (NCSs) are systems in which the sensors or/and the actuators communicate with the controller through a wireless network. Energy saving and robustness to data loss are major challenges in wireless networks, addressed by both communication and control communities. In this paper, we focus on the radio chip as it is admitted that the energy it consumes represents 50% to 80% of the overall consumption of a smart sensor node, see *e.g.* [11], [13], [20]. An often-quoted rule of thumb is that executing 3 million instructions is equivalent to transmitting 1000 bits at a distance of 100 meters in terms of expended energy [15]. It is then interesting to increase the computation load to decide to use the radio or not.

Deep interest has been devoted to intermittent, or event-based, control and estimation, *i.e.*, problems where the measurements or the control input may not be available at some undetermined time because the sensor node switches off to save energy (see *e.g.* [6], [9], [16], [7], [10], [18], [21]). Author in [6] derives a policy to decide when to send measurements to the controller and the optimal associated state-feedback. The event-based policy tested on an example is less efficient than a periodic policy sending more information, but more efficient than a policy sending the same amount of samples at random time instants. Different setups consider several sensor nodes where only one node can transmit at a

time, see *e.g.* [9], or setups where the sensor nodes can adapt the transmission power of the radio chip to face bad channel conditions, or to save energy, see *e.g.* [16].

On the other hand, the communication literature investigates energy saving by switching off only some parts of the node, introducing the notion of energy mode management. While some contributions address setups where entire features of the node (computation, communication, sensing) are turned off (see *e.g.* [17]), we restrict our attention to the case where only the radio chip is switched to low consuming modes, turning off some components, such as the frequency synthesizer, the crystal oscillator, or the voltage regulator within the radio chip (see [5], [12]). However these works do not take into account the application, *i.e.*, the feedback control, in order to choose the radio-mode. To the best of the authors knowledge, there is no work dealing with both a control application and several radio-modes.

The goal of this paper is to derive an event-based radio-mode switching policy to save energy. The novelty of this contribution is to derive a policy jointly taking into account the performance of the feedback loop and several radio-modes. Radio-mode management is not trivial since the intermediate radio-modes consume more energy than the **Sleep** mode, but they have cheaper transition costs to switch to the Transmitting (**Tx**) mode. Also, even if intermediate modes consume less energy than the **Tx** mode, they do not allow data transmission, and the energy needed to switch between modes may result in more wastes than savings. The standard control approach consists in considering only the two extreme radio-modes (typically **Tx** and **Sleep**), defining a reasonable time interval to run open-loop, and switching down the radio periodically when not transmitting. This approach results in a fix energy consumption per period, based on the worst case scenario. Our approach deals with event-based (aperiodic) data transmissions depending on the current system state, with the use of several radio-modes. Not only the sensor node should decide whether to transmit or not, but also, it should decide which of the low consuming modes to switch to when not transmitting.

A switched linear system taking into account several radio-modes and the control application is derived in Section II. The optimal switching policy, computed using Dynamic Programming, is presented in Section III. Simulation results are provided in Section IV and Section V concludes the paper and gives future directions.

Work supported by EU STREP project FeedNetBack FP7-ICT-2007-2. This is an authors' post-print of a paper published in the Proceedings of ACC2012, with some corrections in Section III-B.

Nicolas Cardoso de Castro and Federica Garin are with: INRIA Rhône-Alpes, NeCS Team, Grenoble, FRANCE, emails: {nicolas.cardosodecastro, federica.garin}@inria.fr, and Carlos Canudas de Wit is with: Department of Automatic Control GIPSA-Lab, CNRS, NeCS Team Grenoble, FRANCE, email: carlos.canudas-de-wit@gipsa-lab.grenoble-inp.fr.

II. PROBLEM FORMULATION

A. Setup description

For simplicity, we consider a wireless networked control problem composed of two nodes, as depicted in Fig. 1. The first node has computing capabilities and is then called the smart node. It is in charge of sensing the system output, computing the feedback law and deciding whether or not to send the control input to the second node, in charge of applying the control law to the actuator. The aim of this paper is to save energy at the smart node's radio chip level when the quality of the feedback control is good enough. The radio chip is switched to low consuming modes (*e.g.* Idle, Sleep) to save energy. We are not interested here in the consumption of the second node as we assume that it is co-located with the actuator, and then it has access to an unlimited energy source. This assumption is realistic in many wireless sensor network situations, as motivated in [1].

We define N as the number of radio-modes. The switching decision is denoted by $v_k \in \{1, 2, \dots, N\}$, where $v_k = i$ means that the radio-mode is switched to mode i at time k .

1) *System model*: The system we control is a linear unstable discrete-time controllable system with an additive zero-mean white Gaussian noise, described by Eq. (1).

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (1)$$

where x_k is the system output and u_k is the control input taking values in \mathbb{R}^n and \mathbb{R}^p respectively, and $w_k \sim \mathcal{N}(\mathbf{0}, \bar{W})$ is the measurement noise. A and B have appropriate dimensions.

2) *Channel model*: We consider a simplified memoryless erasure channel where message u_k is dropped with probability ϵ , and otherwise is correctly received. We consider a model where the dropout concerns the real-valued message u_k , not single bits or packets. The transmission successes are modeled by i.i.d. Bernoulli random variables β_k :

$$\beta_k = \begin{cases} 1, & \text{success with probability } 1 - \epsilon, \\ 0, & \text{failure with probability } \epsilon. \end{cases}$$

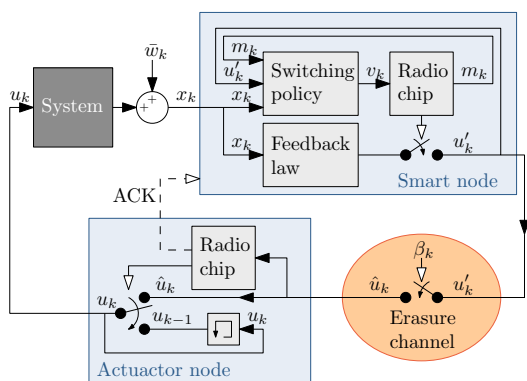


Fig. 1. Block diagram of the problem setup. The smart node measures the noisy state x_k from the system, it computes the feedback law u'_k and decides whether to send it or not to the actuator node. \hat{u}_k is equal to u'_k when a transmission occurs, or to \emptyset otherwise. The receiver is then able to determine if it has received an update or not. β_k equals 1 when the transmission is successful or 0 when there is a dropout.

3) *Feedback law*: The control input applied to the system depends on the arrival of the update, which depends on the transmission success and on the decision to send an update, as described in Eq. (2). If an update is received, then the control law is a state feedback with gain K . This gain is chosen so that the system $x_{k+1} = (A - BK)x_k$ is stable. Otherwise, the control input is held to its previous value as long as no update is received from the smart node.

$$u_k = \begin{cases} -\beta_k K x_k + (1 - \beta_k) u_{k-1}, & \text{if } v_k = 1, \\ u_{k-1}, & \text{otherwise.} \end{cases} \quad (2)$$

4) *Radio chip model*: The radio chip is characterized by the number of radio-modes, N , and the associated costs to stay in a given mode or to switch from a mode to another. The mode transition delays are assumed to be smaller than the sampling time. This implies that the smart node has enough time to switch to the desired mode (and possibly send the update) before the next sampling time.

The first radio-mode (Tx mode) is the only one allowing transmission, and the most consuming one¹. The other modes are intermediate modes where only some components of the radio are turned off, consuming less energy than the Tx mode. The last radio-mode (called the Sleep mode) consumes no or very little energy (less than any other mode), but more energy is needed to switch to the Tx mode from the Sleep mode than from the intermediate modes. We define θ_{ij} as the energy needed to switch from the mode i to the mode j , and θ_{ii} as the energy to stay in the mode i .

The state of the radio chip is the mode at time k , m_k :

$$m_k \in \mathbb{M} \triangleq \{1, 2, \dots, N\},$$

where $1 \equiv \text{Tx}$, and $N \equiv \text{Sleep}$, and we define $\mathbb{M}^* \triangleq \{2, 3, \dots, N\}$. The radio-mode is updated according to the switching decision: $m_{k+1} = v_k$.

The consumption of the radio chip at each sampling time depends on the radio-mode m_k and on the switching decision v_k . The amount of energy E consumed since the commissioning (where $E_0 = 0$) can be computed as follows:

$$E_{k+1} = E_k + \theta_{m_k v_k} = E_k + \theta_{m_k m_{k+1}}.$$

5) *Switching policy*: The sensor node embeds a switching policy η (whose design is the goal of this paper) to assign the radio-mode. The decision to switch between modes is based on the actual system output x_k , the last control input u_{k-1} and the current mode m_k . Note that the smart sensor has a perfect knowledge of the last control input applied to the system, since it receives an acknowledgment when a transmission is successful. The channel from the actuator to the smart sensor is considered reliable since enough energy is available at the actuator side to send the acknowledgement. Introducing $\tilde{u}_k = u_{k-1}$, the memory of the last control input applied to the system, the switching decision is given by $v_k = \eta(x_k, \tilde{u}_k, m_k)$.

¹The smart node is not receiving any other information than the acknowledgements from the actuator, taken into account in θ_{11} , so we do not consider any receiving mode in this setup.

B. Switched system formulation and optimization problem

We formulate the evolution of the system under the different choices of radio-modes as a switched linear system, with as many systems as the number of modes N . Choosing the switching policy at time k is equivalent to choosing the radio-mode. The evolution of the switched system depends on x_k , the state of the system, on $\tilde{u}_k = u_{k-1}$, a memory keeping track of the last applied control input, and on m_k the mode of the radio chip. We define z_k as the system state augmented with the control memory:

$$z_k = [x_k \quad \tilde{u}_k]^\top \in \mathbb{R}^{n+p}, \text{ and also } \omega_k = [w_k \quad \mathbf{0}]^\top.$$

Then, the state of the switched system is (z_k, m_k) and the state space is denoted $\mathbb{X} \triangleq \mathbb{R}^{n+p} \times \mathbb{M}$. The evolution of the system given in Eq. (1) and the control law described in Eq. (2), together with the radio-mode update law η , give rise to the following switched system:

$$\begin{cases} z_{k+1} = f_{v_k}(z_k, \omega_k, \beta_k) \\ m_{k+1} = v_k = \eta(z_k, m_k), \end{cases}$$

where the function f_{v_k} is defined as

$$f_{v_k}(z_k, \omega_k, \beta_k) = \Phi_{v_k}(\beta_k)z_k + \omega_k, \quad (3)$$

and the matrices $\Phi_{v_k}(\beta_k)$, for $v_k \in \mathbb{M}$, are as follows:

1) if $v_k = 1$, i.e., if there is a transmission, then

$$\Phi_1(\beta_k) = \begin{bmatrix} A - \beta_k BK & (1 - \beta_k)B \\ -\beta_k K & (1 - \beta_k)\mathbf{I} \end{bmatrix} \quad (4)$$

$$= \begin{cases} \Phi_{CL} = \begin{bmatrix} A - BK & \mathbf{0} \\ -K & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ \Phi_{OL} = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \text{if } \beta_k = 0. \end{cases} \quad (5)$$

2) if $v_k = j \neq 1$, i.e., if there is no transmission, then

$$\Phi_j(\beta_k) = \Phi_{OL} \quad \forall \beta_k. \quad (6)$$

Our goal is to find a suitable switching policy η , in order to get a good trade-off between the control performance and the energy consumption. To this aim, we define an optimization problem on an infinite time-horizon, where the cost function takes into account the mean of these two criteria:

$$J_\eta(z_0, m_0) = \mathbb{E}_{\{\beta_k\}, \{\omega_k\}} \left[\sum_{k=0}^{\infty} \lambda^k \ell_{v_k}(z_k, m_k, \beta_k) \right],$$

where $m_{k+1} = v_k = \eta(z_k, m_k)$ and $z_{k+1} = f_{v_k}(z_k, \omega_k, \beta_k)$. Here, $\lambda > 0$ is a discount factor, and $\ell_{v_k}(z_k, m_k, \beta_k)$ is the cost-to-go, which is by construction a random variable, designed as follows:

$$\ell_{v_k}(z_k, m_k, \beta_k) = \underbrace{x_k^\top \bar{Q} x_k}_{\text{performance}} + \underbrace{u_k^\top \bar{R} u_k}_{\text{control energy}} + \underbrace{\theta_{m_k v_k}}_{\text{transmission energy}},$$

for some symmetric, positive definite matrices \bar{Q} and \bar{R} , which can be tuned to give different trade-offs between the feedback performance and energy consumption.

Recalling that the control input u_k satisfies Eq. (2), the cost-to-go can be re-written in the following form, which clarifies that it depends on z_k, m_k, v_k and β_k only:

$$\ell_{v_k}(z_k, m_k, \beta_k) = z_k^\top Q_{v_k}(\beta_k) z_k + \theta_{m_k v_k}, \quad (7)$$

where the matrices $Q_{v_k}(\beta_k)$, for $v_k \in \mathbb{M}$, are as follows:

1) if $v_k = 1$, i.e., if $u_k = -\beta_k K x_k + (1 - \beta_k) \tilde{u}_k$, then

$$Q_1(\beta_k) = \begin{bmatrix} \bar{Q} + \beta_k K^\top \bar{R} K & \mathbf{0} \\ \mathbf{0} & (1 - \beta_k) \bar{R} \end{bmatrix} \\ = \begin{cases} Q_{CL} = \begin{bmatrix} \bar{Q} + K^\top \bar{R} K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{if } \beta_k = 1 \\ Q_{OL} = \begin{bmatrix} \bar{Q} & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} & \text{if } \beta_k = 0. \end{cases} \quad (8)$$

2) if $v_k = j \neq 1$, i.e., if $u_k = \tilde{u}_k$, then

$$Q_j(\beta_k) = Q_{OL} \quad \forall \beta_k. \quad (9)$$

The optimization problem is summarized as follows.

Problem.

Find a deterministic stationary policy $\eta^*(z_k, m_k)$ such that

$$J_{\eta^*}(z_0, m_0) = \min_{\eta} J_{\eta}(z_0, m_0),$$

where the cost J_{η} is defined as

$$J_{\eta}(z_0, m_0) = \mathbb{E}_{\{\beta_k\}, \{\omega_k\}} \left[\sum_{k=0}^{\infty} \lambda^k \ell_{v_k}(z_k, m_k, \beta_k) \right],$$

where $m_{k+1} = v_k = \eta(z_k, m_k)$, $z_{k+1} = f_{v_k}(z_k, \omega_k, \beta_k)$ as defined in Eq.s (3)-(6), $\ell_{v_k}(z_k, m_k, \beta_k)$ is the cost-to-go described by Eq.s (7)-(9) and $\lambda \in (0, 1)$ is a discount factor.

Note that we are interested in solving this problem under the following non-triviality assumptions: the system described by Eq. (1) is not stable; the matrix $A - BK$ is not nilpotent; and transmissions have a non-zero cost whatever the previous mode, i.e., $\theta_{m1} > 0 \quad \forall m$.

We are only searching for a stationary policy η^* , because a time-dependent policy η_k on an infinite time-horizon is not implementable. Fortunately, as it is explained in Section III, a stationary policy exists, which is optimal among all policies.

The discount factor λ is used to weight the importance of immediate action versus long-term decision. It is mandatory in our setup because when $\lambda = 1$, on an infinite time-horizon, the cost function is infinite for any policy. However, it has to be noticed that introducing the discount factor $\lambda < 1$ prevents from proving stability with the standard argument in LQ optimal control, using the cost-to-go as a Lyapunov function. To the authors knowledge, stability proof in discounted problems solved with Dynamic Programming is an open issue and beyond the scope of this contribution².

²Authors in [8], [3] discuss about stability of control laws derived using Dynamic Programming based approaches.

III. SOLUTION OF THE OPTIMIZATION PROBLEM BY DYNAMIC PROGRAMMING

A. The Value Iteration method

The optimization problem described in Section II-B can be solved using Dynamic Programming, which is based on Bellman's Principle of Optimality [2]. This methodology is composed of two parts. The first one is run offline, and it provides the optimal switching policy. This part involves heavy computations that can be carried on a computer. It gives an exact solution for every $(z, m) \in \mathbb{X}$. The second part, run online on the smart sensor, consists in computing the optimal switching decision only for the current switched system state (z_k, m_k) at time k , according to the law derived offline.

Notice that the cost-to-go satisfies the positivity assumption, *i.e.*, $\ell_v(z, m, \beta) \geq 0$ for all $(z, m) \in \mathbb{X}$, $v \in \mathbb{M}$ and β . Also notice that the policy $\eta(z, m)$ takes values in the finite set \mathbb{M} . Thanks to these two facts, standard arguments in Dynamic Programming theory (see [4, Ch. 9]) allow to prove that there exists a deterministic stationary policy η which minimizes the cost $J_\eta(z_0, m_0)$, and which can be found by the so-called Value Iteration method, *i.e.*, by the following iterative algorithm:

after initializing $V_0(z, m)$ to be the all-zero function, compute

$$V_{i+1}(z, m) = \min_{v \in \mathbb{M}} \left\{ \mathbb{E}_{\beta, \omega} [\lambda V_i(f_v(z, \omega, \beta), v) + \ell_v(z, m, \beta)] \right\}. \quad (10)$$

The results in [4, Ch. 9] guarantee that such iterations will converge to $V^*(z, m) = J^*(z, m) \triangleq \min_{\eta} J_\eta(z, m)$ as i goes to infinity, and that

$$\eta_i(z, m) \triangleq \arg \min_{v \in \mathbb{M}} \left\{ \mathbb{E}_{\beta, \omega} [\lambda V_i(f_v(z, \omega, \beta), v) + \ell_v(z, m, \beta)] \right\}$$

converges to $\eta^*(z, m)$, the optimal stationary policy.

The derivation of the optimal switching policy consists in computing offline $V^*(z, m)$ and $\eta^*(z, m)$ for all $(z, m) \in \mathbb{X}$. Then, the smart sensor computes online, at each sampling instant, the switching decision v_k as a function of (z_k, m_k) , according to the law η^* .

B. Computation of the Value Function iterations

In theory, the Value Iteration Method gives us an iterative algorithm, converging to the solution of our optimization problem. A caveat is that, at each iteration, we need to compute a function of (z, m) , where (z, m) takes values in an uncountable space \mathbb{X} .

A first way to implement such iterations in practice, is to partition (a portion of) \mathbb{X} in a grid, then compute the Value Function $V_{i+1}(z, m)$ at the grid points only, by using interpolation to find $V_i(f_v(z, \omega, \beta), v)$ when $f_v(z, \omega, \beta)$ is not on the grid. This approach has been taken *e.g.* in [19] and provides a look-up table for $\eta_i(z, m)$ at all grid points. Computationally, it is very heavy, although this is not a major issue since the long computations are done offline, while online the control is chosen by accessing the look-up table. Some minor drawbacks of this approach are that

the solution is limited to a finite domain, and that the numerical approximations and interpolation might introduce some errors giving a sub-optimal solution.

A second approach, inspired both by the results in classic Linear Quadratic (LQ) optimal control and by the work in [14], is to try to compute the functions $V_i(z, m)$ by exploiting some structure that they might have, if any structure exists that is preserved along iterations. For example, in LQ control, for any i the Value Function is a quadratic function of z , *i.e.*, $V_i(z) = z^\top \Pi_i z$. For our problem, at least in the case without noise and packet drops, it is possible to find a structure, although more involved than the classical LQ case, as described below. In the remainder of this subsection, we focus on the simplified problem where there is neither noise ($\omega = 0$) nor packet drops ($\beta = 1$), and we use the following notation: $f_v(z) \triangleq f_v(z, 0, 1)$, $\ell_v(z, m) \triangleq \ell_v(z, m, 1)$, $\Phi_v \triangleq \Phi_v(1)$. With these assumptions and notation, the following expressions hold true.

1) *Structure of the Value Function:* If $V_0(z, m) \equiv 0 \forall (z, m)$, then the iterations (10) give value functions $V_i(z, m)$ such that

$$V_i(z, m) = \min_{(\Pi, \vec{\pi}) \in \mathcal{P}_i} \{z^\top \Pi z + \pi_m\}, \quad (11)$$

where the set \mathcal{P}_i is composed of elements $(\Pi, \vec{\pi})$, where Π is a symmetric matrix and $\vec{\pi} = [\pi_1, \pi_2, \dots, \pi_N] \in \mathbb{R}^N$ is a vector of non-negative scalars, and π_m represents the m^{th} component of $\vec{\pi}$.

Indeed, assuming that Eq. (11) holds, Eq. (10) yields

$$\begin{aligned} V_{i+1}(z, m) &= \min_{v \in \mathbb{M}} \{ \lambda V_i(f_v(z), v) + \ell_v(z, m) \} \\ &= \min_{v \in \mathbb{M}} \left\{ \lambda \min_{(\Pi, \vec{\pi}) \in \mathcal{P}_i} \{ z^\top \Phi_v^\top \Pi \Phi_v z + \pi_v \} + z^\top Q_v z + \theta_{mv} \right\} \\ &= \min_{v \in \mathbb{M}, (\Pi, \vec{\pi}) \in \mathcal{P}_i} \{ z^\top (\lambda \Phi_v^\top \Pi \Phi_v + Q_v) z + \lambda \pi_v + \theta_{mv} \} \\ &= \min_{(\Pi', \vec{\pi}') \in \mathcal{P}_{i+1}} \{ z^\top \Pi' z + \pi'_m \}. \end{aligned}$$

2) *Iterations of the Value Function:* The computation of $V_{i+1}(z, m)$ at iteration $i + 1$ consists in computing the set \mathcal{P}_{i+1} , from the set \mathcal{P}_i :

$$\begin{aligned} \mathcal{P}_0 &= \{(\mathbf{0}, [0, 0, \dots, 0])\} \\ \mathcal{P}_{i+1} &= \mathcal{P}_{i+1}^1 \cup \mathcal{P}_{i+1}^2, \text{ where:} \end{aligned} \quad (12)$$

$$\begin{aligned} \mathcal{P}_{i+1}^1 &\triangleq \left\{ \left((\lambda \Phi_{CL}^\top \Pi \Phi_{CL} + Q_{CL}), \right. \right. \\ &\quad \left. \left. [\lambda \pi_1 + \theta_{11}, \lambda \pi_1 + \theta_{21}, \dots, \lambda \pi_1 + \theta_{N1}] \right) \text{ such that } (\Pi, \vec{\pi}) \in \mathcal{P}_i \right\}, \\ \mathcal{P}_{i+1}^2 &\triangleq \left\{ \left((\lambda \Phi_{OL}^\top \Pi \Phi_{OL} + Q_{OL}), \right. \right. \\ &\quad \left. \left. \left[\min_{v \in \mathbb{M}^*} \{ \lambda \pi_v + \theta_{1v} \}, \min_{v \in \mathbb{M}^*} \{ \lambda \pi_v + \theta_{2v} \}, \dots, \min_{v \in \mathbb{M}^*} \{ \lambda \pi_v + \theta_{Nv} \} \right] \right) \right. \\ &\quad \left. \text{such that } (\Pi, \vec{\pi}) \in \mathcal{P}_i \right\}. \end{aligned}$$

The switching policy at iteration i is given by:

$$\eta_i(z, m) \triangleq \begin{cases} 1 & \text{if } (\Pi_i, \vec{\pi}_i) \in \mathcal{P}_i^1, \\ \arg \min_{v \in \mathbb{M}^*} \{ \lambda \pi_v + \theta_{mv} \} & \text{else,} \end{cases}$$

where $(\Pi_i, \vec{\pi}_i) = \arg \min_{(\Pi, \vec{\pi}) \in \mathcal{P}_i} \{ z^\top \Pi z + \pi_m \}$.

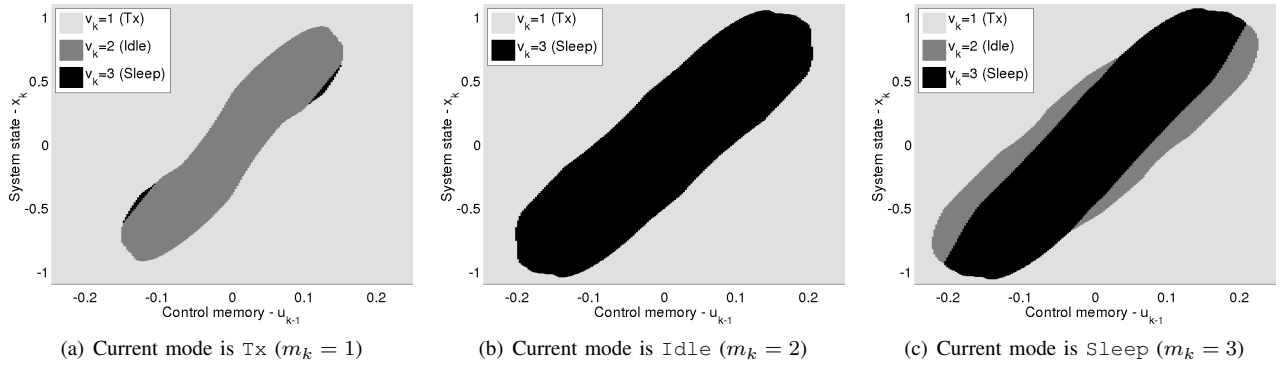


Fig. 2. Optimal switching policy derived from the offline computation, light gray \Leftrightarrow switch to Tx ($v_k = 1$), dark gray \Leftrightarrow switch to Idle ($v_k = 2$) and black \Leftrightarrow switch to Sleep ($v_k = 3$), the control memory $\tilde{u}_k = u_{k-1}$ is on the x-axis and the system output x_k on the y-axis.

3) *Convergence of the Value Function*: The Value Function iterations guarantee that the optimal cost J^* and the optimal switching policy η^* are attained as the number of iterations goes to infinity. This is guaranteed besides having the cardinality of the set \mathcal{P}_i doubled at each iteration (as it can be seen from Eq. (12)). When running the offline computation, the stopping criterion of the iterations is then a trade-off between the complexity of the computations (increasing with the iterations) and the distance to optimality (decreasing with the iterations). One proposition to observe the convergence of the scheme is to define a grid on a finite domain of interest of the state space \mathbb{X} on which, at each iteration, the value function $V_i(z, m)$ is evaluated. The value function evaluated on the grid converges at a rate depending on the precision of the grid, and this is therefore only an indicator.

IV. SIMULATION RESULTS

To illustrate the proposed method, we present here an example of a first order unstable system, with the following parameters:

$$\begin{aligned} x_{k+1} &= 1.074x_k - 1.4808u_k + w_k; & T_s &= 0.05\text{s}; \\ \bar{W} &= 0.02; & K &= -0.23; & \bar{Q} &= 0.01; & \bar{R} &= 0.1; \\ \theta &= \begin{bmatrix} 2.85 & 1.8 & 1.9 \\ 3.2 & 1.4 & 6 \cdot 10^{-5} \\ 3.5 & 3.7 \cdot 10^{-3} & 6 \cdot 10^{-5} \end{bmatrix}; & N &= 3 \\ & & \lambda &= 0.8 \\ & & \epsilon &= 0.3 \end{aligned}$$

Three radio-modes are considered (Tx, Idle, Sleep), the values of the transition costs are given in the θ matrix in [mJ] and are computed from the datasheet of the radio chip Texas Instrument CC1100.

A. Offline results

The offline computation provides the switching policy $v = \eta^*(x, \tilde{u}, m)$. This means that, at any time k , η^* gives the optimal switching decision $v^* \in \mathbb{M}$ for the current measurement x_k , the last control input applied to the system $u_{k-1} = \tilde{u}_k$ and the current radio-mode m_k . This is depicted in Fig. 2 with a subfigure per each current mode m_k .

We observe that the regions where the radio is switched to low consuming modes (colors dark gray and black in Fig. 2) are finite sets around the equilibrium point $(x_k, \tilde{u}_k) = (0, 0)$, and follow the direction $\tilde{u}_k = -Kx_k$ ($K < 0$ in our

example). Outside of these regions, a transmission is forced. This means that when the last control input applied to the system is close to what the state-feedback law would have decided, then the switching policy does not send an update.

Note that we obtain an event-based radio-mode switching policy. Indeed, a switching occurs only when the state of the system crosses one of the regions in Fig. 2.

B. Online results

After deriving the switching policy, we run online temporal simulations to observe the behaviour of the system. In Fig. 3 we compare our event-based switching policy with some periodical ones using the same state feedback law $u = -Kx$, where the radio is alternatively switched to Tx and low consuming modes. We consider various periodic patterns for the radio mode: we will denote by periodic i - j a sequence with period $i+j$ where the mode is Tx for i consecutive sampling intervals and then is Sleep for j intervals. We will then denote by periodic i - j min a sequence with the same period and the same Tx intervals (so that the control performance is unchanged), but where the mode for the non-transmitting intervals is chosen in \mathbb{M}^* so as to minimize the energy consumption. The online simulations include channel dropouts and additive output noise on the system.

Fig. 3(a) shows that the system is stabilized in a set around the equilibrium in both cases. In Fig. 3(b), one can see the switching decisions for both cases. The green triangles indicate time instants where a transmission was intended but a dropout occurred. When an update is dropped, the event-based scheme holds the Tx mode to actually transmit a new control update, while the periodic scheme is not taking dropouts into account. Moreover, the event-based scheme may hold the Sleep mode for a long time interval when transmissions cost more than the deviation observed on the state.

Finally Fig. 3(c) compares the event-based scheme to several periodic patterns. In this figure, the performance of the closed loop system is computed from the cost function $(\sum_{i=0}^{T_{\max}} (x_i^\top \bar{Q} x_i + u_i^\top \bar{R} u_i))$ where $T_{\max}=4\text{s}$ and normalized such that 100% and 0% are the best and the worst performance, respectively. The event-based scheme is very close to the best, although not exactly the best, but it offers the least energy consumption, and especially the best trade-off.

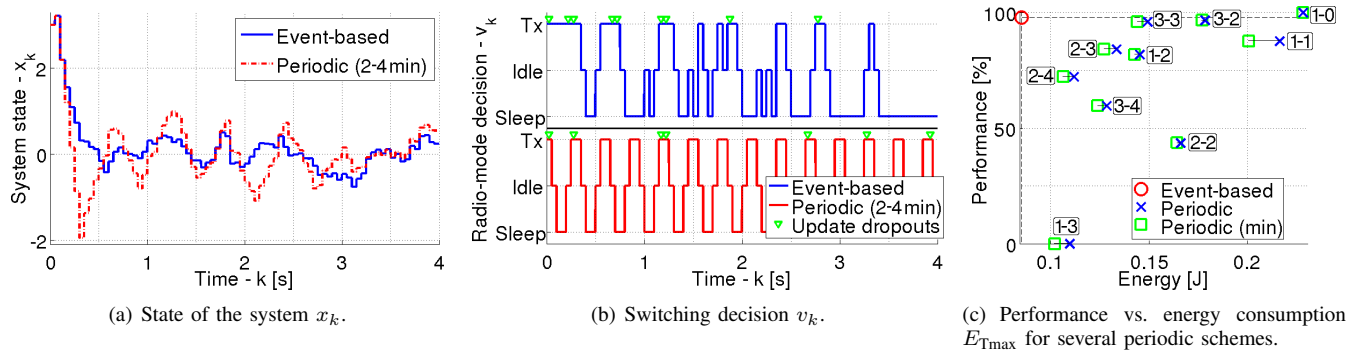


Fig. 3. Online simulations comparing our event-based switching policy with periodic ones. Additive zero-mean white Gaussian noise and channel dropouts are considered. The green triangles in Fig (b) indicate time instants where a transmission was intended but a dropout occurred.

V. CONCLUSION

In this paper we have studied the optimal management of the radio-chip modes of a wireless smart sensor in a networked control problem. The novelty of this paper is to introduce the use of more than two radio-modes in a control problem, whereas previous related control-theoretic literature was focused on the choice between two options (Tx/Sleep). We have considered a networked control problem with an unstable linear system to be stabilized, and with a single smart sensor whose transmissions to the actuator are performed with an optimal choice of the radio-mode.

For this problem, we have defined a suitable cost function, which describes a trade-off between the control performance and the energy consumption, and whose minimum can be computed with an iterative Dynamic Programming algorithm (Value Iteration method). We obtained an event-based policy to switch between radio-modes. Limiting the amount of communication to save energy naturally decreases the closed-loop performance, but we show on an example that an event-based approach permits to keep the performance good and save larger amount of energy than a periodic approach.

A natural extension of this work is to consider an optimization that involves not only the radio-mode, but also the feedback control law for those times where the sensor is transmitting. Moreover, we are currently exploring another approach to this same problem, where the optimization is performed over a finite receding horizon. This MPC framework has the advantage to provide tools to prove stability in the practical-Input-to-State stability sense.

VI. ACKNOWLEDGMENTS

The authors would like to thank Daniel Quevedo for interesting discussions.

REFERENCES

- [1] L. Bao, M. Skoglund, and K.H. Johansson. Encoder decoder design for event-triggered feedback control over bandlimited channels. In *American Control Conference, 2006*, pages 4183–4188, Jun. 2006.
- [2] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [3] D. P. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. In *Fundamental Issues in Control, European J. of Control*, volume 11, 2005.
- [4] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*. Athena Scientific, 1978.
- [5] M. I. Brownfield, T. Nelson, S. Midkiff, and N. J. Davis. Wireless sensor network radio power management and simulation models. *The Open Electrical & Electronic Engineering Journal*, 4:21–31, 2010.
- [6] R. Cogill. Event-based control using quadratic approximate value functions. In *Proceedings of the 48th IEEE Conference on Decision and Control, CDC'09*, pages 5883–5888, Dec. 2009.
- [7] R. Cogill, S. Lall, and J.P. Hespanha. A constant factor approximation algorithm for event-based sampling. In *American Control Conference, 2007. ACC '07*, pages 305–311, 9-13 2007.
- [8] D. Gorges, M. Izak, and S. Liu. Optimal control and scheduling of switched systems. *IEEE Transactions on Automatic Control*, 2011.
- [9] V. Gupta, T. Chung, B. Hassibi, and R. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, Feb 2006.
- [10] O. C. Imer and T. Basar. Optimal estimation with limited measurements. *Int. J. Syst., Control Commun.*, 2:5–29, Jan. 2010.
- [11] G. Joshi, S. Jardosh, and P. Ranjan. Bounds on dynamic modulation scaling for wireless sensor networks. In *3rd International Conference on Wireless Communication and Sensor Networks*, pages 13–16, Dec. 2007.
- [12] R. Jurdak, A.G. Ruzzelli, and G. O'Hare. Adaptive radio modes in sensor networks: How deep to sleep? In *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 386–394, Jun. 2008.
- [13] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2:8–13, 2005.
- [14] B. Lincoln. *Dynamic Programming and Time-Varying Delay Systems*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, May 2003.
- [15] C. R. Mann, R. O. Baldwin, and B. E. Mullins. Wireless sensor networks: guidelines for design and a survey of current research. In *SpringSim*. Dept. of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, USA, 2007.
- [16] D. E. Quevedo, A. Ahlén, and J. Ostergaard. Energy efficient state estimation with wireless sensors through the use of predictive power control and coding. *IEEE Transactions on Signal Processing*, 58(9):4811–4823, Sept. 2010.
- [17] A. Sinha and A. Chandrakasan. Dynamic power management in wireless sensor networks. *Design & Test of Computers, IEEE*, 18(2):62–74, Mar/Apr 2001.
- [18] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, Sept. 2004.
- [19] O. Sundström, D. Ambühl, and L. Guzzella. On implementation of dynamic programming for optimal control problems with final state constraints. *Oil Gas Sci. Technol. - Rev. IFP*, 65(1):91–102, 2010.
- [20] R. Wen, I. Mareels, and B. Krongold. Optimal power analysis for network lifetime balance in wireless sensor networks. In *International Symposium on Communications and Information Technologies*, pages 1161–1165, Oct. 2007.
- [21] Y. Xu and J.P. Hespanha. Estimation under uncontrolled and controlled communications in networked control systems. In *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05*, pages 842–847, Dec. 2005.