52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Power-aware communication for wireless sensor-actuator systems

Konstantinos Gatsis, Miroslav Pajic, Alejandro Ribeiro, and George J. Pappas

Abstract— **This paper considers the design of power-aware communication protocols for a sensor transmitting plant state measurements over a wireless Markov fading channel to a receiver/controller. Communication requires power consumption at transmission adapted to channel fading, and at the receiver, which we model as constant at each transmission. We measure performance with a weighted sum of the average power consumption at both ends and an appropriately defined control task error. We derive an optimal self-triggered protocol where after each transmission devices decide when the next one will take place and switch to a zero-power sleep mode in between. We show that sleep durations need to adapt only to the current channel fading and not the plant state. We then derive an improved protocol allowing the sensor upon wake-up to decide whether to transmit or not based on current plant and channel conditions in an event-based fashion. The power/control performance improvements are illustrated in simulations.**

I. Introduction

Modern control systems are often implemented over wireless networks including sensors and actuators with limited energy resources. Earlier work on networked control systems focused on designing efficient control and estimation schemes under various communication interfaces [1], [2] without accounting for the communication costs. The event-triggered paradigm has been recently introduced for a control system design with low communication requirements.

In event-triggered designs a sensor (or a controller) decides in an on-line fashion, based on the current system state, whether to send new plant measurements (or control inputs) over the network. By appropriate design of the decision rule, communication is used only when necessary, e.g., when the problem state exceeds some threshold. This leads to non-periodic implementations that typically exhibit lower communication rates than the standard periodic ones. Such rules can be designed using Lyapunov techniques [3]–[5], where no transmission is triggered as long as some Lyapunov performance criterion is guaranteed, or dynamic programming formulations where each transmission is penalized by a fixed cost [6]–[9]. For a wireless setup we follow the latter formulation in [10] and explicitly identify the communication cost with the power consumption for transmitting over a random wireless fading channel. In analogy to the event-triggered transmit-or-not rule, we obtain a communication protocol where the transmit power needs to adapt not only to the system state but also to the wireless channel conditions.

This work was supported in part by NSF CNS-0931239, and by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Email: {kgatsis, pajic, aribeiro, pappasg}@seas.upenn.edu.

However, implementing event-based designs in a wireless setup requires the receiver to be continuously listening for possible packet transmissions. Wireless networking experiments [11]–[13] indicate that listening and transmitting over a wireless channel results in a similar power consumption. Thus a communication protocol that turns off the receiver for some time interval after each transmission could reduce power consumption. Such a design is reminiscent of the self-triggered control paradigm [14], where the goal is to use the current system state to choose a constant plant input to be applied, as well as the time of the next control input update.

In this paper we introduce a power-aware communication protocol combining both self and event-triggered elements. We present in Section II our architecture consisting of a sensor transmitting plant state measurements over a Markov wireless fading channel to a controller. As in [10] we explicitly model the sensor's transmit power allowing adaptation to current channel fading and we model the receiver's power when active as constant on each transmission slot. With a fixed controller design, we evaluate the control performance between transmissions by comparing the real plant state with what it would be if the sensor were transmitting at each step. Our goal is to minimize a cost combining the total average control error and power consumption, by designing a communication protocol that controls transmit power and switches the devices to a sleep mode for adaptive time intervals.

In Section III we design an optimal self-triggered protocol and show that, unlike self-triggered control [14], the inter-transmission interval does not depend on the plant state but only on the current channel state. Then in Section IV, keeping this protocol as a reference, we propose a novel one that deviates from the reference by deciding upon wake-up whether to transmit or not in an event-based fashion. The decision is based on current plant and channel states, and guarantees that the future cost will not exceed the reference one. Simulations in Section V illustrate the improved power/control performance trade-off of the proposed protocol and indicate that self-triggered protocols are more advantageous when the receiver has high power consumption. We conclude in Section VI by discussing some implementation issues and future research directions.

Notation: A set of variables $\{\gamma_k, \gamma_{k+1}, \dots, \gamma_{k+t}\}$ is grouped as $\gamma_{k:k+t}$. Subscripts in x_{k-1}, x_k, x_{k+1} denote discrete time and are omitted as x_-, x, x_+ when current time index k is clear from the context.

II. Problem Setup

Consider a discrete-time linear time-invariant plant

$$x_{k+1} = Ax_k + Bu_k + w_k,\ k \geq 0, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the plant's state with x_0 given, $u_k \in \mathbb{R}^m$ is the control input, and $\{w_k \in \mathbb{R}^n, k \geq 0\}$ is an independent and identically distributed Gaussian process noise with zero mean and covariance W. The wireless control system consists of a sensor/transmitter collecting state measurement x_k and transmitting a packet containing x_k with power p_k over a wireless fading channel with coefficient h_k. At the other side of the channel the receiver/controller uses the received information to determine the plant control input u_k.

Due to propagation effects the channel coefficient h_k changes unpredictably [15, Chapter 3], forming a stochastic process $\{h_k, k \geq 0\}$, independent of the plant noise process $\{w_k, k \geq 0\}$. We adopt a Markov modeling whereby the future channel distribution depends only on the current channel state via a transition probability $\mathbb{P}(h_{k+1}|h_k)$. For simplicity let channel states take values in a finite set $\mathcal{H} := \{h^1, \dots, h^L\}$ and group the transition probabilities in an $L \times L$ stochastic matrix M according to

$$\mathbb{P}(h_{k+1} = h^j | h_k = h^\ell) = M_{\ell j}. \tag{2}$$

We assume the Markov chain is irreducible and aperiodic.

At the controller side the received signal consists of the information bearing signal whose power is given by the product $h_k \, p_k$, and additive white Gaussian noise of power N_o. Successful decoding of the transmitted packet depends on the received signal to noise ratio (SNR) defined as $h_k p_k / N_o$. Given the particular type of modulation and forward error correcting (FEC) code used, the SNR determines the probability of successful decoding at the receiver [10]. For modern FEC codes, e.g., turbo codes, that achieve very steep error functions we approximately have a successful packet decoding if SNR is above some threshold and a failed packet decoding otherwise.

In our model, a packet containing x_k is successfully received at the controller if $h_k \, p_k \geq p_0$, and is lost otherwise, with p_0 being some given power level threshold. We assume the sensor has enough transmit power available to guarantee delivery even for the worst channel fading in $\mathcal{H}$. We also assume that the current channel h_k is measured at the beginning of a transmission slot k and is known at both ends, allowing adaptation of transmit power p_k to h_k. Thus the sensor need only choose $p_k = p_0 / h_k$ if it intends to transmit and $p_k = 0$ otherwise. Alternative channel estimation and adaptation setups are discussed in Section VI.

At the other side of the link, the receiver also consumes power to stay awake and listen for the incoming signal on the predefined channel. We model this power by a fixed constant p_a. The devices have the option to switch to a sleep mode with zero power consumption but then no transmission is possible. The total power consumed at slot k is given by

$$p'_k := \begin{cases} p_a + p_k & \text{if awake at } k, \\ 0 & \text{if in sleep mode at } k. \end{cases} \tag{3}$$

Overall the communication can be described by a sequence of indicator variables γ_k, taking value $\gamma_k = 1$ when a packet is received and $\gamma_k = 0$ otherwise, i.e.,

$$\gamma_k := \begin{cases} 1 & \text{if awake at } k \text{ and } h_k \, p_k \geq p_0, \\ 0 & \text{otherwise} \end{cases} . \tag{4}$$

We also let the controller feedback acknowledgment packets containing γ_k to the sensor, as provided by 802.11 and TCP protocols, so that the sensor always knows what information has been received at the controller.

Consider then a given control feedback gain K designed to yield stability and desired plant performance if input $u_k = K x_k$ is applied. Due to the communication protocol, however, the receiver/controller has access to the plant state x_k only when $\gamma_k = 1$. Let then the controller keep an estimate $\hat{x}_k$ of the plant state x_k and apply the input

$$u_k = K \hat{x}_k. \tag{5}$$

We let the state estimate evolve as

$$\hat{x}_k := \begin{cases} x_k & \text{if } \gamma_k = 1 \\ A\hat{x}_{k-1} + Bu_{k-1} & \text{if } \gamma_k = 0 \end{cases} , \tag{6}$$

that is, when no measurement is received $\hat{x}_k$ is updated by propagating the previous estimate through the plant dynamics (1) with process noise replaced by its zero mean.

Our goal is to design communication protocols regulating the transmitter's power levels p_k and the mode of operation (sleep /awake). Such protocols are desired to yield low total power consumption while keeping a satisfying performance of the control task. To define a measure for the latter, consider a time k when a plant state measurement x_k is successfully transmitted ($\gamma_k = 1$), but no packet is received during the following steps ($\gamma_{k+1} = \gamma_{k+2} = \ldots = 0$), so the system is in open loop starting from x_k. If alternatively the sensor had continued transmitting during these steps, the system would have behaved as in a standard closed-loop setup with the process noise as system disturbance. This motivates us to define a hypothetical closed loop system trajectory starting from the plant state at the most recent transmission,

$$x^\circ_{k+1} := \begin{cases} (A + BK)x_k + w_k & \text{if } \gamma_k = 1, \\ (A + BK)x^\circ_k + w_k & \text{if } \gamma_k = 0, \end{cases} \tag{7}$$

with the given initial condition $x^\circ_0 := x_0$. Note that in this definition $\gamma_k = 0$ refers to the actual communication dropout, while x°_{k+1} models what would the state be if $\gamma_k = 1$. We then define the *control error* as the difference between the hypothetical closed loop and the actual trajectory

$$e_k := x^\circ_k - x_k. \tag{8}$$

Intuitively if the plant state is always successfully transmitted the control error is zero. As a side note, other control performance metrics can be incorporated as well. For example, comparing the applied input $K\hat{x}_k$ with the ideal Kx_k gives a control measure $\|K(x_k - \hat{x}_k)\|$ which is proportional to the controller's estimation error - see Remark 1 at the end of Section IV.

Combining the magnitude of the control error e_k with the total power consumption p'_k, we evaluate the performance of a communication protocol by the incurred average infinite-horizon expected cost

$$J := \limsup_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}\left[e_k^T P e_k + \lambda p'_k\right]. \tag{9}$$

Let P be some positive semidefinite matrix and $\lambda > 0$ a constant balancing the control and power considerations. The expectation in this expression is taken over the process noise $\{w_k, k \geq 0\}$ and channel states $\{h_k, k \geq 0\}$.

In Section III we design an optimal communication protocol minimizing (9) within the class of self-triggered protocols. Then, keeping such a protocol as a reference, in Section IV we design a protocol with improved performance that deviates from the reference by appropriately adapting to plant and channel states in an event-triggered fashion.

III. Optimal Self-triggered Protocols

In this section we examine self-triggered communication protocols between the sensor/transmitter and the receiver/controller. Any time the plant state x_k is transmitted, the two devices switch to a sleep mode and wake up after t time steps. Then the new plant state x_{k+t} is sent, and so on. The protocol should guarantee that both devices agree on the same sleep (i.e., inter-communication) time interval when switching to sleep mode. We let t depend on any information available at time k including the current plant, controller, and channel states.

We begin with the following lemma illustrating how the control error evolves between transmissions.[1]

Lemma 1. *Consider the system (1) with the controller described by (5), (6) given the indicators $\{\gamma_k, k \geq 0\}$ of the communication process. Then for any $k \geq 0$, the control error e_k defined by (7),(8) evolves according to*

$$e_{k+1} = \begin{cases} 0 & \text{if } \gamma_k = 1 \\ (A+BK)e_k + BK(x_k - \hat{x}_k) & \text{if } \gamma_k = 0 \end{cases}. \tag{10}$$

Moreover, on the event $\gamma_{k:k+i} = (1, 0, \ldots, 0)$, $i \geq 0$ we have

$$e_{k+i+1} = \sum_{j=0}^{i-1} \left[(A+BK)^{i-j} - A^{i-j}\right] w_{k+j}. \tag{11}$$

The lemma implies that for a self-triggered protocol the control error until the next wake-up time $k+t$ is described by (11) and depends only on the process noise $w_k, w_{k+1}, \ldots$, which are independent of any current information at time k. In particular, they are independent of the current plant state x_k. Thus, unlike the standard self-triggered control paradigm [14], the value of plant state x_k does not play any role in designing the interval t when the goal is to minimize the control error given by (7),(8). On the other hand, the channel h_k during transmission might help predict favorable future channel states, so we let t be a function of h_k. In the remaining of this section we are looking for protocols described as $\tau : \mathcal{H} \to \mathcal{T}$, where $\mathcal{T} := \{1, \ldots, T\}$ and T is some hard upper bound on the sleep duration.

A self-triggered protocol is depicted in Fig. 1. A plant state measurement needs to be transmitted at the time slot when both devices wake up. As described in Section II the current channel h_k is measured by the two devices upon wake-up and the transmitter needs to use $p_k = p_0/h_k$ to guarantee delivery according to (4). Adding the constant power p_a of the receiver the total power consumption at time k of transmission becomes $p_a + p_0/h_k$.

[1]Due to space limitations the proofs of the results are omitted in this paper and can be found in [16].

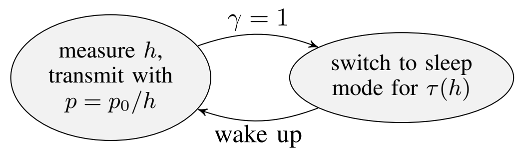

Fig. 1. A self-triggered protocol. Upon wake-up devices measure the current channel state h, plant measurements are transmitted with the appropriate power p_0/h, guaranteeing packet delivery ($\gamma = 1$), and then devices switch to sleep mode for a predefined number $\tau(h)$ of time steps.

We are looking for the optimal protocol τ that minimizes the average performance J given in (9), i.e.,

$$\tau^* := \underset{\tau:\mathcal{H}\to\mathcal{T}}{\operatorname{argmin}} J(\tau). \tag{12}$$

We will find τ^* by leveraging an equivalence to an appropriately constructed Markov Decision Process (MDP). For convenience define the *expected control error and power cost accumulated from the time the devices switch to sleep mode at channel state h^ℓ until they switch again to sleep mode after t slots* by

$$f(h^\ell, t) := \mathbb{E}\left[\sum_{i=1}^{t} e_{k+i}^T P e_{k+i} + \lambda p'_{k+i} \;\;\middle|\; h_k = h^\ell, \gamma_{k:k+t} = (1, 0, \ldots, 0, 1)\right]. \tag{13}$$

Due to the previous lemma this expression indeed depends only on current channel h^ℓ and chosen interval t, and not on the plant state. The expectation is taken over the process noise $w_{k:k+t-1}$ as well as the channel process $h_{k+1:k+t}$. Given the event $\gamma_{k:k+t} = (1, 0, \ldots, 1)$ the control error cost in (13) can be derived by substituting (11) for $i = 0, \ldots, t-1$ and taking the expectation over $w_{k:k+t-1}$. This gives

$$f(h^\ell, t) = \sum_{i=1}^{t-1} (t-i)\, Tr\left(P H_i W H_i^T\right) + \lambda\, p'(h^\ell, t), \tag{14}$$

where $H_i = (A+BK)^i - A^i$, and $p'(h^\ell, t)$ denotes the expected power consumption at time $k+t$, which will be $p_{k+t} = p_a + p_0/h_{k+t}$. Since $h_k = h^\ell$ is given in (13) and channel states are Markov, h_{k+t} is distributed according to the ℓ^{th} row of M^t, giving the expected power consumption

$$p'(h^\ell, t) := p_a + \sum_{j=1}^{L} \frac{p_0}{h^j} M_{\ell j}^t. \tag{15}$$

Now consider the following construction.

Definition 1. Define the Markov Decision Process with:

- State space $\mathcal{H} \times \mathcal{T}$
- Actions $t \in \mathcal{T}$ available at every state
- Given an action $t \in \mathcal{T}$ transitions are described by

$$\mathbb{P}\left[(h^\ell, s-1) \,\middle|\, (h^\ell, s), t\right] = 1, \quad h^\ell \in \mathcal{H},\ 2 \leq s \leq T \tag{16}$$

$$\mathbb{P}\left[(h^j, t) \,\middle|\, (h^\ell, 1), t\right] = M_{\ell j}^t, \quad h^\ell, h^j \in \mathcal{H}. \tag{17}$$

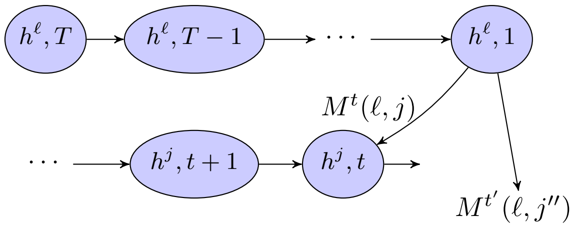

Fig. 2. A representation of the MDP constructed in Definition 1. The process moves from (h^ℓ, T) to $(h^\ell, 1)$ as the counter counts down. At state $(h^\ell, 1)$ different actions $t, t', \ldots \in \mathcal{T}$ are available. If the action t is taken the process moves to one of the states $(h^j, t), (h^{j'}, t), \ldots$ with transition probabilities according to the ℓ^{th} row of the matrix M^t.

- Cost per stage

$$c(h^\ell, s, t) := \begin{cases} f(h^\ell, t) & \text{if } s = 1, \\ 0 & \text{if } 2 \leq s \leq T. \end{cases} \tag{18}$$

A graphical representation of the MDP is shown in Fig. 2. State h^ℓ models the channel state during transmission and state s is a timer counting down till the next transmission. The state $(h^\ell, s = 1)$ models the end of a transmission slot with a current channel h^ℓ when an action $t \in \mathcal{T}$ specifying the next transmission time needs to be taken. At states $(h^\ell, s > 1)$ actions t have no effect and the process moves to $(h^\ell, s-1)$ to reduce counter by 1. It is worth noting that after state $(h^\ell, s = 1)$ with action t the constructed process moves to the channel state h^j that *will* be measured at the next transmission time in t steps, distributed according to the ℓ^{th} row of M^t. Furthermore, the cost at state $(h^\ell, s = 1)$ captures the total accumulated cost $f(h^\ell, t)$ until the next transmission as per (13), while the stage cost is zero as the counter s counts down to 1.

Suppose then that we are looking for a stationary policy $t : \mathcal{H} \times \mathcal{T} \to \mathcal{T}$ in the above MDP that minimizes

$$J_{\text{MDP}}(t) := \limsup_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}\, c(h_k, s_k, t_k). \tag{19}$$

The following proposition establishes the equivalence to our self-triggered communication protocol design problem (12).

Proposition 1. *Let $\tau : \mathcal{H} \to \mathcal{T}$ define a self-triggered communication protocol and $t : \mathcal{H} \times \mathcal{T} \to \mathcal{T}$ be a stationary policy of the MDP defined in Definition 1. If $\tau(h^\ell) = t(h^\ell, 1)$ for all $h^\ell \in \mathcal{H}$ then the corresponding costs $J(\tau)$ according to (9) and $J_{MDP}(t)$ according to (19) are equal.*

The proposition implies that the optimal protocol τ^* in (12) can be derived equivalently from the optimal MDP policy $t^* := \operatorname{argmin}_t J_{\text{MDP}}(t)$ at the points of decision-making $(s = 1)$. Optimal MDP policies can be characterized by standard results in finite state MDPs. In particular, suppose that for any stationary policy t the resulting Markov chain has a single recurrent class. This condition is not restrictive in practice as the Markov channel process is irreducible and aperiodic. Then (see, e.g., [17, Vol.II, Ch. 4.2]) there exists a function $V : \mathcal{H} \times \mathcal{T} \mapsto \mathbb{R}$ and a constant J^* that satisfy the Bellman-like equation

$$V(h^\ell, s) = \min_{t \in \mathcal{T}} \left\{ c(h^\ell, s, t) - J^* + \mathbb{E}[V(h_+, s_+) | h = h^\ell, s, t] \right\} \tag{20}$$

for all h^ℓ, s. Here J^* is the optimal cost of (19), or equivalently (12), and the optimal policy $t^*(h^\ell, s)$ is given by the argument of the right hand side minimization in (20) for each h^ℓ, s. By the established equivalence between $t(h^\ell, 1)$ and $\tau(h^\ell)$ the optimal communication protocol τ^* is the argument of the minimization at the points $(h^\ell, s = 1)$. Substituting in the right hand side of (20) the specific cost and transitions of the MDP at $s = 1$ we find

$$\tau^*(h^\ell) := \operatorname*{argmin}_{t \in \mathcal{T}} \; f(h^\ell, t) - J^* + \sum_{j=1}^{L} V(h^j, t) M^t_{\ell j}. \tag{21}$$

One can readily solve for the triplet $V(h^\ell, s), J^*, \tau^*$ that satisfies (20) employing Value or Policy Iteration algorithms [17, Vol.II, Ch. 4].

The function $V(h^\ell, s)$ in (20) is called the *relative value function* and can be interpreted as the relative value of following the optimal policy t^* when one is at state (h^ℓ, s). Indeed (20) is the standard Bellman's equation for a problem with an infinite-horizon but *non-averaged* objective where the stage cost $c(h^\ell, s, t)$ is reduced by the value J^*, hence the term relative. We will leverage this interpretation of the function $V(h^\ell, s)$ in the following section to construct a new protocol that deviates from τ^* if profitable. In the sequel we only need to consider the values $V(h^\ell, 1)$ at $s = 1$ so we drop the second argument and denote them as $V(h)$.

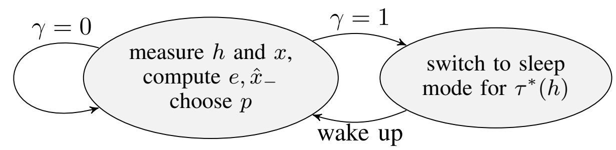

Fig. 3. Proposed protocol based on the optimal self-triggered τ^*. Upon wake-up transmitter adapts power to current channel h as well as all current information - plant state x, control error e, and estimate $\hat{x}_-$. If the sensor transmits, a sleep mode with duration $\tau^*(h)$ follows. If it skips transmission, at the next step the procedure repeats.

IV. Improvements to Self-triggered Protocols

In this section we design protocols that improve upon the self-triggered ones by introducing event-triggered steps between sleep periods. The proposed scheme is shown in Fig. 3 (compare with Fig. 1). When both devices are awake the sensor assesses all current information, including channel and plant states, and decides whether to transmit ($p = p_0/h$) or not ($p = 0$). If it does, the devices follow a self-triggered step and switch to sleep mode according to the optimal $\tau^*(h)$ designed in the previous section, and upon the next wake-up the procedure repeats. On the other hand, if the sensor does not transmit, both devices stay awake and the procedure repeats for the subsequent power p_+. The latter choice might be preferable if, e.g., the current control error e is small and/or the current fading channel h is weak. Note however that an additional power consumption is incurred at the receiver who stays awake for an extra time step.

To decide whether it is profitable to transmit one needs to take into account the future behavior. However it is hard to model all future deviations from the reference τ^*, and thus we adopt a simple approximation depicted in Fig. 4. Suppose

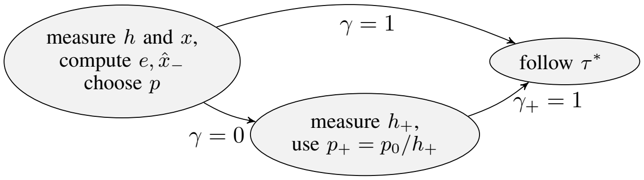

Fig. 4. Model of future behavior after deviation from τ^*. Given all current information, the sensor decides whether to transmit or skip the current slot, assuming that it will transmit at the very next step. In either case the reference protocol τ^* is assumed to follow without deviations.

that if the sensor transmits, which requires $p = p_0/h$, the devices will follow the reference protocol τ^* for all future steps from then on, i.e., without deviations. If on the other hand the sensor decides not to transmit ($p = 0$), at the next step channel h_+ will be measured, the transmitter will send the new plant state x_+ with $p_+ = p_0/h_+$ and from then on the devices will operate according to the reference τ^*.

Based on this model of future behavior we can choose between the two options $p \in \{0, p_0/h\}$ by examining myopically the incurred cost at the current stage, and accounting for the future expected non-averaged cost of following the reference $\tau*$ via the relative value function $V(h)$ derived in the previous section. Let us omit the time indices $k-1, k, k+1$ to simplify the notation. When both devices are awake the sensor measures the current channel h and plant state x, and keeps track of the control error e by (8) and the controller's last estimate $\hat{x}_-$. If the sensor transmits ($p = p_0/h$) the current stage and future expected cost equals

$$V_{\text{tx}}(e, h) := \left[e^T Pe + \lambda(p_a + \frac{p_0}{h}) - J^*\right] + V(h). \quad (22)$$

Here recall that $V(h)$ models the value of following τ^* at the end of a transmission time slot, i.e., when a sleeping period begins.

If the sensor does not transmit ($p = 0$) the current cost is just $e^T Pe + \lambda p_a$. By Fig. 4, at the next step the sensor needs to transmit so we can model the future cost via $V_{\text{tx}}(e_+, h_+)$ that we already defined in the case of (22). Thus the choice $p = 0$ incurs a current stage, expected next stage, and expected future cost equal to

$$V_{\text{skip}}(e, e_+, h) := \left[e^T Pe + \lambda p_a - J^*\right] + \mathbb{E}\left[V_{\text{tx}}(e_+, h_+)|h\right]. \quad (23)$$

The value of h_+ evolves by the Markov channel process (2) while the value e_+ is currently known to the sensor. That is because the choice $p = 0$ implies $\gamma = 0$, so the controller estimate (6) will become $\hat{x} = (A + BK)\hat{x}_-$, and the next control error according to (10) will be $e_+ = (A + BK)e + BK(x - \hat{x})$. All these variables are available at the sensor.

It is important to notice that in order to add the current stage costs to the function $V(h)$ in (22) and (23), they need to be appropriately normalized by the constant J^* as in the right hand side of (20) - see our comments to (20) at the end of the previous section. Now it is profitable to transmit according to the model of Fig. 4 if the condition $V_{tx} < V_{\text{skip}}$ holds. Hence, the optimal myopic decision is

$$p_{\text{myop}} = \begin{cases} p_0/h, & \text{if } V_{\text{tx}}(e, h) < V_{\text{skip}}(e, e_+, h), \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

Upon rearranging terms and removing constants the transmission-triggering condition in (24) for $h = h^\ell$ becomes

$$e_+^T Pe_+ > V(h^\ell) + J^* - \sum_{j=1}^{L} [\lambda(\frac{p_0}{h^j} + p_a - \frac{p_0}{h^\ell}) + V(h^j)] M_{\ell j}, \quad (25)$$

where the only variables are e_+ on the left hand side and h^ℓ at the right hand side.

The power optimization (24) is only myopical at a given time step, and if the model of Fig. 4 were to be followed it would give the same average cost J^*. However we implement the setup of Fig. 3 where profitable deviations from τ^* are allowed at every time step the devices are awake, possibly skipping transmissions repeatedly. There is no guarantee on how much improvement the proposed protocol obtains, but by design the proposed protocol performs at least as good as J^* of the reference τ^*. In addition, simulations in Section V indicate a strictly better performance. Our method can be seen as a variant of rollout algorithms [17, Vol. I, Ch. 6], an approximate dynamic programming technique where, given a base policy with an easily computable cost-to-go, an improved one-step lookahead policy is obtained.

Remark 1. Apart from closed loop control, the introduced protocols can be applied for remote state estimation, where receiver keeps an estimate $\hat{x}_k$ by (6) and estimation errors $\varepsilon_k := x_k - \hat{x}_k$ replace the control errors e_k in our performance metric (9). In analogy to (11) of Lemma 1 the estimation errors during sleep periods $\gamma_{k:k+i} = (1, 0, \ldots, 0)$ are given by $\varepsilon_{k+i} = \sum_{j=0}^{i-1} A^{i-1-j} w_{k+j}$, which are independent of any information at time k, implying again that self-triggered protocols do not need to adapt to current plant state. Thus the designs of Sections III and IV apply with minor modifications. □

V. Simulations

Consider the system with parameters

$$A = \begin{bmatrix} 1.2 & 0 \\ 1 & 0.8 \end{bmatrix}, \; B = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad (26)$$

$W = I$, a controller K for closed loop poles at $0.4, 0.6$, possible channel coefficients $\mathcal{H} = \{0.01, 0.05, 0.1, 0.5, 1\}$, corresponding to a normalized 20dB fading range, and transition probabilities given by

$$M = \begin{bmatrix} 0.6 & 0.4 & 0 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0.2 & 0.6 & 0.6 & 0 \\ 0 & 0 & 0.2 & 0.6 & 0.2 \\ 0 & 0 & 0 & 0.4 & 0.6 \end{bmatrix}. \quad (27)$$

Let $p_0 = 1$ and call $p_{\max} = p_0/h^1$ the required transmit power level for the worst channel. Suppose $p_a = p_{\max}/10$. For different values of λ in (9) we compute the optimal self-triggered protocol τ^* as described in Section III and simulate the proposed improved protocol of Section IV. The results are shown in Fig. 5 with axes corresponding to the average power consumption and the average control error (cf.(9)). The proposed protocol overall yields a better power/control

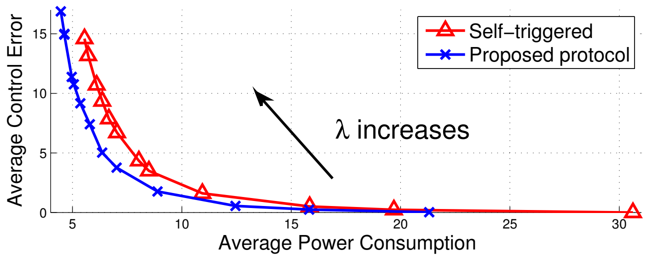

Fig. 5. Power and control performance of the optimal self-triggered protocol τ^* and the proposed improved one for different values of the power weight λ in (9). A better trade-off is achieved by applying the proposed event-triggered deviation in (24). For $\lambda \approx 0$ power is not penalized so $\tau^* \equiv 1$ and the control error is identically zero.

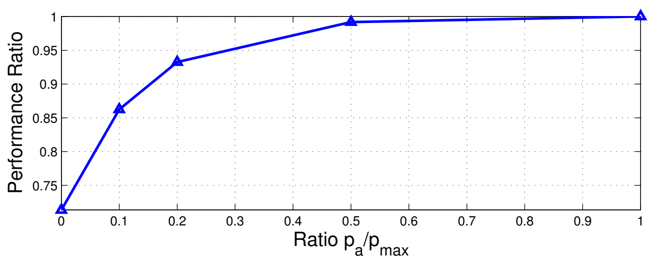

Fig. 6. The cost ratio of the proposed protocol over the optimal self-triggered is plotted versus different ratios of the receiver power consumption p_a over the maximum transmitting power $p_{\max}$ for the worst channel fading. As p_a increases the improvement of the proposed scheme becomes less significant.

trade-off than the self-triggered one. For each value of λ, the event-triggering deviation (24) allows to reduce the average power for only a small increase in the control performance.

We also fix $\lambda = 2$ and vary the constant power cost p_a for staying awake as a percentage of $p_{\max}$. In Fig. 6, we plot the ratio between the total cost J in (9) of the proposed improved protocol and the reference self-triggered τ^*. As p_a increases the two costs become the same, meaning that no deviations from τ^* were found profitable. This should not be interpreted as a weakness of the proposed protocol. It is rather expected that if the power needed to keep a device awake dominates the power to communicate the most efficient strategy is to keep devices at sleep mode most of the time and wake up only to transmit.

VI. Conclusions

In this paper we introduce a power-aware communication protocol for wireless sensor-actuator systems using event and self-triggered elements. After each transmission the devices switch to a zero-power sleep mode for a duration depending on the current channel state, and upon wake-up the transmitter decides based on current plant and channel conditions whether to transmit or not. The proposed protocol by design performs at least as good as the optimal self-triggered protocol.

In our development we assume channel fading is measured before transmission at the beginning of each slot when the devices are awake. An alternative approach, which might be preferable in practice, is to measure the channel during a packet transmission and pass channel information to the transmitter via the acknowledgments. In this case the devices have access to the current channel state h_k after a successful transmission at time k, so the sleep periods $\tau(h_k)$ of the self-triggered protocols (cf. Section III) can be directly followed. However since h_k is not known before transmission the sensor has an imperfect belief on its value, i.e., a distribution m_k on $\mathcal{H}$ based on the last measured channel and the Markov channel model. To guarantee packet delivery given the imperfect channel information m_k, as a self-triggered protocol requires, transmit power needs to increase accordingly. The methodology of Sections III and IV can be adapted in this case at the expense of more cumbersome notation. We note however that for packets with large size the imperfect channel information could cause a significant decrease in energy efficiency of the system, thus measuring channels before transmission might still be beneficial. Alternative channel estimation implementations are left for future investigation as well as extending the protocol design to multi-sensor/actuator wireless networks.

References

[1] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[2] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[3] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[4] M. Donkers and W. Heemels, "Output-based event-triggered control with guaranteed l8-gain and improved event-triggering," in *49th IEEE Conference on Decision and Control, 2010*, pp. 3246–3251, 2010.

[5] W. Heemels, M. Donkers, and A. Teel, "Periodic event-triggered control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847–861, 2013.

[6] Y. Xu and J. Hespanha, "Optimal communication logics in networked control systems," in *43rd IEEE Conference on Decision and Control, 2004*, vol. 4, pp. 3527–3532, 2004.

[7] R. Cogill, S. Lall, and J. Hespanha, "A constant factor approximation algorithm for event-based sampling," in *2007 American Control Conference*, pp. 305–311, 2007.

[8] L. Li and M. Lemmon, "Performance and average sampling period of sub-optimal triggering event in event triggered state estimation," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*, pp. 1656–1661, 2011.

[9] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *48th IEEE Conference on Decision and Control, 2009*, pp. 5832–5838, 2009.

[10] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Optimal power management in wireless control systems," in *Proc. of the 2013 American Control Conference (ACC 2013)*, pp. 1562–1569, 2013.

[11] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "A modeling approach on the telosb wsn platform power consumption," *The Journal of Systems and Software*, vol. 83, no. 1, pp. 1355–1363, 2010.

[12] R. Mangharam, A. Rowe, and R. Rajkumar, "FireFly: A Cross-layer Platform for Real-time Embedded Wireless Networks," *Real-Time System Journal*, vol. 37, no. 3, pp. 183–231, 2007.

[13] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A global time-synchronized link protocol for sensor networks," *Ad Hoc Networks*, vol. 6, no. 8, pp. 1201–1220, 2008.

[14] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[15] A. Goldsmith, *Wireless communications*. Cambr. Univ. Press, 2005.

[16] K. Gatsis, M. Pajic, A. Ribeiro, and G. J. Pappas, "Power-aware communication for wireless sensor-actuator systems: Extended version," *Technical Report*, 2013. Available at `http://www.seas.upenn.edu/~kgatsis/`.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.