2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Co-design for Control and Scheduling over Wireless Industrial Control Networks

Edwin G. W. Peters, Daniel E. Quevedo and Minyue Fu

Abstract— **We investigate the co-design of a scheduler and controller for feedback control over wireless industrial hybrid protocol networks. These hybrid protocols incorporate possibilities for both contention-free and contention-based medium access to support real-time requirements and intermittent communications, respectively. We focus in particular on the possibilities and limitations of feedback control over hybrid networks that are based on the IEEE 802.15.4 protocol, where we propose a controller-scheduler co-design that utilizes both the contention-free and contention-based parts of the protocol. This controller-scheduler is designed to minimize a linear quadratic cost function where probabilities for successful transmissions in the contention-free and contention-based parts also are taken into account. Simulation studies illustrate that careful co-design of the scheduler-controller results in significant performance gains compared to round-robin heuristics.**

I. INTRODUCTION

The emerging development of wireless networks in industrial environments with possibilities for real-time communications has significantly accelerated the research area of networked estimation and control in recent years. Many applications already utilize networked control systems (NCSs) [1]–[3]. Multiple commercial standards for industrial environments are available and implemented such as the WirelessHart, ISA100.11a and ZigBee, which all are based on the IEEE 802.15.4 physical layer [4], [5]. The main feature of the IEEE 802.15.4 standard is low energy consumption with possibilities for real-time requirements in the communication. This standard has, among others, a beaconed operation mode where the superframe is divided into a contention access period (CAP) (where multiple users can transmit) and contention free period (CFP) (where only devices that are assigned a slot can transmit). This is illustrated in Fig. 1.

When departing from periodic sampling, the NCS literature generally focuses on reducing the attention given to the actuators and/or sensors such that actions are performed only when necessary instead of periodically. This is to reduce the network bandwidth usage and/or the power consumption of wireless devices. The literature generally considers two sampling strategies: event-triggered, where the process is measured continuously, and an action is performed when conditions are exceeded [6]–[11], or self-triggered, where the time until a new action is performed is computed, and the process is only measured when the predetermined time elapses [12]–[15]. These can be regarded as closed loop and open loop strategies, respectively. Optimal scheduling with bandwidth limited communication has also been studied for deterministic networks [10], [16]–[19]. Packet loss in networks is frequently modeled as independent and identically distributed (i.i.d.) variables that indicate successful transmissions [11], [20]–[23]. Scheduling where only one sensor or actuator node is allowed to access the network at a time is studied in [17]–[19] where schedules are designed offline, while online scheduling is done in [24]–[27].

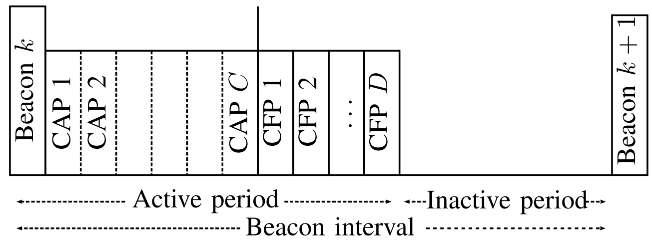

Fig. 1 – The IEEE 802.15.4 superframe structure.

In this work we consider a spatially distributed NCS involving multiple actuators and sensors and a linear-time-invariant system model. These communicate over an IEEE 802.15.4 based network, where only a limited amount of transmission slots are available in each superframe. The main difference to existing work is that we utilize both the CFP and the CAP when co-designing the scheduler and controller, and take the probabilities for packet losses into account.

Notation: We denote $\|x\|_Q^2 = x^T Qx$ where x^T is the transpose of a vector x. For a matrix A, a_{ij} is its ij'th element and a_j is its j'th column vector. The matrix I_n denotes the dimension $n \times n$ identity matrix. A random variable $\omega \sim \mathcal{N}\left(\mu, \sigma^2\right)$ is Gaussian distributed with mean μ and variance σ^2.

II. THE IEEE 802.15.4 NETWORK STANDARD

This section provides a brief introduction to the IEEE 802.15.4 standard and concludes with assumptions that we utilize through the remainder of this paper. For a more detailed explanation on the operation of the standard see, e.g, [28].

The IEEE 802.15.4 standard is designed to work in industrial environments [4], [5]. The standard itself only specifies the lower layers up to the physical layer, whereas the upper layers are to be designed by the implementer. There are multiple upper layer standards available, such as WirelessHART, ZigBee and ISA100.11a. The focus of IEEE 802.15.4 is a low cost and low power standard that has

Edwin G.W. Peters and Minyue Fu are with the School of Electrical Engineering and Computer Science, University of Newcastle, NSW 2308, Australia. Minyue Fu also holds a Qian-ren Scholar position in School of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang, P. R. China edwin.g.w.peters@gmail.com, minyue.fu@newcastle.edu.au.

Daniel E. Quevedo is with the Department of Electrical Engineering (EIM-E) The University of Paderborn, Germany dquevedo@ieee.org.

capabilities to feature real-time communication. The network can be configured as a star topology or peer-to-peer. Multiple networks, where each has a coordinator that manages the network, can be joined to form a mesh network. We however do not impose any specific network topology.

With IEEE 802.15.4, the network can operate in both a (synchronized) beacon enabled and a non-beacon enabled mode. In this paper we use the beacon enabled mode which utilizes superframes that include a CAP, a CFP and an optional inactive period where the entire network can go into a low power state. This superframe is shown in Fig. 1. Here a beacon, which indicates the beginning of a superframe, is transmitted at a fixed beacon interval. This beacon includes a table containing the nodes that are allowed to transmit during the CFP. While a maximum of 7 guaranteed time-slots (GTSs) are allowed to be assigned in one superframe in the IEEE 802.15.4 standard, WirelessHART and ISA100.11a allow for more GTSs in each superframe. The beacon interval and the length of the CAP, CFP and low power state can be defined by the implementer of the network. The IEEE 802.15.4e extends the standard with the options for channel hopping and allows multiple superframes [5].

Users that want to transmit during the CFP have to request a so-called GTS before transmitting. The coordinator will grant the GTSs in a first-come-first-served manner while the maximum number of allowed GTSs (set by the implementer of the network) is not reached. Only users that have a GTS assigned are allowed to transmit during the CFP. This means, that only one user is allowed to transmit in a time slot, thus no packet collisions will occur. However, since it is likely that the network is affected by interference, a small probability for packet dropouts remains.

During the CAP all users are allowed to access the network. Before a user can transmit, it verifies whether it can finish its transmission before the end of the CAP. If this is not possible, the user will delay its transmission until the next superframe. If the transmission can be finished, the user senses the channel for ongoing transmissions. If the channel is available, it transmits. In case the channel is occupied, the user delays its transmission for a random number of slots and retries the procedure. Due to the nature of the CAP, there is still a probability for collisions to occur when a user transmits a data packet. This results in packet dropouts. Successful transmissions in the CAP are confirmed using acknowledgments. If no acknowledgment is received, the packet will be retransmitted if it can finish before the end of the CAP in the current superframe. If this is not the case, the retransmission will be postponed until the next superframe.

Our focus is on the situation where the network is shared. We do not know how many users are competing for the CAP and how frequently they transmit. We furthermore only have a limited amount of GTSs available in each superframe. If the control loop were to maximize the probability to successfully transmit packets it would try to access the channel as often as possible, this leading to network overload. To avoid such issues, and to maintain fairness in the CAP, we only allow the control loop to use a fixed amount of slots in the CAP of each superframe as shown in Fig. 1.

We make the following assumption to ensure, that the time delay from the start of a superframe until the packet is received in that superframe can be neglected:

Assumption 1. *The inactive period of the superframe is much longer than the active period of the superframe. This makes the transmission duration in the superframe negligible compared to the sampling period of the feedback control system.*

Note that as described in the IEEE 802.15.4e standard [5], other superframes can be defined during the inactive period of the control superframe. These superframes can be utilized by other users of the network. Under Assumption 1 we can neglect transmission failures in the CAP when the retransmission is successful later within the same superframe. If the transmission of the packet has to be postponed to the next superframe, the data in the packet might not be optimal anymore when applied to the actuator, and is therefore discarded. The probability for this to occur depends on numerous parameters, such as the number of users on the channel, the frequency at which they want to transmit, the length of the packets, the time in the current CAP at which the user attempts their first transmission etc. This then has to be compared to the length of the CAP. To illustrate the effect of utilizing a CAP that is shared among other users, we make the following assumption in this paper:

Assumption 2. *We consider transmissions in a CAP or CFP that are delayed till the next superframe as being lost. Successful transmissions occur with fixed probabilities p_{CAP} and p_{CFP}, respectively, where $p_{CAP} < p_{CFP}$. We consider the packet loss rates to be constant and known, cf. [29].*[1]

III. Networked Control Architecture of Interest

We consider a spatially distributed large control system with many sensors and actuators which is shown in Fig. 2. The sensors and actuators are placed at different physical locations and are not directly connected to each other. Here $\hat{y}$ and $\hat{u}$ are the measurement and control signals, respectively, that are received after being transmitted over the network. The measurements are transmitted to a state estimator, such that the controller receives the state estimate $\hat{x}$. Since the IEEE 802.15.4 protocol uses acknowledgments to confirm successful transmissions, the controller and state estimator know the transmission outcomes of u.

The IEEE 802.15.4e standard supports multiple superframes. We for simplicity only use one superframe at each sampling instant for the actuators. The beacon interval for this superframe equals the sampling interval of the system. Further, Assumption 1 allows us to neglect the inter-transmission times within the superframe. Also, all the

[1] In case the beacon frame is not received by an actuator node according to the IEEE 802.15.4 standard, the node shall not use its GTS [4]. The control signal might therefore not be received by the actuator. Note that the probability for a beacon frame loss easily can be incorporated in the probabilities for successful transmissions p_{CAP} and p_{CFP}.

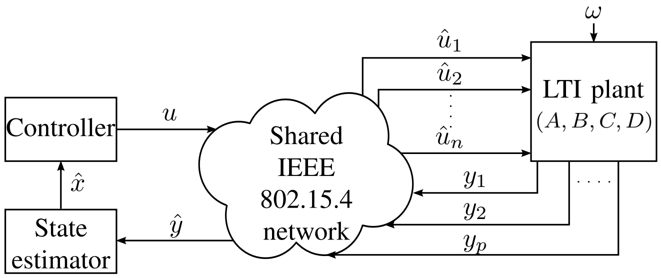

Fig. 2 – The control system with many actuator and sensor nodes.

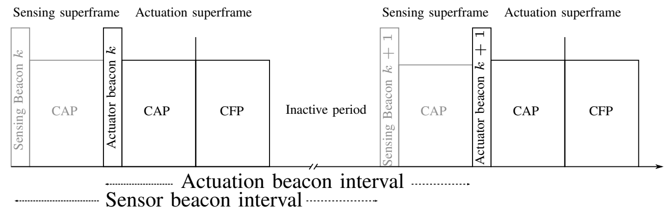

Fig. 3 – Superframe structure for the networked control setup. Here the sensor measurements at time step k are transmitted in the sensing superframe, which finishes right before the actuation superframe commences. The inactive period between actuation superframe k and sensing superframe $k+1$ is significantly longer than the length of the active periods of both superframes.

sensor measurements are transmitted in a separate superframe, where the beacon interval also equals the sampling interval of the system. This superframe finishes just before the actuator superframe begins, which under Assumption 1 allows us to neglect the inter transmission times i.e., the time intervals between from when the sensor measurements $y(k)$ are transmitted to when the control values in $u(k)$ are applied to the actuators. The superframes are shown in Fig. 3.

In the present work we consider the control side and use a state estimator that handles intermittent observations and can easily be designed using methods such as [29][2]. In this work we use acknowledgments to confirm successful transmissions. The state estimator therefore knows whether a packet dropout has occurred or not. This means that separation holds [33]. Since we only consider the scheduling of the actuators, we assume that at every sample instance **all** sensor readings get transmitted jointly in a dedicated sensing superframe. This is illustrated in Fig. 3. We further transmit all sensor data through the less reliable transmission mode, the CAP. Assumption 1 allows us to neglect the time delays included between receiving the sensor data and applying the control signal to the actuators.

The next state of the plant is given by

$$x(k+1) = Ax(k) + B\hat{u}(k) + D\omega(k), \ k \in \mathbb{N} \tag{1}$$

and the output

$$y(k) = Cx(k) + H\nu(k), \tag{2}$$

where $x(k) \in \mathbb{R}^m$, $\hat{u}(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$ and A, B, C are matrices and $D = \mathbf{1}_m$, $H = \mathbf{1}_p$ are vectors of dimensions m and p, respectively, that only contain ones. The plant disturbance $\omega(k) \sim \mathcal{N}\left(0, \sigma_\omega^2\right)$ and measurement disturbance $\nu(k) \sim \mathcal{N}\left(0, \sigma_\nu^2\right)$ are zero-mean Gaussian with variances σ_ω^2 and σ_ν^2, respectively.

As already mentioned in Section II, the actuators are to be controlled over an IEEE 802.15.4 compatible network, which can only support a limited amount of data during the CAP and CFP. We are interested in a situation where there are more actuators than the combined amount of available slots in the CAP and CFP in each superframe. This means, that we have to schedule which actuators to address in every actuation superframe. The sample period of the system is aligned to the length of a superframe, such that at each time step k a new superframe begins, see Fig. 3. Recall further, from Assumption 2, that a packet which at time k is delayed until a future superframe is considered as being lost. We further neglect the difference between the arrival time of the packets in a common superframe according to Assumption 1, such that all control inputs received are applied to the model (1) at the same time.

We let the length n vector $S(k)$ be the schedule at time k, given by

$$S(k) = \left[s_1(k), s_2(k), \ldots, s_n(k)\right]^T. \tag{3}$$

This schedule is represented by the probabilities of successful transmission of the packet. Each entry, $s_j(k)$, indicates whether actuator j is addressed during the CAP, CFP or is not addressed. To indicate in which period actuator j is addressed, we set the entry $s_j(k)$ equal to the probability of successful transmission during the period in the superframe. Thus, $s_j(k)$ belongs to the ternary set

$$s_j(k) \in \{0, p_{\text{CAP}}, p_{\text{CFP}}\}, \tag{4}$$

where p_{CAP} and p_{CFP} are the (known or estimated) probabilities for successful transmissions in the CAP and CFP.

We define $\gamma(k)$ to be the vector that contains all transmission outcomes associated to superframe k. Here $\gamma_j(k) = 1$ if the transmission to actuator j was successful and $\gamma_j(k) = 0$ otherwise. Thus the i.i.d. distribution of $\gamma_j(k)$ is given by

$$\mathbf{Pr}\{\gamma_j(k) = 1\} = s_j(k), \ j \in \{1, 2, \ldots, n\}. \tag{5}$$

For ease of exposition, in what follows we adopt a set-to-zero strategy. Thus, the control signal that is used at the actuators, $\hat{u}(k)$, is given by

$$\hat{u}_j(k) = \begin{cases} u_j(k) & \text{if } \gamma_j(k) = 1 \\ 0 & \text{if } \gamma_j(k) = 0. \end{cases} \tag{6}$$

By defining $B(\gamma(k), k) \triangleq B^{(k)} \operatorname{diag}\{\gamma(k)\}$, where $\operatorname{diag}\{\gamma(k)\}$ creates a $n \times n$ matrix with the entries of $\gamma(k)$ on its diagonal and every column $b_j^{(k)}$ in $B^{(k)}$ is given by

$$b_j^{(k)} = \begin{cases} b_j & \text{if } s_j(k) > 0 \\ 0 & \text{if } s_j(k) = 0, \end{cases} \tag{7}$$

[2]Scheduling for state estimation has been studied recently in works such as [23], [29]–[32].

we can combine (1) and (6) to obtain the NCS model:

$$x(k+1) = Ax(k) + B(\gamma(k), k)\, u(k) + D\omega(k). \quad (8)$$

Within the current setup, scheduling amounts to designing the schedule $S(k)$ in (3). Since $p_{\text{CFP}} > p_{\text{CAP}}$, one would ideally like to assign every actuator to a GTS in every superframe. However, this can only be done if the number of actuators is smaller than the number of available GTSs in a superframe. Otherwise, the CAP needs to be used as well. Further in case there are more actuators than the available amount of slots in the CFP and CAP together, some actuators will not be addressed at all in that superframe. This raises the question: Which actuators to address in which slot in a superframe, which ones to omit and what data should be sent?

To address this question, we will next use a linear quadratic cost function, which we minimize to obtain both an optimal scheduling sequence and also the associated control laws.

IV. Controller-scheduler co-design

We propose to use a finite horizon linear quadratic (LQ) cost function with final state weighting, which we will minimize to obtain the optimal scheduling sequence and associated control laws. We consider the LQ cost for a finite scheduling sequence

$$\overrightarrow{S}(k) \triangleq \{S(k), S(k+1), \dots, S(k+N-1)\}, \quad (9)$$

where $S(k)$ is as in (3):

$$J\left(x(k), \overrightarrow{S}(k), \mu\left(k, x(k), \overrightarrow{S}(k)\right)\right) = \mathbf{E}\left\{ \|x(k+N)\|_W^2 + \sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|u(k+\ell)\|_R^2 \,\middle|\, x(k), \overrightarrow{S}(k) \right\}. \quad (10)$$

The matrices Q and R are positive definite and penalize the cost of the state and the cost of control, respectively, W is a positive definite matrix that penalizes the terminal cost at time N. Further, the control law $\mu\left(k, x(k), \overrightarrow{S}(k)\right)$ maps $x(k)$ into control signals $u(k)$, such that $u(k) = \mu\left(k, x(k), \overrightarrow{S}(k)\right)$. This cost function uses the actual state of the plant $x(k)$. (In the simulations in Section V, these are replaced by estimates, $\hat{x}(k)$.)

Note that, to find the optimal scheduling sequence $\overrightarrow{S}^*(k)$ and associated control policy

$$\pi^* = \left\{\mu^*(\overrightarrow{S}^*(k)), \mu^*(\overrightarrow{S}^*(k+1)), \dots, \mu^*(\overrightarrow{S}^*(k+N-1))\right\}$$

the cost function (10) has to be evaluated for every possible scheduling sequence. For a horizon of length N there are

$$M = \left(\frac{n!}{\hat{D}!\left(n-\hat{D}\right)!} \frac{(n-\hat{D})!}{\hat{C}!\left(n-\hat{D}-\hat{C}\right)!} \right)^N \quad (11)$$

possible scheduling sequences. Here $\hat{D}$ are the GTSs and $\hat{C}$ are the slots in the CAP that are available to the controller.

A. Solution

To state the optimal solution of (10) we first note that, using (5) and (7), the expectation of $b_j(k)$ is given by $\mathbf{E}\{b_j(\gamma_j(k), k) \,|\, s_j(k)\} = \mathbf{Pr}\{\gamma_j(k) = 1 \,|\, s_j(k)\}\, b_j^{(k)} = s_j(k) b_j$. Where the last equality comes from the fact that $s_j(k)$ contains the (estimated or known) probability of a successful transmission, see (4) and (5).

Theorem 3. *Suppose that Q and R are positive definite and the pair (A, B) is controllable. Then for a finite scheduling sequence $\overrightarrow{S}(k)$,*

$$\min_{\mu(\overrightarrow{S}(k))} J\left(x(k), \overrightarrow{S}(k), \mu\left(k, x(k), \overrightarrow{S}(k)\right)\right) = x(k)^T P(\overrightarrow{S}(k)) x(k) + \sum_{\ell=1}^{N} \sigma_\omega^2 D^T P(\overrightarrow{S}(k+\ell)) D. \quad (12)$$

In (12), $P(\overrightarrow{S}(k))$ is given by the recursion

$$\begin{aligned} P(\overrightarrow{S}(k)) &= Q + A^T P(\overrightarrow{S}(k+1)) A \\ &\quad - A^T P(\overrightarrow{S}(k+1)) B \operatorname{diag}\{S(k)\} L(\overrightarrow{S}(k)) \end{aligned} \quad (13)$$

where $\overrightarrow{S}(k+\ell) \triangleq \{S(k+\ell), \dots, S(k+N-1)\}$ and

$$L(\overrightarrow{S}(k)) = \left(R + \operatorname{diag}\{S(k\}\, B^T P(\overrightarrow{S}(k+1)) B\right)^{-1} \times \operatorname{diag}\{S(k)\}\, B^T P(\overrightarrow{S}(k+1)) A, \quad (14)$$

with $P(\overrightarrow{S}(k+N)) = W$ and $L(\overrightarrow{S}(k+N)) = 0$. The minimizing control policy is then given by

$$\begin{aligned} \pi^* = \Big\{ &\mu^*\left(k, x(k), \overrightarrow{S}(k)\right), \dots, \\ &\mu^*\left(k+N-1, x(k+N-1), \overrightarrow{S}(k+N-1)\right) \Big\} \end{aligned} \quad (15)$$

where each control law is defined by

$$\mu^*\left(k, x(k), \overrightarrow{S}(k)\right) \triangleq -L(\overrightarrow{S}(k)) x(k). \quad (16)$$

Proof. The proof is based on results in [34, Section 4.1]. □

Using Theorem 3 we can minimize the cost function (12) for any given scheduling sequence $\overrightarrow{S}(k)$. This allows us to solve the optimal controller and scheduler co-design problem by jointly minimizing (12) for the optimal schedule and control signal as

$$\overrightarrow{S}^*(k) = \operatorname*{argmin}_{\overrightarrow{S}(k)} \Bigg[x(k)^T P(\overrightarrow{S}(k)) x(k) + \sum_{\ell=1}^{N} \sigma_\omega^2 D^T P(\overrightarrow{S}(k+\ell)) D \Bigg]. \quad (17)$$

Clearly, (17) is a combinatiorial problem where the number of possible scheduling sequences for a horizon length N is given by (11), cf. [17]–[19]. The solutions to the recursion (13) and the sum term in (17) can be calculated offline for all possible scheduling policies and stored in a look-up table. In this case the computations for each scheduling sequence reduce to evaluate the quadratic term on $x(k)$ and add constant.

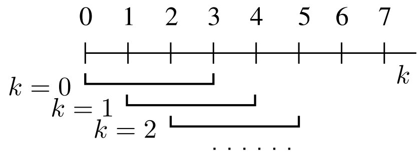

Fig. 4 – Illustration of MPC where $N = 3$.

B. Algorithm

The presented algorithm is implemented as model predictive control (MPC), where at every time step the (estimated) state is used to solve the optimization problem (17). An illustration of this principle is shown in Fig. 4. The algorithm is described in Algorithm 1, which is executed at every time step, where (17) is solved, and the control signal $u(k)$ is given by $u(k) = \mu^* \left(k, x(k), \overrightarrow{S}^*(k)\right)$.

Algorithm 1. MPC - finite horizon

1) At time k the state estimate of $\hat{x}(k)$ is received by the controller from the estimator.
2) The controller computes $\overrightarrow{S}^*(k)$ using (17) and $\pi^*(k)$ using (16).
3) The controller sends $u^*(k) = \mu^* \left(k, \hat{x}(k), \overrightarrow{S}^*(k)\right)$ where $\mu^* \in \pi^*(k)$ over the network. The actuators apply the successfully transmitted control signals to (1).

As in other MPC schemes, see [35], the control performance is affected by the length of the prediction horizon N and the choice of the final state weighting W. Larger N will however increase the complexity of (17) exponentially.

V. Simulation studies

In this section we illustrate the performance gains of the scheduling algorithm presented in Section IV over a simple round robin (RR) heuristic. We simulate a NCS with 3 actuators that compete for 1 GTS and 1 slot in the CAP. The parameters of the model in (1) and (2) are given by

$$A = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 \\ 0 & 1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0.819 & 0 \\ 0 & 0 & 0 & 0.906 & 1 \end{bmatrix} \tag{18}$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 9.063 \\ 0 & 0 & 4.683 \end{bmatrix}. \tag{19}$$

The vector $D = \mathbf{1}_5$ and the plant disturbance $\omega(k) \sim \mathcal{N}(0, 1)$. We can observe all states such that $C = I_5$, $H = \mathbf{1}_5$ and the measurement noise $\nu(k) \sim \mathcal{N}(0, 0.01)$. The weighting matrices Q, R and W are all identity matrices of appropriate dimensions[3]. We fix the probabilities of successful transmissions to constant values to illustrate the effect of packet loss in the closed loop system. These probabilities for successful transmissions are given by $p_{\text{CFP}} = 0.95$, $p_{\text{CAP}} = p_{\text{est}} = 0.75$, where p_{est} is the success probability for the transmissions in the sensing superframe.

The performance is analyzed by the empirical cost averaged over time, which is given by

$$\frac{1}{T} \sum_{k=0}^{T} x(k)^T Q x(k) + u(k)^T R u(k), \tag{20}$$

where Q and R are the weighting matrices of the state and control signal, respectively, and T is the length of the simulation. This is averaged over 1000 simulations, each of 1000 time-steps, of the same system with different initial conditions and noise realizations. We compare the developed algorithms to a simple RR heuristic, where the access to the CAP and CFP is shared equally among the actuators. Further, to show the importance of online schedule design for systems that are affected by disturbances, we also perform simulations where the system (1) is affected by random and unmeasured additive disturbances $\chi(k) \in \mathbb{R}^m$ such that

$$x(k+1) = Ax(k) + B\hat{u}(k) + D\omega(k) + \chi(k)$$

Here every element in $\chi(k)$ is given by

$$\chi_i(k) = \rho_i(k)\upsilon_i(k) + \iota_i(k)\chi_i(k-1), \tag{21}$$

where $\mathbf{Pr}\{\rho_i(k) = 1\} = p_{\text{step}} = 0.05$, $\mathbf{Pr}\{\iota_i(k) = 1\} = p_{\text{length}} = 0.85$, $\upsilon_i(k) \sim \mathcal{U}(-10, 10)$ is uniformly distributed and $\chi_i(-1) = 0$.

In the RR heuristic, each actuator is allowed an equal share of access to the network. Depending on the period length this is however not always possible. In this case the more unstable subsystems in A are granted priority to the CFP. The RR schedules are chosen as

$$\overrightarrow{S}_2 = \left\{ \begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} p_{\text{CFP}} \\ 0 \\ p_{\text{CAP}} \end{bmatrix} \right\}, \overrightarrow{S}_3 = \left\{ \begin{bmatrix} p_{\text{CFP}} \\ p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ p_{\text{CFP}} \\ p_{\text{CAP}} \end{bmatrix} \begin{bmatrix} p_{\text{CAP}} \\ 0 \\ p_{\text{CFP}} \end{bmatrix} \right\},$$

while $\overrightarrow{S}_4 = \left\{ \overrightarrow{S}_2, \overrightarrow{S}_2 \right\}$.[4]

The empirical cost of the simulations, calculated using (20), are shown in Fig. 5 in logarithmic scale. The simulations show that the proposed co-design algorithm performs significantly better than the schedules designed using RR. Here at $N = 4$ the cost using Algorithm 1 is reduced by 6.8 dB and 7.7 dB when the system is affected by Gaussian noise and the noise in (21), respectively. The performance gain is more significant when using shorter horizons. The performance of Algorithm 1 when increasing N improves by 4 dB when $N = 2$ and the system is affected by (21). The performance gain of Algorithm 1 for the simulated system is however limited when $N > 2$.

[3]This system has open loop eigenvalues at $1.5, 1, 0.82$ and two eigenvalues located at 1.1.

[4]Another choice for $\overrightarrow{S}_4$ would have been to use $\overrightarrow{S}_3$ and add a schedule that addresses the actuators of the systems with the highest eigenvalues. This however did result in a significantly reduced control performance. Note however, that other choices of schedules for RR might provide better performance.

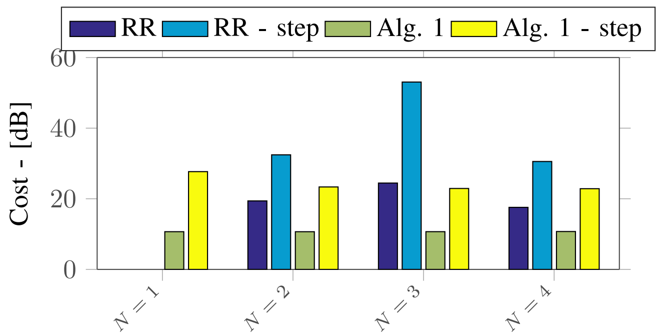

Fig. 5 – Empirical performance (20) of the algorithms averaged over 1000 simulations. Here in "RR - step" and "Alg. 1 - step" the system (1) is affected by the disturbance (21).

VI. Conclusion

We have discussed a possibility for control design over IEEE 802.15.4 networks where both the CFP and CAP are utilized. Simulations illustrate that proper schedule and control co-desing is crucial for the performance of the NCS.

Future work will address the computational complexity of the co-design algorithm and consider a more realistic model of the network congestion instead of i.i.d. dropouts. This can e.g. take into account external users that use a random amount of bandwidth and transmit at random times which interferes with the NCS, cf. [29].

References

[1] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908 – 917, 2011.

[2] T. Namerikawa and T. Kato, "Distributed load frequency control of electrical power networks via iterative gradient methods," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Dec 2011, pp. 7723–7728.

[3] S. Oncu, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "String stability of interconnected vehicles under communication constraints," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 2459–2464.

[4] *IEEE Standard for Local and metropolitan area networks - Part 15.4: Networks Low-Rate Wireless (LR-WPANs)*, IEEE Standards Association, 2011.

[5] *IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, IEEE Standards Association, 2012.

[6] B. Wang and M. Fu, "Comparison of periodic and event based sampling for linear state estimation," in *19th IFAC World Congress of the international Federation of Automatic Control*, 2014.

[7] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2456–2461, Oct 2011.

[8] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems," *Automatic Control, IEEE Transactions on*, vol. 58, no. 2, pp. 470–474, Feb 2013.

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

[10] D. Antunes and W. Heemels, "Rollout event-triggered control: Beyond periodic control performance," *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3296–3311, Dec 2014.

[11] D. E. Quevedo, V. Gupta, W.-J. Ma, and S. Yüksel, "Stochastic stability of event-triggered anytime control," *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, p. 3373, 2014.

[12] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 435–440.

[13] U. Tiberi, C. Fischione, K. Johansson, and M. D. Benedetto, "Energy-efficient sampling of networked control systems over IEEE 802.15.4 wireless networks," *Automatica*, vol. 49, no. 3, pp. 712 – 724, 2013.

[14] X. Wang and M. Lemmon, "Self-triggered feedback control systems with finite-gain l2 stability," *Automatic Control, IEEE Transactions on*, vol. 54, no. 3, pp. 452–467, March 2009.

[15] J. Almeida, C. Silvestre, and A. Pascoal, "Self-triggered output feedback control of linear plants in the presence of unknown disturbances," *Automatic Control, IEEE Transactions on*, vol. 59, no. 11, pp. 3040–3045, Nov 2014.

[16] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 175–184, Feb 2014.

[17] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," *Automatica*, vol. 40, no. 3, pp. 491 – 500, 2004.

[18] ——, "Integration of off-line scheduling and optimal control," in *Real-Time Systems, 2000. Euromicro RTS 2000. 12th Euromicro Conference on*, 2000, pp. 137–143.

[19] D. Hristu and K. Morgansen, "Limited communication control," *Systems & Control Letters*, vol. 37, no. 4, pp. 193 – 205, 1999.

[20] 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, Jan 2007.

[21] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *Automatic Control, IEEE Transactions on*, vol. 52, no. 7, pp. 1314–1319, July 2007.

[22] K. You, M. Fu, and L. Xie, "Mean square stability for Kalman filtering with Markovian packet losses," *Automatica*, vol. 47, no. 12, pp. 2647 – 2657, 2011.

[23] E. Rohr, D. Marelli, and M. Fu, "Kalman filtering with intermittent observations: On the boundedness of the expected error covariance," *Automatic Control, IEEE Transactions on*, vol. 59, no. 10, pp. 2724–2738, Oct 2014.

[24] R. Postoyan, N. Van De Wouw, D. Nesic, and W. Heemels, "Tracking control for nonlinear networked control systems," *Automatic Control, IEEE Transactions on*, vol. 59, no. 6, pp. 1539–1554, June 2014.

[25] E. Henriksson, D. E. Quevedo, E. G. W. Peters, H. Sandberg, and K. H. Johansson, "Multiple loop self-triggered model predictive control for network scheduling and control," *Control Systems Technology, IEEE Transactions on*, 2015, to be published. [Online]. Available: http://arxiv.org/abs/1502.03181

[26] E. Henriksson D. E. Quevedo, H. Sandberg, and K. H. Johansson, "Self-triggered model predictive control for network scheduling and control," in *8th IFAC Symposium on Advanced Control of Chemical Processes*, 2012.

[27] M. Lješnjanin, D. E. Quevedo, and D. Nešić, "Packetized MPC with dynamic scheduling constraints and bounded packet dropouts," *Automatica*, vol. 50, no. 3, pp. 784 – 797, 2014.

[28] A. Koubâa, M. Alves, and E. Tovar, "IEEE 802.15.4 for wireless sensor networks: A technical overview," CISTER - Research Center in Real-Time & Embedded Computing Systems, Tech. Rep., Jul. 2005. [Online]. Available: http://www.cister.isep.ipp.pt/docs/ieee_802_15_4_for_wireless_sensor_networks__a_technical_overview/222/

[29] D. Quevedo, A. Ahlen, and K. Johansson, "State estimation over sensor networks with correlated wireless fading channels," *Automatic Control, IEEE Transactions on*, vol. 58, no. 3, pp. 581–593, March 2013.

[30] C. Ramesh, "State-based channel access for a networked control system," Ph.D. dissertation, KTH Royal Institute of Technology, 2014.

[31] D. Quevedo, J. Ostergaard, and A. Ahlen, "Power control and coding formulation for state estimation with wireless sensors," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 2, pp. 413–427, March 2014.

[32] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[33] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, "Optimal control with unreliable communication: the TCP case," in *American Control Conference, 2005. Proceedings of the 2005*, June 2005, pp. 3354–3359 vol. 5.

[34] D. Bertsekas, *Dynamic Programming and Optimal Control vol. 1*, ser. Athena Scientific optimization and computation series. Athena Scientific, 2005.

[35] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.