Controller and Scheduler Codesign for Feedback Control Over IEEE 802.15.4 Networks

Edwin G. W. Peters, *Student Member, IEEE*, Daniel E. Quevedo, *Senior Member, IEEE*, and Minyue Fu, *Fellow, IEEE*

***Abstract*— In this paper, we examine the impact of the scheduling policy on the performance for feedback control over networks based on a hybrid communication protocol, which incorporates both contention-free and contention-based medium access. We investigate the possibilities and limitations of feedback control in the IEEE 802.15.4 specification and state the problem of scheduling when the controller both utilizes the contention-based and guaranteed parts of the protocol. We propose and compare a number of scheduler–controller codesign algorithms that take both the contention-based and guaranteed parts of the protocol into account using an approximation to an infinite horizon linear quadratic cost function. The simulation studies show that a careful codesign of the scheduler–controller results in significant performance gains, when using both online and offline scheduling policies.**

***Index Terms*— IEEE 802.15.4, networked control systems (NCSs), optimal scheduling, wireless industrial control systems, wireless sensor and actuator networks.**

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), such as robotics, load control, and the smart home [1]–[3]. Multiple commercial standards for industrial environments are available and implemented, such as the WirelessHart, ISA100.11a, and ZigBee standards [4], [5], which all are based on the IEEE 802.15.4 physical layer [6], [7]. The main feature of the IEEE 802.15.4 standard is low energy consumption with possibilities for real-time requirements in the communication. The IEEE 802.15.4 standard has, among others, a beaconed operation mode where the superframe is divided into a contention access period (CAP) and contention-free period (CFP). This is illustrated in Fig. 1. Transmissions in the CAP utilize a carrier sense multiple access with collision avoidance (CSMA/CA) algorithm, whereas time division multiple access (TDMA) is used in the CFP. Here, the so-called guaranteed time slots (GTSs) in the CFP need to be requested by the user at least one superframe ahead. Changes in the allocation of GTSs in the CFP also need to be requested by the user at least one superframe ahead.

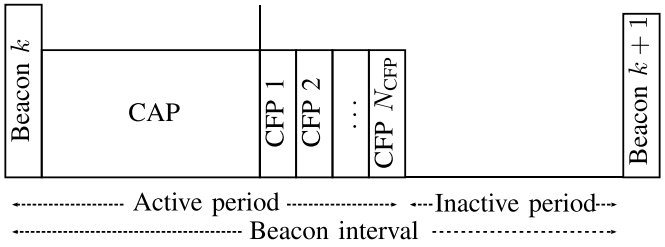

Fig. 1. IEEE 802.15.4 superframe structure.

The CFP can be used for reliable periodic transmissions to actuators or sensors without failure or transmission delays. While the transmission through the CFP leads to a deterministic operation, as specified in [7], only up to seven GTSs are available per IEEE 802.15.4 superframe. WirelessHart and ISA100.11.a have the possibility of assigning more than seven GTSs. A transmission in a GTS has to be requested by the user, after which it may take multiple superframes before the GTS is assigned and transmissions can commence. The CAP can be used for event-based transmissions that are not planned ahead of time. There is, however, a greater probability for transmission failures since multiple users are competing for access to the channel. With the above as background, this paper investigates optimal scheduling and control codesign over IEEE 802.15.4-based networks where we take the advantage of both the CAP and CFP periods. We implement the algorithms using model predictive control (MPC) ideas.

Extensive research has been done in control and sampling over wireless networks, especially in reducing the communication bandwidth requirements and thereby also managing the power consumption. Without getting too exhaustive, we will next revise some works that motivate this paper.

When departing from periodic sampling, the NCS literature generally focuses on reducing the attention to 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

Manuscript received October 15, 2015; accepted December 26, 2015. Date of publication February 2, 2016; date of current version October 14, 2016. Manuscript received in final form January 10, 2016. Recommended by Associate Editor L. Xie.

E. G. W. Peters is with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: edwin.g.w.peters@gmail.com).

D. E. Quevedo is with the Department of Electrical Engineering and Information Technology (EIM-E), Paderborn University, Paderborn 33098, Germany (e-mail: dquevedo@ieee.org).

M. Fu is with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia, and also with the School of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: minyue.fu@newcastle.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2016.2517571

devices. The literature generally considers two sampling strategies: event-triggered, where the process is measured continuously and an action is performed on the event when conditions are exceeded, or self-triggered, where the time until a new action is performed is precomputed and the process is measured only when the predetermined time elapses. These can be regarded as closed-loop and open-loop strategies, respectively. References [8]–[16] consider event- and self-triggered sampling for deterministic networks with reliable links. In [17] and [18], event-triggered, self-triggered, and hybrid algorithms, where the latter combines the self- and event-triggered algorithms, are developed and their performance is compared when minimizing the attention to a system subject to disturbances. Much research is also done on the codesign of the controller and scheduler under communication bandwidth constraints [18]–[22]. While in some cases either actuators or sensors are scheduled, some works consider simultaneous scheduling of both the actuators and sensors. These works, however, do not take network conditions such as packet losses into account. The other research is done on networks with limited bandwidth and delays [23], [24]. Here, optimal scheduling and control codesign is, however, not done. In [25], the quality of service on the IEEE 802.15.4 network is investigated with respect to cyber-physical systems; here the impact of delays on the bandwidth is investigated while multiple transmitters are active in the CAP. Wireless networks are often shared with different types of users that have varying bandwidth demands at different times, which one may not be able to control directly. Furthermore, external noise can interfere with the network and thereby reduce the link quality. This can be modeled as independent and identically distributed (i.i.d.) variables that indicate whether disturbances affected the packet transmission or not [26], [27]. Packet dropouts and delays on the network have been investigated extensively in the literature (see [28]–[31]), whereas [32] studies the maximum number of successive packet dropouts for subsystems in a large distributed setting using event-triggered control. Ulusoy *et al.* [33] propose to use a cooperative medium access control (MAC) protocol as modification to the standard IEEE 802.15.4 MAC layer and combined this with MPC to improve the performance when different disturbances are present on the network. Araujo *et al.* [18] investigate combinations of event and self-triggered control using both the CFP and CAP for transmissions. Here, the power consumption and communication bandwidth management are investigated. Networked control where errors are introduced in the communication for both control and measurement signals is studied in [34]–[37]. Here, dynamic scheduling, when only one sensor or actuator node is allowed to access the network at a time, results in significant performance improvements compared with static schedules. Dynamic programming is used to codesign the schedule and control signals for a deterministic channel in [13] and [19]. The proposed algorithms update the schedule periodically depending on the system state. The algorithms presented utilize only the CFP portion of the IEEE 802.15.4 standard.

References [20]–[22] consider scheduling over networks, where only one actuator is allowed to access the network at each time instant, and generate a periodic schedule and control laws offline, which are then applied online. This is also done for sensor networks, where the schedule under some conditions converges to a periodic sequence [38]. In [39], optimization-based approaches are used to schedule decoupled control loops based on selection of the sampling rate of every control loop. The performance of the presented approaches is evaluated using a WirelessHART test bed and simulated control systems.

In this paper, we consider a spatially distributed large control system involving multiple actuators and sensors and a linear-time-invariant system model. We focus on scheduling of the control data on the link between the controller and actuators that communicate over an IEEE 802.15.4-based network, where we have only a limited number of transmission slots available in each superframe. Whereas the existing literature focuses on increasing the transmission interval to reduce power and bandwidth consumption on the network, scheduling actuators on a deterministic network, or control design with a single link with packet dropouts, we codesign the controller to take network conditions into account. In addition, we focus on optimally sharing a finite resource (the transmission slots over the network) among a number of actuators. The main difference in this paper compared with the existing literature on scheduling and control codesign is that we utilize both the CFP and the CAP in a superframe and take the probabilities for packet losses into account while designing the schedule. We propose to codesign scheduler–controller policies that use both the more reliable but dedicated CFP and the less reliable and shared CAP in every superframe to maximize the control performance by minimizing a linear quadratic (LQ) cost function. The results presented in this paper are extensions to [40]. Finally, we illustrate the performance gains when using optimal scheduling methods compared with simple heuristics through simulation studies.

The basics of the IEEE 802.15.4 protocol are explained in Section II after which we describe the NCS architecture of interest in Section III, where we also present the scheduling and control codesign problem. We present our proposed algorithms that codesign the schedule and control signals in Sections IV and V. Section VI documents the simulation studies of the proposed algorithms and simple heuristics. Conclusions are drawn in Section VII.

Notation: We denote $\|x\|_Q^2 = x^T Q x$, 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 and $\mathbf{1}_m$ the vector of length m containing a one in each entry. A random variable $\omega \sim \mathcal{N}(\mu, \sigma^2)$ is Gaussian distributed with mean μ and variance σ^2. Further let $\mathbb{N}$ and $\mathbb{R}$ be the set of natural and real numbers, respectively.

II. 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 IEEE 802.15.4 standard, the reader is referred to [41].

The IEEE 802.15.4 standard is designed to work in industrial environments [6], [7]. The standard itself specifies only 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 the IEEE 802.15.4 is a low-cost and low-power standard that has capabilities to feature real-time communication. The network can be configured as a star topology or a peer-to-peer one. 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 nonbeacon-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. Fig. 1 shows the IEEE 802.15.4 superframe. In this mode, 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 seven GTSs are allowed to be assigned in one superframe in the IEEE 802.15.4 standard, WirelessHART and ISA100.11a allow 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 the possibility of defining multiple superframes [7].

The CFP uses a TDMA-based algorithm. 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, and thus no packet collision will occur. However, since it is likely that the network is affected by interference, a small probability for packet dropouts remains. The user that sends the packet can decide whether the receiver should return an acknowledgment on successful receipt of the packet or not [6].

During the CAP, all users are allowed to access the network using a slotted CSMA/CA algorithm. 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. Successful transmissions in the CAP are optionally confirmed using acknowledgments [6]. 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 have only 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 would, however, lead to a network overload. To avoid such issues and to maintain fairness in the CAP, we allow only the control loop to use a fixed amount of slots in the CAP of each superframe. This leads to the modified superframe structure shown in Fig. 2.

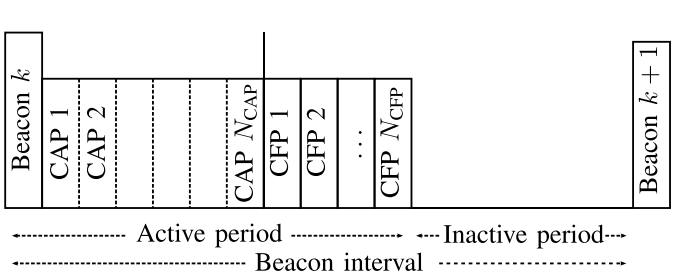

Fig. 2. IEEE 802.15.4 superframe where the CAP is divided into transmission slots.

In this paper, we utilize multiple superframes as described in the IEEE 802.15.4e standard [7]. Here, we utilize one superframe for the controller to transmit the control inputs to the actuators and one superframe to transmit the measurements from the sensors to the state estimator. This is shown in Fig. 4. The sampling time of the system is aligned to the beacon interval of the sensing superframe. Here, we assume that all control inputs are applied at the actuators just before the next sample is taken. This means that the controller computes the control input based on a prediction of the state one sample into the future. This means that the intertransmission times within a superframe do not affect the system. In addition, if failed transmissions are retransmitted successfully within the same superframe, the transmission delays do not affect the system.

In this setup, the controller computes the control input based on an estimation of the plant state one sample in the future.

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 are 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, and the time in the current CAP at which the user attempts their first transmission. This has to be then compared with 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 1: We consider transmissions in a CAP or a CFP that are delayed till the next superframe as being lost. These occur with fixed dropout probabilities p_{CAP} and p_{CFP}, respectively, where $p_{\text{CFP}} < p_{\text{CAP}}$. For simplicity, we consider the packet loss rates to be constant and known.

Remark 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 [6]. The control signal might therefore not be received by the actuator.[1] □

[1]Note that the probability for a beacon frame loss can easily be incorporated in the dropout probabilities p_{CAP} and p_{CFP}.

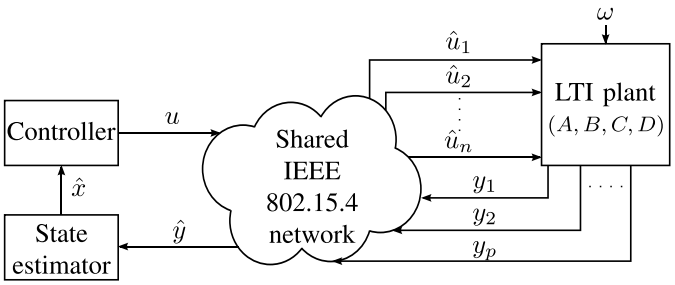

Fig. 3. Control system with many actuator and sensor nodes.

Remark 2: The packet dropout rate on the network is often time varying and will highly dependent on many factors including the network topology that is used, the data transfer rates, the frequency bands being used, and the interference hereon. These conditions can be described with a packet dropout rate, which can be formulated as a Markov chain to capture time-varying packet dropout rates and network conditions. The controller design is in this case, however, similar to the design with independent and identically distributed (i.i.d.) packet dropouts. For ease of exposure, we therefore assume i.i.d. packet dropouts in the controller design in the remainder of this paper. The packet dropout probability can be estimated based on the success rate of past transmissions. Another approach is that one can estimate the dropout probability based on the expected load of the network and keep it constant. The latter can, for example, be done using Markov chain models as suggested in [42]–[44]. For multihop networks, this becomes more involving [45]. □

III. Networked Control Architecture of Interest

We consider a spatially distributed large control system with many sensors and actuators, which is shown in Fig. 3. 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, which then produces the state estimate $\hat{x}$. In this paper, we use the optional acknowledgments from the IEEE 802.15.4 standard to confirm successful transmissions to the actuators. This means that both the controller and the state estimator know the transmission outcomes of u.

The IEEE 802.15.4 standard supports multiple superframes. In this work, we define two superframes: one sensing superframe in which the sensor measurements are transmitted to the estimator and one actuation superframe in which the control signals are transmitted to the actuators. The sampling interval of the system is, as described in Section II, aligned to the beacon interval of the sensing superframe. This is illustrated in Fig. 4.

The actuator signals that the controller calculates from the measure at time k are first applied to the plant at time $k+1$ when the next measurement is taken (see Fig. 4). This means that the controller at time k has to compute the control inputs that are going to be applied to the plant at time $k+1$. Since the estimator at time k does not have any information on the state at time $k+1$, the controller instead gets the prediction of the state at time $k+1$ based on the state at time k, $\hat{x}(k+1|k)$. Since we use acknowledgments for successful transmissions, the estimator knows whether the control signals are transmitted successfully or not. The separation principle therefore holds [46], and we can design the controller assuming that we have state feedback and afterward replace it with the estimated state $\hat{x}(k|k-1)$.

The state estimation problem can be viewed as a dual problem of the control problem. Scheduling for state estimation has been studied recently in [47]. In this paper, we consider the control side and use a state estimator that handles intermittent observations. Since we consider only the scheduling of actuators, we assume that all sensor readings get transmitted at every sample instance. More on state estimation with intermittent observations can be found in [31] and [48]–[50]. In this paper, we focus on the scheduling of the actuators. For ease of exposition and since scheduling of the sensor measurements can be seen as the dual problem of scheduling the actuators, we assume that all sensor measurements are jointly transmitted in a dedicated superframe that is located right before the beacon of the actuator superframe. This is illustrated in Fig. 4. We further transmit all sensor data through the less reliable transmission mode, the CAP.

The next state of the plant is given by

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

and the output is given by

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

where $x(k) \in \mathbb{R}^m$, $\hat{u}(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$, and A, B, and C are matrices of appropriate dimensions. The plant disturbance $\omega(k) \sim \mathcal{N}(0_m, \Sigma_\omega)$ and measurement disturbance $\nu(k) \sim \mathcal{N}(0_p, \Sigma_\nu)$ are zero-mean Gaussian with covariances Σ_ω and Σ_ν, respectively.

The state estimation is performed as described in [48] and [50], leading to the optimal estimator

$$\begin{aligned} \hat{x}(k+1|k) &= A\hat{x}(k|k-1) + B\hat{u}(k) \\ &\quad + K(k)(\hat{y}(k) - C(k)\hat{x}(k|k-1)) \\ P(k+1|k) &= AP(k|k-1)A^T + \Sigma_\omega \\ &\quad - K(k)C(k)P(k|k-1)A^T \end{aligned} \tag{3}$$

where $\hat{y}_i(k) = \psi_i(k)y_i(k)$, with $\psi_i(k) = 1$ if the transmission from sensor i is successful and $\psi_i(k) = 0$ otherwise. Furthermore

$$\begin{aligned} K(k) \triangleq\ & AP(k|k-1)C(k)^T \\ & \times (C(k)P(k|k-1)C(k)^T + \Sigma_\nu)^{-1} \end{aligned} \tag{4}$$

is the Kalman gain. In (3), the random observation matrix is given by

$$C(k) \triangleq \begin{bmatrix} \psi_1(k)C_1 \\ \psi_2(k)C_2 \\ \vdots \\ \psi_p(k)C_p \end{bmatrix} \tag{5}$$

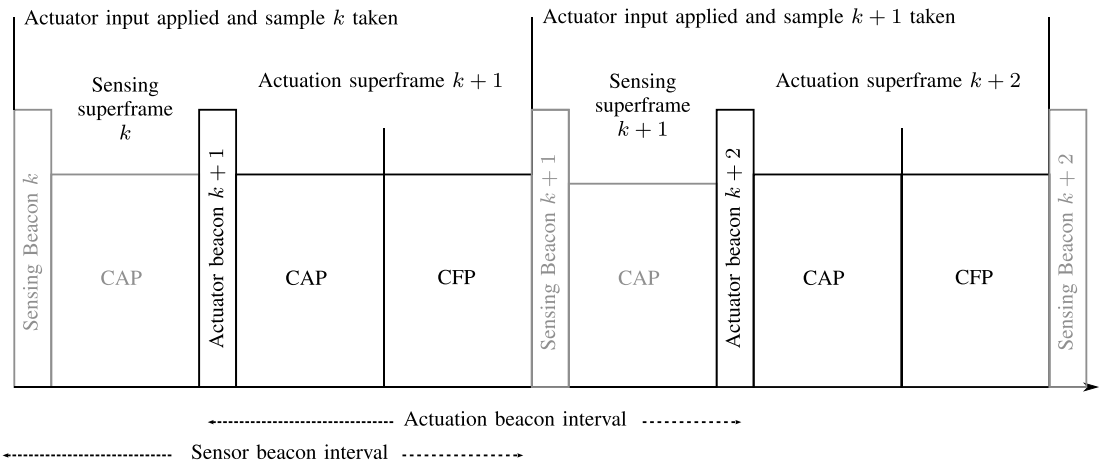

Fig. 4. Superframe structure for the networked control setup. Here, the sensor measurements at time step k are transmitted in the sensing superframe, which finishes right before the actuation superframe commences. The state estimator then predicts the state $\hat{x}(k+1|k)$ that the controller uses to compute the control input $u(k+1)$ and schedule $S(k+1)$.

and

$$\mathbf{Pr}\{\psi_i(k)=1\}=1-p_{\text{est}},\quad i=\{1,2,\ldots,p\}. \tag{6}$$

This state estimator is, as mentioned earlier, aware of failed transmissions in the controller–actuator link and knows the applied control signal $\hat{u}(k)$.

As already mentioned in Section II, the actuators are to be controlled over an IEEE 802.15.4 compatible network, which can support only a limited amount of data during the CAP and CFP. We are interested in a situation where the combined amount of available slots in the CAP and CFP in each superframe is less than the amount of actuator nodes in the system. 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. 4). Recall further from Assumption 1 that a packet that at time k is delayed until a future superframe is considered as being lost. The probabilities for these delays to occur are i.i.d. and are included in the packet dropout probability p_{CAP} or p_{CFP}.

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

$$S(k)=[s_1(k),s_2(k),\ldots,s_n(k)]^T. \tag{7}$$

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 and 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

$$s_j(k)\in\{0,1-p_{\text{CAP}},1-p_{\text{CFP}}\} \tag{8}$$

where p_{CAP} and p_{CFP} are the (known or estimated) packet dropout probabilities in the CAP and CFP, respectively.

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

$$\mathbf{Pr}\{\gamma_j(k)=1\}=s_j(k),\quad j\in\{1,2,\ldots,n\}. \tag{9}$$

There are various possibilities of what the actuator should do in case either it is not addressed at time k or the transmission of the packet containing the control signal failed. The simplest and most common options are set-to-zero and zero-order-hold strategies, where generally neither is superior to the other [51].

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

$$\hat{u}_j(k)=\begin{cases}u_j(k), & \text{if } \gamma_j(k)=1\\ 0, & \text{if } \gamma_j(k)=0.\end{cases} \tag{10}$$

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

$$b_j^{(k)}=\begin{cases}b_j, & \text{if } s_j(k)>0\\ 0, & \text{if } s_j(k)=0\end{cases} \tag{11}$$

we can combine (1) and (10) to obtain the NCS model

$$x(k+1)=Ax(k)+B(\gamma(k),k)u(k)+\omega(k). \tag{12}$$

Example: If in (1) the input matrix $B=I_n$ and the schedule at time k is given by

$$S(k)=[1-p_{\text{CFP}},1-p_{\text{CAP}},1-p_{\text{CAP}},0]^T \tag{13}$$

then $B^{(k)}=\text{diag}\{[1,1,1,0]\}$. If the channel outcomes at time k are given by $\gamma(k)=[1,1,0,0]^T$, then the actuators that are addressed at the plant are given by $B(\gamma(k),k)=\text{diag}\{[1,1,0,0]\}$. □

Remark 3: When using a hold-input strategy, the actuator logic can be equipped with an integrator such that the controller transmits only the changes in the actuator input. This can be incorporated in the current framework using state

augmentation such that the augmented state contains both the system state and the values at the integrators. The A matrix is in this case extended to include the integrators at the actuators. Alternatively, the controller could predict a sequence of future control inputs that all are transmitted at once and stored in a buffer at the actuator. In case no new control signal is received, the controller takes a sample from the buffer (see more in [52]). The actuator logic can further be extended such that it filters the control output to prevent spikes in the actuator positions. More details on prefiltering and postfiltering of the control packets can be found in [53] and [54]. □

Within the current setup, scheduling amounts to designing the schedule $S(k)$ in (7). 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 does not exceed the number of available GTSs in a superframe. Otherwise, the CAP needs to be used as well. Further, if the number of actuators exceeds the total amount of available slots in the CFP and CAP, 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 solve this scheduling and controller codesign problem, ideally one would like to utilize the scheduling sequence and control policies that minimize an infinite horizon LQ cost function. It is in this case, though, not feasible to evaluate an infinite horizon cost function since this results in an infinitely long scheduling sequence and associated control policy. The matrix $B(\gamma(k), k)$ is further stochastic and depends on $S(k)$ for each k. Due to these uncertainties in the system, there might not exist a solution to the infinite horizon cost function (see [55], [56], and references therein). We will therefore use the ideas of MPC with a finite horizon cost function with a suitably chosen final state weighting and a periodic approach to an infinite horizon cost function. It remains future work to analyze under which conditions a solution to the infinite horizon cost function will exist.

IV. Stochastic Control Formulation

In this section, we propose to use a finite horizon cost function to solve the controller–scheduler codesign problem. We consider the LQ cost for a finite scheduling sequence

$$\vec{S}(k) \triangleq \{S(k), S(k+1), \ldots, S(k+N-1)\} \tag{14}$$

where $S(k)$ is as in (7)

$$J(x(k), \vec{S}(k), \pi(\vec{S}(k))) = \mathbf{E}\left\{ \|x(k+N)\|_W^2 + \sum_{\ell=0}^{N-1} \|x(k+\ell)\|_Q^2 + \|u(k+\ell)\|_R^2 \,\middle|\, x(k), \vec{S}(k) \right\} \tag{15}$$

where Q and R are symmetric positive definite matrices that penalize the cost of the state and the cost of control, respectively, and W is a symmetric positive definite matrix that penalizes the terminal cost at time N. More details on how can one choose the final state weighting W will be given in Section V-D. Further, the control policy is given by

$$\pi(\vec{S}(k)) = \{\mu(x(k), \vec{S}(k)), \ldots, \mu(x(k+N-1), \vec{S}(k+N-1))\}$$

where each control law $\mu(x(k), \vec{S}(k))$ maps the state $x(k)$ into control signals $u(k)$, such that $u(k) = \mu(x(k), \vec{S}(k))$. This cost function uses the actual state of the plant $x(k)$. [In the simulations in Section VI, these are replaced by estimates, $\hat{x}(k)$.]

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

$$\pi^* = \{\mu^*(x(k), \vec{S}^*(k)), \ldots, \mu^*(x(k+N-1), \vec{S}^*(k+N-1))\}$$

the cost function (15) has to be evaluated for every possible scheduling sequence. For n actuators and N_{CFP} and N_{CAP} slots in the CFP and CAP, respectively, that are available to the controller, we have for a horizon of length N

$$\left(\frac{n!}{N_{\text{CFP}}!(n-N_{\text{CFP}})!} \frac{(n-N_{\text{CFP}})!}{N_{\text{CAP}}!(n-N_{\text{CFP}}-N_{\text{CAP}})!} \right)^N \tag{16}$$

possible scheduling sequences. Note that since the network is shared with other users, we restrict the controller to use only a limited amount of slots in the CAP and CFP. In (16), the variables N_{CFP} and N_{CAP} represent the amount of slots that the controller is allowed to use in every superframe.

A. Solution

To state the optimal solution of (15), we first note that using (9) and (11), the expectation of $b_j(k)$ is given by $\mathbf{E}\{b_j(\gamma_j(k), k) \mid s_j(k)\} = \mathbf{Pr}\{\gamma_j(k) = 1 \mid s_j(k)\} b_j^{(k)} = s_j(k) b_j$, where the last equality comes from the fact that $s_j(k)$ contains the (estimated or known) probability of a successful transmission (8), (9).

Theorem 1: Suppose that the pair (A, B) is controllable. Then for a finite scheduling sequence $\vec{S}(k)$

$$\min_{\pi(\vec{S}(k))} J(x(k), \vec{S}(k), \pi(\vec{S}(k))) = x(k)^T P(\vec{S}(k)) x(k) + \sum_{\ell=1}^{N} \text{trace}(\Sigma_\omega P(\vec{S}(k+\ell))). \tag{17}$$

In (17), $P(\vec{S}(k))$ is given by the recursion

$$P(\vec{S}(k)) = Q + A^T P(\vec{S}(k+1)) A - A^T P(\vec{S}(k+1)) B \,\text{diag}\{S(k)\}\, L(\vec{S}(k)) \tag{18}$$

where

$$L(\vec{S}(k)) = (R + \text{diag}\{S(k)\} B^T P(\vec{S}(k+1)) B)^{-1} \times \text{diag}\{S(k)\} B^T P(\vec{S}(k+1)) A \tag{19}$$

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

$$\pi^*(\vec{S}(k)) = \{\mu^*(x(k), \vec{S}(k)), \ldots, \mu^*(x(k+N-1), \vec{S}(k+N-1))\} \tag{20}$$

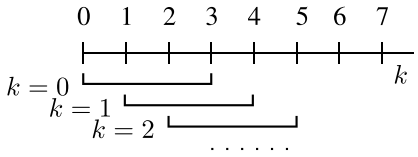

Fig. 5. Illustration of MPC where $N = 3$.

where each control law is defined by

$$\mu^*(x(k), \vec{S}(k)) \triangleq -L(\vec{S}(k))x(k). \tag{21}$$

Proof: The proof is based on the results in [56] and is given in Appendix A. ∎

Using Theorem 1, we can minimize the cost function (17) for any given scheduling sequence $\vec{S}(k)$. This allows us to solve the optimal controller and scheduler codesign problem by jointly minimizing (17) for the optimal schedule and control signal as

$$\vec{S}^*(k) = \underset{\vec{S}(k)}{\operatorname{argmin}} \left[x(k)^T P(\vec{S}(k))x(k) + \sum_{\ell=1}^{N} \operatorname{trace}(\Sigma_\omega P(\vec{S}(k+\ell))) \right]. \tag{22}$$

Clearly, (22) is a combinatorial problem where the amount of possible scheduling sequences for a horizon length N is given by (16). The solutions to the recursion (18) and the sum term in (22) can be calculated offline for all possible scheduling policies and stored in a lookup table. The size of this lookup table can be computed using (16) and depends on the amount of available actuators n, the number of available slots N_{CAP}, N_{CFP}, and the horizon length N. In this case, the computations for each scheduling sequence reduce to evaluating the quadratic term (22) in $x(k)$ and add a constant.

Remark 4: To find the optimal schedule, an exhaustive search has to be performed. Many works that considered actuator selection and scheduling over networks came to this conclusion [20]–[22]. These works propose to use simulated annealing and branch and bound algorithms to solve the combinatorial problem. A heuristic that does not necessarily result in the optimal schedule is presented in [20]. □

B. Algorithm

The finite horizon algorithm is implemented as MPC, where at every time step the (estimated) state is used to solve the optimization problem (22). An illustration of this principle is shown in Fig. 5. The algorithm is described in Algorithm 1, which is executed at every sample instance. Here, (22) is solved at every time step, and the control signal $u(k)$ is given by $u(k) = \mu(x(k), \vec{S}^*(k))$.

Studies show that a longer prediction horizon N results in better control performance [57]. This, however, results in an exponentially increased complexity to find the optimal scheduling sequence (16). It is therefore desirable to maintain the horizon length N relatively short. Another parameter that has an impact on the control performance is the weighting of the terminal cost W. By choosing W with care, one can obtain significant improvements in the performance of the NCS. This motivates the use of an infinite horizon approximation, which is presented next. Further discussions on how to design the final state weighting can be found in Section V-D.

Algorithm 1 MPC—Finite Horizon

1) At time k the estimator computes $\hat{x}(k+1|k)$ using the measurement $y(k)$.
2) The controller computes $\overrightarrow{S}^*(k+1)$ using (22) and $\pi^*(k+1)$ using (21).
3) The controller sends $u^*(k+1) = \mu^*(\hat{x}(k+1), \overrightarrow{S}^*(k+1))$, where $\mu^* \in \pi^*(\overrightarrow{S}^*(k+1))$, over the network.
4) At time $k+1$ the actuators apply the successfully transmitted control signals to (1).

V. Periodic Approximation

In this section, we propose to approximate an optimal infinite length scheduling sequence and associated control laws by assuming that a scheduling sequence $\vec{S}(i)$ of finite length T is repeated periodically. Here, i is the periodic index, which for time index k and a period of length T, is given by $i = \text{rem}\{k/T\}$, where rem {} is the remainder after division. This allows us to state an infinite horizon LQ cost function. The idea of approximating an infinite horizon cost using a periodic sequence is inspired by Henriksson *et al.* [35], [36] and Orihuela *et al.* [38], where they use this to find the sampling interval that minimizes a deterministic LQ cost function. In this paper, we adapt this idea and approximate the infinite horizon cost function by assuming that a periodic scheduling sequence is repeated as the horizon goes to infinity.

In this section, we consider the discounted cost for a finite periodic scheduling sequence

$$\vec{S}(i) = \{S(0), S(1), \ldots, S(T-1)\}$$

given by

$$J_\infty(x(k), \vec{S}(i), \pi(\vec{S}(i))) = \lim_{N\to\infty} \mathbf{E}\left\{ \sum_{\ell=0}^{N-1} \alpha^\ell \left(\|x(k+\ell)\|_Q^2 + \|u(k+\ell)\|_R^2 \right) \middle| x(k), \vec{S}(i) \right\} \tag{23}$$

where the discounting factor $\alpha \in (0, 1)$.

A. Solution

Before we state the solution to (23), we will state the necessary conditions that need to be fulfilled for a stabilizing solution to exist.

Lemma 1: For a scheduling sequence $\vec{S}(k)$, the time-varying system $(A, B(\gamma(k), k))$ is N-step controllable over the time interval $(k, k+N)$ if the system (A, B) is controllable and

$$\operatorname{rank}\{\Gamma(N, \vec{S}(k))\} = m \tag{24}$$

where

$$\Gamma(N, \vec{S}(k)) \triangleq [\bar{B}(k+N-1), A\bar{B}(k+N-2), \ldots, A^{N-1}\bar{B}(k)] \tag{25}$$

where $\bar{B}(k) = B(I_n, k)$.

Proof: A proof for controllability of periodic systems is given in [58] and can be applied directly. The time-varying matrices $\bar{B}(k)$ are given by the scheduling sequence $\vec{S}(k)$. ■

Let $\mathbb{S}$ denote the set containing the length T scheduling sequences that satisfy Lemma 1.

Theorem 2: Suppose that there exists a scheduling sequence $\vec{S}(i) \in \mathbb{S}$ such that the pair $(A, B(\gamma(i), i))$ is controllable. Then

$$\min_{\pi(\vec{S}(k))} J_\infty(x(k), \vec{S}(i), \pi(\vec{S}(i))) = x(k)^T P_{\text{per}}(\vec{S}(i))x(k) + \sum_{\ell=1}^{\infty} \alpha^\ell \operatorname{trace}(\Sigma_\omega P_{\text{per}}(\vec{S}(i+\ell))) \tag{26}$$

where $P_{\text{per}}(\vec{S}(i))$ is the periodic positive definite solution with period length T to the periodic Riccati equation

$$P_{\text{per}}(\vec{S}(i)) = Q + \alpha A^T P_{\text{per}}(\vec{S}(i+1))A - \alpha A^T P_{\text{per}}(\vec{S}(i+1))B \operatorname{diag}\{S(i)\} L_{\text{per}}(\vec{S}(i)) \tag{27}$$

and

$$L_{\text{per}}(\vec{S}(i)) = \alpha(R + \alpha \operatorname{diag}\{S(i)\} B^T P_{\text{per}}(\vec{S}(i+1))B)^{-1} \times \operatorname{diag}\{S(i)\} B^T P_{\text{per}}(\vec{S}(i+1))A. \tag{28}$$

The minimizing control policy is then given by

$$\pi^*_{\text{per}}(\vec{S}(i)) = \left\{\mu^*_{\text{per}}(x(k), \vec{S}(i)), \ldots, \mu^*_{\text{per}}(x(k+T-1), \vec{S}(T-1))\right\} \tag{29}$$

where each control law is defined by

$$\mu^*_{\text{per}}(x(k), \vec{S}(i)) \triangleq -L_{\text{per}}(\vec{S}(i))x(k). \tag{30}$$

Proof: Following the procedure as in the proof of Theorem 1 and using the fact that the Riccati equation becomes periodic due to the periodic scheduling sequence, the periodic Riccati equation and solutions hereof have been analyzed extensively in [59]–[61]. ■

Remark 5: The periodic Riccati equation (27) can be reformulated as a time-invariant Riccati equation using lifting or a cyclic system description [58]. We illustrate a cyclic system description in Appendix B to obtain a time-invariant version of (27). This time-invariant Riccati equation can then be solved either by initializing $\hat{P} > 0$ and then iterating until the solution converges or by using Newton's method [62]. □

Algorithm 2 Periodic—Infinite Horizon

1) At time $k = 0$ the estimator computes $\hat{x}(1|0)$ using the measurement $y(0)$.
2) The controller computes the periodic scheduling sequence and control laws $\overrightarrow{S}^*(1)$ and $\pi^*_{\text{per}}(\overrightarrow{S}(1))$ using (30) and (31), respectively.
3) The control signal $u^*(k+1) = \mu^*_{\text{per}}(\hat{x}(k+1), \overrightarrow{S}^*(i))$, where $\mu^*_{\text{per}} \in \pi^*_{\text{per}}(\overrightarrow{S}^*(1))$, is calculated and sent over the network.
4) At time $k+1$, the actuators apply the successfully transmitted control signals to (1).

Algorithm 3 MPC—Infinite Horizon

1) At time k the estimator computes $\hat{x}(k+1|k)$ using the measurement $y(k)$.
2) The controller computes the periodic scheduling sequence $\overrightarrow{S}^*(i)$ and optimal control policy $\pi^*_{\text{per}}(\overrightarrow{S}(i))$ using (30) and (31), respectively.
3) The control signal $u^*(k+1) = \mu^*_{\text{per}}(\hat{x}(k+1), \overrightarrow{S}(i))$, where $\mu_{\text{per}} \in \pi^*_{\text{per}}(\overrightarrow{S}^*(i))$, is calculated and sent over the network.
4) At time $k+1$, the actuators apply the successfully transmitted control signals to (1).

The optimal scheduling sequence can, using Theorem 2 and the argument as in Section IV-A, be found by

$$\vec{S}^*(i) = \underset{\vec{S}(i)\in\mathbb{S}}{\operatorname{argmin}} \left[x(k)^T P_{\text{per}}(\vec{S}(i))x(k) + \sum_{\ell=1}^{\infty} \alpha^\ell \operatorname{trace}(\Sigma_\omega P_{\text{per}}(\vec{S}(i+\ell))) \right]. \tag{31}$$

The amount of schedules that need to be compared in every iteration depends in this case on the choice of T and can be found by substituting N with T in (16).

B. Algorithms That Provide an Optimal Solution to (26)

We develop three algorithms that utilize the results of Theorem 2. Of these algorithms, one will only minimize the infinite horizon cost function based on the information of the first sample to obtain the optimal scheduling and control policies, which then are applied periodically. This algorithm has a low computational cost since the combinatorial search in (31) has only to be performed once. This is presented in Algorithm 2, which only is executed at time $k = 0$.

The next algorithm uses MPC such that a new control policy and scheduling sequence are computed every time a new measurement is received. This algorithm is presented in Algorithm 3 and is executed at every sample instance. Here, the scheduling sequence is periodic with period T and minimizes (26). Only the first schedule and control law are applied, since a new scheduling sequence and control law are computed when the next sample arrives.

Algorithm 4 MPC—Greedy

1) At time k the estimator computes $\hat{x}(k+1|k)$ using the measurement $y(k)$.
2) The controller finds which actuators to address during the CFP by
$$\overrightarrow{S}^*_{\text{CFP}}(i) = \underset{\overrightarrow{S}_{\text{CFP}(i)}}{\operatorname{argmin}} J\left(\hat{x}(k+1), \overrightarrow{S}_{\text{CFP}}(i), \pi(\overrightarrow{S}_{\text{CFP}}(i))\right).$$
3) The controller finds which of the remaining actuators to address during the CAP using
$$\overrightarrow{S}^*_{\text{CAP}}(i) = \underset{\overrightarrow{S}_{\text{CAP}(i)} \notin \overrightarrow{S}^*_{\text{CFP}(i)}}{\operatorname{argmin}} J\left(\hat{x}(k+1), \bar{S}, \pi(\bar{S})\right),$$
where $\bar{S} \triangleq \overrightarrow{S}^*_{\text{CFP}}(i) \bigcup \overrightarrow{S}_{\text{CAP}}(i)$.
4) The scheduling sequence is then given by $\overrightarrow{S}^*(i) = \overrightarrow{S}^*_{\text{CFP}}(i) \bigcup \overrightarrow{S}^*_{\text{CAP}}(i)$, and the control policy by $\pi^*_{\text{greedy}}(\overrightarrow{S}^*(i)) = \pi^*(\overrightarrow{S}^*_{\text{CFP}}) \bigcup \pi^*(\overrightarrow{S}^*_{\text{CAP}}(i))$.
5) The control signal $u^*(k) = \mu^*(\hat{x}(k+1), \overrightarrow{S}^*(i))$, where $\mu^* \in \pi^*_{\text{greedy}}(\overrightarrow{S}^*(i))$, is sent over the network.
6) At time $k+1$, the actuators apply the successfully transmitted control signals to (1).

C. Algorithm That Provides a Suboptimal Solution to (26)

To reduce the amount of calculations needed to find the optimal scheduling sequence in Algorithm 3, we propose a greedy algorithm. The cost of this is, however, that the scheduling algorithm is suboptimal compared to Algorithm 3. The idea with this greedy algorithm is that it first finds the N_{CFP} actuators that minimize (26) and assigns these to GTSs. Afterward, it finds the N_{CAP} actuators that minimize the remainder of the cost and assigns these in the CAP. This reduces the size of the lookup table in (31) to

$$\left(\frac{n!}{N_{\text{CFP}}!(n-N_{\text{CFP}})!}\right)^T + \left(\frac{(n-N_{\text{CFP}})!}{N_{\text{CAP}}!(n-N_{\text{CFP}}-N_{\text{CAP}})!}\right)^T$$

entries. The implementation of this idea is presented in Algorithm 4.

Remark 6: For the greedy algorithm, the periodic Riccati equations have to be computed for both all possible $\vec{S}_{\text{CFP}}$ and $\vec{S}_{\text{CAP}}$. While this can be done offline, one might in some situations be unable to find positive definite solutions to (27) for $\vec{S}_{\text{CFP}}$ in which case Algorithm 4 will fail. □

D. Choice of Final State Weighting for the Finite Horizon Cost

In this section, we investigate, based on the presented ideas on periodic scheduling, some ideas on the choice of the final state weighting in (15). We suggest that a good choice of final state weighting W in (15) could be to use a solution to the periodic Riccati equation (27). Since for $k \to \infty$ the system will converge toward a steady state that is affected only by the disturbance $\omega(k)$, intuitively a good choice of final state weighting could be the solution to (27) that minimizes (23) when $x(k)$ is at steady state, i.e., $x(k) = \mathbf{0}_m$. In this case, the final state weighting can be computed by

$$\vec{S}^*_W(i) = \underset{\vec{S}(i)}{\operatorname{argmin}} \min_{\pi(\vec{S}(i))} J_\infty(\mathbf{0}_m, \vec{S}(i), \pi(\vec{S}(i))) \tag{32}$$

$$= \underset{\vec{S}(i)}{\operatorname{argmin}} \sum_{\ell=1}^{\infty} \alpha^\ell \operatorname{trace}(\Sigma_\omega P_{\text{per}}(\vec{S}(i+\ell))) \tag{33}$$

where

$$W \triangleq P_{\text{per}}(\vec{S}^*_W(i)). \tag{34}$$

The question that remains is how long the period T should be. Longer period lengths increase the amount of offline computations that need to be performed but do not necessarily result in better performance (see Fig. 6). Further, now both N and T have an influence in the performance of the controller–scheduler. Analytical analysis of the impact of N and T on the performance is, due to the stochastic switching and scheduling in the system, a very difficult problem to solve when the switching is only implicitly characterized via an optimization problem since not only the control input but also the switching is state dependent. We therefore perform Monte Carlo simulations to illustrate the impact of the choices of N and T on the performance of the controller–scheduler. We simulate an NCS with three actuators that compete for one GTS and one slot in the CAP. The parameters of the system models (1) and (2) are given by

$$A = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 \\ 0 & 1 & 1.1 & 0 & 0 \\ 0 & 0 & 0 & 0.819 & 0 \\ 0 & 0 & 0 & 0.906 & 1 \end{bmatrix} \tag{35}$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 9.063 \\ 0 & 0 & 4.683 \end{bmatrix}. \tag{36}$$

The plant disturbance $\omega(k) \sim \mathcal{N}(\mathbf{0}_5, I_5)$. We can observe all states such that $C = I_5$ and the measurement noise $v(k) \sim \mathcal{N}(\mathbf{0}_5, I_5)$. The weighting matrices Q, R, and W are all identity matrices of appropriate dimensions. This system has open-loop eigenvalues at 1.5, 1, and 0.82 and two eigenvalues located at 1.1. We use the method described in Remark 3 when simulating the system with a hold-input strategy. We fix the probabilities of transmission failures to constant values to illustrate the effect of packet loss in the closed-loop system. These probabilities for transmission failures are given by $p_{\text{CFP}} = 0.05$ and $p_{\text{CAP}} = p_{\text{est}} = 0.25$. Furthermore, the discounting factor is set to $\alpha = 0.99$.

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

$$\frac{1}{M}\sum_{k=0}^{M} x(k)^T Q x(k) + u(k)^T R u(k) \tag{37}$$

where Q and R are the weighting matrices of the state and control signal, respectively, and M is the length of

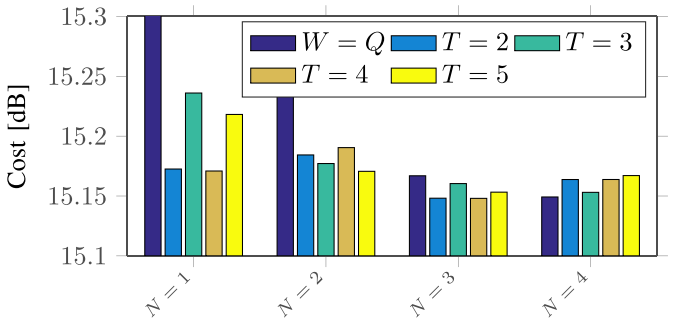

Fig. 6. Impact of the choice of T and N on the performance of Algorithm 1 when W is chosen using (34). For comparison, the final state weighting $W = Q$ is also shown. The empirical cost is evaluated using (37). When $W = Q$ and $N = 1$, the cost is 17.38 dB and is shortened for easier viewing.

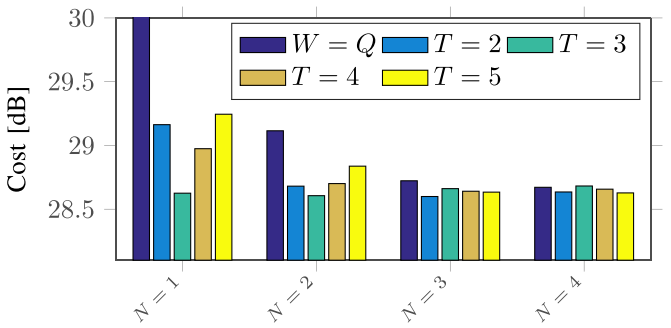

Fig. 7. Impact of the choice of T and N on the performance of Algorithm 1 when W is chosen using (34). Here, the NCS is affected by unknown additive disturbances in the form (38). The empirical cost is evaluated using (37). Simulations for $W = Q$ are shown for comparison. The empirical cost for $N = 1$ with $W = Q$ is 32.85 dB and is truncated for easier viewing.

the simulation. This is averaged over 1000 simulations, each of $M = 1000$ time steps, of the same system with different initial conditions and noise realizations.

The results in Fig. 6 show that for large N, the best performance gains are obtained when $W = Q$. However, for small N, using W as in (31) results in significant performance gains. Fig. 6 further shows that increasing T does not necessarily result in a better performance. The impact of the choice of T on the performance is less than 0.1 dB when $N = 1$. This difference decreases as N increases. There are a few combinations of parameters that provide a better performance than others, here among $N = 2$ with $T = 5$ and $N = 3$ with $T = 2$ or 4. When $N > 3$, the performance gain of using W as in (34) compared with that of using $W = Q$ diminishes. The performance gains at small N are, on the other hand, significant for all combinations of N and T. Also for $N = 1$ with the proper selection of T, the performance is less than 0.1 dB worse than when $N = 3$ or $N = 4$. This means that the number of schedules that need to be compared in Algorithm 1 is reduced significantly while not sacrificing any significant performance.

To further illustrate the importance of online scheduling when the system, besides Gaussian noise, is affected by random and unmeasured additive disturbances, we add the disturbance $\chi(k) \in \mathbb{R}^m$ to (1). Here, every element in $\chi(k)$ is given by

$$\chi_i(k) = \rho_i(k)\upsilon_i(k) + \iota_i(k)\chi_i(k-1) \tag{38}$$

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 $\chi_i(-1) = 0$. The noise generated by (38) is a step of random magnitude $\upsilon_i(k)$ that occurs at every time step with probability ρ_i and lasts l time steps with probability p_{length}^{l-1}.

Fig. 7 shows the results when the NCS is affected by random unknown additive disturbances in the form of (38). In this case, the performance gains of choosing the final state weight W as per (34) are larger than in Fig. 6 for $N < 4$. Note also that a very good performance is obtained for $N = 1$ and $N = 2$ with $T = 3$ and that in this case, the performance is similar to the best performance for $N = 3$ and $N = 4$. Especially, the choice of $N = 1$ with $T = 3$ results in a very good performance for a low online computational cost compared with the performance for large N.

Algorithm 5 Branch and Bound Approach Used for Simulations

1) Input $x(k)$.
2) Solve $J_{\min} = J(x(k), \overrightarrow{S}_1, \pi(\overrightarrow{S}_1))$ and set $i = 2$.
3) If $J_{\min} < J(0, \overrightarrow{S}_i, \pi(\overrightarrow{S}_i))$, optimal scheduling sequence and control laws found. Stop the search.
4) Else $J_{\min} = \min\left\{J_{\min}, J\left(x(k), \overrightarrow{S}_i, \pi(\overrightarrow{S}_i)\right)\right\}$.
5) Increment i and go to Item 3.

VI. Simulation Studies

In this section, we compare the schedule and control design algorithms presented in Sections IV and V through simulations. We first shortly present a branch and bound approach that can significantly reduce the amount of computations needed in the exhaustive search. This is followed by a simulation where three actuators compete for one GTS and one slot in the CAP, after which we simulate a scenario where nine actuators compete for one GTS and seven slots in the CAP.

A. Branch and Bound Algorithm

Although the presented algorithms require an exhaustive search to be performed, the amount of searches required can be reduced significantly using a simple branch and bound algorithm. When the system is in steady state and $x(k)$ is zero, the only part contributing to the costs (15), (23) is the constant part that depends on the variance of the disturbance and the selected schedule. These parts can be calculated offline. By sorting the scheduling sequences, such that the constant part is sorted from low to high values, (22) and (31) can be terminated when

$$J(x(k), \vec{S}_m, \pi(\vec{S}_m)) \leq J(0, \vec{S}_n, \pi(\vec{S}_n)), \quad m < n$$

where n and m are the indexes of the scheduling sequences in the sorted list. The algorithm is illustrated in Algorithm 5.

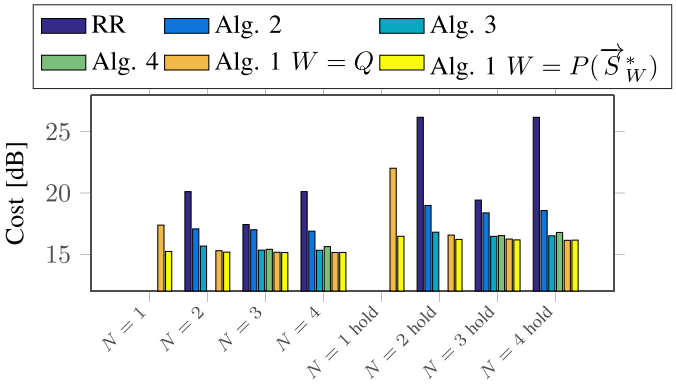

Fig. 8. Empirical cost of the online algorithms analyzed with (37) averaged over 1000 simulations. The results are shown for set-to-zero and hold-input at the actuators.

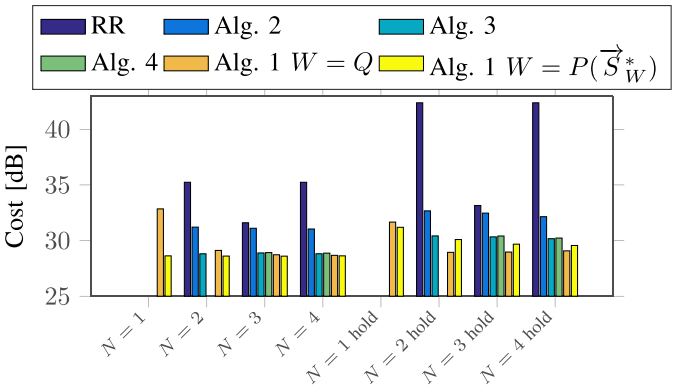

Fig. 9. Empirical cost of the online algorithms analyzed with (37) averaged over 1000 simulations. The results are shown for set-to-zero and hold-input at the actuators for the system affected by the disturbance (38).

B. Small Coupled Systems

In this section, we compare the performance of the presented controller–scheduler codesign algorithms through simulation studies. We use the parameters that are presented in Section V-D and compare the performance using the empirical cost (37). We compare the developed algorithms with a simple round robin (RR) algorithm, where the access to the CAP and CFP are shared equally among the actuators.

The RR schedules are chosen as

$$\vec{S}_2 = \left\{ \begin{bmatrix} 1-p_{\text{CFP}} \\ 1-p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} 1-p_{\text{CFP}} \\ 0 \\ 1-p_{\text{CAP}} \end{bmatrix} \right\}$$

$$\vec{S}_3 = \left\{ \begin{bmatrix} 1-p_{\text{CFP}} \\ 1-p_{\text{CAP}} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1-p_{\text{CFP}} \\ 1-p_{\text{CAP}} \end{bmatrix} \begin{bmatrix} 1-p_{\text{CAP}} \\ 0 \\ 1-p_{\text{CFP}} \end{bmatrix} \right\}$$

while $\vec{S}_4 = \{\vec{S}_2, \vec{S}_2\}$.[2]

The simulations that are performed using Algorithm 1, where the final state weighting is designed as described in Section V-D ($W = P(\vec{S}^*_W)$), use the period lengths that result in the best performance in Figs. 6 and 7.

The empirical cost of the algorithms, calculated using (37), is shown in Fig. 8. Simulations where the NCS (1) is affected by the disturbance (38) are shown in Fig. 9.

The results show that the online scheduling algorithms significantly perform better than offline scheduling algorithms (RR and Algorithm 2). The reason for this is that the schedule can adapt quickly by addressing actuators that control the subsystems where disturbances occur. It is further seen that when $N \neq 3$, Algorithm 2 performs significantly better than the simple RR algorithm. When $N = 3$, the performance of RR is only slightly worse. The reason for this is that when $N = 3$, the amount of actuators fits the period length.

When considering the online algorithms, the greedy algorithm (Algorithm 4) performs slightly worse than Algorithm 3. Using Algorithm 1 with a proper choice of W results in the best performance among all the presented algorithms. Here, as described in Section V-D, the performance gain by choosing $W = P(\vec{S}^*_W)$ over $W = Q$ diminishes for larger N. Also note that Algorithm 1 is the only algorithm that can be used for the given system when the horizon length $N = 1$. The reason for the other algorithms to fail at $N = T = 1$ is that it is not possible to find a positive definite solution to the Riccati equation (27), since the system when using a schedule of length $T = 1$ is not controllable according to Lemma 1. For this reason, Algorithm 4 is also not able to find solutions when $T = 2$ or the hold strategies are used in the actuators. It is seen that while the performance of Algorithm 3 remains fairly constant for larger T, the performance of Algorithm 1 increases for larger N and outperforms Algorithm 3.

TABLE I

MAXIMUM NUMBER OF SCHEDULING SEQUENCES TO EVALUATE IN (22) AND (31) CALCULATED USING (16). THE ACTUAL NUMBER CAN BE LESS THAN THAT SHOWN IN THE TABLE FOR ALGORITHMS 3 AND 4 SINCE SCHEDULING SEQUENCES THAT RESULT IN AN UNCONTROLLABLE SYSTEM ARE OMITTED

Horizon or period length (N or T)	No. of scheduling sequences
1	6
2	36
3	216
4	1296

The amount of scheduling sequences that have to be compared at every time step in the online computations, which is given by (16), is shown in Table I.

When taking the computational complexity into account, Algorithm 1 using a finite horizon LQ cost function where the final state weighting is chosen using a periodic approach to an infinite horizon cost function, as described in Section V, results in the best performance. The reason for this is that it can maintain a low cost, computed by (37), while only six possible scheduling sequences have to be compared.

With the chosen system and noise model, the set-to-zero strategy outperforms the hold-input strategy for all of the online algorithms.

C. Large-Scale Systems

In this simulation, we consider an NCS with nine actuators that compete for seven CAPs and one slot in the CFP.

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

TABLE II
STEADY-STATE EMPIRICAL COST AVERAGED OVER 1000 SIMULATIONS FOR A LARGE-SCALE SYSTEM WITH NINE ACTUATORS SIMULATING OVER A NETWORK FEATURING ONE GTS AND SEVEN SLOTS IN THE CAP

	Gaussian noise [dB]	Noise as in (38) [dB]	Number of scheduling seq.
RR $P = 9$	15.43	31.20	1
RR $P = 3$	14.99	30.57	1
Periodic offline $P = 2$ (Alg. 2)	15.77	30.92	1
Periodic offline $P = 3$ (Alg. 2)	15.43	30.67	1
Finite MPC $N = 1$, $W = Q$ (Alg. 1)	16.18	38.12	72
Finite MPC $N = 2$, $W = Q$ (Alg. 1)	14.72	29.46	5184
Periodic MPC $P = 2$ (Alg.3)	14.72	28.87	5184
Greedy MPC $P = 2$ (Alg. 4)	14.73	-	1444
Finite MPC $N = 1$, $W = P(\vec{S}^*_W)$ (Alg. 1)	14.70	29.19	72
Finite MPC $N = 2$, $W = P(\vec{S}^*_W)$ (Alg. 1)	14.67	28.90	5184

The system is created as $\tilde{A} = \text{diag}\{A, A, A\}$, where A is given in (35), and $\tilde{B} = \text{diag}\{B, B, B\}$ using B from (36). The $\tilde{C}$, $\tilde{Q}$, and $\tilde{R}$ matrices are created in a similar manner. The optimization horizon length $N \in \{1, 2\}$ in Algorithm 1 and period length $T = 2$ in Algorithms 2–4. Algorithm 2 is, however, also simulated with $T = 3$. When the final state weighting is computed as in Section V-D, the period length $T = 2$. For the RR algorithms, we used two schedules. The first one gives every actuator a fair share of the network, and thus every nine time steps, each actuator is addressed once in the CFP and seven times in the CAP and is not addressed once. This gives a periodic schedule with length $T = 9$. The other RR schedule gives priority to the actuators that address the subsystems with the largest eigenvalues. This system has three eigenvalues at 1.5, of which each actuator gets a slot in the CFP every third time step, while the lowest eigenvalues at 0.82 will only be addressed in the CAP twice every three time steps. The subsystems with eigenvalues at 1 are always addressed in the CAP. The resulting scheduling sequence has length $T = 3$. The remaining parameters are as described in Section V-D.

The results of the simulations using set-to-zero at the actuators are shown in Table II. Since there is only one actuator at every time step that is not addressed, the performance deviation among the algorithms is small. It is, however, notable that all online algorithms besides Algorithm 1 with $W = Q$ and $N = 1$ outperform the offline algorithms. In addition, Algorithm 1 with W selected by (34) outperforms the other algorithms by a small margin. When the system is affected by noise as in (38), the margin between the online and offline algorithms increases. In addition, the RR algorithm's performance deviates depending on which schedule is selected. Here, it shows that giving priority to the subsystems with the largest open loop eigenvalues ($T = 3$) results in a better performance than granting every actuator a fair share of the network ($T = 9$). The performance of Algorithm 2, the offline algorithm, is comparable with RR with $P = 9$. The reason for RR with $T = 3$ outperforming Algorithm 2 is that Algorithm 2 computes the schedule based on the first state estimate. In addition, it is in this case fairly easy to create an RR schedule, since all but one actuator can be addressed at every time step.

Finally, it should also be noted that Algorithm 1 with $N = 1$ and $W = P(\vec{S}^*_W)$ results in a good performance, while only having to compare 72 scheduling sequences at every time step with 5184 for the other online algorithms.

VII. CONCLUSION

We have discussed the possibilities of control over networks based on the IEEE 802.15.4 protocol where we utilized both the reliable CFP and the shared CAP. We have shown through simulations that the choice of the scheduling sequence is crucial for the performance of the system. The simulations further show that a finite horizon cost function with a choice of final state weighting, where both the CAP and CFP are taken into account in both the cost function and the final state weighting, results in very good performance, while the amount of scheduling sequences that need to be evaluated remains low.

The computational complexity of the algorithms is only briefly discussed here. Future work could include a tradeoff between the performance and computational cost. It could be of further interest to reduce the amount of possible scheduling sequences in the proposed algorithms, such that the computational cost of the combinatorial part of the optimization can be reduced significantly. This will open up for possibilities of using the algorithms on large-scale NCSs as well as reduce the power consumption, such that the algorithms can be used on battery-powered devices. It could also be of interest to study time delay effects in practice in relation to the arrival times of the different packets in one superframe.

APPENDIX A
DERIVATION OF (17)

Given (15) for a horizon of length N, by applying the dynamic programming algorithm [56], we can state that at stage N, the cost for $x(N)$ and schedule $S(N)$ is given by $J_N(x(N), S(N), \mu(x(N), S(N))) = J^*_N(x(N), S(N)) = \alpha^N x^T(N) Q x(N)$. The cost at stage k for a given scheduling sequence $\vec{S}(k)$ is then given by

$$\begin{aligned} &J_k(x(k), \vec{S}(k), \pi(\vec{S}(k))) \\ &= \min_{u(k)} \mathbf{E}\left\{\|x(k)\|^2_Q + \|u(k)\|^2_R + \alpha J^*_{k+1}(Ax(k) \right. \\ &\qquad \left. + B(\gamma(k), k)u(k) + \omega(k), \vec{S}(k+1)) | x(k)\vec{S}(k)\right\}. \end{aligned}$$

Using the fact that $\omega(k)$ is zero mean, this can be rewritten as

$$\begin{aligned}
&J_k(x(k), \vec{S}(k), \pi(\vec{S}(k)))\\
&= \min_{u(k)} \mathbf{E}\Big\{ x^T(k)(Q + \alpha A^T P(k+1)A)x(k)\\
&\qquad + u^T(k)(R + \alpha B^T(\gamma(k), k)P(k+1)B(\gamma(k), k))u(k)\\
&\qquad + 2\alpha u^T(k)B^T(\gamma(k), k)P(k+1)Ax(k)\\
&\qquad + \sum_{\ell=0}^{N-k-1} \alpha^{\ell+1}\omega(k+\ell)^T P(k+\ell+1)\\
&\qquad \times \omega(k+\ell)|x(k), \vec{S}(k)\Big\}
\end{aligned}$$

where we use the fact that $\omega(k) \sim \mathcal{N}(\mathbf{0}_m, \Sigma_\omega)$ and $P(\vec{S}(k+N)) = W$. The above can be rewritten to

$$\begin{aligned}
&J_k(x(k), \vec{S}(k), \pi(\vec{S}(k)))\\
&= \min_{u(k)} x^T(k)(Q + \alpha A^T P(\vec{S}(k+1))A)x(k) + u^T(k)\\
&\quad \times (R + \alpha\, \mathbf{E}\{B^T(\gamma(k), k)P(\vec{S}(k+1))B(\gamma(k), k)|S(k)\})u(k)\\
&\quad + 2\alpha u^T(k)\,\mathbf{E}\{B^T(\gamma(k), k)|S(k)\}P(\vec{S}(k+1))Ax(k)\\
&\quad + \sum_{\ell=1}^{N-k} \alpha^\ell \,\mathrm{trace}(\Sigma_\omega P(\vec{S}(k+\ell))). \qquad (39)
\end{aligned}$$

Taking the derivative with respect to $u(k)$ and setting the resulting expression equal to zero result in the minimizer

$$u^*(k) = -L(\vec{S}(k))x(k)$$

where

$$\begin{aligned}
L(\vec{S}(k)) = \alpha(R + \alpha\, &\mathbf{E}\{B^T(\gamma(k), k)P(\vec{S}(k+1))\\
&\times B(\gamma(k), k)|S(k)\})^{-1}\\
&\times \mathbf{E}\{B^T(\gamma(k), k)|S(k)\}P(\vec{S}(k+1))A
\end{aligned}$$

and

$$\begin{aligned}
P(\vec{S}(k)) = Q + \alpha A^T &P(\vec{S}(k+1))A\\
&- \alpha\, \mathbf{E}\{A^T P(\vec{S}(k+1))B(\gamma(k), k)\}L(\vec{S}(k)).
\end{aligned}$$

By taking the expectations, the above can be rewritten as

$$\begin{aligned}
P(\vec{S}(k)) = Q &+ \alpha A^T P(\vec{S}(k+1))A\\
&- \alpha A^T P(\vec{S}(k+1))\\
&\times \left(\sum_i \mathbf{Pr}\{\gamma(k) = \gamma^{(i)}|S(k)\}B(\gamma^{(i)}, k)\right)\\
&\times L(\vec{S}(k)) \qquad (40)
\end{aligned}$$

where

$$\begin{aligned}
L(\vec{S}(k)) = \alpha\Bigg(R + \alpha\Bigg(&\sum_i \mathbf{Pr}\{\gamma(k) = \gamma^{(i)}|S(k)\}\\
&\times B^T(\gamma^{(i)}, k)P(\vec{S}(k+1))B(\gamma^{(i)}, k)\Bigg)\Bigg)^{-1}\\
&\times \left(\sum_i \mathbf{Pr}\{\{\}\gamma(k) = \gamma^{(i)}|S(k)\} \times B^T(\gamma^{(i)}, k)\right)\\
&\times P(\vec{S}(k+1))A \qquad (41)
\end{aligned}$$

where $P(\vec{S}(k+N)) = W$ and $L(\vec{S}(k+N)) = 0$.

The result in Theorem 1 follows by letting $\alpha = 1$ and inserting the above into (39), which results in the optimal cost (17) where $P(\vec{S}(k))$, given by (18), follows from rewriting the sum terms in (40) and (41). The optimal feedback gain (19) at stage k is then given by (41).

The result in Theorem 2 follows by letting $N \to \infty$ where $P_{\mathrm{per}}(\vec{S}(i))$ is given by (27), which is obtained using a periodic scheduling sequence, as defined in Section V, and rewriting the sum terms in (40) and (41). The optimal control gains (28) are also given by (41), where $P_{\mathrm{per}}(\vec{S}(i+1))$ is given by the positive definite solution to (27).

Appendix B
Time Invariant Formulation of the Periodic Riccati Equation

The periodic Riccati equation (27) can be formulated into a time-invariant Riccati equation. This can be done using a cyclic system description [58], where

$$\hat{B} = \begin{bmatrix} \mathbf{0}^T & B(\gamma(T-1), T-1) \\ \tilde{B} & \mathbf{0} \end{bmatrix}$$

where $\mathbf{0}$ is a $(T-1)m \times n$ matrix containing only zeros and

$$\tilde{B} = \mathrm{diag}\{B(\gamma(0), 0), B(\gamma(1), 1), \ldots, B(\gamma(T-2), T-2)\}$$

and $\hat{\gamma} = \{\gamma(0), \gamma(1), \ldots, \gamma(T-1)\}$. The cyclic A matrix is given by

$$\hat{A} = \begin{bmatrix} 0 & 0 & \ldots & 0 & A \\ A & 0 & \ldots & 0 & 0 \\ 0 & A & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & A & 0 \end{bmatrix}.$$

We define $\hat{Q} = \mathrm{diag}\{Q, Q, \ldots, Q\} \in \mathbb{R}^{mT \times mT}$ and $\hat{R} = \mathrm{diag}\{R, R, \ldots, R\} \in \mathbb{R}^{nT \times nT}$. This then results in the time-invariant Riccati equation

$$\hat{P} = \hat{Q} + \alpha \hat{A}^T \hat{P} \hat{A} - \alpha \hat{A}^T \hat{P}\, \mathbf{E}\{\hat{B}(\hat{\gamma})|\vec{S}\}\hat{L}$$

and

$$\begin{aligned}
\hat{L} = \alpha(\hat{R} + \alpha\, &\mathbf{E}\{\hat{B}^T(\hat{\gamma})\hat{P}\hat{B}(\hat{\gamma})|\vec{S}\})^{-1}\\
&\times \mathbf{E}\{\hat{B}^T(\hat{\gamma})|\vec{S}\}\hat{P}\hat{A}.
\end{aligned}$$

After calculating the expectations as in Appendix A, the Riccati equation can be written as

$$\hat{P} = \hat{Q} + \alpha \hat{A}^T \hat{P} \hat{A} - \alpha \hat{A}^T \hat{P} \hat{B} \hat{\Gamma} \hat{L}$$

and

$$\hat{L} = \alpha(\hat{R} + \alpha \hat{\Gamma} \hat{B}^T \hat{P} \hat{B})^{-1} \hat{\Gamma} \hat{B}^T \hat{P} \hat{A}$$

where

$$\hat{\Gamma} = \mathrm{diag}\{\mathrm{diag}\{S(0)\}, \mathrm{diag}\{S(1)\}, \ldots, \mathrm{diag}\{S(T-1)\}\}$$

and $\hat{B} \in \mathbb{R}^{mT \times nT}$ is the cyclic B matrix. Here, $P_{\mathrm{per}}(i)$ can be extracted as the ith $m \times m$ block in $\hat{P}$.

REFERENCES

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

[2] T. Namerikawa and T. Kato, "Distributed load frequency control of electrical power networks via iterative gradient methods," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf. (CDC-ECC)*, 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 *Proc. IEEE 51st Annu. Conf. Decision Control (CDC)*, Dec. 2012, pp. 2459–2464.

[4] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Ind. Electron. Mag.*, vol. 5, no. 4, pp. 23–34, Dec. 2011.

[5] *ZigBee PRO Specification*, ZigBee Alliance, Amsterdam, The Netherlands, 2007.

[6] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4-2011, 2011.

[7] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012, 2012.

[8] B.-C. Wang and M. Fu, "Comparison of periodic and event based sampling for linear state estimation," in *Proc. 19th IFAC World Congr. Int. Fed. Automat. Control*, 2014, pp. 5508–5513.

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

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

[11] L. Shi, Y. Yuan, and J. Chen, "Finite horizon LQR control with limited controller-system communication," *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1835–1841, Jul. 2013.

[12] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 847–861, Apr. 2013.

[13] D. Antunes, W. P. M. H. Heemels, and P. Tabuada, "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design," in *Proc. IEEE 51st Annu. CDC*, Dec. 2012, pp. 7212–7217.

[14] X. Meng and T. Chen, "Event-driven communication for sampled-data control systems," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2013, pp. 3002–3007.

[15] M. Mazo, Jr., and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proc. 47th IEEE Conf. Decision Control (CDC)*, Dec. 2008, pp. 435–440.

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

[17] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st Annu. Conf. Decision Control (CDC)*, Dec. 2012, pp. 3270–3285.

[18] J. Araujo, M. Mazo, Jr., A. Anta, P. Tabuada, and K. H. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 175–184, Feb. 2014.

[19] D. Antunes and W. P. M. H. Heemels, "Rollout event-triggered control: Beyond periodic control performance," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3296–3311, Dec. 2014.

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

[21] H. Rehbinder and M. Sanfridson, "Integration of off-line scheduling and optimal control," in *Proc. 12th Euromicro Conf. Real-Time Syst. (Euromicro RTS)*, Jun. 2000, pp. 137–143.

[22] D. Hristu and K. Morgansen, "Limited communication control," *Syst. Control Lett.*, vol. 37, no. 4, pp. 193–205, Jul. 1999.

[23] D. Hristu-Varsakelis and L. Zhang, "LQG control of networked control systems with access constraints and delays," *Int. J. Control*, vol. 81, no. 8, pp. 1266–1280, 2008.

[24] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953–958, Jun. 2006.

[25] F. Xia, A. Vinel, R. Gao, L. Wang, and T. Qiu, "Evaluating IEEE 802.15.4 for cyber-physical systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, 2011, Art. ID 596397.

[26] D. E. Quevedo, V. Gupta, W.-J. Ma, and S. Yüksel, "Stochastic stability of event-triggered anytime control," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3373–3379, Dec. 2014.

[27] D. Antunes, J. P. Hespanha, and C. Silvestre, "Control of impulsive renewal systems: Application to direct design in networked control," in *Proc. 48th IEEE Conf. Decision Control, Held Jointly 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 6882–6887.

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

[29] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1314–1319, Jul. 2007.

[30] 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, Dec. 2011.

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

[32] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 586–601, Mar. 2011.

[33] A. Ulusoy, O. Gurbuz, and A. Onat, "Wireless model-based predictive networked control system over cooperative wireless network," *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 41–51, Feb. 2011.

[34] R. Postoyan, N. van de Wouw, D. Nesic, and W. P. M. H. Heemels, "Tracking control for nonlinear networked control systems," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1539–1554, Jun. 2014.

[35] 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," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2167–2181, Nov. 2015.

[36] E. Henriksson, D. E. Quevedo, H. Sandberg, and K. H. Johansson, "Self-triggered model predictive control for network scheduling and control," in *Proc. 8th IFAC Symp. Adv. Control Chem. Process.*, 2012, pp. 1–7.

[37] 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, Mar. 2014.

[38] L. Orihuela, A. Barreiro, F. Gómez-Estern, and F. R. Rubio, "Periodicity of Kalman-based scheduled filters," *Automatica*, vol. 50, no. 10, pp. 2672–2676, Oct. 2014.

[39] A. Saifullah *et al.*, "Near optimal rate selection for wireless control systems," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 4s, pp. 128:1–128:25, Jul. 2014.

[40] E. G. W. Peters, D. E. Quevedo, and M. Fu, "Co-design for control and scheduling over wireless industrial control networks," in *Proc. 54th IEEE Conf. Decision Control (CDC)*, Dec. 2015, pp. 1–6.

[41] A. Koubâa, M. Alves, and E. Tovar, "IEEE 802.15.4 for wireless sensor networks: A technical overview," CISTER-Res. Center Real-Time Embedded Comput. Syst., Porto, Portugal, Tech. Rep., Jul. 2005, Report No. TR-050702. [Online]. Available: http://www.cister.isep.ipp.pt/docs/ieee_802_15_4_for_wireless_sensor_networks__a_technical_overview/222/

[42] P. Park, C. Fischione, and K. H. Johansson, "Performance analysis of GTS allocation in beacon enabled IEEE 802.15.4," in *Proc. 6th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2009, pp. 1–9.

[43] T. R. Park, T. H. Kim, J. Y. Choi, S. Choi, and W. H. Kwon, "Throughput and energy consumption analysis of IEEE 802.15.4 slotted CSMA/CA," *Electron. Lett.*, vol. 41, no. 18, pp. 1017–1019, Sep. 2005.

[44] P. Park, C. Fischione, and K. H. Johansson, "Modeling and stability analysis of hybrid multiple access in the IEEE 802.15.4 protocol," *ACM Trans. Sensor Netw.*, vol. 9, no. 2, pp. 13:1–13:55, Mar. 2013.

[45] M. Martalò, S. Busanelli, and G. Ferrari, "Markov chain-based performance analysis of multihop IEEE 802.15.4 wireless networks," *Perform. Eval.*, vol. 66, no. 12, pp. 722–741, Dec. 2009.

[46] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. S. Sastry, "Optimal control with unreliable communication: The TCP case," in *Proc. Amer. Control Conf.*, vol. 5. Jun. 2005, pp. 3354–3359.

[47] C. Ramesh, "State-based channel access for a network of control systems," Ph.D. dissertation, School Elect. Eng., KTH Roy. Inst. Technol., Stockholm, Sweden, 2014.

[48] D. E. Quevedo, A. Ahlen, and K. H. Johansson, "State estimation over sensor networks with correlated wireless fading channels," *IEEE Trans. Autom. Control*, vol. 58, no. 3, pp. 581–593, Mar. 2013.
[49] D. E. Quevedo, J. Ostergaard, and A. Ahlen, "Power control and coding formulation for state estimation with wireless sensors," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 413–427, Mar. 2014.
[50] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.
[51] L. Schenato, "To zero or to hold control inputs with lossy links?" *IEEE Trans. Autom. Control*, vol. 54, no. 5, pp. 1093–1099, May 2009.
[52] D. E. Quevedo and D. Nešić, "Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 370–375, Feb. 2011.
[53] G. C. Goodwin, D. E. Quevedo, and E. I. Silva, "Architectures and coder design for networked control systems," *Automatica*, vol. 44, no. 1, pp. 248–257, Jan. 2008.
[54] G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh, "A moving horizon approach to networked control system design," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1427–1445, Sep. 2004.
[55] M. Athans, R. Ku, and S. Gershwin, "The uncertainty threshold principle: Some fundamental limitations of optimal decision making under dynamic uncertainty," *IEEE Trans. Autom. Control*, vol. 22, no. 3, pp. 491–495, Jun. 1977.
[56] D. P. Bertsekas, *Dynamic Programming and Optimal Control* (Athena Scientific Optimization and Computation), vol. 1. Belmont, MA, USA: Athena Scientific, 2005.
[57] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
[58] S. Bittanti and P. Colaneri, *Periodic Systems—Filtering and Control*. Springer-Verlag, London: 2009.
[59] K. V. Ling, J. Maciejowski, A. Richards, and B. F. Wu, "Multiplexed model predictive control," *Automatica*, vol. 48, no. 2, pp. 396–401, Feb. 2012.
[60] S. Bittanti, P. Colaneri, and G. De Nicolao, "The difference periodic Ricati equation for the periodic prediction problem," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 706–712, Aug. 1988.
[61] G. De Nicolao, "Cyclomonotonicity, Riccati equations and periodic receding horizon control," *Automatica*, vol. 30, no. 9, pp. 1375–1388, Sep. 1994.
[62] P. Benner, "Accelerating Newton's method for discrete-time algebraic Riccati equations," in *Proc. MTNS*, 1998, pp. 569–572.

Edwin G. W. Peters (S'15) received the B.Sc. degree in electrical engineering and the M.Sc. degree in signal processing from Aalborg University, Aalborg, Denmark, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree in control systems with The University of Newcastle, Callaghan, NSW, Australia.

His current research interests include networked control and signal processing.

Daniel E. Quevedo (S'97–M'05–SM'14) received the Ingeniero Civil Electrónico and M.Sc. degrees from Universidad Técnica Federico Santa María, Valparaíso, Chile, in 2000, and the Ph.D. degree from The University of Newcastle, Callaghan, NSW, Australia, in 2005.

He holds the Chair in Automatic Control with Paderborn University, Paderborn, Germany. His current research interests include automatic control, signal processing, and power electronics.

Dr. Quevedo was supported by a Full Scholarship from the Alumni Association, during his time with Universidad Técnica Federico Santa María, and received several university-wide prizes upon graduating. He received the IEEE Conference on Decision and Control Best Student Paper Award in 2003, and was a Finalist in 2002. He also received a five-year Research Fellowship from the Australian Research Council in 2009. He is an Editor of the *International Journal of Robust and Nonlinear Control*, and serves as the Chair of the IEEE Control Systems Society Technical Committee on Networks & Communication Systems.

Minyue Fu (F'03) received the B.Sc. degree from the University of Science and Technology of China, Hefei, China, in 1982, and the M.S. and Ph.D. degrees from the University of Wisconsin–Madison, Madison, WI, USA, in 1983 and 1987, respectively, all in electrical engineering.

He was an Assistant Professor with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, from 1987 to 1989. He joined the Department of Electrical and Computer Engineering, The University of Newcastle, Callaghan, NSW, Australia, in 1989, where he is currently a Chair Professor of Electrical Engineering. He has been a Visiting Associate Professor with the University of Iowa, Iowa City, IA, USA, Nanyang Technological University, Singapore, and Tokyo University, Tokyo, Japan. He has held a Changjiang Visiting Professorship with Shandong University, Jinan, China, a Qian-Ren Professorship with Zhejiang University, Hangzhou, China, and a Qian-Ren Professorship with the Guangdong University of Technology, Guangzhou, China. His current research interests include control systems, signal processing, communications, networked control systems, smart electricity networks, and superprecision positioning control systems.

Prof. Fu has been an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the *Journal of Optimization and Engineering*.