



## Brief paper

Adaptive controller placement for wireless sensor–actuator networks with erasure channels<sup>☆</sup>Daniel E. Quevedo<sup>a,1</sup>, Karl H. Johansson<sup>b</sup>, Anders Ahlén<sup>c</sup>, Isabel Jurado<sup>d</sup><sup>a</sup> School of Electrical Engineering & Computer Science, The University of Newcastle, NSW 2308, Australia<sup>b</sup> ACCESS Linnaeus Centre, School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden<sup>c</sup> Department of Engineering Sciences, Signals and Systems, Uppsala University, Uppsala, Sweden<sup>d</sup> Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, Universidad de Sevilla, Spain

## ARTICLE INFO

## Article history:

Received 15 October 2012

Received in revised form

31 July 2013

Accepted 2 August 2013

Available online 18 September 2013

## Keywords:

Wireless sensor–actuator networks

Packet dropouts

## ABSTRACT

Wireless sensor–actuator networks offer flexibility for control design. One novel element which may arise in networks with multiple nodes is that the role of some nodes does not need to be fixed. In particular, there is no need to pre-allocate which nodes assume controller functions and which ones merely relay data. We present a flexible architecture for networked control using multiple nodes connected in series over analog erasure channels without acknowledgments. The control architecture proposed adapts to changes in network conditions, by allowing the role played by individual nodes to depend upon transmission outcomes. We adopt stochastic models for transmission outcomes and characterize the distribution of controller location and the covariance of system states. Simulation results illustrate that the proposed architecture has the potential to give better performance than limiting control calculations to be carried out at a fixed node.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

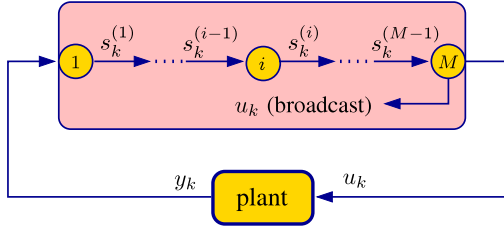
In a Networked Control System (NCS), sensor, controller and actuator links are not transparent, but are affected by bit-rate limitations, packet dropouts and/or delays. This leads to performance degradation and makes the design of NCSs often a challenging task (Antsaklis & Baillieul, 2007; Chen, Johansson, Olariu, Paschalidis, & Stokmenovic, 2011). An interesting aspect is that, when compared to traditional hard-wired control loops, wireless NCSs offer architectural flexibility and additional degrees of freedom. Whilst sensor and actuator functionalities will generally be pre-allocated, often there is no need to pre-assign in a static fashion which nodes carry out control calculations, and which nodes merely relay data. Intuitively, the roles of individual

nodes should depend on the information available at each time instant. In the present work, we examine this question for the case of NCSs with random packet dropouts. Our motivating application is real-time control in the process industry. Several new standards have recently been introduced for multi-hop wireless sensor and actuator networks, e.g., WirelessHART, ISA-100, and this paper proposes a new adaptive controller placement suitable to be implemented over these standards. Note that the plant time constants in process industry are often of the order of seconds or minutes (or even higher), so we make the reasonable assumption that network-induced delays can be neglected. As background to our current work, Goodwin, Quevedo, and Silva (2008) study performance of three static NCS architectures by adopting an additive signal-to-noise ratio constrained channel model. The results in Goodwin et al. (2008) suggest that, in the absence of coding, placing the controller at the actuator node will give better performance than placing it at the sensor node. The work (Robinson & Kumar, 2008) examines NCSs with stochastic packet dropouts using optimal control techniques. *Inter-alia*, the work shows that optimal performance can be achieved if all nodes aggregate their entire history of received data and relay it to the controller–actuator. Depending upon the information available at each node, various optimal control problems can be analyzed. More recently, Pajic, Sundaram, Pappas, and Mangharam (2011) have investigated a distributed control strategy wherein the network itself acts as a controller for a MIMO plant. All nodes (including the actuator nodes) perform linear combinations of internal state

<sup>☆</sup> Research supported by Australian Research Council's Discovery Project DP0988601, by VINNOVA project WiComPI, project Dnr2009-02963, by MCyT (Grant DPI2010-19154), and by the European Commission FeedNetBack Project (Grant 223866). The material in this paper was partially presented at the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12), September 14–15, 2012, Santa Barbara, California, USA. This paper was recommended for publication in revised form by Associate Editor Tongwen Chen under the direction of Editor Ian R. Petersen.

E-mail addresses: [dquevedo@ieee.org](mailto:dquevedo@ieee.org) (D.E. Quevedo), [kallej@ee.kth.se](mailto:kallej@ee.kth.se) (K.H. Johansson), [Anders.Ahlen@signal.uu.se](mailto:Anders.Ahlen@signal.uu.se) (A. Ahlén), [ijurado@cartuja.us.es](mailto:ijurado@cartuja.us.es) (I. Jurado).

<sup>1</sup> Tel.: +61 2 49217072; fax: +61 2 49601712.

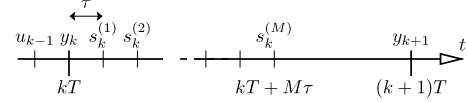


**Fig. 1.** Single-loop control over a wireless sensor–actuator network: forward- and feedback-links are unreliable.

variables of neighboring nodes. In the case of analog erasure channels with i.i.d. dropouts (without acknowledgments), in Pajic et al. (2011) the resulting NCS is then cast, analyzed and designed as a jump-linear system.

The present work studies a single-loop NCS topology which uses a series connection of analog erasure channels. We focus on situations where the wireless nodes have only limited energy and processing power, precluding long data packets. Further, the nodes do not provide local transmission acknowledgments. Feedback (or acknowledgment) is only provided by the actuator node, which broadcasts the applied plant input value over parallel unreliable links to the intermediate nodes. This strategy is plausible as the actuator node is powered, which is a reasonable assumption for most actuators in process industry. Due to random dropouts, the actuator node does not have full information on plant outputs. This constitutes one of the major difficulties when implementing a controller in such a NCS. We address this issue by using an estimation and control structure which is distributed across the network. Instead of tackling optimal control formulations (which depend upon network parameters and may therefore be difficult to implement in practice), we adopt a so-called emulation-based approach, where the controller has been pre-designed; see, e.g., Antunes, Hespanha, and Silvestre (2013), Donkers, Heemels, van de Wouw, and Hetel (2011), Nešić and Teel (2004). To be more specific, we assume that the control policy consists of a pre-designed state feedback-gain combined with a state observer, which, in the absence of network effects, would lead to the desired performance. Within this context, we present a flexible NCS architecture where the role played by individual nodes depends upon transmission outcomes. While all nodes calculate local state estimates at all times, with the algorithm proposed, transmission outcomes determine, at each instant, whether the control input will be calculated at the actuator node, at the sensor node or at one of the intermediate nodes. It turns out that, if individual dropout processes are i.i.d., then the controller location has a stationary distribution, which can be easily characterized. To analyze the performance of the dynamic NCS architecture in the presence of correlated dropouts, we derive a jump-linear system model and adopt the network model recently introduced in Quevedo, Ahlén, and Johansson (2013). This model encompasses temporal and spatial correlations of packet dropouts and is therefore of more practical importance than traditional i.i.d. models. The present paper goes beyond our recent conference contribution (Quevedo, Johansson, Ahlén, & Jurado, 2012), by presenting a closed loop model and considering networks with correlated dropouts. Our approach complements (Quevedo, Johansson et al., 2012) by focusing on a nominal linear estimator, as opposed to a time-varying Kalman filter. This opens the possibility to analyze NCS performance with correlated dropouts using techniques from jump-linear systems.

*Notation:* We write  $\mathbb{N}_0$  for  $\{0, 1, 2, \dots\}$ ;  $\mathbb{R}$  are the real numbers, whereas  $\mathbb{R}_{\geq 0} \triangleq [0, \infty)$ . The  $\ell$ -th unit row-vector in Euclidean space is denoted by  $e_\ell$ , for example,  $e_2 = [0 \ 1 \ 0 \ \dots \ 0]$ ;  $I_n$  is the  $n \times n$  unit matrix,  $0_n \triangleq 0 \cdot I_n$ ;  $\otimes$  refers to the Kronecker product. For any set of column vectors,  $\{u_1, \dots, u_n\}$ ,  $\text{col}(u_1, \dots, u_n) =$



**Fig. 2.** TDMA transmission schedule. The plant input  $u_k$  is applied at time  $kT + M\tau$  and broadcast at time  $(k + 1)T - \tau$ .

$[u_1^T, \dots, u_n^T]^T$ . We adopt the convention  $\sum_{j=1}^0 a_j = 0$ , for all  $a_0, a_1 \in \mathbb{R}$ . A real random variable  $\mu$ , which is zero-mean Gaussian with covariance  $\Gamma$  is denoted by  $\mu \sim \mathcal{N}(0, \Gamma)$ .

## 2. Wireless sensor–actuator network setup

We consider MIMO LTI plant models of the form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k, \quad k \in \mathbb{N}_0 \end{aligned} \quad (1)$$

where  $x_0 \sim \mathcal{N}(\bar{x}_0, P_0)$ ,  $P_0 \succ 0$ . In (1),  $u_k \in \mathbb{R}^m$  is the plant input,  $x_k \in \mathbb{R}^n$  is the state,  $y_k \in \mathbb{R}^p$  is the output, and  $w_k \sim \mathcal{N}(0, Q)$ .  $Q \succ 0$  and  $v_k \sim \mathcal{N}(0, R)$ ,  $R \succ 0$ , are driving noise and measurement noise, respectively. As foreshadowed in the introduction, we focus on a situation where suitable feedback and estimator gains  $L \in \mathbb{R}^{m \times n}$  and  $K \in \mathbb{R}^{n \times p}$  have been pre-designed for the case where the controller  $(K, L)$  has perfect access to plant outputs and inputs. Consequently, we assume that if the control inputs

$$u_k = L\hat{x}_k^{\text{nom}}, \quad k \in \mathbb{N}_0, \quad \text{with} \quad (2)$$

$$\hat{x}_k^{\text{nom}} = A\hat{x}_{k-1}^{\text{nom}} + Bu_{k-1} + K(y_k - C(A\hat{x}_{k-1}^{\text{nom}} + Bu_{k-1})), \quad (3)$$

where  $\hat{x}_k^{\text{nom}}$  denotes an estimate of the state  $x_k$ , were implemented at the plant, then satisfactory performance would be attained. The main theme of the present work is to investigate how to implement the above nominal controller, when using a wireless sensor–actuator network with dropouts.

The sensor node measures the plant output  $y_k$ , whereas the actuator node manipulates the plant input  $u_k$ . The loop is closed over a wireless network, characterized via a (directed) line-graph having  $M$  nodes; see Fig. 1. Transmissions are in sequential Round-Robin fashion  $\{1, 2, \dots, M, 1, 2, \dots\}$  as depicted in Fig. 2. More precisely, the packet  $s_k^{(i)}$  is transmitted from node  $i$  to node  $i + 1$  at times  $kT + i\tau$ , where  $T$  is the sampling period of (1) and  $\tau \ll T/(M + 1)$  refers to the times between transmissions of packets  $s_k^{(i)}$ . The input  $u_k$  is applied at time  $kT + (M + 1)\tau$ . We thus assume that in-network processing is much faster than the plant dynamics (1) and neglect delays introduced by the network.

The network introduces stochastic packet dropouts. To study the situation, we adopt an analog erasure channel model and introduce the binary success processes

$$\gamma_k^{(i)} \in \{0, 1\}, \quad k \in \mathbb{N}_0, \quad i \in \{1, 2, \dots, M - 1\},$$

where  $\gamma_k^{(i)} = 1$  indicates that transmission of the packet  $s_k^{(i)}$  from node  $i$  to node  $i + 1$  at time  $kT + i\tau$ , is successful, i.e., error-free;  $\gamma_k^{(i)} = 0$  refers to a packet-dropout. Throughout this work, we assume that transmission outcomes are known at the corresponding receiver sides and that the sensor node  $i = 1$  has direct access to plant output measurements. For notational convenience, we write  $\gamma_k^{(0)} = 1$ , for all  $k \in \mathbb{N}_0$ . To save energy, the wireless nodes  $i \in \{1, 2, \dots, M - 1\}$  do not provide acknowledgments of receipt of the packets. The actuator node  $M$  provides a feedback mechanism: At time  $(k + 1)T - \tau$ , it broadcasts the control value  $u_k$  to nodes  $i \in \{1, \dots, M - 1\}$ ; see Fig. 1. Due to channel fading, the feedback links between actuator and sensors are also affected by dropouts. We denote the associated success processes via

$$\delta_k^{(i)} \in \{0, 1\}, \quad k \in \mathbb{N}_0, \quad i \in \{1, 2, \dots, M - 1\}.$$

More precisely, if  $u_k$  is successfully received at node  $i$ , then we set  $\delta_k^{(i)} = 1$ ; see also Imer, Yüksel, and Başar (2006) for studies on the importance of acknowledgments in closed loop control. We assume that the actuator node has perfect knowledge of plant inputs, and thus, write  $\delta_k^{(M)} = 1, \forall k \in \mathbb{N}_0$ . Since the actuator node will, in general, have less stringent energy constraints than the other nodes, we focus our attention on situations where the feedback links are more reliable than the forward links moving data from the sensor to the actuator.

Due to packet dropouts, plant output measurements are not always available at the actuator node. On the other hand, the sensor node will, in general, not have perfect information of previous plant inputs. This makes the implementation of the nominal controller (2)–(3) a challenging task. In the sequel, we will present an adaptive controller placement algorithm where the computations leading to the plant inputs are distributed across the network. We foresee that our approach will lead to a dynamic assignment of the role played by the individual network nodes. As will become apparent in the sequel, which tasks are carried out by individual nodes at each time instant, will depend upon transmission outcomes.

### 3. Flexible controller placement

To keep communication overheads low, the packets transmitted by each node  $i$  have only two fields, namely, output measurements and tentative plant inputs (if available):

$$s_k^{(i)} = (y_k, u_k^{(i)}). \quad (4)$$

Plant outputs are transmitted in order to pass on information on the plant state to the nodes  $\{i + 1, i + 2, \dots, M\}$ ; see Fig. 1. On the other hand,  $u_k^{(i)}$  in (4) is the plant input which is applied at the plant provided the packet  $s_k^{(i)}$  is delivered at the actuator node. If  $s_k^{(i)}$  is lost, then following the algorithm described below, the plant input will be provided by one of the later nodes  $\ell > i$ , which thereby takes on the controller role at time  $k$ . In the sequel, we will refer to the node which calculates the plant input at time  $k$  as  $c_k \in \{1, 2, \dots, M\}$ :

$$u_k = L\hat{x}_k^{(c_k)}, \quad k \in \mathbb{N}_0, \quad (5)$$

where  $\hat{x}_k^{(c_k)}$  is the local plant state estimate computed at node  $c_k$ . Intuitively, good control performance will be achieved if the estimate used in (5) is accurate. Clearly, due to the multi-hop nature of the network, nodes which are closer to the sensor will have access to more output measurements; see Fig. 1. On the other hand, one can expect that nodes which are physically located closer to the actuator node will on average receive more plant input acknowledgments, thus, have better knowledge of plant inputs.

While only the node  $c_k$  will provide  $u_k$ , in our formulation all nodes compute local state estimates,  $\hat{x}_k^{(i)}$ , by using the data received from the actuator node and the preceding node. This serves as safeguard for instances when the loop is broken due to dropouts. Motivated by the fact that often feedback links from the actuator to the intermediate sensors are “quite reliable”, we adopt the following simple procedure: If plant output measurements are available at node  $i$ , then state estimators are of the form (3); at instances when the plant input is not available, an open loop estimate is used, thus:

$$\hat{x}_k^{(i)} = A\hat{x}_{k-1}^{(i)} + B\hat{u}_{k-1}^{(i)} + K_k^{(i)}(y_k - C(A\hat{x}_{k-1}^{(i)} + B\hat{u}_{k-1}^{(i)})), \quad (6)$$

where

$$K_k^{(i)} \triangleq \Gamma_k^{(i)}K, \quad \text{and} \quad \Gamma_k^{(i)} \triangleq \prod_{j \in \{0, 1, \dots, i-1\}} \gamma_k^{(j)} \quad (7)$$

is equal to 1 if and only if  $y_k$  is available at node  $i$  at time  $kT + (i - 1)\tau$ . In (6),  $\hat{u}_{k-1}^{(i)}$  is a local plant input estimate. In particular, if  $\delta_{k-1}^{(i)} = 1$ , then  $\hat{u}_{k-1}^{(i)} = u_{k-1}$ . At instances where  $\delta_{k-1}^{(i)} = 0$ , node  $i$  uses  $u_{k-1}^{(i)}$ , the tentative plant input value transmitted in the second field of the previous packet  $s_{k-1}^{(i)}$  (if non-empty), or otherwise sets  $\hat{u}_{k-1}^{(i)} = L\hat{x}_{k-1}^{(i)}$ . More details on the estimator are given in Section 5.

**Remark 1.** Of course, the above transmission and control strategy will in general not be optimal. In particular, nodes do not transmit local state estimates and the control law does not depend upon network parameters, e.g., dropout probabilities; cf., Chiuso and Schenato (2011). The aim of the present work is to develop a simple and practical method, which uses an existing control and estimation policy for implementation over an unreliable network and only requires little communication.  $\square$

**Remark 2.** In Quevedo, Johansson et al. (2012), instead of using (7), the gains  $K_k^{(i)}$  were taken as the Kalman filter gains for a system with intermittent observations; see, e.g., Huang and Dey (2007), Quevedo, Ahlén, Leong, and Dey (2012); Quevedo, Ahlén, and Østergaard (2010). Our subsequent analysis up to (22) can be applied to this structure as well. However, the jump-linear model derived in Section 5 requires a jump-linear estimation model, such as (7).  $\square$

Algorithm 1, run at every node  $i \in \{1, 2, \dots, M\}$ , embodies the adaptive controller allocation method described above. Which calculations are carried out at each node, depends upon transmission outcomes involving the current node (see lines 6, 8, 14, 24 and 37) and also transmission outcomes at previous nodes (see lines 16, 19 and 24). In particular, node  $i$  only calculates a tentative plant input when no tentative plant input is received from node  $i - 1$  and node  $i$  has successfully received  $u_{k-1}$  (see lines 9 and 25). Therefore, preference is given to relay incoming tentative plant input values. The reason for adopting this decision procedure lies in that we assume that data sent from the actuator node to intermediate nodes is often available, whereas transmissions of packets  $s_k^{(i)}$  are less reliable. Thus, nodes closer to the sensor node can be expected to have better state estimates than nodes located further down the line. In particular, the sensor node  $i = 1$  uses as input

$$s_k^{(0)} = (y_k, \emptyset), \quad \gamma_k^{(0)} = 1. \quad (8)$$

If  $\delta_{k-1}^{(1)} = 1$ , then the sensor node calculates a tentative control value and transmits  $s_k^{(1)} = (y_k, L\hat{x}_k^{(1)})$  to node 2. Subsequent nodes relay this packet towards the actuator node. If the packet is dropped along the way, then the next node  $i$  where  $\delta_{k-1}^{(i)} = 1$ , calculates a tentative control  $u_k^{(i)} = L\hat{x}_k^{(i)}$  and transmits  $s_k^{(i)} = (\emptyset, u_k^{(i)})$  to node  $i + 1$ , etc. On the other hand, if  $\delta_{k-1}^{(1)} = 0$ , then  $s_k^{(0)}$  is relayed to subsequent nodes until arriving at some node  $i$  where  $u_{k-1}$  was successfully received. Control calculations are then carried out and the packet  $s_k^{(i)}$  obtained is relayed towards the actuator node, etc. The actuator node implements  $u_k = u_k^{(M)}$ , the value contained in the second field of  $s_k^{(M)}$ .

**Remark 3.** An advantage of allowing the control calculations to be located arbitrarily and in a time-varying fashion, is that it makes more difficult for someone to attack the NCS. The development of secure control strategies based on Algorithm 1 presented remains a topic of future research.  $\square$

**Algorithm 1** Adaptive Controller Placement

---

```

1:  $k \leftarrow 0, \hat{x}_0^{(i)} \leftarrow 0, j \leftarrow 0$ 
2: while  $t \geq 0$  do ▷  $t \in \mathbb{R}_{\geq 0}$  is actual time
3:   while  $t \leq kT + (i-1)\tau$  do ▷ wait-loop
4:      $j \leftarrow j + 1$ 
5:   end while
6:   if  $\gamma_k^{(i-1)} = 0$  then ▷  $s_k^{(i-1)}$  is dropped
7:      $\hat{u}_k^{(i)} \leftarrow L\hat{x}_k^{(i)}$ 
8:     if  $\delta_{k-1}^{(i)} = 1$  then
9:        $s_k^{(i)} \leftarrow (\emptyset, \hat{u}_k^{(i)})$  ▷ a tentative input
10:    else
11:       $s_k^{(i)} \leftarrow (\emptyset, \emptyset)$ 
12:    end if
13:  end if
14:  if  $\gamma_k^{(i-1)} = 1$  then ▷  $s_k^{(i-1)}$  is received
15:     $(y, u) \leftarrow s_k^{(i-1)}$ 
16:    if  $y \neq \emptyset$  then ▷  $y_k$  is available
17:       $\hat{x}_k^{(i)} \leftarrow \hat{x}_k^{(i)} + K(y - C\hat{x}_k^{(i)})$ 
18:    end if
19:    if  $u \neq \emptyset$  then
20:       $u_k^{(i)} = u$ 
21:    else
22:       $u_k^{(i)} \leftarrow L\hat{x}_k^{(i)}$ 
23:    end if
24:    if  $u = \emptyset \wedge \delta_{k-1}^{(i)} = 1$  then
25:       $s_k^{(i)} \leftarrow (y, u_k^{(i)})$  ▷ a tentative input
26:    else
27:       $s_k^{(i)} \leftarrow (y, u)$  ▷  $s_k^{(i-1)}$  is forwarded
28:    end if
29:  end if
30:  while  $t < kT + i\tau$  do ▷ wait-loop
31:     $j \leftarrow j + 1$ 
32:  end while
33:  transmit  $s_k^{(i)}$ 
34:  while  $t \leq (k+1)T - \tau$  do ▷ wait-loop
35:     $j \leftarrow j + 1$ 
36:  end while
37:  if  $\delta_k^{(i)} = 1$  then ▷  $u_k$  is available
38:     $\hat{x}_{k+1}^{(i)} \leftarrow A\hat{x}_k^{(i)} + Bu_k$ 
39:  else ▷ the value in  $s_k^{(i)}$  is used
40:     $\hat{x}_{k+1}^{(i)} \leftarrow A\hat{x}_k^{(i)} + Bu_k^{(i)}$ 
41:  end if
42:   $k \leftarrow k + 1$ 
43: end while

```

---

**4. Dynamic controller location**

With Algorithm 1, which of the nodes calculates the plant input  $u_k$ , depends upon the transmission outcomes. For further reference, we shall denote the set of nodes which calculate a tentative control input (see lines 9 and 25 of Algorithm 1) via  $\mathcal{C}_k \subset \{1, 2, \dots, M\}$ . It is convenient to introduce the processes  $\{\mu_k^{(i)}\}$ , and  $\{c_k^{(i)}\}$ , where  $i \in \{0, 1, \dots, M\}$  and

$$\mu_k^{(i)} \triangleq \begin{cases} 0 & \text{if the second field of } s_k^{(i)} \text{ is empty,} \\ 1 & \text{otherwise,} \end{cases} \quad (9)$$

$$c_k^{(i)} \triangleq \mu_k^{(i)} \max(\mathcal{C}_k \cap \{1, 2, \dots, i\}).$$

Note that  $\mu_k^{(1)} = \delta_{k-1}^{(1)}, \forall k \in \mathbb{N}$ . If  $c_k^{(i)} > 0$ , then  $c_k^{(i)}$  denotes the node where the second field of  $s_k^{(i)}$  was calculated. It is easy to see that, with the algorithm proposed and since the packets  $s_k^{(i)}$  are

communicated sequentially, see Fig. 2, we have  $c_k^{(1)} = \delta_{k-1}^{(1)}$ , for all  $k \in \mathbb{N}_0$ , whereas

$$c_k^{(i)} = \begin{cases} i\delta_{k-1}^{(i-1)} & \text{if } c_k^{(i-1)} = 0 \vee \gamma_k^{(i-1)} = 0, \\ c_k^{(i-1)} & \text{if } c_k^{(i-1)} > 0 \wedge \gamma_k^{(i-1)} = 1, \end{cases} \quad (10)$$

for  $i \in \{2, \dots, M\}$ ,  $k \in \mathbb{N}_0$ . The ‘‘controller node at time  $k$ ’’, i.e., the node where  $u_k$  was calculated is given by

$$c_k \triangleq c_k^{(M)} = \max(\mathcal{C}_k), \quad \forall k \in \mathbb{N}_0; \quad (11)$$

see (5). To derive our results, we introduce the aggregated transmission outcome process  $\{\beta_k\}$ ,  $k \in \mathbb{N}_0$ , where

$$\beta_k \triangleq \sum_{i=1}^{M-1} (2^{M-1} \gamma_{k+1}^{(i)} + \delta_k^{(i)}) 2^{i-1}, \quad k \in \mathbb{N}_0. \quad (12)$$

Note that  $\beta_{k-1} \in \mathbb{I} \triangleq \{0, 1, \dots, 2^{2M-2} - 1\}$  collects the outcomes of all transmissions which occur during the time-interval  $[kT - \tau, kT + M\tau]$ ; see Fig. 2. Thus,  $\beta_{k-1}$  determines  $\mathcal{C}_k$  and  $c_k$ , i.e., the controller location will dynamically adapt to the network conditions. To further elucidate the situation, in the sequel we will regard  $\{\beta_k\}$ ,  $k \in \mathbb{N}_0$  as a stochastic process. We will first assume that the transmission processes are Bernoulli distributed.

**Assumption 4.** The link transmission processes are independent and identically distributed (i.i.d.) with a common success probability  $p \in [0, 1]$ :

$$\Pr\{\gamma_k^{(i)} = 1\} = p, \quad \forall i \in \{1, 2, \dots, M-1\}. \quad (13)$$

The feedback link success processes are i.i.d., with

$$\Pr\{\delta_k^{(i)} = 1\} = q_i, \quad \forall i \in \{1, 2, \dots, M-1\}, \quad (14)$$

for given success probabilities  $q_1, q_2, \dots, q_{M-1} \in [0, 1]$ .  $\square$

Note that while the above assumption imposes that transmission processes are i.i.d., it does take into account the fact that radio connectivity from the actuator node to the other nodes will be distance dependent. It also does not impose that the processes  $\{\mu_k^{(i)}\}$ ,  $k \in \mathbb{N}_0$  for different nodes  $i$  are independent. However, the assumption made does imply stationarity, as apparent from Proposition 5 given below.

**Proposition 5.** Suppose that Assumption 4 holds. Then

$$\Pr\{\mu_k^{(i)} = 1\} = q_i + \sum_{j=1}^{i-1} p^j q_{i-j} \prod_{\ell=0}^{j-1} (1 - q_{i-\ell}) \quad (15)$$

$$\Pr\{i \in \mathcal{C}_k\} = q_i (1 - p \Pr\{\mu_k^{(i-1)} = 1\}) \quad (16)$$

$$\Pr\{c_k = i\} = p^{M-i} \Pr\{i \in \mathcal{C}_k\}, \quad (17)$$

for all  $k \in \mathbb{N}_0$  and  $i \in \{1, 2, \dots, M\}$ , and where  $q_M = 1$ .

The above result shows how the distributions of  $\mu_k^{(i)}$ ,  $\mathcal{C}_k$ , and of  $c_k$  depend upon the communication success probabilities; i.e., the distribution of  $\beta_k$ , here modeled as i.i.d.

**Example 6.** Suppose that Assumption 4 holds and that the feedback links are always available, that is,  $q_i = 1$ , for all  $i \in \{1, \dots, M\}$ . Expression (15) then provides that  $\Pr\{\mu_k^{(i)} = 1\} = 1$ , for all  $i \in \{1, \dots, M\}$ . Since, by (8),  $\Pr\{\mu_k^{(0)} = 1\} = 0$ , Proposition 5 gives that

$$\Pr\{i \in \mathcal{C}_k\} = \begin{cases} 1 & \text{if } i = 1 \\ 1 - p & \text{if } i \in \{2, \dots, M\} \end{cases}$$



**Table 1**  
Set  $\mathcal{C}_k$  (with location  $c_k$  in bold-face) for  $M = 3$ ; see Example 7.

$\delta_{k-1}^{(1)}$	$\gamma_k^{(1)}$	$\delta_{k-1}^{(2)}$	$\gamma_k^{(2)}$	$\mathcal{C}_k, c_k$	$\Pr$
1	1	Any	1	<b>{1}</b>	$q_1 p^2$
1	1	Any	0	<b>{1, 3}</b>	$q_1 p(1-p)$
1	0	1	1	<b>{1, 2}</b>	$q_1(1-p)q_2 p$
1	0	1	0	<b>{1, 2, 3}</b>	$q_1(1-p)q_2(1-p)$
1	0	0	Any	<b>{1, 3}</b>	$q_1(1-p)(1-q_2)$
0	Any	1	1	<b>{2}</b>	$(1-q_1)q_2 p$
0	Any	1	0	<b>{2, 3}</b>	$(1-q_1)q_2(1-p)$
0	Any	0	Any	<b>{3}</b>	$(1-q_1)(1-q_2)$

and the controller location sequence has the following geometric-like distribution

$$\Pr\{c_k = i\} = \begin{cases} p^{M-1} & \text{if } i = 1 \\ (1-p)p^{M-i} & \text{if } i \in \{2, 3, \dots, M\}. \end{cases} \quad (18)$$

On the other hand, if the actuator does not broadcast the plant input values at all ( $q_i = 0, \forall i \neq M$ ), then,  $\forall k \in \mathbb{N}_0$

$$\Pr\{\mu_k^{(i)} = 1\} = 0, \quad \forall i \in \{1, \dots, M-1\},$$

$$\Pr\{c_k = M\} = \Pr\{M \in \mathcal{C}_k\} = 1,$$

and the controller is collocated with the actuator (with probability one). This essentially corresponds to the conclusions made by the previous works (Goodwin et al., 2008; Robinson & Kumar, 2008) for alternative NCS configurations without feedback of plant inputs.  $\square$

**Example 7.** Consider  $M = 3$  and suppose that Assumption 4 holds. In this case, Proposition 5 establishes that

$$\Pr\{c_k = i\} = \begin{cases} q_1 p^2 & \text{if } i = 1, \\ p q_2 (1-p q_1) & \text{if } i = 2, \\ 1 - p q_2 - p^2 q_1 + p^2 q_1 q_2 & \text{if } i = 3. \end{cases}$$

Note that, since  $M$  is small, this result can alternatively be obtained by examining the probabilities of all possible transmission outcomes. This is illustrated in Table 1. Of course, for a large number of nodes, such a procedure is non-practical and use of Proposition 5 is preferable.  $\square$

## 5. Closed loop model

The algorithm proposed embodies a network driven distributed state estimation and control architecture. Closed loop dynamics depend upon transmission outcomes, the plant model (1) and nominal controller/estimator dynamics; see (2)–(3). To derive a compact model, it is convenient to introduce the aggregated state estimation vector  $\hat{x}_k \triangleq \text{col}(\hat{x}_k^{(1)}, \hat{x}_k^{(2)}, \dots, \hat{x}_k^{(M)}) \in \mathbb{R}^{Mn}$ . We also denote the “backup value” for  $u_k$  used at node  $i$  as

$$v_k^{(i)} = \begin{cases} L \hat{x}_k^{(i)} & \text{if } \mu_k^{(i)} = 0, \\ L \hat{x}_k^{(j)}, j = c_k^{(i)} & \text{if } \mu_k^{(i)} = 1, \end{cases}$$

see (9) and note that  $v_k^{(1)} = L \hat{x}_k^{(1)}$ , for all  $k \in \mathbb{N}_0$ . In view of (10), we have

$$v_k^{(i)} = b_k^{(i)} \hat{x}_k, \quad (19)$$

where  $b_k^{(1)} \triangleq e_1 \otimes L \in \mathbb{R}^{m \times Mn}$ , whereas for  $i \geq 2$ ,

$$b_k^{(i)} \triangleq e_\ell \otimes L \in \mathbb{R}^{m \times Mn} \quad (20)$$

$$\ell = \begin{cases} i & \text{if } c_k^{(i-1)} = 0 \vee \gamma_k^{(i-1)} = 0, \\ c_k^{(i-1)} & \text{if } c_k^{(i-1)} > 0 \wedge \gamma_k^{(i-1)} = 1, \end{cases}$$

depends on the realization of  $\beta_{k-1}$ ; see (12).

Since the algorithm gives  $u_k = v_k^{(M)}$ , the plant input estimates used by the state estimators satisfy:

$$\hat{u}_k^{(i)} = \begin{cases} v_k^{(M)} & \text{if } \delta_k^{(i)} = 1, \\ v_k^{(i)} & \text{if } \delta_k^{(i)} = 0 \end{cases}$$

$$= (\delta_k^{(i)}(e_M \otimes I_m) + (1 - \delta_k^{(i)})(e_i \otimes I_m))v_k, \quad (21)$$

where  $v_k \triangleq \text{col}(v_k^{(1)}, v_k^{(2)}, \dots, v_k^{(M)}) \in \mathbb{R}^{Mm}$  forms part of the internal variables used by the  $M$  state estimators.

Now, the plant model can be written as

$$x_{k+1} = Ax_k + Bv_k^{(M)} + w_k \quad (22)$$

and (22), (6) and (21) then provide:

$$\hat{x}_{k+1}^{(i)} = (I_n - K_{k+1}^{(i)}C)(A\hat{x}_k^{(i)} + B\hat{u}_k^{(i)}) + K_{k+1}^{(i)}(CAx_k + CBv_k^{(M)} + Cw_k + v_{k+1})$$

$$= K_{k+1}^{(i)}CAx_k + (I_n - K_{k+1}^{(i)}C)(e_i \otimes A)\hat{x}_k + d_k^{(i)}v_k + K_{k+1}^{(i)}(Cw_k + v_{k+1}), \quad (23)$$

$$d_k^{(i)} \triangleq (1 - \delta_k^{(i)})(I_n - K_{k+1}^{(i)}C)(e_i \otimes B) + ((1 - \delta_k^{(i)})K_{k+1}^{(i)}C + \delta_k^{(i)}I_n)(e_M \otimes B).$$

If we now introduce

$$\Theta_k \triangleq \text{col}(x_k, \hat{x}_k, v_k), \quad n_k \triangleq \text{col}(w_k, v_{k+1}), \quad (24)$$

and use (7), then (23) becomes

$$\hat{x}_{k+1}^{(i)} = \mathcal{D}^{(i)}(\beta_k)\Theta_k + \mathcal{E}^{(i)}(\beta_k)n_k,$$

$$\mathcal{D}^{(i)}(\beta_k) \triangleq [\Gamma_{k+1}^{(i)}KCA \quad (I_n - \Gamma_{k+1}^{(i)}KC)(e_i \otimes A) \quad d_k^{(i)}]$$

$$\mathcal{E}^{(i)}(\beta_k) \triangleq \Gamma_{k+1}^{(i)}K[C \quad I_p].$$

State estimators, thus, follow the dynamic relation

$$\hat{x}_{k+1} = \mathcal{D}(\beta_k)\Theta_k + \mathcal{E}(\beta_k)n_k \quad (25)$$

$$\mathcal{D}(\beta_k) \triangleq \begin{bmatrix} \mathcal{D}^{(1)}(\beta_k) \\ \vdots \\ \mathcal{D}^{(M)}(\beta_k) \end{bmatrix}, \quad \mathcal{E}(\beta_k) \triangleq \begin{bmatrix} \mathcal{E}^{(1)}(\beta_k) \\ \vdots \\ \mathcal{E}^{(M)}(\beta_k) \end{bmatrix}.$$

On the other hand, (19) provides

$$v_{k+1} = \mathcal{F}(\beta_k)\Theta_k + \mathcal{G}(\beta_k)n_k, \quad (26)$$

$$\mathcal{F}(\beta_k) \triangleq \begin{bmatrix} b_{k+1}^{(1)}\mathcal{D}(\beta_k) \\ \vdots \\ b_{k+1}^{(M)}\mathcal{D}(\beta_k) \end{bmatrix}, \quad \mathcal{G}(\beta_k) \triangleq \begin{bmatrix} b_{k+1}^{(1)}\mathcal{E}(\beta_k) \\ \vdots \\ b_{k+1}^{(M)}\mathcal{E}(\beta_k) \end{bmatrix}.$$

Expressions (22), (25) and (26) lead to the jump-linear model

$$\Theta_{k+1} = \mathcal{A}(\beta_k)\Theta_k + \mathcal{B}(\beta_k)n_k, \quad (27)$$

$$\mathcal{A}(\beta_k) \triangleq \begin{bmatrix} A & 0_{Mn} & e_M \otimes B \\ & \mathcal{D}(\beta_k) & \\ & \mathcal{F}(\beta_k) & \end{bmatrix}, \quad \mathcal{B}(\beta_k) \triangleq \begin{bmatrix} I_n & 0 \\ \mathcal{E}(\beta_k) \\ \mathcal{G}(\beta_k) \end{bmatrix}.$$

**Example 8.** Consider  $M = 2$ , in which case  $\beta_k = 2\gamma_{k+1}^{(1)} + \delta_k^{(1)}$  and  $\mathbb{I} = \{0, 1, 2, 3\}$ ; see (12). Since  $c_k^{(1)} = \delta_{k-1}^{(1)}, \delta_{k-1}^{(2)} = 1$  and  $b_{k+1}^{(1)} = e_1 \otimes L$  for all  $k \in \mathbb{N}_0$ , (20) yields:

$$b_{k+1}^{(2)} = \begin{cases} e_2 \otimes L & \text{if } \beta_k < 3, \\ e_1 \otimes L & \text{if } \beta_k = 3. \end{cases}$$

In this case, the matrices in (25) are given by

$$\begin{aligned} \mathcal{D}(\beta_k) &= \begin{bmatrix} KCA & (I_n - KC)(e_1 \otimes A) & d_k^{(1)} \\ \gamma_{k+1}^{(1)}KCA & (I_n - \gamma_{k+1}^{(1)}KC)(e_2 \otimes A) & e_2 \otimes B \end{bmatrix} \\ \mathcal{E}(\beta_k) &= \begin{bmatrix} KC & K \\ \gamma_{k+1}^{(1)}KC & \gamma_{k+1}^{(1)}K \end{bmatrix}, \\ d_k^{(1)} &= \begin{cases} [(I_n - KC)B & KCB] & \text{if } \beta_k \in \{0, 2\}, \\ e_2 \otimes B & \text{if } \beta_k \in \{1, 3\} \end{cases} \end{aligned}$$

thereby, characterizing the model (27).  $\square$

### 6. Performance analysis

To analyze the NCS via (27), we will adopt the stochastic modeling framework of Quevedo et al. (2013). Transmission outcome distributions depend upon the fading radio environment. To allow for temporal and spatial correlations of the radio environment (and possibly also for power and bit-rate control), in Quevedo et al. (2013) we used a Markovian network state,  $\{\mathcal{E}_k\}$ ,  $k \in \mathbb{N}_0$ , which takes values in a finite set, say  $\mathbb{B}$ . Each element of  $\mathbb{B}$  corresponds to a possible configuration of the physical environment, e.g., position of mobile objects. Dropout probabilities of individual channels, when conditioned on the network state, are considered independent. In the particular instance where  $\mathbb{B}$  has only one element, the model describes a situation with independent Bernoulli channels. For the present purposes, the model can be summarized via the following.

**Assumption 9.** The process  $\{\mathcal{E}_k\}$ ,  $k \in \mathbb{N}_0$  is an aperiodic homogeneous Markov Chain with transition probabilities  $p_{ij} = \Pr\{\mathcal{E}_{k+1} = j \mid \mathcal{E}_k = i\}$ ,  $i, j \in \mathbb{B}$  and stationary distribution  $\pi_i = \lim_{k \rightarrow \infty} \Pr\{\mathcal{E}_k = i\}$ ,  $i \in \mathbb{B}$ . The aggregated transmission outcome process  $\{\beta_k\}$  in (12) is conditionally independent given the network state  $\{\mathcal{E}_k\}$ , i.e.,  $\phi_{ij} \triangleq \Pr\{\beta_k = i \mid \mathcal{E}_k = j\}$ , for all  $(i, j) \in \mathbb{I} \times \mathbb{B}$ .  $\square$

It is worth noting that, with the above model, the process  $\beta_k$  is correlated, but not necessarily Markovian. However, the augmented jump process  $(\beta_k, \mathcal{E}_k)$ ,  $k \in \mathbb{N}_0$  forms a finite Markov Chain. Thus, under Assumption 9, (27) belongs to the class of Markov jump-linear systems, as studied for example in Costa, Fragoso, and Marques (2005), Lee and Dullerud (2007). In particular, Theorems 3.9 and 3.33 of Costa et al. (2005) establish necessary and sufficient conditions for mean-square stability (MSS) which can be stated in terms of feasibility of a linear-matrix inequality. The following result characterizes closed loop performance of the flexible networked control systems architecture of interest in the present work. It is tailored directly to the model in Assumption 9 without needing to resort to the augmented jump process  $(\beta_k, \mathcal{E}_k)$ .

**Theorem 10.** Suppose that Assumption 9 holds, that the system (27) is MSS and define  $W \triangleq \text{diag}(Q, R)$ ,

$$\begin{aligned} \bar{\mathcal{A}}_j &\triangleq \mathbf{E}\{\mathcal{A}(\beta_k) \mid \mathcal{E}_k = j\} = \sum_{i \in \mathbb{I}} \phi_{ij} \mathcal{A}(i), \quad j \in \mathbb{B}, \\ \bar{\mathcal{B}}_j &\triangleq \mathbf{E}\{\mathcal{B}(\beta_k) \mid \mathcal{E}_k = j\} = \sum_{i \in \mathbb{I}} \phi_{ij} \mathcal{B}(i), \quad j \in \mathbb{B}. \end{aligned} \tag{28}$$

Then

$$\lim_{k \rightarrow \infty} \mathbf{E}\{\Theta_k \Theta_k^T\} = \sum_{i \in \mathbb{B}} H_i, \tag{29}$$

where  $H_i$ ,  $i \in \mathbb{B}$  satisfy the linear system of equations:

$$H_i = \sum_{j \in \mathbb{B}} p_{ji} \bar{\mathcal{A}}_i H_j \Gamma(\bar{\mathcal{A}}_i)^T + \pi_i \bar{\mathcal{B}}_i W (\bar{\mathcal{B}}_i)^T. \tag{30}$$

In view of (24) and the fact that  $u_k = v_k^{(M)}$ , our result can be used to evaluate the plant state and input covariances.

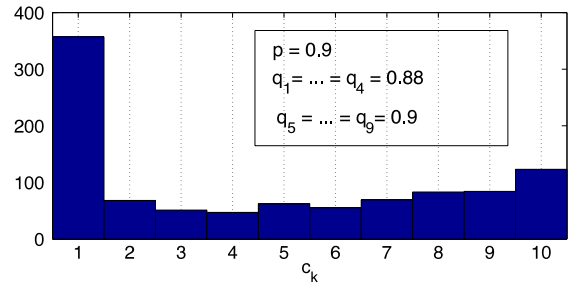


Fig. 3. Histogram of  $c_k$  for an i.i.d. network with success probabilities  $p = 0.9$ ,  $q_1 = \dots = q_4 = 0.88$ , and  $q_5 = \dots = q_9 = 0.9$ .

**Remark 11.** By using Lancaster (1970, Section 5),  $H_i$  in (30) can be written in terms of the solution to a system of linear equations.  $\square$

### 7. Simulation studies

We consider an NCS with  $M = 10$  nodes. In addition to the implementation of the controller via Algorithm 1, we also examine two baseline NCS architectures. In the first one, the controller and the estimator are fixed at the actuator node:

$$\begin{aligned} x_{k+1} &= Ax_k + BL\hat{x}_k^a + w_k, \\ \hat{x}_k^a &= A\hat{x}_{k-1}^a + Bu_{k-1} + \Gamma_k^{(10)}K(y_k - C(A\hat{x}_{k-1}^a + Bu_{k-1})), \end{aligned} \tag{31}$$

where  $\Gamma_k^{(10)}$  is as in (7). In the second scheme, the controller and the estimator are implemented at the sensor node. If the controller output is lost, then the previous plant input is held:

$$\begin{aligned} x_{k+1} &= Ax_k + \Gamma_k^{(10)}BL\hat{x}_k^s + (1 - \Gamma_k^{(10)})Bu_{k-1} + w_k, \\ \hat{x}_k^s &= A\hat{x}_{k-1}^s + B\hat{u}_{k-1}^s + K(y_k - C(A\hat{x}_{k-1}^s + B\hat{u}_{k-1}^s)), \end{aligned} \tag{32}$$

where  $\Gamma_k^{(10)}$  is as in (7) and

$$\hat{u}_{k-1}^s = \begin{cases} u_{k-1} & \text{if } \delta_{k-1}^{(1)} = 1, \\ L\hat{x}_{k-1}^s & \text{if } \delta_{k-1}^{(1)} = 0. \end{cases}$$

*Independent and identically distributed dropouts.* We first consider i.i.d. transmission processes as per Assumption 4. Fig. 3 illustrates a histogram of  $c_k$ , obtained by running the algorithm for 1000 steps with dropout probabilities as indicated. Whilst (18) shows that for small  $p$ , control calculations are at most times, carried out at the actuator node, Fig. 3 illustrates that if links are more reliable, then the controller will be placed at the sensor node at most time steps.

We first consider a noiseless unstable plant model (1), where

$$\begin{aligned} A &= \begin{bmatrix} 1.87 & -0.86 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ C &= [0.048 \quad 0.045] \end{aligned} \tag{33}$$

with Gaussian initial state having mean  $\bar{x}_0 = [5 \ 5]^T$  and covariance  $P_0 = 0.1 \times I_2$ . Controller and estimator gains  $L$  and  $K$  correspond to the steady state LQG/LQR controller with stage cost  $\|x_k\|^2 + \|u_k\|^2/10$ . All nodes use as initial state estimates,  $\hat{x}_0^{(i)} = [0 \ 0]^T$ . The network has i.i.d. dropouts with success probabilities  $p = 0.9$  and  $q_1 = \dots = q_9 = 1$ .

The baseline NCS (32) failed to stabilize the present system. Fig. 4 compares a typical plant output trajectory obtained by using the proposed algorithm (solid line) with that provided by the baseline NCS (31) (dashed line). As can be appreciated, the adaptive controller allocation algorithm presented reacts more quickly to plant outputs. It thereby recovers more quickly from the very bad local initial state estimates and provides control actions leading

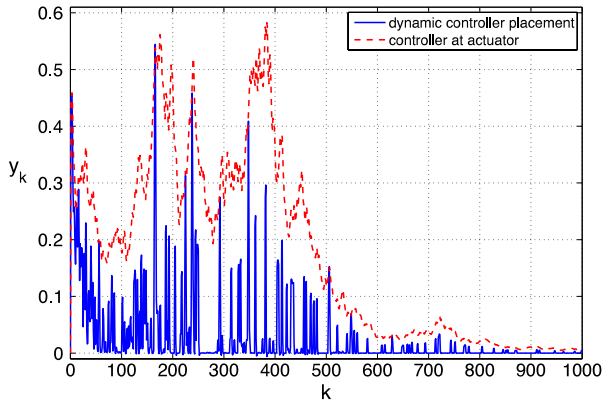


Fig. 4. Output trajectory of the plant model (33) for an i.i.d. network with success probabilities  $p = 0.9, q_1 = \dots = q_9 = 1$ .

Table 2

Performance indices  $J$  when controlling the system (35) over an i.i.d. network with  $q_1 = \dots = q_4 = 0.99, q_5 = \dots = q_9 = 0.995$ .

$p$	NCS (5)–(6)	NCS (31)	NCS (32)
0.95	33.5	38.5	(unstable)
0.97	33.2	37.6	56.1
0.99	33.1	34.7	34.1

to faster convergence to the origin. If we adopt the empirical performance measure

$$J \triangleq \sum_{k=1}^{1000} y_k^2, \tag{34}$$

then, with the dynamic architecture, we obtain  $J \approx 3$ , whereas for the baseline NCS described by (31),  $J \approx 11$ .

We next consider a plant model with an integrator, where

$$A = \begin{bmatrix} 1.8 & -0.8 \\ 1 & 0 \end{bmatrix}, \quad Q = 0.01 \times I_2, \quad R = 0.01 \tag{35}$$

and  $B$  and  $C$  as in (33). The initial state has mean  $\bar{x}_0 = [10 \ 10]^T$ . Table 2 illustrates how the performance gained by using the proposed method depends upon the network reliability. For the situation examined, larger performance gains are obtained with smaller  $p$ . For larger  $p$ , the performance gains become less relevant. This finding is intuitive, since for  $p \approx 1$  the network becomes transparent and overall performance is dominated by the nominal design (1)–(3).

**Network with moving obstacle.** We now focus on a network with an obstacle (e.g., a robot or crane) moving between four different positions; see Fig. 5. We model this as in Section 6, using the network state process  $\Xi_k \in \mathbb{B} = \{1, 2, 3, 4\}$ . The transition probabilities for  $\Xi_k$  are given by:

$$[p_{ij}] = \begin{bmatrix} 0.99 & 0.01 & 0 & 0 \\ 0.003 & 0.99 & 0.007 & 0 \\ 0 & 0.003 & 0.99 & 0.007 \\ 0.007 & 0 & 0.003 & 0.99 \end{bmatrix}.$$

The individual link reliabilities depend on the position of the obstacle. Nodes which are not blocked benefit from high success probabilities  $r \in [0.88, 1]$ . Due to the obstacle, some of the success probabilities will, at times, be lowered to 60%:

$$\Pr\{\gamma_k^{(i)} = 1 \mid \Xi_k = j\} = \begin{cases} 0.6 & \text{if } i \in \{2j - 1, 2j, 2j + 1\} \\ r & \text{in all other cases} \end{cases} \tag{36}$$

$$\Pr\{\delta_{k-1}^{(i)} = 1 \mid \Xi_k = j\} = \begin{cases} 0.6 & \text{if } i \in \{2j, 2j + 1\} \\ r & \text{in all other cases.} \end{cases}$$

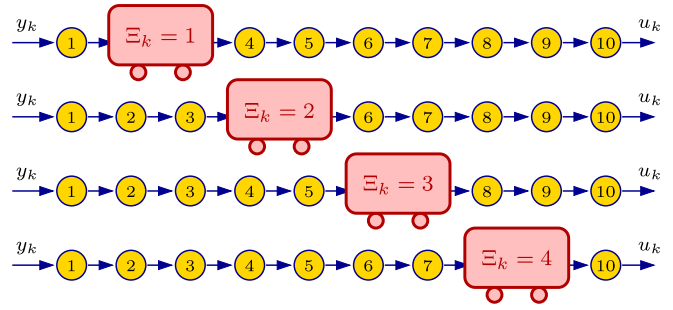


Fig. 5. Sensor-actuator network with moving obstacle.

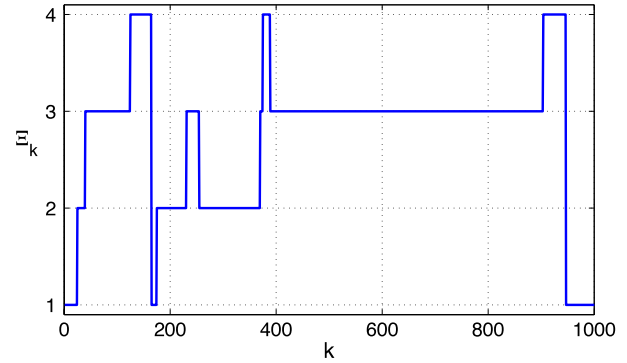


Fig. 6. Network state trajectory,  $\Xi_k$ .

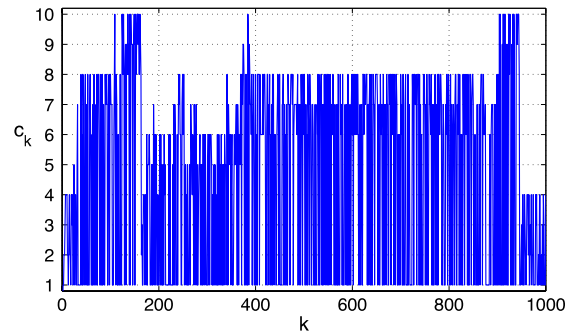
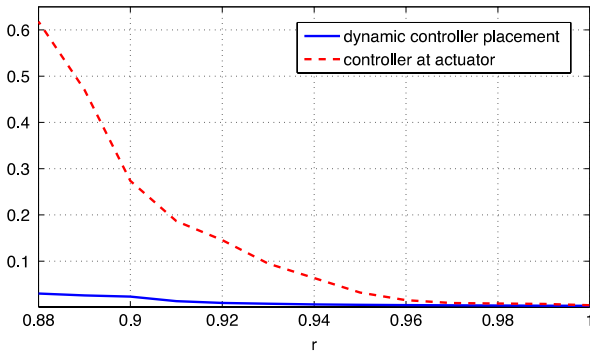


Fig. 7. Controller location  $c_k$  for the network in Fig. 5.

For  $r = 0.99$ , Figs. 6 and 7 illustrate how using Algorithm 1 the controller location depends upon the network state  $\Xi_k$ . It turns out that, in the present case, the plant input is always provided by one of the nodes located between the sensor node and the node immediately following the blocked ones. This behavior can be explained by noting that, in absence of the obstacle, the network is very reliable. In fact, if none of the nodes were blocked, then the algorithm would (almost) always locate the controller at the sensor node.

For (35) and with  $r = 0.99$ , use of the dynamic architecture proposed, gave  $J = 3$ . In contrast, if the controller is fixed at the actuator node, see (31), then  $J = 68.5$  was obtained. This amounts to a performance loss of 33%. In the situations examined, positioning the baseline NCS (32) failed to stabilize the plant model. Fig. 8, obtained using Theorem 10, illustrates how the trace of the covariance of the plant state (35) depends on the network parameter  $r$  in (36). In this figure, the solid line corresponds to the dynamic controller placement method, whereas the dashed line refers to the baseline controller (31). Our results clearly indicate that, without the need for controller re-design, the algorithm proposed in the present work has the potential to give significant performance gains when compared to earlier NCS configurations where node functionalities are fixed.



**Fig. 8.** Trace of the stationary covariance of the plant state  $x_k$  in (35) as a function of the success probability  $r$ ; see (36).

## 8. Conclusions

We have presented a flexible architecture for the implementation of a linear control law over a wireless sensor-actuator network using analog erasure channels without acknowledgments. With the algorithm provided, the role played by individual nodes depends on transmission outcomes. In particular, the controller location is adapted to the availability of past plant input values and transmission outcomes. By deriving a Markovian jump-linear system model, we established a closed form expression for the stationary covariance of the system state in the presence of correlated dropout processes. Future work may include extending the ideas to multiple-loops, to general network topologies, and to controller design.

## Appendix A. Proof of Proposition 5

With Assumption 4,  $\Pr\{\mu_k^{(1)} = 1\} = \Pr\{\delta_{k-1}^{(1)} = 1\} = q_1$ . It is easy to see from lines 11 and 27 of Algorithm 1 that

$$\begin{aligned} i \in \mathcal{C}_k &\iff \delta_{k-1}^{(i)} = 1 \\ &\quad \wedge (\gamma_k^{(i-1)} = 0 \vee (\gamma_k^{(i-1)} = 1 \wedge \mu_k^{(i-1)} = 0)) \\ &\iff \delta_{k-1}^{(i)} = 1 \wedge (\gamma_k^{(i-1)} = 0 \vee \mu_k^{(i-1)} = 0) \end{aligned} \quad (\text{A.1})$$

and that, similarly

$$\begin{aligned} \mu_k^{(i)} = 0 &\iff (\gamma_k^{(i-1)} = 0 \wedge \delta_{k-1}^{(i)} = 0) \\ &\quad \vee (\gamma_k^{(i-1)} = 1 \wedge \mu_k^{(i-1)} = 0 \wedge \delta_{k-1}^{(i)} = 0) \\ &\iff \delta_{k-1}^{(i)} = 0 \\ &\quad \wedge (\gamma_k^{(i-1)} = 0 \vee (\gamma_k^{(i-1)} = 1 \wedge \mu_k^{(i-1)} = 0)) \\ &\iff \delta_{k-1}^{(i)} = 0 \wedge (\gamma_k^{(i-1)} = 0 \vee \mu_k^{(i-1)} = 0) \end{aligned} \quad (\text{A.2})$$

for all  $i \in \{1, 2, \dots, M\}$ . (A.2) provides the recursion

$$\begin{aligned} \Pr\{\mu_k^{(i)} = 1\} &= 1 - \Pr\{\delta_{k-1}^{(i)} = 0\} \Pr\{\mu_k^{(i-1)} = 0 \vee \gamma_k^{(i-1)} = 0\} \\ &= 1 - (1 - q_i)(1 - \Pr\{\mu_k^{(i-1)} = 1 \wedge \gamma_k^{(i-1)} = 1\}) \\ &= 1 - (1 - q_i)(1 - p \Pr\{\mu_k^{(i-1)} = 1\}) \\ &= q_i + p(1 - q_i) \Pr\{\mu_k^{(i-1)} = 1\}, \end{aligned}$$

having explicit solution (15). On the other hand, (A.1) gives

$$\begin{aligned} \Pr\{i \in \mathcal{C}_k\} &= \Pr\{\delta_{k-1}^{(i)} = 1\} (1 - \Pr\{\gamma_k^{(i-1)} = 1\}) \\ &\quad \times \Pr\{\mu_k^{(i-1)} = 1\}, \end{aligned}$$

thus establishing (16). By (11) we obtain

$$\begin{aligned} \Pr\{c_k = i\} &= \Pr\{\max(\mathcal{C}_k) = i\} \\ &= \Pr\{i \in \mathcal{C}_k \wedge \gamma_k^i = \gamma_k^{i+1} = \dots = \gamma_k^{M-1} = 1\} \end{aligned}$$

for  $i \in \{1, \dots, M-1\}$ , whereas for the actuator node, we have  $\Pr\{c_k = M\} = \Pr\{M \in \mathcal{C}_k\}$ . This proves (17).  $\square$

## Appendix B. Proof of Theorem 10

By the law of total expectation, we have

$$\mathbf{E}\{\Theta_{k+1}\Theta_{k+1}^T\} = \sum_{i \in \mathbb{B}} H_{k+1,i}, \quad (\text{B.1})$$

where

$$H_{k+1,i} \triangleq \mathbf{E}\{\Theta_{k+1}\Theta_{k+1}^T \mid \mathcal{E}_k = i\} \Pr\{\mathcal{E}_k = i\}. \quad (\text{B.2})$$

Now, the system Eq. (27) together with the network fading model in Assumption 9 allow one to write

$$\begin{aligned} \mathbf{E}\{\Theta_{k+1}\Theta_{k+1}^T \mid \mathcal{E}_k = i\} &= \mathbf{E}\{(\mathcal{A}(\beta_k)\Theta_k + \mathcal{B}(\beta_k)n_k) \\ &\quad \times (\mathcal{A}(\beta_k)\Theta_k + \mathcal{B}(\beta_k)n_k)^T \mid \mathcal{E}_k = i\} \\ &= \mathbf{E}\{\mathcal{A}(\beta_k)\Theta_k\Theta_k^T\mathcal{A}(\beta_k)^T \mid \mathcal{E}_k = i\} \\ &\quad + \mathbf{E}\{\mathcal{B}(\beta_k)n_k n_k^T \mathcal{B}(\beta_k)^T \mid \mathcal{E}_k = i\} \\ &= \mathbf{E}\{\mathcal{A}(\beta_k)\Theta_k\Theta_k^T\mathcal{A}(\beta_k)^T \mid \mathcal{E}_k = i\} + \overline{\mathcal{B}}_i W (\overline{\mathcal{B}}_i)^T \end{aligned} \quad (\text{B.3})$$

since  $\{n_k\}$  is zero-mean i.i.d. The rule of total expectation, Bayes' rule and the Markovian property of (27) give that

$$\begin{aligned} \mathbf{E}\{\mathcal{A}(\beta_k)\Theta_k\Theta_k^T\mathcal{A}(\beta_k)^T \mid \mathcal{E}_k = i\} &= \sum_{j \in \mathbb{B}} \mathbf{E}\{\mathcal{A}(\beta_k)\Theta_k\Theta_k^T\mathcal{A}(\beta_k)^T \mid \mathcal{E}_k = i, \mathcal{E}_{k-1} = j\} \\ &\quad \times \Pr\{\mathcal{E}_{k-1} = j \mid \mathcal{E}_k = i\} \\ &= \sum_{j \in \mathbb{B}} \mathbf{E}\{\mathcal{A}(\beta_k)\Theta_k\Theta_k^T\mathcal{A}(\beta_k)^T \mid \mathcal{E}_k = i, \mathcal{E}_{k-1} = j\} \\ &\quad \times \Pr\{\mathcal{E}_k = i \mid \mathcal{E}_{k-1} = j\} \Pr\{\mathcal{E}_{k-1} = j\} / \Pr\{\mathcal{E}_k = i\} \\ &= \sum_{j \in \mathbb{B}} p_{ji} \overline{\mathcal{A}}_i \mathbf{E}\{\Theta_k\Theta_k^T \mid \mathcal{E}_{k-1} = j\} (\overline{\mathcal{A}}_i)^T \frac{\Pr\{\mathcal{E}_{k-1} = j\}}{\Pr\{\mathcal{E}_k = i\}}. \end{aligned} \quad (\text{B.4})$$

Substitution of (B.4) into (B.3) and then into (B.2) provides

$$\begin{aligned} H_{k+1,i} &= \sum_{j \in \mathbb{B}} p_{ji} \overline{\mathcal{A}}_i \mathbf{E}\{\Theta_k\Theta_k^T \mid \mathcal{E}_{k-1} = j\} (\overline{\mathcal{A}}_i)^T \\ &\quad \times \Pr\{\mathcal{E}_{k-1} = j\} + \overline{\mathcal{B}}_i W (\overline{\mathcal{B}}_i)^T \Pr\{\mathcal{E}_k = i\} \\ &= \sum_{j \in \mathbb{B}} p_{ji} \overline{\mathcal{A}}_i H_{k,j} (\overline{\mathcal{A}}_i)^T + \overline{\mathcal{B}}_i W (\overline{\mathcal{B}}_i)^T \Pr\{\mathcal{E}_k = i\}. \end{aligned} \quad (\text{B.5})$$

Since the NCS is assumed MSS, it is asymptotically wide-sense stationary (Costa et al., 2005, Theorem 3.33). If we define  $H_i \triangleq \lim_{k \rightarrow \infty} H_{k,i}$ ,  $i \in \mathbb{B}$ , and recall that  $\{\mathcal{E}_k\}$  is aperiodic, then (B.5) becomes (30), and (B.1) establishes (29).  $\square$

## References

- Antsaklis, P., & Baillieul, J. (2007). Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1), 5–8.
- Antunes, D., Hespanha, J., & Silvestre, C. (2013). Stability of networked control systems with asynchronous renewal links: an impulsive systems approach. *Automatica*, 49(2), 402–413.
- Chen, J., Johansson, K. H., Olariu, S., Paschalidis, I. Ch., & Stokmenovic, (2011). Guest editorial special issue on wireless sensor and actuator networks. *IEEE Transactions on Automatic Control*, 56(11), 2244–2246.
- Chiuso, A., & Schenato, L. (2011). Information fusion strategies and performance bounds in packet-drop networks. *Automatica*, 47, 1304–1316.
- Costa, O. L. V., Fragoso, M. D., & Marques, R. P. (2005). *Discrete-time Markov jump linear systems*. London, UK: Springer-Verlag.
- Donkers, M. C. F., Heemels, W. P. M. H., van de Wouw, N., & Hetel, L. (2011). Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9), 2101–2115.
- Goodwin, G. C., Quevedo, D. E., & Silva, E. I. (2008). Architectures and coder design for networked control systems. *Automatica*, 44(1), 248–257.
- Huang, M., & Dey, S. (2007). Stability of Kalman filtering with Markovian packet losses. *Automatica*, 43(4), 598–607.
- Imer, O. C., Yüksel, S., & Başar, T. (2006). Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9), 1429–1439.
- Lancaster, P. (1970). Explicit solutions of linear matrix equations. *SIAM Review*, 12(4), 544–566.



- Lee, J.-W., & Dullerud, G. E. (2007). A stability and contractiveness analysis of discrete-time Markovian jump linear systems. *Automatica*, 43(1), 168–173.
- Nešić, D., & Teel, A. R. (2004). Input-to-state stability of networked control systems. *Automatica*, 40, 2121–2128.
- Pajic, M., Sundaram, S., Pappas, G. J., & Mangharam, R. (2011). The wireless control network: a new approach for control over networks. *IEEE Transactions on Automatic Control*, 56(10), 2305–2318.
- Quevedo, D. E., Ahlén, A., & Johansson, K. H. (2013). State estimation over sensor networks with correlated wireless fading channels. *IEEE Transactions on Automatic Control*, 58(3), 581–593.
- Quevedo, D. E., Ahlén, A., Leong, A. S., & Dey, S. (2012). On Kalman filtering over fading wireless channels with controlled transmission powers. *Automatica*, 48(7), 1306–1316.
- Quevedo, D. E., Ahlén, A., & Østergaard, J. (2010). Energy efficient state estimation with wireless sensors through the use of predictive power control and coding. *IEEE Transactions on Signal Processing*, 58(9), 4811–4823.
- Quevedo, D.E., Johansson, K.H., Ahlén, A., & Jurado, I. (2012). Dynamic controller allocation for control over erasure channels. In *Proc. 3rd IFAC workshop on distributed estimation and control in networked systems*, Santa Barbara, CA.
- Robinson, C. L., & Kumar, P. R. (2008). Optimizing controller location in networked control systems with packet drops. *IEEE Journal on Selected Areas in Communications*, 26(4), 661–671.



**Daniel E. Quevedo** received Ingeniero Civil Electrónico and Magister en Ingeniería Electrónica degrees from the Universidad Técnica Federico Santa María Valparaíso Chile in 2000. In 2005, he received the Ph.D. degree from The University of Newcastle, Australia, where he is currently an Associate Professor. He has been a visiting researcher at various institutions, including Uppsala University, Sweden, KTH Stockholm, Sweden, Aalborg University, Denmark, Kyoto University, Japan, INRIA Grenoble, France, University of Notre Dame, USA, and The Hong Kong University of Science and Technology.

Dr. Quevedo was supported by a full scholarship from the alumni association during his time at the 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 also a finalist in 2002. In 2009, he was awarded a five-year Australian Research Fellowship. His research interests include several areas of automatic control, signal processing, and power electronics.



**Karl H. Johansson** is Director of the KTH ACCESS Linnaeus Centre and Professor at the School of Electrical Engineering, Royal Institute of Technology, Sweden. He is a Wallenberg Scholar and has held a six-year Senior Researcher Position with the Swedish Research Council. He is Director of the Stockholm Strategic Research Area ICT The Next Generation. He received M.Sc. and Ph.D. degrees in Electrical Engineering from Lund University. He has held visiting positions at UC Berkeley (1998–2000) and California Institute of Technology (2006–2007). His research interests are in networked control systems,

hybrid and embedded system, and applications in smart transportation, energy, and automation systems. He has been a member of the IEEE Control Systems Society Board of Governors and the Chair of the IFAC Technical Committee on Networked Systems. He has been on the Editorial Boards of several journals, including *Automatica*, *IEEE Transactions on Automatic Control*, and *IET Control Theory and Applications*. He has been Guest Editor for special issues, including the one on “Wireless Sensor and Actuator Networks” of *IEEE Transactions on Automatic Control* 2011. He was the General Chair of the ACM/IEEE Cyber-Physical Systems Week 2010 in Stockholm and IPC Chair of many conferences. He has served on the Executive Committees of several European research projects in the area of networked embedded systems. In 2009, he received the Best Paper Award of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems. In 2009, he was also awarded Wallenberg Scholar, as one of the first ten scholars from all sciences, by the Knut and Alice Wallenberg Foundation. He was awarded an Individual Grant for the Advancement of Research Leaders from the Swedish Foundation for Strategic Research in 2005. He received the triennial Young Author Prize from IFAC in 1996 and the Peccei Award from the International Institute of System Analysis, Austria, in 1993. He received Young Researcher Awards from Scania in 1996 and from Ericsson in 1998 and 1999. He is a Fellow of the IEEE.



**Anders Ahlén** is full professor and holds the chair in Signal Processing at Uppsala University where he is also the head of the Signals and Systems Division of The Department of Engineering Sciences. He was born in Kalmar, Sweden, and received the Ph.D. degree in Automatic Control from Uppsala University. He was with the Systems and Control Group, Uppsala University from 1984 to 1992 as an Assistant and Associate Professor in Automatic Control. During 1991 he was a visiting researcher at the Department of Electrical and Computer Engineering, The University of Newcastle, Australia. He was a visiting professor at the same university in 2008. In 1992 he was appointed as Associate Professor of Signal Processing at Uppsala University. During 2001–2004 he was the CEO of Dirac Research AB, a company offering state-of-the-art audio signal processing solutions. He is currently the chairman of the board of the same company. Since 2007 he has been a member of the Uppsala VINN Excellence Center for Wireless Sensor Networks, WISENET. His research interests, which include Signal Processing, Communications and Control, are currently focused on Signal Processing for Wireless Communications, Wireless Sensor Networks, Wireless Control, and Audio Signal Processing.

From 1998 to 2004 he was the Editor of Signal and Modulation Design for the *IEEE Transactions on Communications*.



**Isabel Jurado** received the Bachelor and Master degrees in engineering from Seville University in 2007 and 2009, respectively. She is currently working towards the Ph.D. degree in networked control systems at Seville University. Her current research interests are in the fields of robust control, delayed systems and systems controlled over unreliable channels.