

Time-delay estimation and finite-spectrum assignment for control over multi-hop WSN

Emmanuel Witrant, Pangun Park, Mikael Johansson

▶ **To cite this version:**

Emmanuel Witrant, Pangun Park, Mikael Johansson. Time-delay estimation and finite-spectrum assignment for control over multi-hop WSN. Mazumder, Sudip K. (Ed.). Wireless Networking Based Control, Springer, pp.135-152, 2011, ISBN 978-1-4419-7392-4, e-ISBN 978-1-4419-7393-1. 10.1007/978-1-4419-7393-16 . hal-00667148

HAL Id: hal-00667148
https://hal.science/hal-00667148

Submitted on 7 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chapter 6
Time-delay estimation and finite-spectrum assignment for control over multi-hop WSN

Emmanuel Witrant ⋆, Pangun Park, Mikael Johansson ⋆⋆

Abstract

Recent technological developments and the need for global control strategies to meet with stringent performance requirements rendered the use of wireless sensor networks (WSN) of prime importance for today's automation. Such communication devices heavily affect the information transport, especially when a multi-hop configuration is introduced to minimize the energy consumption of the network. The associated time-delays are strongly varying and necessitate the design of a dedicated control approach for remote regulation, as proposed in this chapter. First, the communication process is analyzed and illustrated with the Breath protocol, which results in a description of the delay in the frequency domain. Such analysis motivates the use of a CUMSUM Kalman filter to estimate the delay from round-trip-time (RTT) measurements while selecting the desired frequencies. This estimation then allows for the setup of a state-predictor that ensures a finite spectrum assignment (FSA) on the closed loop spectrum. The efficiency of this approach and the impact of packet losses is finally investigated on an inverted pendulum using experimental network measurements.

Key words: Network controlled systems, wireless networks, multi-hop networks, time-delay estimation, predictive control, finite spectrum assignment.

⋆ E. Witrant is with Control Systems Department, UJF GIPSA-lab, University of Grenoble, Saint Martin d'Hères, France. Corresponding author. Email: emmanuel.witrant@gipsa-lab.inpg.fr.

⋆⋆ P. Park and M. Johansson are with the ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. E-mails: pgpark, mikaelj@ee.kth.se.
This work was supported by the EU project FeedNetBack, the Swedish Research Council, the Swedish Strategic Research Foundation, and the Swedish Governmental Agency for Innovation System.

6.1 Introduction

There are major advantages in terms of increased productivity and reduced installation costs in the use of wireless communication technology in industrial control systems [1, 2]. Several wireless technologies are currently reaching the plant and workshop floor addressing a range of applications. A multi-hop wireless network is one such technology that offers great flexibility, but also challenges when it comes to predictability.

Wireless sensor networks (WSN) are used in emerging applications in industrial automation and dedicated algorithms, such as randomized multi-hop routing protocols [3], were recently shown to be useful in these applications. From the perspective of designing control systems utilizing wireless multi-hop networks, one of the most significant characteristics is the need of hiding the system complexity through suitable abstraction of the networks and their nodes. One simple but important way of doing this is by interpreting the network as a communication channel with time-varying delay.

Experimental results on telerobotics over TCP/IP networks illustrate the importance of varying transmission delays in the control loop [4], where the authors combine a generic estimation/prediction control scheme with a buffering of the received packets, which compensates for the delays jitter. While the buffer may decrease the transient performances of the closed-loop system [5], this work motivates the use of estimation/prediction schemes to achieve robust and efficient control over networks. Another interesting experimental result is provided in [6], where a peer-to-peer wireless network (802.11b) and a UDP protocol are used to transmit the sensors and actuators signals. An inverted pendulum is controlled through this network using LQR techniques without compensating the delay (classical state feedback), which results in relatively large oscillations in the system response (complex poles induced by the delay) even for small latencies (average value less then 3.6 ms). The packet losses (significant effect above 15%) are efficiently compensated using a dynamic adjustment of the sampling interval. These experimental results illustrate the fact that, when the transmissions are fairly deterministic and the control/sensors signals have a significant effect on the network load, simple solutions can be used to compensate for the packet losses and timing issues while the delay effects need a more careful consideration.

The main focus of this chapter is a new scheme for control over wireless multi-hop networks based on time-delay and finite spectrum assignment (FSA). Time-varying delays for the transmission of sensor and control data over the wireless network are caused by a randomized multi-hop routing protocol called Breath [3]. Breath is designed for control applications using WSN and it ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. We also consider the modification of Breath by using an Automatic Repeat reQuest (ARQ) mechanism [7] to improve the packet reception rate (reliability). The characteristics of the routing protocol together with lower-layer network mechanisms give rise to

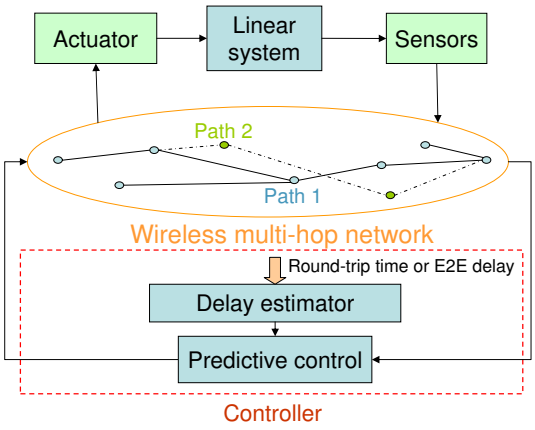


Fig. 6.1 Overview of the system and control setup.

a delay process with high variance and stepwise changing mean. Estimation of the transport delay over multi-hop networks is particularly difficult, since detailed modeling from first principles is prevented by the high complexity and multitude of traffics over the network. Here we use the low-order model proposed in [8] to set a predictive control scheme with a delay estimator. The estimator is based on a Kalman filter with a change detection algorithm [8]. It is able to track the delay mean changes but efficiently attenuate the high frequency jitter. Compared to similar works such as [9], where an output feedback scheme uses the maximal value of the delay and GPS synchronization for remote control over Internet, we *explicitly* compensate for the time-varying delay effect. The control scheme is analyzed and its implementation detailed. Network data from an experimental setup, where the randomized multi-hop routing protocol [3] is implemented on sensor nodes, are used to illustrate the efficiency of the approach. This work is an extended version of [10], with additional insights on the delay estimation and WSN aspects.

The outline of the paper is as follows. Section 6.2 defines the considered problem of control over a wireless multi-hop network. The wireless network and its randomized routing protocol is described in Section 6.3. The time-delay estimator is presented in Section 6.4. Section 6.5 presents the controller and its delay compensator. Experimental results are given in Section 6.6, where the tradeoff between lossless communications and time-delays is discussed.

6.2 Problem description

The problem considered is depicted in Figure 6.1, where a linear system is remotely controlled over a wireless multi-hop network. We suppose that round-trip time or end-to-end delay measurements (i.e., between the sensors and controller, which also implies a proper clock synchronization) are available. A predictive control law uses the filtered measurements to compensate for variations induced by the multi-hop network, which induces communication delays (time-varying average latency and jitter), packet losses and communication bandwidth constraints. More precisely, we consider the remote control of linear systems that write as:

$$\dot{x}(t) = Ax(t) + Bv(t), \quad x(0) = x_0 \tag{6.1}$$

$$y(t) = Cx(t) \tag{6.2}$$

where $x \in R^n$ is the internal state, $v \in R$ is the system input, $y \in R^m$ is the system output, and A, B, C are matrices of appropriate dimensions. The pairs (A, B) and (A, C) are assumed to be controllable and observable, respectively, but no assumption is made on the stability of A. The delays introduced by the network dynamics are defined as:

- $\tau_1(t)$: delay between the sensors and the controller,
- $\tau_2(t)$: delay between the controller and the actuator.

The actuator is event-driven and uses the last received signal. Packet losses are included in this description by considering that the last received and most recent controller output $u(\cdot)$ is taken as the input for the system, which is then:

$$\begin{cases} w(t_k) = u(t_k - \tau_2(t_k)) & \text{if packet } k \text{ is received} \\ w(t_k) = w(t_{k-1}) & \text{else} \end{cases} \tag{6.3}$$

where $w(\cdot)$ is the signal received by the actuator from the network, $u(\cdot)$ is the controller output and $v(t) = w(t_k)\ \forall\, t \in [t_k, t_{k+1})$. The index k is used to differentiate the discrete data packets from the continuous physical variables. The chronology of the signals is indicated in the transmitted data in order to reconstruct their history on the receiver site. The receivers consider only the most recent information, meaning that a packet is also considered as lost if a more recent one is received first.

6.3 Wireless multi-hop networks

In many scenarios of relevant interest, wireless multi-hop networks do not have fixed communication paths [11], but the end-to-end path followed by packets happens according to a dynamic selection of hops [12, 13]. Indeed, it is often impossible to build fixed routing tables for these networks due

to the time-varying communication channel and network topology. In such networks, a packet is routed to a particular next-hop node as long as the node selected for a hop ensures progress toward destination. Furthermore, a major technique to ensure power savings and longer network lifetime is to turn off a node whenever its presence is not strictly required for the correct operation of the network [2, 3, 13, 14]. In this case, each node goes to sleep for a random amount of time depending on traffic and network conditions, which means that the network topology is changing randomly.

Consider the multi-hop network reported in Figure 6.1, where the transmitter to receiver path is composed of a time-varying number of hops $h(t)$. Therefore we can consider the following sources of delay:

- the time α_i to wait before sending a data packet, which is typically a random variable;
- the time F to forward a data packet once the connection with the next node is set up, which is constant and contains the propagation and transmission delay;
- possibly the time β_i induced by an ARQ mechanism, which retransmits a packet for a predefined number of time if no acknowledgement is received.

The end-to-end delay can then be expressed as follows:

$$\tau(t) = h(t)F + \sum_{i=1}^{h(t)} (\alpha_i + \beta_i) \tag{6.4}$$

For the Breath protocol [3] used to set the experimental setup, the routing, medium access control and sleeping discipline are randomized in such a way that the energy consumption of the communication network is minimized while meeting the constraints on packet delay and reliability. Since it is representative of a class of protocol for wireless automation, we take it as a reference for the experimental results. With this protocol, α_i depends on the exponentially distributed wake up rate, which average value $\mu_{c,i}$ is constrained by a function of the traffic rate, channel condition and application requirements. Note that, since $h(t)$, α_i and β_i are random variables, the end-to-end delay clearly results in a random variable.

If one looks at the outcomes of (6.4), there are three possible behaviors:

1. Low frequency delays, due to changes in the number of hops $h(t)$. The end-to-end number of hops is expected to change when an obstacle changes the channel conditions for long time (i.e. the shadow fading of the wireless propagation). Since shadowing changes with low frequency, this causes low frequency changes in the delay.
2. Middle frequency delays, which depend on the random selection of the next awaken hop for a fixed number of clusters (related to $\mu_{c,i}$ in the example considered). This is induced by the fact that a transmitting node has to wait for some node to wake up in the forwarding region. For

example, in the randomized protocol, this phenomenon is directly caused by the wake up rate $\mu_{c,i}$, which is higher than the rate with which the shadow fading changes.

3. High frequency delays, related to the transmissions between nodes and packet loss probability. Since the wireless transmission may be lossy due to fast fading (which causes the execution of the ARQ) there might be highly time-varying delays. Hence, this delay is directly related to the packet loss probability. Note that the frequency of the fast fading is very high in comparison with the shadow fading.

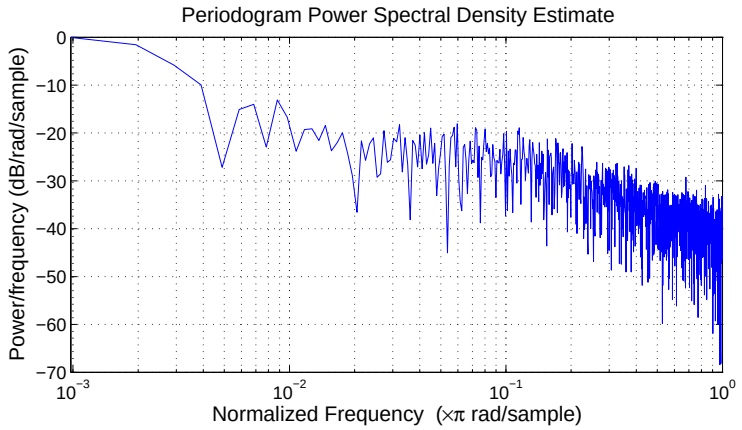


Fig. 6.2 Power spectral density of the delays induced by a three-hops network.

Example 1. Considering the WSN multi-hop network presented in Section 6.6, which describes the specific experimental setup, time-delay measurements are obtained for a three-hops configuration. The power spectral density of these delays is obtained thanks to the periodograms computation and depicted in Figure 6.2. We can clearly notice the dominent low-frequency component of the delays (0 $dB/rad/sample$), followed by the medium (two peaks above -15 $dB/rad/sample$) and high (below -18 $dB/rad/sample$, close to white noise) frequency behaviors.

The measurements can also be used to investigate the class of models that can be associated with such delays. As an example, a spectral model (empirical transfer function estimate) and a time-domain model (auto regressive moving average - ARMA) are presented in Figure 6.3. The ARMA model is set with a second order polynomial on the delay and a first order polynomial on the white noise. While providing for a rough approximation of the signal, we can see that the ARMA model gives a reasonable approximation of the amplitudes at the three frequency levels. This motivates the structure of the delay estimation presented in the next section.

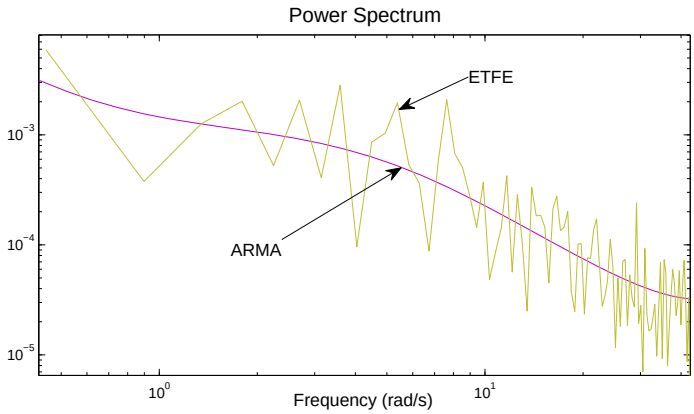


Fig. 6.3 Empirical transfer function estimate (ETFE) and auto regressive moving average (ARMA) model of the network delays.

6.4 Time-delay estimator

A critical component of the proposed control strategy is the delay estimator, which output is used to set the time-varying horizon of the predictive controller. Examining the real delay traces from our network (introduced in the previous example), shown in dashed lines in Figure 6.4, we see that the delay exhibits a large variability and persistent changes in the mean. Although these changes can be traced back to a particular feature of our protocol (discussed in the previous section) similar effects have been reported for a wide range of network quantities and network technologies [15]. Since the controller-actuator latency is not immediately available to the controller, but has to be deduced from the (delayed information about the) controller-actuator round-trip time, we will look for a delay estimator that allows to smooth out the fast variations while being able to react quickly to the changes in mean.

To this end, we propose to use a combination of Kalman filter and CUMSUM change-detection (cf., [8, 16]). The underlying signal model is:

$$\begin{aligned} \tau_{k+1} &= \tau_k + v_k \\ r_k &= c\tau_k + w_k \end{aligned}$$

where τ_k is the one-way latency, r_k is the measured time, c is the number of delays included in r_k (i.e., 1 if the end-to-end delay of the channel is measured and 2 if it is the round trip time), w_k is a white-noise sequence with covariance R_k and:

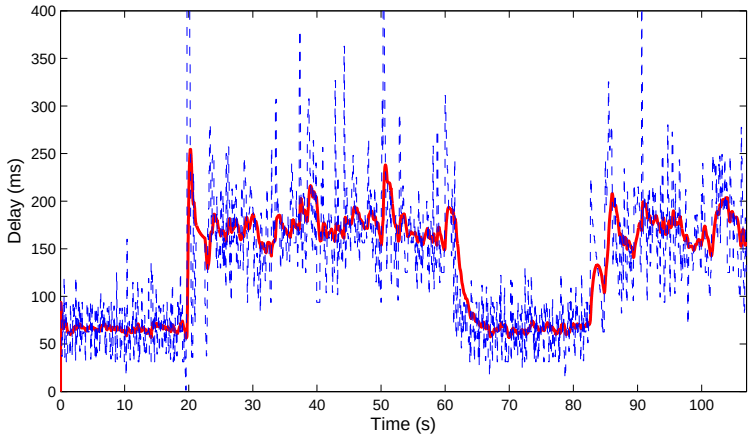


Fig. 6.4 Delay traces and delay estimates computed with the combined Kalman-filter and CUMSUM detector.

$$v_k = \begin{cases} 0 & \text{with probability } 1-q, \\ \nu & \text{with probability } q, \text{ where } \mathrm{Cov}(\nu_k) = q^{-1}Q_k, \end{cases}$$

models the abrupt changes in the mean. The associated Kalman filter updates are given by:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + K_k(r_k - c\hat{\tau}_k), \text{ with}$$
$$K_k = \frac{cP_{k-1}}{R_k + c^2P_{k-1}}, \quad P_k = P_{k-1} - \frac{c^2P_{k-1}^2}{R_k + c^2P_{k-1}} + Q_k$$

The tracking ability of the filter is proportional to the variance of the process noise Q_k, but increasing Q_k also decreases the smoothing effect that we are looking for.

To improve performance, we run a CUMSUM detector in parallel with the Kalman filter. The detector looks at the prediction errors $\varepsilon_k = r_k - c\hat{\tau}_k$, sums up the negative and positive prediction errors:

$$g_k^+ = \max(g_{k-1} + \varepsilon_k - \kappa, 0)$$
$$g_k^- = \max(g_{k-1} - \varepsilon_k - \kappa, 0)$$

and monitors if g_k^+ or g_k^- exceeds a threshold g_{thr}. Here, the parameter κ is a design parameter that specifies the negative drift (see [16] for design guidelines). At these threshold crossings, Q_k is set to a large value, g^+ and g^- are reset to zero, $\hat{\tau}_k$ is set to $c^{-1}r_k$ and the Kalman filter iterations resumed. The combined Kalman-filter and CUMSUM test allows to achieve

both fast tracking of the changes in the mean and good suppression of the high-frequency variations, as illustrated in Figure 6.4 (solid line).

6.5 Predictive control approach

The control approach proposed here is based on the state predictor with a time-varying horizon first proposed in [17]. The main advantage of such method compared to other delay compensation strategies [18] is to allow for a *finite spectrum assignment* on systems with time-varying delays [19, 20]. An overview on the use of state predictors in networked control systems is provided in [5], where it is compared with more classical control approaches.

6.5.1 Ideal delay compensation

We use the results established in [21] and subsequent works, where the delay induced by the network dynamics is explicitly taken into account in the control setup. For simplicity and to focus on the network effect on the closed-loop system, we suppose that the full delayed state[3] $x(t-\tau_1)$ is available to set the control law. This hypothesis can be easily removed using an observer-based control setup as the one proposed in [22]. When there is no packet loss and the delay is fully known (and can be predicted), the main theorem in [21] can be extended to the case where both communication channels experience a delay as follows.

Theorem 1. *Consider the system:*

$$\dot{x}(t) = Ax(t) + Bu(t-\tau_2), \quad x(0) = x_0 \tag{6.5}$$

with (A, B) a controllable pair. Assume that the network dynamics is such that the following holds for τ_1 and τ_2:
A1) $\tau_{1,2} \in \mathcal{C}(R^+, [0, \tau_{max}]) \quad \forall t$,
A2) $\dot{\tau}_{1,2} < 1 \quad \forall t \geq t_0$.
Then the feedback control law:

$$u(t) = -Ke^{A\delta}\left[e^{A\tau_1}x(t-\tau_1) + e^{At}\int_{t-\tau_1}^{t+\delta} e^{-A\theta}Bu(\theta-\tau_2(\theta))d\theta\right] \tag{6.6}$$

$$\dot{\delta}(t) = -\frac{\lambda}{1-d\tau_2(\zeta)/d\zeta}\delta + \frac{d\tau_2(\zeta)/d\zeta + \lambda\tau_2(\zeta)}{1-d\tau_2(\zeta)/d\zeta} \tag{6.7}$$

[3] The time dependency of the variables $\delta(t)$, $\tau_1(t)$ and $\tau_2(t)$ are omitted in the notations from this point.

with $\zeta = t + \delta$, λ a positive constant, $\delta(0) = \delta_0$, ensures that the system trajectories $x(t)$ converge to zero.

Proof. (Outline) First note that (6.6) is obtained by setting the feedback law as:

$$u(t) = -Kx(t + \delta(t))$$

where $x(t + \delta(t)$ is obtained from the fundamental of (6.5) as:

$$x(t+\delta) = e^{A\delta}\left[e^{A\tau_1}x(t-\tau_1) + e^{At}\int_{t-\tau_1}^{t+\delta} e^{-A\theta}Bu(\theta - \tau_2(\theta))d\theta\right]$$

Noting that with $(A1) - (A2)$ the dynamics (6.7) ensures the exponential convergence towards zero of $\delta(t) - \tau(t + \delta(t))$ the closed-loop dynamics is set as:

$$\frac{d}{dt}x(t+\delta) = (A - BK)x(t+\delta)$$

Such dynamics is stable if $(A - BK)$ is Hurwitz and the convergence of $x(t)$ is a direct consequence of the one of $x(t + \delta)$. ◇◇◇

If the communication delay is perfectly known and satisfies A1-A2 (which correspond to a lossless network description), then the closed-loop system behavior can be inferred from the *time-shifted* dynamics:

$$\frac{dx}{d\zeta} = (A - BK)x(\zeta)$$

and its spectrum coincides with the spectrum of the matrix:

$$A - \left|\frac{1+\dot{\delta}}{\dot{\tau}(t+\delta)}\right| e^{-A\delta}BK$$

Considering the worst-case delay properties to set the closed loop spectrum, which correspond to a maximum delay τ_{max} and a maximum delay variation $\dot{\tau}_{max}$, the feedback gain has to be chosen such that the matrix:

$$A - \left|\frac{1}{1-\dot{\tau}_{max}}\right| e^{-A\tau_{max}}BK \tag{6.8}$$

is Hurwitz.

6.5.2 Use of the delay estimator

The control law is set using the delay estimator described in section 6.4. Both delays τ_1 and τ_2 are then replaced by $\hat{\tau}$ in the control setup description

introduced in Theorem 1. This is motivated by the fact that the predictor architecture in used to compensate for the time-variations of the latency (i.e., the slow and medium dynamics of the delay), which are estimated with the CUMSUM Kalman filter. It is reasonable to suppose that the average values of the delays in both communication channels are close and provided by the estimator. The control law is then given as:

$$u(t) = -Ke^{A\delta}\left[e^{A\hat{\tau}}x(t-\tau_1) + e^{At}\int_{t-\hat{\tau}}^{t+\delta} e^{-A\theta}Bu(\theta-\hat{\tau}(\theta))d\theta\right] \tag{6.9}$$

The predictor horizon is computed using the actual values of the estimated delay and delay variation directly instead of the predicted ones (at time ζ). We then approximate δ with $\hat{\delta}$, which dynamics is given by:

$$\dot{\hat{\delta}}(t) = \frac{\dot{\hat{\tau}}(t) + \lambda(\hat{\tau}(t) - \hat{\delta}(t))}{1 - \dot{\hat{\tau}}(t)} \tag{6.10}$$

6.5.3 Control algorithm

Special care has to be taken in the computation of the control law (6.9)-(6.10), especially concerning the integral term used in the predictor's part. Indeed, the discretization of the integral may introduce some numerical instabilities, as illustrated in [23] and explained in [24]. The effect of these instabilities is further increased if the communication channel experiences packet losses and bandwidth limitation. A solution to this problem is to include the discretized integral in a dynamic control law, as described in this section.

First, considering the integral term:

$$\mathcal{I}(t) \doteq e^{At}\int_{t-\hat{\tau}}^{t+\hat{\delta}} e^{-A\theta}Bu(\theta-\hat{\tau}(\theta))d\theta$$

computed at the time instant denoted by k, we have that:

$$\mathcal{I}_k = e^{A\hat{\tau}_k}\int_{0}^{\hat{\tau}_k+\hat{\delta}_k} e^{-A\iota}Bu(\mu+t_k-\hat{\tau}_k-\hat{\tau}(\mu+t_k-\hat{\tau}_k))d\mu$$

Introducing the function $f(\mu) \doteq e^{-A\mu}Bu(\mu+t_k-\hat{\tau}_k-\hat{\tau}(\mu+t_k-\hat{\tau}_k))$ and considering that the delay evolution can not be predicted, the integral term is computed with:

$$f(\mu) = e^{-A\mu}Bu(\min(\mu + t_k - \hat{\tau}_k - \hat{\tau}(\min(\mu + t_k - \hat{\tau}_k, t_k)), t_k))$$

to ensure the causality of the predictor computation.

The second step is to discretize the integral term, which is done according to the guidelines proposed in [24] for the single delay case. More precisely, the integral term is discretized with a trapezoidal rule and included into the dynamic controller:

$$\begin{cases} \dot{z}(t) = -az(t) + bKe^{A(\hat{\delta}_k + \hat{\tau}_k)}\left[x(t-\tau_1) + \dfrac{t_s}{2}\displaystyle\sum_{i=1}^{n_k} f((i-1)t_s) + f(it_s)\right] \\ u_k = z(t_k) \end{cases} \tag{6.11}$$

where $n_k \doteq (\hat{\delta}_k + \hat{\tau}_k)/t_s$, a and b are some positive design parameters, t_s is the sampling time set by the controller clock and n_k is the number of steps used to discretize the integral, which is time-varying. Note that we chose the controller sampling time as the integrand step, since it sets the buffering of the past estimated delays and controller outputs.

Finally, an upper bound $\dot{\bar{\tau}}_{thr}$ is set on the maximum estimated delay variation that can be used for the computation of $\delta(t)$ since the delay estimator may have some fast variations, especially when it is triggered by the alarm signal. The dynamics of the predictor horizon is then set with:

$$\dot{\hat{\delta}}(t) = \frac{\dot{\bar{\tau}}_k + \lambda(\hat{\tau}_k - \hat{\delta}(t))}{1 - \dot{\bar{\tau}}_k} \tag{6.12}$$

where:

$$\begin{cases} \dot{\bar{\tau}}_k = \dfrac{\hat{\tau}_k - \hat{\tau}_{k-1}}{t_s} & \text{if } \dot{\bar{\tau}} < \dot{\bar{\tau}}_{thr} \\ \dot{\bar{\tau}}_k = \dot{\bar{\tau}}_{thr} & \text{else} \end{cases}$$

The issues considered in this section are all critical for a safe implementation of the predictor with a time-varying horizon, especially to control a system over the network considered. They are summarized with the following result.

Result 6.5.1 *Using the delay estimation provided by the CUMSUM Kalman filter described in section 6.4, the proposed predictive control approach is set with the dynamic controller (6.11), where the predictor horizon is set by the dynamics (6.12) and the controller gain is such that the closed-loop matrix (6.8) is Hurwitz.*

6.6 Experimental and simulation results

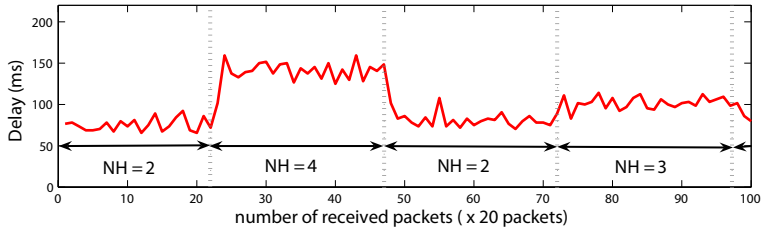
In this section, the network setup is presented along with some experimental results, which are used to estimate the efficiency of the proposed control setup using appropriate simulation tools. More details on the experimental setup and protocol specifications can be found in [3].

6.6.1 Communication protocol

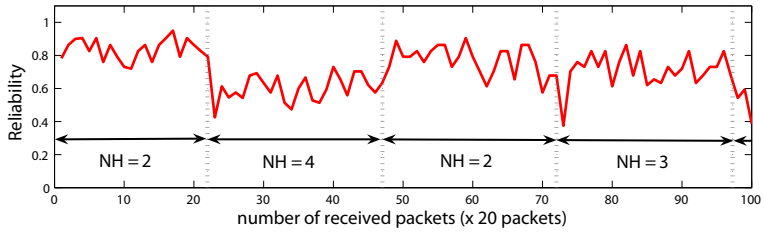
We use the Breath protocol proposed in [3] and briefly discussed in Section 6.3. Breath is designed for WSNs where source nodes attached to a plant must transmit information via multi-hop routing to a sink. Breath ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. The design approach relies on a constrained optimization problem, whereby the objective function is the energy consumption and the constraints are the packet delay and reliability. The optimal working point of the protocol is achieved by a simple algorithm, which adapts to traffic variations and channel conditions with negligible overhead. The protocol has been implemented and experimentally evaluated on a testbed with off-the-shelf wireless sensor nodes [25], and it has been compared with a IEEE 802.15.4 standard solution [26]. Analytical and experimental results show that Breath is tunable and meets delay and reliability requirements. Breath exhibits a good distribution of the working load, thus ensuring a long lifetime of the network. In addition, since this protocol does not include any specific error control mechanism that can ensure a reliable communication, we also introduce a stop-and-wait ARQ algorithm to provide for a more reliable data transfer service. ARQ combines error detection and retransmission to ensure that the data is delivered accurately to the user, despite the errors that may occur during transmission [7]. However there is packet loss due to the limited number of retransmissions with ARQ mechanism (2 in our experiment). In the following, we consider the Breath protocol without and with ARQ mechanism.

6.6.2 Network setup and experimental measurements

Our experiment is based on the Breath protocol using Tmote nodes [25], which are equipped with the radio controller Chipcon CC2420 operating at 2.4 GHz and supporting the IEEE 802.15.4 standard. The source is located at 30 m from the destination node, which is connected to the computer and generates 20 pckts/s. Eighteen nodes are placed at regular intervals (1.58 m) along a straight line without obstacles. Surrounding objects are static, with

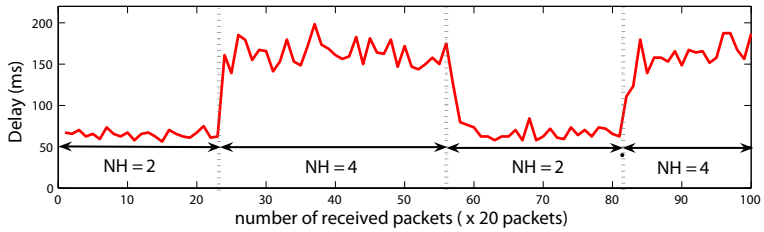


(a) Average end-to-end delay of Breath protocol

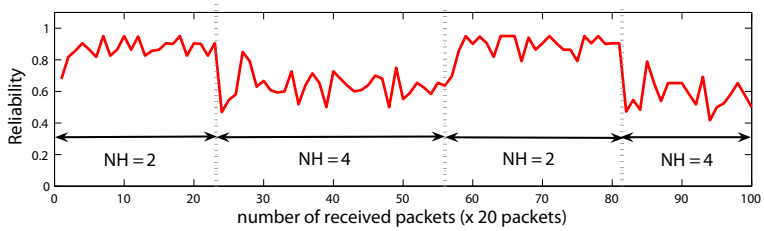


(b) Reliability of Breath protocol

Fig. 6.5 Average end-to-end network-induced delay and reliability of Breath protocol.



(a) Average end-to-end delay of Breath protocol with ARQ



(b) Reliability of Breath protocol with ARQ

Fig. 6.6 Average end-to-end network-induced delay and reliability of Breath protocol with ARQ.

minimal time-varying changes in the wireless channel due to multi-path fading effects. At each location the sensor node is placed in the same orientation to avoid non-linearities in the transmission pattern.

Figures 6.5 and 6.6 show the average end-to-end delay and reliability of the Breath protocol without and with ARQ, respectively. In the figure, note that "NH" refers to the number of hops of the network. We recall that the number of hops is set depending on the channel condition, traffic load and application requirements of the network. It is clear to observe that both the average end-to-end delay and reliability increase as the number of hops increases. In addition, ARQ mechanism increases the end-to-end delay because of the waiting time to receive an acknowledgement. We remark here that there is tradeoff between the packet delay and reliability [2]. Furthermore, increasing the number of hops gives an outstandingly bad reliability if more than four hops are used.

6.6.3 Physical system and control setup

The closed-loop system is presented as a block diagram in Figure 6.7, which depicts the plant, network, controller, and their interconnection. As an example, the linear system considered is similar to the T-shape ECP inverted pendulum presented in [21], an unstable open-loop system with no-minimum phase. Its dynamics is set with the state-space matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.80 & 0 & 12.56 & 0 \\ 0 & 0 & 0 & 1 \\ -2.42 & 0 & -8.33 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 4.57 \\ 0 \\ 0.38 \end{bmatrix}$$

The eigenvalues of A are $\lambda_{1,2} = 0.74 \pm 2.08i$, $\lambda_{3,4} = -0.74 \pm 2.08i$. The choice of an unstable open-loop plant emphasizes the efficiency of the controller to compensate for the transmission perturbations and the performance limitations.

Experimental delay measurements provide for τ_1 and τ_2 by splitting the initial measured data into two sets, one every other sample being attributed to each set. This means that the channels are not symmetric but belong to the same network (same global configuration). Only the last received packets are considered and the packet losses are handled as described in (6.3). The bandwidth limitation is set by zero-order hold with a $50\,ms$ sampling time. The controller includes:

- the time-delay estimator (based on end-to-end delay measurements);
- a reference trajectory block that sets the control objective;
- the predictive control setup.

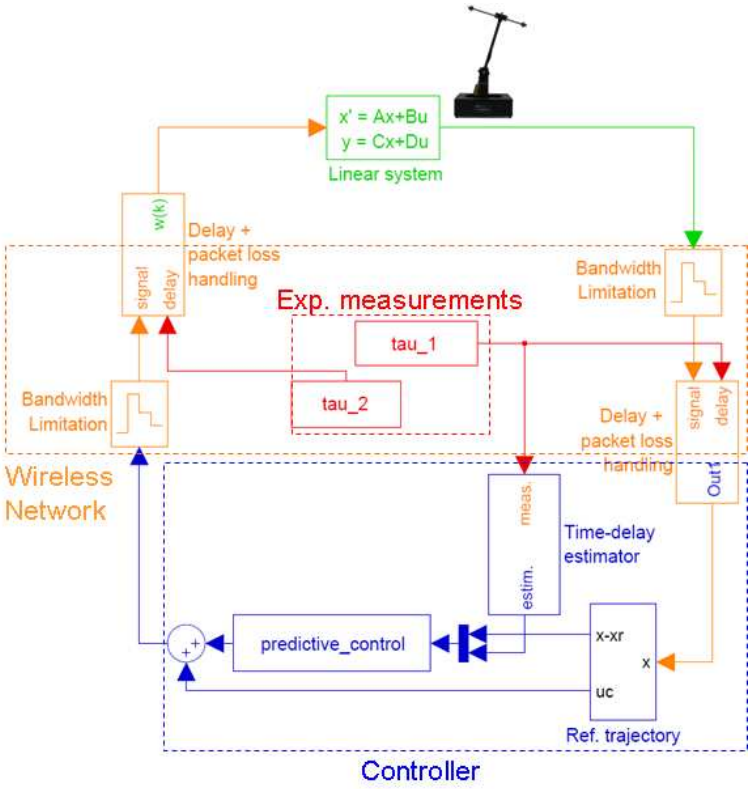


Fig. 6.7 Block diagram of the closed loop system with plant, network, measurements and controller.

The CUMSUM Kalman filter is set with $R = 10\ 000$, $Q = 1$, $\nu = 0.01$ and $g_{thr} = 35$. The reference trajectory, for the pendulum angle $x(3)$, is a filtered square signal. The predictive control is set with $\lambda = 10$, $a = b = 100$ (the introduced controller dynamics is equivalent to a low-pass filter) and $K = [0.48\ 0.36\ 2.20\ 0.57]$ sets the poles of (6.8) to $(-3 + 0.1i; -3 - 0.1i; -4; -4)$.

6.6.4 Simulation results

Figures 6.8 and 6.9 (without and with ARQ) present the actual and estimated time-delays (top), the reference and the system output (middle), and the control output (bottom). In both cases the transmissions are very noisy, in terms of jitter and packet losses (up to 60% of the signal). These figures illustrate the controller efficiency and the proper handling of the high frequency noise induced by the numerical integration problem.

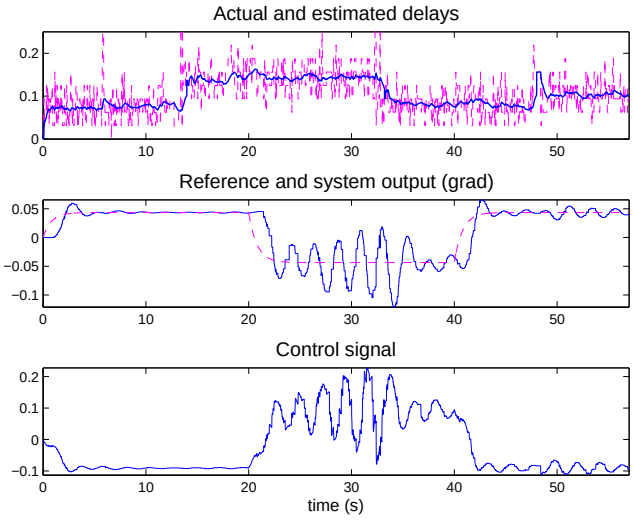


Fig. 6.8 Delays, system output and control signal without ARQ.

The effect of the number of hops on the system response clearly appears in both cases, particularly when a change in the reference sign occurs when there are four hops. Indeed Figure 6.8 depicts a situation where the bad quality of the communication channel introduces some oscillations, which are reduced as soon as the number of hops is set back to two. The benefit of ARQ appears at the beginning of the simulations but is discussable when the decrease in the average reliability is not compensated by the communication algorithm. Comparing the different simulation results, it appears that the system oscillations are induced by the low reliability rather than high end-to-end delay (i.e., comparing the different 4-hops cases). This is due to the fact that the control setup explicitly compensate the delay effects while the robustness with respect to packet losses is not specifically addressed in the feedback design.

These simulations also show that an increase of the number of hops has few effects on the system response when it is established in a steady state. A control-oriented multi-hop wireless network protocol could efficiently use this information to minimize the emission energy (increase the number of hops) according to some measurements of the physical state while ensuring satisfying performances.

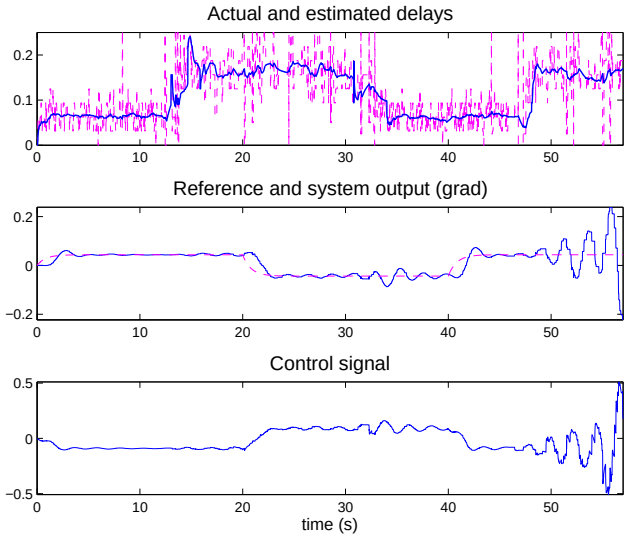


Fig. 6.9 Delays, system output and control signal with ARQ.

6.7 Conclusions

This chapter addressed the problem of remote control over a multi-hop wireless network. The constraints of such a network on the communication channel, such as time-varying delays (jitter), packet losses and bandwidth limitations are considered. The time-delay induced by the network is estimated thanks to a CUMSUM Kalman filter that provides for a fast tracking of the mean and good suppression of the high frequency variations of the delay (jitter). This estimator is used to design a dynamic predictive control law that compensates for the estimated part of the delay explicitly. The causality of this controller is set according to the estimator and specific considerations on safe implementation are detailed. A necessary condition on the assigned closed-loop spectrum is provided but the particular poles location is not specified, which provides for a degree of freedom for optimal or robust control approaches. Based on experimental delay and packet loss, some simulations illustrate the proposed control scheme and the choice of a communication protocol.

References

1. A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, pp. 1130–1151, June 2005.
2. P. Park, "Protocol design for control applications using wireless sensor networks," Licentiate thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2009.
3. P. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-adapting protocol for wireless sensor networks in control and automation," in *IEEE SECON*, 2008.
4. A. Lelevé, P. Fraisse, and P. Dauchez, "Telerobotics over IP networks: towards a low-level real-time architecture," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, Hawai, USA, 2001, pp. 643–648.
5. E. Witrant, C. Canudas-de-Wit, D. Georges, and M. Alamir, "On the use of state predictors in networked control systems," in *Applications of Time-Delay Systems*, ser. Lecture Notes in Control and Information Sciences, J. Chiasson and J.-J. Loiseau, Eds. Springer, 2007, vol. 352.
6. N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireles networks," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 58–71, 2004.
7. A. Leon-Garcia and I. Widjaja, *Counmmunication Netorks*. McGraw Hill, 2004.
8. K. Jacobsson, N. Möller, K. H. Johansson, and H. Hjalmarsson, "Some modeling and estimation issues in traffic control of heterogeneous networks," in *Proc. of the* 16^{th} *Int. Symp. on Math. Th. of Net. and Syst.*, Leuven, Belgium, Jul. 2004.
9. A. Seuret, M. Termens-Ballester, A. Toguyeni, S. E. Khattabi, and J.-P. Richard, "Implementation of an internet-controlled system under variable delays," in *Proc. of the IEEE Conf. on Emerging Technologies and Factory Automation*, Sep. 2006, pp. 675–680.
10. E. Witrant, P. Park, M. Johansson, C. Fischione, and K. Johansson, "Control over wireless multi-hop networks," in *IEEE Conference on Control Applications*, Singapore, Oct. 2007.
11. J. N. A. Karaki and A. E. Kamal, "Routing technques in wireless sensor networks: A survey," *IEEE Wireless Communication Magazine*, vol. 1, pp. 6–28, 2004.
12. J. V. Greuen, D. Petrović, A. Bonivento, J. Rabaey, K. Ramchandran, and A. Sangiovanni-Vincentelli, "Adaptive sleep discipline for energy conservation and robustness in dense sensor networks," *IEEE ICC*, vol. 1, 2004.
13. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," *ACM/IEEE MobiCom*, vol. 1, pp. 70–84, 2001.
14. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM/IEEE MobiCom*, vol. 1, 2001.
15. M. Kim and B. D. Noble, "Mobile network estimation." in *ACM/IEEE MobiCom*, 2001, pp. 298–309.
16. F. Gustafsson, *Adaptive Filtering and Change Detection*. Wiley, 2000.
17. A. W. Olbrot, "On controllability of linear systems with time delays in the control," *IEEE Transactions on Automatic Control*, vol. ac-16, pp. 664–666, 1972.
18. J. P. Richard, "Time delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, Oct. 2003.
19. A. Manitius and A. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE Transactions on Automatic Control*, vol. 24, pp. 541–552, Aug. 1979.
20. Z. Artstein, "Linear systems with delayed control: a reduction," *IEEE Transactions on Automatic Control*, vol. ac-27, no.4, pp. 869–879, Aug. 1982.
21. E. Witrant, C. Canudas-de-Wit, D. Georges, and M. Alamir, "Remote stabilization via time-varying communication network delays: Application to TCP networks," in *Proc. of the IEEE Conference on Control Applications*, Taipei, Taiwan, sep 2004.

22. E. Witrant, C. Canudas-de-Wit, and D. Georges, "Remote output stabilization under two channels time-varying delays," in *Proc. of the* 4^{th} *IFAC Workshop on Time Delay Systems*, Rocquencourt, France, sep 2003.
23. V. Van Assche, M. Dambrine, J.-F. Lafay, and J.-P. Richard, "Some problems arising in the implementation of distributed-delay control laws," in *Proc. of the* 38^{th} *Conference on Decision and Control*, Phoenix, Arizona (USA), Dec. 1999.
24. S. Mondié and W. Michiels, "Finite spectrum assignment of unstable time-delay systems with a safe implementation," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2207–2212, Dec. 2003.
25. *Tmote Sky Data Sheet*, Moteiv, San Francisco, CA, 2006. [Online]. Available: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf
26. *IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, 2006. [Online]. Available: http://www.ieee802.org