

Measurement Delay Estimation for Kalman Filter in Networked Control Systems

Mikael Pohjola and Heikki Koivo

Control Engineering Laboratory, Helsinki University of Technology,
 Espoo, Finland, (e-mail mikael.pohjola@tkk.fi, heikki.koivo@tkk.fi)

Abstract: This work presents a method to estimate an unknown varying delay of measurements and subsequent Kalman filtering in the networked control system (NCS) framework. The delay estimation algorithm is based on the Gaussian error model and a network delay model, either a probability distribution or a Markov-chain. This method is used to tackle the problem with varying delays in a NCS. The undelayed output of the plant is estimated with a Kalman filter and used for control. The probability of a wrong delay estimate is derived. The estimation and control performance is evaluated with simulations. From the control perspective, it has comparable performance to the case with known delays.

1. INTRODUCTION

In low-cost networked control systems (NCS) with commercial-off-the-shelf hardware or wireless sensor networks, the network induces a varying delay, because hard real-time requirements are not incorporated into Ethernet or wireless communication standards. The delay can be known or unknown, depending on time-stamping and clock synchronization of the nodes. In any case, the varying nature of the delay poses some problems to the control loop. Control theory has few tools to handle stochastically varying delay. The research has focused on optimal control (Lincoln and Bernhardsson, 2000) and stability (Zhang *et al.*, 2001).

There are two practical ways to tackle the varying delay control problem. The first is to keep the traditional control loop and tune the controller to be robust to the varying delay. Good results have been achieved with PID controller optimization (Pohjola, 2006), (Eriksson, 2007) and there are some theoretical bounds on stability (Cervin *et al.*, 2004). The other approach is to add an observer, which estimates the current output of the process based on the varying delayed measurements. A suitable observer is e.g. the Kalman filter.

In estimation with varying delayed measurements the Kalman filter (KF) has been applied in many situations where the delay is known. The convergence is proved with LMIs (Linear Matrix Inequalities) (Hespanha and Naghshtabrizi, 2006). Optimal Kalman filtering with varying measurement delay is treated in (Schenato, 2006), where the previous measurements, the state and the covariance estimates are stored in buffers and the filtering is done up to the current time every time a new measurement arrives. This is computationally heavy and the delay must be known. The paper also presents estimation with constant gain.

In (Xu and Hespanha, 2005) and (Schenato, 2006) a "smart sensor" is used. The filtering is done at the sensor and the state estimate is sent over the network. This ensures that the

estimation is optimal, since no measurements are lost, and the current state can be calculated by prediction. It has the downside that the control input to the plant has to be transmitted to the sensor without delay and loss, which is not practically achievable.

Few have studied the case with *unknown* varying measurement delay, which we will do here. In an NCS the measurement from a sensor is sent over a network (wired or wireless) which has a varying delay, Fig. 1. The varying delay can e.g. stem from other control loops communicating over the same network, but the delay is not known at the controller; it has to be estimated. The delay can change every time-step, so delay tracking is not possible. It has to be re-estimated for every received measurement packet. The measurement fusion block resides at the controller. It estimates the delay of the received measurement and performs Kalman filtering to fuse it with the state estimate. It also gives the current estimated plant output to the controller.

The proposed delay estimation algorithm is based on a Gaussian error model and a network delay model, either a probability distribution or a Markov-chain. The delay estimates are of maximum likelihood type and therefore optimal. The filtering is suboptimal, because it assumes correct delay estimates before the fusion.

The output estimation with varying delayed measurement has

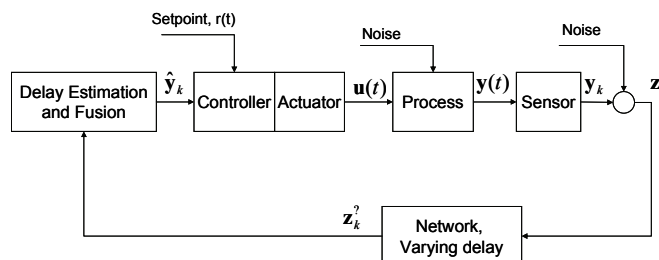


Fig. 1. NCS with delay estimation and fusion.

been treated in (Sanchis *et al.*, 2007), but there the delay is assumed to be known. There are also other Kalman filter based algorithms for varying delayed measurements, e.g. in (Goodwin *et al.*, 2002) the authors treat the delay estimation as errors in variables problem.

This paper summarizes the use of a Kalman filter with varying delayed measurements in the next section. It proposes a maximum likelihood algorithm in Section 3 for estimating the delay at every time-step of an unknown, variably delayed measurement. The probability of wrong delay estimation is also derived. Simulation results are presented in Section 4 and finally conclusions are drawn.

2. KALMAN FILTERING WITH VARYING DELAY

In this section Kalman filtering with variable delayed measurements is summarized. The expansion of the KF to accommodate measurement delay is similar to (Goodwin *et al.*, 2002). The assumed state-space process model with possible multiple inputs and multiple outputs is of the form

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{cases}, \quad (1)$$

where \mathbf{w}_k and \mathbf{v}_k are Gaussian white-noises with covariances \mathbf{Q} and \mathbf{R} . In the constant delay case the delay is usually incorporated into the state-space model, as a process delay.

Some modifications must be done to (1) to add the varying network delay to the model. The model used in the Kalman filter is augmented with delayed process output values such that the state-vector is of the form

$$\mathbf{x}'_k = [\mathbf{x}_k \quad \mathbf{y}_k \quad \cdots \quad \mathbf{y}_{k-d_{max}}]^T, \quad (2)$$

where \mathbf{y}_d , $d = [k, \dots, k-d_{max}]$ are the delayed true (without noise) outputs, and d_{max} is the maximum expected or allowed delay. If measurements are delayed more than this, they may be dropped, as they may not bring significant information to the current process state. A measurement \mathbf{z}_{k-d} received at time k and delayed d time-steps on transmission, is denoted $\mathbf{z}_k^d = \mathbf{z}_{k-d}$.

The modified \mathbf{A}' and \mathbf{B}' matrices become

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & \mathbf{0} & & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & & \\ & \mathbf{I} & \mathbf{0} & \\ & & \ddots & \ddots \\ \mathbf{0} & & & \mathbf{I} & \mathbf{0} \end{bmatrix} \text{ and } \mathbf{B}' = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \quad (3)$$

And n_c is the number of measurements. The measurement is not corrupted by noise as it is delayed in the network, therefore the process noise covariance is accordingly padded with zeros

$$\mathbf{Q}' = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (4)$$

When the Kalman filter update step is performed, a measurement matrix \mathbf{C}^f is used. The matrix changes depending on the delay of the received measurement, such that

$$\mathbf{C}_k^f(d_k) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \cdots \mathbf{0} & \mathbf{I} & \mathbf{0} \cdots \mathbf{0} \end{bmatrix}, \quad (5)$$

when the delay is d_k . The dimensions of the blocks in \mathbf{C}^f are indicated. Here f refers to filter and k to the current time-step. It is assumed that the network induces a delay of at least one time-step. This is natural, since any small transmission delay will be rounded up to the next computation cycle of the time-driven Kalman filter.

The Kalman filtering is done with matrixes \mathbf{A}' , \mathbf{B}' and $\mathbf{C}_k^f(d_k)$, according to the equations (prediction and update) found in Kalman filtering literature, such as (Maybeck, 1979). If no measurement is received on the current time-step, only the update part of the Kalman filter is done. When the estimated output is calculated a constant \mathbf{C}^e matrix can be used such that

$$\mathbf{C}^e(d^e) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \cdots \mathbf{0} & \mathbf{I} & \mathbf{0} \cdots \mathbf{0} \end{bmatrix}, \quad (6)$$

if a d^e delayed output estimate is desired.

After the prediction step of the KF, the measurement is fused with the measurement matrix $\mathbf{C}_k^f(d_k)$. If the measurement delay d_k is unknown, it can be estimated with the algorithm presented in the next chapter.

3. UNKNOWN DELAY ESTIMATION

In case of no time-stamping, or if the clocks of the sender and the receiver are not synchronized, the delay cannot be measured, e.g. d_k is unknown for the received measurements. The goal is to estimate the delay before the KF update step to fuse the measurement correctly.

By using the measurement \mathbf{z} , the true output \mathbf{y} , and the known or estimated delay distribution of the network, the likelihood of the delay can be estimated. When the delay is unknown, the maximum a posteriori delay, at step k , can be estimated by maximizing the conditional probability

$$p_k(d|\mathbf{y}) = \frac{p_k(\mathbf{y}|d)p_k(d)}{p_k(\mathbf{y})}. \quad (7)$$

The right part is obtained by the Bayes' Theorem. The numerator is a scaling factor and can be left out in the maximization. If Gaussian noise is assumed, the error $\mathbf{y}_{k-d} - \mathbf{z}_k^d$, where \mathbf{z}_k^d is a measurement taken at time k and delayed d steps, is Normally distributed. The probability $p_k(\mathbf{y}|d)$ is then proportional to

$$p_k(\mathbf{y}|d) \propto \exp\left(-\frac{1}{2}(\mathbf{y}_{k-d} - \mathbf{z}_k^d)^T \mathbf{R}^{-1}(\mathbf{y}_{k-d} - \mathbf{z}_k^d)\right), \quad (8)$$

where \mathbf{R} is the measurement covariance. The difference of

the true output and the measurement is assigned to the noise. If the difference of the true output delayed by some particular d and the measured output is large, the probability that the measurement is delayed d steps is small. Since d is not known the received measurement \mathbf{z}_k^d is denoted $\mathbf{z}_k^?$.

Instead of the true output \mathbf{y}_{k-d} , which is not available at the fusion block, the estimated delayed output $\hat{\mathbf{y}}_k^d$ can be used. Then the measurement covariance matrix is replaced with the residual covariance $\mathbf{S}_k(d) = \mathbf{C}_k^e(d) \mathbf{P}_{k|k-1} (\mathbf{C}_k^e(d))^T + \mathbf{R}_k(d)$ and

$$p_k(\mathbf{y}|d) \propto \exp\left(-\frac{1}{2}(\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?)^T \mathbf{S}_k^{-1}(d)(\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?)\right). \quad (9)$$

In case of a stationary delay distribution, the probability $p_k(d)$ is constant as a function of k and obtained from the known delay probability density function, $f(d)$ or cumulative distribution function $F(d)$

$$p_k(d) = p(d) = \int_{d-1}^d f(\tau) d\tau = F(d) - F(d-1). \quad (10)$$

This is because the probability that the delay is d time-steps, is equal to the probability mass that the measurement is delayed in the interval $]d-1\dots d]$. The values of $p(d)$ can be calculated in advance.

The maximum a posteriori delay estimate, d^* is then

$$d_k^* = \arg \max_{d=d_{\min}, \dots, d_{\max}} \exp\left(-\frac{1}{2}(\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?)^T \mathbf{S}_k^{-1}(d)(\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?)\right) p_k(d). \quad (11)$$

Note that only delay values from d_{\min} to d_{\max} need to be considered. This can be simplified for online calculations by taking the logarithm and manipulating. The resulting equivalent optimization problem is

$$d_k^* = \arg \min_{d=d_{\min}, \dots, d_{\max}} J_k(d) \quad (12)$$

$$J_k(d) = E(d) + L(d) = \left\| \hat{\mathbf{y}}_k^d - \mathbf{z}_k^? \right\|_{\mathbf{S}_k^{-1}(d)}^2 - 2 \ln(p_k(d))$$

The first term is the $\mathbf{S}_k^{-1}(d)$ -weighted 2-norm of the difference $\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?$ and $L(d)$ can be seen as a regularization term, stemming from the delay distribution. $L(d)$ can be calculated in advance. If the residual covariance is small, the estimation trusts the measurements and the delay that gives the smallest error is chosen. If the residual covariance is large the delay distribution has more weight and the most probable delay is chosen. The algorithm has thus the advantage to naturally incorporate the confidence of the estimator in the delay estimation.

The output of the process needs to change for the optimization to differentiate between the possible delays, otherwise the $E(d)$ term is constant. Note that the matrix inversion in (12) has the dimensions of the measurement vector, not the augmented state-vector.

This method is suboptimal, since it does not re-filter the measurements when more information arrives and a better delay estimate could be obtained. An approach would be to

use relaxed dynamic programming to re-estimate the delay when new information is obtained (Alriksson and Rantzer, 2006). It is however computationally heavier and increases memory requirements. Instead of using a Kalman filter, an output predictor, such as the one described in (Sanchis *et al.*, 2007) could be used, since the state estimate produced by the Kalman filter is not needed in the delay estimation algorithm.

The stability proofs for the case with known varying delay are already tedious. Proofs for the unknown delay case are left for future research. Next a derivation of the probability of a wrong delay estimate is presented.

3.1 Probability of wrong delay estimate

To assess the delay estimation reliability, the probability of a wrong estimate is calculated. We consider $\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?$ as a random variable in (12) and calculate the probability of

$$p\left(J(d^?) \geq \min_{d \neq d^?} J(d)\right), \quad (13)$$

which is the probability that the cost function J has its minimum for a delay other than the true delay, $d^?$. We assume that the Kalman filter has reached steady-state, i.e. the constant state covariance matrix \mathbf{P}_{ss} is the solution of the discrete-time algebraic Riccati equation

$$\mathbf{P} = \mathbf{A}' \left(\mathbf{P} - \mathbf{P} (\mathbf{C}^f)^T \left(\mathbf{C}^f \mathbf{P} (\mathbf{C}^f)^T + \mathbf{R} \right)^{-1} \mathbf{C}^f \mathbf{P} \right) \mathbf{A}^T + \mathbf{Q}. \quad (14)$$

Also the steady-state value of \mathbf{S} is obtained as

$$\mathbf{S}_{ss} = \mathbf{C}^f \mathbf{P}_{ss} (\mathbf{C}^f)^T + \mathbf{R}. \quad (15)$$

The next task is to find the probability distribution of $J(d)$. $\hat{\mathbf{y}}_k^d - \mathbf{z}_k^?$ is Normally distributed with mean $\mathbf{y}_\Delta(d_\Delta)$ and covariance \mathbf{S}_{ss} , where $\mathbf{y}_\Delta(d_\Delta)$ is the change of \mathbf{y} compared to the measurement \mathbf{z} as a function of difference of delay. Naturally $\mathbf{y}_\Delta(0) = 0$, since the expectation $E\{\hat{\mathbf{y}}_k^d - \mathbf{z}_k^d\} = 0$.

The quadratic form $\mathbf{x}^T \mathbf{W} \mathbf{x}$, where \mathbf{x} is a Normally distributed random vector with mean \mathbf{m} and covariance $\mathbf{\Sigma}$, and \mathbf{W} is a positive definite matrix, follows a noncentral chi-square distribution $\chi^2(t|\nu, c)$ if and only if (Tziritas, 1987)

$$tr\left((\mathbf{\Sigma} \mathbf{W})^3\right) tr(\mathbf{\Sigma} \mathbf{W}) = \left(tr\left((\mathbf{\Sigma} \mathbf{W})^2\right)\right)^2 \quad (16)$$

where tr is the trace. The parameters of χ^2 are ν degrees of freedom and noncentrality coefficient c :

$$\nu = \left(tr(\mathbf{\Sigma} \mathbf{W})\right)^2 / tr\left((\mathbf{\Sigma} \mathbf{W})^2\right) \quad (17)$$

$$c = tr(\mathbf{\Sigma} \mathbf{W}) / tr\left((\mathbf{\Sigma} \mathbf{W})^2\right) \mathbf{m}^T \mathbf{W} \mathbf{m}. \quad (18)$$

The quadratic form $\left\| \hat{\mathbf{y}}_k^d - \mathbf{z}_k^? \right\|_{\mathbf{S}_{ss}^{-1}(d)}^2$ satisfies the condition (16), since $\mathbf{\Sigma} = \mathbf{S}_{ss}$ and $\mathbf{W} = \mathbf{S}_{ss}^{-1}$, $\mathbf{\Sigma} \mathbf{W} = \mathbf{I}_{n_c}$ and the traces reduces to n_c . The probability distribution $p_J(t|d)$ of J is thus the shifted (because of the regularization term $L(d)$) noncentral chi-square distribution

$$p_J(t|d) = \chi^2 \left(t + 2 \ln(p_k(d)) \middle| n_c, \mathbf{y}_\Delta(d_\Delta)^T \mathbf{S}_{ss}^{-1} \mathbf{y}_\Delta(d_\Delta) \right). \quad (19)$$

The probability that $J(d^2) \geq J(d)$ is computed with

$$p(J(d^2) \geq J(d)) = \int_{-\infty}^{\infty} p_J(t|d^2) p_J(d < t|d) dt. \quad (20)$$

$p_J(d < t|d)$ is the cumulative distribution function of (19).

3.2 Markov delay model

Instead of using a static delay distribution as a model for the network, a Markov-chain can be used to model correlated network delay. One suitable Markov-chain model is here briefly presented. It has $N = d_{max} - d_{min}$ states, each corresponding to a delay value. The delay is assumed to change at most by one time-step at a time. The Markov chain state-transition matrix is then of the death-birth form

$$P_{delay} = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & 0 \\ p_{21} & p_{22} & p_{32} & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & p_{N,N-1} & p_{NN} \end{bmatrix}. \quad (21)$$

Other chain types are naturally also possible with the presented delay estimation algorithm. Another assumption taken is that the average delay distribution follows the Gamma distribution, since it is typical for computer networks (Mukherjee, 1994). The task is now to find the transition probabilities p_{ij} , so that the delay model meets these criteria.

There are $3N-2$ unknowns. The total transition probabilities of the Markov-chain and the desired average state probability distribution $\Pi = [\pi_{d_{min}} \dots \pi_{d_{max}}]$ give $2N$ equations

$$\sum_j p_{ij} = 1 \text{ and } \sum_i p_{ij} \pi_i = \pi_j. \quad (22)$$

π_j is calculated by integrating the gamma probability density function over the intervals $]d-1..d]$, $d \in [d_{min}, \dots, d_{max}]$

$$\int_{d-1}^d p_{\text{gamma}}(x) dx = \int_{d-1}^d \frac{\alpha}{\Gamma(n)} (\alpha x)^{n-1} e^{-\alpha x} dx, \quad (23)$$

where $P_{\text{gamma}}(x)$ is the incomplete gamma function and Γ is the gamma function

$$\Gamma(n) = \int_0^{\infty} x^{n-1} e^{-x} dx. \quad (24)$$

The average of the gamma distribution is $\bar{d} = n/\alpha$. Due to the truncation of the gamma distribution, the integration limits of the first and the last integrals ($d = d_{min}$ and $d = d_{max}$) are exchanged to 0 and infinity, respectively.

The last $N-2$ equations are obtained by specifying a probability of the delay to change:

$$\begin{cases} p_{21} + p_{23} = c \\ \vdots \\ p_{N-1,N-2} + p_{N-1,N} = c \end{cases}, \quad (25)$$

where c is the change probability $]0..1[$. The change parameter specifies how fast the delay changes. With smaller c the probability to change is little and the delay estimation algorithm should be able to follow the delay changes easier.

Solving these linear equations in p_{ij} results in the desired Markov delay model with desired delay distribution and maximum change rate of 1 step/time-step. To avoid getting stuck with a wrong delay estimate, a small positive constant should be added to every element in the transition matrix. The final Markov chain is

$$P_{Markov} = (P_{delay} + \varepsilon) / (N\varepsilon). \quad (26)$$

When using a Markov chain delay model the delay probabilities in (12) change at every step depending on the estimated delay, so that $p_k(d)$ is the $(d_{k-1}^* - d_{min})^{th}$ row in P_{Markov} . Other Markov-chain delay models could also be used. It could e.g. depend on the congestion of the network.

4. SIMULATION RESULTS

The performance of the delay estimation and filtering is investigated with simulations of two processes. Process 1 (P1) is given in discrete-time as the state-space representation

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.9 & 0 \\ 0.2 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k \quad (27)$$

$$\mathbf{y}_k = \mathbf{x}_k$$

with sample time $h = 0.1$ s. It is controlled by state-feedback, placing both poles at 0.5. The second process (P2) is a continuous-time first-order process

$$G_m(s) = \frac{1}{s+1}. \quad (28)$$

The process model used in the delay estimation is the discretized version of (28) with the sample time h . It is controlled by a discrete-time PID controller of the form:

$$u_{PID}(k) = K_p \left(1 + \frac{h}{T_i(hz-1)} + \frac{T_d N_d (z-1)}{(T_d + N_d h) z - T_d} \right) e(k). \quad (29)$$

The processes are measured with a sensor with a sample time of h . The measurement is transmitted over a simulated network with unknown varying delay. The network model is simulated as the Markov-chain model derived in Section 3.2 with parameters $\alpha = 1$, $n = 2$ (with a mean delay of $\bar{d} = 2$), $d_{min} = 1$, $d_{max} = 6$ ($N = 6$), $c = 0.5$ and $\varepsilon = 0.001/N$. The proposed delay estimation and fusion block resides at the controller and calculates output estimates of the process.

Several fusion variations are compared in simulations:

1. A Kalman filter with a model (2)-(5) that assumes a constant delay $d_k = \bar{d}$, the mean of the network delay, a heuristic to ignore the varying delay.
2. Two Kalman filters with delay estimation as explained in Section 3:
 - a. one with a delay distribution (gamma distribution with parameters $\alpha = 2$, $n = 6$)

- b. one with the Markov-chain of Section 3.2.
- 3. The same Kalman filter as in 2., but with known delay. The delay is obtained by time-stamping and time-synchronization of the sender and receiver.

The same delay realization is used for all the methods to reduce the variance when comparing the methods. All the Kalman filters have diagonal process (**Q**) and measurement (**R**) covariance matrixes with 0.03^2 and 0.01^2 on the respective diagonals. The output of the Kalman filter for P1 is the current state estimate, and for P2 the two-steps delayed estimated output, i.e. $d^e=2$ in (6).

The PID controller is tuned by optimizing the ITSE cost (35) with a constant feedback delay of $2h$, i.e. without the unknown varying delay. The optimal parameters are $K_p = 3.26$, $T_i = 1.17$, $T_d = 0.08$ and a derivative filter $N_d = 10$.

As the delay estimation depends on a change in the output, a step (unit steps every 4 seconds) and a sine reference with amplitude 1, frequency $\pi/2$ rad/s) is used. The step reference emulates the case with periods of steady-state when the delay estimation cannot be performed reliably and the sine case with a constantly changing output.

The integral error costs

$$J_e = \frac{1}{T/h} \sum_{k=1}^{T/h} (\hat{y}_k - y_k)^2 \quad (30) \quad D_{\%} = \frac{100}{T/h} \sum_{k=1}^{T/h} (d_k^* = d_k) \quad (31)$$

$$J_D = \frac{1}{T/h} \sum_{k=1}^{T/h} |d_k^* - d_k| \quad (32) \quad \sigma_D^2 = \frac{1}{T/h} \sum_{k=1}^{T/h} (d_k^* - d_k)^2 \quad (33)$$

$$J_{ISE} = \frac{1}{I} \sum_{i=1}^I \int_0^{\infty} (r^i(t) - y^i(t))^2 dt \quad (34)$$

$$J_{ITSE} = \frac{1}{I} \sum_{i=1}^I \int_0^{\infty} t (r^i(t) - y^i(t))^2 dt \quad (35)$$

are used to evaluate the delay, estimation and control performance of the proposed methods. J_e is the squared output estimation error, $D_{\%}$ the percentage of correct delay estimates, J_D and σ_D^2 the mean and variance of the error in delay estimation in units of time-steps. The control performance is measured with J_{ISE} and J_{ITSE} , which are the mean step response ISE and ITSE costs, respectively, over $I = 250$ runs ($y^i(t)$ refers to the process output of the i^{th} run) when using the different methods 1-3 as estimators. The results of $T = 1000$ s long simulations are collected in the following tables. Table 1 tabulates the delay estimation performance. The output estimation and control performance

Table 1. Delay estimation results for P1 and P2.

Method, reference	$D_{\%}$		J_D		σ_D^2	
	P1	P2	P1	P2	P1	P2
1	33		1.02		1.86	
2a, sine	53	77	0.65	0.30	1.07	0.53
2b, sine	52	78	0.68	0.28	1.10	0.49
2a, step	44	36	1.07	1.19	2.61	2.61
2b, step	44	36	1.07	1.18	2.54	2.94
3	100		0		0	

Table 2. Estimation and control performance of P1.

Method, P1	Sine reference		Step reference	
	J_e	J_{ISE}	J_e	J_{ITSE}
1	0.103	1.89	0.069	1.65
2a	0.084	1.14	0.038	0.268
2b	0.089	1.17	0.039	0.263
3	0.011	0.97	0.011	0.178

for P1 are in Table 2 and for P2 in Table 3.

The plots in Fig. 2 show some histograms of the estimated delay versus the true delay. The areas of the squares are proportional to the number of estimates. In the ideal case there would only be squares on the diagonal.

The theoretical probability of a wrong delay estimation of P2 is calculated. J depends, among other things, on the change of y and the sensor noise **R**. y_{Δ} is modeled to be linear with respect to the delay difference $y_{\Delta}(d_{\Delta}) = d_{\Delta} = (d^e - d)\Delta y$, i.e. the case of a linearly increasing response. In Fig. 3 (13) is illustrated as a function of y_{Δ} (with fixed **R**, the same as in the simulations) and as a function of noise covariance **R** (with fixed $\Delta y = [0.025 \ 0.025]^T$). The examples are for the case of the delay $d^e = 2$.

Examining Table 1 shows that the delay estimation performs better with the sine reference (52-78 % are estimated correctly) compared to the step response (36-44 %). This indicates, as expected, that the output must change constantly for the algorithm to be able to estimate the delay correctly. In the case of the step response the estimator relies more on the delay distribution than on the measurements and hence the squares in Fig. 2 are concentrated on the average delay.

The output estimation results shown in Table 2 and Table 3 are good for both step and sine references, even if the delay estimation with the step response is worse. This can be explained because at steady-state the output is the same regardless of the delay. The estimation error cost, J_e , is a fraction of the error of the naive case 1. The estimate in this case can be far off if the actual delay deviates from the mean.

The control performances with delay estimation are almost equally good as the optimal case with known delay (case 3) and the constant delay assumptions give considerably poorer results. The differences in control performances are small because the closed loop control naturally rejects disturbances, such as estimation errors. Moreover, wrong delay estimates at steady-state with the step reference, do not impact the control performance notably. With both delay estimation variants the estimation and control performance are near or equal to the optimal case with known delay.

Table 3. Estimation and control performance of P2.

Method, P2	Sine reference		Step reference	
	J_e	J_{ISE}	J_e	J_{ITSE}
1	0.039	0.146	0.0147	0.0225
2a	0.0015	0.103	0.0011	0.0093
2b	0.0015	0.103	0.0011	0.0091
3	0.0010	0.103	0.0010	0.0091

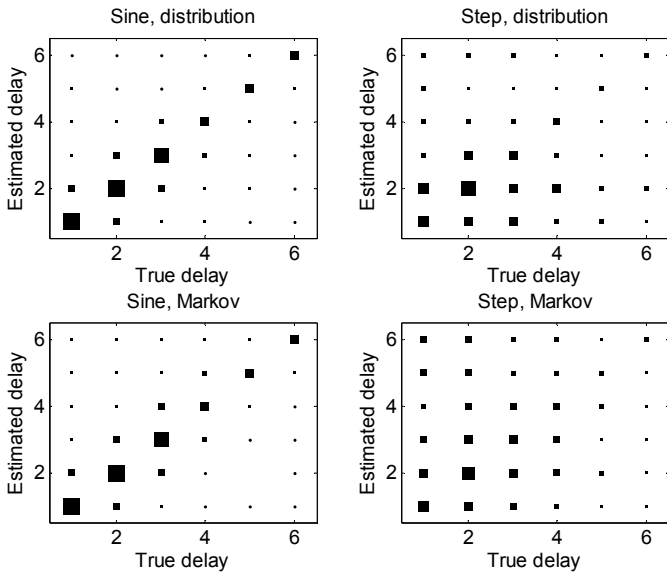


Fig. 2. Estimated delay versus true delay for sine/step reference and distribution/Markov chain delay model with P2.

4. CONCLUSIONS

Kalman filtering with unknown varying delayed measurements was presented. Such situations are common in networked control systems, where the measurements are transmitted over a wireless network. The Kalman filter was used as an observer to estimate the current plant output of the NCS. A delay estimator was proposed, based on the Gaussian error model of the estimated output and a network delay model, either a delay distribution or a Markov-chain delay model. The algorithm naturally takes the estimation confidence into account when estimating the delay. The probability of a wrong delay estimate was derived and some cases were illustrated.

The algorithm is applicable when the delay is unknown and may change at every time-step. The delay model must be known or estimated in advance. The algorithm achieves comparable performance (both from estimation and control point of view) compared to the optimal case with known measurement delay. Even incorrect estimation of the delay at periods with constant output, does not affect the performance of the control system. On the other hand, assuming a constant delay was shown to perform badly. The estimation of delay thus improves the control performance significantly.

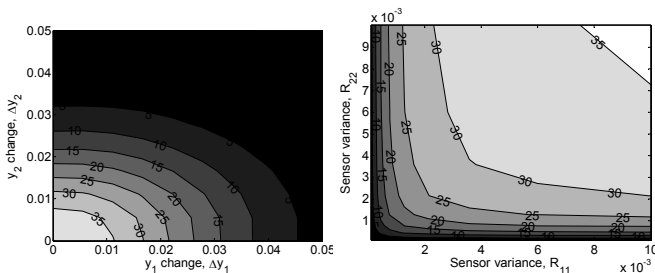


Fig. 3. Probability of wrong delay estimate as a function of change in y (left) and function of noise variance, \mathbf{R} (right).

REFERENCES

Alriksson, P. and A. Rantzer (2006). Observer synthesis for switched discrete-time linear systems using relaxed dynamic programming. In *Proc. 7th International Symposium on Mathematical Theory of Networks and Systems*. Kyoto, Japan.

Cervin, A., B. Lincoln, J. Eker, K-E Årzén, and G. Buttazzo (2004). The jitter margin and its application in the design of real-time control systems. In *Proc. 10th International Conference on Real-Time and Embedded Computing Systems and Applications*. Göteborg, Sweden.

Eriksson, L. M. and M. Johansson (2007). Simple PID Tuning Rules for Varying Time-Delay Systems. To Appear in *Proc. The 46th IEEE Conference on Decision and Control*. New Orleans, LA, USA.

Goodwin, G. C., J. C. Agüero, and Arie Feuer (2002). State estimation for systems having random measurement delays using errors in variables. In *Proc. 15th Triennial IFAC World Congress*. Barcelona, Spain.

Hespanha, J., P. Naghshtabrizi, and Y. Xu (2006). A Survey of Recent Results in Networked Control Systems. In *Proc. of the IEEE Special Issue on Networked Control Systems Technology*, **95**, No 1, 138-162.

Lincoln, B., and B. Bernhardsson (2000). Optimal control over networks with long random delays. In *Proc. 14th International Symposium on Mathematical Theory of Networks and Systems*. Laboratoire de Théorie des Systèmes (LTS), University of Perpignan.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, **1**, pp. 229–230. Academic Press.

Mukherjee, A. (1994). On the dynamics and significance of low frequency components of the internet load. *Internetworking: Research and Experience*, **5**, no. 4, pp. 163–205.

Sanchis, R., I. Peñarrocha, and P. Albertos (2007). Design of robust output predictors under scarce measurements with time-varying delays. *Automatica*, no. 43, pp. 281-289.

Schenato, L. (2006). Optimal estimation in networked control systems subject to random delay and packet drop. In *Proc. 45th IEEE Conference on Decision and Control*. San Diego, CA, USA.

Xu, Y., and J.P. Hespanha (2005). Estimation under uncontrolled and controlled communications in networked control systems. In *Proc. 44th IEEE Conference on Decision and Control, and the European Control Conference*. Seville, Spain.

Zhang, W., M.S. Branicky, and S.M. Phillips (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, **21**, issue 1.

Ziritas, G.G. (1987). On the distribution of positive-definite Gaussian quadratic forms. *IEEE Transactions on information theory*, **33**, no. 6.