THE EFFECT OF RANDOMLY TIME-VARYING SAMPLING AND COMPUTATIONAL DELAY

Martin Sanfridson, Martin Törngren and Jan Wikander

Mechatronics Laboratory
Royal Institute of Technology, SE-100 44 Stockholm, Sweden.
E- mail: {mis, martin, jan}@md.kth.se

Abstract: A randomly time-varying sampling period or control delay affects the performance, robustness and stability of a control loop. The variation in sensing and actuation instants is caused by e.g. multitasking, interactions in distributed systems and varying processing time. The discretization of a continuous-time quadratic loss function and the process is here extended to the case of aperiodic systems, when the control signal exhibits multiple changes during a non-uniform sampling period. *Copyright © 2005 IFAC*.

Keywords: sampled data control, time delay, jitter, discretization, synchronization.

1. INTRODUCTION

The assumption of equidistant sampling and constant control delay is a simplification typically made prior to the design of a discrete controller, applying traditional synthesis methods. Depending on the underlying computer system this might not be an appropriate approximation, or alternatively, it might mislead the designer to over-constrain the real-time behaviour of the computer control system. Hence, it is valuable to assess the effects of time-varying sampling and control delay. The design of a computer control system involves deciding upon the architecture: partitioning, allocation, real-time scheduling, etc. The result of these activities will affect the *timing behaviour* of the controller in terms of *sampling period* and *control delay*. Many computer control systems, for example embedded in vehicles and manufacturing equipment, are distributed having communication channels. The channels convey signals through layers of device drivers, communication protocols and application software, resulting in inherently time-varying *end-to-end* latencies. *Jitter* is here defined as unintentional or intentional time-variations compared to some nominal specification of a time interval. The jitter is usually an unintentional by-product and can from a control viewpoint be seen as a disturbance, compared to the nominal model used for analysis and synthesis.

In this paper a weighted *quadratic measure* applicable for analysing the timing behaviour, including jitter, is proposed. It can also be used for robustness analysis. The main contribution of this paper is the zero-order-hold discretization of the process and weight matrix allowing for non-periodic *multiple changes* in the control signal during one (time-varying) sampling period.

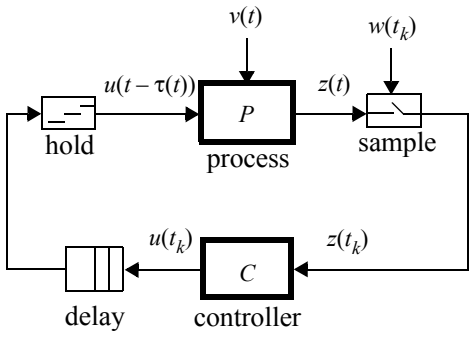

Fig. 1 The closed loop with sample and hold blocks, and with a delay directly after the controller.

In order to compute the loss function, there is a need to model the timing behaviour of the discrete-time controller, to discretize the process model and also the loss function. The process and controller together give the covariance of the closed loop. Finally, this will be combined with the discrete weight matrix to yield the sought measure of deterioration. A few alternative formulations and techniques will be pointed out, such as lifting and linear matrix inequalities. An example is given to show the application of the derived measure. As an application to the theory, the interesting question what type of jitter causes the worst-case performance, is studied by means of a numerical example. The paper is concluded by a note of related work and some final remarks. A summary of the notation is found in the end of the paper. More details and examples are found in (Sanfridson, 2004).

2. SYSTEM MODEL

In this section the timing behaviour and the dynamics of the studied system will be described. The following analysis is not limited to any specific system architecture or scheduling policy. The text book's uniform sampling and constant delay falls out as a special case.

2.1 Timing model, assumptions

Fig. 1 depicts a closed loop with the delay situated after the controller and before the hold-circuit. Let the signals $z(t_k)$, $v(t_k)$, $w(t_k)$ and $u(t_k)$ arise at the same time instant $t_k \in \Re$, corresponding to an integer-valued instance $k \in \{\ldots, -1, 0, 1, \ldots\}$. The time t_k is strictly increasing, $t_k > t_l, \ \forall k > l$. An assumption made here is that an old control signal $u(t_{k-a})$ cannot arrive at the process later than an already arrived newer one, $u(t_{k-b})$ with positive integers $b < a$. It makes sense that old unused samples are overwritten, so called *sample rejection*.

2.2 Timing model, example

Timelines are convenient to describe real-time behaviour. In Fig. 2, there is a pair of timelines: one for the sampling $z(t_k)$ and one for the actuation $u(t_k)$.

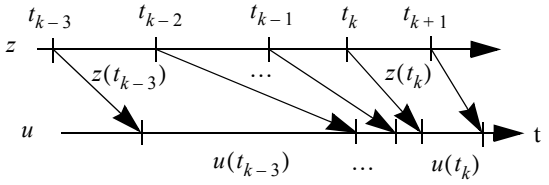

Fig. 2 Non-uniform sampling and actuation events.

A sampling period is the time interval between two consecutive sampling instances. The continuous measurement signal is in the general case not sampled uniformly, but displays jitter at the input to the controller. The time-varying sampling instances give rise to sampling period jitter, when compared to equidistant sampling. Similarly, the output instances $u(t_k)$ exhibit, in the general case, jitter at the output to the process, i.e. at the actuator. The control (or processing) delay is the time interval between a sampling and its corresponding output. The output jitter can be interpreted as a combination of sampling jitter and delay jitter.

During the period $[t_k, t_{k+1}]$ depicted in Fig. 2 the control signal u changes three times in a burst, preceded by a corresponding calm period with an ageing $u(t_{k-3})$ spanning over multiple sampling instances.

The example of a general but bursty timeline in Fig. 2 could arise due to low priority scheduled calls to an A/D unit, followed by the use of buffers for message delivery, and given an event triggered synchronization policy for the controller and actuator downstream. Despite varying latency by interference from higher priority tasks and messages, it is simple to enforce the same sequence of sending and delivery (FIFO-order) by using a counter, thus implementing sample rejection.

2.3 Continuous time process model

The process is LTI and strictly proper. When affected by a time-varying delayed control signal the state equation can be written

$$dx_p(t) = A_p x_p(t)dt + B_p u(t - \tau(t))dt + dv(t) \qquad (1)$$

with state vector x_p. The continuous time state noise v has a zero mean value and incremental covariance $\bar{R}_v dt$. Further, the increments are uncorrelated. The output equation can be written $y(t_k) = C_p x_p(t_k)$ and the measurement equation $z(t_k) = M_p x_p(t_k) + w(t_k)$ is at the sampling instant t_k corrupted by discrete time white noise w of zero mean and covariance R_w. The constant matrices $\{A_p, B_p, C_p, M_p\}$ have the sizes $[np \times np]$, $[np \times nu]$, $[ny \times np]$ and $[nz \times np]$ respectively.

2.4 A given discrete time controller

The linear discrete time controller is given by

$$\begin{aligned} x_c(t_{k+1}) &= \Phi_c x_c(t_k) + \Gamma_c z(t_k) + \Lambda_c U(t_k) \\ u(t_k) &= C_c x_c(t_k) + D_c z(t_k) + E_c U(t_k) \end{aligned} \qquad (2)$$

where $x_c(t_k)$ is the state of size $[nc \times 1]$ and $\{\Phi_c, \Gamma_c, \Lambda_c, C_c, D_c, E_c\}$ are constant matrices of the sizes $[nc \times nc]$, $[nc \times nz]$, $[nc \times (md \cdot nu)]$, $[nu \times nc]$, $[nu \times nz]$ and $[nu \times (md \cdot nu)]$ respectively (md is explained in the next section). The discrete delayed control output vector

$$U(t_k) = \left[u(t_{k-md}) \ \dots \ u(t_{k-1})\right]^T \tag{3}$$

has the size $[(md \cdot nu) \times 1]$ and is updated by

$$U(t_{k+1}) = \begin{bmatrix} 0 & I_{nu(md-1)} \\ 0 & 0 \end{bmatrix} U(t_k) + \begin{bmatrix} 0 \\ I_{nu} \end{bmatrix} u(t_k) \tag{4}$$

when time progresses from t_k to t_{k+1}. The time interval $h_k = [t_k, t_{k+1}]$ between any two rows is in the general case not constant.

2.5 Loss function

Since the vector field is described by a retarded differential equation (1), it makes sense to define a loss function similar to the familiar one commonly used in optimal control,

$$\overline{J} = E \lim_{T \to \infty} \frac{1}{T} \int_0^T \begin{bmatrix} x_p(t) \\ u(t-\tau(t)) \end{bmatrix}^T \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} \\ \overline{Q}_{12}^T & \overline{Q}_{22} \end{bmatrix} \begin{bmatrix} x_p(t) \\ u(t-\tau(t)) \end{bmatrix} dt \tag{5}$$

over some time period T (the same token is also used to denote matrix transpose), with $\overline{Q}_{11} \geq 0$ (positive semi-definite), $\overline{Q}_{22} \geq 0$ and in total $\overline{Q} > 0$. A discrete time equivalent during *one* sampling period is, with $Q > 0$, expressed by

$$J_x(t_k) = E \begin{bmatrix} x_p(t_k) \\ U(t_k) \\ u(t_k) \end{bmatrix}^T Q(t_k) \begin{bmatrix} x_p(t_k) \\ U(t_k) \\ u(t_k) \end{bmatrix} + J_v(t_k) \tag{6}$$

where J_v captures the intersample noise due to v.

3. DISCRETIZATION OF THE PROCESS

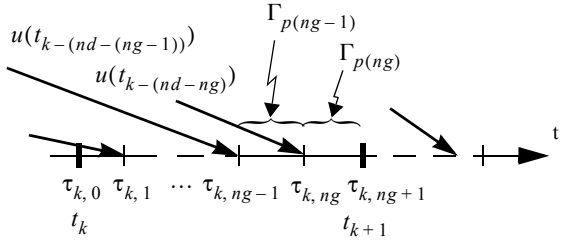

Fig. 3 Arrival of control signals.

The case without delay or with a constant delay is covered in text books, e.g. (Åström and Wittenmark, 1997). With fractional delays, the fact that the control signal is piecewise constant is used. The same concept is used here, and with multiple changes of the control signal, there will be several input matrices. The solution to (1) with a zero initial value,

$$x_p(t) = e^{A_p(t-t_k)} x_p(t_k) \tag{7}$$

$$+ \int_{t_k}^{t} e^{A_p(t-s)} u(s-\tau(t)) ds B_p + \int_{t_k}^{t} e^{A_p(t-s)} dv(s)$$

admits an exact zero-order-hold discretization without approximations. The equivalent process in discrete time is

$$x_p(t_{k+1}) = \Phi_p(t_{k+1}, t_k) x_p(t_k) \tag{8}$$

$$+ \Gamma_{pa}(T_k) u(t_k) + \Gamma_{pb}(T_k) U(t_k) + v(t_{k+1}, t_k)$$

with constant matrices $\{\Phi_p, \Gamma_{pa}, \Gamma_{pb}, M_p\}$ of appropriate sizes: $[np \times np]$, $[np \times nu]$, $[np \times (nd \cdot nu)]$, and $[nz \times np]$ respectively.

Let T_k be a sorted set of offsets or *fractional delays* $\tau_{k,g}$. The offset $\tau_{k,g}$ is the time interval from t_k until $u(t_{k-(nd-g)})$ switches in, with a second subindex $g = 0, 1, \dots, ng+1$. The elements of T_k form a monotonically increasing sequence,

$$0 = \tau_{k,0} \leq \tau_{k,1} \leq \dots \leq \tau_{k,ng} \leq \tau_{k,ng+1} = h_k. \tag{9}$$

The switching of the piecewise constant control signal during an interval of the length $h_k = t_{k+1} - t_k$ is illustrated in Fig. 3.

The relation of the triple $\{md, nd, ng\}$ is $md \geq nd(k) \geq ng(k) \geq 0$ for some k and $md = max(nd(k))$, $\forall k$. The purpose of md is to force matrices to have conform sizes and not depend on k. The relation is sketched geometrically in Fig. 4, with $a, b, c, d \in \{\dots, -1, 0, 1, \dots\}$. Note that any instance c, which corresponds to the newest data used to calculate the output during the studied interval, is uniquely defined, and that c and d do not necessarily coincide.

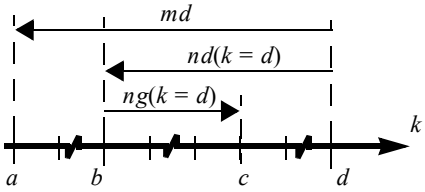

Fig. 4 Relation of the triple $\{md, nd, ng\}$.

Equation (8) can also be written

$$x_p(t_{k+1}) = \Phi_p(h_k)x_p(t_k) + \sum_{g=0}^{ng} \Gamma_{pg}(h_k)u(t_{k-(nd-g)}) + v(t_k) \quad (10)$$

which shows the construction of the tuple $\{\Gamma_{pa}, \Gamma_{pb}\}$ and $U(t_k)$. The system and input matrices at instance k are found by

$$\Phi_p(h_k) = e^{A_p(t_{k+1}-t_k)} \quad (11a)$$

$$\Gamma_{pg}(t) = \int_{\tau_{k,g}}^{\tau_{k,g+1}} e^{A_p(t-t_k-s)}B_p\,ds \quad (11b)$$

for some g. The covariance of the state noise becomes

$$R_v(t_{k+1}, t_k) = E[v(t_{k+1}, t_k)v^T] = \int_0^{h_k} e^{As}\overline{R}_v e^{A^T s}ds \quad (12)$$

since the last term of (7) does not correlate with x_p or u. The state noise does not depend on the delay.

4. CLOSING THE LOOP

By inserting the controller (2) into (8), the closed loop becomes $x(t_{k+1}) = \Phi_{cl}x(t_k) + \Gamma_{cl}e(t_k)$, with

$$\Phi_{cl} = \begin{bmatrix} \Phi_p + \Gamma_{pa}D_cM_p & \Gamma_{pa}E_c + \Gamma_{pb} & \Gamma_{pa}C_c \\ \begin{bmatrix} 0 \\ D_cM_p \end{bmatrix} & \begin{bmatrix} 0 & I \\ 0 & E_c \end{bmatrix} & \begin{bmatrix} 0 \\ C_c \end{bmatrix} \\ \Gamma_cM_p & \Lambda_c & \Phi_c \end{bmatrix} \quad (13a)$$

$$\Gamma_{cl} = \begin{bmatrix} I & \Gamma_{pa}D_c \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ D_c \end{bmatrix} \\ 0 & \Gamma_c \end{bmatrix}, \quad (13b)$$

and with the state vector

$$x(t_k) = \begin{bmatrix} x_p(t_{k+1}) \\ U(t_{k+1}) \\ x_c(t_{k+1}) \end{bmatrix}, \quad (13c)$$

and the noise vector

$$e(t_k) = \begin{bmatrix} v(t_k) \\ w(t_k) \end{bmatrix}. \quad (13d)$$

5. THE DISCRETE WEIGHT MATRIX

The weight matrix is found by identification comparing (5) and (6). After some handwork, the following integrals can be found:

$$q_{x,x} = \frac{1}{h_k}\int_{t_k}^{t_{k+1}} \Phi_p^T(t)\overline{Q}_{11}\Phi_p(t)dt \quad (14a)$$

$$q_{x,g} = \frac{1}{h_k}\int_{t_k+\tau_{k,g}}^{t_k+\tau_{k,g+1}} \Phi_p^T(t)\overline{Q}_{11}\Gamma_{pg}(t)dt \quad (14b)$$

$$q_{x,u} = \frac{1}{h_k}\int_{t_k}^{t_{k+1}} \Phi_p^T(t)\overline{Q}_{12}dt \quad (14c)$$

$$q_{g,g} = \frac{1}{h_k}\int_{t_k+\tau_{k,g}}^{t_k+\tau_{k,g+1}} \Gamma_{pg}^T(t)\overline{Q}_{11}\Gamma_{pg}(t)dt \quad (14d)$$

$$q_{g,u} = \frac{1}{h_k}\int_{t_k+\tau_{k,g}}^{t_k+\tau_{k,g+1}} \Gamma_{pg}^T(t)\overline{Q}_{12}dt \quad (14e)$$

$$q_{u,u} = \frac{1}{h_k}\int_{t_k}^{t_{k+1}} \overline{Q}_{22}dt\,. \quad (14f)$$

The submatrices q depend on the current instance k and the fractional delay instance g. Assembled, the weight matrix becomes

$$Q(t_k) = \begin{bmatrix} q_{x,x} & 0 & q_{x,0} & \cdots & q_{x,ng} & q_{x,u} \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ q_{x,0}^T & 0 & q_{0,0} & \cdots & 0 & q_{0,u} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{x,ng}^T & 0 & 0 & \ldots & q_{ng,ng} & q_{ng,u} \\ q_{x,u}^T & 0 & q_{0,u}^T & \cdots & q_{ng,u}^T & q_{u,u} \end{bmatrix}, \quad (15)$$

where additional rows and columns are inserted to cater for the difference within the triple $\{md, nd(k), ng(k)\}$. The numerical computations of the integrals can be performed by calculating matrix exponentials (Van Loan, 1978), see the appendix.

5.1 Two alternative definitions

Another construction of a loss function in this case, is of course the one traditionally used in optimal control,

$$\overline{J} = E\lim_{T\to\infty}\frac{1}{T}\int_0^T \begin{bmatrix} x_p(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} \\ \overline{Q}_{12}^T & \overline{Q}_{22} \end{bmatrix} \begin{bmatrix} x_p(t) \\ u(t) \end{bmatrix} dt\,. \quad (16)$$

The difference between (5) and (16) depends on where the "probe" to read the control signal is placed; the delay is instead placed after the hold block in Fig. 1. The formulation of the closed loop will stay the same. By identification of (16) with (6), it is found that $q_{x,x}$, $q_{x,g}$ and $q_{g,g}$ yield the weight matrix

$$Q(t_k) = \begin{bmatrix} q_{x,x} & 0 & q_{x,0} & \cdots & q_{x,ng} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ q_{x,0}^T & 0 & q_{0,0} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{x,ng}^T & 0 & 0 & \cdots & q_{ng,ng} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \qquad (17)$$

The difference between (5) and (16) depends on $\overline{Q}_{12}$, (Sanfridson, 2004).

A third alternative — a straight forward assembly which can be seen as a discrete time version from the start — is to discretize $\overline{Q}$ in (16) into Q without delays (Åström and Wittenmark, 1997), and expand it into

$$Q(t_k) = \begin{bmatrix} Q_{11} & 0 & Q_{12} \\ 0 & 0 & 0 \\ Q_{12}^T & 0 & Q_{22} \end{bmatrix} \qquad (18)$$

i.e. not punishing old control signals. This renders a lower value than the main alternative (17).

5.2 State noise

By the above mentioned identification the additional intersample loss becomes

$$J_v(t_k) = \frac{1}{h_k} tr\overline{Q}_{11} \int_{t_k}^{t_{k+1}} (t_{k+1} - s) e^{As} \overline{R}_v e^{A^T s} ds . \qquad (19)$$

5.3 Inserting the control signal

The loop is closed by inserting the control law, substituting $u(t_k)$, getting a new state vector. The transformation $Q_{cl}(t_k) = M^T Q(t_k) M$ based on

$$M = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ D_c M_p & E_c & C_c \end{bmatrix}, \begin{bmatrix} x_p(t_k) \\ U(t_k) \\ u(t_k) \end{bmatrix} = Mx(t_k) \qquad (20)$$

does the trick.

5.4 Measurement noise

When the loop is closed, the variance of measurement noise arising at time t_k should be added which yields the term

$$J_w(t_k) = tr(D_c^T Q_{uu} D_c R_w) . \qquad (21)$$

Q_{uu} is the weight belonging to $u(t_k)$, i.e. $q_{u,u}$, $q_{md,md}$ or Q_{22}.

6. PUTTING IT ALL TOGETHER

The closed loop derived in the previous sections, changes from one sampling instance to another. The timing of the computer system, i.e. the execution and communication, is modelled by a finite discrete Markov chain. The Markov chain models the pattern of time variations and the state covariance is governed by this switching of linear discrete state update models (Costa and Fragoso, 1993; Ji et al., 1991). Each Markov state contains information related to one sampling period. The probability of a jump from i to j is

$$p_{i,j}(t_k) = Prob\{r(t_{k+1}) = j \,|\, r(t_k) = i\} \qquad (22)$$

with states $i, j, r = \{1, \ldots, nr\}$. The constant square Markov state *transition probability matrix*

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1(nr)} \\ \cdots & \cdots & \cdots \\ p_{nr(1)} & \cdots & p_{nr(nr)} \end{bmatrix} \qquad (23)$$

of the homogenous chain (does not depend on t_k), will have a certain structure of zero jump probabilities; the delay cannot increase faster than time itself, i.e. $\dot{\tau}(t) \le 1$, but the delay can disappear instantaneously. A recursive equation giving the steady state *probability vector* is

$$\pi(t_{k+1}) = \begin{bmatrix} \pi_1(t_k) & \ldots & \pi_{nr}(t_k) \end{bmatrix} P = \pi(t_k) P , \qquad (24)$$

which converges to π^∞ when $k \to \infty$ provided the chain is positive recurrent and aperiodic (Cassandras and Lafortune, 1999).

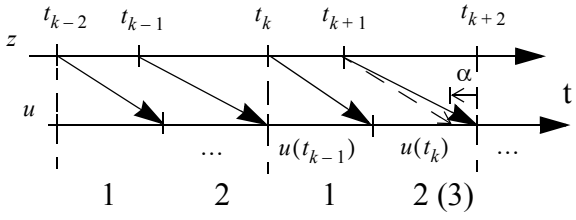

Fig. 5 Example of a periodic sequence.

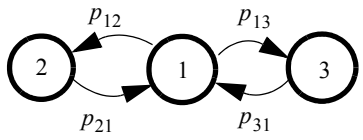

Fig. 6 Example of a periodic Markov chain.

An aperiodic chain can be used to model the general case of random time-variations, and a periodic chain can be used to model e.g. a static schedule. For a periodic chain (24) will not converge and the result depends on the initial value. However, the probability of finding the chain in a specific state is related to the expected recurrence time of that state. The periodic system in Fig. 5 has a periodicity θ of two. The instants $t_{k-2}, t_k, t_{k+2}, \ldots$ are equidistantly spaced. Type 1 corresponds to the interval $(t_k, t_{k+1}]$ and type 2 to $(t_{k+1}, t_{k+2}]$. A sequence of periods can be e.g. $s = 121212\ldots$. Note that the sequence of outputs in the figure does not exhibit jitter.

A chain can be periodic even if some state transitions are random. Let the second instance of type 2 be modified and call this type 3: u arrives at the process a time interval α earlier. A sequence becomes for example $s = 12131212\ldots$. The system is still 2-periodic, since state 1 is always visited every second jump, see Fig. 6. The transition probability matrix becomes

$$P = \begin{bmatrix} 0 & p_{12} & p_{13} \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{25}$$

and the main diagonal is naturally filled with zeros.

The state and measurement noise affects the state covariance, preferably formulated using Kronecker and vector notation. In addition to this, the analysis can be cast as a numerical convex optimization based on linear matrix inequalities, LMIs. The special case of a periodic system also fits in the analysis.

6.1 Steady state covariance

A state-dependent (conditional) covariance at time t_k can be defined by $S_i(t_k) = E(x(t_k)x^T(t_k)\big|r(t_k) = i)$ where $r, i \in \{1, \ldots, nr\}$ denote the state of the chain. The steady state covariance S_i^∞ is found as the limes when $k \to \infty$, provided it exists. From (6) the loss can be written as the sum over N number of non-uniform periods,

$$\begin{aligned} J &= E \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} J_x(t_k) + J_w(t_k) \\ &= E \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} tr(x^T(t_k)Q_{cl}(t_k)x(t_k)) + J_v(t_k) + J_w(t_k) \\ &= \sum_{i=1}^{nr} tr(\pi^\infty(i)S_i^\infty Q_{cl}(i)) + \pi^\infty(i)J_v(i) + \pi^\infty(i)J_w(i) \end{aligned} \tag{26}$$

where $Q_{cl}(i)$, $J_v(i)$ and $J_w(i)$ denote the weight and the expected contribution from the noise, at the state i. At the third equality, the domain changes from the time-line with arguments of t_k to the Markov chain with states i. The trace operator conveniently separates $Q_{cl}(i)$ from S_i^∞. The vectorized covariance matrix

$$\hat{S}(t_k) = \left[vec(\tilde{S}_1(t_k)) \ \ldots \ vec(\tilde{S}_{nr}(t_k))\right] \tag{27}$$

together with $\tilde{S}_i(t_k) = \pi_i(t_k)S_i(t_k)$, satisfies the recurrence equation

$$\hat{S}(t_{k+1}) = \hat{A}\hat{S}(t_k) + \hat{B}. \tag{28}$$

This can be solved by ordinary matrix manipulations, and the steady state conditional covariance $\tilde{S}_i^\infty$ is then retrieved by unfolding the column vector $\hat{S}^\infty$. See (Sanfridson 2004) for the definition of the update matrices in (28) and (Nilsson 1998) for a stringent derivation.

6.2 Periodic system

For a periodic Markov chain, with deterministic jumps only, the loss can be written more generally:

$$\begin{aligned} J = E \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{\theta} \sum_{\kappa=k}^{k+\theta-1} [x^T(t_\kappa)Q_{cl}(t_\kappa)x(t_\kappa) \\ + J_v(t_\kappa) + J_w(t_\kappa)] \end{aligned} \tag{29}$$

All periods of length θ are uniform and it is sufficient to take the expected value over one such period, instead of the average over N number of θ-periods. It must be assured that the covariance $E[x(t_\kappa)x^T(t_\kappa)]$ with $\kappa = k, \ldots, k+\theta-1$ converges. By applying a lifting procedure, the θ-periodic system can be written as a 1-periodic (or aperiodic from a sampled time viewpoint) equivalence (Bittanti et al., 1991). This can be applied when the ratio of e.g. the sampling period and control period is a rational number, in order to obtain a finite dimensional discrete time representation.

For a fully deterministic periodic system with $\theta > 1$, the Markov chain becomes an unnecessary theoretical

overhead. But all transitions in a periodic chain do not necessarily have to be deterministic, cf. the example in Fig. 5. In fact, the framework developed for random jump linear systems is applicable also when the Markov chain is periodic. The transition probability matrix is degenerated, c.f. (25), but the Markov property still holds. The periodicity $\theta > 1$ of the state probability vector can be described by $\pi^{\infty}(t_{k+\theta}) = \pi^{\infty}(t_k)$ for some k in steady state, and the state independent covariance will converge on the same premises as that of an aperiodic Markov chain.

6.3 Stability and robustness

The jump linear system is stable in the *mean square sense* and (28) will converge to an equilibrium if $eig(\hat{A}) < 1$, (Costa and Fragoso, 1992). A stochastic Lyapunov function candidate can be written as a set of coupled matrix inequalities

$$\sum_{j=1}^{nr} p_{ij}\Phi_{cl}^{T}(j)\Sigma_j\Phi_{cl}(j) - \Sigma_i < -\Theta(i) \quad i = 1, \ldots, nr. \quad (30)$$

The jump linear system is stochastically stable, which in turn implies mean square stability (Ji et al., 1991), if (30) holds for searched $\Sigma_i = \Sigma_i^T > 0$ and given $\Theta(i) = \Theta^T(i) \geq 0$, with $\forall i, j \in \{1, \ldots, nr\}$.

Consider the convex optimization problem of minimizing the objective function $f = x_0^T S x_0$, where x_0 is an arbitrary initial state vector, such that (30) holds, and with the interpretation $\Sigma_i = S_i$, $\Theta(i) = Q_{cl}(i)$. A smallest upper bound of the loss will be obtained, $J \leq f$. $Q_{cl}(i)$ can be interpreted as a performance margin or robustness requirement with respect to the perturbations sampling period and delay jitter.

7. EXAMPLE

The example demonstrates a simple yet interesting application of the quadratic measure to period and delay jitter. A nominal period of $h = 0.6$ is used both for analysis and control synthesis (with $\tau = 0$). The process, the fixed optimal controller, and weight matrix are given by

$$A_p = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \quad B_p = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C_p = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \quad M_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (31a)$$

$$D_c = \begin{bmatrix} -0.741 & -0.748 \end{bmatrix} \quad E_c = \begin{bmatrix} 0_{nd-1} & -0.007 \end{bmatrix}, \quad (31b)$$

$$\bar{Q}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \bar{Q}_{12} = \begin{bmatrix} 0.02 \\ 0.02 \end{bmatrix} \quad \bar{Q}_{22} = 1 \quad \text{and} \quad \bar{R}_v = I. \quad (31c)$$

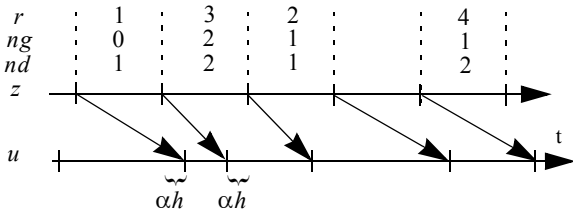

Fig. 7 Case 1: Delay jitter, constant h.

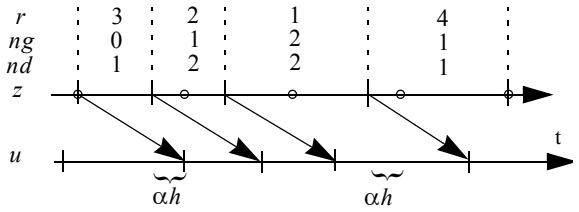

Fig. 8 Case 2: Sampling period jitter, constant $\tau = h$.

Fig. 7 (case 1) and Fig. 8 (case 2) show the timelines to be compared. In the first case, the delay changes with a probability of p or $1-p$, between the values $\{(1-\alpha)h, (1+\alpha)h\}$ for the constant h. In the second case, the period changes with a probability of p or $1-p$ between the values $\{(1-\alpha)h, (1+\alpha)h\}$, for a constant delay $\tau = h$.

Fig. 8 also exemplifies that multiple changes of the control signal are possible despite a constant delay (the rings are equidistantly spaced).

The states of the chain ($nr = 4$) have been selected such that the same transition probability matrix can be used in both cases. Consider the following choice,

$$P = \begin{bmatrix} 0 & 0 & 1-p & p \\ p & 1-p & 0 & 0 \\ p & 1-p & 0 & 0 \\ 0 & 0 & 1-p & p \end{bmatrix} \quad (32)$$

where the single parameter $p \in [0.1, 0.9]$ can be interpreted as the amount of variation, and $\alpha \in [0.1, 0.3]$ as the magnitude.

In Fig. 9 the difference between period and delay jitter is visualized with lines representing similar values. In both cases, the loss increases with p and the effect of α is similar. A higher α gives a larger variation in the result for different p. When α is zero the two cases coincide. A conclusion is that the effect of delay jitter and period jitter is rather similar.

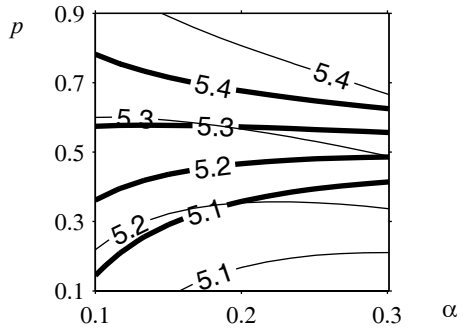

Fig. 9 Delay jitter (bold) vs. period jitter (thin).

8. THE WORST-CASE DELAY JITTER

What kind of jitter distribution will give the worst performance? This is an interesting question, both from a theoretical point of view but also from a practical one. Knowing the worst kind may simplify analysis and synthesis. If the performance is acceptable even for the worst type of jitter, the performance can only be better for other types. It thus becomes an upper bound. The average delay is indeed important, thus a fair way to formulate the problem is to hold the average delay constant when characterizing the shape of the distribution.

The searching for a worst or best distribution of delay jitter will here be made by brute force. One technique suitable for combinatorial problems where the structure is unknown, is simulated annealing. A pseudo code listing is found below:

```
old_value = low_or_high;
Tcur = Tstart;
while Tcur > Tfinal
   for k = 1 to maxtry
      old_mch = mch;
      ind = pick_random();
      mch = change_mch(mch,ind);
      value = calc_energy(mch);
      valid = is_change(value -
         old_value,Tcur);
      if valid
         old_value = value;
      else
         mch = old_mch;
      endif
   endfor
   Tcur = Tcur*Trate;
endwhile
```

The tuning parameters are the start temperature (`Tstart`), the final temperature (`Tfinal`) and the cooling rate (`Trate`). At each temperature level, at most `max_try` number of tries are made in this way: a modified transition matrix P is picked and changed randomly using functions `pick_random` and `change_mch` (each time, a row is changed in random, but there are alternative ways to do this). The loss is calculated by `calc_energy`. If it results in a decrease and in case a minimum value is searched for, it will be accepted. Otherwise, if it results in an increase, a random variable that depends on the current temperature (`Tcur`) will determine if the change is valid or not. The lower the temperature, the smaller the chance of accepting a change in the "wrong" direction. A clear drawback of this method is that it does not help asserting that an optimal solution has been found. Thus, the result has to be interpreted cautiously. The simulating annealing needs a tuning of temperatures and the method of changing parameters to yield an outcome which is satisfactory.

The example system is an inverted pendulum with a minimum variance controller. The delay jitter is modelled as a discrete random variable controlled by a Markov chain. The delay takes on values between and including the extreme pair $\{(1/4)h, (3/4)h\}$ with the split $s > 1$, and a nominal period h for which the closed loop system behaves well. The split is equal to the number of points in the distribution and also equal to the number of Markov states. The average delay is kept at $h/2$ by this jitter construction.

From an arbitrary initial transition probability matrix for a fixed s, two final transition probability matrices are seen in Fig. 10 and Fig. 11, for a (local) minimum and maximum respectively. Recall that the states jumps from a row to a column and that the row sum must equal one. In Fig. 10 there are $s = 2$ states and the figure reveals that if the state switches often, the loss is less than it would be otherwise. In Fig. 11 there are $s = 4$ states. The bars are highest at the edges. The simulated annealing has been interrupted prematurely to yield results that better shows the general direction.

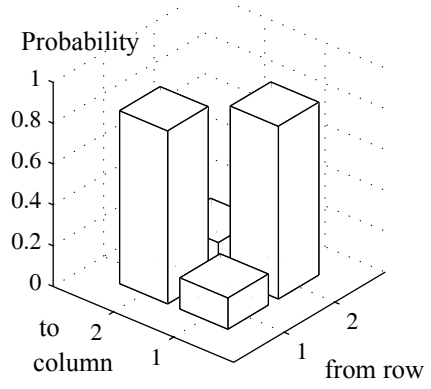

Fig. 10 Final transition probability matrix, P. Searching for the minimum loss with the split $s = 2$.

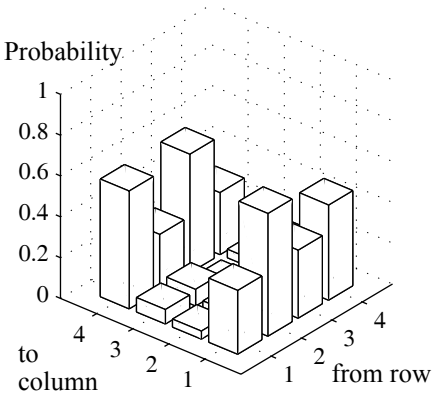

Fig. 11 Final transition probability matrix, P. Searching for the maximum loss with the split $s = 4$.

9. RELATED WORK

An optimal controller addressing random delay, was proposed by Nilsson (1998), and it was shown that the separation theorem holds. The gain of the controller changes according to the time stamped sensor to controller delay, and the expected controller to actuator delay. The total delay is limited to the constant sampling period.

Another method to design an optimal controller for random delay was described by Xiao et al. (2000). An iterating inner loop is applied to find the controller and an outer loop to find a perturbed transition probability matrix. The delay is a multiple of the sampling interval and the calculations are based on LMIs.

A tool called Jitterbug was presented by Lincoln and Cervin (2002), partly following from Nilsson (1998). It handles interconnected blocks describing the timing behaviour, and it is based on a tick defining the granularity. A mix of continuous and discrete time blocks is possible as well as multirate systems.

10. CONCLUSIONS AND FURTHER WORK

With sampling period jitter *or* control delay jitter, multiple switches of the piecewise constant control signal is possible between two sampling instants. In the proposed model of the computer architecture, the delay cannot increase faster than time itself, but may vanish instantaneously. In this paper, it is shown how the process and loss function can be discretized accordingly. The aim is primarily analysis of computer controlled systems, not control synthesis. The result generalizes the procedure with a fractional delay found in many text books. The derivation somewhat resembles multirate systems, which however deal exclusively with periodic events. The approach admits a formulation of LMIs. The loss function can be used at design time as well as at run-time where the timing behaviour of the underlying computer system is measured.

Further work might include adding measurement delay, investigation of the observability, a release of the assumption having a fixed controller, a study of robustness assessment and identification of Markov states from logged data.

11. ACKNOWLEDGEMENTS

This work has been funded by the Swedish Strategic research Foundation via the project FlexCon.

REFERENCES

Bittanti S., Laub A. J. and Willems J. C., (1991). *The Riccati equation*, Springer Verlag, ISBN 0-387-53099-1.

Cassandras C. G.and Lafortune S. (1999). *Introduction to discrete event systems*, Kluwer, ISBN 0-7923-8609-4.

Costa D. and Fragoso M. (1993). *Stability results for discrete-time linear systems with Markovian jumping parameters*, Journal of mathematical analysis and applications 179, p. 154-178.

Ji Y., Chizeck H. J., Feng X., Loparo K.A. (1991). *Stability and control of discrete-time jump linear systems*, Control-Theory and Advanced Technology, Vol. 7, No. 2, pp. 247-270.

Lincoln B. and Cervin A. (2002). *Jitterbug: A tool for analysis of real-time control performance*, In Proc. of the 41st Conf. on Decision and Control, Sydney.

Nilsson J. (1998). *Real-time control systems with delays*, PhD thesis, LTH, ISRN LUTFD2/TFRT--1049-SE.

Sanfridson M. (2004). *Quality of control and real-time scheduling*, PhD thesis, Mechatronics Lab, KTH, ISRN/KTH/MMK--04/7--SE.

Van Loan C. (1978). *Computing integrals involving the matrix exponential*, IEEE Trans. on automatic control, Vol. AC-23, No. 3.

Xiao L., Hassibi A., How J. P. (2000). *Control with random communication delays via a discrete time jump system approach*, Proc. of the American Control Conference.

Åström K. J. and Wittenmark B. (1997). *Computer controlled systems*, Prentice-Hall, ISBN 0-13-314899-8.

NOTATION

A_p, B_p, C_p, D_p	State space matrices for the continuous time dynamic process, p.
$\Phi_p, \Gamma_{pa}, \Gamma_{pb}$ C_p, M_p	Equivalent discrete time description of the process.
$\Phi_c, \Gamma_c, \Lambda_c$ C_c, D_c, E_c	Discrete time controller, c.
Φ_{cl}, Γ_{cl}	Closed loop system matrix and input matrix in discrete time.
E	Expected value operator.
i , j , r	State of the Markov chain.
$\overline{J}, J$	Loss.
h	Denotes period, $h \in \Re$.
P	Transition probability matrix of a Markov chain.
π , π^{∞}	State probability vector, especially stationary, of a Markov chain.
$\overline{Q}, Q, Q_{cl}$	Weight matrices, continuous and discrete time.
R	Noise variance.
S	A notion of covariance.
T	Time horizon in time units, $T \in \Re$. Also denotes the matrix transpose.
τ	Any delay.
θ	Periodicity $\theta \in \{1, 2, 3, \ldots\}$.
tr	The matrix trace operator.
$U(t_k)$	Vector of old outputs.
$u(t), u(t_k)$	Control output, continuous and discrete time.
$v(t_{k+1}, t_k)$ $w(t_k), e(t_k)$	State noise and measurement noise.
$x_p(t)$, $x_c(t_k)$ $x(t_k)$	The state vector of a process, a controller and closed loop.
$y(t), z(t_k)$	Process output, measured value.

APPENDIX

The matrices describing the discrete time state update equation is found by:

$$\Phi_p(h) = \exp(A_p h) = s_0 s_1 \ldots s_{ng} \tag{33}$$

and

$$\Gamma_{pg}(t, \tau_{k,(g+1)}, \tau_{k,g}) = s_{ng} s_{ng-1} \ldots s_{g+1} \sigma_g \tag{34}$$

with

$$\exp\left(\begin{bmatrix} A_p & B_p \\ 0 & 0 \end{bmatrix}(\tau_{k,g+1} - \tau_{k,g})\right) = \begin{bmatrix} s_g & \sigma_g \\ \varnothing & \varnothing \end{bmatrix} \tag{35}$$

Recall that $\exp(0) = I$. The symbol $\varnothing$ does not represent any particular expression but acts as a place holder for parts of the result not useful when obtaining the intended integral. The matrix exponential is split up and calculated for each partial delay, $g = 0, \ldots, ng$:

$$\left(\begin{bmatrix} q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix}\right)_g = (s_{22}^T)_{g+1}(s_{12})_{g+1} - (s_{22}^T)_g (s_{12})_g \tag{36}$$

and

$$\begin{aligned} q_{x,x} &= \frac{1}{t_{k+1} - t_k} \sum_{g=0}^{ng} (q_{11})_g \\ q_{x,g} &= \frac{1}{t_{k+1} - t_k} (q_{12})_g \\ q_{g,g} &= \frac{1}{t_{k+1} - t_k} (q_{22})_g \end{aligned} \tag{37}$$

using

$$\exp\left(\left[\begin{array}{cc|cc} -A_p^T & 0 & \overline{Q}_{11} & \overline{Q}_{12} \\ -B_p^T & 0 & \overline{Q}_{21} & \overline{Q}_{22} \\ \hline 0 & 0 & A_p & B_p \\ 0 & 0 & 0 & 0 \end{array}\right] \tau_{k,g}\right) = \begin{bmatrix} \varnothing & (s_{12})_g \\ \varnothing & (s_{22})_g \end{bmatrix}. \tag{38}$$

Φ_p and Γ_{pg} could also be extracted from this equation, as an alternative to (35). Recall the convention $\tau_{k,0} = 0$ and $\tau_{k,(ng+1)} = h$. Numerical issues might have to be considered when calculating the matrix exponentials.