Graceful Degradation of the Quality of Control through Data Drop Policy

Ning Jia, Ye-Qiong Song, Françoise Simonot-Lion

Graceful Degradation of the Quality of Control through Data Drop Policy

Jia Ning, Song YeQiong, Simonot-Lion Françoise

LORIA – INPL

Campus Scientifique – BP 239

54506 – Vandoeuvre-Lès-Nancy France

{jia,song,simonot}@loria.fr

Abstract—Sharing a network link in a networked control system (NCS) may result in the network overload leading to the non respect of the specified periodicity of control system, this may cause an unpredictable performance degradation of the system. In this paper, we consider an overload management technique which selectively drops the data packets of the NCS to avoid network overload. The investigated problems are: how to design the controller under packet drops and how to distribute the packet drops in a packet delivery sequence so that the quality of control (QoC) is optimized. The paper first gives the optimal LQ-controller using the conventional technique to reduce the deterioration in QoC due to packet drops. Then, the paper proposes a methodology for deriving a distribution of the packet drops in the packet delivery sequence so that the QoC is optimal. Finally, an efficient algorithm is given to reduce the computation complexity of the proposed methodology. This proposal contributes to the co-design of the controller and the resource performance management process at the implementation level.

1. INTRODUCTION

In this paper, we consider a Network Control System (see Figure 1). The sensors and the controller are connected by a network link. The process states are sampled periodically by the sensors, and the samples are transmitted in one packet to the controller through the network. The transmission of this packet introduces a delay between the sampling instant and the control law completion. If the delay is constant, it can be easily taken into account by the control law. Nevertheless, the constant-delay assumption cannot be guaranteed if the network is a resource shared by several applications especially due to network overload periods. The network overload must be handled since the time varying network-induced delay can degrade the QoC and can even destabilize the control system.
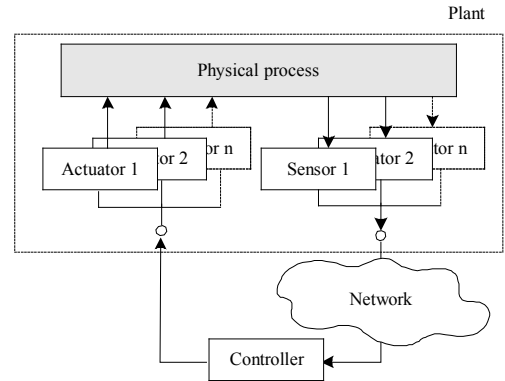

Figure 1. Networked control system

Avoiding network overload requires reducing bandwidth requirement of applications. For control system, this can be achieved by enlarging its sampling period [3][15][16]. As illustrated in Figure 2, different sampling period induces different QoC for a control system, when the sampling period exceeds a certain limit, the control system becomes unstable. In practice, the sampling period limits P_{min} and P_{max} are often determined using classic control theory and practice (e.g. the rule of thumb [1]), and the nominal sampling period is chosen between P_{min} and P_{max}. When the network is overload, the sampling period can be enlarged until P_{max} by sacrificing the QoC to reduce the bandwidth requirement. However, changing the sampling period of a control system alters its dynamics. Another approach for reducing bandwidth requirement is to selectively drop packets without changing the sampling period of control system. The solution discussed in this paper for dealing with the network overload is based on this latter approach.
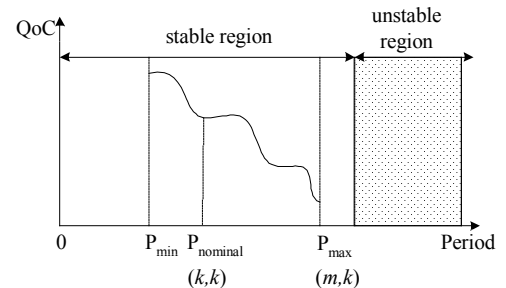

Figure 2. Relationship between sampling period or (*m,k*)-firm constraint and QoC

More specifically, we suppose that the nominal sampling time of control system has been chosen using the conventional sampling period choosing theory, and our solution to handle the network overload is based on the (*m,k*)-firm constraint model [8][14]. The (*m,k*)-firm constraint requires that, at any time, at least *m* packets among any *k* consecutive packets sent from the sensor must be received by the controller, where *m* and *k* are two positive integers with $m \leq k$ (the case where *m=k* is equivalent to the ideal case, noted as (*k,k*)-firm constraint). Obviously, the packet drops tend to degrade the QoC due to the misses of control law updates. However, if a control system is specified to accept a degradation of QoC until *k-m* packet drops among *k* consecutive packets (this can be justified by the observation that most control systems can tolerate misses of the control law updates to a certain extent), the system can then be designed according to the (*m,k*)-firm approach to offer the different levels of QoC, between (*k,k*)-firm (ideal case) and (*m,k*)-firm (worse case), with as many intermediate levels as the possible values between *k* and *m*. This results in a control system with graceful degradation of QoC.

As illustrated in Figure 2, this relation between QoC and (*m,k*)-firm constraint can be illustrated in the same way as for that between QoC and sampling period by simply replacing P_{max} by (*m,k*)-firm constraint and $P_{nominal}$ by (*k,k*)-firm constraint.

The application of such an overload management strategy in control system requires a systematic study of the impact of the packet drops based on the (*m,k*)-firm constraint on the QoC. In most of the prior approaches handling the packet drops, like those in [11][12][17], the packet drop process is regarded as an indeterminist process, therefore only some probabilistic results have been produced. An important contribution is found in [14]. The author proposed a scheduling technique based on the (*m,k*)-firm constraint that drops selectively the control law computation to handle the processor overload and the author also gave a approach for modifying the control law to reduce the deterioration in QoC due to the misses of control law update. However, the issue that how to choose a reasonable (*m,k*)-firm constraint that guarantees control system stability was not addressed, and the approach deriving the optimal control law under control law update misses is not suitable because the control signal and the process state at the moment when the control law update is dropped are not penalized in the cost function. Furthermore, the proposed algorithm implicitly drops the control law computation without justifying the optimality of the resulting control law update sequence from the QoC point of view.

In [9], we presented a formal analysis that derived, for a one-dimensional control system, the *m* and *k* values that guarantee the control system stability. We also proposed an approach for deriving the optimal controller under (*m,k*)-firm constraint. In [5], we extended these results to a multiple dimension control system. We showed how to determine the value of *k* so that the control system remains stable for any possible value of *m,* and how to identify the values of *m* in order to minimize the LQ cost. After having

shown that the distribution of the packet drops in a packet delivery sequence has an important impact on the QoC, we justified that, for the finite time horizon, a judicious choice of packet delivery sequence plays an important role in reducing the degradation of QoC.

In this paper, we will give the optimal LQ-controller under (*m,k*)-firm constraint by using the conventional technique. We will also propose an approach to derive, for infinite time horizon, the packet delivery sequence which optimizes the QoC.

The paper is organized as follows. In section 2, we formalize the system under study. Section 3 gives the optimal LQ-controller under (*m,k*)-firm constraint. Section 4 presents an approach for deriving the optimal packet delivery sequence. A case study of the proposed approach is presented in section 5. Finally, we summarize our work and show the perspectives.

2. CONTROL SYSTEM DESCRIPTION

The control system considered in this work is shown in Figure 3. The system consists of a continuous plant:

$$dx = Ax dt + Bu dt + dv_c \qquad (1)$$

and a discrete linear controller

$$u_{ih} = -L x_{ih} \quad i = 0,1,2,\dots .$$

where $x \in R^n, u \in R^m,$ and A,B,L are matrices of appropriate sizes. x is the state vector containing state variables of the controlled process, and u is the control input. The process v_c has mean value of zero and uncorrelated increments. The incremental covariance of v_c is $R_{1c} dt$.
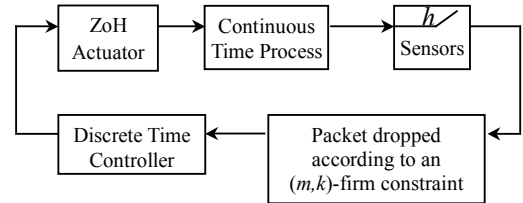

Figure 3. NCS under a dropping policy

The packets containing the process state variables are dropped selectively according to an (*m,k*)-firm constraint during the network overload period. For analyzing the impact of the (*m,k*)-constraint on the performance of control system, we placed ourselves in the worst-case by assuming a long network overload period and the packet drop process drops systematically *k-m* packets every *k*. The packet delivery sequence is thus periodic with period *k*, i.e., if the nth packet is dropped, the $(n+k)^{th}$ packet will be also dropped; otherwise, if the nth packet is received by the controller, the $(n+k)^{th}$ packet will be also received.

The discrete time-variant system model of the system (1) for a given packet delivery sequence is given as follows:

$$x_{i+1} = \Phi_i x_i + \Gamma_i u_i + v_i \quad i = 0,1,2,\dots \qquad (2)$$

where

$$\Phi_i = e^{Af_ih}$$

$$\Gamma_i = \int_0^{f_ih} e^{A\tau}d\tau B$$

and f_i is the number of the consecutively dropped packets plus one. In another word, it gives the distance, in terms of the number of the period h, between any two consecutive control law updates. v_i is a discrete-time Gaussian white-noise process with zero mean value and the following property:

$$Ev_iv_i^T = R_1(f_ih) = \int_0^{f_ih} e^{A\tau}R_{1c}e^{A^T\tau}d\tau \tag{3}$$

Since the packets are dropped periodically within kh according to a (m,k)-firm constraint, therefore we have $f_i + f_{i+1} + .. + f_{i+m-1} = k$ and $f_i = f_{i+m}$. If we only consider the m logical instants inside a kh interval, $\Phi_i = \Phi_{i+m}$, $\Gamma_i = \Gamma_{i+m}$, and the system in (2) is periodic with period m.

3. OPTIMAL CONTROL LAW UNDER (m,k)-FIRM CONSTRAINT

In this section, the optimal LQ controller is derived for the system (2).

We consider a continuous-time cost function:

$$J = E\left(\int_0^{Nh}\left(x^T(t)Qx(t) + u^T(t)Ru(t)\right)dt + x^T(Nh)Q_0x(Nh)\right) \tag{4}$$

where N is a multiple of k.

The cost function discretized for the discrete LQ controller of the system (2) is given by [7] as:

$$J = \sum_{i=0}^{m\frac{N}{k}-1}\left(x_i^T Q_i' x_i + 2x_i^T M_i u_i + u_i^T R_i' u_i\right) + E\left(x_{m\frac{N}{k}}^T Q_0 x_{m\frac{N}{k}}\right) + \sum_{i=0}^{m\frac{N}{k}-1}\bar{J}_i \tag{5}$$

where

$$Q_i' = \int_0^{f_ih}\Phi^T(\tau)Q\Phi(\tau)d\tau \qquad M_i = \int_0^{f_ih}\Phi^T(\tau)Q\Gamma(\tau)d\tau$$

$$R_i' = \int_0^{f_ih}\left(\Gamma^T(\tau)Q\Gamma(\tau) + R\right)d\tau \qquad \bar{J}_i = tr\left(Q\int_0^{f_ih} R_1(\tau)d\tau\right)$$

and $\Phi(t) = e^{At}$ and $\Gamma(t) = \int_0^t e^{A\tau}d\tau B$.

The optimal control law that minimizes the cost function (5) is given by [1] as:

$$u_i = -L_i x_i \qquad i = 0,1,2,\ldots \tag{6}$$

where

$$L_i = \left(\Gamma_i^T S_{i+1}\Gamma_i + R_i'\right)^{-1}\left(\Gamma_i^T S_{i+1}\Phi_i + M_i^T\right) \tag{7}$$

and S_t is obtained from the following recurrent equation:

$$S_{m\frac{N}{k}} = Q_0$$

$$S_i = \Phi_i^T S_{i+1}\Phi_i + Q_i'$$
$$\quad - \left(\Gamma_i^T S_{i+1}\Phi_i + M_i^T\right)^T\left(\Gamma_i^T S_{i+1}\Gamma_i + R_i'\right)^{-1}\left(\Gamma_i^T S_{i+1}\Phi_i + M_i^T\right) \tag{8}$$

Because of the periodicity of Φ_i and Γ_i, the solution of (8) is periodic with period k [2] if the horizon $m\frac{N}{k}-1$ is large, that is $S_i = S_{i+m}$. As a result, the solution of the equation (7) is also periodic with period m:

$$L_i = L_{i+m}$$

4. OPTIMAL PACKET DELIVERY SEQUENCE

In this section, we propose an approach for deriving the optimal packet delivery sequence that minimizes the cost function (5).

Note that finding the optimal packet delivery sequence is to find the suitable value of f_i for $i \in [0,m-1]$ because of the periodicity of packet drops. We first give the derivative of LQ cost function (5) with respect to f_i. This function is then used for deriving the optimal value of f_i for $i \in [0,m-1]$. At the end of this section, a computationally cheaper algorithm is also given to derive the near-optimal packet delivery sequence.

4.1. *Derivative of LQ cost function*

The minimal value of J given by the optimal LQ controller (6) is derived by [7] as:

$$J = m_0^T S_0 m_0 + \sum_{i=0}^{m\frac{N}{k}-1} tr S_{i+1}R_1(f_ih) + \sum_{i=0}^{m\frac{N}{k}-1}\bar{J}_i \tag{9}$$

where

$$\bar{J}_i = tr\left(Q\int_0^{f_ih} R_1(\tau)d\tau\right) \tag{10}$$

When time goes to infinity, i.e. $\lim m\frac{N}{k} \to \infty$, the influence from the initial condition decreases and because of the periodicity of the system, the cost function (9) may be written as:

$$J = \sum_{i=0}^{m-1} tr S_{i+1}R_{Vi} + \sum_{i=0}^{m-1}\bar{J}_i \tag{11}$$

This means that only the stationary cost is regarded and that the cost is scaled by the period of packet delivery sequence k.

In order to use the cost function (11) in the optimization, it is useful to know the derivative with respect to f_i for $i \in [0,m-1]$.

Theorem 1. Given a packet delivery sequence, i.e., $f_0, f_1 .. f_{m-1}$, the first derivative of J with respect to f_i for $i \in [0,m-1]$ is given as

$$\frac{dJ}{df_i} = \sum_{j=0}^{m-1} tr\frac{dS_{j+1}}{df_i}R_{Vj} + tr S_{i+1}\frac{dR_1(f_ih)}{df_i} + \frac{d\bar{J}_i}{df_i}$$

where $\frac{dS_{j+1}}{df_i}$ is obtained from the following recursive equation:

$$\frac{dS_j}{df_i} = \left(\Phi_{j+1} - \Gamma_{j+1}L_{j+1}\right)^T\frac{dS_{j+1}}{df_i}\left(\Phi_{j+1} - \Gamma_{j+1}L_{j+1}\right)$$

for $\mathrm{mod}\left(\dfrac{j}{m}\right)\neq i$

and

$$\frac{dS_i}{df_i}=\left(\Phi_{i+1}-\Gamma_{i+1}L_{i+1}\right)^T\frac{dS_{i+1}}{df_i}\left(\Phi_{i+1}-\Gamma_{i+1}L_{i+1}\right)+W_i$$

with

$$W_i=h\left[\left(\Phi_i-\Gamma_iL_i\right)^T \quad -L_i^T\right]\begin{bmatrix}Q&0\\0&R\end{bmatrix}\begin{bmatrix}\Phi_i-\Gamma_iL_i\\-L_i\end{bmatrix}$$
$$+\left(h\left(\Phi_i-\Gamma_iL_i\right)^T A^T-hL_i^TB^T\right)S_{i+1}\left(\Phi_i-\Gamma_iL_i\right)$$
$$+\left(\Phi_i-\Gamma_iL_i\right)^T S_{i+1}\left(hA\left(\Phi_i-\Gamma_iL_i\right)-hBL_i\right)$$

The matrix $R_1(f_ih)$ can be calculated using lemma 1 in [4] and the derivatives of $R_1(f_ih)$ and $\overline{J}_i$ are obtained directly from Eq (3) and (10).

$$\frac{dR_1(f_ih)}{df_i}=he^{Af_i}R_{1c}e^{A^Tf_i}d\tau$$

$$\frac{d\overline{J}_i}{df_i}=tr\left(QR_1(f_ih)h\right)$$

See Appendix for the proof.   □

4.2 Finding an optimal packet delivery sequence

The optimal packet delivery sequence is that minimizing the cost function (11). This is formulated as the following optimization problem:

Determine f_i for $i\in[0,m\text{-}1]$ that minimize $\quad J$

so that $\displaystyle\sum_{i=0}^{m-1}f_i=k$,

$f_i\geq0$ and f_i is an integer.

$$(12)$$

This optimization problem is a nonlinear integer program which can be solved by the branch-and-bound algorithm [10].

The branch-and-bound algorithm is a general search method for finding optimal solutions of discrete and combinatorial optimization problems. This algorithm aims at exploring, in an intelligent way, the space of feasible solutions of the problem.

According to branch-and-bound algorithm, the original problem (12) is firstly solved as a continuous nonlinear program, ignoring the integrality requirement. If each variable f_i takes integer value, the calculation process is stopped. Otherwise, if a solution f_i is not completely integer-feasible, then the approach is to generate two new subproblems from (12), with additional bounds, respectively

$$f_i\leq\lfloor f_i\rfloor$$

and

$$f_i\geq\lceil f_i\rceil$$

These two subproblems are then solved by ignoring the integrality requirement. The solution given by these two subproblems are compared and we keep the subproblem whose solution corresponds to the best cost value and we do not consider further the other subproblem. The above procedure is then repeated from the remaining subproblem. The whole process terminates when f_i is integer (normally, an approximate value is taken) for $i\in[0,m\text{-}1]$.

To solve the continuous nonlinear problem, a powerful technique is to use Kuck-Tucker conditions. Due to the restriction of the length of the paper, the condition will not be given here, the reader could refer to [6] for a detail information about the Kuck-Tucker conditions .

For the problem (12), the Kuhn-Tucker condition gives that if $\{f_0\,f_2\,..f_{m-1}\}$ is an optimal solution then

$$\left[\frac{dJ}{df_0}\quad\cdots\quad\frac{dJ}{df_{m-1}}\right]^T-\Lambda=0 \qquad(13)$$
$$f_0+f_1+..f_{m-1}-k=0$$

where Λ is a vector of dimension m, in which each element is the Lagrange multiplier λ.

4.3 An approximate version

The approach proposed above includes solving both Riccati and Lyapunov equations, and the branch-and-bound algorithm also needs some time to find the optimal solution. This is expensive and prevents from the implementation of a on-line network load management mechanism which dynamically adjusts the (m,k)-firm constraint and the corresponding packet delivery sequence according to the network load. Therefore a computationally cheaper algorithm for finding the near-optimal packet delivery sequence is desirable.

From the Kuhn-Tucker condition (13), we know that the QoC is optimal when $f_0=f_1..=f_{m-1}$ (The ideal case, i.e. the distribution of packet drops in a packet delivery sequence is absolutely uniform. However, this is not always possible), and one can easily find that the distribution of the packet drops derived by the algorithm proposed is an approximate version of the ideal case. Thus, if the packet drops are distributed as uniformly as possible, we can obtain a near-optimal QoC.

To facilitate the presentation of the packet delivery sequence, we will use a binary word of length k on an alphabet composed by 1 and 0. 1 represents that a packet is received by the controller, and 0 represents a packet drop. For example, the binary word 1001000 represents that the first and the fourth packets in a period of packet delivery sequence are received by the controller, all the other packets are dropped. This corresponds to $f_0=3$, $f_1=4$. We call this binary word the (m,k)-pattern. Therefore, distributing the packet drops as uniformly as possible is to have an (m,k)-pattern in which the letters 0 are uniformly distributed.

From [13], we know that the letters 0 are uniformly distributed in a binary word called upper mechanical word. Therefore, the technique for obtaining an upper mechanical word can be used to derive the (m,k)-pattern. The definition of the upper mechanical word is given as follows:

Definition 1 [13]. A binary word is a upper mechanical if

and only if there exist a real number α such that the nth letter of the word is given by

$$\lceil (n+1) \cdot \alpha \rceil - \lceil n \cdot \alpha \rceil \quad \forall n \geq 0. \tag{14}$$

The proportion of the letters 1 and 0 in the upper mechanical word is given by α. For satisfying the requirement for (m,k)-firm constraint, it is enough to replace α by m/k. Therefore, the nth letter of the (m,k)-pattern for a (m,k)-firm constraint is given by:

$$\left\lceil (n+1) \cdot \frac{m}{k} \right\rceil - \left\lceil n \cdot \frac{m}{k} \right\rceil \quad \forall n \geq 0. \tag{15}$$

Example 1. Given a (7,10)-firm constraint, from (15), we get the (7,10)-pattern 1110110110.

Using this approximate solution instead of the exact one gives a much less computation density. However, the approximation does not represent lower efficiency. We will see in the following section that for our experiment platform, the approximate solution gives the exact optimal solution.

5. CASE STUDY

In this section, we present a example to illustrate how to derive the optimal packet delivery sequence. We also compare our approach with the sampling period enlargement approach throughout an example.

Consider a cart-control system whose objective is to control the position of a cart along a rail according to a position reference. The state variables of the cart (speed and position) are periodically sampled and put in a packet which is then sent to the controller over the network link. For avoiding the network overload, the packets are dropped according to a (m,k)-firm constraint.

The continuous model of the cart system is given by:

$$dx = \begin{bmatrix} 0 & 1 \\ 0 & -12.6559 \end{bmatrix} x dt + \begin{bmatrix} 0 \\ 1.9243 \end{bmatrix} u dt + dv_c$$

v_c is the disturbance on the control signal with the incremental covariance $R_{1c} = \begin{bmatrix} 0 & 0 \\ 0 & 1^{-5} \end{bmatrix}$.

The continuous-time signal x is sampled with a period of 0.01 second. The packets containing the process samples are dropped according to a (3,11)-firm constraint. The problem of finding the optimal packet delivery sequence is set up as follows:

Determine f_i for $i \in [0,2]$ that minimize J

so that $\quad \sum_{i=0}^{2} f_i = 11$,

$f_i \geq 0$ and f_i is an integer.

(16) According to the algorithm proposed in the previous section, the problem (16) is solved as a continuous nonlinear program ignoring the integrality requirement. From the Kuck-Tucker conditions (13), optimal solution for the continuous nonlinear program is $f_0 = f_1 = f_2 = 3.67$. As this is not an integer solution, we thus force f_0 to be integer. To do so, we branch on f_0, creating two new sub-problems. In

one, we add the constraint $f_0 \leq 3$. In the other, we add the constraint $f_0 \geq 4$. This is illustrated in Figure 4.
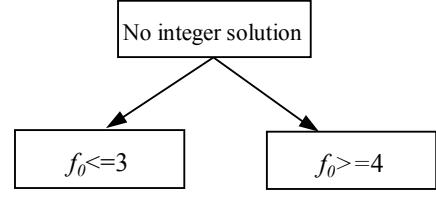

Figure 4. Example - first branching process

Now, any optimal solution to the overall problem must be feasible to one of the subproblems. By solving the two subproblems with the Kuck-Tucker conditions, we get the two solutions:

(1) $f_0 = 3$, $f_1 = 4.15$, $f_2 = 3.85$: $J = 0.00103$
(2) $f_0 = 4.03$, $f_1 = 3.32$, $f_2 = 3.65$: $J = 0.0011$

As the cost obtained with $f_0 = 4.03$ is greater than that obtained with $f_0 = 3$, we thus choose $f_0 = 3$ and branch on f_1. After solving the resulting subproblems, we have the branch and bound tree in Figure 5:
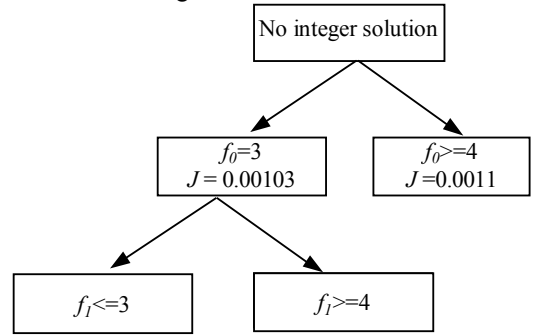

Figure 5. Example - Second branching

Solving the two subproblems gives:

(1) $f_0 = 3$, $f_1 = 3.02$, $f_2 = 4.98$: $J = 0.0012$
(2) $f_0 = 3$, $f_1 = 4$, $f_2 = 4$: $J = 0.0011$

The best integer solution is given by $f_0 = 3, f_1 = 4$ and $f_2 = 4$. The calculation process terminates since all the variables have taken an integer solution. Note that the solution derived by (15) is, in term of (m,k)-pattern, 10010001000 which is exactly the same with the optimal solution.

For a reference position of 0.1 meter, the two monitoring of the cart position obtained with an arbitrary packet delivery sequence and the optimal packet delivery sequence are respectively given in Figure 6 and Figure 7. The position trajectory of the car in the ideal case (i.e. the system without packet drops) is also given in figure 8. The corresponding cost J for each case is given in the figure comments.

Comparing Figure 6 and Figure 7 shows that the position trajectory under the optimal packet delivery sequence is more convergent than that under the arbitrary packet delivery sequence. The comparison of the costs confirms also the improvement in QoC under the optimal packet delivery sequence. Figure 8 shows that the state trajectory in the ideal case is slightly more convergent than that in Figure

7, and the cost is also a little better. Nevertheless, the request for network bandwidth of the cart system under the (3,11)-firm constraint has decreased significantly. The system requires only 27% of required network bandwidth of the ideal system, reducing therefore considerably the network overload.
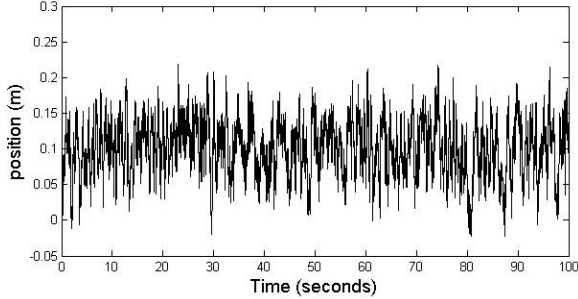

Figure 6. Position trace of the cart under (3,11)-firm constraint, with f_0=3, f_1=1 and f_2=7. J=15.9487.
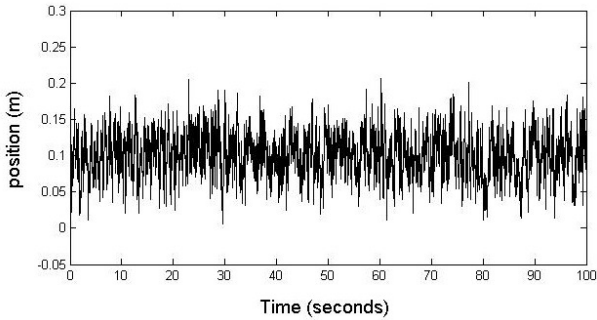

Figure 7. Position trace of the cart under (3,11)-firm constraint, with f_0=3, f_1=4 and f_2=4. J=12.7649.
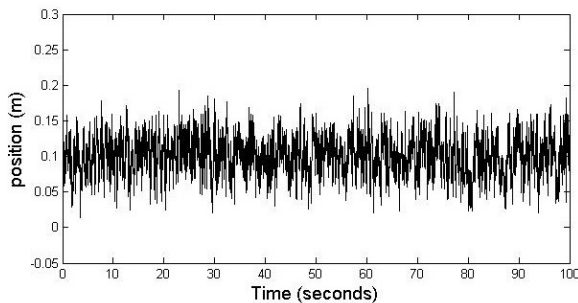

Figure 8. Position trace of the cart without packet drops (or under (1,1)-firm constraint). J=10.7041

As stated earlier, another way of reducing the effective bandwidth requirement is to enlarge the sampling period of control system. For example, to reduce the bandwidth requirement to 27%, we can enlarge sampling period to 3.677 (11/3*0.01) second.

To give an insight of the interest of our approach, we also give in Figure 9 the positions trace of the cart system with sampling period of 3.677 seconds. Comparing figure 7 and figure 9 shows that the there is no notable difference between them. The cost for the cart system with enlarged

sampling period is slightly better. This is obvious because, as noted earlier, the optimal performance is given by a uniform sampling period. However, the (m,k)-firm constraint approach does not alter the dynamics of the subsystem.
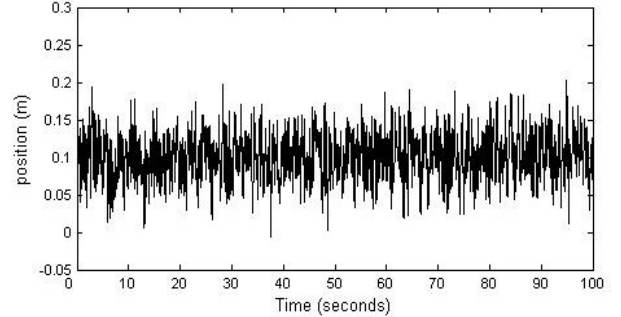

Figure 9. Position trace of the cart with sampling period 3.677. J=12.7547.

6. CONCLUSION AND PERSPECTIVE

In this paper, we propose an overload management technique based on (m,k)-firm constraint that selectively drops the data packets of the NCS to reduce the network bandwidth requirement. The paper first gives the controller design method. Then, it proposes an approach to derive optimal packet delivery sequence. To reduce the computation complexity, a computationally cheaper algorithm which derives the near-optimal packet delivery sequence is also proposed.

The results of this paper provide the theoretic basis for the application of (m,k)-firm model in control systems. As a future work, we plan to further extend the proposed approach so that control application can be adapted not only to cope with the overloads, but also to conform to the control application dynamics and the feasibility of the scheduling. That is, to explore an integrated QoC management framework based on (m,k)-firm model that dynamically tunes the (m,k)-constraints of control systems according to changes in the network load.

REFERENCES

[1] Astrom. K.J. and Wittenmark. B. "Computer Controlled Systems: Theory and Design", 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[2] Bittanti, S. Bittanti, P. Colaneri, and G. De Nicolao, The periodic Riccati equation, in "The Riccati Equation" (S. Bittanti, A. J. Laub, and J. C. Willems, Eds.), pp. 127-162, Springer-Verlag, Berlin, 1991.

[3] Cervin. A, Eker. J, Bernhardsson. B, and Årzn. K.-E, "Feedback feedforward scheduling of control tasks," *Real-Time Syst.*, vol. 23, no. 1-2, pp. 25--53, 2002.

[4] Eker. J, Hagander. P, and Årzén. K.E, "A Feedback Scheduler for Real-Time Controller Tasks", *Control Engineering Practice*, 12, 8, 2000, p. 1369-1378.

[5] Felicioni. F, Jia. N, Song. Y.Q, Simonot-Lion. F, "Impact of a (m,k)-firm data dropouts policy on the quality of control", *6th IEEE International Workshop on Factory Communication Systems - WFCS'2006.* (Torino, Italy). 2006.

[6] Fletcher. R, "Practical Methods of Optimization", second edn, Wiley, UK, 1987.

[7] Gustafsson. K, and Hagander, P (1991). "Discrete-time LQG with cross-terms in the loss function and the noise description" Report TFRT-7475

[8] Quan. G, and Hu. X, " Enhanced Fixed-priority Scheduling with

(m,k)-firm Guarantee ", *Proc. Of 21st IEEE Real-Time Systems Symposium*, pp.79-88, Orlando, Florida, (USA), November 27-30, 2000.

[9] Jia. N, Song. Y.Q, and LIN. R.Z, "Analysis of networked control system with packet drops governed by (m,k)-firm constraint", *6th IFAC International Conference on Fieldbus Systems and their Applications - FeT'2005*. (Puebla, Mexico). 2005.

[10] Land A. H. and Doig. A. G, "An Automatic Method for Solving Discrete Programming Problems". *Econometrica*, Vol.28 (1960), pp. 497-520.

[11] Ling. Q and M.D. Lemmon, "Robust performance of soft real-time networked control systems with data dropouts", *Proc. Of the IEEE Conference on Decision and Control*, 2002.

[12] Ling. Q. and M.D. Lemmon, "Soft real-time scheduling of networked systems with dropouts governed by a Markov chain", *Proc. Of the American Conference on Control*, 2003.

[13] Lothaire. M. "Combinatorics on Word". *Cambridge Academic Press*, second edition, 1997.

[14] Ramanathan. P, "Overload management in Real-Time control applications using (m,k)-firm guarantee ", *IEEE Transactions on Parallel and Distributed Systems*, 10(6) :549-559, Jun 1999.

[15] Seto, D., Lehoczkyn, J. P., Sha, L. and Shin, K. G., "On task schedulability in real-time control systems", *Proceedings of the 17th IEEE Real-Time Systems Symposium*, pp. 13-21, Washington, DC, USA, 1996.

[16] Simon. D, Robert. D and Sename. O, "Robust control/scheduling co-design: application to robot control", *RTAS'05 11th IEEE Real-Time and Embedded Technology and Applications Symposium* march 2005 San Francisco, CA

[17] Zhang. W. M. S. Brannicky, and S. M. Philips, "Stability analysis of networked control systems", *IEEE Control System Magazine*, 21(1):84-89, Feb 2001.

APPENDIX – proof of theorem 1

Proof. From (6) and (7), we get

$$\left(\Gamma_i^T S_{i+1} \Gamma_i + R_i'\right) L_i = \left(\Gamma_i^T S_{i+1} \Phi_i + M_i^T\right) \quad (17)$$

and

$$S_i + L_i^T \left(\Gamma_i^T S_{i+1} \Gamma_i + R_i'\right) L_i = \left(\Phi_i^T S_{i+1} \Phi_i + Q_i\right)^T \quad (18)$$

Regrouping (17) and (18) into a matrix, we get

$$\begin{bmatrix} S_i + L_i^T G_i L & L_i^T G_i \\ G_i L_i & G_i \end{bmatrix} = \begin{bmatrix} \Phi_i^T \\ \Gamma_i^T \end{bmatrix} S_{i+1} [\Phi_i \quad \Gamma_i] + Q_{i,d} \quad (19)$$

where $G_i = \Gamma_i^T S_{i+1} \Gamma_i + R_i'$, and

$$Q_{i,d} = \begin{bmatrix} Q_{i,1d} & Q_{i,2d} \\ Q_{i,2d}^T & Q_{i,3d} \end{bmatrix} = \int_0^{f_i h} e^{\Sigma^T t} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} e^{\Sigma t} dt$$

with

$$e^{\Sigma t} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} t\right) = \begin{bmatrix} \Phi(t) & \Gamma(t) \\ 0 & I \end{bmatrix}$$

To know how *J* depends on the packet delivery sequence, it remains to investigate *S*. Eq. (19) is differentiated with respect to f_i

$$\begin{bmatrix} 0 & 0 \\ \dfrac{dL_i}{df_i} & 0 \end{bmatrix}^T \begin{bmatrix} S_i & 0 \\ 0 & G_i \end{bmatrix} \begin{bmatrix} I & 0 \\ L_i & I \end{bmatrix} +$$

$$\begin{bmatrix} I & 0 \\ L_i & I \end{bmatrix}^T \begin{bmatrix} \dfrac{dS_i}{df_i} & 0 \\ 0 & \dfrac{dG_i}{df_i} \end{bmatrix} \begin{bmatrix} I & 0 \\ L_i & I \end{bmatrix} +$$

$$\begin{bmatrix} I & 0 \\ L_i & I \end{bmatrix}^T \begin{bmatrix} S_i & 0 \\ 0 & G_i \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \dfrac{dL_i}{df_i} & 0 \end{bmatrix} =$$

$$\frac{dQ_{i,d}}{df_i} + \begin{bmatrix} \dfrac{d\Phi_i^T}{df_i} \\ \dfrac{d\Gamma_i^T}{df_i} \end{bmatrix} S_{i+1} [\Phi_i \quad \Gamma_i] + \begin{bmatrix} \Phi_i^T \\ \Gamma_i^T \end{bmatrix} \frac{dS_{i+1}}{df_i} [\Phi_i \quad \Gamma_i]$$

$$+ \begin{bmatrix} \Phi_i^T \\ \Gamma_i^T \end{bmatrix} S_{i+1} \begin{bmatrix} \dfrac{d\Phi_i}{df_i} & \dfrac{d\Gamma_i}{df_i} \end{bmatrix}$$

Rearranging the terms yields

$$\begin{bmatrix} 0 & \dfrac{dL_i^T}{df_i} G \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \dfrac{dS_i}{df_i} & 0 \\ 0 & \dfrac{dG_i}{df_i} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ G \dfrac{dL_i}{df_i} & 0 \end{bmatrix} =$$

$$\begin{bmatrix} \Phi_i^T - L_i^T \Gamma_i^T \\ \Gamma_i^T \end{bmatrix} \frac{dS_{i+1}}{df_i} [\Phi_i - \Gamma_i L_i \quad \Gamma_i] + \begin{bmatrix} I & 0 \\ -L_i & I \end{bmatrix}^T \overline{W}_i \begin{bmatrix} I & 0 \\ -L_i & I \end{bmatrix}$$

$$(20)$$

where

$$\overline{W}_i = \frac{dQ_{i,d}}{df_i} + \begin{bmatrix} \dfrac{d\Phi_i^T}{df_i} \\ \dfrac{d\Gamma_i^T}{df_i} \end{bmatrix} S_{i+1} [\Phi_i \quad \Gamma_i] + \begin{bmatrix} \Phi_i^T \\ \Gamma_i^T \end{bmatrix} S_{i+1} \begin{bmatrix} \dfrac{d\Phi_i}{df_i} & \dfrac{d\Gamma_i}{df_i} \end{bmatrix}$$

From (20) the following equation is obtained

$$\frac{dS_i}{df_i} = \Psi_i^T \frac{dS_{i+1}}{df_i} \Psi_i + \begin{bmatrix} I & -L_i^T \end{bmatrix} \overline{W}_i \begin{bmatrix} I \\ -L_i \end{bmatrix}$$

where $\Psi_i = \Phi_i - \Gamma_i L_i$.

By the same approach, we get

$$\frac{dS_j}{df_i} = \Psi_{j+1}^T \frac{dS_{j+1}}{df_i} \Psi_{j+1} \quad \text{for} \quad \text{mod}\left(\frac{j}{m}\right) \neq i$$

To calculate $\overline{W}_i$, formulas for $\dfrac{dQ_{i,d}}{df_i}$, $\dfrac{d\Phi_i}{df_i}$ and $\dfrac{d\Gamma_i}{df_i}$ are needed:

$$\frac{d\Phi_i}{df_i} = hA\Phi_i \qquad \frac{d\Gamma_i}{df_i} = hA\Gamma_i + hB$$

$$\frac{dQ_{i,d}}{df_i} = h e^{\Sigma^T f_i h} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} e^{\Sigma f_i h} = h \begin{bmatrix} \Phi_i & \Gamma_i \\ 0 & I \end{bmatrix}^T Q_{i,c} \begin{bmatrix} \Phi_i & \Gamma_i \\ 0 & I \end{bmatrix}$$

$\overline{W}_i$ is now written as:

$$\overline{W}_i = h \begin{bmatrix} \Phi_i & \Gamma_i \\ 0 & I \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} \Phi_i & \Gamma_i \\ 0 & I \end{bmatrix}$$

$$+ \begin{bmatrix} (hA\Phi_i)^T \\ (hA\Gamma_i + hB)^T \end{bmatrix} S_{i+1} [\Phi_i \quad \Gamma_i] + \begin{bmatrix} \Phi_i^T \\ \Gamma_i^T \end{bmatrix} S_{i+1} [Ah\Phi_i \quad hA\Gamma_i + hB]$$

Let W be

$$W_i = \begin{bmatrix} I & -L_i^T \end{bmatrix} \overline{W}_i \begin{bmatrix} I \\ -L_i \end{bmatrix} = h \begin{bmatrix} \Psi_i^T & -L_i^T \end{bmatrix} \begin{bmatrix} \Phi_i & \Gamma_i \\ 0 & I \end{bmatrix} \begin{bmatrix} \Psi_i \\ -L_i \end{bmatrix}$$

$$+ \left(h\Psi_i^T A^T - hL_i^T B^T \right) S_{i+1} \Psi_i + \Psi_i^T S_{i+1} \left(hA\Psi_i - hBL_i \right)$$

we get therefore

$$\frac{dS_i}{df_i} = \Psi_i^T \frac{dS_{i+1}}{df_i} \Psi_i + W_i$$

and $\dfrac{dJ}{df_i}$ may be calculated.

$\square$