SAMPLING PERIOD ASSIGNMENT FOR NETWORKED CONTROL SYSTEMS BASED ON THE PLANT OPERATION MODE

Daniel A. Perez*, Ubirajara F. Moreno*, Carlos B. Montez*, Tito L. M. Santos*

*PGEAS - Programa de Pós-Graduação em Engenharia de Automação e Sistemas
DAS - Departamento de Automação e Sistemas, UFSC - Universidade Federal de Santa Catarina
Caixa Postal 476, CEP 88040-900, Florianópolis, Santa Catarina, Brasil

Emails: `dap@das.ufsc.br, moreno@das.ufsc.br, montez@das.ufsc.br, tito@das.ufsc.br`

Abstract— In this paper, a co-design methodology for networked control systems (NCS), based on the feedback scheduling theory is proposed. In the proposed approach, the run-time information of the controlled process in employed to dynamically reassign the computational resources. The policy is to assign a lower bound of resources for each control loop for plants operating in the steady state, and allocate the exceeding resources to control loops of plants that are in a transient behavior.

Keywords— Dynamic Resource Allocation, Networked Control System

Resumo— No presente artigo, uma metodologia de co-design para um sistema de controle via redes (NCS), baseada na teoria de escalonamento via realimentação é proposta. Na abordagem sugerida, são utilizadas informações do processo controlado em tempo de operação para realocar dinamicamente recursos computacionais (largura de banda). A política se dá com a determinação de um limite inferior das necessidades de recursos para cada malha de controle para as plantas que estão operando em regime permanente. Assim os resursos excedentes nas malhas em regime permanente são alocadas para as malhas em regime transitório.

Palavras-chave— Alocação Dinâmica de Recurso, Sistemas de Controle via Redes

1 Introduction

Many approaches for resource allocation in networked control systems are based on fixed parameters of the network's load. They are defined and settled at the initialization stage, and they are kept fixed during all the operation time of the system. At execution time, in general, the resources are shared among the control loops according to static specifications.

As discussed (Martí et al., 2002), it is not necessary to assign the same amount of resources that is demanded to reject a perturbation or a set point change in order to maintain a plant in steady state. This statement suggests that to keep the same distribution of resources during all the time may be seen as a waste of resource in a networked control system (NCS).

In this paper, a control-scheduling co-design methodology that regards the plant output behavior is proposed. The proposed approach employs run-time information from the controlled process to dynamically reassign computational resources (sampling period). The principle of the procedure is to allocate the exceeding bandwidth to those control loops that have their respective plants in transient response.

This paper is organized as follows. In Section 2 the general problem is described and the concept of feedback scheduling and the procedure used to evaluate the control quality degeneration are presented . The co-design methodology of NCS is exposed in section 3. In the Section 4, an illustrative example of the methodology is presented. Finally, this paper is concluded in the Section 5.

2 Problem and Concepts

2.1 Problem

The problem studied in this paper is the real-time control of a set of processes with controllers implemented in a remote computer, interconnected through a computer network with limited bandwidth. There is a set of m continuous plants to be controlled. Associated to each process i, where $i = 1, 2, \ldots, m$, there are two devices physically connected, the sensor i and the actuator i, and a remote component, the controller i.

It is considered the situation which execution time is not an accessible parameter, hence jitter compensation techniques are not considered. The plant is described by the continuous-time linear system $P_i(s)$, the plant output is sampled periodically with interval h by the sensor S_i. The controller is represented by the discrete-time linear system $K_i(z)$, followed by the actuator A_i that includes a zero order hold.

2.2 Problem Modeling

A control cycle can be modeled by a real-time end-to-end task segmented in subtasks with precedence constraints (Sun, 1997). The segmentation of a control cycle T_i^{cc} could be off-line analyzed, and it is divided in three subtasks: sensor-controller $T_{i,1}^{cc}$ message; computation of the control law $T_{i,2}^{cc}$ and controller-actuator $T_{i,1}^{cc}$ message, as presented in Figure 1).

In (Cervin and Eker, 2000) the *feedback scheduling* was proposed. The main idea is to distribute computational resources to optimize the

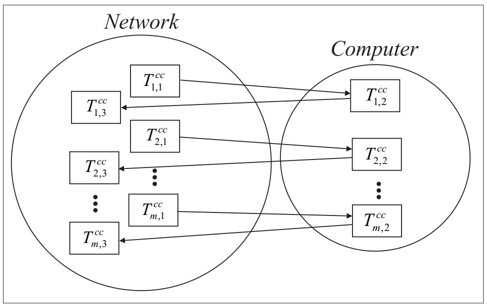


Figure 1: Modeling of a set with m NCSs.

global performance of the control, in a processor susceptible to overloads.

In the feedback scheduling approach, the scheduler feedbacks the consumption of the resources of the system (e.g. execution times of tasks) to determine the load of the system. System parameters, such as periods and priorities of tasks, are reconfigured to lead the utilization to a specific level of reference. If some task comes to overrun, the scheduler may detect the overload and reconfigure the tasks to deal with the overload.

In previous works a scheduling algorithm called *Feedback Control EDF* is presented (Stankovic et al., 1999; Lu et al., 1999). This approach consists in an implementation of a PID controller in the scheduler, that regulates the deadline miss rate for soft real-time tasks with variable execution time, through the adjustment of the processor's utilization. Another approach is used in (Beccari et al., 1999) where sampling intervals are assigned during run-time to prevent overload of the processor. In this same context (Henriksson and Cervin, 2005) can be cited, where tasks reassignment in overload conditions was used. However in the majority of previous work, the plant operation mode was not evaluated for a NCS.

2.3 Estimation of the Control Performance Degeneration

The use of a shared network among the components of a control loop introduces variable delays (delay jitter) in the execution of the control cycle. This uncertainty leads the studied control loops into time-varying systems, disallowing the direct use of linear systems criteria to evaluate the degeneration of the system's stability margins.

On the other hand, there are some criteria to measure the degeneration of the stability margins by implementation factors in linear time-invariant systems. Equations (1) and (2), evaluate the phase lag due to the controller discretization and to the constant delay in the control cycle. $\Delta\varphi_m$ is the sum of the degeneration factors sum.

$$\Delta\varphi_{m(d)} = \frac{\omega_c h}{2} \quad (1)$$

$$\Delta\varphi_{m(a)} = \omega_c L \quad (2)$$

$$\Delta\varphi_m = \Delta\varphi_{m(a)} + \Delta\varphi_{m(d)} \quad (3)$$

An approach to deals with NCS is to turn it into a time-invariant problem, using buffers in the controller and actuator nodes to reduce delay jitters (Luck and Ray, 1990). The system becomes time-invariant, when the release time in buffer is longer than the worst-case response time of transmission and computation of the message between the nodes of the control loop. The main drawback of changing a NCS in a time-invariant system is the unnecessary increase of the control delay, because the average of the delays becomes equal the worst-case response time.

3 The co-design methodology

Using the feedback scheduling concept, adapted to the NCS context, it is possible to assign more computational resources for the control loops that are demanding a better quality of control in each moment of the system operation.

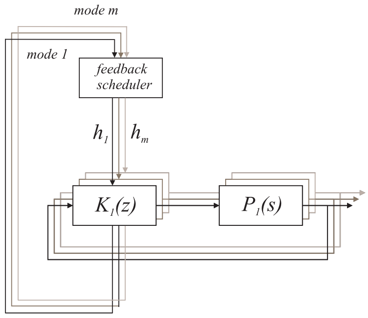


Figure 2: Dynamic allocation of resources through the feedback scheduling.

Differently of the majority of the works carried through on feedback scheduling, the control variable in this proposal is the operation mode of the plants. Thus, the idea is to apply feedback in two levels of the real-time control system, as presented in the Figure 2. There is a standard feedback used by controllers, and a second level that represents the feedback inside of the real-time system to assign dynamically the computational resources between the control loops. The distribution of the resources is based on the current situation of operation of the controlled plants.

The principle of the adopted resources allocation in the methodology comes from (Martí et al., 2002), that suggests that the plant needs different degrees of computational resources to be controlled. The process needs less amount of resources when it is in steady state than in the situations that the control system is rejecting a disturbance or is responding to change in the reference.

Following this approach, operation of the control loops was divided in two modes, *Transient Mode* and *Steady Mode*. In *Transient Mode* all the surplus of resources available will be assigned to the control loop with a settling process in the instant of the evaluation. On the other hand, in *Steady Mode* the minimum amount of resources allowed, kept a lower predefined stability margin, is assigned the control loop with the plant in steady state in the instant of the evaluation.

Each control loop can be switched between these two operation modes during the run-time of the system, in accordance with the actual state of the plant. During the control procedure of the plant, the controller receives the samples of the plant's outputs, which are computed, resulting in the control law. In the other feedback loop, used by the feedback scheduler, at each new control cycle the controller verifies the actual plant state. If the states of the plant have been modified, since the last evaluation, the controller detects this change and feeds the feedback scheduler.

The change in the resources distribution is done through the modification of the sampling periods h_i of each control loop, therefore the quality of control degenerative factors are directly associated to this parameter.

3.1 Definitions and Assumptions

Some assumptions and definitions must be made before the procedure description. First, a technique to transform a NCS in a time-invariant system is applied to obtain a capable metric to give a value of the quality of the control degeneration in each control loop of the system. Thus, (1) and (2) become valid. In order to facilitate the execution of the network and computer schedulability tests, it is also assumed that the maximum time of the control delay R never is greater than the value of the sampling interval h. The assignment method runs every time that a controller detects that its respective controlled plant changes its operation mode.

Considering that, there is a set of m NCS's, each time that the feedback scheduler runs to reconfigure the allocation of resources, the set of control loops is divided in two subsets: TM and SM. The subset denoted by TM is composed by the control loops operating in a transient state, and the subset denoted by SM is composed by the control loops operating in a steady state. The number of elements of each subset is denoted respectively by tm and sm.

The phase margin of the system after its implementation in a NCS and its relative degeneration λ are defined as:

$$\varphi_m^{NCS} = \varphi_m - \Delta\varphi_m \tag{4}$$

$$\lambda = \frac{\varphi_m^{NCS}}{\varphi_m} \tag{5}$$

In accordance with the operation modes, λ_{tm} and λ_{sm} are defined as the relative degeneration of the phase margin of the control loops that belongs to the subsets TM and SM respectively.

By denoting the utilization of the processors by the set of control loops that belong to the system as U_c, the utilizations related to the subsets TM and SM are denoted, respectively, by U_{SM} and U_{TM}.

3.2 Sampling Periods Assignment

The employed methodology consists in assign sampling intervals h such that the control loops which are operating in the same operation mode have the same relative degeneration of the phase margin λ. The sampling interval h of a NCS can be related to its relative degeneration of the phase edge λ.

In accordance with the assumptions, let $L = h$. Thus, the sum of the degeneration factors $\Delta\varphi$ is expressed as in (6) and ω_c is the cross-over frequency.

$$\Delta\varphi_m = \frac{3}{2} \cdot \omega_c \cdot h \tag{6}$$

On the other hand, the sum of the degenerative factors $\Delta\varphi_m$ can be related with the relative degeneration of the phase margin λ, hence from (4) and (5):

$$\Delta\varphi_m = \varphi_m(1 - \lambda) \tag{7}$$

From (6) and (7), the value of h can be obtained.

$$h = \frac{2}{3} \cdot \frac{\varphi_m(1-\lambda)}{\omega_c} \tag{8}$$

Thus, the sampling period assignment for each control loop in the subset SM is done through (8) and by the project parameter λ_{sm}.

The utilization U_{SM} is computed through the sum of the individual utilizations (e_{SM}/h_{SM}) of each control loop in the subset SM and e denotes the execution time. By applying (8), U_{SM} is given by

$$U_{SM} = \frac{3}{2} \cdot \sum_{i=1}^{sm} \frac{e_{SM,i}\omega_{c\ SM,i}}{\varphi_{m\ SM,i}(1 - \lambda_{SM})} \tag{9}$$

Once determined the computational resources consumed by the control loops in steady state, by applying (9), the remaining computational resources are allocated to the control loops in the subset TM, thus, the utilization U_{TM} is given by:

$$U_{TM} = U_c - U_{SM} \tag{10}$$

By developing a similar derivation for the subset TM, as for the subset SM, an expression that relates the utilization U_{TM} and λ_{tm}, could be given by

$$U_{TM} = \frac{3}{2} \cdot \sum_{i=1}^{tm} \frac{e_{TM,i}\omega_{c\ TM,i}}{\varphi_{m\ TM,i}(1-\lambda_{TM})} \tag{11}$$

Differently of the sampling periods assignment in the subset SM, the utilization U_{TM} is the known parameter and the relative degeneration of the phase margin λ_{tm} is the parameter to be calculated. The value of λ_{tm} can be obtained iteratively, starting form an initial value $\lambda_{tm} = \lambda_{sm}$. As described in the proposal of the method, admitting that the component loops of the subset TM will have a smaller degeneration that the ones which form the subset SM, thus $\lambda_{TM} > \lambda_{SM}$ for all the considered cases.

Once the value of λ_{tm} is estimated, the assignment of the sampling period for each NCS k in TM is given by (8).

Thus, the sampling periods of all the control loops in the system are assigned every time that the feedback scheduler is executed.

3.3 Modeling of the System Reconfiguration Procedure

The system reconfiguration can be described in the following way. In each activation, the feedback scheduler determines a new sampling interval for each control loop. The feedback scheduler, then, brings up to date the sampling rates of each controller through the communication between processes in the computer, and brings up to date the sensors and actuators of the NCS through the broadcast of a message containing the new set of sampling intervals, for all the remote devices that compose the system.

The reconfiguration procedure can be modeled as a real-time end-to-end task T^r, subdivided in two subtasks. First subtask T_1^r is composed by the execution of a procedure that computes the new values of the sampling intervals and by the update of these values in the controllers, executed in the computer. The second subtask T_2^r is constituted by the sending of a message, through the communication network, containing the update to the sensors and remote actuators that belong to the system.

Differently of the control cycles, the reconfiguration task of the system is modeled as an aperiodic real-time task. Thus, the release of the feedback scheduler instances has a non periodic behavior. The activations are caused by the occurrence of disturbances or changes in the reference signal that, in general, are events that occur without a predetermined pattern.

The incorporation of the system reconfiguration in the schedulability analysis, and the determination of the worst-case response times of the control cycles, can be made using aperiodic real-time tasks scheduling techniques for the subtask executed in the computer. To consider the aperiodic messages in the communication network, the approach will vary in accordance with the implemented network. For the case of a CAN network, presented in the following topic, is possible to use the *Deferrable Server* (Lehoczky et al., 1987) scheduling strategy.

3.4 Application in a CAN network

A CAN network can be scheduled as a fixed priority non preemptable real-time processor. Thus, established scheduling techniques can be used to compute the schedulability and the response times of a set of real-time messages transmitted through the network.

The modeling of the studied problem looks like the presented in Figure 1, with control cycle end-to-end tasks and the reconfiguration end-to-end task.

The utilization of the network that guarantees the schedulability is given by

$$U_i + u_s + \frac{e_s + b_i}{h_i} \leq U_{RM}(i+1) \tag{12}$$

where U_i and u_s are the periodic tasks and the deferrable server utilizations, respectively, and b_i is the blocking time.

4 Simulations

To illustrate the application of the approach, an example is presented. The example is composed by three control loops, whose controllers are implemented in a remote computer and using the same CAN network to exchange the necessary information to the control. In the computer, the adopted scheduling strategy is rate monotonic. Considering a CAN network with the biggest transmission rate for this technology, $1 Mbits/s$, and messages with constant and equal size 120 $bits$ (average size of a CAN frame), the transmission time of a message is equal $120 \mu s$. The priority assignment of the network nodes is fixed. The computer messages has the biggest priority of the network, the sensors are organized by the decreasing transmission rates when all the NCSs operate in the Transient Mode. It is assumed that the implementation execution times of each controller in the computer are constant and equal 150 μs. The continuous-time plants used in the example are given by (13).

$$\begin{aligned} P_1(s) &= \frac{900}{(s^2+42s+900)} \\ P_2(s) &= \frac{4 \cdot 10^4}{(s-50)(s+50)} \\ P_3(s) &= \frac{5 \cdot 10^7}{s(s^2+100s+2.5 \cdot 10^5)} \end{aligned} \tag{13}$$

The continuous-time controllers are given by (14).

$$\begin{aligned} K_1(s) &= \frac{500(s+70)(s+60)}{s(s+1500)} \\ K_2(s) &= \frac{8\cdot 10^3(s+2.5\cdot 10^5)(s+90)}{(s+2000)(s^2+1.645\cdot 10^4 s+1.35\cdot 10^8)} \\ K_3(s) &= \frac{478(s+2\cdot 10^5)(s^2+160.6s+1.655\cdot 10^5)}{(s+2740)(s+1000)(s^2+2494s+7.109\cdot 10^6)} \end{aligned} \qquad (14)$$

Beginning the co-design procedure, some parameters must be defined before the execution of the simulation. In accordance with rate monotonic theory, the respective utilization that guarantees the schedulability for the network is $U = 0.73$. Consequently, in accord with 12, the reload interval of the deferrable server was chosen as $h_s = 1\,ms$, and the size of the recharge is enough to send a message per cycle, $e_s = 120\,\mu s$. The value of the phase margin relative degeneration for the Steady Mode was assumed to be $\lambda_{sm} = 0.35$, leading to a phase margin after the implementation of $\varphi_m^{ncs} > 20°$ for each mesh.

The proposal, presented in this paper, is characterized by the dynamic change of the resources of allocation, according to the operation mode of the set of NCS. To evaluate possible benefits of the proposal, three cases are proposed in order to explore the operational limits of each plant and the evolution of scenarios in which the assignment of the periods are modified.

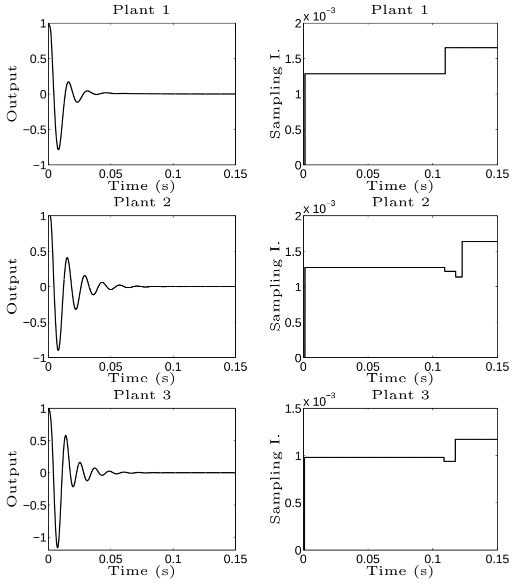
Case 1 - The plants are initially at steady state, the reference of all plants are changed to 0. The goal is to observe the control loops operating, together, in the Transient Mode.

Case 2 - The same reference change of **Case 1** is applied for NCS 2, while the control loops 1 and 3 are kept operating at the steady state (Steady Mode). The objective in this simulation is to place the biggest amount of resources for the control loop 2.

Case 3 - In this scenario, the goal is to change the operation modes of all NCS to show some reconfiguration procedures of the system. To do that, the same disturbance is applied, in $t = 0.03s$ and $t = 0.14s$, in the control loop 2; and a reference change is imposed to NCS 3 at $t = 0.16s$. **Case 1** for NCS 1.

After the simulation of **Case 1**, the results obtained are presented in the Figure 3. This is the configuration which each NCS receives less amount of resources when operating in the Transient Mode. Consequently, the greatest closed loop performance degeneration occurs in this situation.

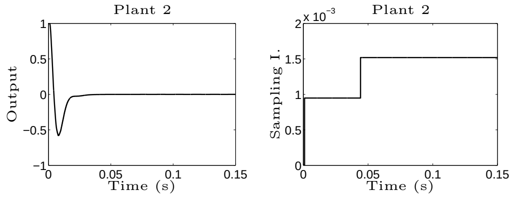
The impact in the plant 2 due to the use of the proposed methodology in **Case 2** is displayed in Figure 4. In this case, the plant 2 holds the greatest availability of resources, since it operates alone in Transient Mode (TM). The instant of commutation t_{com} and the phase margins φ_m^{ncs} of **Case 1** and **Case 2** are shown in the Table I.



(a) (b)

Figure 3: Simulation results to the **case 1**, (a) Plants responses e (b) Sampling interval (in seconds) used during the simulation.

The responses of the control loops of **Case 3** are presented in the Figure 5(a). The values of the sample periods for each NCS during the simulation are displayed in the Figure 5(b), which was



(a) (b)

Figure 4: Simulation results to the **case 2** (a) Plants responses e (b) Sampling interval (in seconds) used during the simulation.

Table 1: Switching time t_{com} and implementation phase margin φ_m^{NCS} (degrees).

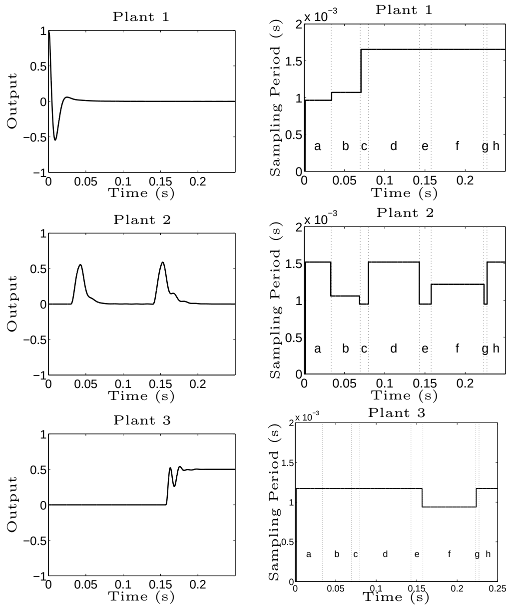
Case 1		Case 2	
$t_{com}(s)$	φ_m^{NCS}	$t_{com}(s)$	φ_m^{NCS}
0.110	28.4	–	–
0.123	28.7	0.045	37.3
0.117	30.6	–	–

Table 2: Implementation phase margin φ_m^{NCS} (degrees) for the **case 3**.

φ_m^{NCS}							
a	b	c	d	e	f	g	h
37.1	34.2	21.8	21.8	21.8	21.8	21.8	21.8
22.0	34.4	37.3	22.0	37.3	32.8	37.3	22.0
23.6	23.6	23.6	23.6	23.6	32.1	23.6	23.6

divided in eight slices (classified in alphabetical order from "a" until "h"), where the limits of each slice are instants of reconfiguration of the distribution of resources in the system. In the Table 2 the phase margins φ_m^{ncs} of the plants for each slice of simulation divided in the Figure 5(b) are presented. As changes of references and disturbances are applied in the control loops, the reassignment of the periods could be verified.

For instance, in the time slice "e", the NCS 2 operates in the Transient Mode and possess $\varphi_{m,2}^{NCS} = 37.3°$, however with the change in the reference (limit between "e" and "f") the control loop 3 control changes the operation mode and requests more computational resources of the system, reducing $\varphi_{m,2}^{NCS}$ to 32.8°. It is possible to verify the effect of the reduction of resources in the quality of control through the responses of the control loop 2, when the same disturbance is applied in the instants of simulation $t = 0.03$ and $t = 0.14$. A similar situation occurs in the slices "a" and "b", between NCS 2 and 3. In the slice "d", all the loops operate in the Steady Mode and use the lesser possible amount of resources.



(a) (b)

Figure 5: Simulation results to the **case 3**, (a) Plants responses e (b) Sampling interval used during the simulation.

5 Final Remarks

By analyzing the results carried out from the simulation examples, it is possible to conclude that, the proposed co-design methodology, presented in the paper, could lead to satisfactory results. In the proposal, results from the theories of control systems and real-systems are combined leading to significant improvements of the global performance of the control and a better use of the available computational resources. In this approach, the feedback scheduler, play a major role, allowing the system to be adaptable to instantaneous changes in operating states of the plants. In despite of the introduction of longer delays, this strategy enhance the performance of the control loops by applying a dynamic re-distribution of the sampling intervals, based on the operation mode of the controlled plants.

References

Beccari, G., Caselli, S., Reggiani, M. and Zanichelli, F. (1999). Rate modulation of soft realtime tasks in autonomous robot control systems, *Proceedings of the 11th Euromicro Conference on RealTime Systems.*

Cervin, A. and Eker, J. (2000). Feedback scheduling of control tasks, *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia.

Henriksson, D. and Cervin, A. (2005). Optimal on-line sampling period assignment for real-time control tasks based on plant state information, *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005*, Seville, Spain.

Lehoczky, J. P., Sha, L. and Strosnider, J. K. (1987). Enhanced aperiodic responsiveness in hard real-time environments, *Proceedings of the IEEE Real-Time Systems Symposium*, pp. 261–270.

Lu, C., Stankovic, J., Tao, G. and Son, S. H. (1999). Design and evaluation of a feedback control edf scheduling algorithm, *Proceedings of the 20th IEEE RealTime Systems Symposium*, pp. 56–67.

Luck, R. and Ray, A. (1990). An observer-based compensator for distributed delays, *Automatica* **26**(5): 903–908.

Martí, P., Fohler, G., Ramamritham, K. and Fuertes, J. M. (2002). Improving quality-of-control using flexible time constraints: Metric and scheduling issues, *Proceedings of the 23nd IEEE Real-Time Systems Symposium.*

Stankovic, J., Lu, C., Son, S. and Tao, G. (1999). The case for feedback control real-time scheduling, *Proceedings of the EuroMicro Conference on Real-Time Systems.*

Sun, J. (1997). *Fixed-Priority End-To-End Scheduling In Distributed Real-Time Systems*, Phd thesis, Department of Computer Science, University of Illinois at Urbana-Champaign.