RT control scheduling to reduce control performance degrading

P. Albertos*, A. Crespo#, I. Ripoll#, M. Vallés* and P. Balbastre#

* Dept. of Systems Engineering and Control

\# Dept of Computer Engineering. Universidad Politécnica de Valencia, Spain

Corresponding author: P. Albertos

Ph. 34 96 3879570 Fax 34 096 387 9579 pedro@aii.upv.es

Abstract. In the framework of RT digital control, two fundamental parameters are defined, the control effort and the control action interval. The first one is related to the strength of the control that, due to the intersampling open-loop control, determines the degrading of performances under unexpected delays. The second one refers to the unavoidable delays in the multitasking environment due to interactions among the tasks. As a consequence, the scheduling policy should consider not only the tasks delays but also their influence in the control loop behavior, being calculated to minimize the overall degrading of performances.

Keywords. Control effort, control delays, scheduling, priority assignment scheme, timing jitter

1. Introduction

Usually, control algorithms are designed according to some required performances and plant characteristics, assuming almost ideal behavior of the digital system implementing the control. But, in highly performant controlled systems, the delays introduced by the digital control due to resources limitations, should be considered. This is mainly the case when the same CPU is used to run a number of control algorithms. In this multitasking environment, the scheduling must consider the effect of these delays on the different control loops.

The implementation of multiloop control systems implies the definition of a set of tasks running under timing constraints. The task scheduling is a fundamental issue in real-time control algorithm implementation. Over the last two decades a significant work has been made in order to make this effort suitable for real time applications. However there are not too much work in the integration of both phases, control design and its implementation, to get the best control performances. The real time task model based on periodic tasks has been very useful to study the scheduling implications but the specific control problems has not been considered.

Recently some works have pointed out the need of the integration of the control design and implementation. In [5], a method to consider the period range of the tasks involved in a control design as criteria to obtain an optimal use of the system resources is proposed. In this integrated approach, the implementation is tuned to execute the control tasks at the maximum frequency while ensuring the system schedulability. The result of this approach is a system with a high system (CPU) utilization.

The main concepts considered in this paper come from the two complementary design aspects. The control effort and the effect of delays in performance degrading arise in the control design [1]. On the other hand, the computing delay and the so-called Control Action Interval, their relation with the task scheme and scheduling parameters assignment (priority and offsets), and the influence of the scheduling are considered in the real time system implementation [2]. Thus, a methodology to find the appropriate scheduling to minimize the degrading of the control performances is investigated.

The task priority is different from the control viewpoint than from the real-time viewpoint. In general, most control loops have the same priority from the control point of view, although larger sampling periods could imply less stringent timing. But the larger the control effort the more sensitive to delays the control loop is. On the other hand, from the real-time viewpoint, the priority is in general related to the task period or deadline. Liu and Layland [3] proved that the rate monotonic algorithm (**RM**) is the optimal fixed priority scheduling algorithm.

The approach in [2] was to reduce the delay variance in the delivering of the control action computed by the different tasks. For that purpose, each task is partitioned into an initial or acquisition part, a mandatory part for the algorithm evaluation and a final part to deliver the action. Minimizing the final part the average delay can be incorporated in the controller design, [4]. In this case, the target is the averaged degrading of control performances which is directly related to both the control effort and the delay. Thus, the scheduling policy distributes the control action interval among the different tasks in order to minimize this index. Recently, several works enforce the strong relation between control and scheduling theories. In [5], an approach to optimize task frequencies within a certain range is presented. Also in [6] a method to minimize the output jitter under an Earliest Deadline First (**EDF**) scheduling algorithm has been reported.

The approach presented in this paper is applied to a multiloop control system. Different controllers providing similar control performances but requiring different control efforts are considered and the scheduling is tuned to minimize the control performance degrading. The obtained scheduling scheme takes into account the control design parameters (control efforts and delays) and build the best priority task scheme from the control implementation point of view.

This paper is organized as follows. First, the relation between the loop phase margin and the control effort is derived, showing the direct effect of the time delay. Then, different strategies to reduce the control action interval are discussed. Although the basic result is stated for state feedback control, simulation results also show its validity for any digitally controlled system. A number of examples are included to illustrate the main result. The synergy between control design and RT task scheduling is summarized in the conclusion section.

2. Control performances degrading by delays

The common feeling in designing a feedback control system is that time delays are a source of trouble in the implementation of the control, although this can be surprisingly different for very special systems, such as conditionally stable systems, as reported in a recent work [7].

Time delays can appear in the process and/or in the actuators, and sometimes in the sensor instrumentation if the measurement requires some sort of transportation. They can be concentrated delays or internal/state delays. But the main goal of this work is to consider delays due to the digital implementation of the control. Moreover, delays that appear in a multitasking environment. In these delays, two parts can be considered. The largest one is due to the control algorithm computation time and can be estimated as a result of the scheduling. In this way it could be taken into account at the control design stage using the classical approaches, such as the well-known Smith predictors or the advanced state estimators, [4]. The second part is not well defined and depends on the task interaction, as later on discussed.

The effect of the delay is not the same in any control task. It is clear that open-loop control system performances are even not very sensitive to time delays. On the other hand, in feedback control systems, the controlled system behavior can be highly degraded if time delays are not included in the design approach. But the main problem is that in controlled systems that seem to be closed-loop well damped and stable may present stability problems if significant time delays happen. In order to evaluate the time delay effect a new concept, the control effort, has been introduced [1]. The point here is that if a great control effort has been used to get the nominal behavior very short time delay becomes significant.

In order to better express this concept, a state feedback control is assumed, although the results are easily extended to a general feedback control loop, [8], and will be practically shown by some examples. Assume that the process is modeled by:

$$\dot{x}(t) = Ax(t) + bu(t) \tag{1}$$

where A is the system matrix dim nxn, and b is the input matrix, dim nx1. Without any loss of generality, a controllable canonical form is assumed. Denote by $a(s)$,

$$a(s) = |sI - A| = \prod_{i=1}^{n}(s - a_i)\,; \qquad A_n = -\sum_{i=1}^{n} a_i \tag{2}$$

the characteristic polynomial of A, where $a_i, i = 1,\dots n$ are the open-loop poles of the system, and A_n is the s^{n-1} coefficient.

Assume that, as a result of a pole assignment or optimization approach, a linear state feedback control law is designed, such as:

$$u(t) = -kx(t) + r(t)\,; \qquad \dot{x}(t) = (A - bk)x(t) + br(t) \tag{3}$$

Denote by

$$p(s) = |sI - A + bk| = \prod_{i=1}^{n}(s - p_i)\,; \qquad P_n = -\sum_{i=1}^{n} p_i \tag{4}$$

the closed-loop characteristic polynomial, where $p_i, i = 1,\dots n$ are the closed-loop poles of the controlled system, and P_n is the s^{n-1} coefficient.

The state feedback controlled system may be easily expressed by the loop transfer function,

$$G(s) = \frac{k(s)}{a(s)} \tag{5}$$

where the coefficients of the polynomial

$$k(s) = p(s) - a(s) \tag{6}$$

are the entries of the feedback matrix k, in (3). The loop frequency response is given by

$$G(j\omega) = \frac{k(j\omega)}{a(j\omega)} = \frac{k_n(j\omega)^{n-1} + \dots + k_1}{(j\omega)^n + A_n(j\omega)^{n-1} + \dots + A_1} \tag{7}$$

Define the **control effort**, k_n, as the shift of the poles from the open to the closed loop system, that is,

$$-\sum_{i=1}^{n}(p_i - a_i) = k_n = P_n - A_n \tag{8}$$

If a delay of Δ units of time in the control action is assumed, the loop frequency response (7) is given by

$$G_\Delta(j\omega) = k(j\omega I - A)^{-1}b.e^{-\Delta j\omega} = \frac{k_n(j\omega)^{n-1} + \dots + k_1}{(j\omega)^n + A_n(j\omega)^{n-1} + \dots + A_1}e^{-\Delta j\omega} \tag{9}$$

Assume that $P_n >> A_n$, that is, the closed loop poles have been shifted well on the left, to improve the time response of the system. In order to estimate the phase margin of the controlled system (3), assuming a high cut-off frequency ω_c, it yields

$$|G(j\omega_c)| = 1, \rightarrow 1 \cong \left|\frac{k_n}{j\omega_c}\right| \rightarrow \omega_c \cong k_n = P_n - A_n \tag{10}$$

and the phase margin is

$$\psi_m = \arg(G(j\omega_c)), \quad \psi_{m,\Delta} = \psi_m - \Delta\omega_c \tag{11}$$

This result can be summarized by the following theorem.

Theorem. For a state feedback controlled plant, such as (1)-(3), define the control effort as the shifting of the poles, from open to closed loop, (8). Under a Δ time delay in the control loop, the reduction in the phase margin is proportional to the cut-off frequency ω_c, (11), and, for sufficiently strong control action, i.e. for $P_n >> A_n$ as defined in (2) (4), it is also proportional to the control effort, (10).

Remark 1: If the initial system is designed assuming a known delay, it is included in the calculation of the initial phase margin ψ_m in (11), and the same conclusion is valid for any additional delay.

Remark 2: For a given phase margin, usually desirable to be around $\pi/3$, the maximum allowable control effort is inversely proportional to the possible time delay.

3. Computation delays and control action interval

Among the different delays in a digital control implementation, [4], there is a variable time delay in delivering the control output due to the processor sharing. This interval, called Control Action Interval (CAI), is considered more critical than the other delays in the control loop because it depends on the computing load and actual number of running tasks and should be computed or estimated based on the analysis of the scheduling policy. The control designer can use the obtained result, at least as an average, to determine the appropriated control algorithm or parameters.

In [2], the effects of the output jitter induced by a monoprocessor task scheduling in a control system implementation are analyzed. The effects of the interference of higher priority task on lower priority tasks produce a strong variation of the task completion time. It implies the variation over the period of the control action. This variation causes a control performance degradation that is more important in lower priority tasks. With reference to the approach described in [5], their task design reinforces the CAI problems in lower priority tasks because they stress the system utilization and, consequently, the delay in delivering the control action is higher. In [2] a method to decrease this variation bounding the CAI by a task partitioning is proposed. The task formulation ensures a fixed delay of the control action and introduces a small variable delay range for each task. This approach drastically reduces the CAI and allows a better control algorithm behavior.

In another work [6], a method to minimize the output jitter, defined as the inter-completion time of successive jobs of the same task, has been proposed. In this approach the scheduling algorithm is the EDF and attempts to reduce the output jitter between two consecutive executions. This approach does not guarantee the minimization all over the periods, and from the control point of view, it is more relevant to bind the delay variance than to reduce the delay between two consecutive executions.

The parameters of a set of tasks with a deadline monotonic (**DM**) priority assignment and the minimum and maximum delay obtained taking into account the scheduling algorithm are shown in Table 1.

Table 1. Multitasking environment. Delays and CAI.

	WCET	Period	Min delay	Max delay	CAI
T1	10	40	10	10	0,0%
T2	20	70	20	30	14,3%
T3	25	150	25	65	26,7%
T4	35	250	50	140	36,0%

It is realized that low priority tasks have both, a higher computing time and a higher delay variation in the control action completion. For instance, in task T4 the control action is sent to actuators with a minimum delay of 50 t.u. and maximum of 140 t.u. It corresponds to a variation of 36% of the T4 period.

As already mentioned, if the delay is fixed and known, it can be incorporated in the control design. The controller becomes more complex and some loss of performances is expected. If the delay is variable, any control design will be degraded when implemented in a real time environment. Thus, the delays must be reduced as much as possible and trying to minimize their effect on the control performances.

4. Delay reduction

To reduce the variable delay of the control activities, a task partitioning is implemented, [2]. Given a schedulable set of tasks T using the DM theory, each task selected to minimize the jitter variance is split into three parts: data acquisition, algorithm evaluation and action delivery. Each part of a task will be considered as a separate task. Each new task redefines a new worst case execution time and holds the other task attributes. From task T_k three tasks T_{ki}, T_{km}, and T_{kf} are defined. The initial task of T_k is defined as

$$T_k = (C_{ki}, D_{ki}, P_{ki}, F_{ki})$$

where C_{ki} is the worst case execution time required by the initial part; D_{ki} is the deadline, P_{ki} is the period, F_{ki} is the offset (initially zero for all parts). In the same way C_{km}, C_{kf} are the worst case execution time required by the mandatory and final parts, respectively. All the tasks resulting on the T_k partition hold the same deadline and period ($D_{ki} = D_{km} = D_{kf}$; $P_{ki} = P_{km} = P_{kf}$). The proposed algorithm will modify the phase value of each task.

The priority of each new task is represented as a function of the task. Thus, Prio(T_{kf}) will denote the priority of the final part. The priority assignment is as follows:

- The priorities are grouped in three priority blocks.
- The final parts (T_{kf}) are located in the highest priority block and inside the block, according to the DM policy.
- The initial parts (T_{ki}) are assigned to the second priority block and also assigned according to the DM.
- The mandatory parts are placed in the lowest priority block and also ordered by the DM.

The proposed priority assignment intends firstly to minimize the variance of the final parts, and last to minimize the execution interval of the initial parts. The group of the mandatory parts has the lowest priority so they can not delay the execution of the final nor the initial parts. As a result of the new task set and priority assignment, the original schedulability can not be assumed and a new schedulability test is required in order to validate the new situation. Since initial parts have a higher priority than mandatory ones, they will be correctly scheduled by a priority driven scheduler. In [2] the schedulability analysis for the proposed scheme has been discussed. As a result, the CAI can be minimized, reducing its impact in the control performances.

In the previous multitasking environment, this scheme gives the results in Table 2.

Table 2. CAI reduction by task splitting

	Initial Part		Final part			Initial
	begin	end	begin	end	CAI	CAI
T1	0	4	15	16	2,5%	0,0%
T2	0	4	32	34	2,9%	14,3%
T3	0	4	64	67	2,0%	26,7%
T4	0	4	136	140	1,6%	36,0%

However, as stated in the previous sections, the control effort required by the regulator and the control action delivering delay have a strong relation with the control performance degradation. Thus, the delay reduction should be balanced not based on the task priority but on the control effort, in such a way that the degrading effect is taken into account. In section 2, a direct connection between phase margin reduction, control effort and delay has been established.

The following methodology allows integrating both design phases: the controller design, leading to a given control effort, and the control implementation, resulting in an estimated average delay.

- As result of the control design, it is possible to determine the control effort of each control loop, (8).
- The schedulability of the partitioned system is analyzed, and the minimum and maximum delays are evaluated (as in Table 1).
- If so required, the control design considers the minimum delays as a parameter to redesign the control algorithm, determining updated control efforts.
- A priority assignment algorithm based on the required control effort is applied to the task set in order to obtain a modified priority assignment. As a result, the priority of tasks with higher control effort is increased. The scheduling algorithm has to ensure the system schedulability.

The priority assignment algorithm starts from a schedulable proposal based on the RM approach. The algorithm promotes tasks with higher control effort to upper priority levels ensuring the schedulability of the task set. The promotion of the task priority involves the three tasks associated to a control loop after the partitioning process.

```
1.  Each task has initially assigned a priority
    according to the DM analysis. Let task T[1] be the
    one with the highest priority and T[n] the lowest
    priority task.
2.  FOR T[k] := highest control effort task TO lowest
    control effort task DO
3.    FOR i := k DOWNTO 2    DO
          IF (T[i].degrading < T[i-1].degrading) AND
           (is_schedulable(T[1],T[2],... T[n]) THEN
          EXCHANGE(T[i],T[i-1]);
          ENDIF
      ENDFOR
4.  ENDFOR
5.  FOR i:=1 TO n DO
      T[i].priority := i;
6.  ENDFOR
```

In fact, as shown in the example below, what is the purpose of the re-scheduling is to get a task scheduling with the minimum control performances degrading. Thus, the criterion to promote a task to a higher priority is based on the reduction of an index such as:

$$J_D = \sum_{i=1}^{N} K_i \Delta_i$$

where K_i is the control effort for the *i*-control task, Δ_i is the average delay attached to this task, and *N* is the number of control tasks. There can be some other tasks that only require some computing time and thus affect the schedulability, but do not degrade the control.

5. Examples

Based on an academic control example, first the influence of the control effort and the delay on the control performances is checked. Then, the effect of the scheduling policy to reduce the average performance degrading is illustrated.

Assume that there are two different simple systems S_1 and S_2, with transfer functions, respectively:

$$G_1(s) = \frac{100}{s+6}, \text{ and } G_2(s) = \frac{100}{s+1}.$$

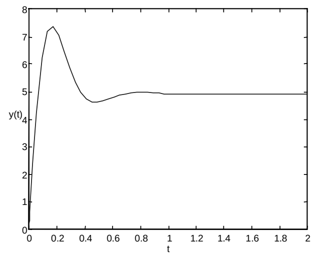


Fig. 1. Desired closed-loop controlled system step response.

As can be seen, S_1 is faster than S_2, with time constants τ_1=1/6 and τ_2=1sec, respectively. The same controlled behavior is required for both, as expressed by the closed-loop transfer function:

$$M(s) = \frac{100(s+7)}{s^2+13s+142},$$

that, graphically is shown in figure 1.

To follow the idea developed in Section 2, a feedback controller is designed such as:

$$u(s) = r(s) - G_{R,i}(s)y(s)$$

The sum of the closed-loop poles is, in both cases, $P_n = -13$, whereas the open loop poles are, respectively, $A_{n1} = -6$ and $A_{n2} = -1$. So, the control effort for S_2 should be greater than for S_1, being also more sensitive to delays. Nevertheless, in this case the controllers are dynamic and they also influence the delay effect.

The controllers required to achieve this behavior are given by

$$G_{R,1}(s) = \frac{1}{s+7} \text{ and } G_{R,2}(s) = \frac{5s+135}{100(s+7)}.$$

The performance degrading can be evaluated by means of the phase margin and the cut-off frequency, (11). They are shown in Table 3.

Table 3. Controlled systems performances

Plant	ω_c (rad/sec)	ψ_m (rad)	Δ_{max}
S_1	7,6	1,41	0,185
S_2	11	1,22	0,111

In order to show the different effect of the delay, both systems are simulated with the same delay, Δ=0.06 sec. The responses are plotted in fig. 2.

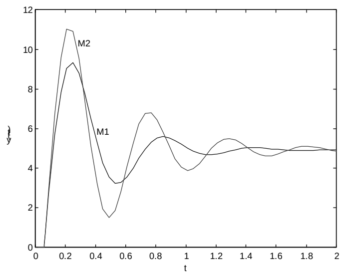


Fig. 2 Closed-loop response of S_1 and S_2 both with a delay of 0.06 sec.

As the ω_c of S_2 is greater than that of S_1 and the initial phase margin of S_2 is smaller than the one of S_1, it can be realized that the S_2 behavior is highly degraded, with lower influence in the S_1 response. The response is even worse if the delay increases, making S_2 to become unstable.

To balance delays and control effort in order to get a similar performance in closed-loop, a constant delay of 0.03 sec has been applied to S_2, while the first system is delayed by 0.08 sec, as can be seen in fig. 3.

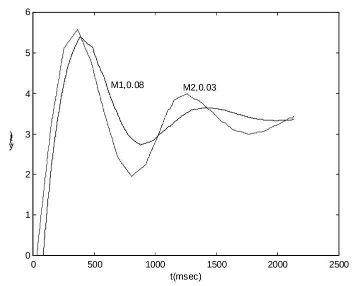


Fig. 3. Step response with an action delay of 0.08 sec for S_1 and 0.03 sec for S_2

The same conclusions can be reached if the system is digitally controlled. In this case, the control scheme is as depicted in fig. 4.

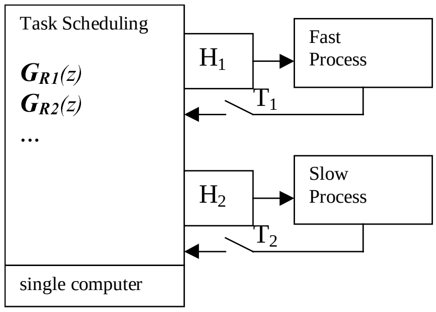


Fig. 4. Single-computer control of a multiprocess

The two processes are sampled at their own rate and the control action computed by the discretized controllers is updated as soon as available, through the hold device.

Let us suppose now that we have three control tasks where the sampling rates are taken as T_2=100 *msec* and T_3= T_4=110 *msec*. The single computer, shared by these three tasks, among some others, is scheduled to balance the performance degrading in both control loops. In the scheduling timing another task, T_1 with a period of 70 msec, is included, assuming that its delay is not relevant although conditioning the overall schedulability.

The scheduling based on the DM algorithm is shown in table 4.

Table 4. DM scheduling of the 4 tasks

	WCET	T	Δ_{min} msec	Δ_{max} msec	$\bar{\Delta}$ msec	CAI %	Contr. Effort K	Degrad. (rad) K*Δ	**TOTAL deg.**
T1	22	70	22	22	22	0,0	0	0	
T2	15	100	15	37	26,0	22,0	7,6	0,198	
T3	17	110	17	54	35,5	35,5	11	0,391	**1,309**
T4	19	110	36	95	65,5	65,5	11	0,720	

The computed degrading is not too bad, but the negative effect of the high CAI has not been considered. Thus, in some periods, the task T4 can reach a maximum delay of 95 msec (an 86,36% of the period), becoming almost unstable. If a CAI reduction technique, as applied, as in [2], the new figures are shown in Table 5.

Table 5. DM scheduling with CAI reduction

	WCET	T	Δ_{min} msec	Δ_{max} msec	$\bar{\Delta}$ msec	CAI %	Contr. Effort K	Degrad. (rad) K*Δ	**TOTAL deg.**
T1	22	70	27	28	27,5	1,4	0	0	
T2	15	100	39	41	40,0	2,0	7,6	0,304	
T3	17	110	53	56	54,5	2,7	11	0,600	**1,927**
T4	19	110	91	95	93,0	3,6	11	1,023	

In this case, the variability in the delay, as expressed by the CAI, has been highly reduced, but the average time delay has been increased and the control degrading is much higher. Again, the process related to the task T4 will be almost unstable. Now, the re-scheduling methodology is applied. The final results are shown in Table 6.

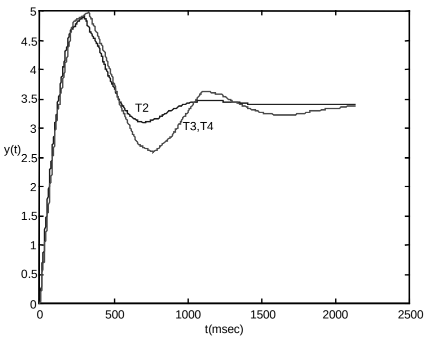
Table 6. Re-scheduling minimizing the control performance degrading

	WCET	T	Δ_{min} msec	Δ_{max} msec	$\overline{\Delta}$ msec	CAI %	**Priority**	Degrad (rad) K*Δ	**TOTAL deg.**
T1	22	70	56	60	58	5,7	**3**	0	
T2	15	100	94	97	95,5	3,0	**4**	0,726	
T3	17	110	22	23	22,5	0,9	**1**	0,247	**1,392**
T4	19	110	38	40	39	1,8	**2**	0,429	

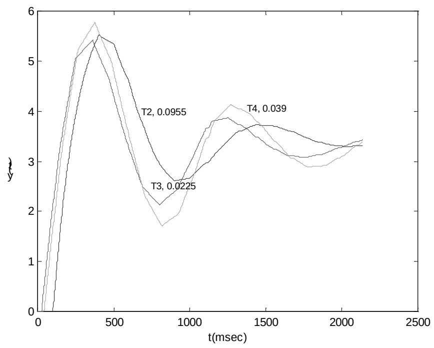
As a result, the tasks related to the control of the slow process have been promoted to the highest priority. On the other hand, to keep the system schedulable, the priority of the control task T2 cannot be higher than that of the task T1, which period is lower.

The scheduling policy allows balancing the performance degrading in both control loops. In this sense, the expected average delay in the faster open loop system is more than twice the second one, which requires a stronger control effort to achieve a similar behavior.

The simulation results are shown in fig. 5.



a)



b)

Fig. 5 Digital control: a) without delays, b) with balanced delays

6. Conclusions

Time delays degrade the control performances. If these delays are known in advance, they can be taken into account by the control algorithm. But the effect of the delay is not the same in all the control loops. The first conclusion is that the effect of the delay is strongly connected to the control effort. The control effort is defined as the shift of the poles, from the position in the process to that in closed-loop. The result is only valid in extreme conditions, but it is a rule of thumb for any feedback control system.

In digital control systems, the control signal is always delivered with some delay. In a multitasking environment, the delay is variable, depending on the tasks' interaction. The priority assigned to the tasks determines the respective delay.

In the paper, a scheduling policy to minimize the control performance degrading has been presented. As a result, the priorities can be changed, assuring the schedulability of the whole system and reducing the need of a variable tuning of the control.

Some illustrative results have been presented, the approach being under further application in laboratory systems.

References

[1] Albertos, P. and Olivares, M. (1999). "Time delay limitations in control implementation" *European Control Conference* Karlsrhue.

[2] Crespo, A., Ripoll, I., and Albertos, P. (1999). " Reducing Delays in RT control: the Control Action Interval" *IFAC World Congress*, Beijing.

[3] Liu, C. L. and Layland, J. W. (1973). "Scheduling algorithms for multi-programming in a hard real-time environment," *JACM*, vol. 20, no. 1, pp. 46-61.

[4] Albertos, P. and Crespo, (1999). "A. Real-Time Control of Non-Uniformly Sampled Data Systems". *Control Engineering Practice*.

[5] Seto D., Lheoczky J., Sha L., Shin K.G., (1996). "On Task Schedulability in Real-Time Control Systems". 17th IEEE Real-Time Systems Symp., December 4-6. Washington, DC. Pp: 13-21

[6] Baruah S., Buttazzo G., Gorinsky S., Lipari G. (1999). "Scheduling periodic task systems to minimize output jitter". Sixth Conference on Real-Time Computing Systems and Applications. HK, December 13-15, Pp: 62-69.

[7] Albertos, P. Salgado, M and Olivares, M. (2000) "Are delays in digital control implementation always bad?" *Asian Control Conference* Shanghai. July 2000.

[8] Albertos, P. and Olivares, M. (2000) "The control effort and its implications" *IFAC Workshop in Advances in Control Education*. Gold Coast, Australia, December 2000.