Subtask Scheduling and Predictive-Delay Control
Comparison and Hybridization

Zakaria Sahraoui[1], Abdenour Labed[1], Mohamed Ahmed-Nacer[2] and Emmanuel Grolleau[3]
[1]*Computer Science Department, École Militaire Polytechnique, BP 17, Bordj-Elbahri, Algiers, Algeria*
[2]*Computer Science Department, Université des Sciences et de la Technologie HOUARI BOUMEDIENE, Algiers, Algeria*
[3]*LIAS, ENSMA, Téléport 2,1, Av. Clément Ader, BP 40109, 86961 Chasseneuil Futuroscope, Cedex, France*

Keywords: Input-output Latency, Subtask Scheduling, Real-time control, Predictive-Delay, Quality of Control, Feedback-scheduling, TrueTime.

Abstract: Amongst real-time scheduling community, several methods aim at enhencing the performance of the control. Subtask scheduling is one of the embedded convenient methods that reduce the input-output latency in the control loops. The predictive-Delay control is a new method based on input-output latency prediction in order to reduce the impact of this artefact on the quality of the control. Combining both subtask scheduling and predictive delay methods can be of a great help in combatting the impairments induced by this scheduling artifact.

1 INTRODUCTION

In real-time multi-task control, the choice of performance criteria is guided by multiple design constraints. On one hand, a part of these constraints is related to control design, whereas the others rely on the real-time scheduling theory. But the challenging question is what can be the dependence between the two sides of these constraints ?

For instance, it is well-known that in control theory, selection of appropriate task periods is one of the most prevailing constraints, while in scheduling theory, the processor overload is a fundamental constraint. The choice of a processor in an embedded system is initially based on these two parameters, which means that there is a relationship between the period and the processor load. Furthermore, insuring schedulability does not necessarily mean control with high performance, and reducing the task periods is not necessarily increasing the quality of control (Sahraoui et al., 2016). More explicitly, since control tasks are of recurrent nature, the first step in control design is to identify the closed loop frequency of the controlled process which provides a first idea about the control task periods. As a matter of fact, with coarse values of the execution times in hand, an estimate of the processor load and at the same time its capacity are generally deduced from the control task periods.

In this context, some recent theories and research results may be of valuable help. For example, in (Cervin, 2003)(Sahraoui et al., 2014), it has been shown that a higher processor bound test does not necessarily lead to a better quality of control. It has also been proven that input-output latency, is a significant artifact which may deteriorate the control if it is not taken into account.

In this context, we aim through the present work at testing the quality of the control for second and third order processes under the subtask model conditions. The main points of the analysis 0out in (Sahraoui et al., 2014) are resumed to focus the variation of some parameters.

The execution time confidence interval is widened to ensure convergent behavior of the quality of the control (QC) in the simulation set. Execution-time with a wider confidence interval may also mean a mode change. This also can reveal overload situation required to highlight some scheduling artefacts. These characterizations give more in deep sight and help the reader discern between the extent of research works.

In this paper, using two case studies and intensive simulation where computing duration varies, we first show that classic FBS fails in stabilizing processes controlled by low priority tasks in case of processor overload. We show that both subtask scheduling method studied in (Cervin, 2003) and Predictive-Delay Control (P-DC) proposed in (Sahraoui et al.,

Sahraoui, Z., Labed, A., Ahmed-Nacer, M. and Grolleau, E.
Subtask Scheduling and Predictive-Delay Control - Comparison and Hybridization.
DOI: 10.5220/0005977701010109
In *Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016)*, pages 101-109
ISBN: 978-989-758-199-1

2016) improve the QC. Finally, after this analysis and comparison, we show that combining both of these methods leads to even better result.

2 PREVIOUS WORKS

Several Works have studied the scheduling and control codesign problem. They generally investigate methods either able to enhance the control performance or to recover the process stability. The real-time community have been working on this subject for 20 years. The seminal work presented in (Seto et al., 1996), solves an optimization problem based on a non linear criterion, then in (Ryu et al., 1997) other criteria are proposed for the optimization of control performance as a function of the period and the computing latency. Later, there has been suggestions to resolve other optimization problems on-line to fit the scheduling constraints as schedulability or task periods selection, like in (Robert et al., 2005) by RST & H∞ algorithms together or by the LPV method (Sename et al., 2008; Robert et al., 2010) . These solutions are referred to as the indirect feedback scheduling (FBS).

Methods that suggest priority assignment, like in (Xu et al., 2014) with the LQG method or in (Bini and Cervin, 2008; Yepez et al., 2003; Xia et al., 2006) are called direct FBS. In the class of the direct FBS we also find the solution of (Henriksson et al., 2002; Henriksson and Åkesson, 2004) based on the Predictive Control Model.

Particularly, authors in (Cervin and Eker, 2000; Cervin, 2003) have studied the impact of the scheduling jitters on the QC using the jitterbug tool (Cervin, 2003) and then those of the latencies on the QC using the TrueTime tool (Cervin et al., 2003). The authors, proposed an indirect FBS to rescale tasks periods, based on a processor load estimator. Then, this study has been taken back in-details in (Sahraoui et al., 2014)(Sahraoui et al., 2016), where it is accounted for other scheduling artefacts and constraints. For more details about feedback scheduling the reader can refer to (Sahraoui et al., 2014).

Regarding the subtask solution, it is considered by (Gerber and Hong, 1993; Gerber and Hong, 1997) in order to enhance the schedulability under fixed priority (FP) scheduling or by (Crespo et al., 1999; Albertos and Crespo, 1999; Balbastre et al., 2000) to minimize the input-output jitter. Finally, in (Cervin, 2003) the subtask scheduling is used to improve the QC.

3 TASK MODEL AND EXPERIMENTAL SETTINGS

Lets first introduce the classical task model with the associated notation: we call tasks system the set of tasks $S = \{\tau_1, \ldots \tau_N\}$ involved in a given real-time system and denote the number of tasks by N. In addition, two jobs of a task are considered perfectly interchangeable in that they perform identical treatment.

A given task τ_i is characterized by its period h_i, its observed execution-time C_i(k) at time index k, its worst execution time C_i and the date of its first arrival (or offset) O_i. The tasks systems studied in this work have implicit-deadlines (i.e., tasks must terminate before their next release). Each periodic task generates a potentially infinite set of jobs τ_i(k), where k refers to the k^{th} sampling period : every sub-request job is released every h_i time unit.

3.1 Task Division into Calculate-Output and Update-State

A typical model to get the minimum latency from the measure input to the control output is to split the controller code into two segments: Calculate-Output and Update-State. The control output is send to the process before the Update-State segment (Åström and Wittenmark, 1997), see Listing 1. We implement the P-DC with this model for two reasons:

i) to conform, in terms of matching and comparison, the P-DC method with the subtask scheduling which is based on this typical model,

ii) to check the efficiency of this method with the minimum of latencies not due to scheduling artifacts.

In Figure 1 the execution time of the Calculate-Output segment C_{co} is a rate of $C_i(k)$ (in %). This means that the delay from the jobs start time to the end of the Calculate-Output segment will be at least $C_{co} C_i$(k). However, preemption from higher priority tasks may induce a longer delay, where the time from the jobs release/arrival time until its start time is noted the sampling latency Ls and Lio is the Input-Output latency representing the C_{co} segment latency.

The second segment returns $C_{us}(\%)$ of $C_i(k)$, which is reserved to update the PID state variables. This duration can be also subject of preemption from higher priority tasks and noted by Lus as an Update-State Latency. Finally the response time latency is defined by

$$\text{Lresp} = \text{Ls} + \text{Lio} + \text{Lus}.$$

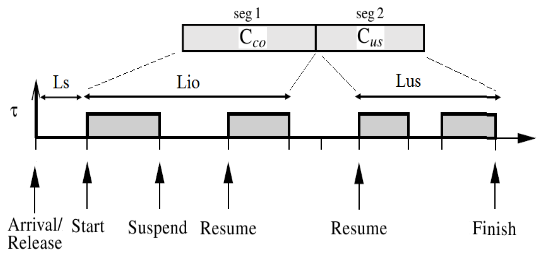

Figure 1: Task division into Calculate-Output and Update-State.

It is important to know that in the P-DC method the task scheduling is assumed by the RM scheduler under the FBS. Nevertheless, in subtask scheduling, we assign the priorities to the tasks segments (subtask model) where the scheduling is assumed with the FP protocol. This technique is proposed in (Cervin, 2003) and implemented under the TrueTime tool. The subtask scheduling method is detailed in section 4.

In the sequel, we specify the used FBS as well as the servo-motor and the pendulum processes to be controlled with a PID controller and finally specify the cost criterion of the QC.

3.2 Physical Processes

The first case study application presented in Figure 2 concerns three second order processes. It consists of three similar servo-motors, each one described by the transfer function

$$G(s) = \frac{1000}{s(s+1)} . \quad (1)$$

We define by r the reference signal, y_i the measure of i^{th} process and u_i the control send to this process. The second case study consists of three inverted-pendulum which are a convolution of the inverted pendulums, carts, motors and the pulley chain mechanisms as specified by the transfer functions (Figure 3).

The Inverted pendulum is often considered as reference benchmark in control design problems. For

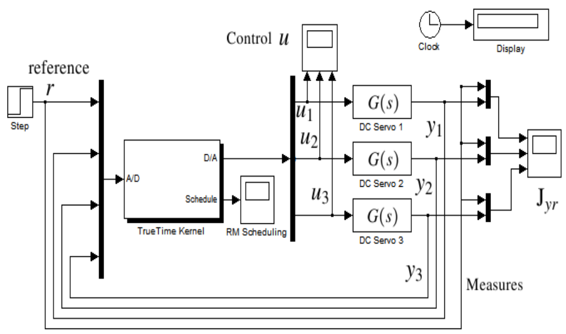

Figure 2: Three servo-motors under TrueTime/Simulink.

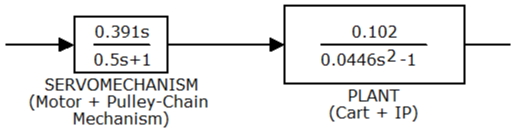

Figure 3: The inverted pendulum, version on cart.

our simulation, the pendulum starts from the center which corresponds to an angle of 0 rad. It will be constrained to an impulsion of 0.0873 rad (about 5 degrees), applied on the cart two seconds after the beginning of the simulation.

3.3 Feedback Scheduling

The FBS is used on-line, generally to supersede the off-line scheduling analysis.

Job durations of the three controller tasks τ_1, τ_2 and τ_3 are generated according to a Weibull distribution as in (Sahraoui et al., 2016). This distribution is defined by three parameters : the localization parameter l which fixes the best case execution-time, the shape factor λ and the scale factor μ. Variation in task execution-times during the simulation is accompanied by task periods rescaling, in order to achieve an observed processor utilization equal to the Liu and Layland (L&L) (Liu and Layland, 1973) RM utilization bound.

At the end of each job $\tau_i(k)$, the execution time $\hat{C}_i(k)$ is smoothed by a low pass filter. The FBS relies on this value, to calculate an estimate for the CPU utilization factor $\hat{U}(t) = \sum_{i=1}^{N} \hat{C}_i(k)/h_i(t)$.

3.4 Tasks Systems

The tasks systems used in the present work are described in Tables 1 and 2, where durations are given in *ms*.

Table 1: The three servo-motors tasks system for scheduling artifacts characterization.

	h_i^{nom}	C_i	l	μ	λ
τ_1	6	4	3.1	0.0009	3
τ_2	13	4	3.1	0.0009	3
τ_3	14	4	3.1	0.0009	3

The shape factor μ is chosen high enough to ensure wide confident interval of the $C_i(k)$ values. This may not introduce processor overload situation in simulations, but such situation can occur for the subtask scheduling case.

The system defined in Table 2 is simulated with the same range of processor utilization U_i as in the three servo-motors example, where periods and execution times are both multiplied by a factor of 1.6. It is worth noting that the task sampling period never exceeds the divergence threshold of 27 ms for the servomotor and 60 ms for the inverted pendulum. These thresholds are related to the PID setting described in the next subsection.

Table 2: The inverted pendulums tasks system characterization.

	h_i^{nom}	C_i	l	μ	λ
τ_1	9.6	7.5	5	0.0014	3
τ_2	20.8	7.5	5	0.0014	3
τ_3	22.4	7.5	5	0.0014	3

3.5 PID Controller

The PID controller defined by equations (2-7) is used. This controller is developed in (Åström and Hägglund, 1995). Given the fact that we rescale periods by FBS to ensure estimated schedulability, a_d and b_d parameters are recomputed according to formulas (5) and (6). Thus, a derivative term is computed using backward differences and a low pass filter (equation (4)) is used.

$$P(k) = K(\beta * r(k) - y(t_k)), \quad (2)$$

$$I(k) = I(k-1) + K * \frac{h}{T_i}(r(k) - y(t_k)), \quad (3)$$

$$D(k) = a_d * D(k-1) + b_d * (y(t_{k-1}) - y(t_k)), \quad (4)$$

$$a_d = \frac{T_d}{N * h + T_d}, \quad (5)$$

$$b_d = \frac{N * K * T_d}{N * h + T_d}, \quad (6)$$

$$u(k) = P(k) + I(k) + D(k). \quad (7)$$

PID parameters (K, T_i, T_d, N) are tuned in a way to obtain a system closed-loop bandwidth of ω_c = 20 rd/s and a relative damping $\xi = 0.707$. This excludes the fact that the controller design and discretization may be a source of instability for the range of the sampling periods h_i. For such convergence the cost (8) has been specified to respect a threshold of 0.36. This outset for divergent costs is taken for a simulation time T_{sim}=5 ms.

$$J_{yr_i} = \int_0^{T_{sim}} |r - y_i| dt \,. \quad (8)$$

3.6 Impact of the Input-output Latency on QC

For the tasks system presented in Table1, the QC may diverge because of high input-output latency of lower priority tasks, due to preemption from tasks of higher priority level. Figure 4 confirms this behavior. The motor controlled by the task τ_3 diverges.

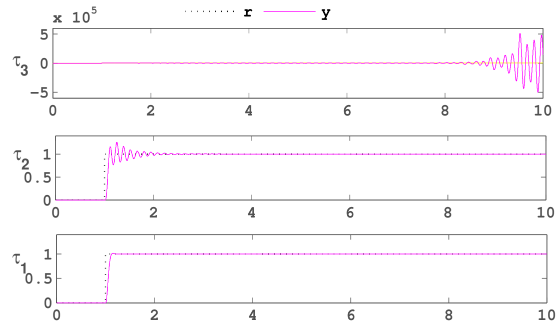

Figure 4: The three servo-motors example with the subtask model and wide range of C_i(k).

4 SUBTASK SCHEDULING

To simulate the subtask scheduling, the task model presented in subsection 3.1 is used. With a fixed priority assignment scheduling protocol, we assign the highest priority to the Calculate-Output segment (time critical part) and the lowest priority to the Update-State segment (must respect the period as deadline). It is obvious that the improvement will concern τ_3, the task which has the lowest priority.

Nevertheless, for overload situation, it can happen that τ_3 is blocked most of the time.

Scheduling of this case is shown in figure 5. The output measure y_i for each task τ_i of the tasks system defined in Table1 is shown in Figure 6.

Undesirable breaks in the diagram testify the overload situation under subtask scheduling method. In this marginal case, tasks τ_2 and τ_3 are concerned within the interval times [2.5, 3.5] and [2.5 3.8], respectively. Figure 6 shows the divergence of tasks with lower priority τ_3 and then τ_2 as a consequence to the overload situation.

4.1 Schedulability

It is noted in (Cervin, 2003) that the ideal case of subtask scheduling under FP scheduling suggests that all Calculate-Output tasks segments have higher priorities than all Update-State tasks segments. Unfortunately, such priority assignment may render the tasks system unschedulable. In cases where this approach does not work, an iterative algorithm is used. Given a schedulable original tasks system, the iterative algorithm attempts to minimize the deadlines of the Calculate-Output segments while maintaining schedulability.

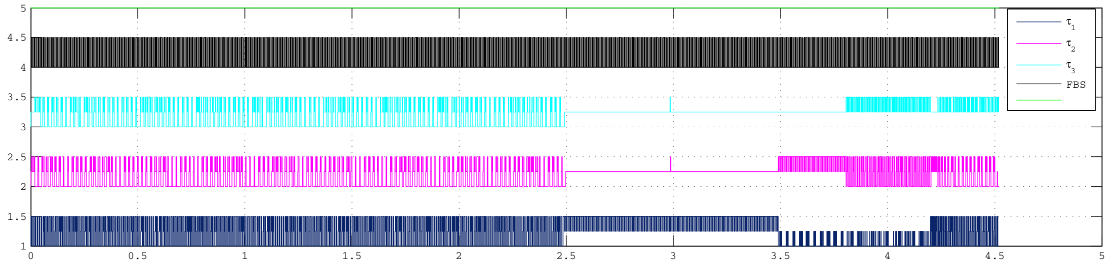

Figure 5: Scheduling diagram of the three servo-motors example with the subtask scheduling method under overload situation.

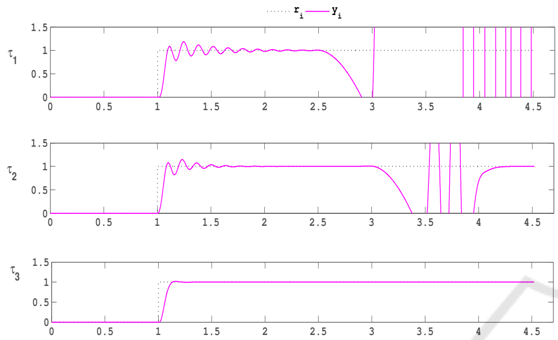

Figure 6: Output measures of the three servo-motors example with the subtask scheduling method under overload situation.

Listing 1: Implementation of subtask scheduling under fixed priority scheduling.

```
t := CurrentTime;
SetPriority(P_CO);
LOOP
ReadInput;
Calculate−Output;
WriteOutput;
SetPriority(P_US);
Update−State;
t := t + h;
SetPriority(P_CO);
SleepUntil(t);
END;
}
```

In our work, the used FBS does not care about job overruns and the basic FP implementation technique of (Cervin, 2003) is used. Since the TrueTime tool supports dynamic changes of priorities, we simply insert the TrueTime instruction “SetPriority” in the code when entering a new segment (i.e., subtask in this model), see Listing 1. Note that the priority changes may introduce additional context switches, which can degrade the performance in a real system.

It has been established in (Cervin, 2003) that the input-output latency Lio is reduced to 42 % and the used cost (an LQG function based on the control and the output signals), is reduced up to 26 %. Nevertheless, it is also noted that even if the latency is fixed and known, delay compensation can only recover part of the performance loss. This fact is illustrated by an example where the control cost of an integrator is given by $J \approx 0.79h + L$, for details, see (Cervin, 2003).

5 PREDICTIVE-DELAY CONTROL

To improve the QC, the P-DC method brings up a predicted response time latency $Lresp_i$ of the concerned task τ_i to calculate the control signal u_i. This artifice helps bypassing several practical problems like schedulability, convergence and computation time from which suffer most of proposed solutions. The method relies on an estimate $Lresp_i$, the current and the previous measures to extrapolate the forthcoming measure y_i required in the PID control calculus. Without the P-DC, the measure to be used in the PID will be obsolete. reference

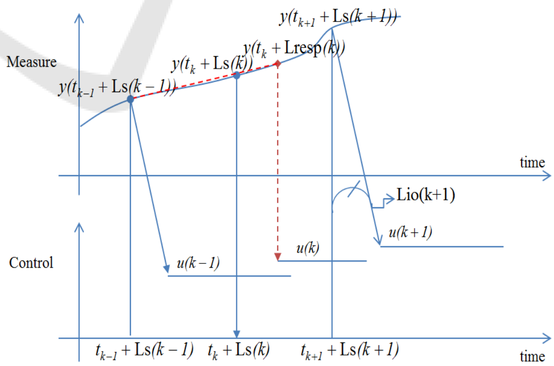

Figure 7: Predictive measures based on Lio (Sahraoui et al., 2016).

With the observed $C_i(k)$, within the overloaded case of the subtask simulation of section 4 we obtain the P-DC result presented in figures 8 and 9.

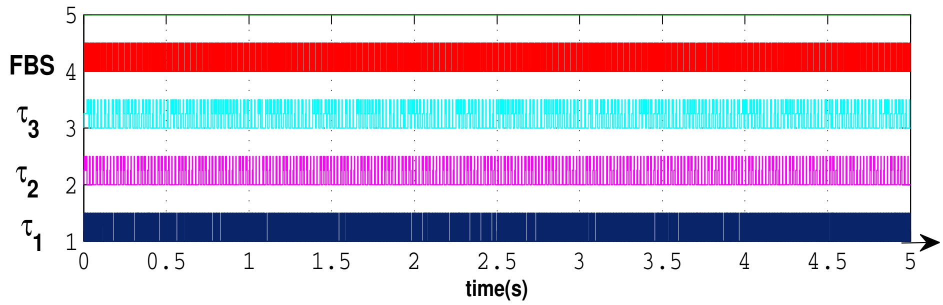

Figure 8: Scheduling of three servo-motors under P-DC with an estimate Lresp.

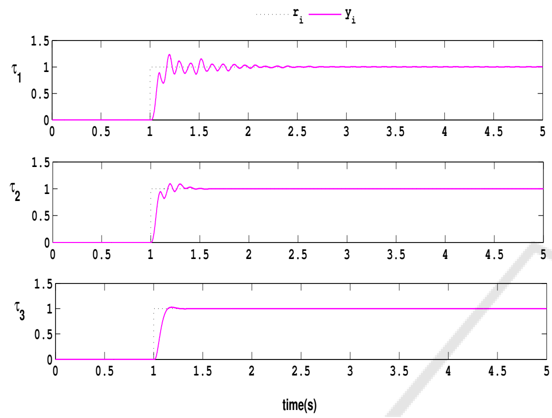

Figure 9: The three servo-motors controls converge to the set point with a low cost for an overloaded system.

6 COMPARISON AND HYBRIDIZATION

The first column in Table 3 sums up a comparison among seven solutions proposed to enhance the QC of tasks τ_2 and τ_3 without hybridization. Column two, summarize a comparison when the hybridization solution is involved.

We observe that when the hybrid solution is not involved in the comparison tests, for mild to moderate deterioration as in the case of task τ_2, or for obvious deterioration like in the case of task τ_3, using estimated Lresp (line 4,5) or its previous values (line 2 and 3), the P-DC solution may be of great help.

It is observed through more than 20000 simulations that the improvement amounts of divergent controls (e.g., Figure 10.a), based either on the previous or on an estimated Lio which are computed on the basis of the previous and an estimated Lresp, respectively is sensibly the same. Figure 10.c shows the QC improvement when using actual Lio in case of task τ_3. This result is not far from the improvement based on the subtask solution shown in Figure 10.b.

It can be concluded that subtask scheduling combined with P-DC leads to a solution that outperforms those obtained using P-DC or the subtask scheduling solely whatever the task is (Figure 10.d show the step response of the task τ_3 with a $J_{\mathrm{yr_i}}$ lower than 0.007).

Implementation of solutions based on the previous Lresp or Lio needs system calls to save the response time and eventually the sampling latency for each job termination. However, solutions with the response-time calculated on the basis of upper bounds may show significant improvements.

To verify these results, we plot the Lio impact on the QC of the 20000 samples for each technique. Figure 11.a shows the improvement of the QC when the previous value of Lio is used as an estimate. The result in Figure 11.b is based on actual Lio and is similar to the one obtained when the previous Lio is used.

The smoothed Lio in Figure 11.c, can be considered as the easiest prediction if we use a simple filter; the same as the one used to smooth the execution-time values. Figure 11.d, show the QC of the hybride solution which gives the best cost where $\mathrm{J}_{\mathrm{yr}_3}$ is always lower than $0.12 < 0.15$. In all the tested cases, it is noticed that J_{yr_3} never exceeds the value of 0.36 which is considered as a threshold in our specification (section 3.2).

It is also important to recall that, due to the overload situation, it was very difficult to accomplish the 20000 simulations samples for subtask solution.

For the example of the inverted pendulum, which is considered as a benchmark with a more sensitive cost, where $\mathrm{J}_{\mathrm{yr}} < 0.09$ for convergent control situation.

Figures 12.a and 12.b show the impact of the input-output latency on the QC for 20000 simulation samples of 5 s. The Cost J_{yr_3} converges for all the samples, which confirms the result obtained for the first example of three servo-motors.

It is also noticed that the P-DC method is more appropriate for impulse response systems like in the pendulum case.

Table 3: Summary statement in comparison and hybridization between subtask scheduling and P-DC for the case of the three servo-motors.

		τ_2		τ_3	
		Comparison	Hybridization involved	Comparison	Hybridization involved
1	Actual Lio	0%	0%	31%	1%
2	Previous Lio	0%	4%	22%	0%
3	Previous Lresp	0%	4%	0%	0%
4	Lresp^{ub}(WCET)	79%	0%	0%	0%
5	$\text{Lresp}^{ub}(\hat{C}_i(\text{k}))$	0%	0%	0%	0%
6	Smoothed $\text{Lresp}^{ub}(\hat{C}_i(\text{k}))$	0%	51%	0.5%	0%
7	Subtask only	21%	0%	46.5%	0%
8	Subtask & P-DC		41%		99%

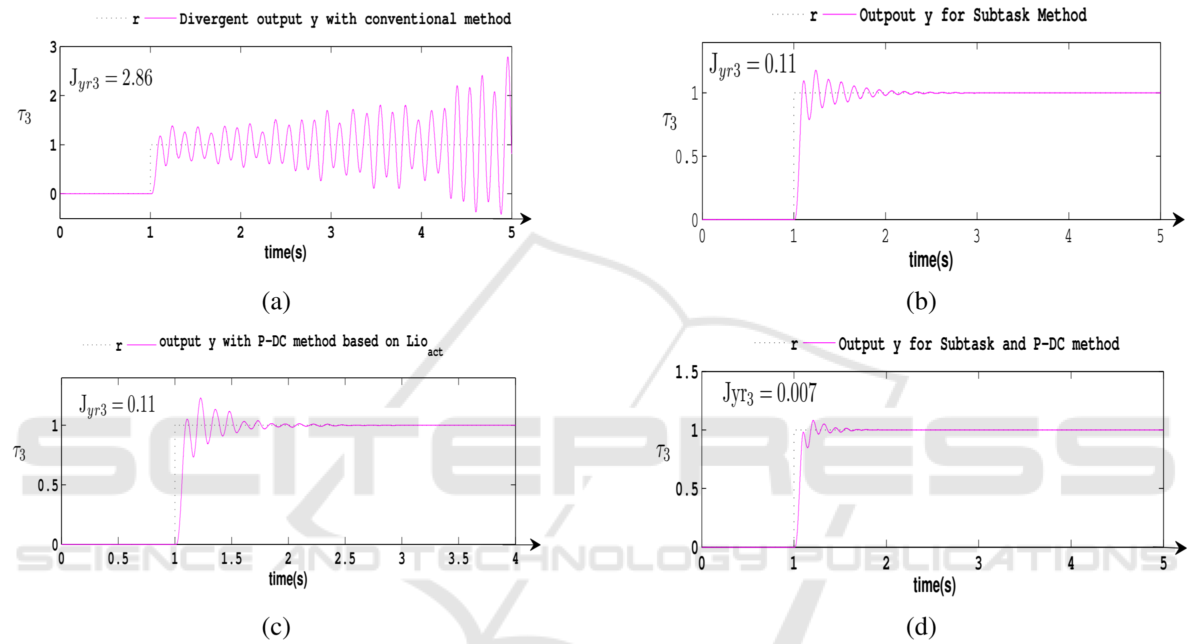

Figure 10: Output and costs for a divergent QC (a) and for improved QC (b, c, d), case of the three servo-motors.

7 CONCLUSIONS

A comparison between the P-DC and the subtask scheduling techniques is performed experimentally by simulation under TrueTime tool. We found out that the hybridization of both techniques under an FP protocol is a promising path that helps improving significantly the quality of the control.

Indeed, hybridization can suggest a better quality than a scheduling or a feedback scheduling based solely on the Predictive-Delay control. Hence, it can be deduced that the Predictive-Delay Control would be used not only to make up for scheduling latency but also to recover the control signal in overload/overrun situations. This recovering should be difficult to handle under indirect feedback scheduling or any other scheduling algorithm like the subtask scheduling techniques.

To sum up concluding remarks; reducing the input-output latency, through a subtask scheduling technique, can help boosting the P-DC method.

For further works, we can compare the P-DC technique with other methods like the control server (Aminifar et al., 2013) or the subtask scheduling under the Earliest deadline first (EDF) scheduler, where some other techniques to avoid overruns or overload situations are suggested.

REFERENCES

Albertos, P. and Crespo, A. (1999). Real-time control of non-uniformly sampled systems. *Control Engineering Practice*, 7(4):445–458.

Aminifar, A., Bini, E., Eles, P., and Peng, Z. (2013). Designing bandwidth-efficient stabilizing control

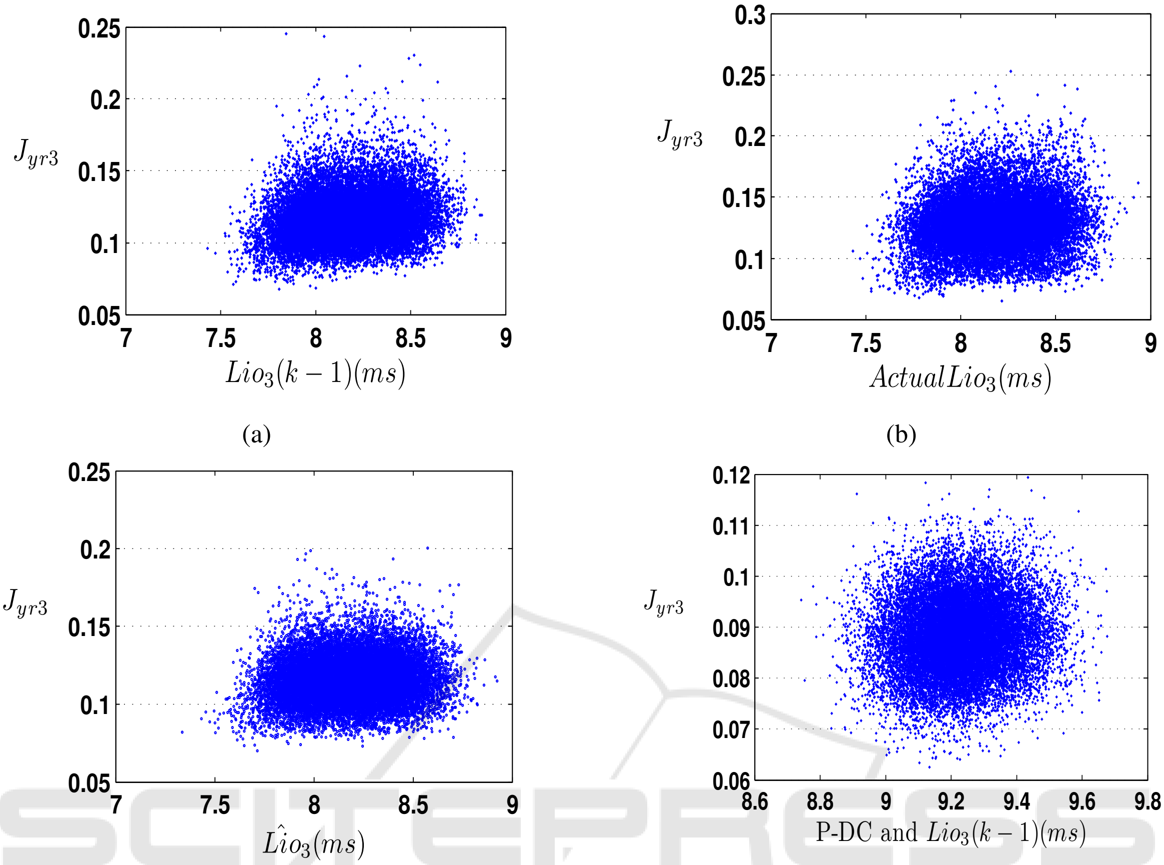

Figure 11: Improved QC and performances comparison between proposed solutions, case of the three servo-motors

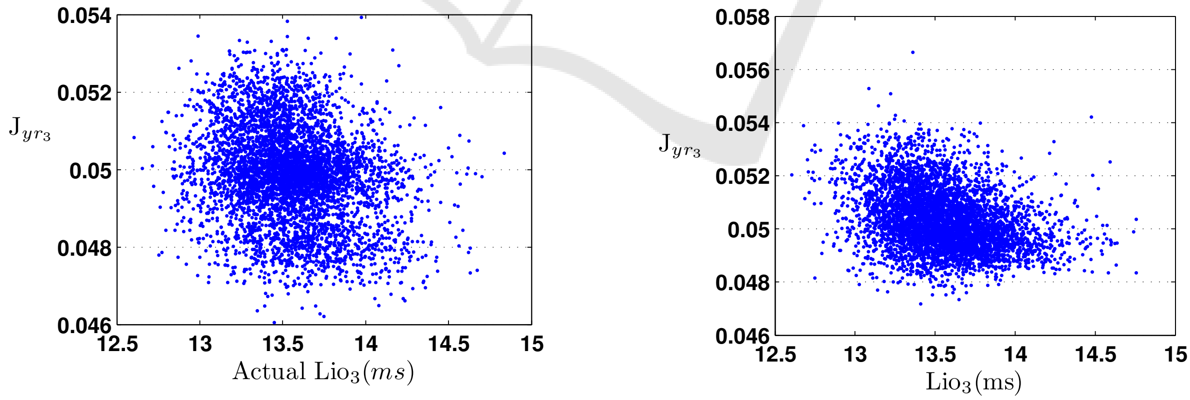

Figure 12: Improved QC and performance comparison, case of the three Inverted Pendulum.

servers. In *Proceedings of 34th IEEE Real-Time Systems Symposium (RTSS)*, pages 298–307, Vancouver, British Columbia. IEEE.

Åström, K. J. and Hägglund, T. (1995). PID controllers : Theory, design, and tuning - 2nd ed. *Instrument Society of America, 1995.*

Åström, K. J. and Wittenmark, B. (1997). *Computer-controlled Systems (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Balbastre, P., Ripoll, I., and Crespo, A. (2000). Control tasks delay reduction under static and dynamic scheduling policies. In *Real-Time Computing Systems and Applications, 2000. Proceedings. Seventh International Conference on*, pages 522–526. IEEE.

Bini, E. and Cervin, A. (2008). Delay-aware period assignment in control systems. In *Proceedings of the Real-Time Systems Symposium (RTSS)*, pages 291–300, Barcelona, Spain. IEEE.

Cervin, A. (2003). *Integrated Control and Real-Time Scheduling*. PhD thesis, Lund University.

Cervin, A. and Eker, J. (2000). Feedback scheduling of control tasks. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 5, pages 4871–4876 vol.5.

Cervin, A., Henriksson, D., Lincoln, B., Eker, J., and Årzén, K.-E. (2003). How does control timing affect performance ? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine*, 23(3):16–30.

Crespo, A., Ripoll, I., and Albertos, P. (1999). Reducing delays in rt control: the control action interval. In *Proceedings of the 14th IFAC World Congress*, pages 257–262.

Gerber, R. and Hong, S. (1993). *Semantics-based compiler transformations for enhanced schedulability.* Citeseer.

Gerber, R. and Hong, S. (1997). Slicing real-time programs for enhanced schedulability. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 19(3):525–555.

Henriksson, D. and Åkesson, J. (2004). *Flexible Implementation of Model Predictive Control Using Suboptimal Solutions*. Institutionen för reglerteknik, Lunds tekniska högskola. Lund University.

Henriksson, D., Cervin, A., Åkesson, J., and Årzén, K.-E. (2002). On dynamic real-time scheduling of model predictive controllers. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 1325–1330, Las Vegas, NV.

Liu, C. L. and Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61.

Robert, D., Sename, O., and Simon, D. (2005). Sampling period dependent RST controller used in control/scheduling co-design. In *Proceedings of the 16th International Federation of Automatic Control (IFAC) World Conference*, Czech Republic.

Robert, D., Sename, O., and Simon, D. (2010). An H_∞ LPV design for sampling varying controllers experimentation with a T-inverted pendulum. *IEEE Transactions on Control Systems Technology,*, 18(3):741–749.

Ryu, M., Hong, S., and Saksena, M. (1997). Streamlining real-time controller design : From performance specifications to end-to-end timing constraints. In *Proceedings of the Third IEEE Real-Time Technology and Applications Symposium (RTAS)*, pages 91–99, Montreal, Canada.

Sahraoui, Z., Grolleau, E., Mehdi, D., Ahmed-Nacer, M., and Abdenour, L. (2016). Predictive-delay control based on real-time feedback scheduling. *to appear in Simulation Modelling Practice and Theory under DOI: 10.1016/j.simpat.2016.02.013*.

Sahraoui, Z., Grolleau, E., Nacer, M. A., Mehdi, D., and Bauer, H. (2014). Antinomy between schedulability and quality of control using a feedback scheduler. In *Proceedings of the 22nd International Conference on Real-Time Networks and Systems (RTNS), Versaille, France, October 8-10*, page 171. ACM.

Sename, O., Simon, D., and Ben Gaïd, M. E. M. (2008). A LPV approach to control and real-time scheduling codesign : application to a robot-arm control. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, pages 4891–4897, Cancun, Mexico.

Seto, D., Lehoczky, J. P., Sha, L., and Shin, K. G. (1996). On task schedulability in real-time control systems. In *Proceedings of the 17th IEEE Real-Time Systems Symposium*, pages 13–21, Washington, DC, USA.

Xia, F., Dai, X., Sun, Y., and Shou, J. (2006). Control oriented direct feedback scheduling. *International Journal of Information Technology*, 12(3):21–32.

Xu, Y., Årzén, K.-E., Bini, E., and Cervin, A. (2014). Response time driven design of control systems. In *Proceedings of the 19th International Federation of Automatic Control (IFAC) World Congress, Cape Town, South Africa*.

Yepez, J., Fuertes, J., and Martí, P. (2003). The large error first (LEF) scheduling policy for real-time control systems. In *Proceedings of the Real-Time Systems Symposium WIP*, pages 63–66, Cancun, Mexico.