

Key aspects for co-designing real-time and control systems*

Manel Velasco, Pau Martí, Rosa Castañé, Ricard Villà and Josep M. Fuertes

Automatic Control Department, Technical University of Catalonia

Pau Gargallo 5, 08028 Barcelona

{manel.velasco,pau.marti,rosa.castane,ricard.villa,josep.m.fuertes}@upc.edu

Abstract

Real-time systems and control systems are designed to meet performance specifications. Real-time systems must fulfill timing constraints whereas control systems are intended to achieve certain desired closed-loop system characteristics. In real-time control systems, the interplay of real-time constraints and control specifications can be somewhat subtle. Hence, the selection of the sampling period becomes crucial to ensure both the expected real-time behavior and control performance. We examine how timing affects control and real-time performance. This study a) reveals some misleading practices, b) provides insights for accurate analysis and design, and c) claims that co-design is essential for building correct real-time control systems.

1 Introduction

The main objective of real-time computing is to ensure tasks executions within given timing constraints. [3]. In real-time systems, resource allocation and scheduling policies restrict the selection of task timing constraints, and this can impede meeting control specifications.

The goal of controller design is to achieve certain desired closed-loop system characteristics. In a formal specification of the problem, system characteristics are given through design parameters, which can be met by specifying the location of continuous time closed-loop poles [1]. When a controller is executed on a computer, models and/or designs must be transferred to digital form, where the sampling period (h) becomes a decisive design parameter. In the end, a discrete-time controller is an algorithm to be executed every sampling period. For real-time control systems, this may prevent

task set feasibility.

Therefore, the interplay of timing constraints and control specifications can be somewhat subtle. Since the selection of the sampling period is crucial, we examine how timing constraints affect control and real-time performance. To do so, we assess well-established control methods for obtaining discrete-time controllers, such as discretization of continuous-time design, or direct discrete-time design. This allows us to identify which parameters are of prime importance in analyzing and designing real-time control systems, clarifying the interaction between performance specifications of control and real-time systems.

The study we present a) reveals some misleading practices, b) provides insights for accurate analysis and design, and c) claims that co-design is essential for building correct real-time control systems.

2 Discretization of continuous-time design

One way to design a computer controlled system is to make a discrete-time approximation of a continuous-time controller.

2.1 Formal aspects

We examine this approximation when the continuous-time controller, considered as a dynamical system, is given in terms of either a transfer function or a state-space model.

2.1.1 Pulse transfer function

This technique is based on transforming the continuous transfer function $G(s)$ of the controller to a pulse transfer function $G(z)$. This is obtained by replacing the argument s in $G(s)$ by an s' that approximates the mapping of the s -plane into the z -plane. For example, the Tustin's approximation (1) implies the replacement

*This work has been partially supported by Spanish Ministerio de Ciencia y Tecnología Project ref. DPI2002-01621.

given by (2).

$$z = e^{sh} \approx \frac{1 + sh/2}{1 - sh/2} \quad (1)$$

$$s' = \frac{2}{h} \cdot \frac{z - 1}{z + 1} \quad (2)$$

Such type of approximations can be modeled as a hold circuit followed by the continuous-time system. And for small sampling periods, the transfer function of the hold circuit can be approximated by a time delay of half a sampling period, $e^{-sh/2}$ [1]. Therefore, the approximation will deteriorate the behavior of the closed-loop system.

2.1.2 State space model

Assume that the controlled system is represented by (3) where x is the controller state. And assume that the state feedback gain (4) controls system (3).

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (3)$$

$$u(t) = -Lx(t) \quad (4)$$

Controller (4) can be implemented in digital form by sampling the states and holding the control signal constant over the sampling period, or by applying more sophisticated approaches. Equation (5) gives an example of a discrete-time approximation of (4) [1].

$$\tilde{L} = L[I + (A - BL)h/2] \quad (5)$$

In any case, such approaches also deteriorate the closed-loop system response [1].

2.2 Illustrative example

Consider an inverted pendulum, whose continuous-time linearized state space model is given by (6).

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.018 & -1.370 & 0.027 \\ 0 & 0 & 0 & 1 \\ 0 & 0.027 & 17.180 & -0.347 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1.803 \\ 0 \\ -2.773 \end{bmatrix} u \\ y &= [0 \ 0 \ 1 \ 0] x \end{aligned} \quad (6)$$

2.2.1 Continuous-time controller

We specify for the closed-loop system dominant modes a damping $\xi = 0.7$ and a natural frequency $\omega_n = 1.4$ rad/s. Thus, the dominant poles are $s_1 = -1 + i$ and $s_2 = -1 - i$, and the non-dominant poles are placed arbitrarily at $s_3 = -20$ and $s_4 = -30$ (to prevent their

influence in the system dynamics)¹. Using pole placement design, the continuous-time state feedback gain is (7). The inverted pendulum's angle dynamics is shown in Figure 1 (a).

$$L = [-44.142 \ -48.725 \ -297.630 \ -50.293] \quad (7)$$

2.2.2 Continuous-time controller approximation

To apply (5), a sampling period must be selected. We use the rule-of-thumb based on the number N of samples per period of the dominating mode of the closed-loop system (8). It is recommended $N \approx 10 \div 20$ at least [1].

$$N = \frac{2\pi}{\omega_n h \sqrt{1 - \xi^2}} \quad (8)$$

Therefore, $h \approx 0.6283s \div 0.3142s$.

Gain (5) depends on the sampling period, and approximates a continuous-time controller. Figure 1 shows the dynamics of the pendulum's angle when applied (a) the continuous-time controller (7), and three approximations (9) obtained using (5) for (b) $h = 0.030s$, (c) $h = 0.035s$, and (d) $h = 0.038s$. As outlined in Section 2.1.2, even for small sampling periods, the system response deteriorates with increasing h , failing to meet the desired characteristics.

$$\begin{aligned} L &= [-9.953 \ -11.656 \ -79.071 \ -15.562] \\ L &= [-4.255 \ -5.478 \ -42.645 \ -9.774] \\ L &= [-0.836 \ -1.771 \ -20.789 \ -6.301] \end{aligned} \quad (9)$$

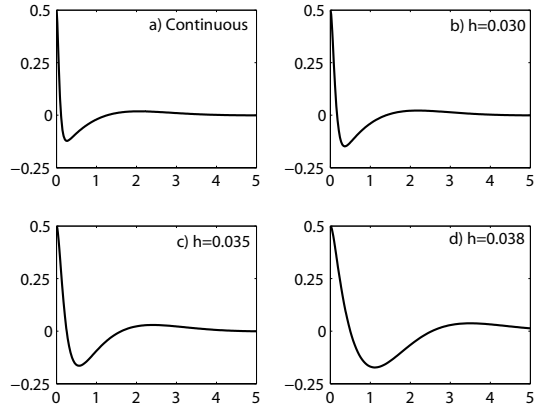


Figure 1. Dynamics with continuous-time controller approximation

¹The closed-loop poles represent the system's autonomous behavior. A system subject to a perturbation follows a specific trajectory. Its characteristics depend on the closed-loop poles.

2.2.3 System dynamics

Figure 3 (a) shows the integral of the absolute error (IAE) of the angle response for a wider range of sampling periods (see [2] for a detailed explanation of the IAE index). As it can be deduced, from $h \approx 0.040$, the closed-loop system is unstable ($\text{IAE} \rightarrow \infty$).

2.3 Implications

Discretization of continuous-time design works well if the sampling period is sufficiently small. In any case, the discrete-time desing will be only an approximation of the continuous one.

3 Discrete-time design

Discrete-time design methods usually consider the system through the values of the system inputs and outputs at the sampling instants. To do this, a sampled version of the continuous system model is derived. Then, a wide range of direct discrete-time controller design methods can be applied for obtaining the controller. In this case, we will focus on controller design methods using state-space models, although the analysis would hold for methods using transfer functions.

3.1 Design specifications

From the specification of the desired dominant modes for the closed-loop system, we obtain the location of the continuous-time closed-loop poles. From them, and using the exact mapping between s and z (10), we obtain the location of the discrete-time closed-loop poles.

$$z_i = e^{s_i h} \quad (10)$$

3.2 Pole placement approach

The design problem is formulated in terms of obtaining a closed-loop system with specified poles. For a sampled version (11) of the system (3), where Φ and Γ have been obtained using(12), the solution of the pole placement problem for the single-input-single-output case is given by a linear state feedback with the gain (13).

$$\begin{aligned} x(kh + h) &= \Phi x(kh) + \Gamma u(kh) \\ y(kh) &= Cx(kh) + Du(kh) \end{aligned} \quad (11)$$

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} ds B \end{aligned} \quad (12)$$

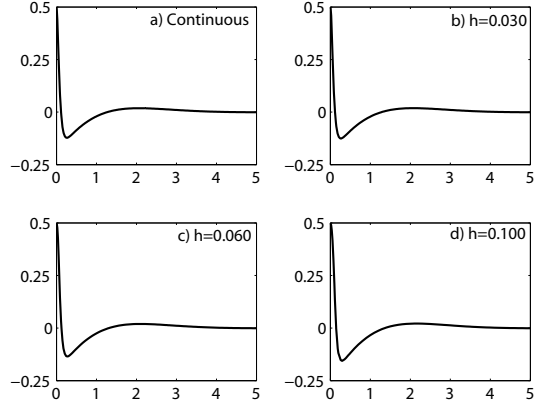


Figure 2. Dynamics with discrete-time controller

$$L = [0 \ 0 \ \dots \ 1] W_c^{-1} P(\Phi) \quad (13)$$

In (13), W_c is the reachability matrix of system (11) and $P(\Phi)$ is the characteristic polynomial of the closed-loop system (which is given by the desired location of its closed loop poles) evaluated in Φ .

3.3 Example

Using the inverted pendulum (6) of the previous example, and following the approach described in Section 3.2, we have designed three discrete-time gains (14) for $h = 0.030, 0.060$, and 0.100 s, respectively, that place the closed-loop poles at s_1, s_2, s_3 and s_4 (taking into account (10)).

$$\begin{aligned} L &= [-21.329 \ -23.950 \ -151.390 \ -26.910] \\ L &= [-11.291 \ -12.978 \ -86.319 \ -16.253] \\ L &= [-5.491 \ -6.558 \ -47.910 \ -9.693] \end{aligned} \quad (14)$$

Figure 2 shows a) the original dynamics, and the dynamics of the angle when controllers (14) are applied (sub-figures b), c) and d), respectively). Compared to Figure 1, it can be observed that the response meets the closed-loop system characteristics regardless of the sampling period (even for larger sampling periods than before). Figure 3 (b) shows the IAE of the angle response for the same rang of sampling periods as Figure 3 (a). As it can be seen, the IAE is almost the same for any period.

3.4 Implications

Direct-discrete time design works better than discretization of continuous time design. However, as it can be seen in Figure 3 (b), there is still a smooth degradation as the sampling period increases. This seems to

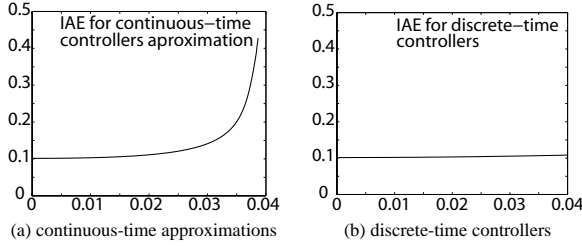


Figure 3. IAE

contradict the design procedure. Provided that the sampling period is appropriately selected, the exact mapping between s -poles and z -poles (10), followed by discrete pole placement should yield a closed-loop behavior at the sampling instants equal to that obtained by the continuous-time controller. However, this is not the case as we illustrate in Figure 4. As it can be seen, the response of the discrete-time controller for $h = 0.100\text{s}$ does not coincide with the response of the continuous-time controller. Therefore, further analysis is required.

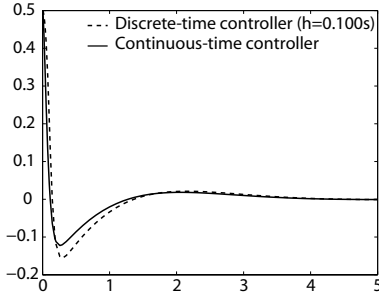


Figure 4. Dynamics with discrete-time controller

3.5 Non-dominant modes

The selection of the sampling period was made taking into account the rule-of-thumb based on (8). However, it was applied according to the dominant poles s_1 and s_2 , regardless of the location of the non-dominant poles s_3 and s_4 .

Some misleading practices in computer control focus the selection of the sampling period upon the dominant modes only. However, as we show next, the non-dominant poles can have a significant effect on the closed-loop system behavior.

To do so, we will place the continuous-time poles s_3 and s_4 at $-\infty$. Hence, the corresponding discrete-time poles are $z_3 = z_4 = 0$ (forcing a transient deadbeat dynamics). Figure 5 shows the angle dynamics for the same periods as before ($h = 0.030, 0.060$, and

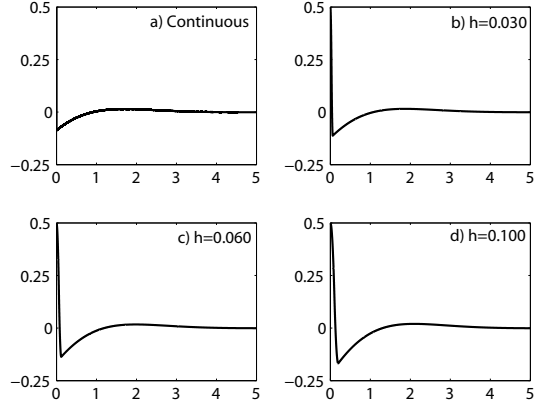


Figure 5. Dynamics with discrete-time controller when non-dominant modes are considered

0.100s) when the discrete-time controller is designed according to the new pole location. As it can be seen, increasing the sampling period drags the deadbeat dynamics, deteriorating the overall system response. That is, when the rise time due to the non-dominant poles is faster than the sampling period, the overshoot is delayed with respect to the expected one. Figure 6 further illustrates this phenomena when longer sampling periods ($h = 0.100 \dots 0.600\text{s}$) are applied. Note that for illus-

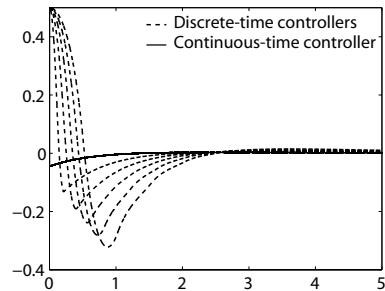


Figure 6. Detailed view

trative purposes we have shown an extreme case (non-dominant deadbeat poles). However, this effect also holds for “too” fast poles if the sampling period is not selected appropriately.

3.6 Correct selection of the sampling period

Notwithstanding the preceding, if the sampling period is selected a) considering the original continuous-time non-dominant poles ($s_3 = -20$ and $s_4 = -30$), and b) according to the rule-of-thumb that states that the sampling should be 4 to 10 per rise time (given

the fastest pole, s_4), we obtain a minimum period of $h = 0.0083$ s. Figure 8 shows the continuous response versus the response obtained when the controller has been well designed. As it can be seen, there is not appreciable difference.

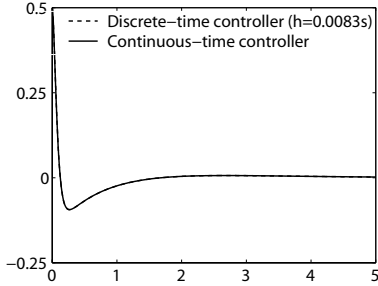


Figure 7. Correct design

4 Co-design

In the previous analysis we have seen that meeting all control specifications with a discrete-time controller design requires short periods, which may prevent task set schedulability. On the other hand, if the task period is specified in order to obtain a feasible task set, the control performance specifications may not be completely achieved. Therefore, a good design must trade-off control performance vs real-time performance.

4.1 Approaches

In the previous example, the sampling period was required to be $h = 0.0083$ s. Let us suppose that the real-time specification for the control task requires a task period of $h = 0.100$ s. Clearly, these specifications are contradictory. If the second prevails, two approaches can be considered, which are analyzed next.

1. not to change the control specifications (continuous time pole locations) and run a controller designed for the specified sampling period without accounting for the non-dominant poles (as it was done Section 3.3), or
2. changing the control specifications (moving the continuous time pole locations) in such a way that the specified sampling period is congruent with the new pole locations.

4.2 Example

Following the previous example (Section 3.3), let us suppose that we design the controller for $h = 0.100$ s,

thus choosing approach 1. In fact, the angle dynamics for this controller can be seen in Figure 4 dashed line. There is an detectable degradation (difference between the continuous and the discrete response).

Let us suppose that we consider approach 2. In this case, we change the fast poles location to $s_3 = -1.10$ and $s_4 = -1.25$, in such a way the $h = 0.100$ becomes then a well-selected sampling period according to the discussion in Section 3.6. Moving poles means having different closed-loop dynamics. Figure 8 shows the angle dynamics. The deviation of this response with respect to zero (IAE) is higher than the one of the previous response (Figure 4 dashed line), that is, a worse response is obtained. Therefore for this example, approach 1 is better than 2.

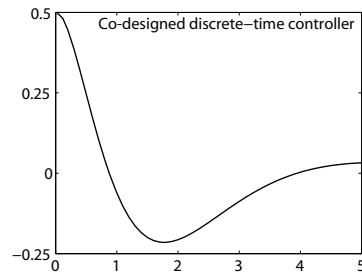


Figure 8. Co-designed controller

4.3 Discussion

Notice that approach 1 introduces a “non-predicted” dynamics error, whereas approach 2, introduces a higher but predicted error. In fact, it is interesting to point out that for a range of periods below the specified period, controllers designed by approaches 1 and 2 will give *different* class of responses. For approach 1, as the period decreases, better responses are obtained (in terms of IAE, for example). Not so for approach 2. This is illustrated next.

In Figure 9, and following the example of Section 4.2, we show a) the IAE for those controllers designed without accounting for the non-dominant poles (approach 1) and b) the IAE for those controllers accounting for non-dominant poles when co-design is applied (approach 2). As it can be seen, for approach 1, the curve descends as the sampling gets smaller (there is a decay from $h = 0.100$ to $h \approx 0$ of approx. 0.13 in the y-axis). However, for approach 2, the curve is almost horizontal (there is an insignificant decay from $h = 0.100$ to $h \approx 0$ of approx. 0.002 in the y-axis²). In Figure 9, the curve of approach 1 is always below (better performance) the

²This is mainly due to precision errors.

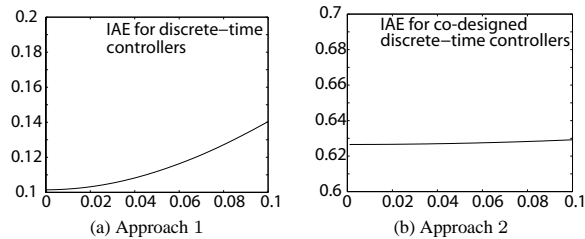


Figure 9. IAE

curve of approach 2, for the given range of h . Therefore, for this case, approach 1 could be used for those control and real-time system co-design methods that optimize control performance by shortening sampling rates (see for example [4] or [5]).

However, the relative position of these two curves change depending on many factors, such as the nature of plants, control specifications (e.g., pole locations), or real-time specifications (e.g., range of periods). Figure 10 shows the most relevant possibilities that will determine co-design. The right sub-figure shows the previ-

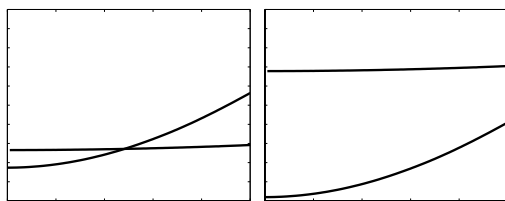


Figure 10. Co-designed approaches

ous case. It also may happen that the almost-horizontal curve (approach 2) lies below the other. The left sub-figure shows when both curves cross each other. This two cases have the following implications for co-design approaches³:

1. Slack-reclaiming based co-design approaches to improve control performance: these approaches (e.g.,[4] or [5]) should be based on approach 1 as far as its performance is always better (as in sub-figure right) than approach 2.
2. Co-design approaches for overload conditions management: these approaches, based on increasing periods in a controlled way (e.g.,[2]), must carefully examine how the increase of periods affects control performance. In this case, any of the

³Co-design approaches are generally based on performance cost functions (also called performance indexes) that depend on the error of the controlled systems (e.g., IAE), or on more sophisticated functions that depend also on the cost of the control signals (as it is usually done in optimal control problems)

two approaches, or a combination of them, may be feasible.

Finally, two important issues should be pointed out. First, since curves of approach 1 and 2 come from different continuous-time specifications, they should be normalized (e.g., dividing the IAE of the response of the discrete-time controller by the response of the continuous-time controller that they approximate) in order to have proper comparisons. Second, the initial conditions (i.e., perturbations) that affect each controlled plant affect the derivative of each curve. Hence, they have to be considered in the analysis. However, since they affect both curves proportionally, again, the curves can be properly compared.

5 Conclusions

We have reviewed well-known control system design methods to understand how they affect control performance. A first consequence has been to identify and classify which shapes of cost functions are usually used in the co-design of control and real-time systems. Further, we have provided guidelines on how to establish co-design approaches aimed at improving control performance and/or coping with overload conditions.

References

- [1] K. J. Åström and B. Wittenmark. *Computer-Controlled Systems. Third Edition*. Prentice-Hall, 1997.
- [2] G. Buttazzo, M. Velasco, P. Marti, and G. Fohler. Managing quality-of-control performance under overload conditions. In *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, July 2004.
- [3] Giorgio Buttazzo. *Hard Real-Time Computer Systems. Predictable Scheduling Algorithms and Applications*. Kluwer Academic Publishers, 1997.
- [4] A. Cervin, J. Eker, B. Bernhardsson, and K-E. Årzén. Feedback-feedforward scheduling of control tasks. *Real-Time Systems*, 23:25–53, July 2002.
- [5] P. Marti, C. Lin, S. Brandt, M. Velasco, and J. M. Fuertes. Optimal state feedback based resource allocation for resource-constrained control tasks. In *Proceedings of the 25th IEEE Real-Time Systems Symposium*, December 2004.