Toward Flexible Scheduling of Real-Time Control Tasks: Reviewing Basic Control Models

Pau Martí and Manel Velasco

Technical University of Catalonia, Automatic Control Department, Pau Gargallo 5, 08028 Barcelona, Spain
pau.marti@upc.edu
http://webesaii.upc.edu

Abstract. We review state-space control models in order to identify timing properties that can favour flexible scheduling of real-time control tasks. First, from the state-space model of a linear time-invariant discrete-time control system with time delay, we derive a new model that involves computing the control signal with a predicted state vector at the actuation instant. Second, by allowing irregular sampling instants, we show that the new model only forces a single synchronization point (actuation instant) while having a feasible implementation. This augurs better schedulability for a set of real-time control tasks, and can provide robustness against scheduling induced jitters.

1 Introduction

The state space model for a linear time-invariant discrete-time control system with time delay [1] implies a synchronous timing at the sampling and actuation instants that can be perfectly met by a single periodic real-time control task. Assuming that sampling and actuation occurs at the beginning and at the end of each task execution [2], the timing of the control task execution corresponds to the timing of the model if task period and deadline are set equal to the sampling period and time delay of the model. However, in a multitasking real-time control system, the tight specification of these timing constraints for control tasks impairs schedulability in the general case [3].

Relaxing this specification by setting the deadline greater than the time delay introduces jitters in control tasks executions that violate the synchronous timing of the model, causing control performance degradation [2]. To maintain synchronism, abstract computational models that force two synchronization points (sampling and actuation) for each control task have been developed ([4] or [5]).

Such computational models introduce three main drawbacks. First, they impose an artificially longer time delay in the closed loop system. Second, they constrain system schedulability by requiring two synchronization points. Third, if state feedback control is used, they involve computing the control signal considering a state vector that becomes outdated at the actuation instant.

The contribution of this paper is to derive a novel state-space model for real-time control tasks aimed at solving the above identified deficiencies. First, we

A. Bemporad, A. Bicchi, and G. Buttazzo (Eds.): HSCC 2007, LNCS 4416, pp. 710–713, 2007.

derive a new model, *prediction-based model*, that involves computing the control signal using the updated (predicted) state vector at the actuation instant, eliminating the possibly unknown and varying time delay between sampling and actuation. Second, by allowing in the new model irregular sampling instants, we show that the new model only forces a single synchronization point (actuation instant) while having a feasible implementation. This offers flexibility for real-time control tasks scheduling, which can improve schedulability and can provide robustness against jitters.

2 Standard Model

Consider the state space model of a linear time-invariant continuous-time system with time delay τ

$$\begin{aligned} \tfrac{dx(t)}{dt} &= Ax(t) + Bu(t-\tau) \\ y(t) &= Cx(t). \end{aligned} \tag{1}$$

where $x(t)$ is the plant state, $u(t)$ and $y(t)$ are the input and output of the plant, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$ are the system, input and output matrices respectively. The time delay can model an input/output latency that appears due to the computation of the control algorithm or due to the insertion of a network within a control loop. For periodic sampling, with sampling period h, where $\tau \le h$, we obtain the standard discrete time model

$$\begin{aligned} x_{k+1} &= \Phi(h)x_k + \Phi(h-\tau)\Gamma(\tau)u_{k-1} + \Gamma(h-\tau)u_k \\ y_k &= Cx_k, \end{aligned} \tag{2}$$

where matrices $\Phi(t)$, $\Gamma(t)$ are obtained using the following

$$\Phi(t) = e^{At}, \quad \Gamma(t) = \textstyle\int_0^t e^{As}Bds. \tag{3}$$

Note that eq. (2) slightly differs from conventional notation. The purpose of the new notation is to explicitly indicate dependencies on h and τ. An state space model of the system (2) is given by

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi(h) \; Phi(h-\tau)\Gamma(\tau) \\ 0 \qquad\qquad 0 \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} \Gamma(h-\tau) \\ I \end{bmatrix} u_k \tag{4}$$

where $z_k \in \mathbb{R}^{m\times 1}$ are the values that represent the past control signals. For closed loop operation, the control signal will be

$$u_k = \begin{bmatrix} L_1 \; L_2 \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} = L_1x_k + L_2z_k \quad \text{with} \quad L_1 \in \mathbb{R}^{1\times n}, \quad L_2 \in \mathbb{R}^{1\times m}. \tag{5}$$

Remark 1. The closed loop model given by (4) and (5) is based on two synchronization points, the sampling and actuation instants. At time t_k the k^{th} sample[1] (x_k) is taken, and at time $t_{k+\tau}$ the k^{th} control signal (u_k) is sent out. The sampling period h is defined from t_k to t_{k+1}, and the time delay τ from t_k to $t_{k+\tau}$.

[1] Sample is used to refer to the full state vector, regardless of whether it has been sampled or observed.

Remark 2. The closed loop model given by (4) and (5) involves computing the control signal to be sent out at time $t_{k+\tau}$ using a sample taken at t_k, τ time units before.

Remark 3. The closed loop model given by (4) and (5) holds the control signal u_k from $t_{k+\tau}$ to $t_{k+1+\tau}$.

3 Prediction-Based Model

A more consistent model could involve computing the control signal with the updated state vector at the actuation instant $t_{k+\tau}$. Instead of (5), we will have

$$u_k = Lx_{k+\tau}. \tag{6}$$

Remark 4. With (6), the sampling period h is the time elapsed from $t_{k+\tau}$ to $t_{k+1+\tau}$. Moreover, no delay is present. An u_k still is held from $t_{k+\tau}$ to $t_{k+1+\tau}$. Recall remarks 1, 2 and 3.

The new closed loop dynamics can be modeled by (6) and

$$x_{t+\tau+1} = \Phi(h)x_{k+\tau} + \Gamma(h)u_k. \tag{7}$$

Remark 5. Although u_k in (6) uses $x_{k+\tau}$, the sample is still taken at time t_k.

According to remark 5, $x_{k+\tau}$ has to be predicted from x_k, that is

$$x_{k+\tau} = \Phi(\tau)x_k + \Gamma(\tau)u_{k-1}. \tag{8}$$

Proposition 1. *All closed loop dynamics given by (6), (7) and (8) can be obtained by (4) and (5).*

Proof. Let L be the state feedback gain of (6). Substituting (8) into (6) we have

$$u_k = L\Phi(\tau)x_k + L\Gamma(\tau)u_{k-1}. \tag{9}$$

By comparing (9) to (5), we obtain that the state feedback gain of (5) is $L_1 = L\Phi(\tau)$ and $L_2 = L\Gamma(\tau)$ (when $z_k \in \mathbb{R}^{1\times 1}$). □

Proposition 2. *Not all closed loop dynamics given by (4) and (5) can be obtained by (6), (7) and (8).*

Proof. If $L_1 = L\Phi(\tau)$ and $L_2 = L\Gamma(\tau)$, by simple algebraic manipulations we obtain that $L_2 = L_1\Phi^{-1}(\tau)\Gamma(\tau)$. This is the family of controllers of model (4) and (5) that can be obtained by model (6), (7) and (8). □

Proposition 3. *All closed loop dynamics given by (4) and (5) can be obtained by (6), (7) and (8) if $m = n$.*

Proof. By adding $L_1 = L\Phi(\tau)$ and $L_2 = L\Gamma(\tau)$, and isolating L we obtain $L = (L_1 + L_2)(\Phi(\tau) + \Gamma(\tau))^{-1}$. This is only feasible if $m = n$. □

Remark 6. By proposition (1) and (2) we deduce that the model given by (4) and (5) is more general than the specified by (6), (7) and (8).

Remark 7. The closed loop system given by (6) and (7) is based on one synchronization point, $t_{k+\tau}$, that is, on the actuation instant.

Taking advantage of remark 7, we can relax the constraint imposed by remark 5. That is, the sample may be taken at times different than t_k.

Proposition 4. *For irregular sampling with $t_k \in (t_{k-(h+\tau)} \quad t_{k+\tau})$, model (4) and (5) can not be applied while model (6), (7) and (8) still holds.*

Proof. In terms of two consecutive irregular sampling instants, t_k and t_{k+1}, eq. (5) is given by

$$u_k = \begin{bmatrix} L_1(h_k) \; L_2(h_k) \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} \tag{10}$$

where $h_k = t_{k+1} - t_k$. Since at the actutaion instant ($t_{k+\tau}$), the next sampling instant (t_{k+1}) is not known, the feedback gain can not be correctly computed. However, in (6) L depends on h, which is constant since it goes from actuation to actuation instant (recall remark 4). The irregular sample only influences eq. (8), where $\tau = t_{k+\tau} - t_k$. Since t_k can be known at the actutaion instant ($t_{k+\tau}$), the new model accepts irregular sampling. □

4 Discussion and Conclusions

The discrete-time closed loop model given by (6), (7) and (8), a) eliminates the input/output delay, b) is based on a single synchronization point, c) uses an updated (predicted) state at the actuation instants, and more important, d) can absorb irregular sampling. From the real-time perspective, accepting irregular sampling has two main benefits. It can eliminate the degradation introduced by jitters, and it can improve schedulability (since task releases can be performed earlier). Future work will focus on a) analysing the impact of the new model on the schedulability of a set of real-time control tasks, and b) evaluating in a multitasking real-time control system the control performance achieved by a set of tasks implementing control laws based on the new model with respect to the standard one or to those using existing abstract computational models.

References

1. Åström, K.J., Wittenmark, B.: Computer controlled systems. Prentice Hall (1997)
2. Årzén, K.-E., Cervin, A., Eker, J., Sha, L.: An introduction to control and scheduling co-design. In Proc. of the 39th IEEE Conference on Decision and Control. (2000)
3. Martí, P., Fuertes, J.M., Villà, R., Fohler, G: On real-time control tasks schedulability. In European Control Conference. (2001)
4. Cervin, A., Eker, J.: The Control Server: a computational model for real-time control tasks. In Proc. of the 15th Euromicro Conference on Real-Time Systems. (2003)
5. Henzinger, T.A., Horowitz, B., and Kirsch, C.M.: Giotto: a time-triggered language for embedded programming. In Proc. of the First International Workshop on Embedded Software, LNCS **2211** (2001) 166–184