Real-Time Syst (2008) 40: 44–76
DOI 10.1007/s11241-008-9048-3

Quantifying and suppressing the measurement disturbance in feedback controlled real-time systems

Mehdi Amirijoo · Jörgen Hansson · Svante Gunnarsson · Sang H. Son

Published online: 14 February 2008

Abstract In the control of continuous and physical systems, the controlled system is sampled sufficiently fast to capture the dynamics of the system. In general, this property cannot be applied to the control of computer systems as the measured variables are often computed over a data set, e.g., deadline miss ratio. In this paper we quantify the disturbance present in the measured variable as a function of the data set size and the sampling period, and we propose a feedback control structure that suppresses the measurement disturbance. The experiments we have carried out show that a controller using the proposed control structure outperforms a traditional control structure with regard to performance reliability.

Keywords Adaptive scheduling · Feedback control · Measurement disturbance · System disturbance

This work was funded, in part by CUGS (the National Graduate School in Computer Science, Sweden), CENIIT (Center for Industrial Information Technology) under contract 01.07, ISIS (Information Systems for Industrial Control and Supervision), and NSF grants IIS-0208578 and CCR-0329609. This work was partly performed when J. Hansson was with Linköping University.

M. Amirijoo (✉)
Dept. of Computer and Information Science, Linköping University, Linköping, Sweden
e-mail: meham@ida.liu.se

J. Hansson
Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: hansson@sei.cmu.edu

S. Gunnarsson
Dept. of Electrical Engineering, Linköping University, Linköping, Sweden
e-mail: svante@isy.liu.se

S.H. Son
Dept. of Computer Science, University of Virginia, Charlottesville, VA, USA
e-mail: son@cs.virginia.edu

1 Introduction

In recent years a new class of soft real-time systems has emerged, e.g., web applications, e-commerce, and data-intensive applications. These applications typically operate in open and unpredictable environments, in which arrival patterns and resource requirements of tasks are in general unknown. For soft real-time systems operating in open environments, tasks cannot be subject to exact schedulability analysis given the lack of a priori knowledge of the workload, making transient overloads inevitable. Furthermore, these systems are becoming larger and more complex, and at the same time they are being used in applications where performance guarantees are needed. Feedback control scheduling has been introduced as a promising foundation for performance control of complex real-time systems (Hellerstein et al. 2004; Lu et al. 2001, 2002; Parekh et al. 2002; Amirijoo et al. 2006). It has been shown that feedback control is highly effective to support the specified performance of dynamic systems, which are both resource insufficient and exhibit unpredictable workloads.

When controlling physical and continuous systems, the sampling period selection is of paramount importance. The sampling period must be chosen such that the dynamics of the controlled system is captured and in general the sampling rate is set to the maximum that the controller and the AD/DA converters can manage (Franklin et al. 1998). However, when controlling computer systems, one cannot sample the controlled system arbitrarily fast. Usually, the measured variables are computed over a data set, e.g., utilization or deadline miss ratio. To form these metrics requires an underlying data set, which must be large enough to give an acceptable accuracy of the behavior of the controlled system. To obtain a large data set we have to set the sampling period to a large value, meaning that we gather data over a larger time window. Doing so, however, results in an unresponsive system as the controller is rarely invoked and, hence, does not react fast enough to failures or changes in workload. Therefore, ideally we want to choose a low sampling period, to react to changes in the controlled system, still being able to base the control actions on a valid and accurate representation of the controlled system. This enables controllers to be more efficient in keeping the actual performance at the reference performance. The latter increases the reliability of the system and implies a more controlled worst-case performance of the closed-loop system and faster convergence toward the desired performance.

Our contributions are as follows. We present a model and quantification method of the measurement disturbance of utilization, deadline miss ratio, and average quality of tasks. An optimal time variant estimator is introduced to suppress the measurement disturbance. The measurement disturbance quantifier and the optimal estimator are combined. This gives a feedback control structure that is less sensitive to the uncertainty of the measured variable compared to a feedback structure without an estimator. We show through experiments that this approach results in the controlled variable, which defines the system performance, to be significantly closer to the desired level compared to a traditional feedback structure where the disturbance is not suppressed.

The remainder of this paper is organized as follows. The assumed system model and the problem formulation are given in Sects. 2 and 3, respectively. This is followed by Sect. 4, which describes a state-space model of the controlled system. In Sect. 5 a feedback control structure, which suppresses the measurement disturbance is given. The performance of the proposed feedback structure is evaluated in Sect. 6. In Sect. 7 we give an overview on related work, followed by Sect. 8, where conclusions and future work are discussed. A table of commonly used variables is given on page 74.

2 Assumed system model

2.1 Task model

We consider a single processor real-time system as the controlled system. A task model similar to that used by Lu et al. (2002) is adopted in this paper. A task τ_i is classified as either a periodic or an aperiodic task. A task τ_i has $N_i \geq 2$ service levels given by $S_i = \{s_{i1}, \ldots, s_{ij}, \ldots, s_{iN_i}\}$. A service level s_{ij} gives the quality $q_i(s_{ij})$ of the result of τ_i, where $0 \leq q_i(s_{ij}) \leq 1$. The result quality $q_i(s_{ij})$ increases monotonically as j increases, and in particular we have that $0 \leq q_i(s_{i1}) < q_i(s_{i2}) < \cdots < q_i(s_{iN_i}) \leq 1$. We say that a task is terminated when it has completed or missed its deadline. Let $s_i^t \in S_i$ denote the service level of τ_i at the termination of τ_i. The probability of τ_i having service level s_i^t at the point of termination is denoted with $P_i(s_i^t)$. We assume that the result quality of the tasks solely depends on the service level and not on the result quality of other tasks.

Let us elaborate on what introducing service levels implies. For example, we may have two service levels, i.e., $N_i = 2$, where s_{i1} models a rejection at admission control and s_{i2} models an admission. The output quality is zero when a task is rejected whereas the output quality is one when the task is admitted, i.e., $q_{i1} = 0$ and $q_{i2} = 1$. Further, an admitted task may deliver results of varying quality or precision as given by the imprecise computation model (Liu et al. 1994). For example, following the milestone approach (Liu et al. 1994), j increases as the execution time given to a task increases, thus, enhancing the quality of the task result. As such, the service level of a task may be constant or vary during its execution. A task may start with a certain service level and terminate with another. In general, the imprecise computation model is employed by a wide range of applications, e.g., numerical algorithms (Fausett 2003), graph algorithms (Zilberstein and Russell 1996), real-time databases (Davidson and Watters 1988; Vrbsky and Liu 1993), web server systems (Abdelzaher et al. 2002), multimedia (Brandt et al. 1998), and control (Cervin et al. 2002). The number of service levels and the quality associated with each service level depend on the particular type of application.

Define $\hat{z}$ to be the estimate of the true variable z. A task has the following characteristics, which depend on the service level: period $p_i(s_{ij})$ (periodic tasks), mean inter-arrival time $r_i(s_{ij})$ (aperiodic tasks), relative deadline $d_i(s_{ij})$, execution time $x_i(s_{ij})$, and load $l_i(s_{ij})$. See Table 1 for a complete task model. The deadline d_i

Table 1 The assumed task model

Attribute	Periodic tasks	Aperiodic tasks
$d_i(s_{ij})$	$d_i(s_{ij}) = p_i(s_{ij})$	$d_i(s_{ij}) = r_i(s_{ij})$
$\hat{l}_i(s_{ij})$	$\hat{l}_i(s_{ij}) = \hat{x}_i(s_{ij})/p_i(s_{ij})$	$\hat{l}_i(s_{ij}) = \hat{x}_i(s_{ij})/\hat{r}_i(s_{ij})$
$l_i(s_{ij})$	$l_i(s_{ij}) = x_i(s_{ij})/p_i(s_{ij})$	$l_i(s_{ij}) = x_i(s_{ij})/r_i(s_{ij})$

of a task is set to the inter-arrival time of the task. The load l_i of each task is defined as the ratio of execution time to the inter-arrival time. Upon arrival a task presents its estimated load $\hat{l}_i(s_{ij})$ and its relative deadline d_i to the system. The actual load of the task $l_i(s_{ij})$ is not known due to, e.g., variations in execution time.

The task model presented above is general in that it embraces periodic as well as aperiodic tasks. Further, the model captures inaccuracies in task parameters, e.g., we differentiate between the estimated and the true execution time. Note that the task model does not pose any particular assumptions regarding the distribution of inter-arrival times and execution times, since these are heavily dependent on the particular types of application. We assume for simplicity that tasks are independent in that a task does not miss its deadline due to data dependencies with other tasks. This assumption is applicable to, e.g., real-time databases (Amirijoo et al. 2006), digital controllers (Franklin et al. 1998), and video coding and signal processing (Wiegand et al. 2003). For example, a video encoder is for simplicity implemented as a single task, which does not have any data dependencies with other tasks.

2.2 Performance metrics

We adopt the following notation of describing discrete variables in the time domain. A sampled variable $a(k)$ refers to the value of the variable a at time kT, where $T > 0s$ is the sampling period and k is the sampling instant. We let the interval $](k-1)T, kT]$ to denote all $x \in R$ that satisfy $(k-1)T < x \leq kT$, i.e., $\{x \in R | (k-1)T < x \leq kT\}$. For the rest of this paper, we sometimes drop k where the notion of time is not of primary interest. We start by defining the measured variables used throughout this paper.

Definition 1 (Measured Utilization) Let $n_B(k)$ be the number of time units that the system is busy computing and n_T is the total number of monitored time units in the time interval $](k-1)T, kT]$. The measured utilization,

$$u_m(k) = \frac{n_B(k)}{n_T} \tag{1}$$

is the ratio of time that the system is busy computing during the time interval $](k-1)T, kT]$.

The ratio of tasks missing their deadline is given by the measured deadline miss ratio and is formally defined as follows.

Definition 2 (Measured Deadline Miss Ratio) Let $n_\Theta(k)$ be the number of tasks terminated during the time interval $](k-1)T, kT]$. The number of tasks that have missed their deadline during the time interval $](k-1)T, kT]$ is denoted with $n_M(k)$. The measured deadline miss ratio,

$$m_m(k) = \begin{cases} \frac{n_M(k)}{n_\Theta(k)}, & n_\Theta(k) > 0 \\ 0, & n_\Theta(k) = 0, \end{cases} \tag{2}$$

is the ratio of tasks that have missed their deadline during the time interval $](k-1)T, kT]$.

Further, we have assumed that each task has a set of service levels, where a level is associated with an execution time and a quality of the task result. The service levels are chosen such that schedulability is maintained. The measured average task quality gives the precision of the task results and is defined as follows.

Definition 3 (Measured Average Task Quality) Let $\Theta(k)$ and $n_\Theta(k)$ be the set of tasks and number of tasks terminated during the time interval $](k-1)T, kT]$, respectively. The measured average task quality,

$$q_m(k) = \begin{cases} \frac{1}{n_\Theta(k)} \sum_{\tau_i \in \Theta(k)} q_i(s_i^t), & n_\Theta(k) > 0 \\ 0, & n_\Theta(k) = 0, \end{cases} \tag{3}$$

is the average result quality of tasks terminated during the time interval $](k-1)T, kT]$.

Next we give the feedback control architecture used in this paper.

2.3 Feedback architecture

The feedback control structure assumed in this paper is shown in Fig. 1. Throughout the paper, we let $y(k)$ denote the controlled variable, and as the controlled variable we use the utilization $u(k)$, deadline miss ratio $m(k)$, or the average task quality $q(k)$. Hence, $y(k)$ denotes $m(k)$, $u(k)$, or $q(k)$. The measurement of a controlled

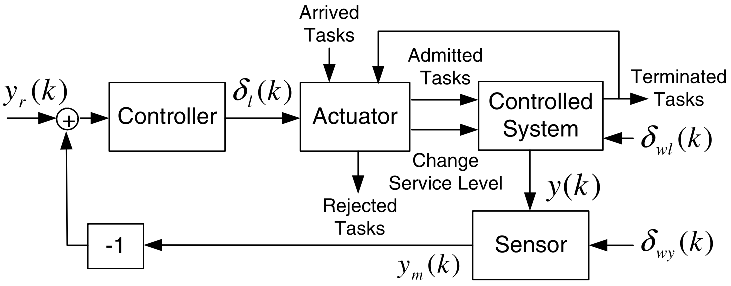


Fig. 1 The assumed feedback loop structure

variable is denoted with $y_m(k)$, i.e., $y_m(k)$ represents $u_m(k)$, $m_m(k)$, or $q_m(k)$. The controlled variable $y(k)$ is sampled using a sensor which measures the controlled variable according to (1)–(3). Input to the controller is the difference between the reference $y_r(k)$, representing the desired level of the controlled variable, and $y_m(k)$. Based on the performance error $y_r(k) - y_m(k)$ the controller computes a change $\delta_l(k)$ to the admitted workload.

The requested change in workload $\delta_l(k)$ is enforced by an actuator, as shown in Fig. 1. We assume in this paper that the actuator is either an admission controller, a task service level controller, or both. The main objective of the actuator is to enforce the workload change as given by $\delta_l(k)$. More specifically, the actuator forms the sum $\sum_{n=0}^{n=k} \delta_l(n)$ over all previous control inputs, which gives the desired level of admitted workload. The actual admitted workload is then adjusted by altering the service level of one or several tasks and/or adjusting the admission of tasks. As such, there is an inner loop, which continuously adjusts the admitted workload according to $\sum_{n=0}^{n=k} \delta_l(n)$. The actuator may be executed periodically or aperiodically, e.g., when a new task arrives, and it runs with a higher frequency than the controller in the outer loop from $y(k)$ to $\delta_l(k)$. Note that the actuator needs to monitor the terminated tasks, i.e., those leaving the systems, in order to maintain the admitted workload.

The control problem is how to compute the manipulated variable $\delta_l(k)$ such that the difference between $y_r(k)$ and $y(k)$ is minimized, i.e., for each k we want to minimize $(y_r(k) - y(k))^2$.

3 Problem formulation

We observe that variations in $y(k)$, see Fig. 1, are caused by the predictable changes in admitted workload $\delta_l(k)$, i.e., those computed by the controller, and unpredictable changes in workload. Recall from Sect. 2.1 that the actual workload may deviate from the true workload as a result of inaccurate execution time estimates and resource conflicts causing blocking, restart, and abortion of tasks. This causes unpredictable changes in workload, which gives rise to the system disturbance as defined by the following.

Definition 4 (System Disturbance) Let $A(k)$ be the set of admitted tasks at time kT. The system disturbance δ_{wl} represents unpredictable variations in admitted workload due to errors in the estimated execution time $\hat{x}_i$ of all tasks $\tau_i \in A(k)$.

Obviously, variations in $y(k)$ and, consequently, $y_m(k)$ are caused by $\delta_l(k)$ and $\delta_{wl}(k)$. However, there is also a third component contributing to variations in $u_m(k)$, $m_m(k)$, and $q_m(k)$, namely, the disturbance arising from the averaging operation in (1)–(3) called the measurement disturbance. Starting with $u_m(k)$ and $m_m(k)$, we note that for a given constant load the variance in $u_m(k)$ and $m_m(k)$ increases as n_T and $n_\Theta(k)$ decrease. This occurs as the sample sizes n_T and $n_\Theta(k)$ over which $u_m(k)$ and $m_m(k)$ are formed may not be sufficiently large to give an accurate estimate of the utilization and the deadline miss ratio, as illustrated in Fig. 2. There exists an analogous measurement disturbance for $q_m(k)$. When measuring $q_m(k)$, as defined

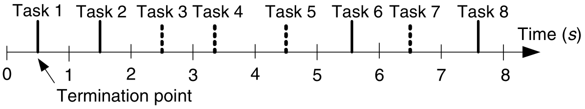


Fig. 2 A *dashed line* indicates a deadline miss, whereas a *solid line* represents a timely completion. If we choose $T = 1s$, then $m_m(1) = 0, m_m(2) = 0, m_m(3) = 1$ and so on. Increasing T to say $4s$, then $m_m(1) = 0.5$ and $m_m(2) = 0.5$. Thus as T and consequently $n_\Theta(k)$ increase, the variance in the measured variable $m_m(k)$ decreases

by (3), we observe that for a given constant load the variance in $q_m(k)$ increases as $n_\Theta(k)$ decreases. In the extreme case when $n_\Theta(k)$ is one and $q_i(s_{ij})$ is equal to zero or one for all tasks, then $q_m(k)$ is equal to zero or one. We are now ready to define the measurement disturbance.

Definition 5 (Measurement Disturbance) The measurement disturbance δ_{wy} represents unpredictable variations in measurements due to the uncertainty in the data set over which the metrics $u_m(k)$, $m_m(k)$, and $q_m(k)$ are computed.

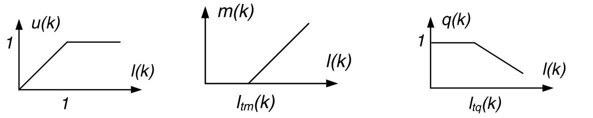
The efficiency of the control in terms of minimizing $(y_r(k) - y(k))^2$ strictly depends on the accuracy of $y_m(k)$. If $y_m(k)$ is accurate then we have a good representation of the controlled variable, thus, enabling improved control performance. Now, we know that the accuracy of $y_m(k)$ is enhanced by increasing the sample sizes over which the measurements are taken, i.e., increasing n_T and $n_\Theta(k)$. The only way of increasing $n_\Theta(k)$ is to increase T. However, choosing a longer sampling period degrades the responsiveness of the controller (Franklin et al. 1998), resulting in a slower reaction to changes in $y(k)$. Ideally, we want to have a short sampling period, to respond promptly to changes in the controlled variable, while experiencing a low measurement disturbance.

The problems we address in this work are the following: How can we model and quantify the system and measurement disturbances? Given a sampling period T, how can we efficiently suppress the measurement disturbance to obtain a more accurate representation of the behavior of the controlled system? What is the gain in performance control, with respect to minimizing $(y_r(k) - y(k))^2$, when suppressing the measurement disturbance? In summary, our findings provide an insight on how the computation of $y_m(k)$ gives rise to measurement disturbance and also how we can use this knowledge to reduce the impact of the measurement disturbance to achieve better performance control.

4 Modeling the controlled system

We adopt the linear and time-invariant model presented by Lu et al. (2002) due to its simplicity and sufficiently precise description of the dynamics of a real-time system. We extend the model to capture the system disturbance and the measurement disturbance. Recall from Sect. 2.3 that the actuator controls the admitted workload such that it equals $\sum_{n=0}^{n=k} \delta_l(n)$. The actuator runs at a higher rate than the controller

Fig. 3 Model of the controlled system



a. Utilization b. Deadline miss ratio c. Average task quality

and, hence, we assume that a desired change in workload $\delta_l(k)$ is executed before the next sampling instant $k+1$. As such, the workload $l(k+1)$ of admitted tasks in the next sampling period is changed due to the manipulated variable $\delta_l(k)$ and the system disturbance $\delta_{wl}(k)$, given by

$$l(k+1) = l(k) + \delta_l(k) + \delta_{wl}(k). \tag{4}$$

As mentioned previously in Sect. 3, the system disturbance $\delta_{wl}(k)$ arises from incomplete knowledge about the controlled system, e.g., unknown execution times and resource conflicts. We now model $u_m(k)$, $m_m(k)$, and $q_m(k)$.

4.1 Utilization

We say that an output signal is saturated when it remains unchanged even though the input signal is altered. The relationship between the admitted workload $l(k)$ and the utilization $u(k)$ is non-linear due to saturation, as follows,

$$u(k) = \begin{cases} l(k), & l(k) \leq 1 \\ 1, & l(k) > 1. \end{cases} \tag{5}$$

As shown in Fig. 3(a), when $l(k)$ is less or equal to 1, i.e. the CPU is underutilized, then $u(k)$ is not saturated and is equal to $l(k)$. However, when $u(k)$ is saturated, i.e., $l(k)$ is greater than 1, then $u(k)$ remains at 1, despite changes to $l(k)$. When the CPU is underutilized, we add a utilization measurement disturbance $\delta_{wu}(k)$ to $u(k)$, obtaining the measured utilization,

$$u_m(k) = u(k) + \delta_{wu}(k). \tag{6}$$

Recall that $\delta_{wu}(k)$ represents uncertainties in the utilization measurement due to incomplete data set over which the utilization is formed. The variations in $u_m(k)$ increase as the intensity of $\delta_{wu}(k)$ increases. We have above derived a model from the control input $\delta_l(k)$ to the measured utilization $u_m(k)$, capturing the system and the measurement disturbances.

4.2 Deadline miss ratio

Continuing with the deadline miss ratio $m(k)$, the relationship between the admitted workload $l(k)$ and $m(k)$ is non-linear due to saturation, as shown in Fig. 3(b). Let $l_{tm}(k)$ be the greatest workload threshold in the interval $](k-1)T, kT]$ for which admitted tasks are schedulable. We note that $m(k)$ is saturated when $l(k) \leq l_{tm}(k)$, and remains zero despite changes to $\delta_l(k)$, i.e., $m(k) = 0$ when $l(k) \leq l_{tm}(k)$. When not saturated, $m(k)$ increases non-linearly with $l(k)$. In this paper we use methods of linear control theory and, therefore, we linearize the relationship between $l(k)$ and $m(k)$ by forming the derivative,

$$g_m = \frac{dm(k)}{dl(k)}$$

of $m(k)$ at the vicinity of the reference $m_r(k)$. Hence, we get that,

$$m(k) = g_m l(k), \quad l(k) > l_{tm}(k). \tag{7}$$

To capture the deadline miss ratio measurement disturbance $\delta_{wm}(k)$, we model the measured deadline miss ratio as,

$$m_m(k) = m(k) + \delta_{wm}(k). \tag{8}$$

We have above derived a model from the control input $\delta_l(k)$ to the measured deadline miss ratio $m_m(k)$, capturing the system and the measurement disturbances.

4.3 Measured average task quality

As in the case of utilization and deadline miss ratio, the relationship between the average task quality and the load in the system is non-linear due to saturation, as shown in Fig. 3(c). As the load in the system increases, the quality of the results produced by tasks is lowered to guarantee task schedulability, resulting in decreasing $q(k)$. Let $l_{tq}(k)$ be the greatest workload threshold in the k^{th} period for which admitted tasks are schedulable at their greatest service level, i.e., $s_i^t = s_{iN_i}$ for all tasks τ_i. Hence for loads less than $l_{tq}(k)$, the average task quality $q(k)$ is saturated and equals one, as $q_i(s_i^t) = 1$ for all tasks τ_i. When $q(k)$ is not saturated (i.e., $l(k) > l_{tq}(k)$), the service levels have to be degraded to maintain schedulability, implying lower task result qualities and a decrease in $q(k)$. We linearize the relationship between $l(k)$ and $q(k)$, by forming the derivative,

$$g_q = \frac{dq(k)}{dl(k)}$$

of $q(k)$ at the vicinity of the reference $q_r(k)$. Hence, we get that,

$$q(k) = g_q l(k), \quad l(k) > l_{tq}(k). \tag{9}$$

To capture the measurement disturbance of the average task quality $\delta_{wq}(k)$, we model the measured average task quality as,

$$q_m(k) = q(k) + \delta_{wq}(k). \tag{10}$$

We have above derived a model from the control input $\delta_l(k)$ to the measured average task quality $q_m(k)$, capturing the system and the measurement disturbances.

4.4 A state-space model of the controlled system

Below we develop a state-space model of the controlled system, which is useful when analyzing and suppressing the measurement disturbance. We say that $l(k)$ is the state of the system. The following state space model is used to describe $y(k)$ and $y_m(k)$,

$$l(k+1) = l(k) + \delta_l(k) + \delta_{wl}(k) \tag{11a}$$

$$y(k) = g_y l(k) \tag{11b}$$

$$y_m(k) = g_y l(k) + \delta_{wy}(k). \tag{11c}$$

We now obtain the following results from (4)–(10):

- Under the condition $l(k) \leq 1$, we obtain the state-space model (11), where $y(k) = u(k)$, $y_m(k) = u_m(k)$, $g_y = 1$, and $\delta_{wy} = \delta_{wu}$.
- Under the condition $l_{tm} < l(k)$, we obtain the state-space model (11), where $y(k) = m(k)$, $y_m(k) = m_m(k)$, $g_y = g_m$, and $\delta_{wy} = \delta_{wm}$.
- Under the condition $l_{tq} < l(k)$, we obtain the state-space model (11), where $y(k) = q(k)$, $y_m(k) = q_m(k)$, $g_y = g_q$, and $\delta_{wy} = \delta_{wq}$.

Recall from Sect. 2.1, where we assumed that tasks are independent. This means that a task missing its deadline does not induce other tasks to miss their deadlines and, as such, there is no correlation between individual task deadline misses. Therefore, we assume that $\delta_{wm}(k)$ does not depend on its previous nor future values, i.e., there is no correlation between $\delta_{wm}(k)$ and $\delta_{wm}(k+b)$, where $k \neq b$. Also, in Sect. 2.1 we assumed that there are no relationships between the result qualities of the tasks, i.e., the result quality of a task does not depend on the result quality of other tasks. Hence, we may assume that there is no correlation between $\delta_{wq}(k)$ and $\delta_{wq}(k+b)$.

Now, we do not have a complete knowledge of the controlled system, e.g., we have inaccurate execution time estimates and, consequently, $\delta_{wl}(k)$ is unknown and cannot be determined. For simplicity we assume that $\delta_{wl}(k)$ is uncorrelated with $\delta_{wl}(k+b)$, meaning that a change in $\delta_{wl}(k)$ does not affect $\delta_{wl}(k+b)$. Since the measurement and the system disturbances are uncorrelated in time, we model the system and the measurement disturbances as white noise (Oppenheim and Willsky 1996) with expected values of zero,

$$E\{\delta_{wl}(k)\} = E\{\delta_{wu}(k)\} = E\{\delta_{wm}(k)\} = E\{\delta_{wq}(k)\} = 0. \tag{12}$$

We now define a measure of the magnitude of the system and the measurement disturbances.

Definition 6 (Disturbance Variance) Let the variance of the system and the measurement disturbances be,

$$\begin{gathered} R_{\delta wl}(k) = E\{\delta_{wl}^2(k)\}, R_{\delta wu}(k) = E\{\delta_{wu}^2(k)\}, \\ R_{\delta wm}(k) = E\{\delta_{wm}^2(k)\}, R_{\delta wq}(k) = E\{\delta_{wq}^2(k)\}. \end{gathered} \tag{13}$$

The system and measurement disturbances increase in magnitude as $R_{\delta wu}(k)$, $R_{\delta wm}(k)$, and $R_{\delta wq}(k)$ increase. In the following sections we use this model to design control structures that suppress the measurement disturbance.

5 Suppressing the measurement disturbance

We use a technique that is widely used in control theory where the problem of noisy measurements is often present (see, e.g., Glad and Ljung 2000 and Franklin et al. 1998). Let $\hat{z}(k|b)$ be the estimated value of $z(k)$, predicted at time bT. For example, $\hat{z}(k|k-1)$ refers to the estimated value of $z(k)$, predicted at time $(k-1)T$. We now define a closed loop estimator, which is called an observer in this paper.

Definition 7 (Observer) The observer is defined by,

$$\hat{l}(k+1|k) = \hat{l}(k|k) + \delta_l(k) \tag{14a}$$

$$\hat{l}(k|k) = \hat{l}(k|k-1) + K_y(k)(y_m(k) - g_y\hat{l}(k|k-1)) \tag{14b}$$

$$\hat{y}(k) = \hat{y}(k|k) = g_y\hat{l}(k|k) \tag{14c}$$

where $\hat{y}(k)$ is the estimate of $y(k)$ and $K_y(k)$ is the estimator feedback gain.

Assume that the current time is kT. The next estimated load $\hat{l}(k+1|k)$ is the sum of the current estimated load $\hat{l}(k|k)$ and $\delta_l(k)$. The current estimated load $\hat{l}(k|k)$ is the previously predicted current estimated load $\hat{l}(k|k-1)$, which is adjusted with respect to the measured variable $y_m(k)$. Remember that $y_m(k)$ is related to $l(k)$ and, hence, $y_m(k)$ is an indirect measure of the current load. By setting $K_y(k)$ to a large value, the estimate follows the true system state to a larger extent, as the impact of a difference in measurement and estimate, i.e., $y_m(k) - \hat{y}(k)$, is large. However, a large $K_y(k)$ implies that the measurement disturbance has a large influence on the state estimate. Hence, if the measurement disturbance is small and the system disturbance is large, then we should choose a large $K_y(k)$. In contrast, if the measurement disturbance is large and the system disturbance is small, then we set $K_y(k)$ to a small value to suppress the measurement disturbance. Applying this principle to the control of deadline miss ratio, we show in Sect. 5.1 that a large $n_\Theta(k)$ implies a small measurement disturbance and, hence, $K_m(k)$ should be set to a large value. However, if $n_\Theta(k)$ is small, meaning that the measurement disturbance is large, then $K_m(k)$ should be set to a small value to eliminate the disturbance due to the averaging operation. By utilizing the assumptions of the system disturbance and the measurement disturbance in Sect. 4.4, we obtain the following result.

Theorem 1 *The $K_y(k)$ that minimizes the variance of the estimation error $l(k) - \hat{l}(k)$ is,*

$$K_y(k) = \frac{g_y H_y(k)}{R_{\delta wy}(k) + g_y^2 H_y(k)} \tag{15}$$

where

$$H_y(k) = \frac{R_{\delta wy}(k-1)H_y(k-1)}{R_{\delta wy}(k-1) + g_y^2 H_y(k-1)} + R_{\delta wl}(k-1). \tag{16}$$

Proof Given the model (11), where $\delta_{wl}(k)$ and $\delta_{wy}(k)$ are white noise, and the observer (14), the optimal choice of $K_y(k)$ follows directly from the Kalman filter, see e.g. (Franklin et al. 1998). □

In this regard, the observer (14) where $K_y(k)$ is set according to Theorem 1 is an optimal estimator, meaning that it produces estimates that are closest to the true system state among all estimators. Now, let us study how $K_y(k)$ is affected by certain $R_{\delta wl}(k)$ and $R_{\delta wy}(k)$ when applying (15) and (16). It can be shown that $K_y(k)$ reaches steady state when $R_{\delta wl}(k)$ and $R_{\delta wy}(k)$ are constant (Franklin et al. 1998). Since it is difficult to analyze the time varying $K_y(k)$, for analysis purposes we consider $R_{\delta wl}(k)$ and $R_{\delta wy}(k)$ to be constant and study the value of $K_y(k)$ during steady state. During steady state we have that $K_y(k) = K_y(k-1) = K_y$, $R_{\delta wl}(k-1) = R_{\delta wl}(k) = R_{\delta wl}$, and $R_{\delta wy}(k-1) = R_{\delta wy}(k) = R_{\delta wy}$. Since we assume that $R_{\delta wy}(k)$ and $R_{\delta wl}(k)$ are constant, it must hold that $H_y(k) = H_y(k-1) = H_y$. Hence, by replacing $H_y(k-1)$ and $H_y(k)$ by H_y in (16) and solving for H_y we obtain,

$$H_y = \frac{R_{\delta wl}}{2} + \sqrt{\frac{R_{\delta wl}^2}{4} + \frac{R_{\delta wl} R_{\delta wy}}{g_y}}. \tag{17}$$

Hence, (17) gives the value that H_y converges to if the variance of the system and measurement disturbances are kept constant. Figure 4 plots K_y as a function of $R_{\delta wl}$ and $R_{\delta wy}$ according to (15) and (17), where $g_y = 1$. There are some interesting issues to consider here. We notice that K_y decreases as the measurement disturbance $R_{\delta wy}$ increases relatively to $R_{\delta wl}$, i.e., increasing $\frac{R_{\delta wy}}{R_{\delta wl}}$, meaning that the measured values have less impact on the estimate. We recall that the system disturbance represents variations in load, and as the system disturbance increases the estimation should rely more on the measurements to achieve better tracking of the true state of the system. Hence, K_y should increase as $R_{\delta wl}$ increases relatively to $R_{\delta wy}$, as is shown in Fig. 4.

Above we have derived a state estimator that given the magnitude of the measurement and system disturbances produces optimal estimates of the controlled system state. Next we quantify the measurement disturbance variance and propose a control structure that suppresses the measurement disturbance.

5.1 Quantifying the measurement disturbance variance

In this section we provide the theory for computing $R_{\delta wu}$, $R_{\delta wm}$, and $R_{\delta wq}$.

5.1.1 Deadline miss ratio

To distinguish between the measurement and the system disturbances we consider the variations in $m_m(k)$ to originate from $\delta_{wm}(k)$ only, i.e., we consider $l(k)$ to be constant.

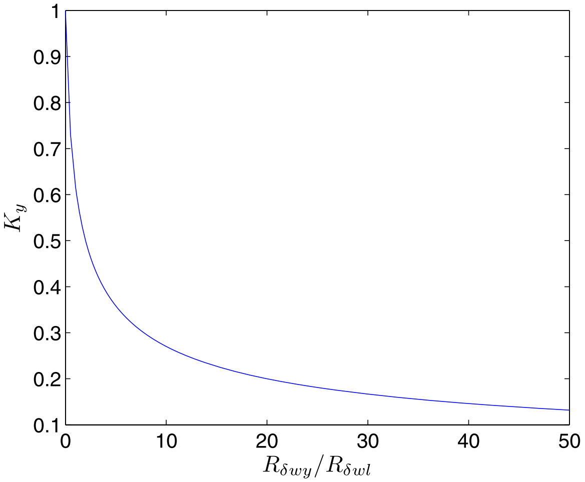


Fig. 4 The steady state value of $K_y(k)$ given $R_{\delta wl}(k)$ and $R_{\delta wy}(k)$

Theorem 2 *For a $n_\Theta(k) > 0$, the variance $R_{\delta wm}(k)$ of the deadline miss ratio measurement disturbance $\delta_{wm}(k)$ is,*

$$R_{\delta wm}(k) = \frac{\bar{m}(k) - \bar{m}^2(k)}{n_\Theta(k)}. \tag{18}$$

Proof From (11) and (12) we obtain that $\bar{m}_m(k) = \bar{m}(k) = g_m\bar{l}(k)$. Since $l(k)$ is constant, we observe that $l(k) = \bar{l}(k)$ and, hence, $\bar{m}_m(k) = g_m\bar{l}(k) = g_m l(k)$. From (11) and (14) we have that,

$$R_{\delta wm}(k) = E\{(m_m(k) - g_m l(k))^2\} = E\{(m_m(k) - \bar{m}_m(k))^2\}. \tag{19}$$

Now, consider a set of terminated tasks $\{\tau_1, \ldots, \tau_i, \ldots, \tau_{n_\Theta(k)}\}$ with the random variables $\{v_1, \ldots, v_i, \ldots, v_{n_\Theta}\}$, where $v_i = 1$ means that τ_i has missed its deadline and $v_i = 0$ means that τ_i met its deadline. Since we are assuming that tasks are independent, then the random variables $\{v_1, \ldots, v_{n_\Theta}\}$ are independent as well. Hence, the outcome of v_i does not affect the outcome of v_j, where $j \neq i$. The probability of $v_i = 1$ is $\bar{m}(k)$, whereas the probability of $v_i = 0$ is $1 - \bar{m}(k)$, i.e., the probability distribution function (PDF) is,

$$P_m(v_i) = \begin{cases} \bar{m}(k), & v_i = 1 \\ 1 - \bar{m}(k), & v_i = 0, \end{cases}$$

with,

$$E\{v_i\} = E\{v_i^2\} = \bar{m}(k). \tag{20}$$

The measured deadline miss ratio

$$m_m(k) = \frac{v_1 + \cdots + v_{n_\Theta}}{n_\Theta(k)} = \frac{1}{n_\Theta(k)} v_1 + \cdots + \frac{1}{n_\Theta(k)} v_{n_\Theta} \tag{21}$$

is then the average of the random variables. Using (19) and (21) we get,

$$R_{\delta wm}(k) = \frac{E\{(v_1 - \bar{v}_1)^2\} + \cdots + E\{(v_{n_\Theta} - \bar{v}_{n_\Theta})^2\}}{n_\Theta^2(k)}$$

and from (20) it follows that,

$$R_{\delta wm}(k) = \frac{\overbrace{\bar{m}(k) - \bar{m}^2(k) + \cdots + \bar{m}(k) - \bar{m}^2(k)}^{n_\Theta(k)}}{n_\Theta^2(k)} = \frac{\bar{m}(k) - \bar{m}^2(k)}{n_\Theta(k)}. \qquad \square$$

By analyzing the result of Theorem 2 we observe the following. The variance $R_{\delta wm}(k)$ of the deadline miss ratio measurement disturbance decreases monotonically to zero as $n_\Theta(k)$ increases. Now, we can safely assume that the number of terminated tasks $n_\Theta(k)$ increases as T increases. This is intuitive as more tasks are terminated when the sampling period increases. Considering this, we get the following result.

Corollary 1 *If* $\bar{m}(k) = 0$ *or* $\bar{m}(k) = 1$, *then* $R_{\delta wm}(k) = 0$ *for all* T. *Under the assumption* $0 < \bar{m} < 1$,

$$\lim_{T\to\infty} R_{\delta wm}(k) = 0$$

if and only if the number of terminated tasks $n_\Theta(k)$ *increases monotonically with the sampling period* T.

Proof The case of $\bar{m}(k) = 0$ or $\bar{m}(k) = 1$ follows directly from Theorem 2. For $0 < \bar{m} < 1$, we break the proof into two parts.

($\Rightarrow$) If $n_\Theta(k)$ increases monotonically with T, then $n_\Theta(k) \to \infty$ as $T \to \infty$ and, hence, Corollary 1 follows from Theorem 2.

($\Leftarrow$) Proof by contradiction. Assume that $\lim_{T\to\infty} R_{\delta wm}(k) = 0$ and the number of terminated tasks $n_\Theta(k)$ does not increase monotonically with the sampling period T. However, since $n_\Theta(k)$ does not increase monotonically with the sampling period T and $0 < \bar{m} < 1$, the disturbance variance $R_{\delta wm}(k)$ may not converge to zero, as $T \to \infty$. This is realized if for example $n_\Theta(k)$ is constant although T increases. This contradicts our assumption, hence, if $\lim_{T\to\infty} R_{\delta wm}(k) = 0$, then number of terminated tasks $n_\Theta(k)$ must increase monotonically with the sampling period T. □

This result suggests that we need an infinite sampling period to achieve zero deadline miss ratio measurement disturbance. An infinite sampling period is not feasible in practice and as such, the measurement disturbance will always be present in the measurement.

Now, by forming the partial derivative of (18) with respect to $\bar{m}(k)$ and solving,

$$\frac{\partial R_{\delta wm}}{\partial \bar{m}(k)} = \frac{1 - 2\bar{m}(k)}{n_\Theta(k)} = 0$$

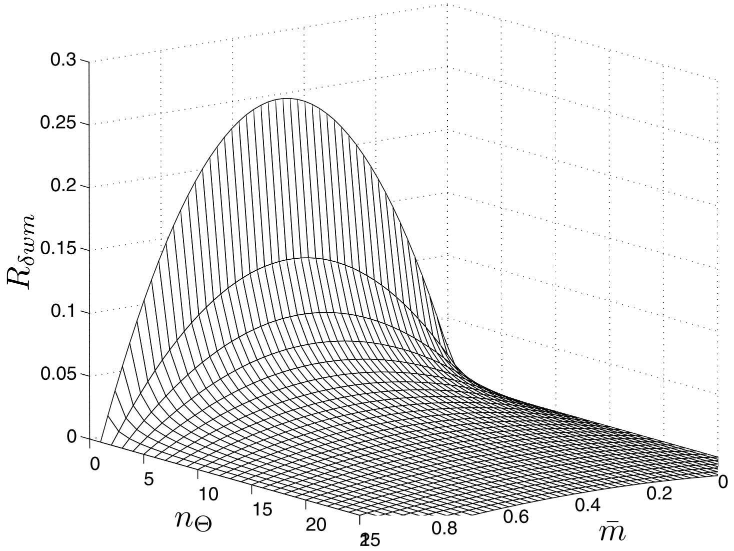


Fig. 5 $R_{\delta wm}(k)$ as a function of $\bar{m}(k)$ and $n_\Theta(k)$

we find that $R_{\delta wm}(k)$ is maximized at $\bar{m}(k) = \frac{1}{2}$ for any $n_\Theta(k)$. This is shown in Fig. 5 where $R_{\delta wm}(k)$ is plotted as a function of $\bar{m}(k)$ and $n_\Theta(k)$. As expected, $R_{\delta wm}(k)$ decreases as $n_\Theta(k)$ increases, meaning that the measurement disturbance originating from the averaging operation decreases in variance. For a certain $n_\Theta(k)$, the variance of the measurement disturbance peaks when the average deadline miss ratio $\bar{m}(k)$ is $\frac{1}{2}$. The variance $R_{\delta wm}(k)$ is zero when $\bar{m}(k)$ is zero or one, which is expected as we have no variation in $m(k)$ at zero or one deadline miss ratio.

5.1.2 Utilization

The derivation of $R_{\delta wu}(k)$ is analogous to the derivation of $R_{\delta wm}(k)$. To distinguish between the measurement and the system disturbances we consider the variations in $u_m(k)$ to originate from $\delta_{wu}(k)$ only, i.e., we consider $l(k)$ to be constant.

Theorem 3 *For a $n_T > 0$, the variance $R_{\delta wu}(k)$ of the utilization measurement disturbance $\delta_{wu}(k)$ is,*

$$R_{\delta wu}(k) = \frac{\bar{u}(k) - \bar{u}^2(k)}{n_T}. \tag{22}$$

Proof From (11) and (12) we obtain that $\bar{u}_m(k) = \bar{u}(k) = \bar{l}(k)$. Since $l(k)$ is constant, we observe that $l(k) = \bar{l}(k)$ and, hence, $\bar{u}_m(k) = \bar{l}(k) = l(k)$. From (11) and (13) we have that,

$$R_{\delta wu}(k) = E\{(u_m(k) - l(k))^2\} = E\{(u_m(k) - \bar{u}_m(k))^2\}. \tag{23}$$

Recall that $u(k)$ is measured over the interval $](k-1)T, kT]$, which we divide into n_T time units. We introduce the independent random variables $b_1, \dots, b_i, \dots, b_{n_T}$, where $b_i = 1$ means that the CPU is busy and $b_i = 0$ means that the CPU is free during

$$\left](k-1)T + (i-1)\frac{T}{n_T}, (k-1)T + i\frac{T}{n_T}\right].$$

The probability of $b_i = 1$ is $\bar{u}(k)$, whereas the probability of $b_i = 0$ is $1 - \bar{u}(k)$, i.e., the PDF is,

$$P_u(b_i) = \begin{cases} \bar{u}(k), & b_i = 1 \\ 1 - \bar{u}(k), & b_i = 0, \end{cases}$$

with,

$$E\{b_i\} = E\{b_i^2\} = \bar{u}(k). \tag{24}$$

The measured utilization

$$u_m(k) = \frac{b_1 + \cdots + b_{n_T}}{n_T} = \frac{1}{n_T} b_1 + \cdots + \frac{1}{n_T} b_{n_T} \tag{25}$$

is then the average of the random variables. Using (23), (24), and (25) we get,

$$R_{\delta wu}(k) = \frac{\bar{u}(k) - \bar{u}^2(k)}{n_T}.$$

□

By analyzing the result of Theorem 3 we observe the following. The variance $R_{\delta wu}(k)$ of the utilization measurement disturbance decreases monotonically to zero as n_T increases. This gives the following result.

Corollary 2 *If $\bar{u}(k) = 0$ or $\bar{u}(k) = 1$, then $R_{\delta wu}(k) = 0$ for all T. Under the assumption $0 < \bar{u} < 1$,*

$$\lim_{T \to \infty} R_{\delta wu}(k) = 0$$

if and only if the number of monitored time units n_T increases monotonically with the sampling period T.

Proof The proof follows the same line of argument as the proof for Corollary 1. □

As for the case of deadline miss ratio, this result suggests that we need an infinite sampling period to achieve zero utilization measurement disturbance. Hence, the measurement disturbance will always be present in the measurement. Similarly to deadline miss ratio, $R_{\delta wu}(k)$ is maximized at $\bar{u}(k) = \frac{1}{2}$ for any n_T. The measurement disturbance variance is zero at $\bar{u}(k) = 0$ and $\bar{u}(k) = 1$.

5.1.3 Average task quality

To distinguish between the measurement and the system disturbances we consider the variations in $q_m(k)$ to originate from $\delta_{wq}(k)$ only, i.e., we consider $l(k)$ to be constant.

Theorem 4 *For the set of terminated tasks* $\Theta(k) = \{\tau_1, \dots, \tau_{n_\Theta(k)}\}$ *with* $n_\Theta(k) > 0$, *the variance* $R_{\delta wq}(k)$ *of the average task quality measurement disturbance* $\delta_{wq}(k)$ *is*,

$$R_{\delta wq}(k) = \frac{R_{q_1} + \cdots + R_{q_{n_\Theta(k)}}}{n_\Theta^2(k)}, \tag{26}$$

where,

$$R_{q_i} = \sum_{j=1}^{N_i} q_i^2(s_{ij}) P_i(s_{ij}) - \left(\sum_{j=1}^{N_i} q_i(s_{ij}) P_i(s_{ij}) \right)^2 . \tag{27}$$

Proof From (11) and (12) we obtain that $\bar{q}_m(k) = \bar{q}(k) = g_q \bar{l}(k)$. Since $l(k)$ is constant, we observe that $l(k) = \bar{l}(k)$ and, hence, $\bar{q}_m(k) = g_q \bar{l}(k) = g_q l(k)$. From (11) and (14) we have that,

$$R_{\delta wq}(k) = E\{(q_m(k) - g_q l(k))^2\} = E\{(q_m(k) - \bar{q}_m(k))^2\}. \tag{28}$$

Now, consider the set of terminated tasks $\Theta(k) = \{\tau_1, \dots, \tau_i, \dots, \tau_{n_\Theta(k)}\}$. We recall from Definition 3 that,

$$q_m(k) = \frac{1}{n_\Theta(k)} q_1(s_1^t) + \cdots + \frac{1}{n_\Theta(k)} q_{n_\Theta(k)}(s_{n_\Theta(k)}^t) \tag{29}$$

where s_i^t is the service level of task τ_i upon the termination of τ_i. Hence, it follows from (28) and (29) that,

$$R_{\delta wq} = E\{(q_m(k) - \bar{q}_m(k))^2\} = \frac{R_{q_1} + \cdots + R_{q_{n_\Theta(k)}}}{n_\Theta^2(k)}.$$

The variance R_{q_i} of random variable q_i is by definition,

$$\begin{aligned} R_{q_i} &= E\{(q_i - \bar{q}_i)^2\} = E\{q_i^2\} - \bar{q}_i^2 \\ &= \sum_{j=1}^{N_i} q_i^2(s_{ij}) P_i(s_{ij}) - \left(\sum_{j=1}^{N_i} q_i(s_{ij}) P_i(s_{ij}) \right)^2 . \end{aligned}$$

□

The following example shows how Theorem 4 is applied.

Example 1 Assume that a task τ_i has N_i service levels. We let $q_i(s_{ij}) = \frac{j}{N_i}$ and $P_i(s_{ij}) = \frac{1}{N_i}$. Hence, the quality of task results increases linearly with the service

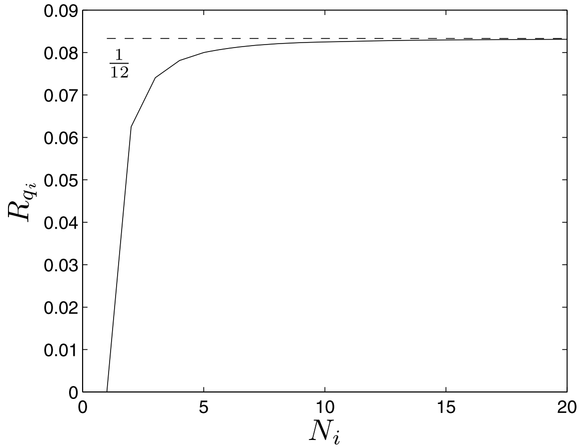


Fig. 6 R_{q_i} as a function of N_i

level and the probability of a task terminating with a service level is evenly distributed among the service levels. From (27) we get,

$$\begin{aligned} R_{q_i} &= \sum_{j=1}^{N_i}\left(\frac{j}{N_i}\right)^2 \frac{1}{N_i} - \left(\sum_{j=1}^{N_i} \frac{j}{N_i}\frac{1}{N_i}\right)^2 \\ &= \frac{(N_i+1)(2N_i+1)}{6N_i^2} - \frac{(N_i+1)^2}{4N_i^2} = \frac{N_i^2-1}{12N_i^2}, \end{aligned}$$

which inserted into (26) finally gives,

$$R_{\delta wq}(k) = \frac{\frac{N_1^2-1}{12N_1^2} + \cdots + \frac{N_{n_\Theta(k)}^2-1}{12N_{n_\Theta(k)}^2}}{n_\Theta^2(k)}.$$

Figure 6 shows how R_{q_i} varies as a function of N_i. The dashed line represents $\lim_{N_i\to\infty} \frac{N_i^2-1}{12N_i^2} = \frac{1}{12}$. If we assume that $N_i > 3$ for all tasks τ_i, then $R_{q_i} \approx 0.08$, and in that case,

$$R_{\delta wq}(k) \approx \frac{0.08 + \cdots + 0.08}{n_\Theta^2(k)} = \frac{0.08}{n_\Theta(k)}.$$

In Example 1 we assume that the probability of a task terminating with a service level is evenly distributed among the service levels. This assumption is restrictive and will not hold for all systems and workloads. For example, it is likely that a task terminates with one or a subset of the service levels when the system is underutilized. In this case, all tasks may terminate at their highest service levels giving the highest output quality. Hence, we have that $P_i(s_{ij}) = 1$ for $j = N_i$ and $P_i(s_{ij}) = 0$ for all other j. In reality we may have little information regarding the distribution P_i, which may even change during run-time (e.g., from an underutilized state to an overloaded state). Rather, we may monitor the service level of the tasks upon their termination and estimate P_i during run-time.

Let us now analyze whether $R_{\delta wq}(k)$ decreases as T increases. We say that the result quality of a task τ_i is subject to variation, if and only if $R_{q_i} > 0$, i.e., τ_i delivers results of varying quality. The following shows that $R_{\delta wq}(k)$ converges to zero as $n_\Theta(k)$ and T go toward infinity.

Corollary 3 *If the quality level of all tasks is constant, i.e., $R_{q_i} = 0$ for all $1 \le i \le n_\Theta(k)$, then $R_{\delta wq} = 0$ for all T. Under the assumption that the result quality of at least one task τ_i is subject to variation, i.e., $R_{q_i} > 0$, then*

$$\lim_{T\to\infty} R_{\delta wq}(k) = 0$$

if and only if the number of terminated tasks $n_\Theta(k)$ increases monotonically with the sampling period T.

Proof If $R_{q_i} = 0$ for all $1 \le i \le n_\Theta(k)$, then $R_{q_i} = 0$ follows directly from Theorem 4. For the case where the result quality of at least one terminated task τ_i is subject to variation we divide the proof in two parts.

($\Rightarrow$) We first prove that $\lim_{n_\Theta(k)\to\infty} R_{\delta wq}(k) = 0$. Since $0 \le q_i(s_{ij}) \le 1$ and $0 \le P_i(s_{ij}) \le 1$ it holds that,

$$\sum_{j=1}^{N_i} q_i^2(s_{ij}) P_i(s_{ij}) \le N_i.$$

Further,

$$R_{q_i} = \sum_{j=1}^{N_i} q_i^2(s_{ij}) P_i(s_{ij}) - \left(\sum_{j=1}^{N_i} q_i(s_{ij}) P_i(s_{ij})\right)^2 \le N_i \le N \tag{30}$$

where $N = \max_{\forall \tau_i \in \Theta(k)} N_i$. From (26) and (30) we find that,

$$R_{\delta wq}(k) = \frac{R_{q_1} + \cdots + R_{q_{n_\Theta(k)}}}{n_\Theta^2(k)} \le \frac{N}{n_\Theta(k)}.$$

Consequently, under the condition that $n_\Theta(k)$ increases monotonically with T, we obtain that for any finite number of service levels,

$$\lim_{T\to\infty} R_{\delta wq}(k) = \lim_{n_\Theta(k)\to\infty} R_{\delta wq}(k) = 0.$$

($\Leftarrow$) See proof of Corollary 1. □

As in the case of deadline miss ratio and utilization, the measurement disturbance variance for the average task quality equals zero if an infinite sampling period is selected. As the latter cannot hold, we cannot avoid the presence of the measurement disturbance in this case either.

5.2 Estimating the system disturbance variance

We recall from (16) that $R_{\delta wl}(k)$ must be known to compute the estimator gain $K_y(k)$ in (14). In general it is very difficult to obtain the exact value of the system disturbance variance and often one has to resort to estimations (Franklin et al. 1998). The system disturbance represents the uncertainties in the system (e.g., resource conflicts and varying execution time) and, therefore, it is impossible to accurately compute $R_{\delta wl}(k)$. If we can exactly determine $R_{\delta wl}(k)$, then this implies that the probability distribution of $\delta_{wl}(k)$ is known. To know the probability distribution of $\delta_{wl}(k)$ requires that probability distribution of the task execution times are available. This leads to a contradiction since we do not know the execution time distribution of the tasks as assumed in Sect. 2.1. Instead we present an estimator that produces estimates of $R_{\delta wl}(k)$.

Let us start with reviewing Definition 6. By using (11a) we find that,

$$R_{\delta wl}(k) = E\{\delta_{wl}^2(k)\} = E\{(l(k+1) - l(k) - \delta_l(k))^2\}.$$

As the load $l(k)$ in the system is unknown, due to inaccurate execution time estimates and resource conflicts, we have to resort to approximations of $l(k)$. We estimate $l(k)$ by the measured estimated workload of admitted tasks,

$$l_{ad}(k) = \sum_{\forall \tau_i \in A(k)} \hat{l}_i$$

where $A(k)$ is the set of admitted tasks at time kT. This means that we want to evaluate,

$$E\{(l_{ad}(k+1) - l_{ad}(k) - \delta_l(k))^2\}.$$

Therefore, we introduce the estimator,

$$\hat{R}_{\delta wl}(k) = \frac{1}{W} \sum_{p=1}^{W} \big(l_{ad}(k+1-p) - l_{ad}(k-p) - \delta_l(k-p)\big)^2, \tag{31}$$

that forms the average of $(l_{ad}(k+1) - l_{ad}(k) - \delta_l(k))^2$ over a subset of previous samples, where $W \in Z^+$ is the estimation window. The choice of W is a tradeoff between tracking and accuracy. If the system disturbance variance changes frequently, then choosing a small W is beneficial as the estimator can better track changes in the variance. However, if system disturbance variance does not change considerably, then a large W must be chosen to eliminate errors in the estimate, hence, achieve more accurate estimates. As an alternative to (31), we may use the moving average of $(l_{ad}(k+1) - l_{ad}(k) - \delta_l(k))^2$, i.e.,

$$\hat{R}_{wl}(k) = (1-\beta)\hat{R}_{wl}(k-1) + \beta(l_{ad}(k) - l_{ad}(k-1) - \delta_l(k-1))^2, \quad 0 < \beta\ < 1$$

which is computationally more efficient than (31). The parameter β has similar properties as W.

5.3 A measurement disturbance suppressive feedback control structure

The feedback control structure that suppresses the measurement disturbance consists of the classical feedback loop and the additional observer, as shown in Fig. 7. Note, we either use $u_m(k)$ and $n_T(k)$, $m_m(k)$ and $n_\Theta(k)$, or $q_m(k)$ and $n_\Theta(k)$ depending on the choice of the controlled variable. The observer (14) is used to estimate the controlled variable $y(k)$. According to the separation principle (Franklin et al. 1998; Glad and Ljung 2000), the design of the controller and the observer is disjoint, meaning that the tuning of one does not affect the other one. Hence, the separation principle significantly reduces the design complexity. The controller is designed using profiling data and a tuning method, e.g., (Lu et al. 2002), and the observer is designed using the profiling data and (14).

The following takes place during run-time. At time kT, the measured variable $y_m(k)$ is formed by the sensor and returned to the observer along with the current admitted load $l_{ad}(k)$ and either $n_\Theta(k)$ or $n_T(k)$ (depending on the controlled variable). The measurement disturbance variance $R_{\delta wy}(k)$ is then computed using (18), (22), or (26). The system disturbance variance $R_{\delta wl}(k-1)$ is estimated using (31), thus, $\hat{R}_{\delta wl}(k-1)$ is obtained. Having $R_{\delta wy}(k)$ and $\hat{R}_{\delta wl}(k-1)$, the estimator feedback gain $K_y(k)$ is computed according to (15) and (16), where $R_{\delta wl}(k-1)$ is replaced with $\hat{R}_{\delta wl}(k-1)$. Once, the feedback gain is updated, the observer (14) is used to compute $\hat{y}(k)$. The controller then computes $\delta_l(k)$ using the estimate $\hat{y}(k)$.

The effect of using an observer is the following. For simplicity assume that $R_{\delta wl}(k)$ is constant and let us study the case when we have varying $n_\Theta(k)$ and $n_T(k)$ (we consider the opposite below). As $n_\Theta(k)$ or $n_T(k)$ increases, then we should trust $y_m(k)$ more as the measured variable is based on a larger data set. An increase in $n_\Theta(k)$ or $n_T(k)$ corresponds to a decrease in $R_{\delta wy}(k)$, which in turn results in an increase in $K_y(k)$ (see Figs. 4 and 5). An increase in $K_y(k)$ corresponds to the estimation relying more on the measured variable rather than the prediction. Similarly, as $n_\Theta(k)$ or $n_T(k)$ decrease, estimates are based more on predictions and less on the measurements.

Now, consider the opposite where $R_{\delta wy}(k)$ is constant (i.e., $n_\Theta(k)$ or $n_T(k)$ are invariant) and the admitted load has reached stationarity in the sense that the load does not change considerably. In other words, the system disturbance $R_{\delta wl}(k)$ is very small. If suddenly an unforeseen event that rapidly changes the admitted load occurs, then an increase in $R_{\delta wl}(k)$ is detected by the estimator (31), hence, resulting in an increase in $\hat{R}_{\delta wl}(k)$. At this point we shall rely more on the measurements to better track the changes in admitted load. An increase in $\hat{R}_{\delta wl}(k)$ results in an increase in $K_y(k)$, meaning that the estimate of $y(k)$ must be based more the measurements rather than the predictions, which by definition do not account for unpredictable changes.

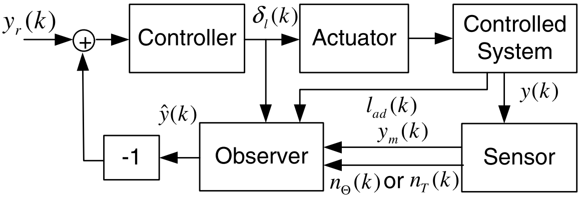


Fig. 7 The feedback structure for measurement disturbance suppression. Note, the task flow in Fig. 1 has been removed to enhance the presentation

Table 2 The main constituent, prediction or measurement, of the estimates

	Low $\hat{R}_{\delta wl}(k)$	High $\hat{R}_{\delta wl}(k)$
Low $R_{\delta wy}(k)$	Prediction and Measurement	Measurement
High $R_{\delta wy}(k)$	Prediction	Prediction and Measurement

The above is captured in Table 2, where four cases are depicted. Although, an estimate of the controlled variable is always based on both the measurement and the prediction, Table 2 gives the key constituent, either the prediction or the measurement, when forming the estimate. In two of the cases, namely, when both disturbances are either low or high, neither the prediction nor the measurement dominates when computing the estimate. Rather, the prediction and the measurement are weighted equally when computing the estimate. The overall result is an efficient way of suppressing disturbances present in $y_m(k)$. This means that we present a more accurate representation of the system state to the controller which is able to enhance the performance management in terms of lowering $(y_r(k) - y(k))^2$.

6 Performance evaluation

The goal of the performance evaluation is to determine the suitability of the proposed approach, namely, using an observer to suppress the measurement disturbance. In this regard we perform an experiment where the performance of a feedback loop with an observer is compared with a set of baselines. Deadline miss ratio is used as the controlled variable due to the simplicity of the sensor used to measure the deadline miss ratio (compared to, e.g., average task quality) and its well-defined behavior for varying load. Next we describe the simulator, baselines, and the results of the performance evaluation.

6.1 Simulator setup

The simulated workload consists of aperiodic tasks, as an aperiodic task set implies an increased unpredictability in the workload, hence, a greater challenge on the control of performance. The general outline of the feedback control scheduling architecture is given in Fig. 8. We assume a workload model where each task has two service levels, i.e. $N_i = 2$, $q_i(s_{i1}) = 0$, $q_i(s_{i2}) = 1$, $\hat{x}_i(s_{i1}) = x_i(s_{i1}) = 0$, $\hat{x}_i(s_{i2}) > 0$, and $x_i(s_{i2}) > 0$ (see Sect. 2.1), i.e., a task is either admitted for execution or rejected. Input to the controlled system is the set of arriving submitted tasks and the change to the admitted estimated workload $\delta_l(k)$. Output from the controlled system is the set of terminated tasks and $m_m(k)$. The goal is to minimize $(m_r - m(k))^2$ for each k. Based on $\delta_l(k)$, the admission controller enforces the workload adjustment. A task τ_i is admitted if its estimated load added to the estimated load of admitted tasks, i.e. $\hat{l}_i(s_{i2}) + l_{ad}(k)$ is less than the integrated value of $\delta_l(k)$. The workload model of the tasks is described as follows. The estimated execution time $\hat{x}_i(s_{i2})$ of a task τ_i is uniformly distributed according to $U : (50\text{ ms}, 300\text{ ms})$. Upon generation of a task an actual execution time given by the normal distribution $N : (\hat{x}_i(s_{i2}), \sqrt{\hat{x}_i(s_{i2})})$ is associated with τ_i. The deadline is set to $a_i + \hat{x}_i(s_{i2}) \times$ slackfactor, where a_i denotes the

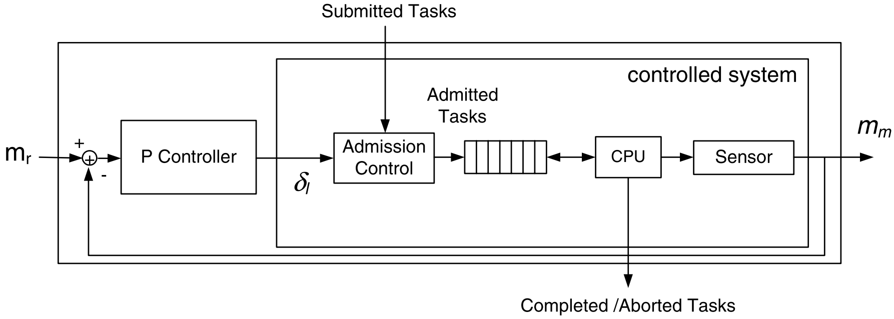


Fig. 8 The simulated system architecture

arrival time of τ_i and slackfactor is uniformly distributed according to $U:(10,30)$. The inter-arrival time is exponentially distributed with the mean inter-arrival time set to $\hat{x}_i(s_{i2}) \times$ slackfactor.

In our experiments, one simulation run lasts for 1000 seconds of simulated time. For all the performance data, we have taken the average of 10 simulation runs and derived 95% confidence intervals.

6.2 Modeling and controller design

We first describe the tuning of g_m in the model (11), followed by the tuning procedure of the controller used. The system is profiled in open-loop, i.e., without a controller, and the admitted load is increased in steps of 0.10, while measuring the deadline miss ratio. The derivative of the measured deadline miss ratio $m_m(k)$ is formed at the vicinity of the reference ($m_r = 0.10$) giving that $g_m \approx 1$.

By forming the $\mathcal{Z}$-transform of the model (11), we find that the transfer function from the control input $\delta_l(k)$ to the controlled variable $m(k)$ is $G_m(z) = \frac{g_m}{z-1}$. We employ P control (Franklin et al. 1998; Glad and Ljung 2000), i.e., $\delta_l(k) = K_P(y_r(k) - \hat{y}(k))$, where K_P is the P controller parameter. Using a P controller gives that the closed-loop transfer function from $m_r(k)$ to $m(k)$ is $G_{m,c}(z) = \frac{K_P g_m}{z-(1-K_P g_m)}$. Assuming that the closed-loop system $G_{m,c}(z)$ is stable, then the final value theorem (Franklin et al. 1998) gives that the steady-state error of $m(k)$,

$$E = m_r - \lim_{k\to\infty} m(k) = m_r - \lim_{z\to 1}(z-1)\frac{m_r z}{z-1}\frac{K_p g_m}{z-(1-K_p g_m)} = m_r - m_r = 0$$

is zero. The zero steady-state error can be directly observed since the controlled system has an integral part $\frac{1}{z-1}$ and, thus, an integral part for the controller is not needed to remove the steady-state error. Therefore we use a P controller and we tune K_p such that a pole at zero is obtained, i.e., $K_P = \frac{1}{g_m}$. Hence, the closed-loop system is stable since the pole is within the unit circle (Franklin et al. 1998) and, as such, the steady-state error of the controlled variable is zero according to above. The system profiling gave that $g_m \approx 1$, hence, $K_P = 1$ according to above. We use the same controller for all experiments, i.e., the same controller is used whether or not the observer is used.

6.3 Performance metrics

In Sect. 3 we argued that the goal of feedback control is to minimize the difference between the controlled variable $m(k)$ and its reference $m_r(k)$. Recall that it is not possible to obtain the value of $m(k)$, as we only measure $m_m(k)$. Therefore, we distinguish the performance of controllers by how well they force $m_m(k)$ to follow $m_r(k)$. We introduce the performance metrics,

$$J_a = \frac{1}{S}\sum_{k=1}^{S}\left|m_r - m_m(k)\right|$$

$$J_s = \frac{1}{S}\sum_{k=1}^{S}\left(m_r(k) - m_m(k)\right)^2$$

where S is the number of samples taken. The metric J_a gives the average difference between $m_m(k)$ and $m_r(k)$, whereas J_s gives the average squared difference. The lower J_s and J_a are, the better a controller is able to keep $m_m(k)$ near $m_r(k)$, and also the faster $m_m(k)$ converges toward $m_r(k)$.

6.4 Baselines

We compare the approach presented in this paper with the following baselines. First, we consider the baseline where no observer is used, i.e., the measured deadline miss ratio $m_m(k)$ is fed back and compared with the reference $m_r(k)$.

For the second baseline we use a moving average filter where the estimate of $m(k)$ is computed according to,

$$\hat{m}(k) = (1-\beta)\hat{m}(k-1) + \beta m_m(k). \tag{32}$$

The forgetting factor $0 < \beta \leq 1$ must be set to achieve a good balance between tracking of deadline miss ratio and efficient suppression of the measurement disturbance.

As a third baseline we have used a sliding window filter,

$$\hat{m}(k) = \alpha_0 m_m(k) + \alpha_1 m_m(k-1) + \cdots + \alpha_n m_m(k-n) \tag{33}$$

where $0 \leq \alpha_i \leq 1$ and $\alpha_0 + \cdots + \alpha_n = 1$. The difficulty in using a sliding window lies in the choice of n and α_i. As a general rule, n should be large enough to suppress the measurement disturbance and the weights should be chosen such that $\alpha_n < \alpha_{n-1} < \cdots < \alpha_0$ to track recent changes in $m(k)$.

6.5 Evaluation of controller performance

We divide the evaluation into two parts in order to clarify the presentation of the results.

6.5.1 Results for the observer

We know that the control performance is directly related to $R_{\delta wm}(k)$, which in turn depends on $n_\Theta(k)$. Therefore it is interesting to observe the performance of a feedback loop as $n_\Theta(k)$ and, hence, T varies. We show that using the deadline miss ratio observer (14), where $y = m$, significantly reduces J_a and J_s for low sampling periods. This implies that the measurement disturbance is suppressed, resulting in a more efficient control of the deadline miss ratio. In this experiment we vary m_r according to $0.05, 0.10, \ldots, 0.30$, and vary T according to $0.25, 0.50, 1.00, 1.50, \ldots, 5.00$ s. Varying the sampling period from 0.25 s to 5.00 s enables us to study the effect of using an observer for systems that need fast response (low T), and also systems that tolerate lower sampling rates and slower response (great T). For example, a sampling period of 0.50 s is chosen in (Lu et al. 2002), whereas a sampling period of 5.00 s is chosen in (Amirijoo et al. 2006). The results of the experiments are shown in Figs. 9–14.

Considering Fig. 9(a), we see that the number of terminated tasks $n_\Theta(k)$ increases as T increases and, hence, as T increases we expect a lower measurement disturbance variance. Further, $n_\Theta(k)$ also increases as the deadline miss ratio reference m_r increases. An increase in T and, consequently, an increase in $n_\Theta(k)$ results in a lower measurement disturbance, meaning that we can rely on the measured deadline miss ratio to a greater extent, which corresponds to an increase in $K_m(k)$. This is shown in Fig. 9(b), where $K_m(k)$ increases as T increases. Note that $K_m(k)$ is slightly greater for $T = 0.25$ s compared to $T = 0.50$ s. Recall that $K_m(k)$ decreases as the ratio $\frac{R_{\delta wm}(k)}{R_{\delta wl}(k)}$ increases. In our measurements we have noted that $\frac{R_{\delta wm}(k)}{\hat{R}_{\delta wl}(k)}$ increases as T decreases, except for $T = 0.25$ s, where we have actually noted a decrease in $\frac{R_{\delta wm}(k)}{\hat{R}_{\delta wl}(k)}$ and, as such, an increase in $K_m(k)$. The decrease in $\frac{R_{\delta wm}(k)}{\hat{R}_{\delta wl}(k)}$ is due to a significant increase in $\hat{R}_{\delta wl}(k)$, which is caused by the measurement disturbance affecting the manipulated variable $\delta_l(k)$. However, as we will see, the increase in $K_m(k)$ does not affect J_s and J_a considerably and we achieve much better performance control compared to the case when an observer is not used.

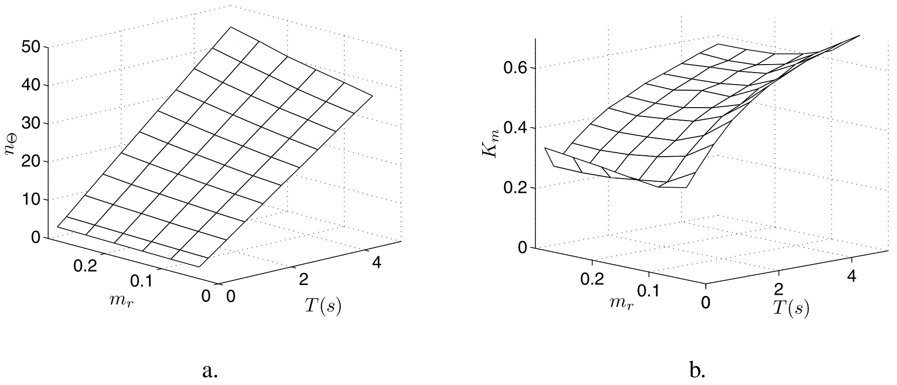
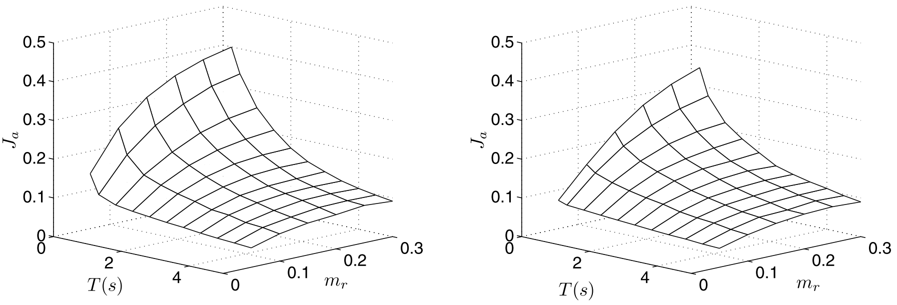


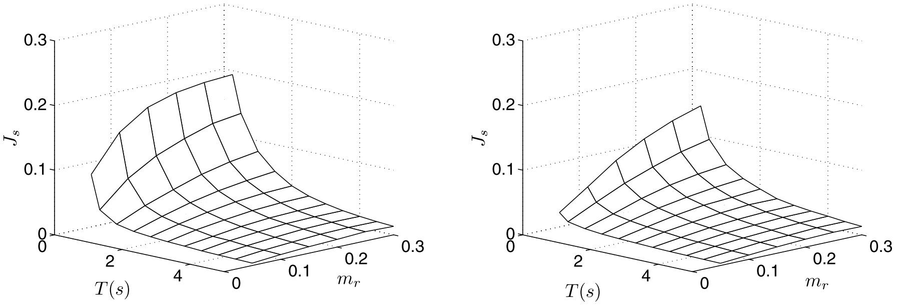
Fig. 9 Measured n_Θ (figure a) and K_m (figure b) when varying m_r and T



a. Observer not used

b. Observer used

Fig. 10 Measured J_a



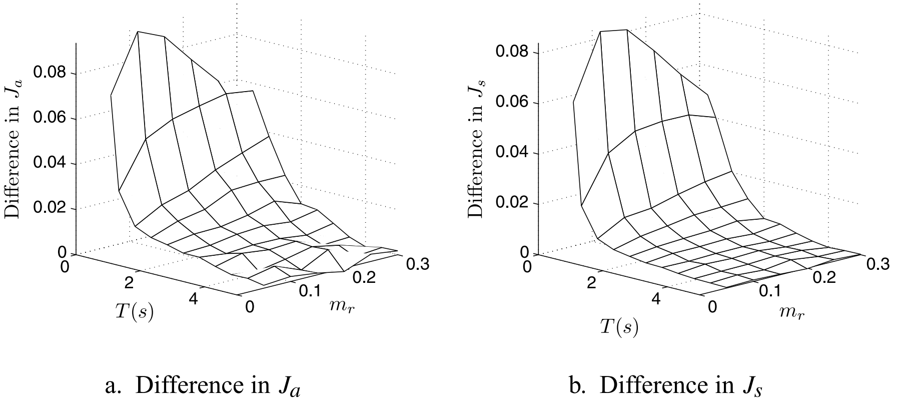
a. Observer not used

b. Observer used

Fig. 11 Measured J_s

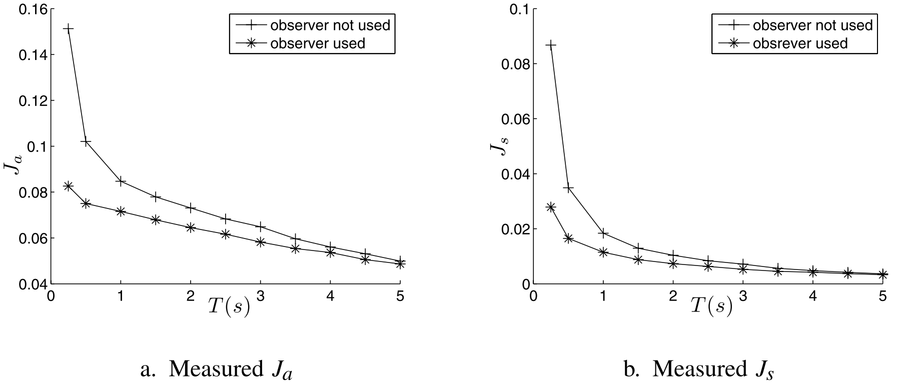
The measured J_a and J_s when an observer is used (figure b), respectively not used (figure a), are presented in Figs. 10 and 11. Remember that as J_a and J_s decrease, the actual system performance is closer to the desired system performance. The difference in performance of a traditional feedback loop and a feedback loop with an observer is given in Fig. 12. The difference in J_a and J_s between the two approaches decreases as T increases due to the decreasing variance of the measurement disturbance. For low sampling periods, a significant decrease in J_a and J_s is achieved when an observer is used as compared to the case when an observer is not used. This clearly shows the gain in performance for low sampling periods when an observer is used to suppress the measurement disturbances.

The case when $m_r = 0.05$ is given in Fig. 13. At $T = 0.25$ s, we have that J_a is 0.08 ± 0.00 when an observer is used, compared to 0.15 ± 0.01 when an observer is not used, i.e., we have a difference of about 0.07. This means that in average, using an observer improves the performance such that $m_m(k)$ is closer to the reference by 0.07. Similarly, the difference in J_s is 0.06. Hence, the measured deadline miss ratio is significantly closer to its reference when an observer is used for low sampling periods.



a. Difference in J_a b. Difference in J_s

Fig. 12 Difference in performance with and without an observer



a. Measured J_a b. Measured J_s

Fig. 13 Measured J_s and J_s when $m_r = 0.05$

We also study $m_m(k)$ in the time domain to obtain a better understanding of how a certain J_a corresponds to variations in $m_m(k)$. Figure 14 shows the deadline miss ratio in the time domain for the experiment corresponding to $T = 0.50$ s and $m_r(k) = 0.10$. As we can see $m_m(k)$ oscillates heavily around the reference in the absence of an observer, as shown in Fig. 14(a). However, the deviations are significantly reduced when using an observer, as shown in Fig. 14(b). In the absence of the observer, large deviations in $m_m(k)$ due to the measurement disturbance are not filtered. As a consequence, the controller tries to compensate for the changes in $m_m(k)$ by changing the requested workload, which results in an over compensation and, hence, $m_m(k)$ deviates even more from $m_r(k)$. However, an observer is able to suppress variations in $m_m(k)$ due to the measurement disturbance or equivalently the averaging operation. Consequently, a less noisy measurement is presented to the controller, which in turn enhances the control performance.

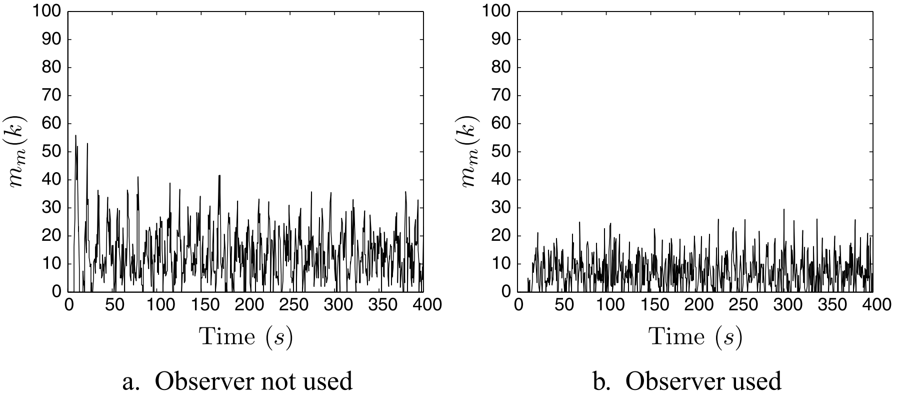


Fig. 14 $m_m(k)$ in the time domain

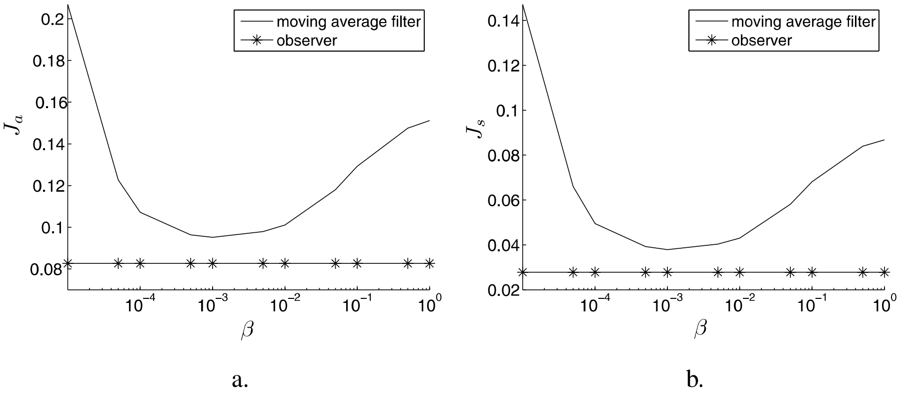


Fig. 15 Control performance when a moving average filter is used and $T = 0.25$ s and $m_r = 0.05$. The corresponding value for the observer is taken from Fig. 13. Note the logarithmic scale on the x-axis

6.5.2 Results for the moving average and sliding window filters

Let us now turn to the two remaining baselines, namely the moving average and the sliding window filters. For the following experiments we set the sampling period to 0.25 s, i.e., $T = 0.25$ s and $m_r = 0.05$.

Recall from Sect. 6.4 that we have to tune the parameter β of the moving average filter (32) in order to obtain a good performance. For this reason we vary β from 10^{-5} to 1 and the result is shown in Fig. 15. The performance of the controller when using an observer is always better than the performance when using a moving average filter. For $\beta < 10^{-3}$ the moving average filter does not adequately track changes in deadline miss ratio and this causes a significant overshoot in $m(k)$ resulting in greater J_a and J_s.

Now we consider the control performance under weighted sliding window filtering (33). We consider two windows, i.e., $n = 1$ with $\alpha_0 = \alpha$ and $\alpha_1 = 1 - \alpha$. We vary α from 0.5 to 1.0 and the results are shown in Fig. 16. The control performance

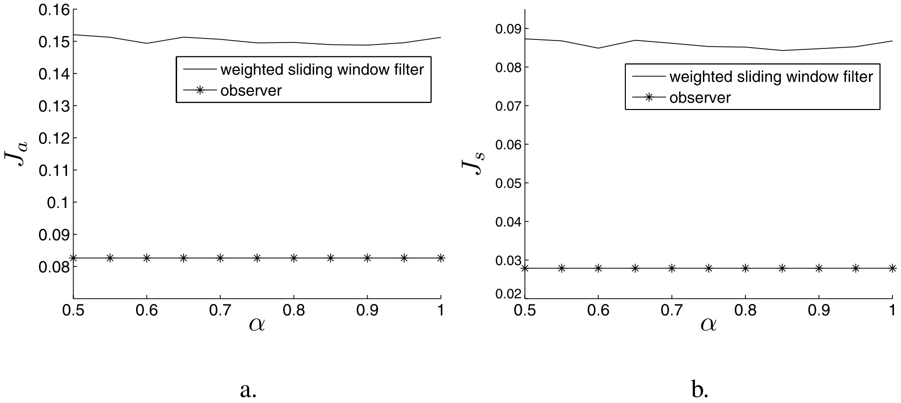


Fig. 16 Control performance when a sliding window filter is used and $T = 0.25$ s and $m_r = 0.05$. The corresponding value for the observer is taken from Fig. 13

under sliding window filtering is inferior to the case when an observer is used. We also consider the case when $n = 5$, which results in a similar control performance as when $n = 1$. Hence, the experiments show that the control performance is significantly worse when using a sliding window filter compared to when using an observer.

Note that in the experiments above we have not redesigned the controller (recomputing K_p) for each value of β (or α_i). To improve the control performance we need to take a co-design approach where we iteratively try to find the best combination of β (or α_i) and K_p. Using the observer (14) presented in this paper alleviates this difficulty, since the design of the controller and the observer are disjoint (see Sect. 5.3).

6.5.3 Performance summary

In summary we have shown that a lower sampling period increases the disturbance in the measurements. The performance is improved when using an observer for suppressing the measurement disturbance. We have observed that $m_m(k)$ is closer to the reference $m_r(k)$, which demonstrates achieved improved performance reliability. Hence, by using an observer, the actual system performance gets closer to the desired system performance compared to the case when the observer is not used, and moving average and sliding window filtering is used.

7 Related work

Lu et al. (2002) introduced a feedback control scheduling framework for controlling utilization and deadline miss ratio. In their model they assume that each task has several QoS-levels giving results of varying quality. Each QoS-level is characterized by a set of attributes, such as period, deadline, and utilization. This work by Lu et al. (2002) is extended in this paper by adding a control structure that suppresses the measurement disturbance.

Li and Nahrstedt (1998) proposed a task model for quality of service (QoS) control. In their model, there are dependencies among the tasks, characterized by input quality and output quality. The goal is to design controllers that force the target task to maintain the same output quality at a desired QoS reference. Changing the periodicity of a set of tasks in response to load variations has been suggested by Buttazzo and Abeni (2002). If the estimated load is found to be greater than a threshold, task periods are enlarged to find the desired load. In the work of Cervin et al. (2002), an approach is presented for optimizing the performance of a set of control tasks. The rate of the control tasks is adjusted, such that the utilization is kept close to a reference point.

Controlling the queue length is the key to guarantee timely processing of requests and packets in server systems and networks. Parekh et al. (2002) use feedback control scheduling to control the length of a queue of remote procedure calls arriving at a server. Abdelzaher et al. (2003) presented control algorithms for managing service delay and queue length of requests arriving at web servers. Robertson et al. (2003) used a nonlinear fluid model expressed in terms of a differential equation. The model is linearized and a PI controller is tuned to control the queue length. Sha et al. (2002) used a feedback controller in combination with a queuing model predictor to adjust the queue length of services. The results show that it is beneficial to use a combined approach, resulting in the actual queue length to be closer to the reference compared to a traditional approach of only using feedback control.

Efforts have been carried out trying to reduce energy consumption in real-time systems, while preserving timely completion of tasks (Zhu and Mueller 2004). In this case execution times are monitored and the voltage and, thus, frequency of the CPU is varied such that the power consumption is reduced and tasks are executed in a timely manner. A similar problem was studied by Sharma et al. in the context of servers (Sharma et al. 2003).

None of the approaches above have considered the effects of the sampling period on the measured variable. In our earlier work we discussed the effects of the sampling period and we introduced a measurement suppressive control structure to reduce the effects of the disturbance arising when measuring the utilization and deadline miss ratio (Amirijoo et al. 2005). In this work we have extended our previous results by the following contributions: (1) A new controlled variable, namely the average task quality, has been added. (2) An optimal time variant observer is introduced to suppress the measurement disturbance, while our previously published observer is suboptimal. (3) The system disturbance estimation is extended such that off-line profiling is no longer needed, hence, the estimation is carried out during run-time. (4) A new method for quantifying the measurement disturbance is introduced, where the computational complexity of the new method is substantially reduced.

8 Conclusions

The emergence of real-time systems operating in open and unpredictable environments has resulted in a paradigm shift in techniques for managing system resources. Using feedback control has shown to be effective for a large class of real-time systems with unpredictable workload characteristics. Although there is a great body of

Table 3 Table of commonly used variables

Attribute	Description	Defined on page
δ_l	Change to the estimated admitted workload	49
δ_{wl}	System disturbance	51
δ_{wu}	Measurement disturbance of utilization	51
δ_{wm}	Measurement disturbance of deadline miss ratio	52
δ_{wq}	Measurement disturbance of average task quality	52
J_a	Average absolute performance error	67
J_s	Average squared performance error	67
K_y	Estimator feedback gain	54
l	Admitted workload	51
l_{ad}	Measured estimated admitted workload	63
l_i	Average load of task τ_i	46
m	Deadline miss ratio	48
m_m	Measured deadline miss ratio	48
n_Θ	Number of terminated tasks	48
n_T	Total number of monitored time units	47
q	Average task quality	48
q_i	Out quality of task τ_i when terminating	46
q_m	Measured average task quality	48
R	Disturbance variance	53
s_{ij}	Service level of task τ_i	46
T	Sampling period	47
u	Utilization	48
u_m	Measured utilization	47
x_i	Execution time of task τ_i	46

knowledge in the control community dealing with the control of dynamic systems, not all techniques and results can be directly mapped into the domain of computer performance control. The measured variables typically used to describe the performance of computer systems are formed over a data set. In this paper we have shown how the sampling period selection influences the characteristics of the measurements and, hence, the control performance. The disturbance in the measurement increases as the sampling period decreases, due to the decreasing size of the data set that is used to compute the measured variable. Still a large sampling period is not desired as the control would become less responsive to changes in the controlled variable. To solve the problem of the sampling period selection we have proposed an approach consisting of choosing a suitable sampling period to capture the system dynamics, and an observer that produces estimates of the controlled variable. Experimental results show that this approach results in improved performance control as the actual performance is closer to the desired level. This increases the reliability of the system and implies a more controlled worst-case performance and faster convergence toward the desired performance.

In our future work we will consider other types of controlled variables, e.g., response time and average queue length. Also we aim at extending our task model such that dependence among the tasks is captured.

References

Abdelzaher TF, Shin KG, Bhatti N (2002) Performance guarantees for web server end-systems: a control-theoretical approach. IEEE Trans Parallel Distrib Syst 13(1):80–96

Abdelzaher TF, Stankovic JA, Lu C, Zhang R, Lu Y (2003) Feedback performance control in software services. IEEE Control Syst Mag 23(3):74–90

Amirijoo M, Hansson J, Gunnarsson S, Son SH (2005) Enhancing feedback control scheduling performance by on-line quantification and suppression of measurement disturbance. In: Proceedings of the IEEE real-time and embedded technology and applications symposium (RTAS)

Amirijoo M, Hansson J, Son SH (2006) Specification and management of QoS in real-time databases supporting imprecise computations. IEEE Trans Comput 55(3):304–319

Brandt S, Nutt G, Berk T, Mankovich J (1998) A dynamic quality of service middleware agent for mediating application resource usage. In: Proceedings of the IEEE real-time systems symposium

Buttazzo GC, Abeni L (2002) Adaptive workload management through elastic scheduling. Real-time Syst 23(1/2)

Cervin A, Eker J, Bernhardsson B, Årzén K (2002) Feedback-feedforward scheduling of control tasks. Real-time Syst 23(1/2)

Davidson S, Watters A (1988) Partial computation in real-time database systems. In: Proceedings of the workshop on real-time software and operating systems

Fausett LV (2003) Numerical methods: algorithms and applications. Prentice Hall, New York

Franklin GF, Powell JD, Workman M (1998) Digital control of dynamic systems, 3rd edn. Addison–Wesley, Reading

Glad T, Ljung L (2000) Control theory—multivariable and nonlinear methods. Taylor and Francis, London

Hellerstein JL, Diao Y, Parekh S, Tilbury DM (2004) Feedback control of computing systems. Wiley/IEEE Press, London

Li B, Nahrstedt K (1998) A control theoretical model for quality of service adaptations. In: Proceedings of the international workshop on quality of service

Liu JWS, Shih W-K, Lin K-J, Bettati R, Chung J-Y (1994) Imprecise computations. Proceedings of the IEEE 82

Lu C, Stankovic JA, Tao G, Son SH (2002) Feedback control real-time scheduling: framework, modeling and algorithms. Real-time Syst 23(1/2)

Lu Y, Saxena A, Abdelzaher TF (2001) Differentiated caching services; a control-theoretical approach. In: Proceedings of the international conference on distributed computing systems (ICDCS)

Oppenheim AV, Willsky AS (1996) Signals and systems, 2nd edn. Prentice Hall, New York

Parekh S, Gandhi N, Hellerstein J, Tilbury D, Jayram T, Bigus J (2002) Using control theory to achieve service level objectives in performance management. Real-time Syst 23(1/2)

Robertson A, Wittenmark B, Kihl M (2003) Analysis and design of admission control in Web-server systems. In: Proceedings of American control conference (ACC)

Sha L, Liu X, Lu Y, Abdelzaher T (2002) Queuing model based network server performance control. In: Proceedings of real-time systems symposium (RTSS)

Sharma V, Thomas A, Abdelzaher T, Skadron K, Lu Z (2003) Power-aware QoS management in Web servers. In: Proceedings of real-time systems symposium (RTSS)

Vrbsky SV, Liu JWS (1993) APPROXIMATE—a query processor that produces monotonically improving approximate answers. IEEE Trans Knowl Data Eng 5(6):1056–1068

Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H. 264/AVC video coding standard. IEEE Trans Circ Syst Video Technol 13

Zhu Y, Mueller F (2004) Feedback EDF scheduling exploiting dynamic voltage scaling. In: Proceedings of the IEEE real-time and embedded technology and applications symposium (RTAS), pp 84–93

Zilberstein S, Russell SJ (1996) Optimal composition of real-time systems. Artif Intel 82(1–2):181–213

Mehdi Amirijoo received his M.Sc. degree in computer science and engineering, and Ph.D. degree in computer science from Linköping University, Sweden, in 2002 and 2007 respectively. His interests include real-time systems, QoS management, automatic control, software engineering, and telecommunication. His research has focused on QoS management of real-time data services using automatic control for managing uncertainties in workload. This includes performance specification models, modeling of real-time systems performance, and architectures based on feedback control. He has published over 15 papers on real-time systems and QoS management using automatic control. He is currently affiliated with Ericsson Research working on automatic operation of radio networks including optimization, control, and configuration.

Jörgen Hansson received the B.Sc. and M.Sc. degrees from the University of Skövde, Sweden, in 1992 and 1993, respectively. He received the Ph.D. degree in 1999 from Linköping University, Sweden, with which he is also affiliated as an associate professor. He is a senior member of the technical staff at the Software Engineering Institute at Carnegie Mellon University. His current research interests include real-time systems and real-time database systems. His research has focused on techniques and algorithms for ensuring robustness and timeliness in real-time applications that are prone to transient overloads, mechanisms and architectures for handling increasing amounts of data in real-time systems, and algorithms to ensure data quality in real-time systems. His current research interests include modeling and validation of software and system architectures using model-based engineering frameworks, and software architectures for embedded and real-time systems. He is a member of the IEEE and ACM.

Svante Gunnarsson was born in Tranås, Sweden, 1959. He received the M.Sc. degree in Applied Physics and Electrical Engineering from Linköping University, Sweden, in 1983. He received his Ph.D. in Automatic Control from Linköping University, Sweden, in 1988, and is currently Professor in the Control group at the Department of E.E., Linköping University, Sweden. His research interests are in the areas of system identification, iterative learning control, and robot control.

Sang H. Son is a Professor at the Department of Computer Science of University of Virginia. He received the B.S. degree in electronics engineering from Seoul National University, M.S. degree from KAIST, and the Ph.D. in computer science from University of Maryland, College Park.

Prof. Son is the Chair of the IEEE Technical Committee on Real-Time Systems. He is serving as an Associate Editor for IEEE Transactions on Computers, Real-Time Systems Journal, Journal of Information Processing Systems, and Journal of Computing Science and Engineering. He has served as an Associate Editor of IEEE Transactions on Parallel and Distributed Systems, and Journal of Business Performance Management.

His research interests include real-time systems, database and data services, QoS management, wireless sensor networks, and information security. He has written or co-authored over 250 papers and edited/authored four books in these areas. He has served as the Program Chair or General Chair of several real-time and database conferences, including IEEE Real-Time Systems Symposium and International Conference on Networked Sensing Systems. He is the Program Chair of the IEEE Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), 2008. He received the Outstanding Contribution Award at the IEEE Conference on Embedded and Real-Time Computing Systems and Applications in 2004.