

Enhancing Feedback Control Scheduling Performance by On-line Quantification and Suppression of Measurement Disturbance*

Mehdi Amirijoo*, Jörgen Hansson*, Svante Gunnarsson[◇]

*Dept. of Computer and Information Science

[◇]Dept. of Electrical Engineering

Linköping University, Sweden

{meham,jorha}@ida.liu.se

Sang H. Son

Dept. of Computer Science

University of Virginia, Charlottesville, Virginia, USA

son@cs.virginia.edu

Abstract

In the control of continuous and physical systems, the controlled system is sampled sufficiently fast to capture the system dynamics. 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 quantize the disturbance present in the measured variable as a function of the sampling period and we propose a measurement disturbance suppressive control structure. 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 and adaptation.

1. Introduction

In recent years a new class of soft real-time systems operating in open environments 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 the resource requirements of tasks are in general unknown. Feedback control scheduling has been introduced as a promising foundation for performance control of complex real-time systems [2, 5–7, 9]. It has been shown that feedback control is

highly effective to support the specified performance of dynamic systems that are both resource insufficient and that 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 [3]. 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. This in turn 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. For example, consider a service provider streaming video to a set of clients. The service provider sets the reference quality of the streams to a certain level, and also gives requirements on worst-case quality and how fast the quality should converge toward the ref-

* 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 CCR-0098269 and IIS-0208758.

erence quality in the case of a transient overload or a disturbance in the system (e.g., a server goes down). Now, using a more accurate measurement of the controlled variable, enables the service provider to provide streams with more reliable quality and faster performance adaptation, as compared to using less accurate measurements. Hence, the quality is regulated much more efficiently even in the case of transient system overloads.

The contributions of this paper are as follows: (i) a model of the controlled system incorporating the uncertainty in measurements, induced by the data set size, (ii) a method for quantizing the uncertainty of a measurement, given the data set size, and (iii) a feedback control structure that is less sensitive to the uncertainty of the measured variable. To the best of our knowledge, this is the first paper proposing a model of the measurement disturbance and describing a feedback control structure that suppresses the measurement disturbance. We show through experiments that this approach results in a faster QoS adaptation and reliability in the face of changes to the controlled system, compared to using a traditional feedback structure where no disturbance suppression is done.

The remainder of this paper is organized as follows. The problem formulation and the task model are given in Sections 2 and 3. In Section 4 we present our approach, and in Section 5, we evaluate the performance of the disturbance suppressive feedback structure. In Section 6 we give an overview on related work, followed by Section 7, where conclusions and future work are discussed.

2. Problem Formulation

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 is the sampling period and k is the sampling instant. For the rest of this paper, we sometimes drop k where the notion of time is not of primary interest. Typically, one is interested in controlling the performance of real-time systems using the metrics utilization or deadline miss ratio. We define the utilization as, $u(k) = \frac{t_B(k)}{t_T}$, where $t_B(k)$ is the number of time units that the system is busy computing, and t_T is the total number of monitored time units. We say that a task is terminated when it has completed or missed its deadline. Let the deadline miss ratio, $m(k) = \frac{n_M(k)}{n_T(k)}$, be the ratio of tasks that have missed their deadline, where $n_M(k)$ is the number of tasks that have missed their deadline and $n_T(k)$ is the number of terminated admitted tasks in the time interval $[(k-1)T, kT]$.

The deadline miss ratio or the utilization are controlled using the feedback structure, as shown in Figure 1. Let $y(k)$ denote $m(k)$ or $u(k)$. Input to the controller is the difference between the reference $y_r(k)$, representing the desired

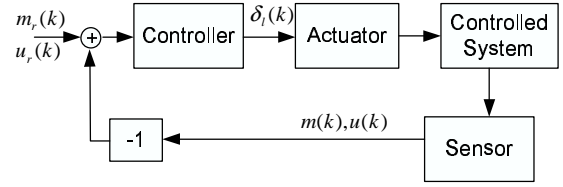


Figure 1. Feedback loop structure.

performance of the controlled system, and the actual system performance given by the measured variable $y(k)$. Based on the performance error $y_r(k) - y(k)$ the controller computes a change $\delta_l(k)$ to the estimated admitted workload $l(k)$. The control problem is how to compute the manipulated variable $\delta_l(k)$ such that the difference between the desired performance and the actual performance is minimized, i.e., we want to minimize $(y_r(k) - y(k))^2$. We note that it is impossible to keep $y(k)$ at $y_r(k)$. Before justifying this claim, let \bar{y} denote the average value of $y(k)$ and let the variance of $y(k)$ be the expected value of the squared deviation from \bar{y} , i.e., $R_y = E\{(y(k) - \bar{y})^2\}$.

Intuitively, the variance in utilization, i.e. $R_u = E\{(u(k) - \bar{u})^2\}$, and the variance in deadline miss ratio, i.e. $R_m = E\{(m(k) - \bar{m})^2\}$, are caused by transient increases or decreases in load, due to inaccurate execution time estimation errors, resource conflicts causing blocking, restart, and abortion of tasks. We refer to unpredictable changes in load as the system disturbance, causing variance in $u(k)$ and $m(k)$. However, there is also a second component contributing to variations in $u(k)$ and $m(k)$, namely, the disturbance arising from the averaging operation. For example, consider that we are measuring deadline miss ratio over a set of tasks for a given load. We note that R_m increases as $n_T(k)$ decreases. In the extreme case when $n_T(k)$ is one, $m(k)$ is equal to one or zero. Since $n_T(k)$ is small we obtain a poor presentation of the actual system performance, which in turn implies a negative effect on the feedback control scheduling performance, i.e., we observe large deviations between $m(k)$ and $m_r(k)$. We view the type of variance that arises from the averaging operation as a measurement disturbance that deteriorates the actual representation of the system. Now, we know that $n_T(k)$ increases as T increases and, thus, by increasing the sampling period we lower R_u and R_m . However, this typically degrades the responsiveness of the controller [3], resulting in a slower reaction to changes in $u(k)$ and $m(k)$. 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 formalize and model system and measurement distur-

Attribute	Periodic Tasks	Aperiodic Tasks
d_i	$d_i = p_i$	$d_i = r_{A,i}$
$l_{E,i}[s_j]$	$l_{E,i}[s_j] = x_{E,i}[s_j]/p_i$	$l_{E,i}[s_j] = x_{E,i}[s_j]/r_{E,i}$
$l_{A,i}[s_j]$	$l_{A,i}[s_j] = x_{A,i}[s_j]/p_i$	$l_{A,i}[s_j] = x_{A,i}[s_j]/r_{A,i}$

Table 1. The assumed task model.

bances? How can we quantify the measurement disturbance? 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 utilization and deadline miss ratio give rise to measurement disturbance and also how we can use this knowledge to lower the measurement disturbance to achieve better performance control.

3. Task Model

We consider a real-time system as the controlled system, where there is one CPU as the main processing element. A task τ_i is classified as either a periodic or an aperiodic task. See Table 1 for a complete task model. A task τ_i has a set of QoS levels $\{s_1, \dots, s_j, \dots, s_N\}$, where $N \geq 2$. A task has the following characteristics: period p_i (periodic tasks), estimated mean inter-arrival time $r_{E,i}$ (aperiodic tasks), actual mean inter-arrival time $r_{A,i}$ (aperiodic tasks), relative deadline d_i , estimated execution time $x_{E,i}[s_j]$, actual execution time $x_{A,i}[s_j]$, estimated load, $l_{E,i}[s_j]$, actual load, $l_{A,i}[s_j]$, and quality of result $q_i[s_j]$ produced when terminated, where $0 \leq q_i[s_j] \leq 1$.

The quality $q_i[s_j]$ of τ_i increases as j increases. Similarly, $x_{E,i}[s_j]$ and $x_{A,i}[s_j]$ increase as j increases. In the simplest case when $N = 2$, $q_i[s_1] = 0$, $q_i[s_2] = 1$, $x_{E,i}[s_1] = x_{A,i}[s_1] = 0$, $x_{E,i}[s_2] > 0$, $x_{A,i}[s_2] > 0$, represents a task model where tasks are either rejected or admitted for execution. Upon arrival, a task presents its estimated average load $l_{E,i}[s_j]$ and its relative deadline d_i to the system. The actual load of the task $l_{A,i}[s_j]$ is not known in advance due to variations in execution time.

4. Approach

4.1. An Overview of the Approach

Generally, the performance of a controller improves with decreasing sampling period as the closed-loop system becomes more responsive to the difference between the controlled variable and its reference [3]. The reference may vary over time as a result of changes to the desired performance, or the controlled variable may vary due to changes

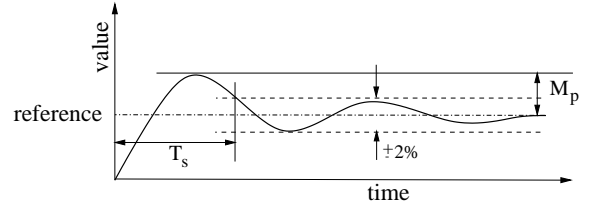


Figure 2. Definition of settling time (T_s) and overshoot (M_p).

in workload characteristics, e.g., a sudden change in execution time estimation error. The desired transient-state performance is usually expressed in terms of the maximum overshoot and the settling time [3, 5, 6], as shown in Figure 2. The maximum overshoot M_p is the worst-case system performance in the transient system state. The settling time T_s is the time for the transient overshoot to decay and reach the steady state performance and, hence, it is a measure of system adaptability.

Now assume that the sampling period is T . At time kT the feedback loop is closed, meaning that the controlled system is sampled, the manipulated variable is recomputed, and the actuator is invoked. Between each sampling no feedback information is sent to the controller and the system is running in open-loop and unresponsive to changes in the controlled variable. By definition, if the reference or the controlled variable suddenly changes at kT , then it takes T_s seconds for the controlled variable to reach the steady-state. However, consider the case when the controlled variable suddenly increases immediately after kT , e.g., the deadline miss ratio increases due to an execution time estimation error. Then, in the worst-case it takes $T_{wcs} = T + T_s$ for the controlled variable to reach the steady-state, as the change in the controlled variables is not detected until the next sampling instant at time $(k+1)T$. This is shown in Figure 3, where at time $4.5s$ the controlled variable suddenly increases. This increase is not detected until time $8s$ when the controlled variable is sampled. As T_s is $4s$, it takes a total of $7.5s$ for the controlled variable to reach steady-state. Hence, T must be chosen such that T_{wcs} satisfies the desired response time.

Further T_s may be influenced by T depending on the methodology used when tuning the controllers. Assume that we want to control $m(k)$ by using a proportional-controller (P-controller) [3] that is tuned by following the method presented in [6]. We apply a step to the reference to simulate the effect of a change in reference or a transient overload [6]. The settling time of $m(k)$ is $9s$ when $T = 1s$, whereas the settling time of $m(k)$ is $90s$ when $T = 10s$. Hence, the settling time increases as the sampling period increases, meaning that the controller is less responsive to

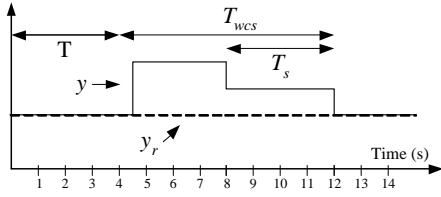


Figure 3. The worst-case settling time depends on the sampling period.

changes in the reference or transient overloads.

We advocate an approach where the sampling period is chosen such that the transient-state specification based on the worst-case settling time, e.g. $T_{wcs} \leq 10s$ and $M_p \leq 10\%$, is satisfied and where the effects of the measurement disturbance due to the chosen sampling period is suppressed using estimators. For example, if T_{wcs} must be less than $10s$, and T_s is simulated to be $9s$, then T must be less or equal to $1s$. In the following sections we provide a model of the controlled system and we show that the measurement disturbance due to the averaging operation increases significantly as the sampling period decreases. We propose a feedback control structure that suppresses the measurement disturbance, resulting in an enhancement in the performance control.

4.2. A State-Space Model of the Controlled System

We adopt the linear and time-invariant model presented by Lu et al. [6], describing the utilization and the deadline miss ratio in the face of changes to estimated requested workload. We choose this model 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.

Starting from the control input, the workload of admitted tasks $l(k+1)$ in the next sampling period is changed through the manipulated variable $\delta_l(k)$ and the system disturbance δ_{wl} , given by

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

As mentioned previously in Section 2, the system disturbance $\delta_{wl}(k)$ arises from incomplete knowledge about the controlled system, e.g. unknown execution times and resource conflicts.

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. When $l(k)$ is less or equal to 100% , 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 100% , then $u(k)$ remains at 100% , despite changes to $l(k)$. When the CPU is underutilized, we add a utilization measurement disturbance $\delta_{wu}(k)$ to $l(k)$ when forming $u(k)$, i.e.,

$$u(k) = \begin{cases} l(k) + \delta_{wu}(k), & l(k) + \delta_{wu}(k) \leq 100\% \\ 100\%, & l(k) + \delta_{wu}(k) > 100\%. \end{cases} \quad (2)$$

Continuing with $m(k)$, the relationship between the actual workload $l(k)$ and $m(k)$ is non-linear due to saturation. Let $l_{Th}(k)$ be the workload threshold of tasks in the k^{th} period for which admitted tasks are schedulable. $m(k)$ is saturated when $l(k) \leq l_{Th}(k)$, and remains zero despite changes to $\delta_l(k)$, i.e., $m(k) = 0$ when $l(k) \leq l_{Th}(k)$. When not saturated, $m(k)$ increases non-linearly with $l(k)$. Since feedback control relies on linear systems, we linearize the relationship between $l(k)$ and $m(k)$ at the vicinity of the performance reference m_r , i.e., $g_M = \frac{dm(k)}{dl(k)}$. To capture the deadline miss ratio measurement disturbance $\delta_{wm}(k)$, we model $m(k)$ as,

$$m(k) = g_M l(k) + \delta_{wm}(k). \quad (3)$$

Finally, under the conditions $l(k) \leq 100\%$ and $l_{Th} < l(k)$, we obtain the following state-space model from (1)-(3),

$$\begin{aligned} l(k+1) &= l(k) + \delta_l(k) + \delta_{wl}(k) \\ u(k) &= l(k) + \delta_{wu}(k) \\ m(k) &= g_M l(k) + \delta_{wm}(k), \end{aligned} \quad (4)$$

where $l(k)$ is the state of the system and $u(k)$ and $m(k)$ are the measured outputs of the system. Above, we have developed the state-space model (4) representing the dynamics of $u(k)$ and $m(k)$ given the input $\delta_l(k)$, the load disturbance $\delta_{wl}(k)$, and utilization and deadline miss ratio measurement disturbances $\delta_{wu}(k)$ and $\delta_{wm}(k)$, respectively. In the following sections we use this model to design control structures that suppress the measurement disturbance.

4.3. Suppressing the Measurement Disturbance

In the remaining of this section we focus on suppression of the deadline miss ratio disturbance $\delta_{wm}(k)$, as the suppression of the utilization measurement disturbance is analogous. Define $\hat{m}(k)$ as the estimation of a true signal $m(k)$. An initial step to suppressing the measurement disturbance is to use the open-loop estimator [4] given by,

$$\begin{aligned} \hat{l}(k+1) &= \hat{l}(k) + \delta_l(k) \\ \hat{m}(k) &= g_M \hat{l}(k). \end{aligned} \quad (5)$$

Basically, this estimator is running open-loop and not utilizing any incoming measurements. Eventually the estimations

would diverge from the true system state due to incomplete knowledge of the controlled system, e.g., inaccurate execution time estimations. However, if we compute the difference between the measured output and the estimated output and constantly correct the model with this error, the divergence is avoided. The idea is to construct a feedback system around the open-loop estimator (5) with the estimation error as feedback. Let $\hat{y}(k|b)$ be the estimated value of $y(k)$, predicted at time bT . For example, $\hat{y}(k|k-1)$ refers to the estimated value of $y(k)$, predicted at time $(k-1)T$. The new estimator is given by,

$$\begin{aligned}\hat{l}(k+1|k) &= \hat{l}(k|k) + \delta_l(k) \\ \hat{l}(k|k) &= \hat{l}(k|k-1) + K_m(m(k) - g_M \hat{l}(k|k-1)) \\ \hat{m}(k|k) &= \hat{m}(k) = g_M \hat{l}(k|k)\end{aligned}\quad (6)$$

where K_m is the deadline miss ratio 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 $m(k)$. Remember that $m(k)$ is related to $l(k)$ and, hence, $m(k)$ is an indirect measure of the current load. By setting K_m to a large value, the estimation follows the true system state to a larger extent, as the impact of the difference in measurement and estimation, i.e., $m(k) - \hat{m}(k)$, is large. However, a large K_m implies that the measurement disturbance has a large influence on the state estimation. Hence, if the measurement disturbance is small and the system disturbance is large, then we should choose a large K_m . In contrast, if the measurement disturbance is large and the system disturbance is small, then we set K_m to a small value to suppress the measurement disturbance. Applying this principle to the control of deadline miss ratio, we show in Section 4.4 that a large $n_T(k)$ implies a small measurement disturbance and, hence, K_m should be set to a large value. However, if $n_T(k)$ is small, meaning that the measurement disturbance is large, then K_m should be set to a small value to eliminate the disturbance due to the averaging operation.

To compute a suitable value for K_m , we need to model and quantify the magnitude of the system and measurement disturbances. 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 $k \neq b$. Similarly we assume that $\delta_{wu}(k)$ is uncorrelated with $\delta_{wu}(k+b)$, and that $\delta_{wl}(k)$ is uncorrelated with $\delta_{wl}(k+b)$. Hence, we model the system and the measurement disturbances as white noise [8], with zero average values, i.e.,

$$E\{\delta_{wu}(k)\} = E\{\delta_{wm}(k)\} = E\{\delta_{wl}(k)\} = 0. \quad (7)$$

and we define the variance of the system and the measure-

ment disturbances as,

$$\begin{aligned}R_{\delta_{wu}} &= E\{\delta_{wu}^2(k)\}, R_{\delta_{wm}} = E\{\delta_{wm}^2(k)\}, \quad (8) \\ R_{\delta_{wl}} &= E\{\delta_{wl}^2(k)\}.\end{aligned}\quad (9)$$

As $R_{\delta_{wu}}(k)$ and $R_{\delta_{wm}}(k)$ increase, the magnitude of the measurement disturbances increases. Having modeled the disturbances, the next step becomes to compute K_m , which is given by the following theorem.

Theorem 1 *The K_m that minimizes the variance of the estimation error $(l(k) - \hat{l}(k))^2$ is,*

$$K_m = \frac{g_M H_m}{g_M^2 H_m + R_{\delta_{wm}}}\quad (10)$$

where

$$H_m = \frac{R_{\delta_{wl}}}{2} + \sqrt{\frac{R_{\delta_{wl}}^2}{4} + \frac{R_{\delta_{wl}} R_{\delta_{wm}}}{g_M}}.\quad (11)$$

Proof: Given the model (4), where $\delta_{wl}(k)$ and $\delta_{wm}(k)$ are white noise, and the estimator (6), the optimal choice of K_m , follows directly from the Kalman filter, see e.g. [3, 4]. Equation (11) is the solution to the corresponding riccati equation [3, 4]. \square

In this regard, the estimator (6) where K_m is set according to Theorem 1 is an optimal estimator, meaning that it produces estimations that are closest to the true system state among all estimators. The corresponding estimator for the utilization is,

$$\begin{aligned}\hat{l}(k+1|k) &= \hat{l}(k|k) + \delta_l(k) \\ \hat{l}(k|k) &= \hat{l}(k|k-1) + K_u(u(k) - \hat{l}(k|k-1)) \\ \hat{u}(k|k) &= \hat{u}(k) = \hat{l}(k|k)\end{aligned}\quad (12)$$

where,

$$\begin{aligned}K_u &= \frac{H_u}{H_u + R_{\delta_{wu}}}\quad (13) \\ H_u &= \frac{R_{\delta_{wl}}}{2} + \sqrt{\frac{R_{\delta_{wl}}^2}{4} + R_{\delta_{wl}} R_{\delta_{wu}}}.\end{aligned}$$

Figure 4 plots K_m as a function of $R_{\delta_{wl}}$ and $R_{\delta_{wm}}$ according to (10) and (11), where g_M is one. There are some interesting issues to consider here. We notice that K_m decreases as the measurement disturbance $R_{\delta_{wm}}(k)$ increases, meaning the measurement values have less impact on the estimation. We recall that the system disturbance represents variations in load and model inaccuracies, 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, as $R_{\delta_{wl}}(k)$ increases K_m should increase as is shown in Figure 4.

Above we have derived a state estimator that given the magnitude of the measurement and system disturbances produces optimal estimations of the controlled system state.

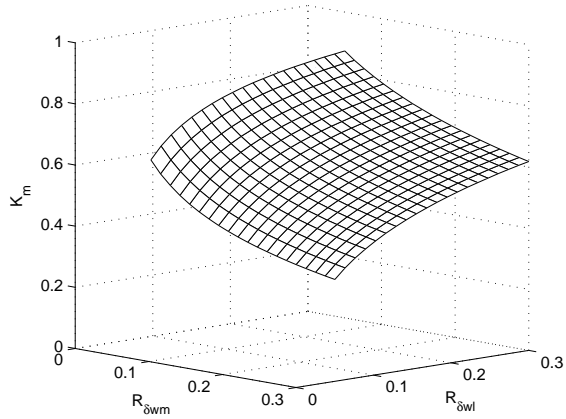


Figure 4. The optimal choice of K_m given $R_{\delta wl}$ and $R_{\delta wm}$.

4.4. Quantifying the Measurement Disturbance Variance

In this section we provide the theory for computing $R_{\delta wm}$ and $R_{\delta wu}$. As in the previous section we only show the derivation for $R_{\delta wm}$ as computing $R_{\delta wu}$ is analogous. Consider the model given by (4). Using (7), we notice that,

$$E\{m(k)\} = E\{g_M l(k)\} = g_M E\{l(k)\} = g_M \bar{l} = \bar{m},$$

i.e., the average deadline miss ratio is equal to the average of $g_M l(k)$, which we denote with \bar{m} . To distinguish between the measurement and the system disturbances we consider the variations in $m(k)$ to originate from $\delta_{wm}(k)$ only, i.e., we consider $g_M l(k)$ to be constant and equal to \bar{m} . Under this assumption the following theorem shows how $R_{\delta wm}$ is computed.

Theorem 2 Let $n_T \geq 1$ denote the number terminated tasks, n_M denote the number of tasks that have missed their deadlines, and \bar{m} denote the average deadline miss ratio. Then the variance in $\delta_{wm}(k)$ is,

$$R_{\delta wm} = \sum_{n_M=0}^{n_T} \left(\frac{n_M}{n_T}\right)^2 \frac{n_T!}{n_M!(n_T - n_M)!} \bar{m}^{n_M} (1 - \bar{m})^{n_T - n_M} - \bar{m}^2. \quad (14)$$

Proof: Utilizing the assumption $\bar{m} = g_M l(k)$, we get from (4) and (8) that,

$$R_{\delta wm} = E\{(m(k) - \bar{m})^2\} = E\{m^2(k)\} - \bar{m}^2. \quad (15)$$

By definition,

$$E\{m^2(k)\} = \sum_{\text{all } i} m_i^2(k) P_m(n_M(k), n_T(k)) \quad (16)$$

where $m_i(k)$ denotes the possible values that $m(k)$ can take and P_m is the discrete probability density function giving the probability of $m_i(k)$ assuming the value $\frac{n_M(k)}{n_T(k)}$. For example, if $n_T(k) = 1$, then $m_1(k) = 0$ and $m_2(k) = 1$, and similarly if $n_T(k) = 2$, then $m_1(k) = 0$, $m_2(k) = 0.5$, and $m_3(k) = 1$. In general, $m_i(k)$ can assume the values,

$$m_i(k) = \frac{n_M(k)}{n_T(k)}, 0 \leq n_M(k) \leq n_T(k). \quad (17)$$

To compute P_m , consider a set of tasks $\{\tau_1, \dots, \tau_i, \dots, \tau_{n_T}\}$ with the deadline miss indicators $\{d_1, \dots, d_i, \dots, d_{n_T}\}$, where $d_i = 1$ means that τ_i has missed its deadline and $d_i = 0$ means that τ_i did not miss its deadline. The probability of $d_i = 1$ is \bar{m} , whereas the probability of $d_i = 0$ is $1 - \bar{m}$. The probability of n_M out of n_T tasks missing their deadlines is given by the binomial distribution and, hence, the probability of m_i assuming the value $\frac{n_M}{n_T}$ is,

$$P_m(n_M, n_T) = \frac{n_T!}{n_M!(n_T - n_M)!} \bar{m}^{n_M} (1 - \bar{m})^{n_T - n_M}. \quad (18)$$

Substituting m_i and P_m in (16) by (17) and (18), gives,

$$E\{m^2(k)\} = \sum_{n_M=0}^{n_T} \left(\frac{n_M}{n_T}\right)^2 \frac{n_T!}{n_M!(n_T - n_M)!} \bar{m}^{n_M} (1 - \bar{m})^{n_T - n_M} \quad (19)$$

and by inserting (19) in (15) we finally obtain,

$$R_{\delta wm} = \sum_{n_M=0}^{n_T} \left(\frac{n_M}{n_T}\right)^2 \frac{n_T!}{n_M!(n_T - n_M)!} \bar{m}^{n_M} (1 - \bar{m})^{n_T - n_M} - \bar{m}^2. \quad \square$$

The derivation of $R_{\delta wu}$ is analogous to the derivation of $R_{\delta wm}$, hence,

$$R_{\delta wu} = \sum_{t_B=0}^{t_T} \left(\frac{t_B}{t_T}\right)^2 \frac{t_T!}{t_B!(t_T - t_B)!} \bar{u}^{t_B} (1 - \bar{u})^{t_T - t_B} - \bar{u}^2, \quad (20)$$

where $\bar{u} = E\{l(k)\}$.

Figure 5 shows how $R_{\delta wm}$ varies as a function of \bar{m} and n_T . As expected, $R_{\delta wm}$ decreases as n_T increases, meaning that the measurement disturbance originating from the averaging operation decreases in intensity. For a certain n_T , the intensity of the measurement disturbance peaks when the average deadline miss ratio \bar{m} is 0.5. The intensity $R_{\delta wm}$ is zero when \bar{m} is zero or one, which is expected as we have no variation in $m(k)$ at zero or one deadline miss ratio.

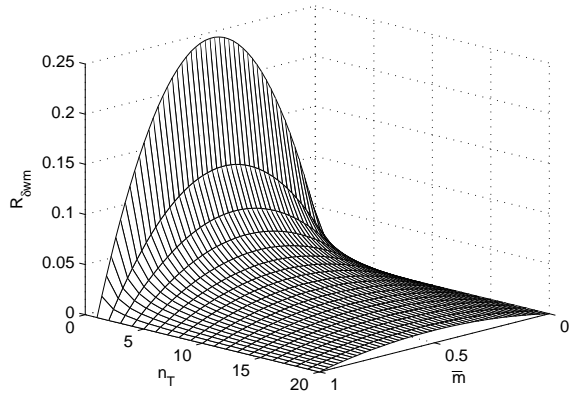


Figure 5. $R_{\delta_{wm}}$ as a function of \bar{m} and n_T .

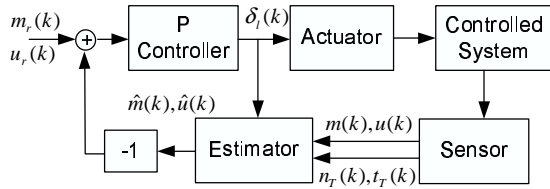


Figure 6. The Feedback Structure for Measurement Disturbance Suppression.

4.5. 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 estimator, as shown in Figure 6. Note, we either use $m(k)$ and $n_T(k)$, or $u(k)$ and $t_T(k)$ depending on the choice of the controlled variable. P-control [3,4] is employed, where $\delta_l(k) = K_P(m_r(k) - \hat{m}(k))$ when deadline miss ratio is the controlled variable, whereas, $\delta_l(k) = K_P(u_r(k) - \hat{u}(k))$ when utilization is the controlled variable. The estimators (6) and (12) are used to estimate the deadline miss ratio and the utilization. According to the separation principle [3,4], the design of the controller and the estimator is disjoint, meaning that the tuning of one does not affect the other one. The controller is designed using profiling data and a tuning method, e.g. [6], and the estimator is designed using the profiling data and (6). Hence, the separation principle significantly reduces the design complexity.

The variance of the system disturbance $R_{\delta_{wl}}$ is measured by profiling the controlled system during open-loop where we have no feedback control. $R_{\delta_{wl}}$ is set to the average of

the squared differences between the admitted load at each sampling instance and the average load. During run-time, $m(k)$ and $u(k)$ are formed by the sensor and returned to the estimator along with $n_T(k)$ and $t_T(k)$ at time kT . $R_{\delta_{wm}}(k)$ and $R_{\delta_{wu}}(k)$ are then computed using (14) and (20). Having obtained $R_{\delta_{wm}}(k)$ and $R_{\delta_{wu}}(k)$, the estimator feedback gains are computed according to (10) and (13). Note that updating the feedback gains requires a time-variant estimator [3,4]. However, in this work we approximate the time-variant estimator with a time-invariant estimator as an initial approach. In our future work we extend the estimator to a time-variant estimator.

Once, the feedback gains are updated, the estimators (6) and (12) are used to compute $\hat{m}(k)$ and $\hat{u}(k)$. The P-controller then computes $\delta_l(k)$ using the estimates $\hat{m}(k)$ and $\hat{u}(k)$. The effect of using an estimator is the following. As $n_T(k)$ or $t_T(k)$ increases, then we should trust $m(k)$ and $u(k)$ more as the deadline miss ratio and utilization are based on a larger data set. An increase in $n_T(k)$ or $t_T(k)$ corresponds to a decrease in $R_{\delta_{wm}}(k)$ and $R_{\delta_{wu}}(k)$ which in turn results in an increase in K_m and K_u (see Figure 4 and 5). An increase in K_m and K_u corresponds to the estimators relying more on the measured data rather than predictions. Similarly, as $n_T(k)$ or $t_T(k)$ decrease, estimations are based more on predictions and less on the measurements. The result of this is an efficient way of suppressing disturbances present in $m(k)$ and $u(k)$ that arise from the averaging operation. This means that we present a more accurate representation of the system state to the P-controller which is able to enhance the performance management in terms of lowering $(m_r^2(k) - m(k))^2$ and $(u_r^2(k) - u(k))^2$. The latter implies a more reliable performance and system adaptability as the actual system performance is closer to the desired system performance.

5. Performance Evaluation

The main objective of the performance evaluation is to determine the suitability of the proposed approach, namely, using an estimator to suppress the measurement disturbance. In this regard we perform an experiment where the performance of a feedback loop with an estimator is compared to the performance of a traditional feedback loop without an estimator. The following describes the simulator used to carry out the performance evaluation, followed by definition of performance metrics and the result of the experiments.

5.1. Simulator Setup

The simulated workload consists of aperiodic tasks, as an aperiodic task set implies an increased unpredictability in workload, hence, a greater challenge on the control of

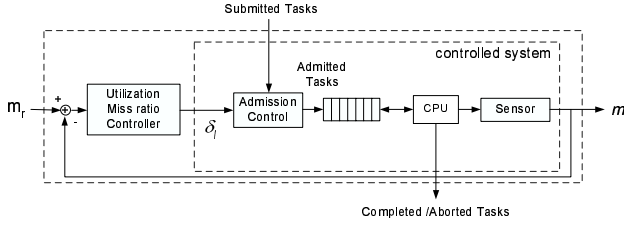


Figure 7. The Simulated System Architecture.

performance. The general outline of the feedback control scheduling architecture is given in Figure 7. We assume a workload model where each task has two QoS levels, i.e. $N = 2$, $q_i[s_1] = 0$, $q_i[s_2] = 1$, $x_{E,i}[s_1] = x_{A,i}[s_1] = 0$, $x_{E,i}[s_2] > 0$, and $x_{A,i}[s_2] > 0$ (see Section 3), 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 $m(k)$. The goal is to minimize $(m_r - m(k))^2$. Based on $\delta_l(k)$, the admission controller enforces the workload adjustment. A task is admitted if its load added to the admitted load is less than $l(k)$. The workload model of the tasks is described as follows. The estimated execution time $x_{E,i}[s_2]$ of a task τ_i is uniformly distributed according to $U : (50ms, 300ms)$. Upon generation of a task an actual execution time given by the normal distribution $N : (x_{E,i}[s_2], \sqrt{x_{E,i}[s_2]})$ is associated with τ_i . The deadline is set to $a_i + x_{E,i}[s_2] \times \text{slack factor}$, where a_i denotes the arrival time of τ_i . The slack factor is uniformly distributed according to $U : (10, 30)$. The inter-arrival time is exponentially distributed with the mean inter-arrival time set to $x_{E,i}[s_2] \times \text{slack factor}$.

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

5.1.1. Performance Metrics In Section 2 we argued that the goal of feedback control is to minimize the difference between the actual system performance and the desired system performance. Therefore, we distinguish the performance of controllers by how well they force $m(k)$ to follow m_r , despite presence of system and measurement disturbances. In our simulations we evaluate the controllers with respect to $J_a = \frac{1}{S} \sum_{k=1}^S |m_r - m(k)|$ and $J_s = \frac{1}{S} \sum_{k=1}^S (m_r - m(k))^2$, where S is the number of samples taken. By J_s and J_a we can establish how well the controllers are able to keep $m(k)$ near m_r . The lower J_s and J_a are, the better a controller is able to keep $m(k)$ near m_r , and also the faster $m(k)$ converges toward m_r .

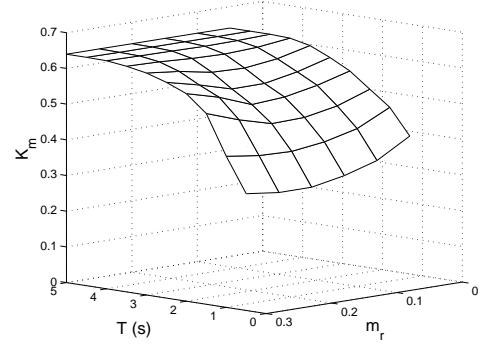


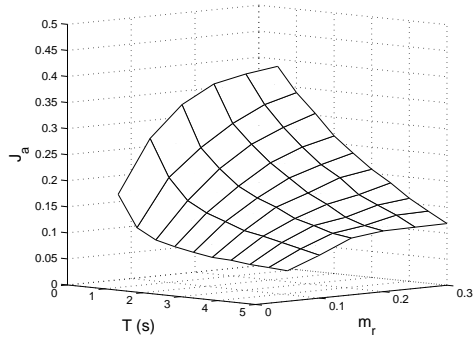
Figure 8. Measured K_m

5.2. Evaluation of Controller Performance

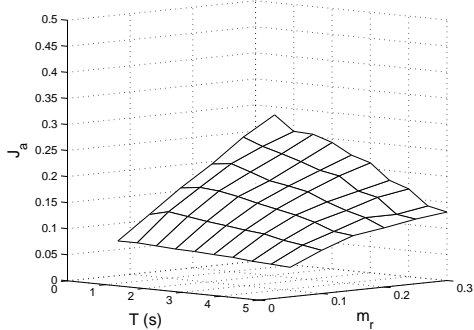
We know that the control performance is directly related to $R_{\delta_{wm}}$, which in turn depends on n_T . Therefore it is interesting to observe the performance of a feedback loop as n_T and, hence, T varies. We show that using the deadline miss ratio estimator (6) significantly reduces J_a and J_s 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, ..., 0.30, and vary T according to 0.50, 1.00, ..., 5.00s. The results are shown in Figures 8-11.

Number of terminated tasks n_T increases as T increases and, hence, for larger T we expect a lower magnitude of the measurement disturbance. Further, n_T also increases as the deadline miss ratio reference m_r increases. An increase in T and, consequently, an increase in n_T 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 . This is shown in Figure 8, where K_m increases as T increases.

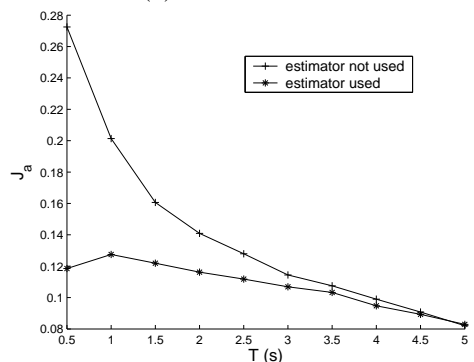
The measured J_a and J_s when an estimator is used (figure b), respectively not used (figure a), are presented in Figures 9 and 10. The case when $m_r = 0.10$ is given in Figure 9(c). Remember that as J_a and J_s decrease, the actual system performance is closer to the desired system performance. The difference in J_a decreases as T increases due to the decreasing magnitude of the measurement disturbance. For low sampling periods, a significant decrease in J_a is achieved when an estimator is used as compared to the case when an estimator is not used. Considering Figure 9(c) at $T = 0.5s$, J_a is 0.12 when an estimator is used, compared to 0.28 when an estimator is not used, i.e., we have a difference of about 0.16. Similarly, the difference in J_a is 0.07 for the case $T = 1s$, showing that even at the sampling period of one second we have a great difference in J_a . Hence, the actual deadline miss ratio is closer to its reference when



(a) Estimator not used



(b) Estimator used

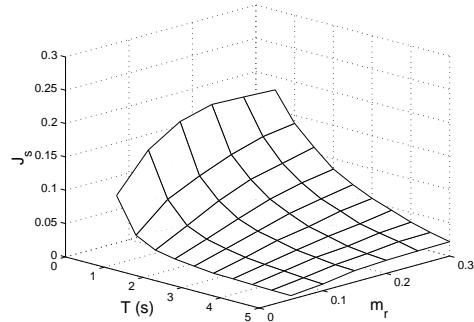


(c) $m_r = 0.10$

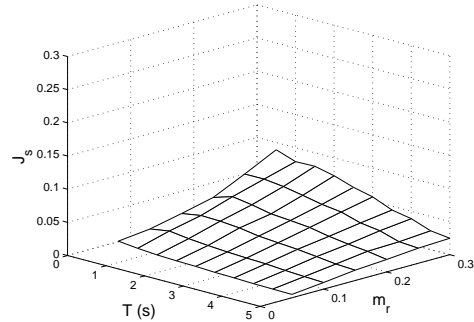
Figure 9. Measured J_a

an estimator is used.

We also study $m(k)$ in the time domain to obtain a better understanding of how a certain J_a corresponds to variations in $m(k)$. Figure 11 shows the deadline miss ratio in the time domain for the experiment corresponding to $T = 1s$ and $m_r = 0.10$. As we can see $m(k)$ oscillates heavily around the reference in the absence of an estimator, as shown in Figure 11(a). However, the deviations are significantly reduced when using an estimator, as shown in Figure 11(b). In the absence of the estimator, large deviations in $m(k)$ due to the measurement disturbance are not filtered. As a consequence, the controller tries to compensate for the changes in $m(k)$ by changing the requested workload, which results in



(a) Estimator not used



(b) Estimator used

Figure 10. Measured J_s , estimator off

an over compensation and, hence, $m(k)$ deviates even more from m_r . However, an estimator is able to suppress variations in $m(k)$ 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.

In summary we have shown that a lower sampling period increases the disturbance in the measurements. The performance is improved when using estimators for suppressing the measurement disturbance. We have observed that $m(k)$ is closer to the reference m_r , implying that we achieve improved performance reliability and adaptation. Hence, by using an estimator, the actual system performance gets closer to the desired system performance, compared to the case when the estimator is not used.

6. Related work

Lu et al. introduced a feedback control scheduling framework for controlling utilization and deadline miss ratio [6]. In our previous work we controlled the performance of real-time databases using feedback control [2]. Parekh et al. use feedback control scheduling to control the length of a queue of remote procedure calls (RPCs) arriving at a server [9]. Abdelzaher et al. presented control algorithms for managing service delay and queue length of requests arriving at

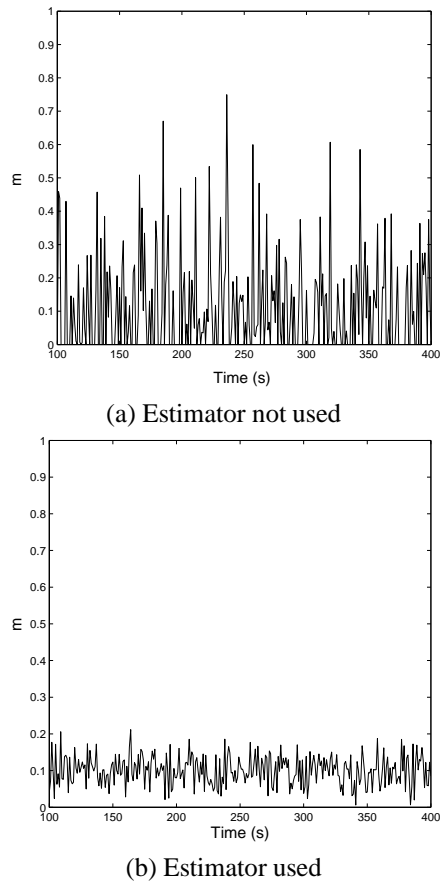


Figure 11. $m(k)$ in the time domain.

web servers [1]. None of the approaches above have considered the effects of the sampling period on the measured variable. Lu et al. [7] introduced an architecture for differentiated caching services. The desired relations between the hit ratios of different content classes are enforced using per-class feedback control loops. The authors note that the variance of the hit ratio for a class may be large for small sampling periods. They use a low pass filter to smoothen out large deviations in hit ratio. However, they do not model and quantify the measurement disturbance and a method for tuning the low pass filter is not given. In contrast, we have modeled and quantified the measurement disturbance and provided a method for tuning an estimator for the controlled variable.

7. Conclusions

Using feedback control has shown to be effective for a large class of real-time systems with unpredictable workload characteristics. The measured variables typically used to describe the performance of com-

puter 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 estimator that produces estimations of the controlled variable. Experimental results show that this approach results in improved control performance 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 this work we considered the metrics of deadline miss ratio and utilization. In our future work, we will generalize our approach by including the quality of the results that the tasks produce. Also we will consider time-variant estimators, where the estimation parameters are varied according to changes in the controlled system.

References

- [1] T. F. Abdelzaher, J. A. Stankovic, C. Lu, R. Zhang, and Y. Lu. Feedback performance control in software services. *IEEE Control Systems Magazine*, 23(3):74–90, June 2003.
- [2] M. Amirijoo, J. Hansson, S. H. Son, and S. Gunnarsson. Robust quality management for differentiated imprecise data services. In *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS)*, 2004.
- [3] G. F. Franklin, J. D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, third edition, 1998.
- [4] T. Glad and L. Ljung. *Control Theory - Multivariable and Nonlinear Methods*. Taylor and Francis, 2000.
- [5] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. Wiley-IEEE Press, 2004.
- [6] C. Lu, J. A. Stankovic, G. Tao, and S. H. Son. Feedback control real-time scheduling: Framework, modeling and algorithms. *Journal of Real-time Systems*, 23(1/2), July/September 2002.
- [7] Y. Lu, A. Saxena, and T. F. Abdelzaher. Differentiated caching services; a control-theoretical approach. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2001.
- [8] A. V. Oppenheim and A. S. Willsky. *Signals and Systems*. Prentice-Hall, second edition, 1996.
- [9] S. Parekh, N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, and J. Bigus. Using control theory to achieve service level objectives in performance management. *Journal of Real-time Systems*, 23(1/2), July/September 2002.