A Feedback Scheduler for Real-Time Controller Tasks

Eker, Johan; Hagander, Per; Årzén, Karl-Erik

Published in:
Control Engineering Practice

DOI:
10.1016/S0967-0661(00)00086-1

2000

A Feedback Scheduler for Real-Time Control Tasks
Johan Eker, Per Hagander, Karl-Erik Årzén,
Control Engineering Practice, Volume 8, Issue 12, pages 1369-1378
Pergamon:
Alternative location: [http://dx.doi.org/10.1016/S0967-0661(00)00086-1]

In IFAC Control Engineering Practice 2000

A FEEDBACK SCHEDULER FOR REAL-TIME CONTROLLER TASKS

Johan Eker Per Hagander Karl-Erik Årzén

Department of Automatic Control
Lund Institute of Technology
Box 118, S-221 00 Lund
Sweden
Fax: +46-[0]46 13 81 18
johan.eker|karlerik.arzen|per.hagander@control.lth.se

Keywords Real-time; Feedback scheduling; Linear quadratic control; Optimization.

Abstract

The problem studied in this paper is how to distribute computing resources over a set of real-time control loops in order to optimize the total control performance. Two subproblems are investigated: how the control performance depends on the sampling interval, and how a recursive resource allocation optimization routine can be designed. Linear quadratic cost functions are used as performance indicators. Expressions for calculating their dependence on the sampling interval are given. An optimization routine, called a feedback scheduler, that uses these expressions is designed.

1. INTRODUCTION

Control design and task scheduling are in most cases today treated as two separate issues. The control community generally assumes that the real-time platform used to implement the control system can provide deterministic, fixed sampling periods as needed. The real-time scheduling community, similarly, assumes that all control algorithms can be modeled as periodic tasks with constant periods, hard deadlines, and known worst case execution times. This simple model has made it possible for the control community to focus on its own problem domain without worrying how scheduling is being done, and it has released the scheduling community from the need to understand how scheduling delays impact the stability and performance of the plant under control. From a historical perspective, the separated development of control and scheduling theories for computer based control systems has produced many useful results and served its useful purpose.

Upon closer inspection it is, however, quite clear that neither of the above assumptions need necessarily be true. Many of the computing platforms that are commonly used to implement control systems, are not able to give any deterministic guarantees. This is especially the case when commercial off-the-shelf operating systems, e.g. Windows NT or Linux, are used. These systems are, typically, designed to achieve good average performance rather than high worst-case performance. Many control algorithms are not periodic, e.g. internal combustion engine control, or they may switch between a number of different fixed sampling periods. Control algorithm deadlines are not always hard. On the contrary, many controllers are quite robust towards variations in sampling period and response time. It is in many cases also possible to compensate for the variations on-line by, e.g., recomputing the controller parameters, see Nilsson (1998). It is also possible to consider control systems that are able

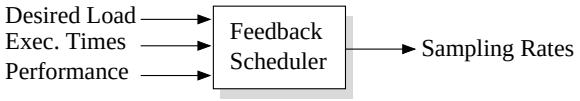

Fig. 1 New sampling rates are calculated based on the desired CPU load, the execution times of the controllers, and the performance of the controllers.

to do a tradeoff between the available computation time, i.e., how long time the controller may spend calculating the new control signal, and the control loop performance.

For more demanding applications, requiring higher degrees of flexibility, and for situations where computing resources are limited, it is therefore desirable to study more integrated approaches to scheduling of control algorithms. The approach taken in this paper is based on dynamic feedback from the scheduler to the controllers, and from the controllers to the scheduler. The idea of feedback has been used informally for a long time in scheduling algorithms for applications where the dynamics of the computation workload cannot be characterized accurately. The VMS operating system, for example, uses multi-level feedback queues to improve system throughput, and Internet protocols use feedback to help solve the congestion problems. The idea of feedback has also been exploited in multi-media scheduling R&D, recently, under the title of quality of service (QoS). An application demonstrating QoS reasoning in a real-time control system is found in Abdelzaher, Atkins and Shin (1997).

In this paper the goal is to schedule a set of real-time control loops in order to maximize their total performance. The number of control loops and their execution times may change over time and hence the task schedule must be adjusted to maintain optimality and schedulability. Since the optimizer works in closed loop with the control tasks, it is referred to as a feedback scheduler. The feedback scheduler adjusts the control loop frequencies to optimize the control performance while maintaining schedulability. The input signals are the performance levels of each loop, their current execution times, and the desired workload level, see Fig. 1. Two problems are studied when designing such a feedback scheduler. The first is to find a suitable performance index and calculate how it depends on the sampling frequency. The second problem is to design an optimization routine. It is assumed that measurements or estimates of the execution times and system workload are available from the real-time kernel at run-time.

As a performance index, a linear quadratic (LQ) formulation is used. The controllers are state feedback algorithms designed to minimize this quadratic cost function. The performance index is calculated as a function of the sampling interval. Given this control performance indicator an optimization routine is designed. The optimization routine aims to find the control performance optimum, given a desired CPU utilization level (workload) and the execution times for the tasks. The global cost function that should be minimized consists of the sum of the cost function of each control loop. The minimization is performed subject to a schedulability constraint. The minimization of the global cost function constitutes a nonlinear programming problem. The solution of this problem is facilitated by the knowledge of the first and second derivatives of the cost functions with respect to the sampling rate. Expressions for the derivatives are calculated and used in the optimization.

Using a feedback scheduling strategy it is now possible to design real-time control systems that are more robust against uncertainties in execution times and workload. When a control task performs a mode change and this leads to a change in its execution time the feedback scheduler adjusts the sampling frequencies so that the system remains schedulable.

Deadlines may be missed if a change in execution time causes the system to overload. In such a situation, the feedback scheduler adjusts the sampling rates to regain schedulability, but deadlines may, however, still be missed. If instead the feedback scheduler is aware of a forthcoming mode change, it could avoid an overload by changing the sampling rates in advance. This can be dealt with by establishing a communication channel between the feedback scheduler and the control tasks. The control tasks notify the feedback scheduler, by sending a request, prior to a mode change and may not proceed with the mode change until given permission to do so. The module that handles this is called the *admission controller*. It is also responsible for handling the arrival of new tasks and the termination of old tasks. Fig. 2 shows a block diagram of the feedback scheduler with the admission controller.

1.1 *Outline*

An overview of related work is given in Section 2. Section 3 defines the problem, and the LQ-based cost function calculation are presented in Section 4. Section 5 describes the design of the feedback scheduler.

2. INTEGRATED CONTROL AND SCHEDULING

In order to achieve on-line interaction between control algorithms and the scheduler a number of issues must be considered. Control design methods must take schedulability constraints into account. It must be possible to dynamically adjust task parameters, e.g., task periods, in order to compensate for changes in workload. It can also be advantageous to view the task parameters adjustment strategy in the scheduler as a controller. In this section an overview is given of the work that has been performed in these areas. A more detailed survey on control and scheduling can be found in Årzén, Bernhardsson, Eker, Cervin, Nilsson, Persson and Sha (1999).

2.1 *Control and scheduling co-design*

A prerequisite for an on-line integration of control and scheduling theory is the ability to make an integrated off-line design of control algorithms and scheduling algorithms. Such a design process should ideally allow an incorporation of the availability of computing resources into the control design by utilizing the results of scheduling theory. This is an area where relatively little work has been performed so far. In Seto, Lehoczky, Sha and Shin (1996) an algorithm was proposed that translates a system performance index into task sampling periods, considering schedulability among tasks running with pre-emptive priority scheduling. The sampling periods were considered as variables, and the algorithm determined their values so that the overall performance was optimized subject to the schedulability constraints. Both fixed priority rate-monotonic and dynamic priority, Earliest Deadline First (EDF) scheduling were considered. The loop cost function was heuristically approximated by an exponential function. The approach was further extended in Seto, Lehoczky and Sha (1998).

A heuristic approach to optimization of sampling period and input-output latency subject to performance specifications and schedulability constraints was also presented in Ryu, Hong and Saksena (1997) and Ryu and Hong (1998). The control performance was specified in terms of steady state error, overshoot, rise time, and settling time. These performance parameters were expressed as functions of the sampling period and the input-output latency. An iterative algorithm was proposed for the optimization of these parameters subject to schedulability constraints.

2.2 *Task attribute adjustments*

A key issue in any system allowing dynamic feedback between the control algorithms and the on-line scheduler is the ability to dynamically adjust task parameters. Examples of task parameters that could be modified are period and deadline.

In Shin and Meissner (1999) the approach in Seto, Lehoczky, Sha and Shin (1996) is extended, making on-line use of the proposed off-line method for processor utilization allocation. The approach allows task period changes in multi-processor systems. A performance index for the control tasks is used, weighting the importance of the task to the overall system, to determine the value to the system of running a given task at a given period.

In Buttazzo, Lipari and Abeni (1998) an elastic task model for periodic tasks is presented. A task may change its period within certain bounds. When this happens, the periods of the other tasks are adjusted so that the overall system is kept schedulable. An analogy with a linear spring is used, where the utilization of a task is viewed as the length of a spring that has a given spring coefficient and length constraints. The MART (Modification and Adjustment of Real-time Tasks) scheduling algorithm (Kosugi, Takashio and Tokoro, 1994; Kosugi, Mitsuzawa and Tokoro, 1996; Kosugi and Moriai, 1997) also supports task period adjustments. MART has been extended to also handle task execution time adjustments. The system handles changes both in the number of periodic tasks and in the task timing attributes. Before accepting a change request the system analyzes the schedulability of all tasks. If needed, it adjusts the period and/or execution time of the tasks to keep them schedulable with the rate monotonic algorithm.

2.3 *Feedback scheduling*

An on-line scheduler that dynamically adjusts task attributes can be viewed as a controller. Important issues that must be decided are what the right control signals, measurement signals, and set-points are, what the correct control structure should be, and which process model that may be used.

So far, very little has been done in the area of real-time feedback scheduling. A notable exception is presented in Stankovic, Lu, Son and Tao (1999), where a PID controller is used as an on-line scheduler. The measurement signal (the controlled variable) is the deadline miss ratio for the tasks, and the control signal is the requested CPU utilization. Changes in the

requested CPU are effectuated by two mechanisms (actuators). An admission controller is used to control the flow of workload into the system, and a service level controller is used to adjust the workload inside the system. The latter is done by changing between different versions of the tasks with different execution time demands. A simple liquid tank model is used as an approximation of the scheduling system.

Using a controller approach of the above kind, it is important to be able to measure the appropriate signals on-line, e.g., to be able to measure the deadline miss ratio, the CPU utilization, or the task execution times.

An event feedback scheduler is proposed in Zhao and Zheng (1999). Several control loops share a CPU, and only one controller may be active at each time instant. The strategy used is to activate the controller connected to the plant with the largest error. Similar ideas are found in Årzén (1999), which suggests an event-based PID controller that only executes if the control error is larger than a specified threshold value.

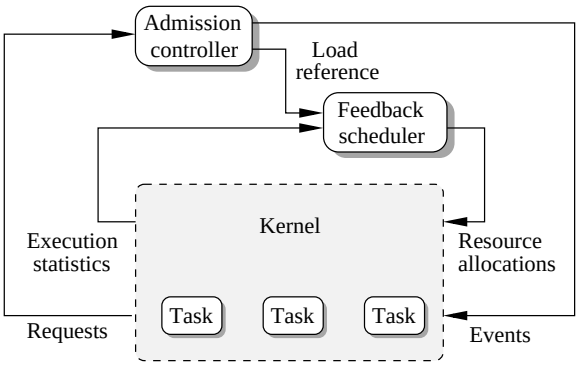

Fig. 2 The real-time kernel is connected in a feedback loop with the feedback scheduler and the admission controller.

3. PROBLEM STATEMENT

Consider a control system where several control loops share the same CPU. Let the execution times of all tasks and the system workload be available from the real-time kernel at all times. Furthermore, associate each controller with a function, which indicates its performance. The execution times, the number of tasks, and the desired workload may vary over time. The aim of the feedback scheduler is to optimize the total control performance, while keeping the workload at the desired level.

Two separate modules are used, see Fig. 2. The feedback scheduler adjusts the sampling intervals to control the workload. The workload reference is calculated by the admission controller which contains high-level logic for coordinating tasks.

The tasks and the feedback scheduler communicate using requests and events. A task can send a request for more computing resources and the scheduler may grant this by replying with the appropriate event. The kernel manages the tasks at the low level, i.e. task dispatching, etc. The feedback scheduler gets execution statistics, such as the actual execution times of the tasks, and the system workload from the kernel. This information is then used to calculate how CPU resources should be allocated in an optimal fashion.

Let the number of tasks on the system be n, and let each task execute with a sampling interval h_i (task period) and have the execution time C_i. Each task is associated with a cost function $J_i(h_i)$, which gives the control cost as a function of sampling interval h_i. The following optimization problem may now be formulated

$$\text{minimize} \quad J = \sum_{i=1}^{n} J_i(h_i), \tag{1}$$

subject to the constraint $\sum_{i=1}^{n} C_i/h_i \leq U_{ref}$, where U_{ref} is the desired utilization level.

4. COST FUNCTIONS

In this section the LQ cost function and its derivatives with respect to the sampling interval are calculated. These functions will be used in the feedback scheduler algorithm, as described in Section 5.

Let the system be given by the linear stochastic differential equation

$$d\mathbf{x} = \mathbf{A}\mathbf{x}dt + \mathbf{B}udt + d\mathbf{v}_\mathbf{c}. \tag{2}$$

where A and B are matrices and v_c is a Wiener process with the incremental covariance $R_{1c}dt$. A continous stochastic system description is used as the goal is to study the cost of sampling the system at different rates. The sampled system is given by

$$\mathbf{x}(kh+h) = \mathbf{\Phi}\mathbf{x}(kh) + \mathbf{\Gamma}u(kh) + \mathbf{v}(kh),$$

where $\mathbf{\Phi} = e^{\mathbf{A}h}$, $\mathbf{\Gamma} = \int_0^h e^{\mathbf{A}s}ds\mathbf{B}$, and $\mathbf{v}(kh)$ is white noise with the following property:

$$\mathrm{E}\mathbf{v}(kh)\mathbf{v}^{\mathrm{T}}(kh) = \mathbf{R}_1(h) = \int_0^h e^{\mathbf{A}\tau}\mathbf{R}_{1c}e^{\mathbf{A}^{\mathrm{T}}\tau}d\tau. \tag{3}$$

The cost function for the controller is chosen as

$$J(h) = \frac{1}{h}\mathrm{E}\int_0^h \begin{bmatrix}\mathbf{x}^{\mathrm{T}}(t) & \mathbf{u}^{\mathrm{T}}(t)\end{bmatrix}\mathbf{Q}_c\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{bmatrix}dt,$$

where

$$\mathbf{Q}_c = \begin{bmatrix}\mathbf{Q}_{1c} & \mathbf{Q}_{2c}\\ \mathbf{Q}_{2c}^{\mathrm{T}} & \mathbf{Q}_{3c}\end{bmatrix}.$$

The cost per time unit for the discrete LQ-controller is given as:

$$\begin{aligned} J &= E(\frac{1}{Nh}\int_0^{Nh}\begin{bmatrix}\mathbf{x}^{\mathrm{T}}(t) & \mathbf{u}^{\mathrm{T}}(t)\end{bmatrix}\mathbf{Q}_c\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{bmatrix})\\ &= E(\frac{1}{Nh}\sum_{k=0}^{N}\int_{kh}^{kh+h}\begin{bmatrix}\mathbf{x}^{\mathrm{T}}(t) & \mathbf{u}^{\mathrm{T}}(t)\end{bmatrix}\mathbf{Q}_c\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{bmatrix}dt)\end{aligned}$$

When time goes to infinity, $lim_{N\to\infty}$,the influence from the initial conditions decreases, and the cost may be written as

$$J = E(\frac{1}{h}\int_0^h \begin{bmatrix}\mathbf{x}(t) & \mathbf{u}(t)\end{bmatrix}\mathbf{Q}_c\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{u}(t)\end{bmatrix}dt)$$

This means that only the cost in stationarity is regarded, and that the cost is scaled by the time horizon, i.e. the sampling interval h. The controller $\mathbf{u} = -\mathbf{L}\mathbf{x}(t)$ that minimizes the cost is given by solving the stationary Riccati equation with respect to the matrices $\mathbf{S}$ and $\mathbf{L}$.

$$\begin{bmatrix}\mathbf{S}+\mathbf{L}^{\mathrm{T}}\mathbf{G}\mathbf{L} & \mathbf{L}^{\mathrm{T}}\mathbf{G}\\ \mathbf{G}\mathbf{L} & \mathbf{G}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}^{\mathrm{T}}\\ \mathbf{\Gamma}^{\mathrm{T}}\end{bmatrix}\mathbf{S}\begin{bmatrix}\mathbf{\Phi} & \mathbf{\Gamma}\end{bmatrix} + \mathbf{Q}_d \tag{4}$$

where $\mathbf{G} = \mathbf{\Gamma}^{\mathrm{T}}\mathbf{S}\mathbf{\Gamma} + \mathbf{Q}_{3d}$, and

$$\begin{aligned}\mathbf{Q}_d &= \begin{bmatrix}\mathbf{Q}_{1d} & \mathbf{Q}_{2d}\\ \mathbf{Q}_{2d}^{\mathrm{T}} & \mathbf{Q}_{3d}\end{bmatrix} = \int_0^h e^{\mathbf{\Sigma}^{\mathrm{T}}t}\mathbf{Q}_c e^{\mathbf{\Sigma}t}dt,\\ \mathbf{\Sigma} &= \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\end{aligned} \tag{5}$$

The minimal value of J is given by Åström and Wittenmark (1997) and Gustafsson and Hagander (1991) as:

$$\begin{aligned}\min J(h) &= \frac{1}{h}(\mathrm{tr}\mathbf{S}\mathbf{R}_1 + \bar{J}), \qquad \text{where}\\ \bar{J} &= \mathrm{tr}(\mathbf{Q}_{1c}\int_0^h \mathbf{R}_1(\tau)d\tau)\end{aligned}$$

In order to use the cost function in the optimization it is useful to know the derivatives with respect to the sampling period.

Theorem 1
The first derivative of J is given as

$$\begin{aligned}\frac{dJ}{dh} =& \frac{1}{h}(\mathrm{tr}\frac{d\mathbf{S}}{dh}\mathbf{R}_1 + \mathrm{tr}\mathbf{S}\frac{d\mathbf{R}_1}{dh} + \frac{d\bar{J}}{dh}) -\\ & \frac{1}{h^2}(tr\mathbf{S}\mathbf{R}_1 + \bar{J})\end{aligned}$$

where

$$\begin{aligned}\frac{d\mathbf{S}}{dh} =& (\mathbf{\Phi}-\mathbf{\Gamma}L)^{\mathrm{T}}\frac{d\mathbf{S}}{dh}(\mathbf{\Phi}-\mathbf{\Gamma}L) + W,\\ \mathbf{W} =& \begin{bmatrix}\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L}\\ -\mathbf{L}\end{bmatrix}^{\mathrm{T}}\mathbf{Q}_c\begin{bmatrix}\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L}\\ -\mathbf{L}\end{bmatrix} +\\ & \{(\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L})^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} - \mathbf{L}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\}\mathbf{S}(\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L}) +\\ & (\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L})^{\mathrm{T}}\mathbf{S}\{\mathbf{A}(\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{L}) - \mathbf{B}\mathbf{L}\},\end{aligned}$$

The matrix R_1 is calculated using Lemma 1 in the appendix and derivative of R_1 is given directly from Eq. 3.

$$\begin{aligned}\frac{d\mathbf{R}_1}{dh} =& e^{\mathbf{A}h}\mathbf{R}_{1c}e^{\mathbf{A}^{\mathrm{T}}h},\\ \frac{d\bar{J}}{dh} =& \mathrm{tr}\mathbf{Q}_{1c}\mathbf{R}_1\end{aligned}$$

□

See Appendix A for the proof. Expression for the second derivative of the cost functions is also given there.

Example 1
Consider the linearized equations for a pendulum:

$$\begin{aligned}d\mathbf{x} &= \begin{bmatrix}0 & 1\\ \alpha\omega_0^2 & -d\end{bmatrix}\mathbf{x}dt + \begin{bmatrix}0\\ \alpha b\end{bmatrix}udt + d\mathbf{v}_c\\ y &= \begin{bmatrix}1 & 0\end{bmatrix}\mathbf{x}, \qquad \mathbf{R}_{1c} = \begin{bmatrix}0 & 0\\ 0 & \omega_0^4\end{bmatrix}\end{aligned}$$

The natural frequency is ω_0, the damping $d = 2\zeta\omega_0$, with $\zeta = 0.2$, and $b = \omega_0/9.81$. If $\alpha = 1$, the equations describe the pendulum in the upright position, and if $\alpha = -1$ they describe the pendulum in the downward position. The incremental covariance of v_c is $\mathbf{R}_{1c}dt$, which corresponds to a disturbance on the control signal.

The cost functions J for the closed loop control of the inverted pendulum as a function of the sampling interval is shown in Fig. 3. The corresponding function for the stable pendulum is shown in Fig. 4. Fig. 4 clearly demonstrates that faster sampling not necessarily gives better control performance. Sampling the pendulum as slowly as this is however unrealistic. The rule of thumb from Åström and Wittenmark (1997) is to choose the sampling rate as $\omega_0 h \approx 0.2 - 0.6$. The time scale for Fig. 4 is out of bounds for

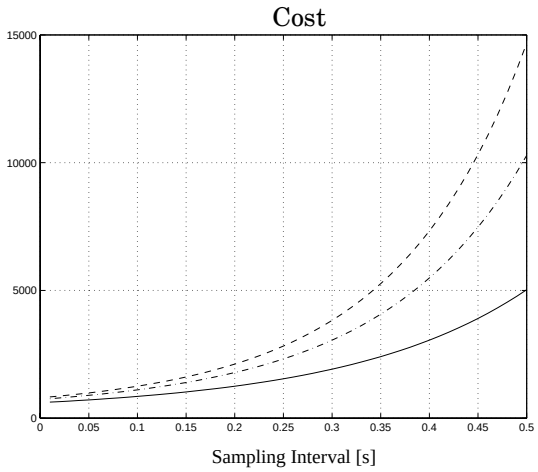

Fig. 3 The cost $J_i(h)$ as a function of the sampling interval for the inverted pendulum. The plot shows the graph for $\omega_0 = 3.14$(full), 3.77(dot-dashed), and 4.08(dashed).

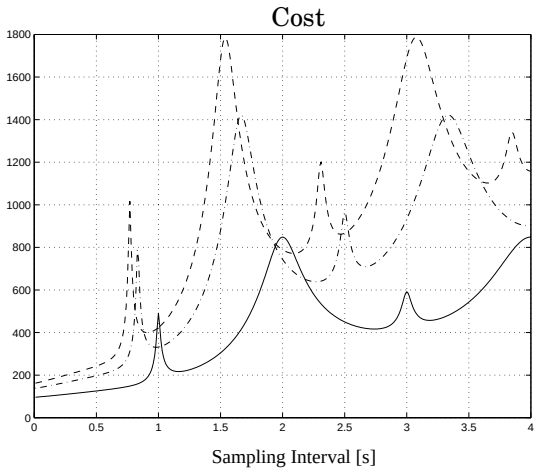

Fig. 4 The cost $J_i(h)$ as a function of the sampling interval for the pendulum. The plot shows the graph for $\omega_0 = 3.14$(full), 3.77(dot-dashed), and 4.08(dashed). The peaks are due to the resonance frequencies of the pendulum.

any practical use for the current application. It does however give an indication of what could happen in a system with a high frequency mechanical resonance. □

5. A FEEDBACK SCHEDULER

In this section a feedback scheduler, see Fig. 2, for a class of control systems with convex cost functions is proposed. Standard nonlinear programming results are used as a starting point for the feedback scheduler design. First, an algorithm, using the cost functions presented in the previous section is designed, and then an approximate, less computation-intense algorithm is presented. Simulation results for a system with three control loops are also given.

5.1 *Static Optimization*

For the class of systems for which the cost functions are convex, ordinary optimization theory, e.g. steepest decent search or constraint Newton, may be applied. The optimization criterion in Eq. (1) has nonlinear constraints and is first rewritten. By optimizing over the frequencies instead of the sampling intervals, the following optimization problem is given:

$$\begin{aligned} \min_f V(f) &= \sum_{i=1}^{n} J_i(1/f_i), \\ \text{subject to} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{f} \le U_{ref} \\ & \mathbf{f} = [f_1, \ldots, f_n]^{\mathrm{T}} \end{aligned}$$

The Kuhn-Tucker conditions, see for example (Fletcher, 1987), give that if $\bar{f} = [\bar{f}_1, \ldots, \bar{f}_n]^{\mathrm{T}}$ is an optimal solution then:

$$\begin{aligned} V_f(\bar{\mathbf{f}}) + \lambda \mathbf{c} &= 0 \\ \lambda[U_{ref} - \mathbf{c}^{\mathrm{T}}\bar{\mathbf{f}}] &= 0, \\ \lambda &\ge 0 \end{aligned} \tag{6}$$

where the column vector V_f is the gradient, $\mathbf{c} = [C_1, \ldots, C_n]^{\mathrm{T}}$, and λ is the Lagrange multiplier. Since $V_i(f_i) = J_i(1/f_i) = J_i(h_i)$, the derivative of V_i is $\frac{dV_i}{f_i} = -h^2 \frac{dJ_i}{h_i}$. This gives $V_f^{\mathrm{T}} = [\frac{dV_i}{f_1}, \ldots, \frac{dV_i}{f_n}] = [-h_1^2 \frac{dJ_i}{h_1}, \ldots, -h_n^{\mathrm{T}} \frac{dJ_n}{h_n}]$.

5.2 *Recursive Optimization*

Since changes in the computer load U_{ref} and the execution times C change the optimization problem, they need to be resolved at each step in time. In principle, one can repeat the solution to the static optimization problem at each step in time. However, instead of looking for a direct solution a recursive algorithm will be considered. Assume that there is a solution $f(k)$, at step k, which is optimal or close to optimal. Now a recursive algorithm to compute new values for f and λ will be constructed. Let the execution

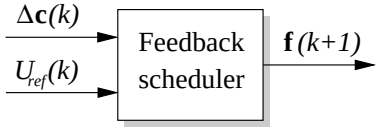

Fig. 5 The feedback scheduler calculates new sampling frequencies to accomodate for changes in the execution times C or in the desired workload level.

time vector $C(k)$ be time-varying. A control loop that adjusts the sampling frequencies so that the control performance cost is kept at the opti-

mum is to be designed, see Fig. 5. Let

$$\begin{aligned}\mathbf{f}(k+1) &= \mathbf{f}(k) + \Delta\mathbf{f}(k)\\ \lambda(k+1) &= \lambda(k) + \Delta\lambda(k)\\ \mathbf{c}(k+1) &= \mathbf{c}(k) + \Delta\mathbf{c}(k)\end{aligned}$$

Assume that $\lambda > 0$, i.e. that the CPU constraint is active. Linearization of Eq. (7) around the optimum gives

$$\begin{aligned}V_f + V_{ff}\Delta\mathbf{f} + (\lambda + \Delta\lambda)(\mathbf{c} + \Delta\mathbf{c}) &= 0\\ (\mathbf{c}^{\mathrm{T}} + \Delta\mathbf{c}^{\mathrm{T}})(\mathbf{f} + \Delta\mathbf{f}) &= U_{ref}\end{aligned}$$

If the quadratic delta terms are disregarded

$$\begin{bmatrix} V_{ff} & \mathbf{c} \\ \mathbf{c}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{f} \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -(V_f + \lambda\mathbf{c}) - \lambda\Delta\mathbf{c} \\ U_{ref} - \mathbf{c}^{\mathrm{T}}\mathbf{f} - \Delta\mathbf{c}^{\mathrm{T}}\mathbf{f} \end{bmatrix}$$

is obtained. Now the increments in f and λ are given as

$$\begin{bmatrix} \Delta\mathbf{f} \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} V_{ff} & \mathbf{c} \\ \mathbf{c}^{\mathrm{T}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -(V_f + \lambda\mathbf{c}) \\ U_{ref} - \mathbf{c}^{\mathrm{T}}\mathbf{f} - \Delta\mathbf{c}^{\mathrm{T}}\mathbf{f} \end{bmatrix} \tag{7}$$

A solution exists if V_{ff} is positive (or negative) definite and $\mathbf{c} \neq 0$. Note that the matrix V_{ff} is diagonal.

$$V_{ff} = diag(\begin{bmatrix} \frac{\partial^2 V}{\partial f_1^2} & \dots & \frac{\partial^2 V}{\partial f_n^2}, \end{bmatrix})$$

Thus, there is a solution to Eq. (8) if $\frac{\partial^2 V}{\partial f_i^2} > 0, \forall i$.

Remark It can be shown that this optimization routine corresponds to constrained Newton optimization, see for example Gill, Murray and Wright (1981). The algorithm has mostly been investigated through simulation and it seems very stable. In general a constrained Newton algorithm would not necessarily converge, but there may be reasonable assumptions that could guarantee the convergence seen in simulations.

Example 2

A single CPU-system is used to control three inverted pendulums of different lengths. Every pendulum is controlled by one LQ-controller, designed using the same weight matrix $\mathbf{Q}_c$ as in Example 1. Each control loop has an execution time C_i and a sampling frequency $=f_i$. Let the initial sampling frequencies be f_i^0. One of the tasks operates in two modes with very different execution times. The feedback scheduler adjusts the sampling frequencies for the tasks, so that the total control performance is optimized, given the current execution times and the workload reference. The workload reference is given by the admission controller.

Fig. 6 shows some plots from a simulation, where execution times, sampling frequencies and load are plotted as functions of the iteration steps. From the initial frequencies $\mathbf{f}^0 = [4, 4.5, 5]$, an optimal solution is found after 7-8 iterations. At step 15, Task #2 sends a request to the admission controller for more execution time, due to a forthcoming mode change. The admission controller must first lower the workload reference, before any task is allowed to increase its execution time, in order to avoid overload.

The new workload reference at step 15 is chosen so that there will enough room to allow task#2 to increase its execution time. The size of the CPU reserved for change in task#2 is based on the current sampling rates. In this example $C_1 = 0.04, C_2 = 0.05, C_3 = 0.07$ and the frequencies at step 14 are $f_1 = 5.84, f_2 = 6.50, f_3 = 6.31$. If task#2 doubles its execution time, the workload will increase with approximate 30%. The workload reference is therefore set to 0.7 to prevent overload. At step 20, the actual workload is sufficiently near the reference and the admission controller grants the request from Task #2, which then immediately performs a mode change. At the same time the admission controller changes the workload reference back to 1. The final frequencies are $\mathbf{f} = [\,4.822 \quad 4.378 \quad 5.276\,]$. □

5.3 *An approximate version*

The feedback scheduling algorithm proposed above, includes solving both Riccati and Lyapunov equations on-line. This is expensive, and a computationally cheaper algorithm is desirable. From Fig. 3 it seems that the cost functions could be approximated as quadratic functions of the sampling interval h. The relation between performance and sampling rates for LQG controllers was also discussed in Åström (1963), where it was shown that the cost is indeed quadratic for small sampling intervals.

Let the approximative cost $\tilde{J}(h)$ be defined as

$$\tilde{J}(h) = \alpha + \beta h^2.$$

By choosing a suitable nominal sampling interval h_0, and using $\tilde{J}(h_0) = J(h_0)$, $\tilde{J}_h(h_0) = J_h(h_0)$ the coefficients are given as:

$$\begin{aligned}\beta &= J_h(h_0)/(2h_0)\\ \alpha &= J(h_0) - \beta h_0^2\end{aligned}$$

The following approximate cost function and

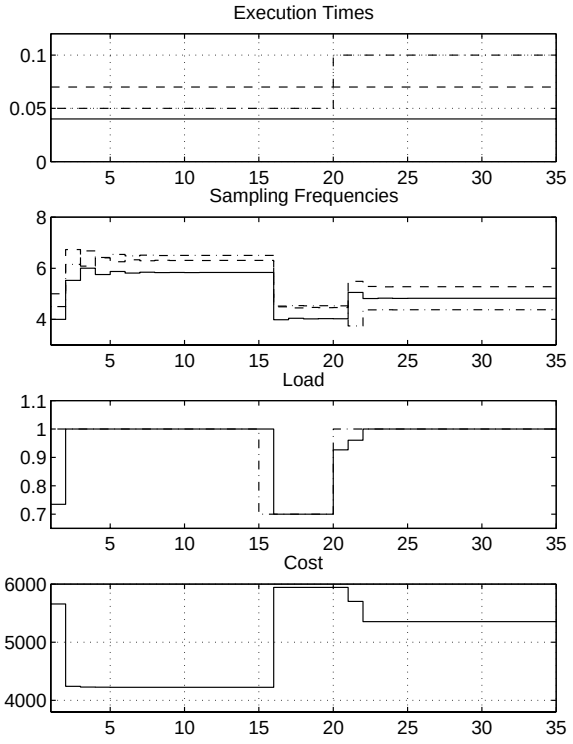

Fig. 6 Plots from Example 2. The top plot shows how the execution time for task #2 doubles at step 20. The second plot from the top shows how the sampling rates are adjusted by the feedback scheduler (Task #1–full, Task #2–dash-dotted, Task#3–dashed). The load plot shows how the workload reference (dash-dotted) goes from 100% to 70% at step 15, and back to 100% at step 20. The full line is the actual workload. The bottom plot shows the sum of the cost functions, i.e. the optimization criterion.

derivate are then obtained:

$$\tilde{V}(\mathbf{f}) = \sum_i^n \alpha_i + \beta_i/f^2$$
$$\tilde{V}_f(f) = -2\,[\,\beta_1/f^3, \quad \ldots \quad ,\beta_n/f^3\,]^{\mathrm{T}}$$
$$V_{ff} = 6\ \mathrm{diag}\,[\,\beta_1/f^4, \quad \ldots \quad ,\beta_n/f^4\,]$$

Using these approximate functions instead of the exact ones will give a much less computation intense problem.

The optimal solution is obtained from Eq. (7), and by inserting the approximative expression for V_f

$$2\beta_i/f_i^3 = \lambda C_i \ \Rightarrow \ f_i = (2\beta_i/(\lambda C_i))^{1/3}$$
$$\Rightarrow U = \sum_1^n C_i f_i = \sum_1^n C_i (2\beta_i/(\lambda C_i))^{1/3}$$

is obtained. Now, an explicit expression for the optimal solution is given by

$$\begin{cases} \lambda = (1/U \sum_1^n C_i^{2/3} (2\beta_i)^{1/3})^3 \\ f_i = (2\beta_i/(\lambda C_i))^{1/3} \end{cases} \qquad (8)$$

Example 2 continued
Again, consider the system with three pendulum controllers. From the explicit analytical expressions in Eq. (9) the following frequencies are calculated:

$$\mathbf{f} = [\,4.866 \quad 4.347 \quad 5.29\,]$$

Note that these frequencies are very close to those previously calculated in Example 2. This example shows that the approximative cost functions are sufficient to achieve good optimization results for some processes.

A similar approximative version can be designed for cost functions that are approximately linear in the sampling interval h, i.e. $\tilde{J} = \alpha + \beta h$. For example, for short sampling intervals in Fig. 4 the cost is almost linear. □

Comment 1
The rate of the feedback scheduler, and hence the rate of the changes to the sampling frequencies, is assumed to be much slower than the sampling frequencies. If the sampling frequencies are adjusted at a too high rate there will be problems with large jitter and possibly stability. □

6. CONCLUSION

In this paper a novel feedback scheduler has been proposed. For a class of control systems with convex cost functions, the feedback scheduler calculates the optimal resource allocation pattern. The calculation of cost functions and their dependence on sampling intervals has been investigated. Formulaes for calculating the cost functions and their derivatives have been presented. The feedback scheduler is demonstrated on a three control loops system, and the result is promising. The algorithm is, however, complex and quite computation-intense. The execution time needed for solving the optimization problem would probably exceed the execution times of most control tasks. By instead using approximative cost functions good results may be achieved at much less computational cost.

Many questions have been left out in this paper's discussion on feedback scheduling. One thing that has been neglected is what happens during the transient phase, when a sampling rate is changed. This problem must be treated differently depending on the used scheduling algorithm, e.g EDF or RMS.

Using a feedback scheduler approach it would be possible to design real-time, plug-and-play control systems. Tasks and resources are allowed to change and the system adapts on-line.

REFERENCES

Abdelzaher, T., Atkins, E. and Shin, K.: 1997, QoS negotiation in real-time systems, and its application to flight control, *Proceedings of the 18th IEEE Real-Time Systems Symposium*, San Francisco, CA, USA, pp. 228–238.

Årzén, K.-E.: 1999, A simple event based PID controller, *Proceedings of the 14th World Congress of IFAC*, Vol. Q, IFAC, Beijing, P.R. China, pp. 423–428.

Årzén, K.-E., Bernhardsson, B., Eker, J., Cervin, A., Nilsson, K., Persson, P. and Sha, L.: 1999, Integrated control and scheduling, *Technical Report ISRN LUTFD2/TFRT–7586–SE*, Department of Automatic Control, Lund, Sweden.

Åström, K. J.: 1963, On the choice of sampling rates in optimal linear systems, *Technical Report RJ-243*, San José Research Laboratory, IBM, San José, California.

Åström, K. J. and Wittenmark, B.: 1997, *Computer-Controlled Systems*, third edn, Prentice Hall, Englewood Cliffs, New Jersey, USA.

Buttazzo, G., Lipari, G. and Abeni, L.: 1998, Elastic task model for adaptive rate control, *Proceedings of the 19th IEEE Real-Time Systems Symposium*, Madrid, Spain.

Fletcher, R.: 1987, *Practical Methods of Optimization*, second edn, Wiley, UK.

Gill, P. E., Murray, W. and Wright, M. H.: 1981, *Practical Optimization*, Academic Press, UK.

Gustafsson, K. and Hagander, P.: 1991, Discrete-time LQG with cross-terms in the loss function and the noise description, *Technical Report TFRT-7475*, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Kosugi, N., Mitsuzawa, A. and Tokoro, M.: 1996, Importance-based scheduling for predictable real-time systems using MART, *Proceedings of the 4th Int. Workshop on Parallel and Distributed Systems*, Honolulu, Hawaii, USA, pp. 95–100.

Kosugi, N. and Moriai, S.: 1997, Dynamic scheduling for real-time threads by period adjustment, *Proceedings of the First World Congress on Systems Simulation*, Singapore, pp. 402–406.

Kosugi, N., Takashio, K. and Tokoro, M.: 1994, Modification and adjustment of real-time tasks with rate monotonic scheduling algorithm, *Proceedings of the Second Workshop on Parallel and Distributed Systems*, Cancun, Mexico, pp. 98–103.

Nilsson, J.: 1998, *Real-Time Control Systems with Delays*, PhD thesis, ISRN LUTFD2/TFRT–1049–SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Ryu, M. and Hong, S.: 1998, Toward automatic synthesis of schedulable real-time controllers, *Integrated Computer-Aided Engineering* **5**(3), 261–277.

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

Seto, D., Lehoczky, J. P. and Sha, L.: 1998, Task period selection and schedulability in real-time systems, *Proceedings of the 19th IEEE Real-Time Systems Symposium*, Madrid, Spain, pp. 188–198.

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

Shin, K. G. and Meissner, C. L.: 1999, Adaptation of control system performance by task reallocation and period modification, *Proceedings of the 11th Euromicro Conference on Real-Time Systems*, York, UK, pp. 29–36.

Stankovic, J. A., Lu, C., Son, S. H. and Tao, G.: 1999, The case for feedback control real-time scheduling, *Proceedings of the 11th Euromicro Conference on Real-Time Systems*, York, UK, pp. 11–20.

van Loan, C.: 1978, Computing integral involving the matrix exponential, *IEEE Transactions on Automatic Control* (AC–23), 395–404.

Zhao, Q. C. and Zheng, D. Z.: 1999, Stable real-time scheduling of a class of pertubed hds, *Proceedings of the 14th World Congress of IFAC*, Vol. J, IFAC, Beijing, P.R. China, pp. 91–96.

APPENDIX A – PROOF AND CONTINUATION OF THEOREM 1

Proof To find out how J depends on the sampling interval it remains to investigate $\mathbf{S}$. Eq. (5) is differentiated with respect to h:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \frac{d\mathbf{L}}{dh} & \mathbf{0} \end{bmatrix}^T \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix} +$$
$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{d\mathbf{S}}{dh} & \mathbf{0} \\ \mathbf{0} & \frac{d\mathbf{G}}{dh} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix} +$$
$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \frac{d\mathbf{L}}{dh} & \mathbf{0} \end{bmatrix} =$$
$$\frac{d\mathbf{Q}_d}{dh} + \begin{bmatrix} \frac{d\mathbf{\Phi}^{\mathrm{T}}}{dh} \\ \frac{d\mathbf{\Gamma}^{\mathrm{T}}}{dh} \end{bmatrix} \mathbf{S} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}^{\mathrm{T}} \\ \mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \frac{d\mathbf{S}}{dh} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}^{\mathrm{T}} \\ \mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \mathbf{S} \begin{bmatrix} \frac{d\mathbf{\Phi}}{dh} & \frac{d\mathbf{\Gamma}}{dh} \end{bmatrix}$$

Rearranging the terms yields

$$\begin{bmatrix} \mathbf{0} & \frac{d\mathbf{L}^{\mathrm{T}}}{dh}\mathbf{G} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \frac{d\mathbf{S}}{dh} & \mathbf{0} \\ \mathbf{0} & \frac{d\mathbf{G}}{dh} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{G}\frac{d\mathbf{L}}{dh} & \mathbf{0} \end{bmatrix} =$$
$$\begin{bmatrix} \mathbf{\Phi}^{\mathrm{T}} - \mathbf{L}^{\mathrm{T}}\mathbf{\Gamma}^{\mathrm{T}} \\ \mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \frac{d\mathbf{S}}{dh} \begin{bmatrix} \mathbf{\Phi} - \mathbf{\Gamma}\mathbf{L} & \mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{L} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \bar{\mathbf{W}} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{L} & \mathbf{I} \end{bmatrix} \quad (9)$$

where the block matrix $\bar{\mathbf{W}}$ is defined as

$$\bar{\mathbf{W}} = \frac{d\mathbf{Q}_d}{dh} + \begin{bmatrix} \frac{d\mathbf{\Phi}^{\mathrm{T}}}{dh} \\ \frac{d\mathbf{\Gamma}^{\mathrm{T}}}{dh} \end{bmatrix} \mathbf{S} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}^{\mathrm{T}} \\ \mathbf{\Gamma}^{\mathrm{T}} \end{bmatrix} \mathbf{S} \begin{bmatrix} \frac{d\mathbf{\Phi}}{dh} & \frac{d\mathbf{\Gamma}}{dh} \end{bmatrix}$$

The following Lyapunov equations for $\frac{d\mathbf{S}}{dh}$ and $\frac{d\mathbf{L}}{dh}$ are obtained by extracting elements from Eq. (10), and introducing $\mathbf{\Psi} = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{L})$.

$$\frac{d\mathbf{S}}{dh} = \mathbf{\Psi}^{\mathrm{T}} \frac{d\mathbf{S}}{dh} \mathbf{\Psi} + \begin{bmatrix} \mathbf{I} & -\mathbf{L}^{\mathrm{T}} \end{bmatrix} \bar{\mathbf{W}} \begin{bmatrix} \mathbf{I} \\ -\mathbf{L} \end{bmatrix}$$
$$\mathbf{G}\frac{d\mathbf{L}}{dh} = \mathbf{\Gamma}^{\mathrm{T}} \frac{d\mathbf{S}}{dh} \mathbf{\Psi} + \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{W}} \begin{bmatrix} \mathbf{I} \\ -\mathbf{L} \end{bmatrix}$$

The derivative $\frac{d\mathbf{L}}{dh}$ is needed later for the calculation of the second derivative of J. To calculate $\bar{\mathbf{W}}$ formulas for $\frac{d\mathbf{Q}_d}{dh}$, $\frac{d\mathbf{\Phi}}{dt}$, and $\frac{d\mathbf{\Gamma}}{dh}$ are needed. Since

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} h \right) = e^{\mathbf{\Sigma} h}$$

and $\frac{\mathbf{Q}_d}{dh}$ is given from Eq.uation (6) it is straightforward to calculate

$$\frac{d\mathbf{\Phi}}{dh} = \mathbf{A}\mathbf{\Phi}, \qquad \frac{d\mathbf{\Gamma}}{dh} = \mathbf{A}\mathbf{\Gamma} + \mathbf{B}, \qquad \frac{d\mathbf{Q}_d}{dh} = e^{\mathbf{\Sigma}^{\mathrm{T}} h} \mathbf{Q}_c e^{\mathbf{\Sigma} h}$$

$\bar{\mathbf{W}}$ can now be written as

$$\bar{\mathbf{W}} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{\mathrm{T}} \mathbf{Q}_c \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} +$$
$$\begin{bmatrix} \mathbf{A}\mathbf{\Phi} & \mathbf{A}\mathbf{\Gamma} + \mathbf{B} \end{bmatrix}^{\mathrm{T}} \mathbf{S} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \end{bmatrix}^{\mathrm{T}} \mathbf{S} \begin{bmatrix} \mathbf{A}\mathbf{\Phi} & \mathbf{A}\mathbf{\Gamma} + \mathbf{B} \end{bmatrix}$$

and now let $\mathbf{W}$ be

$$\mathbf{W} = \begin{bmatrix} \mathbf{I} & -\mathbf{L}^{\mathrm{T}} \end{bmatrix} \bar{\mathbf{W}} \begin{bmatrix} \mathbf{I} \\ -\mathbf{L} \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi} \\ -\mathbf{L} \end{bmatrix}^{\mathrm{T}} \mathbf{Q}_c \begin{bmatrix} \mathbf{\Psi} \\ -\mathbf{L} \end{bmatrix} +$$
$$\{\mathbf{\Psi}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} - \mathbf{L}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\}\mathbf{S}\mathbf{\Psi} + \mathbf{\Psi}^{\mathrm{T}}\mathbf{S}\{\mathbf{A}\mathbf{\Psi} - \mathbf{B}\mathbf{L}\}.$$

which now leads to the following expression, and $\frac{dJ}{dh}$ may now be calculated.

$$\begin{aligned}\frac{d\mathbf{S}}{dh} &= \mathbf{\Psi}^{\mathrm{T}}\frac{d\mathbf{S}}{dh}\mathbf{\Psi} + \mathbf{W} \\ \mathbf{G}\frac{d\mathbf{L}}{dh} &= \mathbf{\Gamma}^{\mathrm{T}}\frac{d\mathbf{S}}{dh}\mathbf{\Psi} + [\mathbf{\Gamma}^{\mathrm{T}} \quad \mathbf{I}]\,\mathbf{Q}_c\begin{bmatrix}\mathbf{\Psi}\\ -\mathbf{L}\end{bmatrix} + \mathbf{\Gamma}^{\mathrm{T}}\mathbf{S}(\mathbf{A}\mathbf{\Psi} - \mathbf{B}\mathbf{L}) + (\mathbf{A}\mathbf{\Gamma} + \mathbf{B})^{\mathrm{T}}\mathbf{S}\mathbf{\Psi}\end{aligned}$$

Theorem 2
The second derivative is given by

$$\begin{aligned}\frac{d^2J}{dh^2} = &\frac{1}{h}\Big\{tr(\frac{d^2\mathbf{S}}{dh^2}\mathbf{R}_1) + 2tr(\frac{d\mathbf{S}}{dh}\frac{\mathbf{R}_1}{dh}) + tr(\mathbf{S}\frac{d^2\mathbf{R}_1}{dh^2}) + \frac{d^2\bar{J}}{dh^2}\Big\} \\ &- \frac{2}{h^2}\Big\{tr(\frac{d\mathbf{S}}{dh}\mathbf{R}_1) + tr(\mathbf{S}\frac{\mathbf{R}_1}{dh}) + \frac{d\bar{J}}{dh}\Big\} + \frac{2}{h^3}\Big\{tr(\mathbf{S}\mathbf{R}_1) + \bar{J}\Big\}\end{aligned}$$

where

$$\begin{aligned}\frac{d^2\mathbf{S}}{dh^2} =&\mathbf{\Psi}^{\mathrm{T}}\frac{d^2\mathbf{S}}{dh^2}\mathbf{\Psi} + \mathbf{W}_2, \qquad \mathbf{W}_2 = \frac{d\mathbf{\Psi}^{\mathrm{T}}}{dh}\frac{d\mathbf{S}}{dh}\mathbf{\Psi} + \mathbf{\Psi}^{\mathrm{T}}\frac{d\mathbf{S}}{dh}\frac{d\mathbf{\Psi}}{dh} + \frac{d\mathbf{W}}{dh} \\ \frac{d\mathbf{\Psi}}{dh} =&\mathbf{A}\mathbf{\Phi} - (\mathbf{A}\mathbf{\Gamma} + \mathbf{B})\mathbf{L} - \mathbf{\Gamma}\frac{d\mathbf{L}}{dh}\end{aligned}$$

$$\begin{aligned}\frac{d\mathbf{W}}{dh} = &\begin{bmatrix}\frac{d\mathbf{\Psi}}{dh}\\ -\frac{d\mathbf{L}}{dh}\end{bmatrix}^{\mathrm{T}}\mathbf{Q}_c\begin{bmatrix}\mathbf{\Psi}\\ -\mathbf{L}\end{bmatrix} + \begin{bmatrix}\mathbf{\Psi}\\ -\mathbf{L}\end{bmatrix}^{\mathrm{T}}\mathbf{Q}_c\begin{bmatrix}\frac{d\mathbf{\Psi}}{dh}\\ -\frac{d\mathbf{L}}{dh}\end{bmatrix} + \\ &\{\mathbf{\Psi}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} - \mathbf{L}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\}(\frac{d\mathbf{S}}{dh}\mathbf{\Psi} + \mathbf{S}\frac{d\mathbf{\Psi}}{dh}) + \\ &(\mathbf{\Psi}^{\mathrm{T}}\frac{d\mathbf{S}}{dh} + \frac{d\mathbf{\Psi}^{\mathrm{T}}}{dh}\mathbf{S})\{\mathbf{A}\mathbf{\Psi} - \mathbf{B}\mathbf{L}\} + \\ &\{\frac{d\mathbf{\Psi}^{\mathrm{T}}}{dh}\mathbf{A}^{\mathrm{T}} - \frac{d\mathbf{L}^{\mathrm{T}}}{dh}\mathbf{B}^{\mathrm{T}}\}\mathbf{S}\mathbf{\Psi} + \mathbf{\Psi}^{\mathrm{T}}\mathbf{S}\{\mathbf{A}\frac{d\mathbf{\Psi}}{dh} - \mathbf{B}\frac{d\mathbf{L}}{dh}\}\end{aligned}$$

$$\frac{d^2\bar{J}}{dh^2} = tr(\mathbf{Q}_{1c}\frac{d\mathbf{R}_1}{dh}), \qquad \frac{d^2\mathbf{R}_1}{dh^2} = \mathbf{A}\mathbf{\Phi}R_{1c}\mathbf{\Phi}^{\mathrm{T}} + \mathbf{\Phi}R_{1c}\mathbf{A}^{\mathrm{T}}\mathbf{\Phi}^{\mathrm{T}}$$

Proof Proven with similar techniques as Theorem 1.

Lemma 1
The integral

$$\int_0^t e^{\mathbf{A}\tau}\mathbf{Q}e^{\mathbf{A}^{\mathrm{T}}\tau}d\tau$$

is calculated using the following equations taken from van Loan (1978). Consider the linear system

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\mathbf{A} & \mathbf{Q}\\ \mathbf{0} & \mathbf{A}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}, \tag{10}$$

which has the following solution

$$\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} = \exp(\begin{bmatrix}-\mathbf{A} & \mathbf{Q}\\ \mathbf{0} & \mathbf{A}^{\mathrm{T}}\end{bmatrix}t)\begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix}$$

If

$$\begin{bmatrix}\mathbf{\Psi}_{11} & \mathbf{\Psi}_{12}\\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_{22}\end{bmatrix} = \exp(\begin{bmatrix}-\mathbf{A} & \mathbf{Q}\\ \mathbf{0} & \mathbf{A}^{\mathrm{T}}\end{bmatrix}t)$$

then

$$x_1(t) = \mathbf{\Psi}_{11} x_1(0) + \mathbf{\Psi}_{12} x_2(0)$$
$$x_2(t) = \mathbf{\Psi}_{21} x_1(0) + \mathbf{\Psi}_{22} x_2(0)$$

An alternative way of solving Eq. (11) is

$$x_2(t) = e^{\mathbf{A}^{\mathrm{T}} t} x_2(0) \quad \Rightarrow \dot{x}_1(t) = -\mathbf{A} x_1(t) + \mathbf{Q} e^{\mathbf{A}^{\mathrm{T}} t} x_2(0)$$
$$\Rightarrow \quad x_1(t) = e^{-\mathbf{A}t} x_1(0) + e^{-\mathbf{A}t} \int_0^t e^{\mathbf{A}\tau} \mathbf{Q} e^{\mathbf{A}^{\mathrm{T}} \tau} d\tau x_2(0)$$

Identification of the terms from the different solutions now gives

$$I(t) = \int_0^t e^{\mathbf{A}\tau} \mathbf{Q} e^{\mathbf{A}^{\mathrm{T}} \tau} d\tau = \mathbf{\Psi}_{22}^{\mathrm{T}} \mathbf{\Psi}_{12},$$

which concludes the lemma.

Comment
The double integral $\int_0^t I(\tau) d\tau$ may be calculated in a similar fashion by extending the equations above and using the following system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\mathbf{A} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{Q} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

□