Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

PI Control and Scheduling Design for Embedded Control Systems

Sven Reimann * **Wei Wu** * **Steven Liu** *

* *Institute of Control Systems, University of Kaiserslautern P.O. Box 3049, 67653 Kaiserslautern, Germany (e-mail: reimann@eit.uni-kl.de, wu@eit.uni-kl.de, sliu@eit.uni-kl.de).*

Abstract: This paper addresses the control and scheduling design for control systems sharing a limited computation resource, especially embedded control systems (ECSs). The complementary design of controller and scheduler is for improving the control performance and achieving an efficient utilization of the limited computation resources. A novel PI control design method as well as a novel output-based scheduling approach are introduced to achieve setpoint-tracking for multiple plants. For verifying stability the whole system, namely all plants with the scheduled controllers, is modeled as a discrete-time switched linear system. The stability criteria is eventually formulated as a linear matrix inequality (LMI) feasibility problem. The effectivenes of the proposed control and scheduling approach is illustrated by a practical implementation where two DC motors are controlled by one processor.

Keywords: Embedded systems, Scheduling algorithms, PI controllers, Target tracking, Lyapunov stability

1. INTRODUCTION

In embedded control system, the computational resources are generally limited. This usually means that several control loops have to be implemented on one processor. Therefore an efficient usage of the resources is essential requiring an intelligent scheduling algorithm. In literature this general problem is approached differently.

One class of publication on control task scheduling for embedded control systems focuses on finding a tradeoff in the distribution of the resources for the control tasks under a constraint resource utilization, see Seto et al. [1996], Eker et al. [2000], Cervin et al. [2002, 2011], Martí et al. [2004, 2009], Henriksson and Cervin [2005], Castañé et al. [2006]. Thereby the control tasks are scheduled preemptively under an assigned sampling period. The sampling periods are determined in a frame of optimization which considers the relationship between the sampling period and the quality of control. In an early approach Seto et al. [1996] assume that the controllers are designed in the continuous-time domain and afterwards discretized. A performance index quantifying the performance of the digitalized control law of a given sampling period is introduced. The optimal sampling periods result from an offline optimization problem subject to some schedulability constraint and are used for discretizing the controllers.

Later publications extend the work of Seto et al. [1996] to an online sampling period assignment, see Eker et al. [2000], Cervin et al. [2002, 2011], Martí et al. [2004, 2009], Henriksson and Cervin [2005], Castañé et al. [2006]. This is often referred to as feedback scheduling. Eker et al. [2000] propose an online optimization method for adjusting the sampling period online based on the current processor load. Therefore, a state feedback controller is considered and the LQ cost function is expressed as a function of the sampling period. This is extended by Cervin et al. [2002] to general linear dynamic controllers. Thus, the approaches by Eker et al. [2000] and Cervin et al. [2002] focus on the resource allocation under a time-varying processor load.

In the approaches by Martí et al. [2004, 2009], Henriksson and Cervin [2005], Castañé et al. [2006], Cervin et al. [2011] the current plant states are incorporated in the online optimization, i.e. the sampling periods are adapted online based on a state-based performance index. Martí et al. [2004, 2009] introduce a heuristic cost function taking current states into account for the sampling period assignment. Another approach by Henriksson and Cervin [2005] makes use of a quadratic cost function related to the sampling period and the current states for adjusting the sampling period online. Thereby also the expected future plant noise is taken into account. Henriksson and Cervin [2005] assume a state feedback control law whereas the approach in Castañé et al. [2006] generalizes for linear dynamic controllers. Cervin et al. [2011] complements the feedback scheduler from Castañé et al. [2006] by taking the computational delay into account and by adding a noise estimator as the evaluation of the cost function depends on the noise intensity. Further a practical evaluation is given in Cervin et al. [2011].

In a parallel line of research a complementary design of scheduler and controller is investigated in Rehbinder and Sanfridson [2000], Lincoln and Bernhardsson [2002], Görges et al. [2007], Cervin and Alriksson [2006], Gaid et al. [2009], Reimann et al. [2013], also referred to as control and scheduling codesign. Those approaches can be distinguished on the basis of offline and online scheduling. In the approaches Rehbinder and Sanfridson [2000], Lincoln and Bernhardsson [2002], Görges et al. [2007] an optimal state feedback controller is designed jointly with a control task sequence in an offline optimization problem. This means that the control tasks are scheduled non-preemptively according to the offline determined sequence which is repeated perpetually. The benefit of this codesign approach is that the controller and scheduler are harmonized with each other. This means that the jitter due to the scheduling is incorporated in the model and thus taken into account in the control design. In the work of Rehbinder and Sanfridson [2000] a constant time slot size is assumed where one control task is released at the beginning of a time slot and completed before the end of the slot. By fixing a task sequence with a defined periodicity the control design is formulated as a periodic linear quadratic control problem. The optimal control task sequence is then derived by exhaustive search among all task sequence permutations. To make the problem computationally tractable a maximum periodicity is introduce. The resulting sampling period of each control task is then a multiple of the slot size. The method by Lincoln and Bernhardsson [2002] determines the control task sequence via dynamic programming with tree pruning. Both approaches Rehbinder and Sanfridson [2000], Lincoln and Bernhardsson [2002] deal with the optimal control and scheduling codesign problem in the sense of LQG. In Görges et al. [2007] the lifting technique is applied to transform the control and scheduling codesign problem to an optimal periodic control problem in the sense of LQR. The optimal control task sequence is then derived by exhaustive search. Additional to Rehbinder and Sanfridson [2000], Lincoln and Bernhardsson [2002] the computational delay is included in the modeling and codesign.

Opposite to the offline scheduling approaches Cervin and Alriksson [2006], Gaid et al. [2009], Reimann et al. [2013] propose an extension to online scheduling where the control task sequence is adapted online based on current plant state information. In the approach by Cervin and Alriksson [2006] the controller and scheduler are determined jointly by solving an infinite-horizon optimization problem using relaxed dynamic programming. However this approach involves a large scheduling overhead. Gaid et al. [2009] decompose the control and scheduling problem into two subproblems. In the first subproblem a periodic control task sequence is determined using the branch and bound method. Based on the derived control task sequence the optimal control gains are derived in the second subproblem applying the lifting technique. Further a state feedback scheduler is proposed to improve the control performance with respect to the offline scheduling sequence. The approach in Reimann et al. [2013] proposes a sub-optimal control and scheduling design where the upper bound of the LQR costs is optimized. Additional to the other approaches an output-based scheduling is derived which includes an observer design for the state estimation. However in this research line only state feedback control is considered. An extension to linear dynamic controllers is still missing in literature to the authors' knowledge which is the main motivation for the proposed work.

In the proposed approach the idea of determining the control task sequence online will be investigated for a control task scheduling problem with linear dynamic controllers. Hereby we focus on proportional-integral (PI) control for tracking a reference signal. This is also motivated by the fact that PI control is still one of the most widely used feedback control methods. Further, the online scheduling allows to react on disturbances and reference changes. For designing the scheduler and the controller ideas from event-based PID control, see Årzén [1999], Durand and Marchand [2009], Tiberi et al. [2012], Lehmann et al. [2012], are adapted. The proposed approach can be also extended to PID control with derivative action.

The rest of the paper is outlined as follows. Section 2 formulates the problem and explains the non-preemptive online scheduling setup. In Section 3 the PI controller is introduced which is adapted automatically with respect to the time-varying sampling period. The time-varying sampling period of the control loops results from the scheduling which is introduced in Section 4. Section 5 discusses the stability under the introduced controller and scheduler. Therefore, the holistic system is modeled as discrete-time switched system. In Section 6 the results from a practical implementation of the proposed control and scheduling approach are presented. Thereby, two DC motors are controlled to a reference speed by one embedded processor. Section 7 concludes the paper and gives an outlook for future work.

Throughout the paper diag$(\cdot)$ denotes a block-diagonal matrix and $\|\boldsymbol{x}\|_2$ denotes the Euclidean norm of a vector $\boldsymbol{x}$. Furthermore, a matrix $\left(\begin{smallmatrix}\boldsymbol{A} & *\\ \boldsymbol{B} & \boldsymbol{C}\end{smallmatrix}\right)$ represents a symmetric matrix $\left(\begin{smallmatrix}\boldsymbol{A} & \boldsymbol{B}^T\\ \boldsymbol{B} & \boldsymbol{C}\end{smallmatrix}\right)$.

2. PROBLEM FORMULATION

Consider a set of M independent linear time-invariant n_i-order SISO systems P_i, $i \in \{1, \ldots, M\}$ controlled by a set of control tasks T_i, $i \in \{1, \ldots, M\}$ using a single embedded processor. Each plant is described by a continuous-time state equation

$$\begin{aligned}\dot{\boldsymbol{x}}_{\mathrm{p}i}(t) &= \boldsymbol{A}_{\mathrm{p}i}\boldsymbol{x}_{\mathrm{p}i}(t) + \boldsymbol{B}_{\mathrm{p}i}u_i(t)\\ y_i(t) &= \boldsymbol{C}_{\mathrm{p}i}\boldsymbol{x}_{\mathrm{p}i}(t)\end{aligned} \tag{1}$$

where $\boldsymbol{x}_{\mathrm{p}i}(t) \in \mathbb{R}^{n_i}$ is the plant's state, $u_i(t) \in \mathbb{R}$ is the control signal and $y_i(t) \in \mathbb{R}$ is the output.

The input-delay, i.e. the time delay from the measurement of the output until the actuation is neglected in the model, see Reimann et al. [2013] for a detailed discussion on the input-delay.

Although the plants are independent, using a common processing unit with a limited computation capacity leads to an interconnected large-scale system. Hence, achieving the main objective is a complex problem. In order to solve this complex problem, an intelligent scheduling policy besides an implementation-aware controller are indispensable. In our approach the time line is partitioned into time slots of a given length h similar to Gaid et al. [2009]. At the time instant t_k, the beginning of each time slot, the outputs of all plants are sampled simultaneously and fed to the scheduler. Based on this information a scheduling law is applied for making a decision about the control task $\mathrm{T}_{j(k)}$ to be executed, with $j(k) \in \mathbb{J} = \{1, \ldots, M\}$. The computed control signal $u_{j(k)}(t_k)$ is finally forwarded to the plant

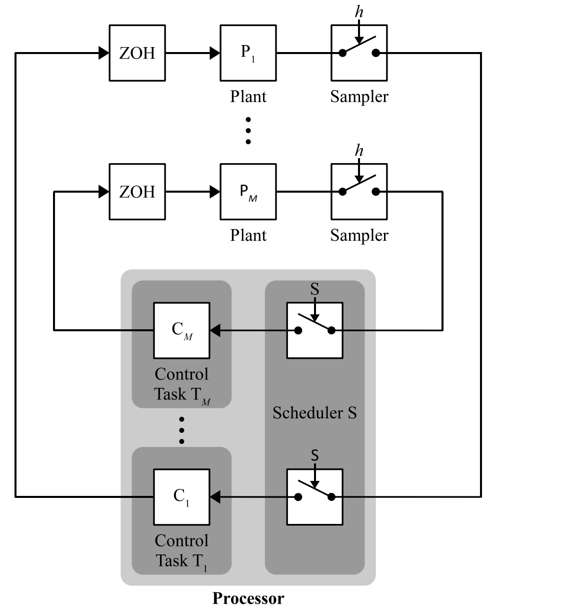

Fig. 1. General architecture of an embedded control system

$P_{j(k)}$. For the other plants $P_i \neq P_{j(k)}$, the control signal is held constant until a new one is delivered. The length of the time slot $h = t_{k+1} - t_k$ is chosen larger than or equal to the worst-case execution time of the scheduler S and the control task $T_{j(k)}$ such that the execution of the scheduler and a control task can be finished within one time slot. As each plant is controlled by a PI controller with an equivalent structure and order the worst-case execution time differs little with respect to the control task index $j(k)$. The remaining idle time can be devoted to executing non-control tasks, see Fig. 2. Additionally, by choosing the slot size h appropriately a certain percentage of the processor resources can be reserved for non-control tasks.

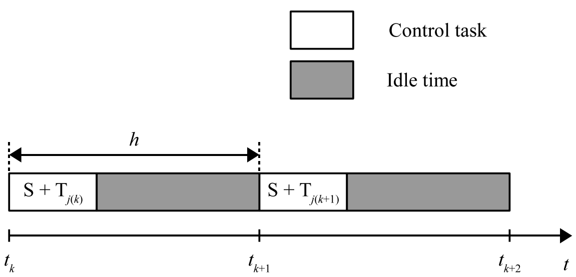

Fig. 2. ECS timing diagram

Due to the discrete nature of the computation platform, a discrete-time representation of the continuous-time state equation (1) is crucial. The discretization is applied with the discretization interval h equal to the length of a time slot. Thus, discretizing (1) using zero-order hold leads to the discrete-time model

$$\begin{aligned} \boldsymbol{x}_{\mathrm{p}i}(k+1) &= \boldsymbol{\Phi}_i \boldsymbol{x}_{\mathrm{p}i}(k) + \boldsymbol{\Gamma}_i u_i(k) \\ y_i(k) &= \boldsymbol{C}_i \boldsymbol{x}_{\mathrm{p}i}(k) \end{aligned} \tag{2}$$

with

$$\boldsymbol{\Phi}_i = e^{\boldsymbol{A}_{\mathrm{p}i} h}, \quad \boldsymbol{\Gamma}_i = \int_0^h e^{\boldsymbol{A}_{\mathrm{p}i} s} ds \boldsymbol{B}_{\mathrm{p}i}, \quad \boldsymbol{C}_i = \boldsymbol{C}_{\mathrm{p}i}.$$

3. PI CONTROL DESIGN

As we focus on tracking a constant reference signal a PI controller is applied which is given by its digital implementation. For defining the PI controller, we distinguish whether the control signal is updated or not within the discretization interval $t_k \leq t < t_{k+1}$. If $i = j(k)$, then the integrator state and the control signal is updated, i.e.

$$x_{\mathrm{c}i}(k) = x_{\mathrm{c}i}(k-1) + \beta_i(k)\big(r_i(k) - y_i(k)\big) \tag{3a}$$

$$u_i(k) = K_{\mathrm{P}i}\big(r_i(k) - y_i(k)\big) + K_{\mathrm{I}i} x_{\mathrm{c}i}(k). \tag{3b}$$

If $i \neq j(k)$, then the integrator state and the control signal is not updated at all, i.e.

$$x_{\mathrm{c}i}(k) = x_{\mathrm{c}i}(k-1) \tag{4a}$$

$$u_i(k) = u_i(k-1) \tag{4b}$$

where $x_{\mathrm{c}i}(k) \in \mathbb{R}$ is the integrator state, $K_{\mathrm{P}i}, K_{\mathrm{I}i} \in \mathbb{R}$ are the proportional and integrator gain respectively. $r_i(k) \in \mathbb{R}$ is the reference signal which is assumed to be piecewise constant and $\beta_i(k)$ indicates the integrator update rate. The parameter $\beta_i(k)$ needs to be chosen appropriately with respect to the control update interval which is time-varying due to the scheduling.

There are different possibilities for defining $\beta_i(k)$, see Durand and Marchand [2009] for a detailed discussion. The intuitive choice for $\beta_i(k)$ would be the elapsed time since the previous control update $h_{\mathrm{act}_i}(k)$, i.e. the effective control update interval. However, the control update interval may become large for one plant due to the absence of a control update for this plant, for instance in the case that the plant is in the steady state and the resources are used for the control tasks of the other plants. When the reference signal changes the integral part explodes which can result in a large overshoot. Therefore the control update interval is saturated, i.e. the parameter $\beta_i(k)$ is reset to h when the control update interval exceeds a defined boundary $h_{\max_i}$. Thus, $\beta_i(k)$ is defined as

$$\beta_i(k) = \begin{cases} h_{\mathrm{act}_i}(k) & \text{if} \quad h_{\mathrm{act}_i}(k) \leq h_{\max_i} \\ h & \text{if} \quad h_{\mathrm{act}_i}(k) > h_{\max_i}. \end{cases} \tag{5}$$

Under the given assumption of a constant discretization interval h the possible values of $\beta_i(k)$ are given by a set $\beta_i(k) \in \mathbb{H}_i = \{h, 2h, ..., N_i h\}$ with $h_{\max_i} = N_i \cdot h$.

The logical variable

$$\delta_{ij(k)} = \begin{cases} 0 & \text{if} \quad i \neq j(k) \\ 1 & \text{if} \quad i = j(k) \end{cases}$$

is introduced to model the general control law with respect to the task index $j(k)$ which is later used for modeling the closed loop system.

We introduce an error $e_i(k)$ defined as

$$\begin{aligned} e_i(k) &= \big(y_i(k) - r_i(k)\big) - \big(\hat{y}_i(k) - \hat{r}_i(k)\big) \\ &= \big(\boldsymbol{C}_i \boldsymbol{x}_{\mathrm{p}i}(k) - r_i(k)\big) - \big(\boldsymbol{C}_i \hat{\boldsymbol{x}}_{\mathrm{p}i}(k) - \hat{r}_i(k)\big) \end{aligned} \tag{6}$$

where $\hat{y}_i(k)$ and $\hat{r}_i(k)$ indicate the output and reference respectively at the time instant of the previous control update of the plant P_i before time instant k. This error gives an indication if we are in the transient state or in the steady state based on which we can deduce the necessity for a control update. Assuming that the reference signal is constant, i.e. $r_i(k) = \hat{r}_i(k)$, a large error indicates the transient phase where a control update is necessary and a small error indicates that we are in the steady state. If the reference signal changes the error increases such that we can conclude that an increasing error $e_i(k)$ indicates the requirement for a control update which is later used for the scheduling law. This error is equivalent to the error used by Årzén [1999] for the event generator.

Defining the error $e_i(k)$ allows to rewrite the control law (4) for the case the control input is not updated as

$$x_{\mathrm{c}i}(k) = x_{\mathrm{c}i}(k-1) \tag{7a}$$

$$u_i(k) = K_{Pi}\big(\hat{r}_i(k) - \boldsymbol{C}_i\hat{\boldsymbol{x}}_{\mathrm{p}i}(k)\big) + K_{Ii}x_{\mathrm{c}i}(k). \tag{7b}$$

As $\hat{r}_i(k) - \boldsymbol{C}_i\hat{\boldsymbol{x}}_{\mathrm{p}i}(k) = r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k) + e_i(k)$ the control law is then rewritten with respect to the task index $j(k)$

$$x_{\mathrm{c}i}(k) = x_{\mathrm{c}i}(k-1) + \delta_{ij(k)}\beta_i(k)\big(r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k)\big) \tag{8a}$$

$$\begin{aligned} u_i(k) = K_{Pi}\big(&r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k) + (1-\delta_{ij(k)})e_i(k)\big) \\ &+ K_{Ii}x_{\mathrm{c}i}(k). \end{aligned} \tag{8b}$$

4. SCHEDULING CRITERIA

Consider the scheduling law

$$j(k) = \arg\max_{i\in\mathbb{J}}\left\{\lambda_{\mathrm{e}i}\|e_i(k)\|_2^2 + \lambda_{\mathrm{r}i}\|r_i(k) - y_i(k)\|_2^2\right\} \tag{9}$$

with the design parameters $\lambda_{\mathrm{e}i} > 0$ and $\lambda_{\mathrm{r}i} > 0$. As explained in Section 3 a large value of the first term $\|e_i(k)\|_2^2$ in the argument indicates the necessity for an update of the control input. However even if $\|e_i(k)\|_2$ is zero there still maybe a steady state control error such that an update of the control input is required. This is incorporated in the second term $\|r_i(k) - y_i(k)\|_2^2$ which measures the control error of the output to the reference signal. It is worth emphasizing that from $\|r_i(k) - y_i(k)\|_2 = 0$ we cannot conclude that the control error has converged to zero. The control error can also be zero at certain time instants in case of an oscillation around the reference signal. A high value of each term indicates the necessity of updating the control signal. Therefore both terms are considered in the scheduling law. Including only one of the terms may result in an unsatisfactory performance.

5. STABILITY ANALYSIS

5.1 Modeling of the closed-loop system

In order to derive a stability criterion which can guarantee the asymptotic stability of all plants a model of the holistic closed-loop system is built. Combining the discrete-time plant (2) and the PI controller (8) leads to

$$\begin{aligned} \boldsymbol{x}_{\mathrm{p}i}(k+1) = {}& \boldsymbol{\Phi}_i\boldsymbol{x}_{\mathrm{p}i}(k) \\ &+ \boldsymbol{\Gamma}_iK_{Ii}\big(x_{\mathrm{c}i}(k-1) + \delta_{ij(k)}\beta_i(k)\big(r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k)\big)\big) \\ &+ \boldsymbol{\Gamma}_iK_{Pi}\big(r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k) + (1-\delta_{ij(k)})e_i(k)\big) \end{aligned} \tag{10}$$

$$x_{\mathrm{c}i}(k) = x_{\mathrm{c}i}(k-1) + \delta_{ij(k)}\beta_i(k)\big(r_i(k) - \boldsymbol{C}_i\boldsymbol{x}_{\mathrm{p}i}(k)\big) \tag{11}$$

Further the dynamic behavior of the error $e_i(k)$ is analyzed. Based on (6) the error prediction $e_i(k+1)$ is given by

$$\begin{aligned} e_i(k+1) = {}& \big(y_i(k+1) - r_i(k+1)\big) - \big(y_i(k) - r_i(k)\big) \\ &+ (1-\delta_{ij(k)})e_i(k) \end{aligned} \tag{12}$$

In the following we assume that the reference signal is constant, i.e. $r_i(k+1) = r_i(k)$. Thus, we have

$$e_i(k+1) = \boldsymbol{C}_i\big(\boldsymbol{x}_{\mathrm{p}i}(k+1) - \boldsymbol{x}_{\mathrm{p}i}(k)\big) + (1-\delta_{ij(k)})e_i(k). \tag{13}$$

Combining (13) and (10)-(11) the closed-loop system of the plant P_i is given by

$$\boldsymbol{x}_i(k+1) = \boldsymbol{A}_{ij(k)}(k)\boldsymbol{x}_i(k) + \boldsymbol{F}_{ij(k)}(k)r_i(k) \tag{14}$$

with $\boldsymbol{x}_i(k) = \big(\boldsymbol{x}_{\mathrm{p}i}(k)\ x_{\mathrm{c}i}(k-1)\ e_i(k)\big)^T$ and the matrices (15), (16).

We see that the time dependency of the matrices $\boldsymbol{A}_{ij(k)}(k)$ and $\boldsymbol{F}_{ij(k)}(k)$ caused by the parameter $\beta_i(k)$ only appears for the case $\delta_{ij(k)} = 1$, i.e. the control input of the plant P_i is updated. If the control input of the plant P_i is not updated, i.e. $\delta_{ij(k)} = 0$, at a time instant k the closed-loop system matrix $\boldsymbol{A}_{ij(k)}$ is time-invariant for given i and $j(k)$. In order to deal with the time-varying parameter $\beta_i(k)$ for the case $\delta_{ij(k)} = 1$ another switching index ℓ is introduced which subsumes the possibilities for the integrator update rate $\beta_i(k) \in \mathbb{H}_i$. Substituting $\beta_i(k) = \ell \cdot h$ in (15) and (16) with ℓ arbitrary from the set $\{1, ..., N_{j(k)}\}$ (14) can be rewritten as

$$\boldsymbol{x}_i(k+1) = \tilde{\boldsymbol{A}}_{ij(k)\ell}\boldsymbol{x}_i(k) + \tilde{\boldsymbol{F}}_{ij(k)\ell}r_i(k) \tag{17}$$

Thus, the time dependency is modeled by the additional switching index ℓ. Even though the switching index ℓ is only relevant for $i = j(k)$ the closed-loop model given by (17) is also valid for $i \neq j(k)$.

The overall closed-loop system is then described by the block-diagonal discrete-time switched linear system

$$\boldsymbol{x}(k+1) = \tilde{\boldsymbol{A}}_{j(k)\ell}\boldsymbol{x}(k) + \tilde{\boldsymbol{F}}_{j(k)\ell}\boldsymbol{r}(k) \tag{18}$$

with

$$\begin{aligned} \boldsymbol{x}(k) &= \big(\boldsymbol{x}_1^T(k)\ \cdots\ \boldsymbol{x}_M^T(k)\big)^T \in \mathbb{R}^n \\ \boldsymbol{r}(k) &= \big(r_1(k)\ \cdots\ r_M(k)\big)^T \in \mathbb{R}^M \\ \tilde{\boldsymbol{A}}_{j(k)\ell}(k) &= \mathrm{diag}\left(\tilde{\boldsymbol{A}}_{1j(k)\ell}(k), \ldots, \tilde{\boldsymbol{A}}_{Mj(k)\ell}(k)\right) \in \mathbb{R}^{n\times n} \\ \tilde{\boldsymbol{F}}_{j(k)\ell}(k) &= \mathrm{diag}\left(\tilde{\boldsymbol{F}}_{1j(k)\ell}(k), \ldots, \tilde{\boldsymbol{F}}_{Mj(k)\ell}(k)\right) \in \mathbb{R}^{n\times M} \end{aligned}$$

and $n = n_1 + ... + n_M + 2M$. In the model (18) the index $j(k) \in \{1, ..., M\}$ is determined by the scheduler (9) whereas the index ℓ is assumed to be arbitrary from the set $\{1, ..., N_{j(k)}\}$ as it depends on the actual and the previous scheduling indices of the controlled plant.

$$\boldsymbol{A}_{ij}(k) = \begin{pmatrix} \boldsymbol{\Phi}_i - \boldsymbol{\Gamma}_i(K_{Pi} + \delta_{ij(k)}\beta_i(k)K_{Ii})\boldsymbol{C}_i & \boldsymbol{\Gamma}_iK_{Ii} & (1-\delta_{ij(k)})\boldsymbol{\Gamma}_iK_{Pi} \\ -\delta_{ij(k)}\beta_i(k)\boldsymbol{C}_i & 1 & 0 \\ \boldsymbol{C}_i\big(\boldsymbol{\Phi}_i - \boldsymbol{\Gamma}_i(K_{Pi} + \delta_{ij(k)}\beta_i(k)K_{Ii})\boldsymbol{C}_i - \boldsymbol{I}\big) & \boldsymbol{C}_i\boldsymbol{\Gamma}_iK_{Ii} & (1-\delta_{ij(k)})(\boldsymbol{C}_i\boldsymbol{\Gamma}_iK_{Pi} + 1) \end{pmatrix} \tag{15}$$

$$\boldsymbol{F}_{ij}(k) = \begin{pmatrix} \boldsymbol{\Gamma}_i(K_{Pi} + \delta_{ij(k)}\beta_i(k)K_{Ii}) \\ \delta_{ij(k)}\beta_i(k) \\ \boldsymbol{C}_i\boldsymbol{\Gamma}_i(K_{Pi} + \delta_{ij(k)}\beta_i(k)K_{Ii}) \end{pmatrix} \tag{16}$$

5.2 Stability criterion

Without loss of generality we assume that $r_i(k) = 0$ in the following derivation. Further, we assume that the control and scheduling parameters $K_{\mathrm{P}i}$, $K_{\mathrm{I}i}$, $\lambda_{\mathrm{r}i}$ and $\lambda_{\mathrm{e}i}$ are given. For analyzing the stability the scheduling law (9) is transformed to

$$\begin{aligned} j(k) &= \arg\max_{i\in\mathbb{J}} \left\{ \lambda_{\mathrm{e}i} e_i^T(k) e_i(k) + \lambda_{\mathrm{r}i} \boldsymbol{x}_{\mathrm{p}i}(k)^T \boldsymbol{C}_i^T \boldsymbol{C}_i \boldsymbol{x}_{\mathrm{p}i}(k) \right\} \\ &= \arg\max_{i\in\mathbb{J}} \boldsymbol{x}_i^T(k) \boldsymbol{Q}_i \boldsymbol{x}_i(k) \end{aligned} \tag{19}$$

with $\boldsymbol{Q}_i = \mathrm{diag}\big(\lambda_{\mathrm{r}i} \boldsymbol{C}_i^T \boldsymbol{C}_i, 0, \lambda_{\mathrm{e}i}\big)$. Based on (19) we know that for a given task index $j(k)$ we have

$$\boldsymbol{x}_{j(k)}^T(k) \boldsymbol{Q}_{j(k)} \boldsymbol{x}_{j(k)}(k) > \boldsymbol{x}_i^T(k) \boldsymbol{Q}_i \boldsymbol{x}_i(k) \quad \forall i \neq j(k). \tag{20}$$

Summing (20) over all $i \neq j(k)$ yields

$$(M-1)\boldsymbol{x}_{j(k)}^T(k) \boldsymbol{Q}_{j(k)} \boldsymbol{x}_{j(k)}(k) > \sum_{i=1, i\neq j(k)}^{M} \boldsymbol{x}_i^T(k) \boldsymbol{Q}_i \boldsymbol{x}_i(k)$$

which is equivalent to

$$\boldsymbol{x}^T(k) \tilde{\boldsymbol{Q}}_{j(k)} \boldsymbol{x}(k) > 0. \tag{21}$$

where $\tilde{\boldsymbol{Q}}_{j(k)} = \mathrm{diag}\big(-\boldsymbol{Q}_1, ..., (M-1)\boldsymbol{Q}_{j(k)}, ..., -\boldsymbol{Q}_M\big)$.

For the stability analysis consider the multiple Lyapunov function, see e.g. Daafouz et al. [2002]

$$V(k) = \boldsymbol{x}^T(k) \boldsymbol{P}_{j(k)} \boldsymbol{x}(k) \tag{22}$$

with $\boldsymbol{P}_{j(k)} \in \mathbb{R}^{n\times n}$ symmetric and positive definite and $j(k) \in \mathbb{J}$. In order to prove asymptotic stability the inequality $\Delta V(k) = V(k+1) - V(k) < 0$ must be satisfied for all $\boldsymbol{x}(k) \neq \boldsymbol{0}$ under the scheduling law (19).

For each task index $j(k) \in \{1, \ldots, M\}$ the stability condition is then rewritten as

$$\begin{aligned} &\Delta V(k) = \boldsymbol{x}^T(k) \left(\tilde{\boldsymbol{A}}_{j(k)\ell}^T \boldsymbol{P}_i \tilde{\boldsymbol{A}}_{j(k)\ell} - \boldsymbol{P}_{j(k)} \right) \boldsymbol{x}(k) < 0 \\ &\text{for all } \boldsymbol{x}(k) \text{ such that } \boldsymbol{x}^T(k) \tilde{\boldsymbol{Q}}_{j(k)} \boldsymbol{x}(k) > 0 \end{aligned} \tag{23}$$

for all $i \in \{1, \ldots, M\}$ and $\ell \in \{1, ..., N_{j(k)}\}$. Applying the lossless S-procedure to (23), see e.g. [Boyd et al., 1994, Sec. 2.6.3] for details, the following Theorem is deduced.

Theorem 1. If there exist a set of symmetric and positive definite matrices $\boldsymbol{P}_{j(k)}$ and non-negative scalars $\mu_{j(k)}$, $j(k) = \{1, ..., M\}$ satisfying

$$\boldsymbol{P}_{j(k)} - \tilde{\boldsymbol{A}}_{j(k)\ell}^T \boldsymbol{P}_i \tilde{\boldsymbol{A}}_{j(k)\ell} - \mu_{j(k)} \tilde{\boldsymbol{Q}}_{j(k)} > \boldsymbol{0} \tag{24}$$

for all $(j(k), i, \ell) \in \mathbb{J} \times \mathbb{J} \times \{1, ..., N_{j(k)}\}$ then the closed-loop system (18) is globally asymptotically stabilized by the scheduling law (9).

In the case of a constant reference signal $r_i(k) \neq 0$ the equilibrium point of the controlled system (14) is not the origin but $\boldsymbol{x}_{\mathrm{eq}i} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{p}i}(\infty) & x_{\mathrm{c}i}(\infty) & 0 \end{pmatrix}^T$. The output converges to the reference value, i.e. $r_i(\infty) = \boldsymbol{C}_{\mathrm{p}i} \boldsymbol{x}_{\mathrm{p}i}(\infty)$. In the equilibrium point we have $0 = \boldsymbol{A}_{\mathrm{p}i} \boldsymbol{x}_{\mathrm{p}i}(\infty) + \boldsymbol{B}_{\mathrm{p}i} u_i(\infty)$ with $u_i(\infty) = K_{Ii} x_{\mathrm{c}i}(\infty)$ such that

$$\begin{pmatrix} \boldsymbol{0} \\ r_i(\infty) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{\mathrm{p}i} & \boldsymbol{B}_{\mathrm{p}i} K_{Ii} \\ \boldsymbol{C}_{\mathrm{p}i} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{\mathrm{p}i}(\infty) \\ x_{\mathrm{c}i}(\infty) \end{pmatrix}. \tag{25}$$

Therefore

$$\begin{pmatrix} \boldsymbol{x}_{\mathrm{p}i}(\infty) \\ x_{\mathrm{c}i}(\infty) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{\mathrm{p}i} & \boldsymbol{B}_{\mathrm{p}i} K_{Ii} \\ \boldsymbol{C}_{\mathrm{p}i} & 0 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{0} \\ r_i(\infty) \end{pmatrix}. \tag{26}$$

As the reference signal is constant in the equilibrium point we have $e_i(\infty) = 0$. By a coordinate change that shifts the equilibrium point into the origin the stability can be analyzed with the Lyapunov function

$$V(k) = \big(\boldsymbol{x}(k) - \boldsymbol{x}_{\mathrm{eq}}\big)^T \boldsymbol{P}_{j(k)} \big(\boldsymbol{x}(k) - \boldsymbol{x}_{\mathrm{eq}}\big) \tag{27}$$

with $\boldsymbol{x}_{\mathrm{eq}} = \big(\boldsymbol{x}_{\mathrm{eq}1}^T \cdots \boldsymbol{x}_{\mathrm{eq}M}^T\big)^T$.

For a general piecewise constant reference signal changing in a stepwise way based on the Lyapunov function (27) the stability is still guaranteed by shifting the origin to the new equilibrium point.

6. EXPERIMENTAL IMPLEMENTATION

Given two equivalent DC motors (Maxon RE-max29) with the model

$$\dot{n}(t) = -\frac{k_{\mathrm{m}}}{2\pi J R k_{\mathrm{e}}} n(t) + \frac{k_{\mathrm{m}}}{2\pi J R} u_{\mathrm{a}}(t),$$

where $u_{\mathrm{a}}(t)$ is the armature voltage and $n(t)$ is the motor speed in rounds per second which is controlled to a piecewise constant reference speed. The parameter $k_{\mathrm{m}} = 0.0258 \,\frac{\mathrm{Nm}}{\mathrm{A}}$ is the torque constant, $k_{\mathrm{e}} = 6.1667 \,\frac{1}{\mathrm{Vs}}$ is the motor velocity constant, $R = 3.26\,\Omega$ is the total resistance and $J = 8.79\cdot 10^{-6}\,\mathrm{kgm}^2$ is the moment of inertia. The initial motor speed of both DC motors is zero, i.e. $n(0) = 0\,\mathrm{s}^{-1}$. This leads to the model (1) with $\boldsymbol{A}_{\mathrm{p}i} = -23.237$, $\boldsymbol{B}_{\mathrm{p}i} = 143.296$ and $\boldsymbol{C}_{\mathrm{p}i} = 1$ for all $i = \{1, 2\}$.

The discretization interval is set to $h = 0.01\,\mathrm{s}$ such that the effective control update interval is h or a multiple of h depending on the scheduling decision. The control parameters for the control law (8b) are chosen $K_{\mathrm{P}i} = 0.0233$ and $K_{\mathrm{I}i} = 1$ and the factor $\beta_i(k)$ is saturated by $h_{\max_i} = 4\cdot h$ for all $i = \{1, 2\}$. The scheduling parameters of the scheduling law (9) are $\lambda_{\mathrm{e}i} = 1.0$ and $\lambda_{\mathrm{r}i} = 0.1$ for $i = \{1, 2\}$.

Before implementing the controller and scheduler in the processor the stability is verified by Theorem 1. The feasibility of the LMI problem (24) is successfully verified using YALMIP Löfberg [2004] with the SeDuMi solver Sturm [1999].

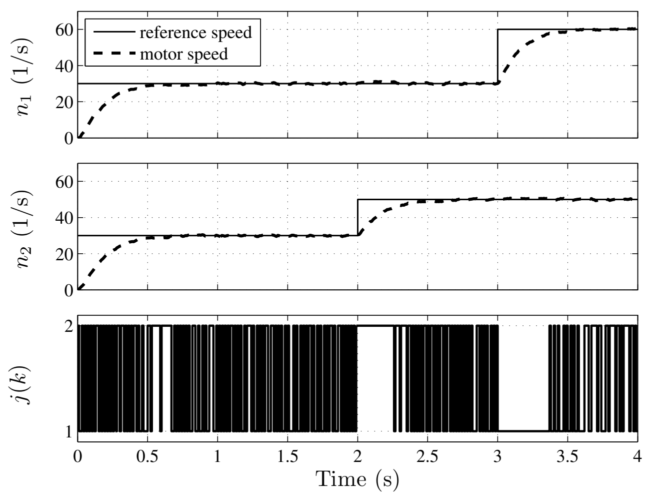

Fig. 3. Implementation results

For the practical implementation a microcontroller NXP LPC2294 is used as the processor. Fig. 3 shows the reference speed and the controlled motor speed for both motors as well es the scheduling index $j(k)$. At the beginning of the experiment both outputs are controlled to a reference speed $r_i(k) = 30\,\mathrm{s}^{-1}$. Therefore the scheduling index switches constantly between 1 and 2. At the time instant $t = 2\,\mathrm{s}$ and $t = 3\,\mathrm{s}$ we can see how the scheduler reacts on the reference change and gives all computation resources to the control task T_2 and T_1, respectively. When both plants are again in the steady state the scheduling index again switches constantly between 1 and 2. Obviously the scheduling is also affected by the measurement noise. This can be seen after 0.5 seconds where more computation resources are given to control task T_2 due to the noise.

7. CONCLUSION AND FUTURE WORK

In this paper we propose an output-based control and scheduling method to achieve set-point tracking using PI control for embedded control systems. A novel PI control design and scheduling method is introduced to realize an efficient usage of the limited resources. For investigating the stability the holistic closed-loop system is modeled as a switched system. Based on a multiple Lyapunov function approach stability can be proven by solving an LMI feasibility problem.

An open question for the proposed approach is the determination of the control and scheduling parameters. Therefore, a codesign method for determining those parameters simultaneously will be investigated in future work. It is also interesting to introduce a performance criterion which is then optimized. Further, integrator windup caused by actuator saturation will be analyzed.

REFERENCES

K.-E. Årzén. A simple event-based PID controller. In *14th World Congress of IFAC*, pages 423–428, 1999.

S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

R. Castañé, P. Martí, M. Velasco, A. Cervin, and D. Henriksson. Resource management for control tasks based on the transient dynamics of closed-loop systems. In *Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 171–182, 2006.

A. Cervin and P. Alriksson. Optimal on-line scheduling of multiple control tasks: A case study. In *Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 141–150, Dresden, Germany, July 2006.

A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Årzén. Feedback-feedforward scheduling of control tasks. *Real-Time Systems*, 23(1-2):Springer Netherlands, July 2002.

A. Cervin, M. Velasco, P. Martí, and A. Camacho. Optimal online sampling period assignment: Theory and experiment. *IEEE Transactions on Control Systems Technology*, 19(4):902–910, 2011.

J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883–1887, 2002.

S. Durand and N. Marchand. Further results on event-based PID controller. In *Proceedings of the 2009 European Control Conference*, pages 1979–1984, 2009.

J. Eker, P. Hagander, and K.-E. Årzén. A feedback scheduler for real-time controller tasks. *Control Engineering Practice*, 8(12):1369–1378, 2000.

M. E. M. Ben Gaid, A. Çela, and Y. Hamam. Optimal real-time scheduling of control tasks with state feedback resource allocation. *IEEE Transactions on Control Systems Technology*, 17(2):309–326, 2009.

D. Görges, M. Izák, and S. Liu. Optimal control of systems with resource constraints. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 1070–1075, 2007.

D. Henriksson and A. Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, pages 4469–4474, 2005.

D. Lehmann, G. A. Kiener, and K. H. Johansson. Event-triggered PI control: Saturating actuators and anti-windup compensation. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 6566–6571, 2012.

B. Lincoln and B. Bernhardsson. LQR optimization of linear system switching. *IEEE Transactions on Automatic Control*, 47(10):1701–1705, 2002.

J. Löfberg. YALMIP : a toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium*, pages 284–289, 2004.

P. Martí, C. Lin, S. A. Brandt, M. Velasco, and J. M. Fuertes. Optimal state feedback based resource allocation for resource-constrained control tasks. In *Proceedings of the 25th IEEE International Real-Time Systems Symposium*, pages 161–172, 2004.

P. Martí, C. Lin, S. A. Brandt, M. Velasco, and J. M. Fuertes. Draco: Efficient resource management for resource-constrained control tasks. *IEEE Transactions on Computers*, 58(1):90–105, 2009.

H. Rehbinder and M. Sanfridson. Integration of off-line scheduling and optimal control. In *Proceedings of the 12th Euromicro Conference on Real-Time Systems*, pages 137–143, 2000.

S. Reimann, S. Al-Areqi, and S. Liu. Output-based control and scheduling codesign for control systems sharing a limited resource. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 4042–4047, 2013.

D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin. On task schedulability in real-time control systems. In *17th IEEE Real-Time Systems Symposium*, pages 13–21, 1996.

J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones (updated for version 1.03). *Optimization Methods and Software*, 11-12:625–653, 1999.

U. Tiberi, J. Araújo, and K. H. Johansson. On event-based PI control of first-order processes. In *IFAC Conference on Advances in PID Control*, pages 448–453, Brescia, Italy, March 2012.