Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 10301–10306

Settling Time Optimization in Wire Bonder Systems via Extremum-Seeking Control [1]

Wouter Weekers * Dragan Kostić ** Alessandro Saccon * Nathan van de Wouw *

* *Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven, The Netherlands (e-mail: {w.weekers, a.saccon, n.v.d.wouw}@tue.nl).*
** *ASMPT, Center of Competency, Beuningen, The Netherlands (e-mail: dragan.kostic@asmpt.com).*

Abstract: Adequate tuning of control laws is essential for high positioning accuracy, large system throughput, and reliability in high-end mechatronic and robotic systems. However, a population of such systems generally shows slight variations in dynamic responses due to, e.g., manufacturing tolerances, different disturbance situations, or position-dependent dynamics. Given the time-consuming nature of controller design, even by experienced control engineers, typically just one control law is designed for the whole system population based on worst-case bounds on variations in dynamic responses, resulting in a loss of individual system performance. The main contribution of this paper is the development of an automated controller tuning approach, based on extremum-seeking control, for settling time optimization via individual controller tuning. While other automated controller tuning methods exist, the developed approach allows inclusion of closed-loop stability and robustness constraints based solely on non-parametric frequency-response measurements of open-loop plant dynamics, and therewith directly optimizes transient system performance in a purely data-based manner. The proposed approach has been applied in simulation in an industrial case study for settling time optimization in point-to-point motions of a wire bonder system. In this case study, the effectiveness of the approach has been shown by achieving significant performance increases of 39.4% and 40.6% compared to controllers designed by experienced control engineers using manual loop-shaping techniques and a frequency-based auto-tuner, respectively, without needing manual tuning effort.

Keywords: extremum seeking, data-driven control, transient performance, feedback control

1. INTRODUCTION

Motion systems used in high-end mechatronic and robotic systems, such as, e.g., semiconductor manufacturing equipment, require well-tuned control laws to achieve high positioning accuracy, large system throughput, and reliability. In production facilities, often a large population of the same type of motion system is used for the manufacturing process; in this paper, we consider wire bonder equipment for the back-end semiconductor industry as a use case. These systems generally show slight variations in their dynamic responses due to manufacturing tolerances, different disturbance situations, or position-dependent dynamics. The performance of these systems is typically characterized by their ability to accurately execute quick point-to-point motions, and to achieve fast settling times while adhering to maximal error bounds after settling; both to ensure high system throughput. However, the variation in the dynamic responses of these systems results in differences in closed-loop system performance using the same controller. To complicate the matter, the relation between controller parameters and the system performance is often not directly clear due to, e.g., unknown disturbance situations. Due to the time-consuming and costly nature of controller design, even by experienced control engineers, typically only one control law is designed for the whole population of systems, based on worst-case bounds on differences in responses. Such conservative design approach results in a loss of individual system performance.

Automated, data-driven methods for controller tuning exist and allow for faster design of individually tuned controllers, without the need for human intervention. Moreover, such dedicated controller tuning could outperform the conservative controller designed based on worst-case bounds on differences in system responses. Examples of data-driven controller tuning techniques include iterative-learning control (ILC) (Bristow et al., 2006; Ahn et al., 2007), iterative feedback tuning (IFT) (Hjalmarsson et al., 1998; Hjalmarsson, 2002), and extremum-seeking control (ESC) (Krstić and Wang, 2000; Teel and Popović, 2001; Tan et al., 2010). These data-driven methods are similar in the sense that they iteratively update controller parameters to minimize a cost function. In ILC, a given motion task is repeated to iteratively update feedforward controller parameters via feedback in the iteration domain. In IFT, feedback controller parameters are updated iter-

[1] This publication is part of the project Digital Twin with project number P18-03 of the research program TTW Perspective which is (partly) financed by the Dutch Research Council (NWO).

10.1016/j.ifacol.2023.10.946

atively using gradient-based methods, and the required gradient is estimated by three offline experiments per iteration. Similarly, in ESC, feedback or feedforward controller parameters are (typically) updated iteratively using gradient-based methods as well (although non-gradient-based methods exist too, see, e.g., Khong et al. (2013b)). However, in ESC, the required gradient is estimated online via small controller parameter perturbations for each cost function evaluation. In Khong et al. (2016), it is shown that ILC can also be reformulated in an ESC framework.

In this paper, we will focus on automated tuning of feedback controllers, as typically seen in IFT and ESC. Although these methods have been successfully applied for controller tuning in several applications (see e.g. (Hamamoto et al., 2003; McDaid et al., 2012) for IFT, and (Killingsworth and Krstić, 2006; Killingsworth et al., 2007; Chen et al., 2018; Hazeleger et al., 2021) for ESC), the performance in these applications is typically measured in terms of an error norm. Here, we are instead interested in settling time optimization, which is not guaranteed by the minimization of an error norm. In fact, the settling time is a discontinuous function of feedback controller parameters, and thus standard gradient-based IFT and ESC techniques cannot be employed for controller tuning in this context. Furthermore, closed-loop stability and its robustness to perturbations of open-loop system dynamics are important requirements in feedback controller design, which cannot be guaranteed during the iterative steps of standard IFT and ESC methods. In this paper, we therefore develop an approach for automated tuning of parameters of a given feedback controller structure, that optimizes the settling time and guarantees closed-loop stability and its robustness.

The main contribution of this paper is the development of such approach based on global ESC (Khong et al., 2013b). In contrast to the steady-state performance optimization problems typically encountered in ESC (Tan et al., 2010), we thus optimize transient system performance. To deal with the discontinuous dependence of the settling time on controller parameters, the proposed method is based on the DIRECT algorithm (Jones et al., 1993; Jones and Martins, 2021), a global, non-gradient-based optimization algorithm previously employed in the context of ESC in Khong et al. (2013a,b). The approach guarantees closed-loop stability, and allows introduction of robustness constraints, based solely on non-parametric frequency-response measurements of open-loop plant dynamics. It therefore automatically tunes controller parameters in a purely data-based manner. A second contribution involves application of the proposed approach to transient performance optimization of point-to-point motions in a wire bonder system, and a comparison of the performance to that obtained by dedicated controllers designed by experienced control engineers using manual loop shaping, as well as by a frequency-based tuning method used in industry.

The remainder of this paper is organized as follows. In Section 2, the problem formulation is presented. In Section 3, the ESC-based approach for automated tuning of feedback controllers is described. Numerical simulations are presented in Section 4. In Section 5, we present concluding remarks.

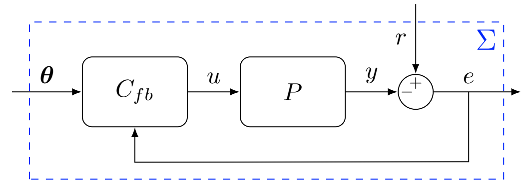

Fig. 1. Closed-loop system Σ, consisting of the feedback interconnection of a plant P and a controller $C_{fb}(\boldsymbol{\theta})$.

2. PROBLEM DESCRIPTION

In this section, we provide a system description and, subsequently, formulate the performance optimization problem to be solved.

2.1 System description

For sake of ease of presentation, we consider a linear, single-input-single-output plant P, see Figure 1, reflecting the dynamics of a motion system. It is described by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \tag{1a}$$
$$y(t) = \mathbf{c}^{\mathsf{T}}\mathbf{x}(t) + du(t), \tag{1b}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_{\mathbf{x}}}$ is the plant state, $t \in \mathbb{R}$ is time, $u(t) \in \mathbb{R}$ is the system input, $y(t) \in \mathbb{R}$ is the plant output, and $\mathbf{A} \in \mathbb{R}^{n_{\mathbf{x}} \times n_{\mathbf{x}}}$, $\mathbf{b} \in \mathbb{R}^{n_{\mathbf{x}}}$, $\mathbf{c} \in \mathbb{R}^{n_{\mathbf{x}}}$, and $d \in \mathbb{R}$ are the system matrices. The system input $u(t)$ is generated by the linear feedback controller $C_{fb}(\boldsymbol{\theta})$ with tunable parameters $\boldsymbol{\theta} \in \mathbb{R}^{n_{\boldsymbol{\theta}}}$, i.e., the plant input is generated by

$$\dot{\mathbf{z}}(t) = \mathbf{E}(\boldsymbol{\theta})\mathbf{z}(t) + \mathbf{f}(\boldsymbol{\theta})e(t), \tag{2a}$$
$$u(t) = \mathbf{g}^{\mathsf{T}}(\boldsymbol{\theta})\mathbf{z}(t) + h(\boldsymbol{\theta})e(t), \tag{2b}$$

where $\mathbf{z}(t) \in \mathbb{R}^{n_{\mathbf{z}}}$ is the controller state, and $\mathbf{E}(\boldsymbol{\theta}) \in \mathbb{R}^{n_{\mathbf{z}} \times n_{\mathbf{z}}}$, $\mathbf{f}(\boldsymbol{\theta}) \in \mathbb{R}^{n_{\mathbf{z}}}$, $\mathbf{g}(\boldsymbol{\theta}) \in \mathbb{R}^{n_{\mathbf{z}}}$, and $h(\boldsymbol{\theta}) \in \mathbb{R}$ are the controller system matrices, that are continuous in $\boldsymbol{\theta}$. In (2), $e = r - y$ is the servo error during tracking of the time-varying reference signal $r(t)$. The feedback interconnection of (1) and (2) results in the multi-input-single-output closed-loop system Σ, with inputs $\boldsymbol{\theta}$ and r, and output e, as illustrated in Figure 1. In the context of ESC, $\boldsymbol{\theta}$ denotes the to-be-optimized parameters, and the output e is related to the performance measure of interest. We pose the following assumption on the closed-loop system Σ.

Assumption 1. For a given frequency-dependent robustness weight W_p, there exists a nonempty, compact set $\boldsymbol{\Theta} \subset \mathbb{R}^{n_{\boldsymbol{\theta}}}$ such that for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the closed-loop system (1)-(2) is asymptotically stable (AS) and satisfies

$$|W_p(j\omega)S(j\omega)| < 1, \, \forall \omega \tag{3}$$

with $S := (1 + PC_{fb}(\boldsymbol{\theta}))^{-1}$ the sensitivity function.

This is a natural assumption for feedback controller design, where closed-loop stability and its robustness (as determined by satisfaction of (3), since choosing, e.g., $W_p(j\omega) = \frac{1}{2}$ for all ω implies gain and phase margins of at least 2 and 30°, respectively) are standard requirements (Skogestad and Postlethwaite, 2005, Section 2.4.3).

2.2 Performance optimization problem description

We consider point-to-point motion tasks defined on the time interval $[0, T]$, see Figure 2, each consisting of

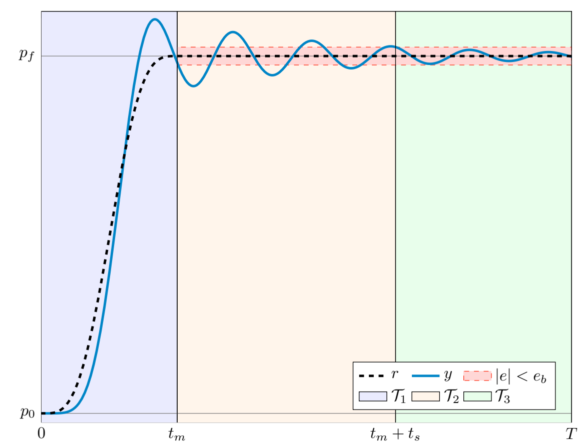

Fig. 2. Motion profile r, from an initial position p_0 to a final position p_f, with motion time window $\mathcal{T}_1$, settling time window $\mathcal{T}_2$, and process time window $\mathcal{T}_3$.

- a motion time window $\mathcal{T}_1$ of length t_m, during which the system is supposed to track a time-varying reference signal from an initial position p_0 to a final position p_f or vice versa;
- an idle time window $\mathcal{T}_2$ of length t_s, during which the system output e converges, satisfying a desired error bound e_b at the end of the interval, i.e.
$$|e(t_m + t_s, \boldsymbol{\theta})| \le e_b, \tag{4}$$
where $e(t, \boldsymbol{\theta})$ denotes the time evolution of the system output e for a given $\boldsymbol{\theta}$;
- a process time window $\mathcal{T}_3$ of length $t_p = T - t_s - t_m$, during which accurate positioning is crucial to perform the machine operation and e should respect the bound $|e| \le e_b$.

Combining these motion tasks results in a $2T$-repetitive motion profile r. Note that t_m and t_p are typically determined by the used setpoint profile and the time needed for the machine operation, respectively. Consequently, the production throughput is maximized by minimizing the settling time
$$t_s(\boldsymbol{\theta}) := \max\{s \in [0,\, T - t_m] : |e(t_m + s, \boldsymbol{\theta})| > e_b\}, \tag{5}$$
as this in turn would enable to decrease T and hence would increase the throughput. This leads to the constrained optimization problem:
$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \; t_s(\boldsymbol{\theta}) \tag{6}$$
$$\text{s.t. } |e(t, \boldsymbol{\theta})| \le e_b, \;\; \forall t \in \mathcal{T}_3. \tag{7}$$
The constraint $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ in (6) comes from the fact that $\boldsymbol{\theta}$ are parameters of a feedback controller, and thus a wrong choice of parameters could result in an unstable and/or non-robust controller setting. Solving the optimization problem (6)-(7) numerically, however, is difficult since t_s is not a continuous function of $\boldsymbol{\theta}$: an arbitrarily small change in $\boldsymbol{\theta}$ could cause t_s to jump discontinuously, as illustrated in Figure 3. Since optimizing a discontinuous function is generally difficult, we reformulate the optimization problem (6)-(7) in the next section to minimize t_s.

3. A CASCADED EXTREMUM-SEEKING APPROACH

In this section, we describe our proposed approach to solve the optimization problem (6)-(7) in an online, and purely data-based fashion. To this end, first, we reformulate this discontinuous optimization problem as two cascaded optimization problems with continuous cost functions. Next, we describe the global ESC approach that we use to solve them.

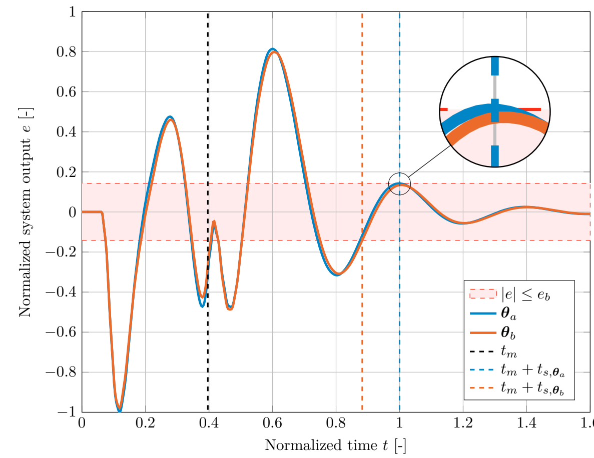

Fig. 3. Small changes in the output e as a result of small changes in parameters $\boldsymbol{\theta}$ from $\boldsymbol{\theta}_a$ to $\boldsymbol{\theta}_b$ can result in changes in the violation of the error bound e_b, causing a discontinuous jump in t_s from $t_{s,\boldsymbol{\theta}_a}$ to $t_{s,\boldsymbol{\theta}_b}$.

3.1 Optimization problem reformulation

Let us introduce an extra optimization variable τ, and reformulate optimization of (6)-(7) as follows:
$$\min_{\tau \in [0,\, T - t_m]} \; \tau \tag{8}$$
$$\text{s.t. } J^*(\tau) \le e_b, \tag{9}$$
where
$$J^*(\tau) := \min_{\boldsymbol{\theta} \in \Theta} \; J(\tau, \boldsymbol{\theta}) \tag{10}$$
with $J(\tau, \boldsymbol{\theta}) := \sup_{t \ge 0} \; |w(t,\, \tau) e(t, \boldsymbol{\theta})|$, and
$$w(t,\, \tau) := \begin{cases} 1, \text{ if } t \in [t_m + \tau,\, T], \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$
The reformulated optimization problem (8)-(10) forms a new, cascaded optimization problem. This problem solves the discontinuous optimization problem (6)-(7) by solving the continuous upper-level optimization problem (8)-(9), for which the continuous lower-level optimization problem (10) is repeatedly solved to be able to evaluate the constraint (9). The problem reformulation allows minimization of t_s as follows. The combination of the indicator function w in (11) and the supremum in the definition of the cost J ensures the optimal parameters $\boldsymbol{\theta}^*$ minimize the peak servo error over the time-interval $t \in [t_m + \tau,\, T]$. Furthermore, (9) ensures that for the optimal parameter τ^* this peak error is within the required bounds, meaning the servo error has settled within the required bounds at time $t_m + \tau^*$. Finally, the constraint $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ in (10) guarantees that optimal parameters $\boldsymbol{\theta}^*$ result in a robustly stable closed-loop system.

3.2 The cascaded optimization algorithm

Standard gradient-based ESC methods cannot be applied to solve the lower-level optimization problem (10), since it is 1) non-smooth, due to the supremum and absolute value in the definition of the cost J, and 2) likely not even convex, thus potentially has multiple local minima. However, despite its non-smooth and non-convex nature, optimization of (10) is less challenging than optimization of the discontinuous function (6), due to its continuity. To deal with the non-smooth and non-convex nature of (10), we employ the DIRECT algorithm, a global optimization algorithm, with some adaptations aimed at 1) reducing the number of costly cost function evaluations the original algorithm needs, and 2) the inclusion of the feasibility constraint $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. The DIRECT algorithm is a deterministic, non-smooth, non-convex optimization algorithm, that minimizes a given cost function by repeated sampling from the parameter space $\Omega := \{\boldsymbol{\theta} = [\theta_1, \dots, \theta_{n_{\boldsymbol{\theta}}}]^\mathsf{T} : \theta_i \in [\underline{\theta}_i, \overline{\theta}_i],\ i = 1, \dots, n_{\boldsymbol{\theta}}\}$, and subdividing it into hyper-rectangles. Here, $\underline{\theta}_i$ and $\overline{\theta}_i$ denote lower and upper bounds on parameter θ_i, respectively. The DIRECT algorithm has previously been used in the context of ESC and ILC in Khong et al. (2013a,b) and Khong et al. (2016).

Algorithm 1 Modified DIRECT algorithm

1: **Input:** $\Omega = \{\boldsymbol{\theta} = \left[\theta_1, \dots, \theta_{n_{\boldsymbol{\theta}}}\right]^\mathsf{T} : \theta_i \in [\underline{\theta}_i, \overline{\theta}_i],\ i = 1, \dots, n_{\boldsymbol{\theta}}\}$, cost function J, maximum number of function evaluations N, maximum number of iterations M, error bound e_b, and $\varepsilon > 0$
2: **Output:** optimal parameters $\boldsymbol{\theta}^*$ and corresponding value J^*
3: **procedure** DIRECT(Ω, J, N, M, e_b, ε)
4: Let $\boldsymbol{\theta}_1$ be the center point of Ω
5: $J_1 \leftarrow J(\boldsymbol{\theta}_1)$. Perform a motion experiment if $\boldsymbol{\theta}_1 \in \boldsymbol{\Theta}$, otherwise assign a penalty value $\tilde{J}$.
6: $l \leftarrow 0$, $k \leftarrow 1$, $J^* \leftarrow J_1$
7: **while** $J^* > e_b$ **and** $k < N$ **and** $l < M$ **do**
8: Identify set $\mathcal{S}$ of all indices $m \in [1, \dots, k]$ for which $\exists L > 0$ such that
$J_m - Ld_m \leq J_i - Ld_i,\ \forall i = 1, \dots, k$
$J_m - Ld_m \leq J^* - \varepsilon|J^*|$.
Here d_m denotes the longest side length of the mth hyper-rectangle, and $\varepsilon > 0$ is small.
9: **for** every $m \in \mathcal{S}$ **do**
10: Let q be the direction along which the mth hyper-rectangle has side length d_m. In case the mth hyper-rectangle has multiple sides of length d_m, q is the direction with the least number of total subdivisions.
11: Let $\boldsymbol{\theta}_m$ be the center point of the mth hyper-rectangle, and $\boldsymbol{\sigma}_q$ the unit vector along q.
12: $\delta \leftarrow d_m/3$
13: $J_{k+1} \leftarrow J(\boldsymbol{\theta}_m - \delta\boldsymbol{\sigma}_q)$ ▷ See line 5
14: $J_{k+2} \leftarrow J(\boldsymbol{\theta}_m + \delta\boldsymbol{\sigma}_q)$ ▷ See line 5
15: Trisect the mth hyper-rectangle along q.
16: $k \leftarrow k + 2$
17: **end for**
18: $l \leftarrow l + 1$, $J^* \leftarrow \min_{i=1,\dots,k} J_i$
19: Set $\boldsymbol{\theta}^*$ to be the parameters associated with J^*.
20: **end while**
21: **end procedure**

We describe our modified version of the DIRECT algorithm in Algorithm 1. For a detailed description of the original algorithm, and visualizations thereof, we refer to Jones et al. (1993); Jones and Martins (2021). The following three modifications to the original DIRECT algorithm can be recognized in Algorithm 1:

1) line 7: the algorithm is terminated before utilizing the full function-evaluation budget N or iteration budget M when $J^* \leq e_b$. This modification prevents unnecessary function evaluations in case (9) is satisfied and τ can be decreased.
2) line 8: the size of the mth hyper-rectangle is measured by its longest side length, instead of its center-to-vertex distance. This modification results in a stronger bias for local search (Gablonsky, 2001), resulting in less function evaluations necessary to find parameters $\boldsymbol{\theta}$ such that (9) is satisfied.
3) lines 9-17: the mth hyper-rectangle is sampled and trisected along a single side, instead of along all sides of length d_m. This modification accelerates convergence according to (Jones, 2001).

Besides these modifications, a feasibility check for parameters $\boldsymbol{\theta}$ is performed before implementing $C_{fb}(\boldsymbol{\theta})$ and conducting a motion experiment to evaluate the cost J (indicated by $J(\cdot)$ in Algorithm 1). Since we presume that a parametric model of the plant is not available, we perform a data-based analysis of closed-loop stability based on frequency response function (FRF) data of the open-loop plant P ($u \rightarrow y$, cf. Figure 1). Given the controller parameters $\boldsymbol{\theta}$, we evaluate $C_{fb}(\boldsymbol{\theta})$ for the same frequency range as P, and combine the FRF data of the two to draw the Nyquist plot of the open-loop system using linear interpolation between data points. The net amount of encirclements of the critical point is determined automatically by evaluating the number of up- and downward negative real axis crossings to the left of the critical point. Similarly, the magnitude of the sensitivity function S is computed on the basis of FRF data of $C_{fb}(\boldsymbol{\theta})$ and P, and linearly interpolated to check if (3) is satisfied. The new, to-be-evaluated, parameters $\boldsymbol{\theta}$ are only marked as feasible in case the closed-loop system with controller $C_{fb}(\boldsymbol{\theta})$ is concluded to be asymptotically stable (AS), and (3) is satisfied, i.e., if $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. In case of infeasible $\boldsymbol{\theta}$, the controller $C_{fb}(\boldsymbol{\theta})$ is not implemented, and no motion experiment is performed for this (unstable and/or non-robust) controller setting. Instead, a (high) surrogate penalty function value $\tilde{J}$ is assigned to J according to the constraint handling approach proposed by Gablonsky (2001, Section 3.4.3). This modification essentially turns the constrained optimization problem (10) into the unconstrained problem

$$J^*(\tau) = \min\ I_{\boldsymbol{\Theta}}(\boldsymbol{\theta})J(\tau, \boldsymbol{\theta}) + (1 - I_{\boldsymbol{\Theta}}(\boldsymbol{\theta}))\tilde{J} \tag{12}$$

with indicator function $I_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ equal to one if $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and zero otherwise, ensuring closed-loop stability and its robustness during iterations of the DIRECT algorithm.

Given $J^*(\tau)$ for a fixed value of τ, and a desired tolerance τ^{tol}, we update τ with a bisection search aimed at solving the upper-level optimization problem (8)-(9). This bisection search repeatedly updates the lower and upper bound of the search interval, denoted by $\underline{\tau}$ and $\overline{\tau}$ respectively, based on satisfaction of (9), as illustrated in Algorithm 2. The (initial) values of $\underline{\tau}$, $\overline{\tau}$, and e_b follow from the application context. The combination of the closed-loop system (1)-(2) and the modified DIRECT algorithm (Algorithm 1) forms a sampled-data ESC loop in the spirit of Khong et al. (2013a,b). Addition of the bisection search (Algorithm 2) to this sampled-data ESC loop results in the cascaded ESC loop illustrated in Figure 4. This cascaded ESC loop only requires specification of lower and upper bounds on each of the optimization parameters τ and θ_i, $i = 1, \dots, n_{\boldsymbol{\theta}}$, and the tuning of a single parameter ε of the DIRECT algorithm (whose value does not have a large influence on algorithm performance (Jones et al., 1993)). By only

Algorithm 2 Bisection search

Inputs: initial lower bound $\underline{\tau}$, initial upper bound $\overline{\tau}$, tolerance τ^{tol}, and error bound e_b
Output: found optimum τ^*
procedure BISECT($\underline{\tau}$, $\overline{\tau}$, τ^{tol}, e_b)
  while $|\underline{\tau} - \overline{\tau}| > \tau^{tol}$ **do**
    $\tau \leftarrow (\underline{\tau} + \overline{\tau})/2$
    Solve (10) ▷ See Algorithm 1
    if $J^*(\tau) < e_b$ **then**
      $\overline{\tau} \leftarrow \tau$
    else
      $\underline{\tau} \leftarrow \tau$
    end if
  end while
  $\tau^* \leftarrow (\underline{\tau} + \overline{\tau})/2$
end procedure

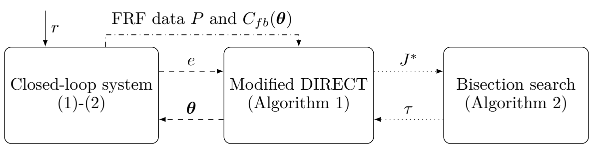

Fig. 4. The cascaded ESC loop. Arrows indicate data exchange at function evaluations (dash dotted), motion experiments (dashed), or bisection updates (dotted).

requiring lower and upper bounds, the algorithm is easier to tune than standard gradient-based ESC methods, which typically require tuning of multiple parameters (e.g, the dither amplitude, waiting time, optimizer gain, and initial parameter values). Furthermore, the DIRECT algorithm allows global optimization, whereas gradient-based ESC methods are local optimizers (Tan et al., 2010).

4. INDUSTRIAL CASE STUDY

In this section, we validate the effectiveness of the proposed cascaded ESC approach in a simulation study. As a use case we consider the plant P to be the x-stage of a wire bonder system. A wire bonder typically consists of a motion stage (Figure 5) that moves along its x-, y-, and z-axes to make wired interconnections between a semiconductor die and its packaging. To ensure satisfactory quality of the finished integrated circuit, the servo error should not violate a given bound e_b at the moment a bond is made. Furthermore, to ensure high system throughput, the servo error should converge to within this bound as quickly as possible, i.e., the settling time t_s should be minimized. The bound e_b is typically in the (sub-)micrometer range, while the settling time t_s should typically be in the order of milliseconds. Our goal is to minimize t_s by appropriate tuning of the parameters of a fixed feedback controller structure designed based on engineering insight. Here, we opt for a proportional-integral-derivative (PID) controller structure combined with an additional low-pass filter, whose transfer function is given by $C_{fb}(s;\boldsymbol{\theta}) = 2\pi f(k_d s^2 + k_p s + k_i)/(s(s + 2\pi f))$. The proportional gain k_p, integral gain k_i, derivative gain k_d, and low-pass filter cut-off frequency f serve as the optimization variables which we aim optimize using the proposed ESC approach, i.e., $\boldsymbol{\theta} := [k_p \; k_i \; k_d \; f]^\mathsf{T}$. The reference motion for the wire bonder is given by seventh-order polynomial trajectories with idle periods of several milliseconds in-between to allow for the servo error to converge, as shown in Figure 2. The trajectories consist of

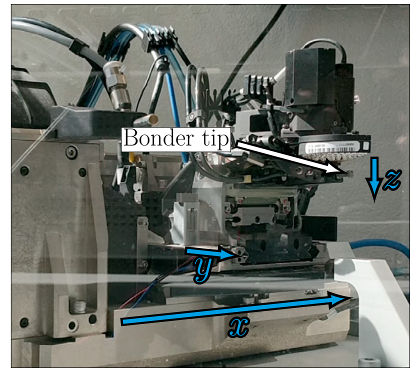

Fig. 5. Isolated wire bonder motion stage.

displacements of several millimeters along the x-axis and back. To ensure sufficient robustness margin, we choose $W_p(j\omega) = 1/(20\log_{10}(6))$, $\forall\omega$. We use the MATLAB implementation of the DIRECT algorithm from Finkel (2004) containing the constraint handling approach described by Gablonsky (2001, Section 3.4.3) as a basis, and adapt it to include the modifications described in Section 3.

We perform numerical validation of the cascaded ESC approach using a (confidential) multibody model of the motion stage shown in Figure 5 used by the manufacturer for initial validation of novel control techniques. In this model, interconnections between the motion axes are modeled using springs and dampers, and nonlinear effects of rotations, and time delays in the control loop of the real system are included. We perform the feasibility check described in Section 3 for numerous random parameter combinations, to get a rough a priori estimate of $\boldsymbol{\Theta}$, and choose $\underline{\theta}_i$ and $\overline{\theta}_i$, $i = 1, \ldots, n_{\boldsymbol{\theta}}$, such that this estimate is contained in Ω. Furthermore, we choose the lower and upper bounds on τ based on engineering insight, and set the desired tolerance τ^{tol} equal to the sample time of the real motion stage. We compare the performance of the optimized controller to controllers designed by experienced control engineers using standard loop-shaping (LS) techniques, and a frequency-based auto-tuner (FBA) that aims to achieve the largest amount of low-frequency error suppression given an upper bound on the sensitivity function designed by the engineer. Both controllers are tuned based on the same FRF data as those used in the feasibility check described in Section 3. Simulation parameter values and results [2] are summarized in Table 1. Comparing the servo error obtained with the controller tuned using the ESC approach to those obtained with the LS and FBA controllers shows significant reductions in t_s of 39.4% and 40.6%, respectively, as illustrated by Figure 6. With the number of simulated motion experiments reported in Table 1, and each reference motion lasting in the order of tens of milliseconds, the total optimization time is only in the order of 15 minutes, showing feasibility of the proposed approach in practice.

5. CONCLUSION

We developed an approach for data-driven optimization of transient system performance based on constrained global extremum-seeking control. The approach minimizes

[2] Results are normalized for confidentiality reasons.

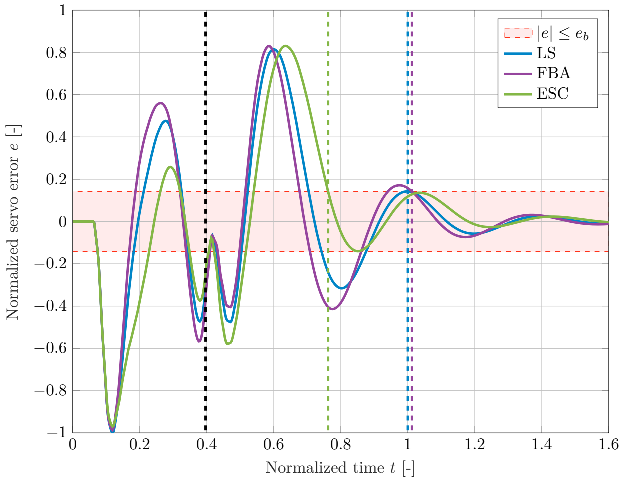

Fig. 6. Simulated servo error $e(t)$ for the controllers obtained with manual LS (LS), the FBA (FBA), and the ESC approach (ESC). Time instances t_m (black) and $t_m + t_s$ (colored) are indicated by vertical dashed lines. For ESC, t_s is significantly reduced compared to LS (39.4%) and FBA (40.6%).

Table 1. Simulation parameters and results.

	ESC	LS	FBA
Normalized settling time	0.366	0.604	0.616
Max. number of function evaluations per iteration N			5,000
Max. number of iterations M			5,000
Number of bisection search iterations used			8
Total number of function evaluations used			28,220
of which required simulation of motion experiments			3,549

the settling time in point-to-point motions by automated tuning of feedback controller parameters using bisection search and the DIRECT optimization algorithm. Closed-loop stability and its robustness are guaranteed using constraints based solely on non-parametric frequency-response function measurements. In an industrial case study, the effectiveness of the approach is validated in simulation by applying it to settling time optimization in an industrial wire bonder system. The controller optimized using the proposed approach achieves significant performance increases of 39.4% and 40.6%, respectively, compared to time-consuming dedicated controller tuning by experienced control engineers using manual loop-shaping techniques and a frequency-based auto-tuner used in industry. Future research will focus on experimental validation of the approach on a real wire bonder, and extending it to minimize position-dependent performance differences to ensure uniform performance.

REFERENCES

Ahn, H.S., Chen, Y., and Moore, K.L. (2007). Iterative learning control: brief survey and categorization. *IEEE Trans. Syst., Man, Cybern., Part C (Applications Rev.)*, 37(6), 1099–1121.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE Control Syst. Mag.*, 26(3), 96–114.

Chen, Q., Tan, Y., Li, J., and Mareels, I. (2018). Decentralized PID control design for magnetic levitation systems using extremum seeking. *IEEE Access*, 6, 3059–3067.

Finkel, D.E. (2004). Research and codes. URL `https://ctk.math.ncsu.edu/Finkel_Direct/`. (Accessed Sep. 23, 2022).

Gablonsky, J.M.X. (2001). *Modifications of the DIRECT algorithm*. PhD thesis, North Carolina State University.

Hamamoto, K., Fukuda, T., and Sugie, T. (2003). Iterative feedback tuning of controllers for a two-mass-spring system with friction. *Control Eng. Pract.*, 11(9), 1061–1068.

Hazeleger, L., Beerens, R., and Van de Wouw, N. (2021). Proportional–integral–derivative-based learning control for high-accuracy repetitive positioning of frictional motion systems. *IEEE Trans. Control Syst. Technol.*, 29(4), 1652–1663.

Hjalmarsson, H. (2002). Iterative feedback tuning - an overview. *Int. J. Adapt. Control Signal Process.*, 16(5), 373–395.

Hjalmarsson, H., Gevers, M., Gunnarson, S., and Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE Control Syst. Mag.*, 18(4), 26–41.

Jones, D.R. (2001). DIRECT global optimization algorithm. In C.A. Floudas and P.M. Pardalos (eds.), *Encyclopedia of Optimization*, 431–440. Springer US, Boston.

Jones, D.R. and Martins, J.R.R.A. (2021). The DIRECT algorithm: 25 years later. *J. Glob. Optim.*, 79(3), 521–566.

Jones, D.R., Perttunen, C.D., and Stuckman, B.E. (1993). Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.*, 79(1), 157–181.

Khong, S.Z., Nešić, D., and Krstić, M. (2016). Iterative learning control based on extremum seeking. *Automatica*, 66, 238–245.

Khong, S.Z., Nešić, D., Manzie, C., and Tan, Y. (2013a). Multidimensional global extremum seeking via the DIRECT optimisation algorithm. *Automatica*, 49(7), 1970–1978.

Khong, S.Z., Nešić, D., Tan, Y., and Manzie, C. (2013b). Unified frameworks for sampled-data extremum seeking control: global optimisation and multi-unit systems. *Automatica*, 49(9), 2720–2733.

Killingsworth, N.J., Aceves, S.M., Flowers, D.L., and Krstić, M. (2007). Extremum seeking tuning of an experimental HCCI engine combustion timing controller. In *2007 Am. Control Conf.*, 3665–3670. IEEE.

Killingsworth, N.J. and Krstić, M. (2006). PID tuning using extremum seeking: online, model-free performance optimization. *IEEE Control Syst. Mag.*, 26(1), 70–79.

Krstić, M. and Wang, H.H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4), 595–601.

McDaid, A.J., Aw, K.C., Haemmerle, E., and Xie, S.Q. (2012). Control of IPMC actuators for microfluidics with adaptive "online" iterative feedback tuning. *IEEE/ASME Trans. Mechatronics*, 17(4), 789–797.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable feedback control: analysis and design*. John Wiley & Sons Ltd, Chichester, second edition.

Tan, Y., Moase, W.H., Manzie, C., Nešić, D., and Mareels, I.M.Y. (2010). Extremum seeking from 1922 to 2010. In *Proc. 29th Chin. Control Conf.*, 14–26. IEEE.

Teel, A.R. and Popović, D. (2001). Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In *Proc. 2001 Am. Control Conf.*, volume 3, 2394–2399. IEEE.