PID-based learning control for performance optimization of repetitive motion systems with friction

Sekandari, H.

Award date:
2019

University of Technology Eindhoven
Department of Mechanical Engineering
Dynamics and Control group

Master Thesis Report

PID-based learning control for performance optimization of repetitive motion systems with friction

DC2019.095

Supervisor
prof. dr. ir. N. van de Wouw

Coaches
ir. L. Hazeleger
ir. R. Beerens

Author
H. Sekandari 0821565

Eindhoven, November 6, 2019

Acknowledgements

Throughout this project I have received a great deal of support and assistance. I would first like to express my gratitude to my supervisor, prof. dr. ir. N. van de Wouw, who gave me the opportunity to work on this very promising and interesting project, his critical vision and good suggestions during our meetings that give me more insight in understanding of the research topic. Especially, I want to acknowledge my coaches, ir. L. Hazeleger and ir. R. Beerens, for their valuable guidance, ongoing support, feedback, patience in teaching me, excellent cooperation, critical view, time and effort they put in this project. My coach, ir. L. Hazeleger, I want to thank u, for your critical view and your valuable discussions during our bi-weekly meetings, which give me more insight in solving the issues that I faced and thereby becoming a better engineer. My other coach, ir. R. Beerens, I want to thank u, for your positive attitude and contribution towards me and this project, as well as your equanimous manner of explanations. Finally, I want to thank all the students in the DCT-lab for providing happy distraction to rest my mind outside of my research during our exciting table soccer games.

Abstract

Many industrial motion systems that perform repetitive positioning tasks suffer from frictional effects, limiting the achievable system performance. Classical PID control is employed in the vast majority of industrial motion systems with dry friction, motivated by the intuitive, easy design and tuning rules available. However, PID control suffers from severe system performance limitations in terms of position accuracy and settling time. Namely, undesirable limit cycling (hunting) behaviour occurs when integral control is applied in systems with unknown *Stribeck friction*. The system's performance is highly dependent on its particular friction characteristic, which is generally uncertain or unknown, and may be time- and position-dependent. To achieve optimal system performance in terms of improved setpoint positioning accuracy and enhanced transient response by minimizing overshoot for frictional motion systems in a repetitive motion setting, a PID-based controller with a *time-varying* learning integrator gain is proposed. The time-varying integrator gain design is adaptively tuned using a *model-free* sampled-data extremum seeking control approach. The advantage of the proposed approach is that it does not rely on prior knowledge of the friction characteristics. The working principle and effectiveness of the proposed approach are demonstrated by means of a simulation study and experimental application on an industrial high-precision positioning stage setup of a high-end electron microscope.

Contents

CHAPTER 1

Introduction

1.1 Control of frictional motion systems

Many high-tech motion control systems that perform repetitive tasks, such as robotic arms, printing systems, pick and place machines, and the manipulation stage of electron microscopes (see Figure 1.1) suffer from frictional effects, which limit the achievable system performance. Friction is described as the resistance to motion when two surfaces slide against each other. It is a nonlinear phenomenon which can cause undesirable behaviour in motion control systems such as stick-slip limit cycling, non-zero steady-state errors, and large settling times [1–5, 15], when classical control solutions are employed.

In the literature, many control approaches have been presented that deal with motion systems subject to friction, which can be divided into model-based friction compensation and non-model based control techniques. Often in model-based friction compensation techniques, a parametric friction model is used either in a feedback or feedforward control loop in order to compensate for friction [2–8]. Model-based friction compensation methods [2–8] can be inaccurate and insufficient due to the fact that these frictional effects are often hard to model accurately, as these are in general uncertain, possibly position-dependent, and subject to changing characteristics due to, e.g., changing lubrication conditions, temperature, wear, humidity, etc., resulting in modelling errors. Adaptive control [9–12] can be employed to cope with uncertain or time-varying friction circumstances by a continuous online update of the friction model parameters used in the control law. However, unavoidable modelling errors still remain, resulting in limited positioning performance.

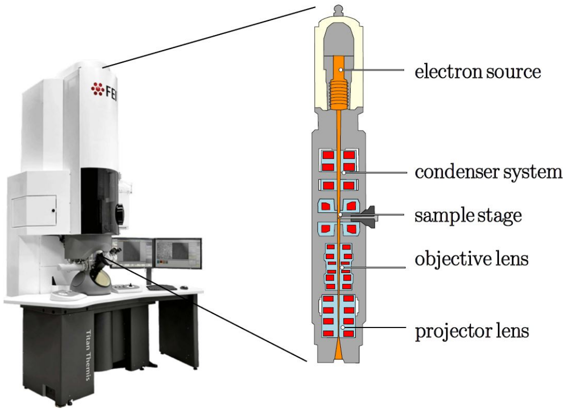

Figure 1.1: Electron microscope [Figure 1.1 in [18]].

Non-model based friction compensation techniques have the advantage that these techniques do not rely on knowledge of the friction characteristic. In non-model based techniques, specific control signals are applied to change the system behaviour in order to obtain the desired system performance despite the presence of friction. For example, non-model based friction compensation techniques are impulsive control strategies [13,14] and dithering-based controllers [20, 21]. In dithering-based techniques [20, 21], high-frequent vibrations are employed to smoothen the discontinuity induced by the friction. The impulsive control strategies [13,14] use impulsive control signals when the system gets in a stick phase with non-zero position error, in order to escape such a stick phase. The disadvantage of both non-model based friction compensation techniques is that these may excite high-frequency system dynamics which can damage the motion system. Therefore, non-model based friction compensation techniques are not appealing for being used in industrial applications [1, 13, 14, 16].

Despite the existence of these control techniques, the classical PID feedback controller [1, 15, 16, 19, 22] is most commonly applied for frictional motion systems in industry, due to the existence of intuitive tuning tools (loop-shaping), and knowledge of control practitioners. Conventional P(I)D-based feedback control is however prone to performance limitations.

In the absence of integral action, a non-zero steady-state error may be obtained [3]. For motion control systems subject to *Stribeck* friction, including integral action, it is possible to escape the stick behaviour, since the integral action eventually compensates for the static friction by integrating the position error. When the error is non-zero, the integrator builds up control force, eventually leading to a control effort exceeding the unknown static friction force. However, due to the velocity-weakening Stribeck effect, the integrator action overcompensates the friction force which can lead to overshooting the desired setpoint, and subsequent a stick phase occurs. The process of (re)filling and depletion of the integrator induces the undesired limit cycle behaviour (*hunting*) [3, 15, 16]), thereby losing asymptotic stability of the setpoint. Furthermore, in absence of Stribeck effect, the slow build up (depletion and (re)filling) of the integration buffer can lead to large settling times which are highly undesired as well [1]. A solution for this problem is presented in [1]. A reset integral control approach is proposed in [1], which circumvents the time-consuming process of depletion and refilling of the integrator buffer (whenever the system overshoots the setpoint). Resetting the integral action, results in shorter periods of stick and thereby significantly faster settling.

The main focus of this project resides on improving PID-based control for repetitive positioning of frictional systems because PID-based control is still applied in the vast majority of industrial motion systems. Given the performance limitations in conventional PID-based control when considering frictional motion systems in a repetitive motion setting, PID-based control with a *time-varying integrator gain* design is proposed to improve setpoint positioning performance. Due to the fact that the friction characteristics are uncertain, the optimal tuning of this time-varying integrator gain is unknown. As such, in this project, a data-based learning approach is pursued that iteratively improves the system performance by adaptive tuning of the time-varying integrator gain.

1.2 Data-driven performance optimization

In the literature, many optimisation-based methods to adaptively tune system parameters to achieve improved performance have been presented. A commonly used optimisation-based method to improve the system performance is Iterative Learning Control (ILC) [25, 28]. The optimisation-based method ILC can be employed to iteratively learn the scheduling of the integrator gain for repetitive motion profiles. Iterative Learning Control (ILC) is a learning based method which executes the same task multiple times and under the same operating conditions. ILC improves the performance by using the previous error information in the control input for subsequent iterations. In general, ILC is employed to modify the feedforward control input under the condition that the system is reset at the beginning of every task [25]. In addition, ILC can compensate for exogenous signals, such as repeating disturbances by learning from previous iterations. One of the methods for ILC is the norm-optimal ILC [30,31], in which the optimization problem is defined as finding the minimum of a quadratic objective function. In [38], a generalization of norm-optimal ILC for nonlinear systems with constraints is discussed. It is an extension of the conventional normal-optimal ILC for linear time-invariant (LTI) systems in [30]. A model-based iterative learning control (ILC) for nonlinear systems is proposed in [39]. The control law makes use of the inverse of the nonlinear model. Another frequently used ILC methodology is ILC with basis functions [29, 32, 33]. The basis functions are used to parameterize the inverse model of the real system. Moreover, basis functions are exploited to enable performance enhancement through iterative learning, while providing flexibility with respect to task variations. There are also other ILC methodologies for nonlinear systems, such as the high-order internal model (HOIM) ILC [40], robust ILC [28], machine learning ILC [41].

In the literature, there are also model-free PID tuning methods, such as unfalsified control [42,43,46] and iterative feedback tuning (IFT) [44–46]. In unfalsified control, a controller can be designed that meets a given performance criterium, without knowledge of the system model (based on output measurements). In unfalsified control, an adaptive algorithm is used to update the PID controller gains based on whether or not the controller satisfies a given criterion (performance specifications) [46]. IFT can be used to iteratively optimize the PID controller gains with respect to an objective function, obtained from output measurements.

The optimisation-based ILC methods and the model-free PID tuning methods are not suitable to adaptively tune the *time-varying* integrator gain. The vast majority of the conventional optimisation-based ILC methods have the disadvantage that these methods rely on accurate knowledge of the plant model (including the friction characteristic). It is hard to model friction due to changing environmental conditions. Moreover, in most ILC methods, a feedforward signal is adaptively tuned and not the feedback controller. The model-free PID tuning methods are tailored to adaptively tune *constant* PID controller gains only. Therefore, in this project, a *model-free* sampled-data extremum seeking control framework is proposed, to adaptively tune the time-varying integrator gain design [48, 51, 52]. The sampled-data extremum seeking control is able to cope with the nonlinear behaviour of the motion system, because it does not rely on model knowledge. The optimization problem is described in terms of a model-free sampled-data extremum seeking control problem [27, 47, 48, 51, 53] in which basis functions are used to parametrize the time-varying integrator gain. The extremum seeking control framework is able to optimize the system performance by adaptive tuning of the parameters in the basis functions.

1.3 Problem statement

As described in Section 1.1, motion control systems subject to friction are challenging to control accurately to a setpoint, due to the fact that friction is generally unknown or uncertain. Classical PID control is commonly used in industry for frictional systems, but PID control has performance limitations, such as limited setpoint accuracy, limit cycling, and large settling times. The main focus of this project is on improving PID-based control in order to overcome the performance limitations for repetitive positioning of frictional systems.

Therefore, in this project, the research control goal is formulated as follows:

”Design a PID-based learning controller for motion systems subject to an unknown friction characteristic that achieves optimal setpoint positioning accuracy and an optimal transient response, for a repetitive motion profile”.

The related sub-goals, also reflecting the approach taken, are:

(i) Design of a time-varying integrator gain to facilitate improved setpoint accuracy and transient responses for repetitive point-to-point motion of systems subject to friction.

(ii) Development of an automatic tuning procedure to improve performance optimization, by learning the optimal time-varying integrator gain.

(iii) Experimentally verify the proposed PID-based learning controller on an industrial nano-positioning motion stage setup.

1.4 Outline of thesis

This report is organized as follows. In Chapter 2, the considered frictional dynamical system is given and discussed. Furthermore, the performance limitations of classical PID control for frictional motion systems are illustrated by means of a numerical simulation example. In addition, various designs of the time-varying integrator gain are proposed. The effectiveness and the working principle are illustrated by numerical simulation examples. In Chapter 3, the *model-free* sampled-data extremum seeking control framework is introduced and a detailed explanation is given. Simulations results in which the sampled-data ESC framework is applied on the considered frictional motion system with the proposed time-varying integrator gain controller are presented and discussed. Chapter 4 focusses on the application of the proposed sampled-data ESC framework on an industrial high-precision positioning stage setup. Its working principle and effectiveness are demonstrated via several experiments. This thesis is closed with conclusions and recommendations given in Chapter 5.

CHAPTER 2

PID-based control for motion systems with Stribeck friction

In this chapter, the performance limitations of classical PID control for frictional motion systems are discussed to support the problem statement of the current research. Furthermore, a PID-based controller with time-varying integral gain is presented and discussed. The proposed designs of the time-varying integrator gain are applied on the single-sliding mass motion system and their working principle and effectiveness are illustrated using a numerical simulation example, see also [23].

2.1 PD and PID control for motion systems with friction

2.1.1 Description of a single-mass system with Stribeck friction

Consider a single sliding mass m, which is subject to a control force u_c, and a friction force f_f, as shown in Figure 2.1. The goal is to control the single sliding mass to the constant setpoint $(x,\dot{x}) = (r,0)$, with x the displacement and $\dot{x}$ the velocity. Using Newton's second law, the equation of motion can be written as:

$$m\ddot{x} = u_c - f_f, \tag{2.1}$$

where the friction force f_f is given by the following set-valued friction law [3, 17]:

$$f_f \in F_s\text{Sign}(\dot{x}) + \alpha\dot{x} - f(\dot{x}), \tag{2.2}$$

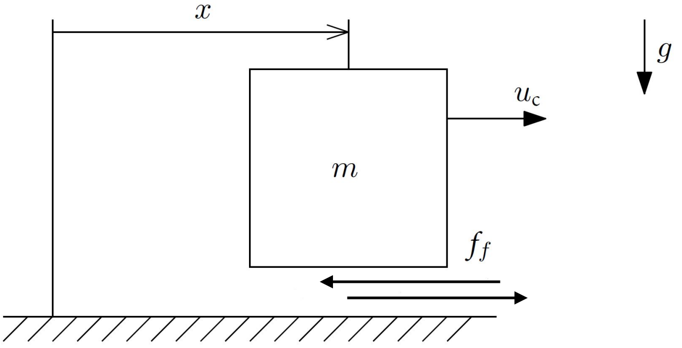

Figure 2.1: Single sliding mass m subject to friction and controlled by u_c [17].

In Figure 2.2, the friction model in (2.2) is depicted with its individual components and below every individual component in (2.2) is discussed:

1. The first term $F_s\text{Sign}(\dot{x})$ in (2.2) is the discontinuous, set-valued static friction model, where F_s denotes the static friction. The set-valued sign function $\text{Sign}(\dot{x})$ is defined as follows:

$$\text{Sign}(\dot{x}) = \begin{cases} \{1\} & \text{if } \dot{x} > 0, \\ [-1, 1] & \text{if } \dot{x} = 0, \\ \{-1\} & \text{if } \dot{x} < 0. \end{cases} \tag{2.3}$$

   Static friction is the friction that exists between a stationary object and the surface on which it is resting. When the static friction has been overcome by the control input u_c the object starts to move (i.e., when $u_c > f_f$, see (2.1)).

2. The second term $\alpha\dot{x}$ in (2.2) is the viscous friction, which is a linear function of the velocity, with $\alpha > 0$ the viscous friction parameter.

3. The last term $f(\dot{x})$ in (2.2) enables the modelling of the velocity-weakening (Stribeck) effect, which can, for example, be modelled as an exponentially decrease from static friction F_s to Coulomb friction F_c.

The control force u_c is generated by a PID controller and is defined as:

$$\begin{aligned} u_c &= k_p e + k_d \dot{e} + k_i x_3, \\ \dot{x_3} &= e, \end{aligned} \tag{2.4}$$

where e is the position error (e = r - x), k_p denotes the proportional gain, k_d denotes the derivative gain, k_i denotes the integral gain, and x_3 the integrator state.

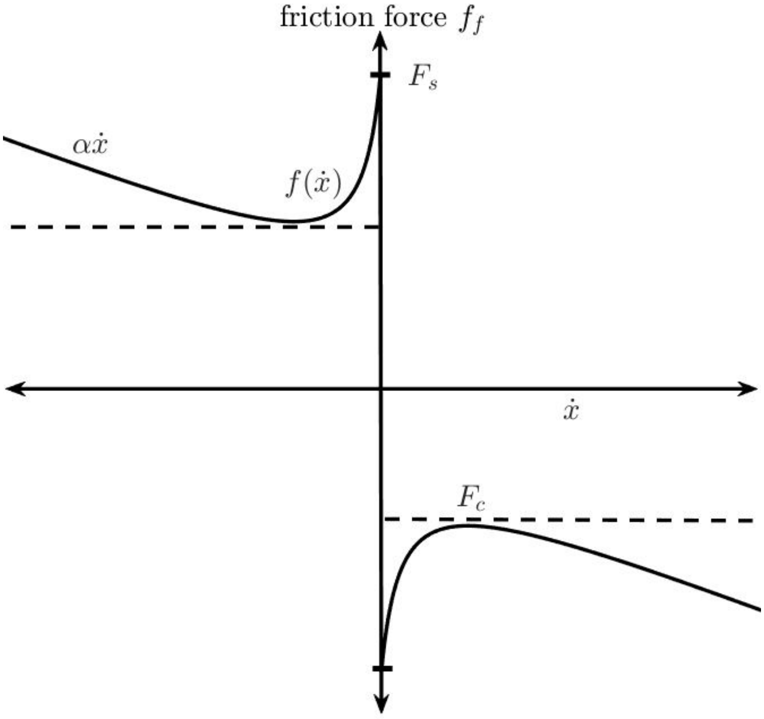

Figure 2.2: Friction characteristic.

Using the following change of variables $x_1 := x$, $x_2 := \dot{x}$, (2.1) can be rewritten in the form of two first-order differential equations, and a differential inclusion:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} \in (-F_s \mathrm{Sign}(x_2) - \alpha x_2 + f(x_2) - k_p(x_1 - r) - k_d x_2 - k_i x_3)/m \\ \dot{x_3} = x_1 - r, \end{cases} \tag{2.5}$$

where x_1 is the position of the mass, and x_2 the velocity.

The equilibria set of (2.5) is found by evaluating (2.5) at $\dot{x_1} = \dot{x_2} = \dot{x_3} = 0$ and $f(0) = 0$ and solving for x_1, x_2 and x_3, resulting in

$$\mathcal{E}_{PID} = \left\{ (x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_1 = r, x_2 = 0, x_3 \in [-\tfrac{F_s}{k_i}, \tfrac{F_s}{k_i}] \right\}. \tag{2.6}$$

The PID controller results in an equilibrium set in x_3, and, the setpoint (r,0) is the only equilibrium point on the (x_1,x_2)-plane. The set $\mathcal{E}_{PID}$ is not attractive, and solutions generally do not converge to the set $\mathcal{E}_{PID}$, but a limit cycle occurs instead in the presence of a velocity-weakening effect [19]. When $f(\dot{x}) = 0$, no limit cycling occurs for a single-sliding mass motion system and the setpoint (r,0) is asymptotically stabilized by the PID controller (2.4), if the following assumption holds [16, 19]:

Assumption 1. *The control parameters k_p, k_d, k_i satisfy $k_i > 0$, $k_p > 0$, $k_p k_d / m > k_i$*

For (2.5) without friction, according to the Routh-Hurwitz stability criterion, global exponential stability of the equilibrium $x_1 = r$, $x_2 = x_3 = 0$ is ensured under *Assumption 1.*

The equilibrium set of the system (2.5) without the integrator term ($k_i = 0$) is given as follows:

$$\mathcal{E}_{PD} = \left\{ (x_1, x_2) \in \mathbb{R}^3 \mid x_1 \in [r - \tfrac{F_s}{k_p}, r + \tfrac{F_s}{k_p}], x_2 = 0] \right\}. \tag{2.7}$$

This means that the PD controller results in an equilibrium set in x_1. All the solutions tend to the equilibrium set, but not necessarily to the setpoint (r,0) [3].

2.1.2 Repetitive motion control setting and performance measure

Consider a repetitive motion profile defined on the time interval t $\in$ [0,T], where T is the period time of the motion profile. The time window [0,T] can be divided in two time intervals:

1. t $\in$ [0,T_A); this is the transient time window in which the system starts and moves to a constant setpoint reference r and where transient response is to be improved. T_A is the time instant where the system is required to be at the setpoint as accurately as possible.

2. t $\in$ [T_A, T]; in this time window, the system should be at rest (standstill, velocity should be zero) with optimal setpoint accuracy.

The desired performance of the motion control system, minimized overshoot (transient response) on the time interval [0, T_A], and an optimal setpoint accuracy on [T_A,T], is captured by the following objective function:

$$\mathbf{J}(e) = w_1 \int_0^{T_A} |e(\tau)|^2 \, d\tau + w_2 \int_{T_A}^{T} |e(\tau)|^2 \, d\tau, \tag{2.8}$$

where $e : = r$ - x_1, and w_1 and w_2 are suitable weighting factors. These weighting factors can be chosen, depending on the performance requirements (transient performance versus setpoint accuracy). If for example, the system behaviour during transient time window is not of interest and the focus lies only on setpoint accuracy, the weight w_1 can be chosen as zero or a small number and weight w_2 can be chosen as 1 or higher.

2.1.3 Analysis of the control system performance

To illustrate the performance limitations of using conventional linear PD and PID feedback control for motion control systems subject to Stribeck friction, the closed-loop system (2.1), (2.2), and (2.4) is simulated using a numerical time-stepping method [26, chap. 6]. The numerical time-stepping method is used to effectively deal with the discontinuous set-valued friction without the need for event detection. In Figure 2.3, the state x_1 (displacement) with the boundary of the equilibrium set (2.7), x_2 (velocity), control force u_c (2.4) and the total friction force f_f (2.2) are depicted. The friction characteristic are as follows; $F_s = 0.981$ N and the velocity-weakening Stribeck effect is given by [2,3]:

$$f(x_2) = \frac{(F_s - F_c)\eta x_2}{1 + \eta |x_2|}, \tag{2.9}$$

where F_c denotes the Coulomb friction, η is the Stribeck shape parameter and α the viscous friction parameter. The simulations are done using the following values: $m = 1$ kg, $g = 9.81$ m/s^2, $k_p = 18$ N/m, $k_d = 2.5$ Ns/m, $k_i = 30$ N/(ms), $x_1(0) = 0$ m, $x_2(0) = 0$ m/s, $F_c = F_s/2$ N, $\eta = 20$, $\alpha = 0.5$ Ns/m, and $r = 0.1$ m.

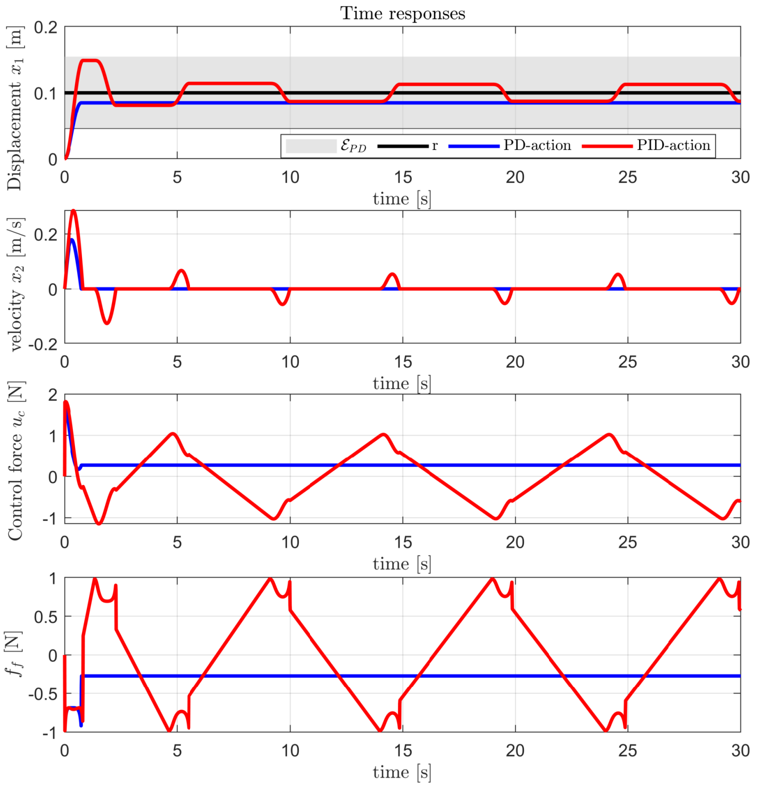

Figure 2.3: The state x_1 (displacement) with the boundary of the equilibrium set (2.7), x_2 (velocity), control force u_c (2.4) and the total friction force f_f (2.2).

Consider first a PD controller for input u_c (2.4). It is illustrated in Figure 2.3 that the error response results in a nonzero steady-state error due to the presence of the static friction F_s, see the equilibrium set for the PD control case in (2.7). As the error becomes small, the control force u_c generated by the PD controller may be smaller than the static friction F_s. As as result, the stick phase has been reached. The size of the equilibrium set (2.7) depends inversely proportional on k_p.

Second, consider the PID controller. Adding an integrator to the controller allows to compensate for the static friction F_s by building up control force, and allows the system to break free from the stick phase. As a result of the Stribeck effect, the controller overcompensates the friction during the slip phase that follows, leading to overshoot, and a subsequent stick phase occurs. Subsequently, the integration buffer first has to deplete in order to apply a control force in the opposite direction (refilling of the integration buffer) to overcome the static friction again. The process of (re)filling and depletion of the integrator repeats itself and induces the undesired stick-slip oscillations (limit cycle) around the setpoint. It can be seen from Figure 2.3 that the equilibrium set $\mathcal{E}_{PID}$ (2.6) is not attractive, and solutions do not converge to the set $\mathcal{E}_{PID}$ (2.6), but a limit cycle occurs instead in the presence of the Stribeck effect. In Figure 2.3 the settling, depletion and refilling behaviour of the integration buffer are depicted.

In absence of the Stribeck effect, the process of depletion and refilling of the integrator buffer (change of sign of the integrator state) causes long periods of stick, resulting in large settling times (the system converges increasingly slow towards its setpoint). This process takes increasingly more time with a decreasing position error, and as a result long periods of stick occur. Therefore, the settling performance of the motion system is limited. In [1], a reset integral control approach is proposed, which circumvents these (time-consuming) long periods of stick. The integrator is reset whenever the system overshoots the setpoint, and when the system enters a stick phase. Resetting the integral action results in shorter periods of stick, which increases the settling performance significantly.

2.2 PID-based control with time-varying integrator gain

2.2.1 Intuitive feedback control design

As have been demonstrated in Section 2.1.3, using the classical PID controller, and in particular the integral action, on motion control systems subject to Stribeck friction results in stick-slip limit cycling. The stick-slip limit cycling process motivates the design of a time-varying integrator gain $k_i(t)$. The resulting PID-based controller is defined as follows:

$$\begin{aligned} u_c &= k_p e + k_d \dot{e} + k_i(t) x_3 \quad \text{for } t \in [0, T] \ , \\ \dot{x_3} &= e, \end{aligned} \tag{2.10}$$

where the gains k_p and k_d are defined beforehand using loop-shaping techniques and the proposed $k_i(t)$ denotes the time-varying integrator gain. Intuitively, a time-varying integrator gain $k_i(t)$ design that enables improved system performance can have the following shape: during the transient time window, initially, a high integrator gain is desired to overcome the static friction F_s and escape possible stick. Subsequently, a low (or even negative) integrator gain can be needed to avoid overcompensation of friction and minimize overshoot, and achieve a high setpoint accuracy at the end of the transient window. Finally, at the setpoint, a zero integrator gain is desired to prevent limit cycling and create robustness close to the setpoint. Namely, if $k_i = 0$, the system cannot escape the stick phase (standstill of the system), since the integral action is and remains zero, preventing unnecessary control force. Therefore, the total control force at the setpoint is significantly lower (proportional action is low) than the level of static friction F_s, resulting in robustness against force disturbances. A possible shape of the time-varying integrator gain $k_i(t)$ is demonstrated in Figure 2.4. In the next section, potential designs of the time-varying integrator gain are presented and discussed.

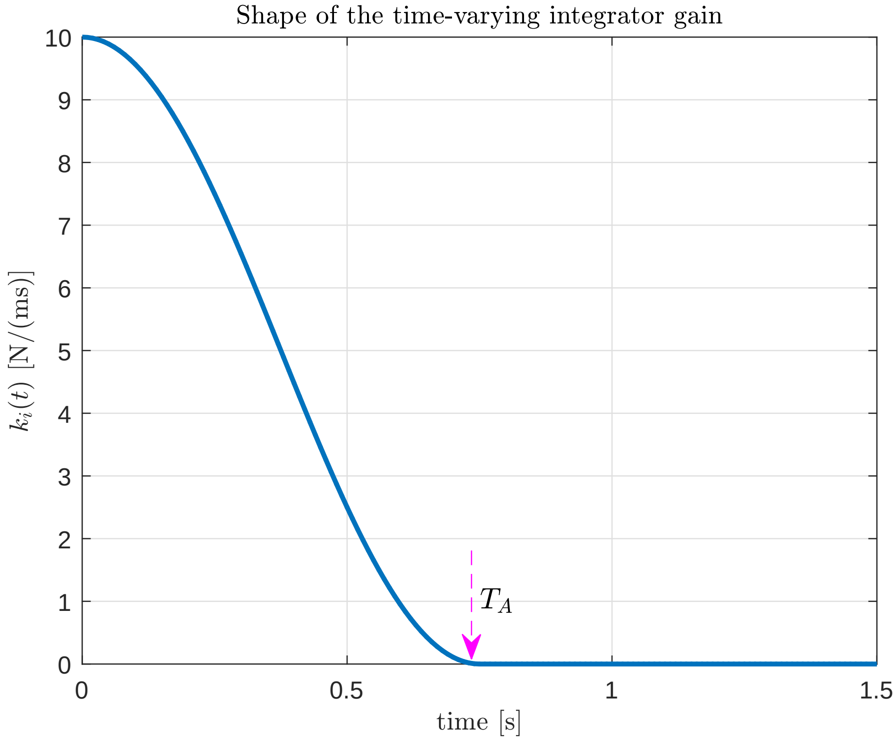

Figure 2.4: Qualitative shape of the time-varying integrator gain $k_i(t)$.

2.2.2 Time-varying integrator gain design

To design the time-varying integrator gain, a parametrization is proposed using a finite set of basis functions Ψ, and a parameter vector $u \in \mathbb{R}^p$ to be constructed, as follows:

$$k_i(t) = \sum_{j=1}^{p} u^{(j)} \Psi^{(j)}(t), \tag{2.11}$$

where u is a vector with to be designed parameters, $u^{(j)}$ indicates the jth element of the vector u, and p is the number of elements in vector u. In literature, various basis functions have been proposed, including step-like basis functions [23], Fourier basis functions [24], polynomial basis functions [29], rational basis functions [33], etc. In this project, three potential choices for the basis function Ψ are investigated, which are step-like, Fourier and linear basis functions.

Example 1: Step-like basis functions

The step-like basis function and the corresponding parameter vector are given by [23]:

$$\Psi^{(j)}(t) := \begin{cases} 1, & t \in [(j-1)t_s, jt_s) \\ 0, & t \notin [(j-1)t_s, jt_s) \end{cases} \quad \text{for } j = 1, 2, ..., p, \tag{2.12}$$

$$u^T = [u^{(1)} \quad u^{(2)} \cdots u^{(p)}], \tag{2.13}$$

where t_s satisfies $T = pt_s$. Note that T is the period time of the repetitive motion profile. To investigate the potential of this specific parametrization of the time-varying integrator gain $k_i(t)$ (2.11-2.13), a simulation example is presented in Section 2.3, and the resulting time responses are compared with other examples.

Example 2: Fourier basis functions

Based on intuition, a high k_i is needed to overcome the stick phase, a reduced k_i is required to prevent overcompensation of friction and a zero k_i at the setpoint is required to put the system at rest (rule out limit cycling); the shape of the time-varying integrator gain $k_i(t)$ can be captured by a first-order cosine Fourier (periodic) basis function, see Figure 2.4. As a result, the number of elements in vector u is reduced to $p = 2$, and the parametrization of $k_i(t)$ is almost everywhere continuous differentiable.

The PID controller is given by (2.10) and the parametrization of the integrator gain $k_i(t)$ is according to $k_i(t) = u\Psi(t)$, in which the Fourier basis function and the parameter vector are defined as follows [24]:

$$\Psi(t) := \begin{cases} \begin{bmatrix} \cos(\omega t) \\ 1 \end{bmatrix}, & t \in [0, T_A) \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, & t \in [T_A, T] \end{cases} \tag{2.14}$$

$$u^T = \begin{bmatrix} A & \delta \end{bmatrix}, \tag{2.15}$$

where $\omega = 2\pi f$ is the angular frequency. The parameters A and δ are the amplitude and offset, respectively.

In (2.14) the integrator gain $k_i(t)$ is set to zero in the time window $[T_A,T]$ to rule out limit cycling. To enforce continuity of $k_i(t)$, the parameter ω is fixed to $\omega = \dfrac{\pi + \cos^{-1}(\frac{\delta}{A})}{T_A}$. Note that for $\delta/A > 1$ and $\delta/A <$ -1 there are no solutions for ω and thus also for the control input force u_c (2.10).

Example 3: Linear basis functions

In this case, the control input force u_c is given by (2.10), where the parametrization of $k_i(t)$ is given by (2.11) in which the basis function is defined as [29]:

$$\Psi^{(j)}(t) := \begin{cases} \frac{t-(j-2)t_s}{(j-1)t_s-(j-2)t_s}, & t \in [(j-2)t_s, (j-1)t_s) \\ \frac{jt_s-t}{jt_s-(j-1)t_s}, & t \in [(j-1)t_s, jt_s) \\ 0, & \text{otherwise} \end{cases} \qquad \text{for } j = 1, 2, ..., p, \tag{2.16}$$

where t_s satisfy $T = pt_s$. The corresponding parameter vector u in (2.11) is given by:

$$u^T = [u^{(1)} \quad u^{(2)} \cdots u^{(p)}] \tag{2.17}$$

2.3 Simulation study: Parametrization of time-varying integrator gain

In this section, numerical simulation examples are presented to illustrate the effectiveness of all the proposed parametrizations of the time-varying integrator gain, as discussed in Section 2.2.2. Consider the single-mass sliding motion system as defined in Section 2.1, which is subject to a control force u_c (2.10) and a friction force f_f (2.2). In this simulation study, two types of Stribeck characteristics are applied to show the robustness of the time-varying integrator gain $k_i(t)$ for different friction characteristics. The velocity-weakening (Stribeck) effect is given by (2.9), where F_s = 0.981 N. The two types of Stribeck characteristic (weak and strong Stribeck effect, see Figure 2.5) are defined by the following parameters:

$$\text{case a) } F_c = F_s/2,\ \eta = 20 \text{ and } \alpha = 0,$$
$$\text{case b) } F_c = F_s/4,\ \eta = 60 \text{ and } \alpha = 0.5$$

A repetitive motion profile is considered on the interval [0, T], with T = 1.5 s, T_a = 0.75 s and a constant reference r = 0.1 m. The following numerical values are used: $x_1(0)$ = 0 m, $x_2(0)$ = 0 m/s, k_p = 18 N/m, k_d = 2.5 Ns/m and $k_i(t)$ is parametrized by a basis function, and a parameter vector, as given in Section 2.2.2.

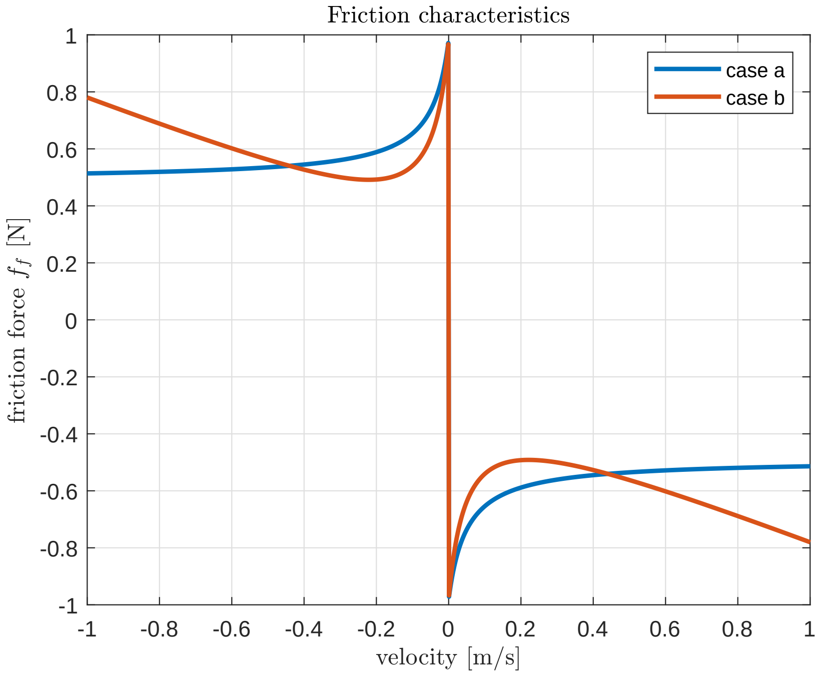

Figure 2.5: Weak (case a) and strong (case b) Stribeck effect.

Simulation 1: Step-like basis functions

For this simulation example of the step-like basis function (2.6), the parameter vector u is given by:

$$u^T = [u^{(1)} \quad u^{(2)} \quad u^{(3)} \quad u^{(4)} \quad u^{(5)} \quad u^{(6)}], \tag{2.18}$$

where $p = 6$ and from $T = pt_s$ it can be derived that $t_s = 0.25$ s. The parameter $u^{(1)}$ is fixed at the value of 10 N/(ms) and $u^{(j)} = 0$ N/(ms) for all $j = 4, ..., p$. In this simulation example, the parameters $u^{(2)}$ and $u^{(3)}$ are chosen as follows:

case a) $u^{(2)} = 5$ and $u^{(3)} = 1$,
case b) $u^{(2)} =$ -4 and $u^{(3)} =$ -8.

The parameters $u^{(2)}$ and $u^{(3)}$ are chosen based on the intuition that the optimal time-varying integrator gain $k_i(t)$ is expected to have decaying shape, as explained in Section 2.2.1. Note that for case b) negatives values of the parameters $u^{(2)}$ and $u^{(3)}$ are chosen, which are needed to resist the rapid reduction of the friction force due to this kind of severe Stribeck effect.

In Figure 2.6, the position error e and control force u_c with fixed $k_i = 0$ N/(ms), and $k_i =$ 10 N/(ms), and time-varying integrator gain $k_i(t)$ for both the friction cases are depicted. It is illustrated in Figure 2.6 that for a fixed $k_i = 0$ N/(ms), the response results in a nonzero constant steady-state error and for $k_i = 10$ N/(ms), the response results in overshoot of the setpoint and subsequently hunting occurs (note that the time window in Figure 2.6 is too short to observe the limit cycle). So, on the one hand, $k_i = 0$ N/(ms)(PD control), and on the other hand $k_i = 10$ N/(ms) both results in limited setpoint accuracy and transient performance. When a time-varying integrator gain $k_i(t)$ is applied, the system performance can be improved; the setpoint accuracy and transient performance are significantly enhanced for both the friction cases compared to the classical P(I)D control, as depicted in Figure 2.6.

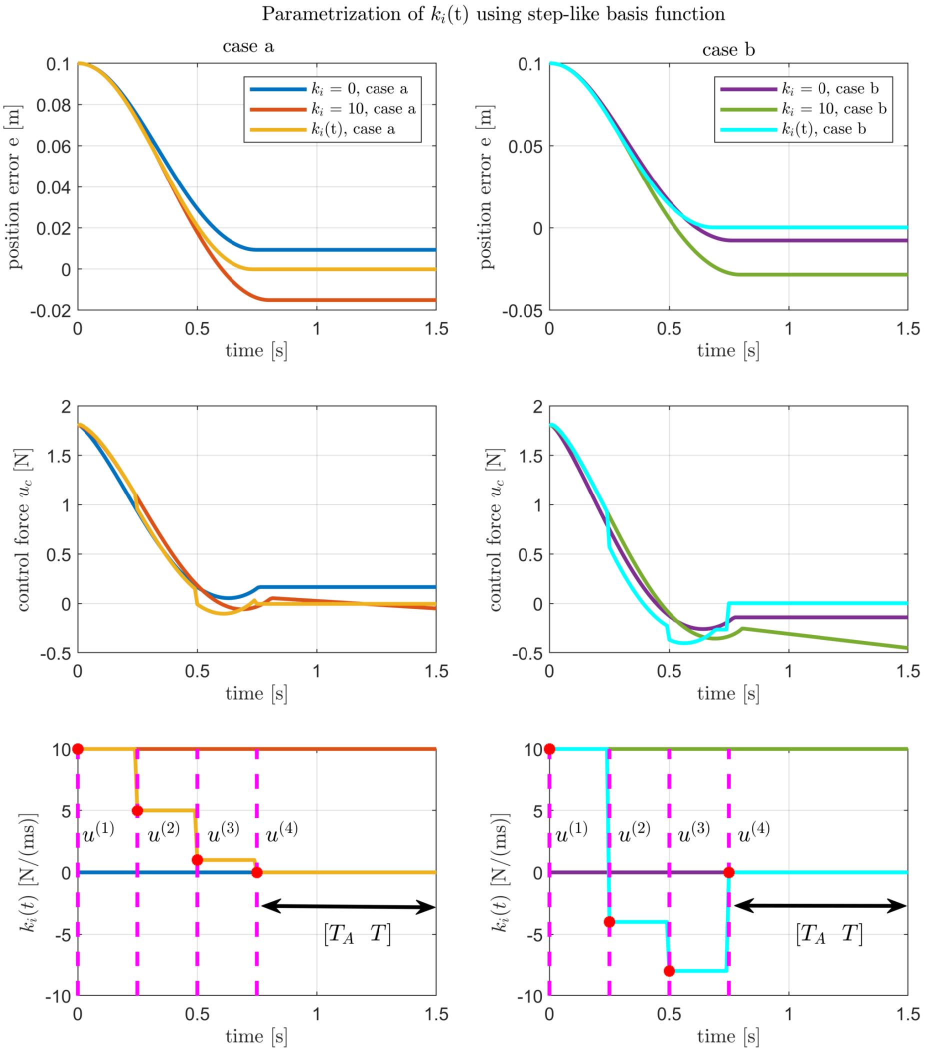

Figure 2.6: Position error e, control force u_c and step-like basis function parametrization of the time-varying integrator gain $k_i(t)$ (2.12).

Simulation 2: Fourier basis functions

A numerical example is presented to validate if the Fourier basis function parametrization of $k_i(t)$ (2.14) improve the desired system performance. For this simulation example, the parameters A and δ in the parameter vector (2.15) are chosen as follows:

$$\text{case a) } A = 5 \text{ and } \delta = 5,$$
$$\text{case b) } A = 8 \text{ and } \delta = 2.$$

It can be seen from Figure 2.7, that when a time-varying integrator gain $k_i(t)$ is parametrized as a Fourier basis function, the desired performance is improved; the position error at the setpoint has decreased significantly and transient performance is improved for both the friction cases, compared to $k_i = 0$ N/(ms) and $k_i = 10$ N/(ms). To have a more clarified figure, the results of fixed $k_i = 0$ N/(ms) for case b), and $k_i = 10$ N/(ms) for case b) are omitted in Figure 2.7, but these results are already depicted in Figure 2.6.

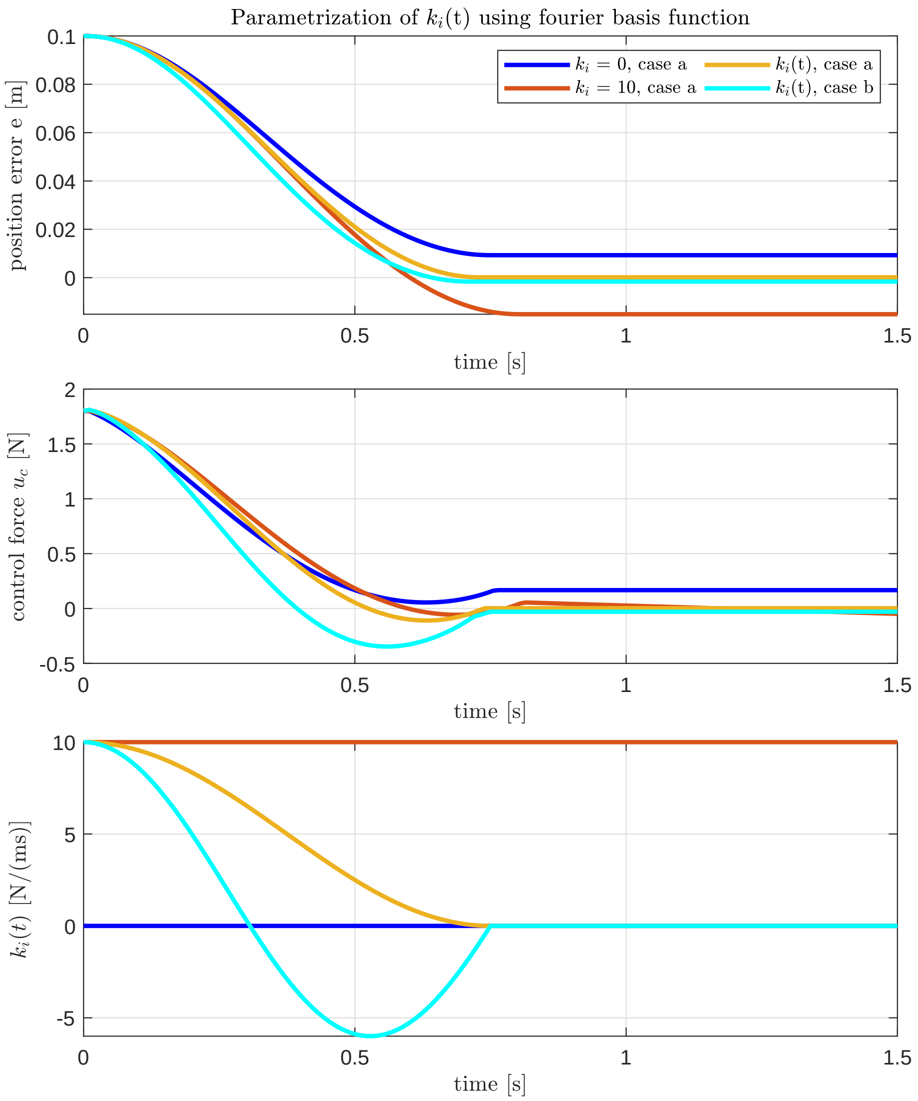

Figure 2.7: Position error e, control force u_c and Fourier basis function parametrization of the time-varying integrator gain $k_i(t)$ (2.14).

Simulation 3: Linear basis functions

The effectiveness of the linear basis function parametrization for $k_i(t)$ (2.16) is also validated by a numerical simulation example. In this simulation example, the number of elements in u is $p = 6$, resulting in $t_s = 0.25$ s. The parameter $u^{(1)}$ is fixed at the value of 10 N/(ms) and $u^{(j)} = 0$ N/(ms) for all $j = 4, ..., p$. The parameters $u^{(2)}$ and $u^{(3)}$ are chosen as follows:

case a) $u^{(2)} = 7$ and $u^{(3)} = 2$,
case b) $u^{(2)} = 2$ and $u^{(3)} =$ -8.

Between the parameters $u^{(1)}$ and $u^{(2)}$; and between the parameters $u^{(2)}$ and $u^{(3)}$; and between the parameters $u^{(3)}$ and $u^{(4)}$ a linear interpolation polynomial function is used, which is given by (2.16).

For the linear basis function parametrization of $k_i(t)$ the same conclusions can be drawn as from the simulation examples of the Fourier and linear basis functions parametrizations of $k_i(t)$. It can be seen from Figure 2.8 that with this linear basis function parametrization of the time-varying integrator gain $k_i(t)$ the setpoint accuracy is significantly improved and overshoot/undershoot (transient performance) is minimized with respect to both the friction cases, compared to $k_i = 0$ N/(ms) and $k_i = 10$ N/(ms).

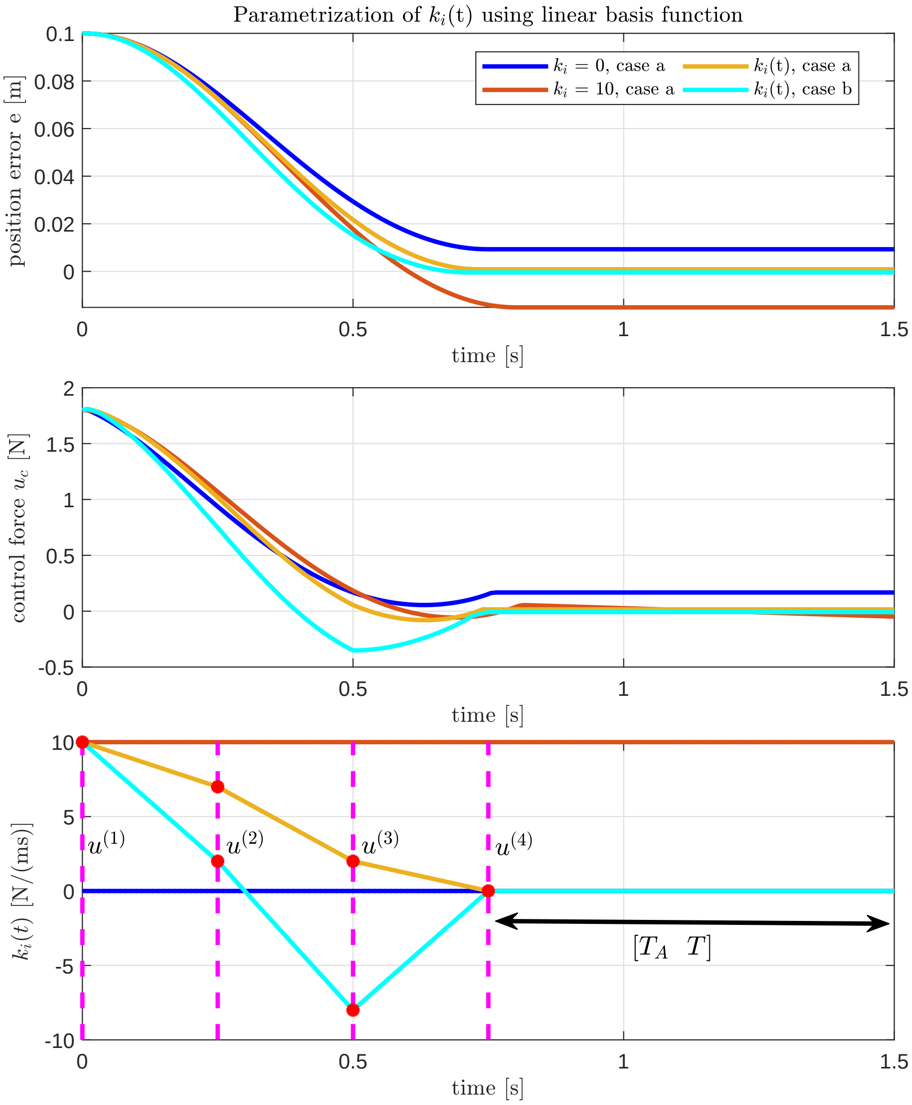

Figure 2.8: Position error e, control force u_c and linear basis function parametrization of the time-varying integrator gain $k_i(t)$ (2.16).

2.4 Discussion

In Section 2.3 it has been shown that for all the three proposed basis functions parametrizations of the time-varying integrator $k_i(t)$, both the setpoint accuracy and transient response can be improved. The integrator gain is set to zero at the setpoint, which prevents build up of the control force. As such, the system cannot escape the stick phase (ruling out limit cycling), and provide robustness against force disturbances. So, the design of the time-varying integrator gain $k_i(t)$ is a good approach to achieve the setpoint control goal as defined in Section 1.3. Nevertheless, some of these basis function parametrizations have some disadvantages, which makes them less suitable to use in practice.

For example, a disadvantage of the step-like basis function parametrization of $k_i(t)$ is that there are jumps (discontinuity) in the time-varying integrator gain and thus in the control force u_c. Due to these jumps in the control force u_c, higher-order motion system dynamics can be excited [1, 16], which may cause, e.g., instability of the closed-loop system, or the machinery can be damaged as well.

The major disadvantage of the Fourier basis function parametrization of the time-varying integrator gain is that the two parameters A and δ are strongly dependent on each other. A small change in A (in the wrong direction), can deteriorate the performance, or δ has to change accordingly to compensate for the change in A, to still achieve optimal performance. Another disadvantage is, there are constraints on the parameters A and δ, $\delta/A < 1$ and $\delta/A >$ -1, which makes the optimization problem more complex. One way to solve this constrained optimization problem is by using the gradient projection method [55]. If the solution is not in the feasible set, then the gradient projection method ”projects” it onto the feasible set. The idea of the gradient projection method is explained in Appendix A.1. Moreover, due to these constraints, not all realisations of the parameter vector u are possible. An advantage of the Fourier basis function parametrization of the time-varying integrator gain is that it is almost everywhere continuous differentiable.

The linear basis function parametrization of $k_i(t)$ has many advantages which satisfy the requirements:

(i) The control force u_c (2.10) is continuous (smooth), which prevents higher-order motion system dynamics to be excited unnecessarily,

(ii) No strong dependency and no constraints between the parameters in the parameter vector u. So, the linear basis function parametrization offers more flexibility in the shape of the parameter vector u.

Therefore, for the remainder of the thesis, the PID controller, in which $k_i(t)$ is parametrized as a linear basis function is employed.

To obtain optimal performance (setpoint accuracy and transient response), the parameter vector u has to be designed optimally. Due to the fact that friction is generally unknown or uncertain, the optimal realisation of the parameter vector u is typically unknown and, therefore, difficult to tune. In Chapter 3, a model-free sampled-data extremum seeking control framework is discussed, which enables data-driven performance optimization by adaptively tuning u and therefore, the optimal vector u can be found.

CHAPTER 3

Extremum seeking control for transient performance optimization

In this chapter, the sampled-data extremum seeking control (ESC) framework in an iterative learning context is presented. Furthermore, its working principles are illustrated by means of a simulation study in which the sampled-data ESC framework is applied to optimally tune the time-varying integrator gain design for improved setpoint positioning in frictional motion systems.

3.1 Sampled-data extremum seeking control for transient performance optimization

3.1.1 Background on extremum seeking control

Extremum seeking control is a data-driven and model-free control technique, which is employed to locate an optimal operating regime of an unknown steady-state input-output map of general nonlinear systems in real-time [47, 48]. Extremum seeking control assumes the existence of a steady-state input-output relation (input-output performance map) between the tunable input and the measured steady-state output [49]. In literature, there are two main types of extremum seeking methods. The first type is continuous-time extremum seeking, which uses periodic excitation signals to explore the local behaviour of the dynamical system to be optimized, and continuously steers the system's input to one that results in optimal steady-state performance [49, 50]. The second type is sampled-data extremum seeking, which can be classified as an online extension of classical nonlinear programming (NLP) problem of function minimization within a periodic sampled-data framework [53]. Given an objective function $Q : \mathbb{R}^m \to \mathbb{R}$, the goal is to find one input $u^* \in \mathbb{R}^m$ (optimal point) such that $Q(u^*)$ is minimal. In extremum seeking control, the objective function is not given analytically but information on its value (and the values of its derivatives) at any selected u is only obtained experimentally by probing the system, which is called the evaluator of the objective function Q [53]. Within extremum seeking the gradient of the objective function Q is typically estimated based on the measured data, which is subsequently used to steer the input to the optimal input that minimizes the objective function Q, thereby achieving optimal steady-state performance.

Extremum seeking control can be used when: the plant/disturbances are unknown, performance depends on tunable system parameters/measurable output, and it is desired to optimize steady-state performance. The classical extremum seeking control is meant to optimize the *steady-state* performance, as described in [53]. In this project, it is desired to optimize the *transient performance*, i.e., minimize the tracking error over the time period of the setpoint, by adaptively tuning a time-varying integrator gain using extremum seeking control. The difference with the classical extremum seeking control is that now for each new input the system does not have to wait to settle down (reach steady-state) before taking the output measurements that reflect steady-state performance.

There are several reasons to consider sampled-data ESC method instead of continuous-time ESC method. Sampled-data ESC:

1. covers a larger class of optimization methods, for example, it is able to apply global optimization algorithms. Therefore, it can deal with non-differentiable (nonsmooth) and nonconvex input-output mappings that depend on many input parameters in a highly coupled manner [51, 53];
2. can deal with nonsmooth (infinite-dimensional) dynamical systems [51, 53].
3. is a discrete optimization framework. For the proposed time-varying integrator gain, the controller is updated after each trial to find the optimal parameter vector u, which is a discrete operation. Therefore, the sampled-data ESC is particularly suitable to find the optimal settings of u.

Therefore, in this project the main focus lies on performance optimization using a sampled-data extremum seeking control approach.

3.1.2 Sampled-data extremum seeking control framework

In this section, an iterative learning algorithm is proposed, which enables data-driven performance performance optimization by adaptively tuning u. The setpoint control problem is formulated as a model-free sampled-data ESC optimization problem by using the linear basis functions parametrization of the time-varying integrator gain $k_i(t)$ in (2.16) [23, 47]. Based on the cascade connection of the PID-controlled motion system given by ((2.1), (2.2) and (2.4)), with the linear basis function parametrization of $k_i(t)$ by (2.16), and the objective function $\mathbf{J}(e)$ in (2.8), the *unknown* input-output mapping is defined as follows:

$$Q(u) = \mathbf{J}(e) = w_1 \int_0^{T_A} |e(\tau)|^2 \, d\tau + w_2 \int_{T_A}^{T} |e(\tau)|^2 \, d\tau. \tag{3.1}$$

Note that the realized error profile for e depends on the parameters u of $k_i(t)$.

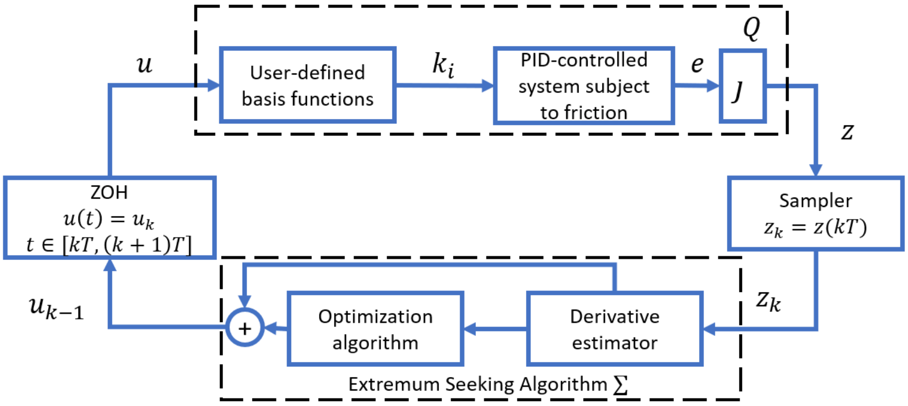

Figure 3.1: Sampled-data ESC framework.

Using output measurements, extremum seeking is able to adaptively find the optimal parameter vector u that minimizes the objective function (3.1). In Figure 3.1, a sampled-data extremum seeking framework is depicted in which the PID-controlled system subject to friction is given by (2.1), (2.2) and (2.4). In this framework a *Sampler*, *Zero-Order Hold* and an *Extremum Seeking Algorithm* $\sum$ is employed. The sampler samples the continuous output signal of the motion system such that it can be employed as an input for the sampled-data extremum seeking algorithm $\sum$. The sampling operation is given as follows [47]:

$$z_k := z(kT) \quad \text{for all } k = 1, 2, ..., \tag{3.2}$$

where $z_k = Q(u_{k-1})$ are the output measurements.

The ZOH is used to transform the sampled-based output of the sampled-data extremum seeking algorithm $\sum$ to a continuous signal such that it can be used as an input for the motion system. The zero-order hold (ZOH) operation is defined as follows [47]:

$$u(t) := u_k \quad \text{for all } t \in [kT, (k+1)T], \tag{3.3}$$

with k = 0,1,2,..., and $u(t)$ the input signal, with $T > 0$ is the sampling period. Note that T is the sampling period of the extremum-seeking controller (not the motion system), which can be chosen the same as the period time of the repetitive motion profile. In the next section, the design of the extremum seeking controller is discussed.

3.1.3 Extremum seeking controller design

In the literature, many optimization algorithms $\sum$ are presented within the sampled-data ESC framework that solves the optimization problem of finding the minimum of $Q(u)$:

$$z^* := \min_u Q(u). \tag{3.4}$$

The extremum seeking algorithm $\sum$ consists of an optimization algorithm and a derivative estimator. In this thesis, the steepest descent optimization algorithm [53] is employed to optimize the vector u in (2.17):

$$u_k = u_{k-1} - \gamma \nabla Q(u_{k-1}), \tag{3.5}$$

where γ is a fixed optimizer gain. The optimizer gain γ determines how fast the extremum seeking learning controller reaches the optimal value. A large value for the optimizer gain results in that the extremum seeking learning controller reaches an optimum value faster (convergence). However, when close to an optimum value, a large value of the optimizer gain results in overshooting the optimal value. When the same optimizer gain γ is chosen, this would result in that for some cases the system converges to the optimal value, and for other cases, it overshoots the optimal value and oscillate without ever converging. So, there is a trade-off between region of attraction (accuracy) and convergence rate.

In general, the gradient $\nabla Q(u_{k-1})$ is not available, since the objective function Q is not analytically known. It is only possible to perform measurements to estimate the objective function Q and its derivative. Using these measurements and specific input signals, the gradient of Q can be estimated based on finite-differences [23, 53]:

$$\nabla Q(u_{k-1}) = \underbrace{\frac{1}{\tau} \begin{bmatrix} Q(u_{k-1} + \tau d_1) - Q(u_{k-1}) \\ \vdots \\ Q(u_{k-1} + \tau d_j) - Q(u_{k-1}) \end{bmatrix}}_{\text{derivative estimator}} \quad \text{with } j = 1, ..., p_o, \tag{3.6}$$

where τ is the step (dither) size of the derivative estimator (a real number), k denotes the iteration index , p_o denotes the number of parameters in the parameter vector u that are dithered to determine one entry of (3.6), and d_j are vectors (dither sequence) with jth element equal to one and all others equal to zero. The dither size τ determines the size of the fluctuation around the optimal value. Increasing the dither size τ would increase the size of fluctuation around the optimal value, resulting in deteriorating the accuracy. The parameter p_o is defined as the number of elements in the parameter vector u that have to be optimized. The vectors τd_j are called dither (perturbation) signals, which needs to be applied in a sequential order. The purpose of the perturbation signal is to provide sufficient excitation to accurately estimate the gradient of the objective function Q. The dither sequence and the gradient descent method (3.5) can be combined in an extremum seeking algorithm to estimate the gradient of the objective function Q (3.6).

The extremum seeking algorithm is defined as follows [23]:

$$u_k = \begin{cases} u_{k-n} + \tau d_n & \text{if } n \neq 0 \\ u_{k-(p_o+1)} - \gamma \nabla Q(u_{k-(p_o+1)}) & \text{if } n = 0 \end{cases}, \text{ for all } k = 1, 2, ..., \tag{3.7}$$

where $n = \text{mod}(k,p_o + 1) \in \mathbb{N}$. Hence, for $n = 0$, the parameter vector u is updated, while for $n \neq 0$ the input is excited by dither in order to estimate the gradient. The gradient $\nabla Q(u_{k-(p_o+1)})$ in (3.7) is given by:

$$\nabla Q(u_{k-(p_o+1)}) = \frac{1}{\tau} \begin{bmatrix} Q(u_{k-p_o}) - Q(u_{k-(p_o+1)}) \\ \vdots \\ Q(u_{k-1}) - Q(u_{k-(p_o+1)}) \end{bmatrix}, \tag{3.8}$$

The working principle of the learning algorithm in (3.7) and (3.8) is as follows:

1. First, the system is probed with an input u_{k-1} (base experiment).

2. The next step is to probe the system with an input $u_{k-1} + \tau d_1$ (perturbation of the first parameter in vector u), then with an input $u_{k-1} + \tau d_2$ (perturbation of the second parameter in vector u), etc.

3. After collecting the output measurements of all the inputs, a new input u_k (controller update) can be computed using the extremum seeking algorithm (3.7) and (3.8), that brings the input closer to the one that minimizes the objective function. These steps are repeated, until the extremum is obtained.

In the next section, the extremum seeking learning algorithm, given by (3.7) and (3.8), and the sampled-data ESC framework are employed, to find the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ in (2.17) and where $k_i(t)$ is parametrized by linear basis function (2.16). Moreover, the effectiveness of the extremum seeking learning algorithm is illustrated by means of a simulation study.

3.2 Simulation study: Optimal tuning of a time-varying integrator gain for motion systems with friction

3.2.1 Input-output mapping

In Section 2.3, it has been demonstrated that the proposed linear basis function parametrization of the time-varying integrator gain improves the desired performance. To obtain optimal performance (setpoint accuracy and transient response), the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ in (2.17) have to be obtained by solving (3.4).

There are many optimization algorithms that can solve the optimization problem of finding the minimum of $Q(u)$ (3.4). The unknown input-output mapping $Q(u)$ (3.1) is visualized to determine which optimization algorithm suffices to solve the optimization problem. If $Q(u)$ has only one global optimum, or finding a local optimum is sufficient, the classical gradient descent method (3.5) or Newton method can be used [34, 35]. If $Q(u)$ has multiple optima (local and global), then global optimization methods (e.g. DIRECT and Shubert) can be used [36, 37], to find the global optimum. Moreover, an advantage of visualizing the input-output mapping $Q(u)$ is that the initial parameter vector u can be chosen close to the optimum, which results in faster convergence. Another benefit of visualizing the input-output mapping $Q(u)$ is to evaluate the extremum seeking learning control results.

In Section 2.3, the free 'tunable' number of elements p_o for the proposed linear basis function parametrization of the time-varying integrator $k_i(t)$ (2.16) is chosen as 2. If the number of parameters p_o increase, the vector u becomes larger, which makes the controller more complex and may required significantly more experiments to obtain optimal performance. Another major disadvantage of choosing more number of elements p_o to be optimized, is that the optimization problem (input-output mapping) becomes less intuitive. In general, it is more difficult to gain insight in the optimization problem through visualization for an increasing amount of parameters. For instance, if the number of elements $p_o = 15$, this results in an optimization problem with 15 inputs and 1 output. As a result, 1) the optimal settings cannot be visualized, and 2) it is unknown if the obtained optimum is a local one, or a global one. Moreover, it becomes more difficult to decide on which optimization algorithm is best to use for this optimization problem.

The input-output mapping $Q(u)$ of the linear basis function parametrization of the time-varying integrator $k_i(t)$ is visualized for both friction characteristics in Figures 3.2 and 3.3 below. The number of elements that have to be optimized is chosen as $p_o = 2$ and the unknown input-output mapping $Q(u)$ is given by (3.1). For the input-output mapping $Q(u)$ in Figures 3.2 and 3.3, the following conditions are used $k_p = 18$ N/m, $k_d = 2.5$ Ns/m, $w_1 = 10^{-10}$, $w_2 = 1$, $x_1(0) = 0$ m, $x_2(0) = 0$ m/s, $T = 1.5$ s, $T_A = 0.75$ s and $r = 0.1$ m. The weight w_1 is chosen low, because obtaining optimal setpoint accuracy is more of interest, compared to transient performance (minimized overshoot). In the extremum seeking literature it is a common requirement that the input-output mapping $Q(u)$ is independent of initial conditions [23]. However, in this project, the transient behaviour is partly dependent on the initial conditions. Therefore, to have an uniquely defined input-output mapping $Q(u)$, the states x_1 and x_2 have to be *re-initialized* after every setpoint operation. In practice, an automatic hooming procedure is employed to reset the system.

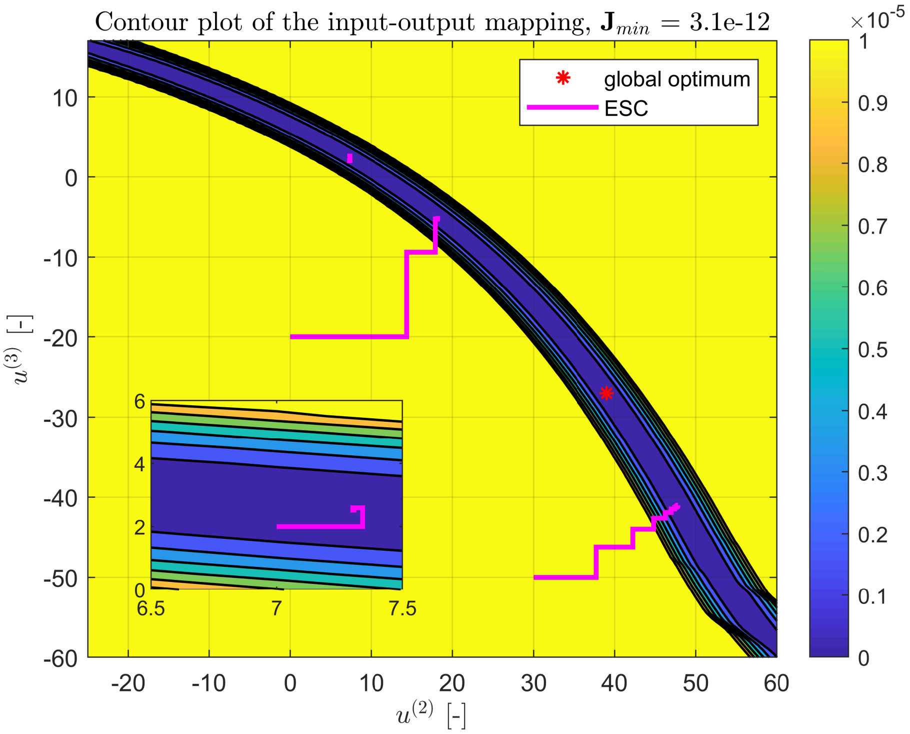

Figure 3.2: Contour plot of the input-output mapping (3.1) for the linear basis function parametrization of k_i(t) with respect to friction case a).

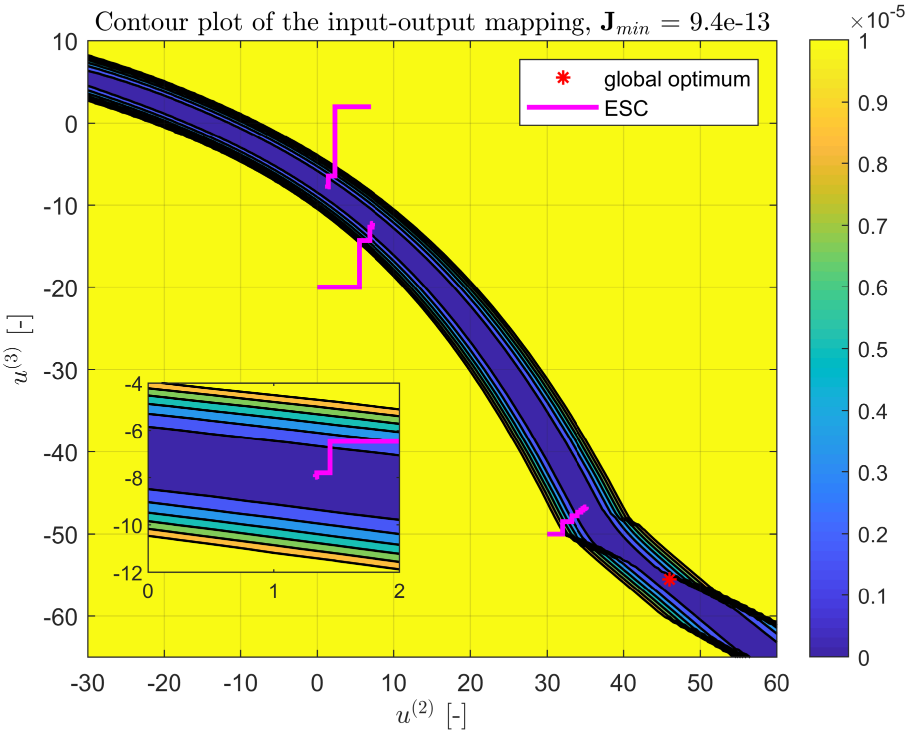

Figure 3.3: Contour plot of the input-output mapping (3.1) for the linear basis function parametrization of k_i(t) with respect to friction case b).

To improve the visualization of the input-output mapping $Q(u)$, the cost $\mathbf{J}(e)$ (2.8) in Figures 3.2 and 3.3 is saturated. All values of $Q(u)$ that are larger than or equal to 10^{-5} m^{2}s are set to 10^{-5} m^{2}s. Furthermore, the lowest cost $\mathbf{J}_{min}$ is indicated by a red asterisk marker in both Figures 3.2 and 3.3. In Figures 3.2 and 3.3, the contour plots of the input-output mapping (3.1) are depicted for the linear basis function parametrization of $k_i(t)$ regarding both friction characteristics. It can be seen from Figures 3.2 and 3.3 that there exists a set of optima (dark blue area). The desired performance is achieved, as long as the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ are attained in the set of optima (dark blue area in Figures 3.2 and 3.3). The optimization problem at hand is nonconvex because there are combinations of points in the optimal area (dark blue area) for which the line segment between the points lies outside the optimal area. The classical gradient descent optimization algorithm (3.5) can be used to solve the optimization problem at hand (finding a local optimum is sufficient). Note that there are small differences in the values of the cost $\mathbf{J}(e)$ in the optimal area (dark blue area in Figures 3.2 and 3.3), which are not visible due to the chosen saturation of the cost $\mathbf{J}(e)$ (2.8).

The classical gradient descent optimization algorithm (3.5) only finds local optima. Therefore, the resulting $k_i(t)$ (optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$), as determined by the ESC learning algorithm, depends on the initialization of the time-varying integrator gain $k_i(t)$. From Figures 3.2 and 3.3 it can be observed that for a different friction case (Stribeck effect) the input-output mapping $Q(u)$ looks different. As a result, the extremum seeking controller converges to different optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$.

Remark 1. In this section, only the input-output mappings of the linear basis function parametrization of $k_i(t)$ are visualized. The input-output mappings of both the step-like and Fourier basis function parametrizations of $k_i(t)$ are presented in Appendix A.2.

3.2.2 Application to the time-varying learning integrator gain

In this section, the sampled-data extremum seeking control framework is applied on the single sliding mass motion system as described in Section 2.1, to find the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ in the parameter vector u (2.17). The extremum seeking learning controller adaptively optimize the parameters $u^{(2)}$ and $u^{(3)}$ in the parameter vector u (2.17), resulting in a optimized $k_i(t)$ parametrized by linear basis functions, which achieves the desired performance (setpoint accuracy and transient response). The objective function is given by (2.8) and the step size of the gradient estimator in the extremum seeking learning algorithm (3.7) is chosen as $\tau = 10^{-8}$. Furthermore, the adopted optimizer gain γ and the initial parameter vector u_0 for all the parametrizations of $k_i(t)$ are given in Table 3.1.

Table 3.1: The optimization gain γ, the initial parameter vector u_0, the error at T_A and the iteration index k.

	γ	u_0^T	e_{T_A} [m]		k (experiments)	
Basis function	*case a) and b)*	*case a) and b)*	*case a)*	*case b)*	*case a)*	*case b)*
Step-like	350000	$[u^{(2)} \ \ u^{(3)}] = [7 \ 2]$	$8.5 \cdot 10^{-10}$	$1.7 \cdot 10^{-10}$	9 (27)	11 (33)
Fourier	15000	$[A \ \delta] = [8 \ 2]$	$8.4 \cdot 10^{-10}$	$6.1 \cdot 10^{-10}$	22 (66)	90 (270)
Linear	100000	$[u^{(2)} \ \ u^{(3)}] = [7 \ 2]$	$1.7 \cdot 10^{-10}$	$2.7 \cdot 10^{-10}$	8 (24)	9 (27)
$k_i = 0$	N.A	N.A	$9.3 \cdot 10^{-3}$	$-6.7 \cdot 10^{-3}$	N.A	N.A
k_i =10	N.A	N.A	$-1.4 \cdot 10^{-2}$	$-2.7 \cdot 10^{-2}$	N.A	N.A

Remark 2. The aim is to control the system towards a setpoint with a desired accuracy e_{spec} $= 10^{-9}$. In this section, it is said that convergence is achieved (the objective function (2.8) is effectively minimized), when $e_{spec} = 10^{-9}$ m is attained. Moreover, the parameter e_{T_A} is defined as the error at time instant T_A where the system is required to arrive at the setpoint.

The setpoint error (e_{T_A}) at the time $t = T_A$ of the linear basis parametrization of $k_i(t)$ for both the cases are shown in Table 3.1 and in the setpoint error plot of Figures 3.5 and 3.7. It is illustrated in Table 3.1 and in Figures 3.4-3.7 that optimal transient response (no overshoot) and optimal setpoint accuracy with respect to both friction characteristics are achieved by the learning controller. It can be seen from Figures 3.4 and 3.7, compared to the first iteration, the setpoint accuracy at the iteration where convergence is reached (see Table 3.1), is approximately by a factor 10^6 (on average) improved. Compared to a fixed $k_i = 0$ N/(ms) or $k_i = 10$ N/(ms), the setpoint accuracy with the learning $k_i(t)$ is approximately by a factor 10^7 and 10^8 improved, respectively, see Table 3.1 for the exact values. Note that the setpoint accuracy (e_{T_A}) as given in Table 3.1 holds for the time window [T_A, T]. In the $k_i(t)$ plot of Figures 3.4 and 3.6, it is depicted that in the time window [T_A, T], $k_i(t)$ is set to zero. Due to a zero $k_i(t)$, build up of the control force is avoided, resulting in that the system cannot escape the stick phase ($u_c < F_s$), which eliminates limit cycling and robustness to force disturbances is obtained. When looking at the shape of the obtained $k_i(t)$, it matches the engineering intuition (decaying shape) as described in Section 2.2.1.

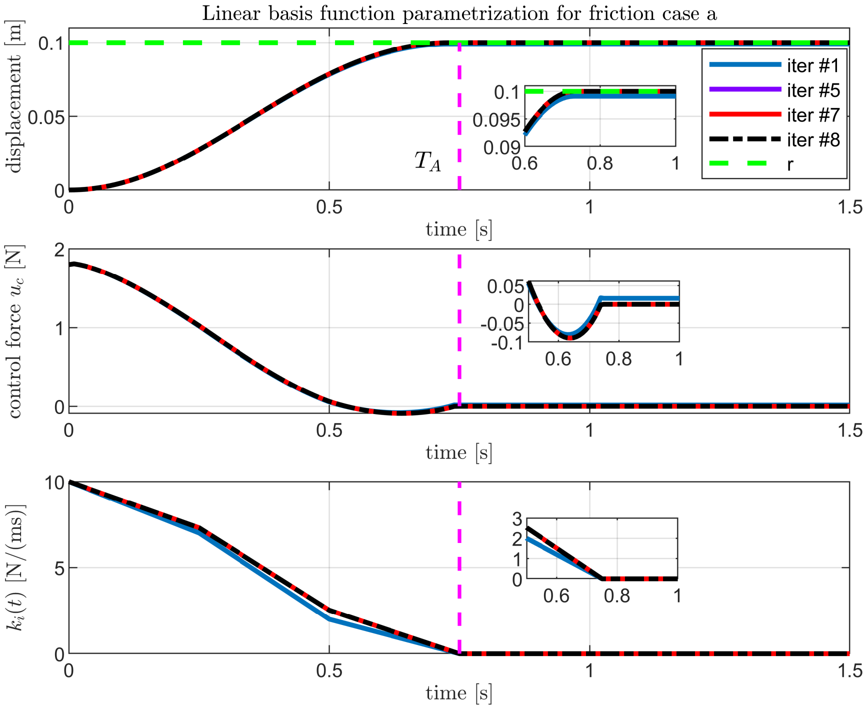

Figure 3.4: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the linear basis function parametrization of $k_i(t)$ with respect to friction case a).

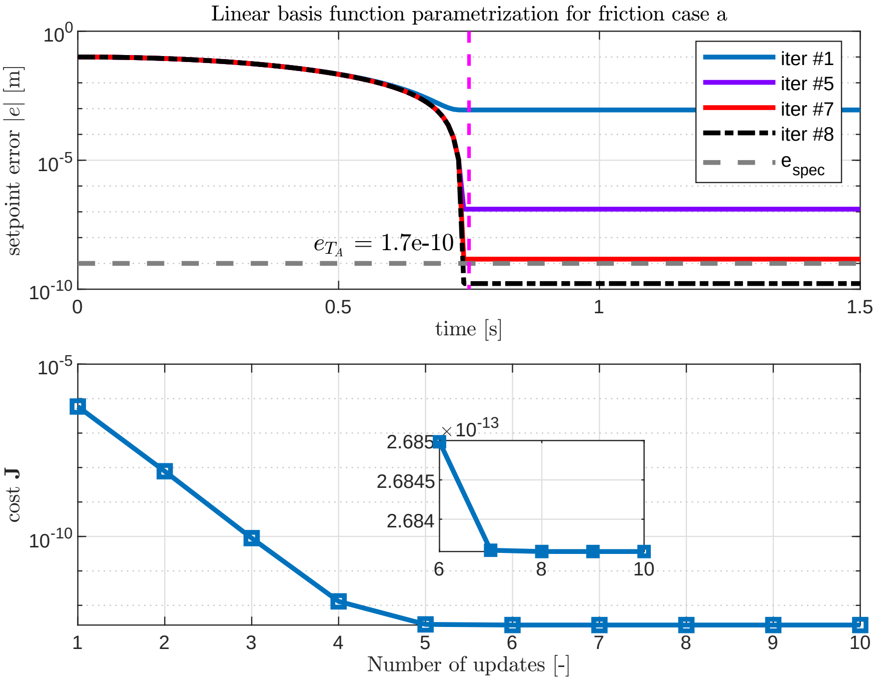

Figure 3.5: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the linear step-like basis function parametrization of $k_i(t)$ with respect to friction case a).

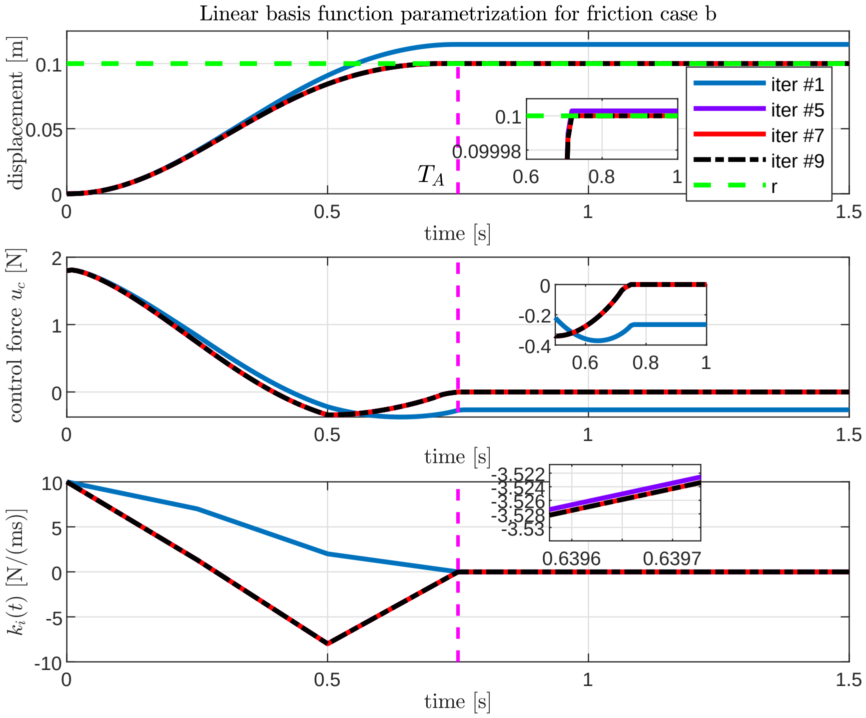

Figure 3.6: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the linear basis function parametrization of $k_i(t)$ with respect to friction case b).

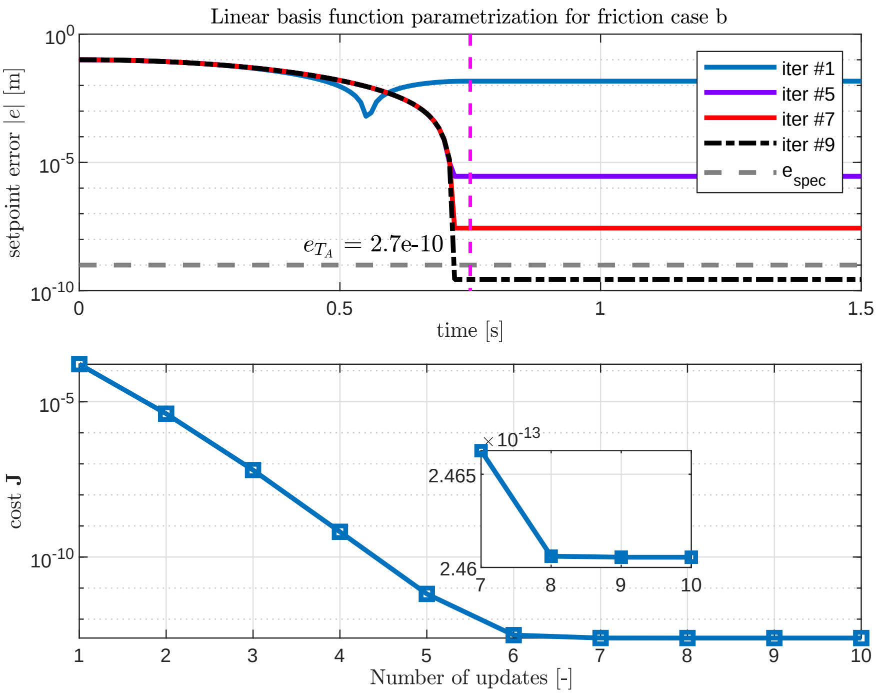

Figure 3.7: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the linear basis function parametrization of $k_i(t)$ with respect to friction case b).

Table 3.1 and the objective function plot of Figures 3.5 and 3.7 show that for friction case a), after eight controller updates, and for friction case b), after nine controller updates the objective function (2.8) is effectively minimized. The evolution of $k_i(t)$, parametrized as a linear basis function, during the iterative process of the learning controller is illustrated in the $k_i(t)$ plot of Figures 3.4 and 3.6 with respect to both friction cases. Considering Figure 3.6, it can be observed that the time-varying integrator gain $k_i(t)$ takes negative values to achieve the optimal performance by the sampled-data ESC framework. From a classical PID-feedback control point-of-view for linear systems (when $F_s = 0$), a negative *constant* integrator gain would almost always yields unstable closed-loop behaviour, see *Assumption 1*. For frictional motion systems with Stribeck effect, a negative *time-varying* integrator gain $k_i(t)$ is needed (temporarily) to counteract the rapid reduction of the friction force due to a severe Stribeck effect. From the displacement time response plot in Figures 3.6, it can be seen that no unstable behaviour occurs. It is also remarkable, that a small change in the obtained optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$, results in a significantly increase of the setpoint accuracy, see the zoomed-in $k_i(t)$ plot in Figures 3.4-3.7.

In the input-output mappings $Q(u)$ (Figures 3.2 and 3.3) in Section 3.2.1, the evolution of the optimized parameters $u^{(2)}$ and $u^{(3)}$ by the learning controller, for different initial parameter vector u_0 are depicted (the magenta colored lines, labelled as 'ESC'). It is demonstrated, that for several initial conditions of the parameter vector u, the optimal u is obtained in the set of optima (dark blue area in Figures 3.2 and 3.3), with respect to both friction cases. So, the classical gradient descent method (3.5) seems a suitable optimization algorithm to solve the optimization problem of finding the minimum of $Q(u)$. The classical gradient descent algorithm only finds local optima, therefore, the resulting optimal settings depends on the chosen values for parameters τ and γ and on the initialization of the integrator gain (the initial parameter vector u_0). Moreover, the extremum seeking algorithm converges to a point (optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$) other than the global minimum. There are two possible causes for this. The first possibility is that a local minimum is found, which can occur for a non-convex optimization problem. The second possibility is that the cost $\mathbf{J}(e)$ (2.8) at a point other than the global minimum is below the threshold value (tolerance) of the time-stepping solver, preventing further convergence to the global minimum. Note that for the initial parameter vector u_0 of the optimized parameters as given in Table 3.1, the time response results of the extremum seeking controller are shown in Figures 3.4-3.7. Regarding other initial parameter vector u_0 only the evolution of the optimized parameters are illustrated in the input-output mappings of Figures 3.2 and 3.3.

Remark 3. In this section, only the results of the linear basis function parametrization of $k_i(t)$ are discussed. The results of the step-like and Fourier basis function parametrizations of $k_i(t)$ are discussed in Appendix A.3. The conclusions drawn in this section for the linear basis function parametrization of $k_i(t)$ also hold for both the step-like and Fourier basis function parametrizations of $k_i(t)$.

3.3 Discussion

The sampled-data extremum seeking control framework is employed to adaptively tune the input u and, therefore, construct the optimal realisation of the parameter vector u (parameters $u^{(2)}$ and $u^{(3)}$) that minimizes the objective function. The input-output mapping is visualized with respect to both friction cases. The simulations results of the input-output mapping show that there is a set of optima (optimal area), which results in optimal setpoint accuracy. It is illustrated that for various initial conditions of the parameter vector u, the obtained optimal u (parameters $u^{(2)}$ and $u^{(3)}$) by the classical gradient descent method is attained in the set of optima (optimal area), with respect to both the friction cases. Therefore, it is concluded that the classical gradient descent method is a suitable optimization algorithm to solve the nonconvex optimization problem at hand. Furthermore, the obtained optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ by the extremum seeking learning controller depends on the initial parameter vector. As a result, the extremum seeking learning controller converges to different optimal settings of the paramters $u^{(2)}$ and $u^{(3)}$ for different initial parameter vectors.

Moreover, application of the PID-based learning controller on the single-sliding mass motion system using ESC achieves, both optimal setpoint accuracy and optimal transient response with respect to both Stribeck effect cases. In case of a strong Stribeck effect, negative optimal settings of the parameter vector u are obtained by the sampled-data ESC framework to obtain optimal performance. Typically, a negative *constant* integrator gain would almost always result in an unstable closed-loop behaviour from a conventional PID-feedback control point-of-view for linear systems. However, for nonlinear systems a negative *time-varying* integrator gain is needed to counteract the rapid reduction of the friction force due to a severe Stribeck effect.

CHAPTER 4

PID-based control of an industrial nano-positioning motion stage

In this chapter, the working principle and the effectiveness of the proposed learning controller are demonstrated through various experiments on an industrial nano-positioning stage. The nano-positioning stage is employed as an experimental setup, which is representative for the R_x-rotation of a sample manipulation stage of an electron microscope.

4.1 Experimental setup

4.1.1 System description

The experimental setup (schematically) depicted in Figures 4.1 and 4.2, is an industrial high-precision positioning stage which represents a sample manipulation stage of an electron microscope [56]. An electron microscope is a type of microscope that uses a beam of accelerated electrons to illuminate a sample and create a magnified image. The function of the sample manipulation stage is to position the sample under the electron beam. This high-precision stage can provide movement in 5 degrees of freedom (DOF), which are the three translations (x, y, z) and two rotations (R_x, R_y). The four DOF (x, y, z and R_y) are considered frictionless. However, the rotation R_x suffers from significant friction, limiting the achievable system performance.

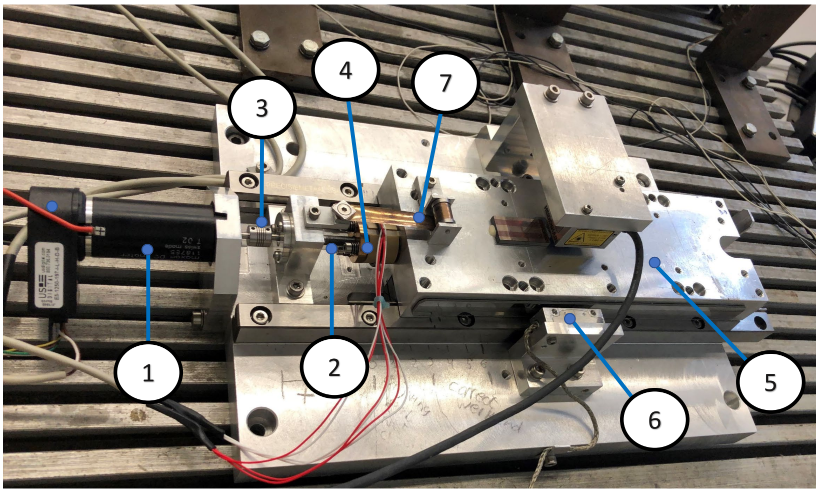

Figure 4.1: The nano-positioning motion stage experimental setup.

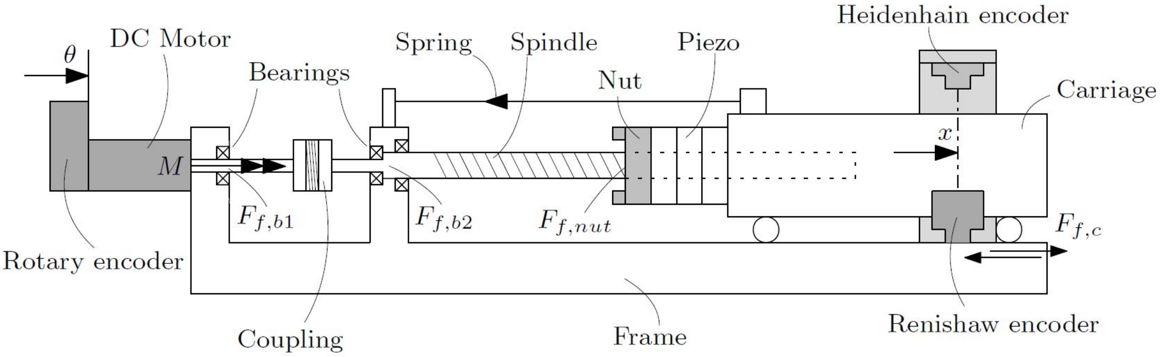

Figure 4.2: Schematic view of the nano-positioning motion stage experimental setup [18].

The nano-positioning motion stage is depicted in Figure 4.1. The experimental setup consists [1, 18] of a DC servo motor (1), which is connected to a spindle (2), by a coupling (3). The coupling drives a nut (4), converting the rotary motion of the spindle to a translational motion of the attached carriage (5). The coupling is stiff in the rotational direction, but flexible in the translational direction. The position of the carriage is measured by a linear Renishaw encoder (6) with a resolution of 1 nm and peak noise level of 4 nm. To eliminate any backlash between the spindle and the nut, the carriage is connected to the fixed world frame via a coiled spring (7).

The experimental setup suffers from frictional effects, induced by different sources of friction, such as the bearings $F_{f,b1}$ and $F_{f,b2}$ supporting the motor axis and the spindle, as depicted in Figure 4.2. Friction is also induced by the contact between the spindle and the nut $F_{f,nut}$, as illustrated in Figure 4.2. The friction $F_{f,nut}$ is a lubricated contact, which caused the Stribeck effect. Therefore, the friction $F_{f,nut}$ contributes dominantly to the total friction force in (2.2) compared to the other friction sources [1].

The frictional nano-positioning motion stage is hard to control accurately towards a setpoint, due to the fact that friction is generally unknown or uncertain. Classical PID has severe performance limitations such as limited setpoint accuracy, or stick-slip behaviour. Therefore, the proposed PID-based learning controller, described in Section 3.1, is implemented on the nano-positioning motion stage to investigate the achievable performance benefits in practice, in terms of setpoint accuracy and transient response. The motion profile is designed such that the setup follows a third-order reference trajectory to the setpoint, which is a displacement of 1 μm. The prescribed velocity to reach this setpoint is 0.1 m/s. The period time T of the motion profile is set to 3 s and T_A is 1.5 s.
The goal is to control the system towards a desired setpoint within an accuracy of ± 10 nm in the time window $[T_A \;\; T]$ (at the setpoint).

4.1.2 Classical PD vs PID-control experiment

To illustrate the performance limitations of using a conventional P(I)D controller, experiments are performed on the nano-positioning motion stage experimental setup, employing a classical P(I)D controller given by (2.4). The PID controller gains are set to $k_p = 10^7$ N/m , $k_d = 2{\cdot}10^3$ Ns/m and $k_i = 10^8$ N/(ms) [1]. These PID controller gains are obtained by linear loop-shaping techniques and satisfy the linear stability criterion, see *Assumption 1.*

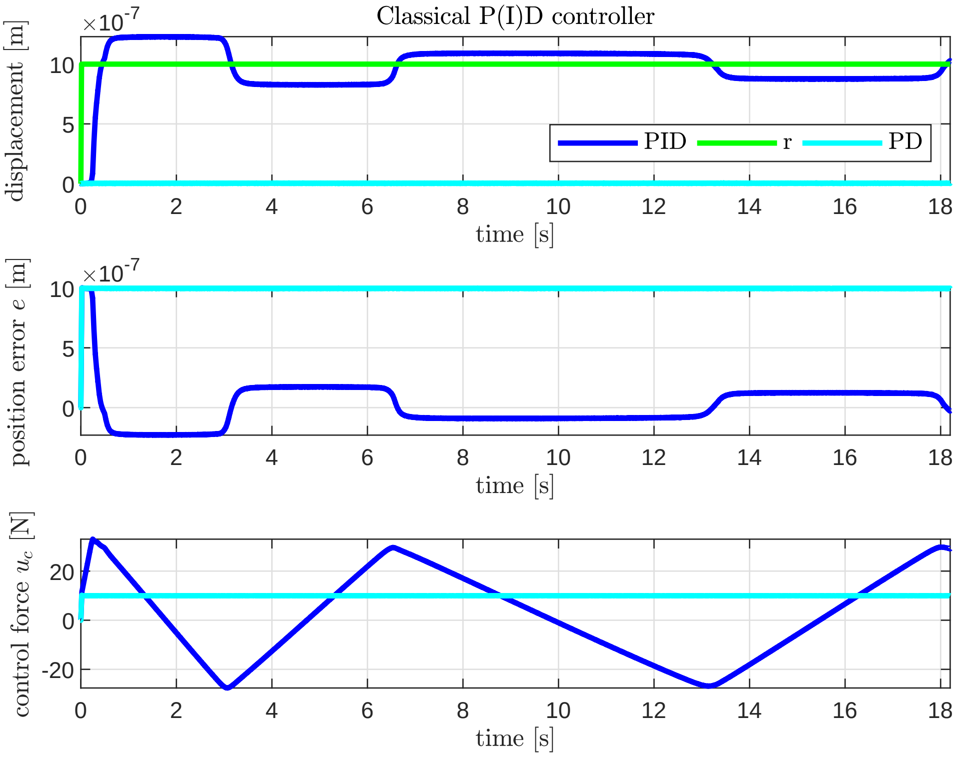

Figure 4.3: Measured time responses using classical P(I)D controller.

Figure 4.3 shows that, when using a PD controller, the system stays in stick. This is a result of not enough control force having built up to overcome the static friction F_s. When using a PID controller with a constant integrator gain, limit cycling occurs due to the presence of Stribeck friction. Hence, the performance limitations of a classical P(I)D are also visible on the experimental setup. To cope with these performance limitations the proposed PID controller with a time-varying integrator gain is implemented on the nano-positioning motion stage experimental setup in the next section.

4.2 PID-based control design

4.2.1 Control design

The working principle and the effectiveness of the PID controller (2.10) with a time-varying integrator gain $k_i(t)$ is demonstrated through an experiment on the nano-positioning motion stage experimental setup. The time-varying integrator gain $k_i(t)$ is parametrized by linear basis functions (2.16) and the parameter vector u is given by (2.17). The adopted motion profile is defined in Section 4.1.1. The number of elements p in the parameter vector u equals 6, therefore, $t_s = \frac{T}{p} = 0.5$ s. The proportional gain k_p and the derivative gain k_d are set at $k_p = 10^7$ N/m and $k_d = 2{\cdot}10^3$ Ns/m. The parameter $u^{(1)}$ in the parameter vector u is fixed at 10^8 N/(ms) and $u^{(j)} = 0$ for all $j = 4, ..., p$. In this experiment, the parameters $u^{(2)}$ and $u^{(3)}$ have the values $8{\cdot}10^7$ N/(ms) and $7{\cdot}10^7$ N/(ms), respectively. Note that this experiment is performed without the application of the extremum seeking learning algorithm.

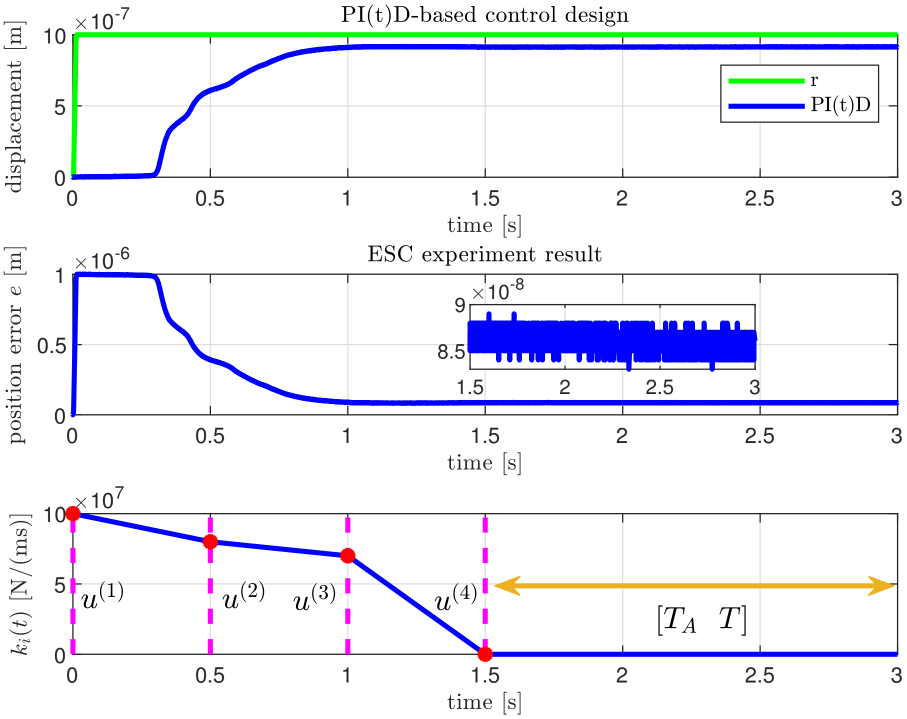

Figure 4.4: PID controller with a time-varying integrator gain.

Figure 4.4 shows that the PID controller with the time-varying integrator gain design, results in significantly improved setpoint accuracy, enhanced transient performance and elimination of limit cycling. So, the proposed $k_i(t)$ design is also effective on the nano-positioning motion stage experimental setup. However, the setpoint accuracy that is achieved with these settings of the parameters $u^{(2)}$ and $u^{(3)}$ is approximately 85 nm, which is not the desired setpoint accuracy of ± 10 nm in the time window $[T_A \quad T]$. So, in order to achieve 1) optimal setpoint accuracy, and 2) optimal transient response by minimizing overshoot, the unknown optimal setting of the parameters $u^{(2)}$ and $u^{(3)}$ have to be found. The optimal settings for $u^{(2)}$ and $u^{(3)}$ can be adaptively find by the proposed learning algorithm, described in Section 3.1. In Section 4.3.2, the extremum seeking learning PID controller is employed and results of the PID controller with the application of the extremum seeking learning algorithm are demonstrated.

4.2.2 Input-output mapping

In order to perform appropriate experiments on the experimental setup using the learning controller, first, the optimization problem is analyzed by visualizing the unknown experimental input-output mapping. The experimental input-output mapping is given by (3.1). It is desirable to visualize the input-output mapping (3.1) using the time-varying integrator gain design on the experimental setup to determine which optimization algorithm can be used to solve the optimization problem of finding the minimum of $Q(u)$. Furthermore, the initial parameter vector u_0 (parameters $u^{(2)}$ and $u^{(3)}$) can be chosen more accurately (close to the optimum), which results in faster convergence of the extremum seeking controller. Another benefit of visualizing the input-output mapping is to support the analysis of the results of the extremum seeking learning controller.

The input-output mapping $Q(u)$ of the experimental setup, for the proposed linear basis function parametrization of the time-varying integrator $k_i(t)$, is visualized in Figures 4.5 and 4.6. The number of elements that have to be optimized is chosen as $p_o = 2$ (same as in the simulation study). The weights of the objective function (2.8) are chosen as $w_1 = 0$ and $w_2 = 1$. The weight w_1 is set to zero, because obtaining optimal setpoint accuracy is more of interest, compared to transient performance (minimized overshoot). For the input-output mapping in Figures 4.5 and 4.6, the numerical values from Section 4.2.1 and the motion profile as defined in Section 4.1.1 are adopted. In the input-output mapping experiment, measurements are performed for multiple combinations of the parameters $u^{(2)}$ and $u^{(3)}$ in the linear basis parametrization of the time-varying integrator gain $k_i(t)$ given by (2.16-2.17).

The measured combinations of the parameters $u^{(2)}$ and $u^{(3)}$ are depicted by red squares in Figure 4.5. Moreover, to improve the visualization of the input-output mapping $Q(u)$ in Figure 4.5, the cost $\mathbf{J}(e)$ (2.8) in Figure 4.6 is saturated. All values of the cost $\mathbf{J}(e)$ (2.8) that are larger than or equal to $1.5 \cdot 10^{-10}$ m^{2}s correspond to the yellow color. Furthermore, the lowest cost $\mathbf{J}_{min}$ is indicated by a cyan asterisk marker in both Figures 4.5 and 4.6. It is demonstrated in Figure 4.6 that a high integrator gain action (e.g., $u^{(2)} = u^{(3)} = 10^8$) and a low integrator gain action (e.g., $u^{(2)} = u^{(3)} = 0$) both result in significant overshoot and undershoot, respectively. In Figure 4.7, the designs of a high and low time-varying integrator gain action are illustrated. The purpose to consider these two cases is that between the settings of the high and low integrator gain there are cases (settings of the parameters $u^{(2)}$ and $u^{(3)}$), which achieve the desired setpoint accuracy. From Figure 4.6, it can be observed that between the high and low integrator gain settings there exist a set of optima (combinations $u^{(2)}$ and $u^{(3)}$ which results in appropriate setpoint accuracy).

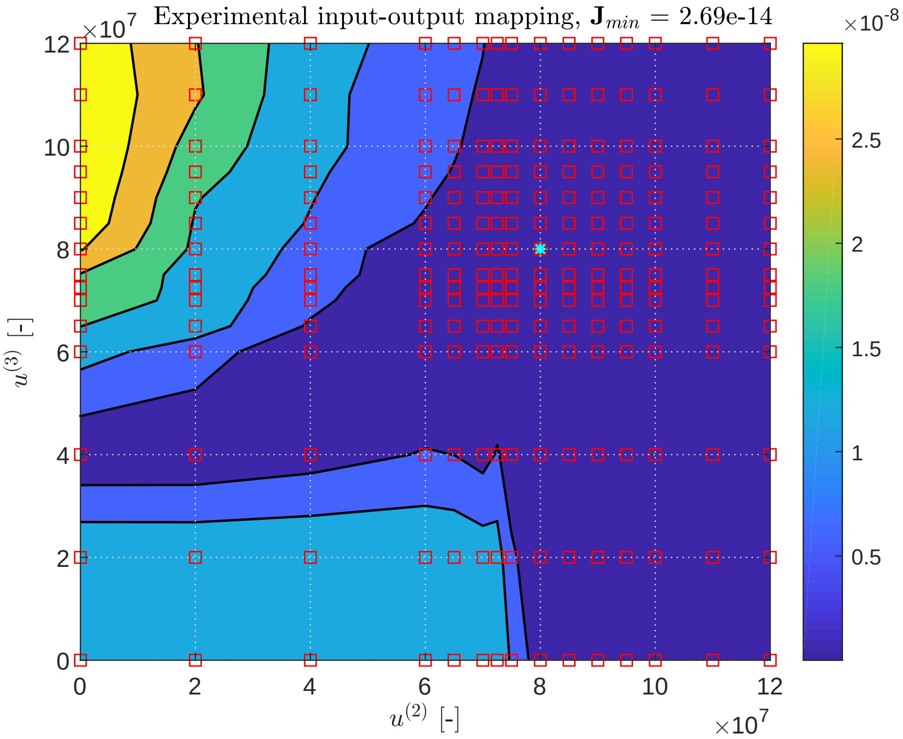

Figure 4.5: Contour plot of the input-output mapping.

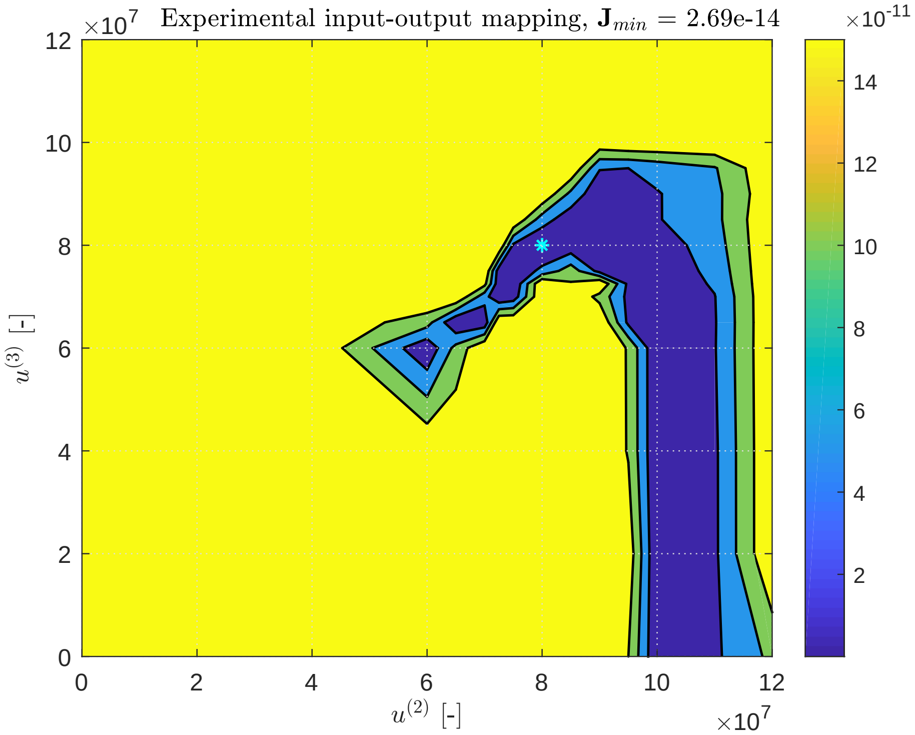

Figure 4.6: Saturated contour plot of the experimental input-output mapping.

From Figure 4.6, it is concluded that mainly combinations of a high $u^{(2)}$ (10^8) and a lower $u^{(3)}$ ($\approx < 9 \cdot 10^7$) result in good performance (appropriate setpoint accuracy), satisfying the intuition of the (decreasing) time-varying integrator gain $k_i(t)$. Intuitively, a high $u^{(2)}$ (i.e., 10^8) is desired to overcome the static friction F_s and a lower $u^{(3)}$ ($\approx < 9 \cdot 10^7$) is needed to avoid overcompensation of friction. It can also be seen that a very high $u^{(2)}$ ($\approx > 10^8$), independently of the value of $u^{(3)}$ results in significant overshoot (bad performance).

From the input-output mapping in Figure 4.6, it can be observed that the optimization problem at hand is nonconvex because there exist combinations of points (optima) in the optimal area (dark blue area in Figure 4.6) for which the line segment between the points lies outside the optimal area. Moreover, for $6 \cdot 10^7 \leq u^{(2)} \leq 8 \cdot 10^7$ and $6 \cdot 10^7 \leq u^{(3)} \leq 8 \cdot 10^7$ there exist multiple regions of optima, both local optima and global optimum. In general, the classical gradient descent optimization algorithm (3.5) is not suitable to solve a nonconvex optimization problem (it can only find local optima). Nevertheless, it is presented in Section 4.3.2 that the gradient descent optimization algorithm (3.5) is sufficient to solve the optimization problem at hand, and that for several initial values of the parameter vector u the desired setpoint accuracy is achieved.

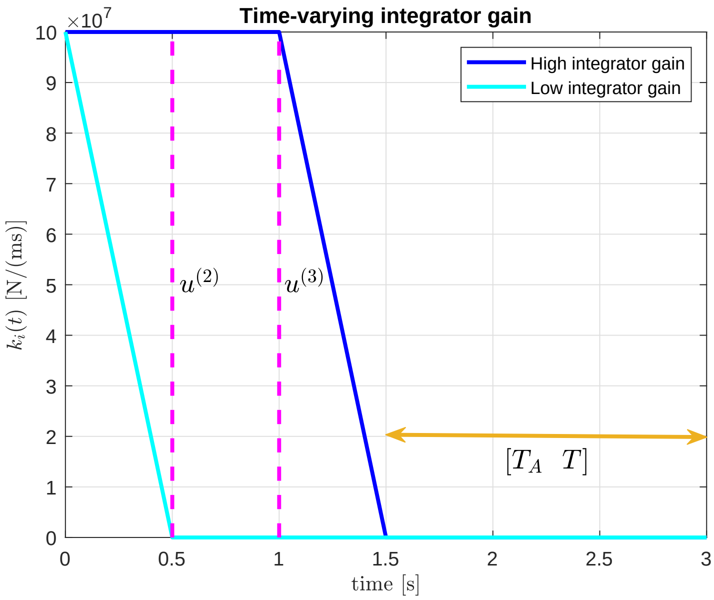

Figure 4.7: Designs of a high ($u^{(2)} = u^{(3)} = 10^8$) and low ($u^{(2)} = u^{(3)} = 0$) $k_i(t)$.

4.3 PID-based learning control design

4.3.1 Extremum seeking control design

In Section 4.2.1, it has been demonstrated that application of the PID controller where $k_i(t)$ is parametrized by linear basis functions parametrization on the experimental setup results in significantly improved setpoint accuracy and transient performance. However, the desired setpoint accuracy is not achieved due to the fact that the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ are unknown. Therefore, in this section, the PID controller where $k_i(t)$ is parametrized by linear basis functions is extended with the proposed learning algorithm presented in Section 3.1. The PID-based learning controller adaptively optimizes the parameters $u^{(2)}$ and $u^{(3)}$ in the parameter vector u, resulting in an optimized $k_i(t)$ which achieves optimal setpoint accuracy. The adopted numerical values of the PID controller and the time-varying integrator gain design are given in Section 4.2.1. Recall that the weighting factors of the objective function (2.8) are set at $w_1 = 0$ and $w_2 = 1$. The weight w_1 is set to zero, because, at first, the focus lies more on optimal setpoint accuracy and less on transient performance. The adopted motion profile is given in Section 4.1.1. At the setpoint, in the time window $[T_A \quad T]$, the aim is to achieve a setpoint accuracy of ± 10 nm. A requirement for the extremum seeking control framework is that the system has to be *re-initialized* after every setpoint operation which is accomplished by an automatic hooming procedure.

4.3.2 PID-based learning control results

The first extremum seeking learning controller experiment results are depicted in Figures 4.8 and 4.9. In this experiment, the step size of the gradient estimator τ and optimizer gain have the values $3 \cdot 10^6$ and $\gamma = 6.5{\cdot}10^{22}$, respectively. The initial parameter vector is set to $u_0^T = [u^{(2)} \quad u^{(3)}] = [0 \quad 10^8]$. Figures 4.8 and 4.9 show that application of the extremum seeking learning controller achieves the desired setpoint accuracy of ± 10 nm, and transient response is improved (despite $w_1 = 0$), compared to a classical P(I)D controller.

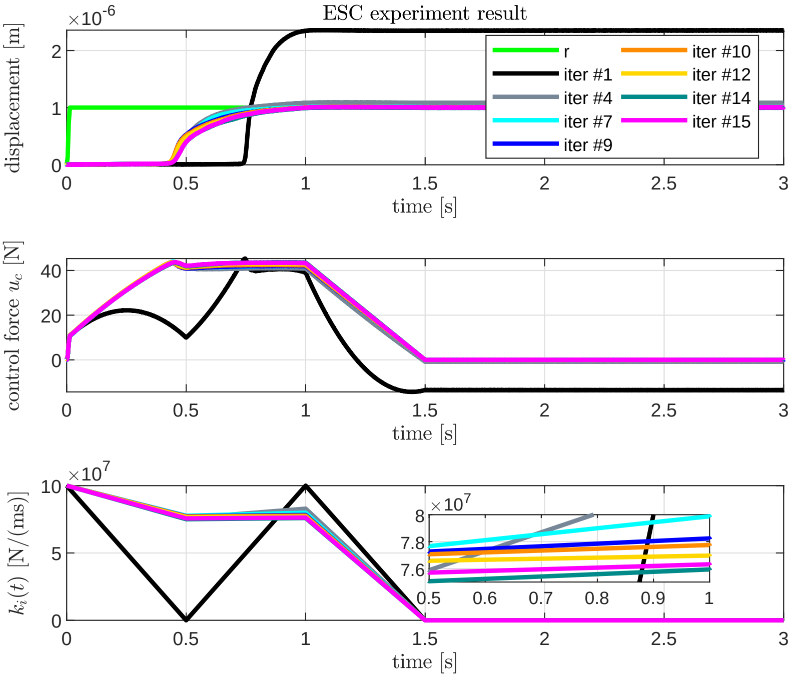

Figure 4.8: Measured time responses of the displacement, control force u_c and $k_i(t)$ with initial parameter vector $u_0^T = [u^{(2)} \quad u^{(3)}] = [0 \quad 10^8]$ using ESC.

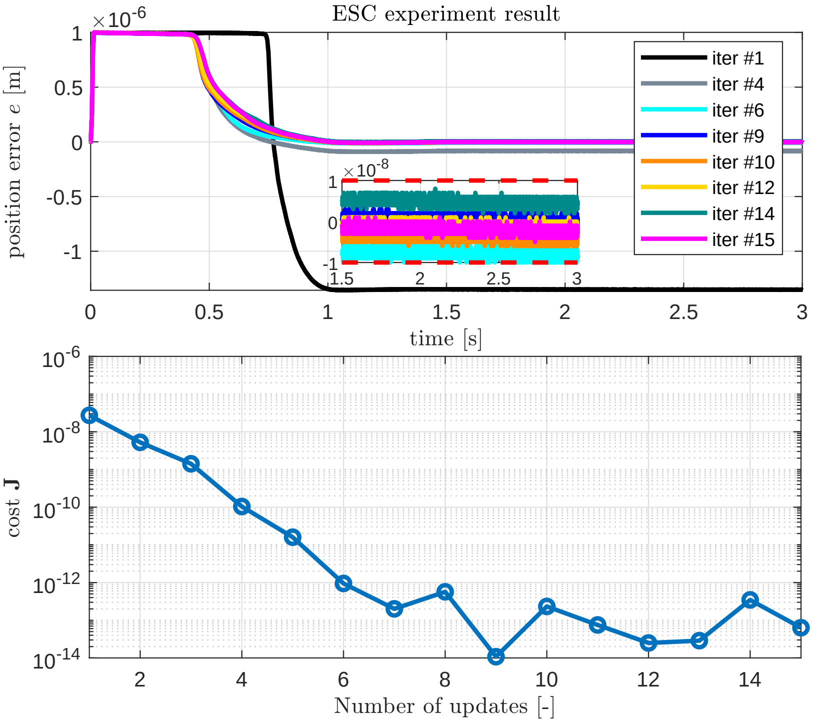

Figure 4.9: Measured time responses of the error e and objective function $\mathbf{J}$ for $k_i(t)$ with initial parameter vector $u_0^T = [u^{(2)} \quad u^{(3)}] = [0 \quad 10^8]$ using ESC.

The top subplot of Figure 4.8 shows that for the first iteration (input) the system is in the stick phase for the first 0.75 s while for other iterations it takes approximately 0.45 s to escape the stick phase. This difference is caused by the fact that not enough control force is built up to overcome the static friction F_s (approximately 45 N). It can be seen from the top subplot of Figure 4.8 that a combination of a very low $u^{(2)}$ and high $u^{(3)}$ (first iteration) results in significant overshoot. This is induced by a larger integrator buffer (state x_3), resulting in a larger integrator force and, therefore, in significant overshoot, compared to the other iterations. It is also visible that the integrator state x_3 keeps decreasing instead of converging to a set, which could lead to unstable behaviour. Moreover, it can be concluded from the control force and time-varying integrator gain plot in Figure 4.8, the higher parameter $u^{(2)}$, the faster the system breaks free from the stick phase.

The top subplot in Figure 4.9, demonstrates that starting from iteration six, the desired performance is achieved and that the cost $\mathbf{J}(e)$ (2.8) is minimized. The cost $\mathbf{J}(e)$ fluctuates after iteration six. In practice this means that the extremum seeking learning controller is close to an optimum value, and thus, fluctuations around an optimal value are observed, caused by the chosen dither size τ and the optimizer gain γ. This problem can be solved by adapting step size τ and optimizer gain γ in the learning algorithm. In case the gradient is (approximately) zero, the extremum seeking learning controller is at an optimum, and the dither size τ and γ can be adjusted accordingly. As a result, the fluctuations around the optimal value decrease and convergence takes place.

Moreover, the proposed learning algorithm can be employed for calibration procedures in industrial applications, or as a continuous online adapting control tool in industrial applications to deal with unknown friction and time-varying disturbances. In case of a calibration procedure, the extremum seeking learning algorithm can be stopped when the optimal settings are obtained (the desired performance is achieved). Furthermore, the lower and upper bound of each error response in the zoomed-in plot of Figure 4.9 is related to the peak noise level of the encoder, which is approximately ± 4 nm, as given in Section 4.1.1.

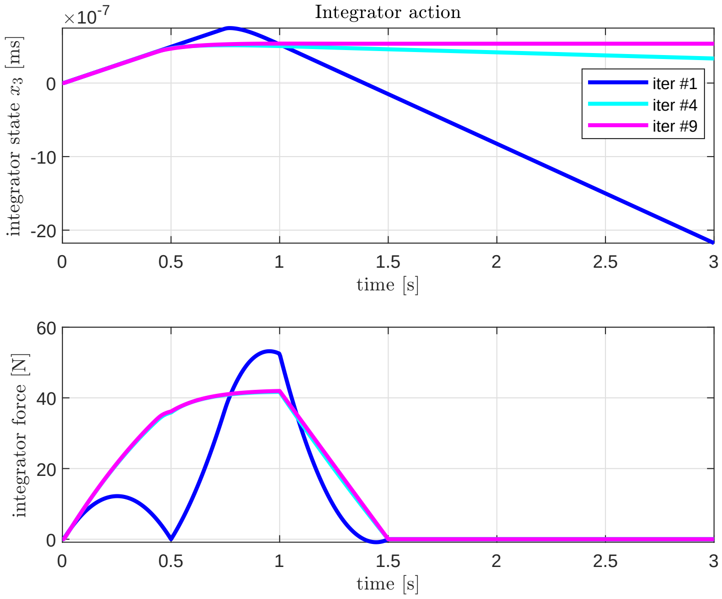

Figure 4.10: Integrator force with initial parameter vector $u_0^T = [u^{(2)} \;\; u^{(3)}] = [0 \;\; 10^8]$.

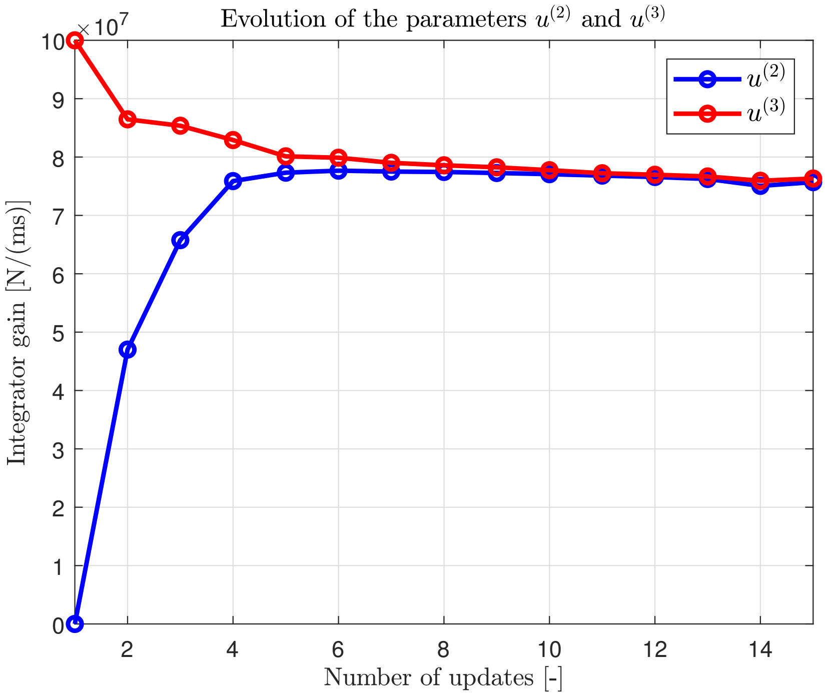

Figure 4.11: Evolution of the parameters $u^{(2)}$ and $u^{(3)}$ for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [0 \;\; 10^8]$.

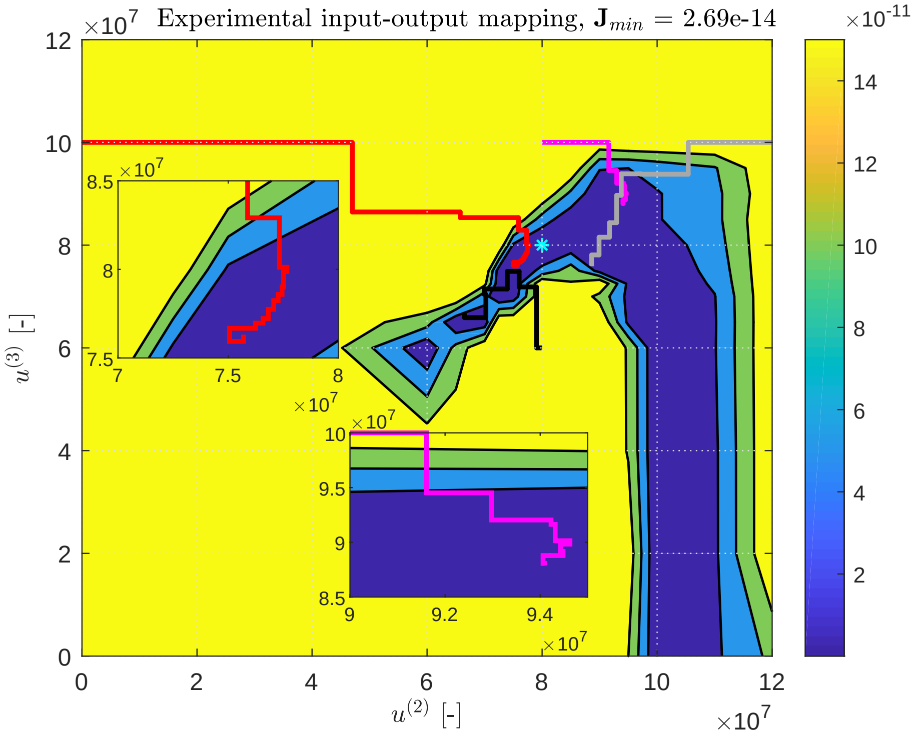

Figure 4.12: Saturated contour plot of the experimental input-output mapping. The evolution of $k_i(t)$ are depicted for initial parameter vectors $u_0^T = [u^{(2)} \;\; u^{(3)}] = [0 \;\; 10^8]$ (—), $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ (—), $u_0^T = [u^{(2)} \;\; u^{(3)}] = [12 \cdot 10^7 \;\; 10 \cdot 10^7]$ (—) and $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 6 \cdot 10^7]$ (—).

The evolution of $k_i(t)$ (the parameters $u^{(2)}$ and $u^{(3)}$) for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [0 \;\; 10^8]$ during the iterative process of the extremum seeking learning controller is illustrated in Figures 4.8, 4.11 and in the input-output mapping of Figure 4.12. The obtained optimal settings by the extremum seeking learning PID controller for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [0 \;\; 10^8]$ are $u_{opt}^T = [u^{(2)} \;\; u^{(3)}] = [7.57 \cdot 10^7 \;\; 7.63 \cdot 10^7]$. Figure 4.12 shows that the obtained optimal settings of the extremum seeking learning controller are in the set of optima (dark blue area in Figure 4.12), indicated by red line. For these settings, the required setpoint accuracy is met. Therefore, the classical gradient descent method is a sufficient optimization algorithm to find suitable settings of the parameters $u^{(2)}$ and $u^{(3)}$ for which the required setpoint accuracy is obtained. Additionally, it is shown that a slight change in the parameters $u^{(2)}$ and $u^{(3)}$ results in different time responses. Therefore, the optimal performance depends on the combination of the parameters $u^{(2)}$ and $u^{(3)}$.

In Figures 4.13 and 4.14, the time responses of an experiment with the learning controller for an initial parameter vector $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ are presented. For this experiment, the step size of the gradient estimator τ and optimizer gain are set to $3 \cdot 10^6$ and $\gamma = 1.2 \cdot 10^{22}$, respectively. The same conclusions are drawn as from the previous extremum seeking learning controller experiment depicted in Figures 4.8 and 4.9. The extremum seeking learning controller achieves the desired position error accuracy band of 10 nm, and overshoot is significantly reduced. Furthermore, the cost $\mathbf{J}(e)$ (2.8) is significantly reduced as well fluctuates, starting from iteration eight. Moreover, in Figure 4.12 it is illustrated that the obtained optimal settings by the extremum seeking learning PID controller for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ are in the optimal area (dark blue area in Figure 4.12), see the magenta line. The obtained optimal settings are $u_{opt}^T = [u^{(2)} \;\; u^{(3)}] = [9.41 \cdot 10^7 \;\; 8.79 \cdot 10^7]$.

It can be observed that these optimal settings (the parameters $u^{(2)}$ and $u^{(3)}$) are different from the settings found previously, because the initial conditions are different. The optimum found using the classical descent gradient method, which can only find local optima, depends on the initialization of the parameter vector u_0. In case of employing global optimization methods, the resulting time-varying integrator gain (settings of the parameters $u^{(2)}$ and $u^{(3)}$) is independent on the initialization of the integrator gain, since the global optimum is always found. However, global optimization methods have two weaknesses. First, they can be slow in reaching the global optimum with high accuracy, and second, many evaluations are spend exploring local optima [57]. Furthermore, the optimum that is found by the extremum seeking learning algorithm also depends on the step size τ and the optimizer gain γ, which have different values for both the experiments. These parameters determine which optimum is attained and how fast the optimum is reached (convergence rate). Additionally, the evolution of the parameters $u^{(2)}$ and $u^{(3)}$ for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ with respect to each update of the extremum seeking learning controller is illustrated in Figure 4.15.

More experiments with the extremum seeking learning controller are performed on the experimental setup for different initial parameter vectors ($u^{(2)}$ and $u^{(3)}$). The time responses of these experiments are presented in Figures A.14-A.17 of Appendix A.4. For these experiments the same conclusions are drawn as from the previous experiments described in this section. The evolution of $k_i(t)$ for various initial vectors is shown in Figure 4.12. It can be seen that for other initial parameter vectors the optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ also end up in the optimal area (dark blue area in Figure 4.12). Therefore, the classical gradient descent method (3.5) is a sufficient optimization method to find appropriate settings of the parameters $u^{(2)}$ and $u^{(3)}$ for which the desired setpoint accuracy is achieved.

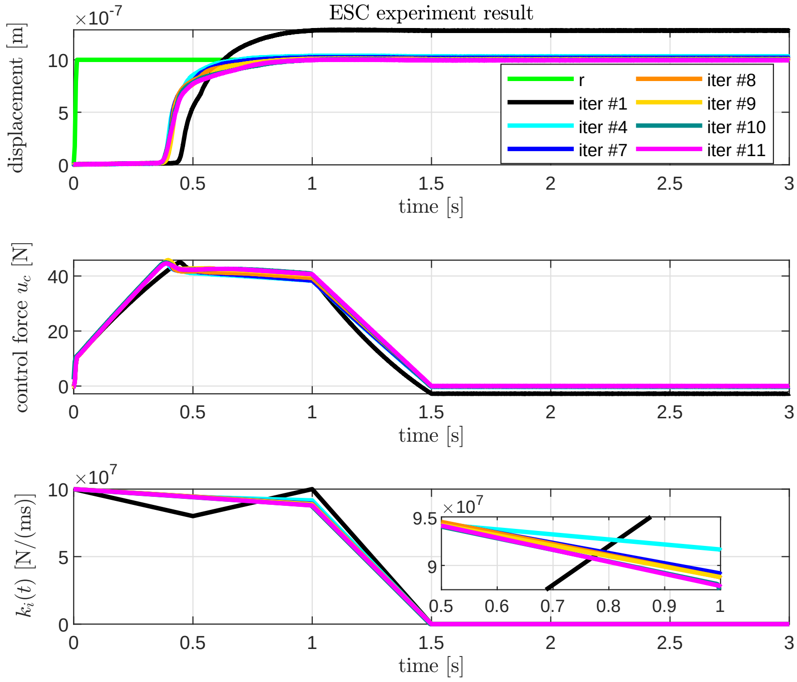

Figure 4.13: Measured time responses of the displacement, control force u_c and $k_i(t)$ with initial parameter vector $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ using ESC.

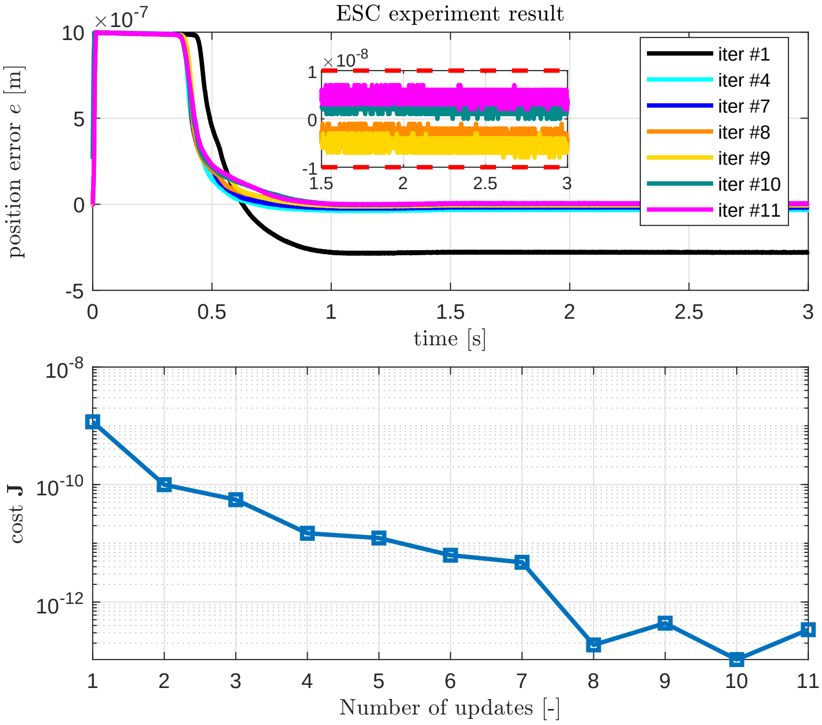

Figure 4.14: Measured time responses of the error e and objective function **J** for initial parameter vector $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$ using ESC.

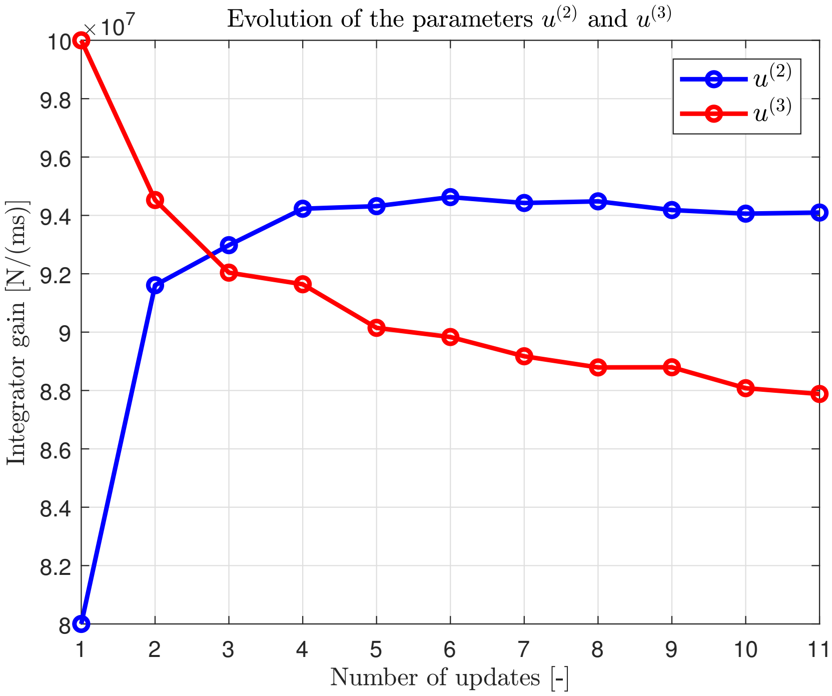

Figure 4.15: Evolution of the parameters $u^{(2)}$ and $u^{(3)}$ for $u_0^T = [u^{(2)} \;\; u^{(3)}] = [8 \cdot 10^7 \;\; 10 \cdot 10^7]$.

4.4 Discussion

Various experiments are performed on an industrial nano-positioning motion stage experimental setup using the linear basis function parametrization of $k_i(t)$. The experiment with the classical PID controller shows occurrence of limit cycling due to the presence of Stribeck friction. From the input-output mapping experiment, it can be observed that mainly combinations of a high $u^{(2)}$ (10^8) and a lower $u^{(3)}$ ($\approx < 9 \cdot 10^7$) result in good performance (appropriate setpoint accuracy), satisfying the intuition of the (decreasing) time-varying integrator gain $k_i(t)$. Moreover, it can be observed that the optimization problem at hand is nonconvex, since there exist combinations of points (optima) in the optimal area for which the line segment between the points lies outside the optimal area. In addition, there exist multiple regions of optima, including both local optima and a global optimum.

From the extremum seeking learning PID controller experiments, it is shown that application of the extremum seeking learning PID controller for various initial parameter vectors achieves the desired setpoint accuracy of ± 10 nm, and enhanced transient response, compared to a classical PID controller. Therefore, the classical gradient descent optimization method is a sufficient optimization method to find the suitable settings of the parameters $u^{(2)}$ and $u^{(3)}$ for which the desired setpoint accuracy is achieved. It has been illustrated that when the extremum seeking learning PID controller is close to an optimum value and the extremum seeking algorithm is still active, it results in fluctuations around the optimal value. This is caused by the chosen dither size τ and the optimizer gain γ. The obtained optimal settings of the parameters $u^{(2)}$ and $u^{(3)}$ are located in the optimal area of the input-output mapping, but for different initial parameter vectors the obtained optimal settings are different. So, the resulting time-varying integrator gain (the parameters $u^{(2)}$ and $u^{(3)}$), as determined by the extremum seeking learning algorithm, depends on the initialization of the integrator gain, the optimizer gain γ and the dither size τ of the gradient estimator.

CHAPTER 5

Conclusions and Recommendations

The conclusions drawn from this research, and recommendations for future research are presented in this chapter.

5.1 Conclusions

The vast majority of repetitive industrial motion systems with friction employ conventional PID-based control. For motion systems with *unknown* Stribeck friction, classical PID-based control results in severe performance limitations, such as stick-slip limit cycling and non-zero steady-state position errors, mainly induced by the integrator action. The performance of the motion systems highly depends on its unknown, time- and position-dependent friction characteristics. To deal with this kind of complicated friction characteristics, a PID-based learning controller with a time-varying integrator gain is presented to obtain optimal setpoint positioning accuracy and enhanced transient response by minimizing overshoot, thereby eliminating the performance limitations of the conventional PID controller.

The time-varying integrator gain is parametrized by a set of basis functions, and a to-be-constructed parameter vector. The optimal realisation of the parameter vector is typically unknown due to the fact that friction is generally unknown or uncertain. Therefore, the specific time-varying integrator gain tuning is adaptively obtained with the use of a *model-free* sampled-data extremum seeking approach. The working principle and effectiveness of the learning PID-based controller are demonstrated by a simulation study, and by an experimental case study, using a high-precision positioning stage application that suffers from friction.

Using a classical PID controller on the experimental setup results in limit cycling, due to the presence of Stribeck friction. From both simulation and experimental study results, it is concluded that the adaptively obtained PID-based learning controller by the ESC framework for various initial parameter vectors and different friction characteristics achieves 1) a significantly improved setpoint accuracy, 2) an enhanced transient response by minimizing overshoot, compared to a classical P(I)D controller. The resulting time-varying integrator, determined by the learning algorithm, depends on the initialization of the integrator gain, the optimizer gain γ and the step size τ of the gradient estimator.

The simulation study results show that in case of a strong Stribeck effect, negative optimal settings of the parameter vector are obtained by the sampled-data ESC framework to achieve optimal performance. Typically, a negative *constant* integrator gain would almost always result in an unstable closed-loop behaviour from a conventional PID-feedback control point-of-view for linear systems. However, for frictional motion systems with Stribeck effect, a negative *time-varying* integrator gain is needed to resist the rapid reduction of the friction force due to a severe Stribeck effect.

The simulation and experimental visualizations of the input-output mapping show that there exist regions of optima. It is illustrated that for various initial conditions of the parameter vector, the obtained optimal parameter vector is attained in the region of optima. Therefore, it is concluded that the classical gradient descent method is a suitable optimization algorithm to solve the nonconvex optimization problem at hand because finding local optima is sufficient to achieve the desired system performance.

5.2 Recommendations

In this section, open problems and possible extensions with regards to the research results are listed:

(i) The classical gradient descent optimization algorithm only finds local optima. The resulting time-varying integrator gain, as determined by the learning algorithm, results in the desired setpoint accuracy and significantly reduced overshoot by finding local optima. However, in case of more parameters in the parameter vector have to be optimized, it may be that global optimization methods such as DIRECT, or Shubert have to be employed to find the global optimum, in order to achieve the desired system performance.

(ii) From the ESC experimental results, it is concluded that when the extremum seeking learning controller is close to an optimum value and the extremum seeking algorithm is still active, it results in fluctuation around the optimal value. This is caused by the chosen dither size τ and the optimizer gain γ. The dither size τ determines the size of the fluctuation around the optimal value. The optimizer gain γ determines how fast the extremum seeking learning controller reaches the optimal value. It is possible that when the dither size τ is chosen to be small, but the optimizer gain γ is chosen to be large, it still results in overshooting the optimal value due to a large input update, caused by the large optimizer gain γ. This issue can be dealt with by extending the extremum seeking algorithm with an adaptation mechanism for τ and γ. That is, when the extremum seeking learning controller is close to the optimum (gradient is approximately zero), the dither size τ and optimizer gain γ can be adapted accordingly. As a result, the fluctuations around the optimal value become small and convergence takes place.

(iii) From the ESC simulation results, it is concluded that, for a strong Stribeck effect case, the extremum seeker finds negative optimal values for the time-varying integrator gain. For linear systems, a negative *constant* integrator gain would often yields in unstable closed-loop behaviour. For frictional motion systems with Stribeck effect, a negative *time-varying integrator* gain is needed to counteract the rapid reduction of the friction force caused by a severe Stribeck effect. The simulation results presented in Section 3.2.2, suggest asymptotic stability of the setpoint. Nevertheless, it has been observed on the experimental setup that a negative *time-varying* integrator gain indeed yield in unstable closed-loop behaviour (out of scope for this report). In literature, both the stability and convergence of the ESC control approach are proven. Using these existing proofs, a stability and convergence analysis of the proposed *time-varying* learning controller approach using ESC may be derived.

(iv) In this research, the integrator gain is only adaptively tuned by the ESC framework. It is possible that tuning other parameters, e.g., proportional and derivative gains, or the complete controller force may also be obtained through data-based learning.

Furthermore, instead of optimizing two parameters in the parameter vector of the time-varying integrator gain, it is possible to optimize only one parameter and fix the other parameters based on intuition and the input-output mapping results. As a result, the optimal performance depends only on one parameter and less experiments are required to obtain the optimal performance. Additionally, the first element in the parameter vector is fixed in both the simulation and the experimental study at values of respectively 10 N/(ms) and 10^8 N/(ms). The impact of the choice of these fixed values on the obtained optimal learning controller, and thus on the system performance, could be investigated.
In future research, these possible extensions can be investigated in order to verify if the proposed learning controller is still effective.

(v) Currently, the proposed extremum seeking learning controller is only applied for a motion profile with only a forward movement. In the future, the effectiveness of the proposed extremum seeking learning controller for a motion profile with a sequential forward and backward movement could be investigated.

(vi) In the experimental input-output mapping not all the points are measured. Between the measured data points, linear interpolation is used to estimate the output. Contour lines between the measured points do not necessarily give a reliable representation of the system behaviour. In the future, more points could be measured to have a more accurate input-output mapping.

Bibliography

[1] R. Beerens, A. Bisoffi, L. Zaccarian, W.P.M.H. Heemels, H. Nijmeijer, and N. van de Wouw (2018), 'Reset integral control for improved settling of motion systems with friction', in Automatica, vol. 107, pp. 483-492.

[2] B. Armstrong-Helouvry, P. Dupont, and C. Canudas de Wit (1994), 'A survey of models, analysis tools and compensation methods for the control of machines with friction', in Automatica, vol. 30, no. 7, pp. 1083-1138.

[3] D. Putra, N. van de Wouw and H. Nijmeijer (2007), 'Analysis of undercompensation and overcompensation of friction in 1DOF mechanical systems', in Automatica, vol. 43, no. 8, pp. 1387-1394.

[4] C. Canudas de Wit, H. Olsson, K. Astrom, and P. Lischinsky (1995), 'A new model for control of systems with friction', in IEEE Transactions on Automatic Control, vol. 40, no. 5, pp. 419-425.

[5] L. Freidovich, A. Robertsson, A. Shiriaev and R. Johansson (2010), 'LuGre-Model-Based Friction Compensation', in IEEE Transactions on Control Systems Technology, vol. 18, no. 1, pp. 194-200.

[6] C. Makkar, G. Hu, W.G. Sawyer and W.E. Dixon (2007), 'Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction', in IEEE Trans. Autom. Control, vol. 52, no. 10, pp. 1988-1994.

[7] H. Olsson and K. Astrom (1996), 'Observer-based friction compensation', in Proceedings 1996 IEEE Conference on Decision and Control, pp. 4345-4350.

[8] L. Marton and B. Lantos (2009), 'Control of mechanical systems with Stribeck friction and backlash', in Systems and Control Letters, vol. 58, no. 2, pp. 141-147.

[9] C. Canudas De Wit and P. Lischinsky (1997), 'Adaptive friction compensation with partially known dynamic friction model', in International Journal of Adaptive Control and Signal Processing, vol. 11, no. 1, pp. 65-80.

[10] A. Amthor, S. Zschaeck and C. Ament (2010), 'High precision position control using an adaptive friction compensation approach', in IEEE Trans. on Autom. Control, vol. 55, no. 1, pp. 274-278.

[11] E. Panteley, R. Ortega and M. Gafvert (1998), 'An adaptive friction compensator for global tracking in robot manipulators', in Systems and Control Letters, vol. 33, pp. 307-313.

[12] J. Na, Q. Chen, X. Ren, and Y. Guo (2014), 'Adaptive prescribed performance motion control of servo mechanisms with friction compensation', in IEEE Transactions on Industrial Electronics, vol. 61, no. 1, pp. 486-494.

[13] N. van de Wouw and R. Leine (2012), 'Robust impulsive control of motion systems with uncertain friction', in Int. J. of Robust and Nonlinear Control, vol. 22, pp. 369-397.

[14] Y. Orlov, R. Santiesteban and L.T. Aguilar (2009), 'Impulsive control of a mechanical oscillator with friction', In Proceedings of the 2009 IEEE American Control Conference, pp. 3494-3499.

[15] R. Hensen, M. van de Molengraft, and M. Steinbuch (2003), 'Friction induced hunting limit cycles: A comparison between the LuGre and switch friction model', in Automatica, vol. 39, no. 12, pp. 2131-2137.

[16] R. Beerens, A. Bisoffi, L. Zaccarian, W.P.M.H. Heemels, H. Nijmeijer and N. van de Wouw (2018), 'Hybrid PID control for transient performance improvement of motion systems with friction', in 2018 Annual American Control Conference (ACC), pp. 539-544.

[17] R. Beerens, H. Nijmeijer, M. Heemels, and N. van de Wouw (2017), 'Set-point Control of Motion Systems with Uncertain Set-valued Stribeck Friction', in IFAC PapersOnLine, vol. 50, no. 1, pp. 2965-2970.

[18] D.T. Fresen (2018), 'Setpoint control of motion systems with set-valued friction', MSc thesis, Eindhoven University of Technology.

[19] A. Bisoffi, M.D. Lio, A.R. Teel and L. Zaccarian (2018), 'Global asymptotic stability of a PID control system with Coulomb friction', in IEEE Transactions on Automatic Control, vol. 63, no. 8, pp. 2654-2661.

[20] L. Iannelli, K.H. Johansson, U.T. Jonsson and F. Vasca (2006), 'Averaging of nonsmooth systems using dither', in Automatica, vol. 42, pp. 669-676.

[21] J.J. Thomsen (1999), 'Using fast vibrations to quench friction-induced oscillations', in Journal of Sound and Vibration, vol. 228, no. 5, pp. 1079-1102.

[22] L. Cao and M.H. Schwartz (2000), 'Stick-slip friction compensation for PID position control', in Proceedings of the American Control Conference, vol. 2, pp. 1078-1082.

[23] L. Hazeleger, R. Beerens and N. van de Wouw (2019), 'A sampled-data extremum-seeking approach for accurate setpoint control of motion systems with friction', in Proceedings of 11^{th} symposium on nonlinear control systems, pp. 1415-1420.

[24] G. Konidaris, S. Osentoski, and P. Thomas (2011), 'Value function approximation in reinforcement leanring using the fourier basis', in AAAI.

[25] D. A. Bristow, M. Tharayil and A. G. Alleyne (2006), 'A survey of iterative learning control', in IEEE Control Systems Magazine, vol. 26, no. 3, pp. 96-114.

[26] R. I. Leine and H. Nijmeijer (2004), 'Dynamics and Bifurcations of Non-Smooth Mechanical Systems', in 1st ed. Berlin: Springer-Verlag.

[27] S.Z. Khong, D. Nešić and M. Krstić, (2016), 'An extremum seeking approach to sampled-data iterative learning control of continuous-time nonlinear systems', in National Science Foundation and the Australian Research Council, vol. 49, no. 18, pp. 962-967,

[28] J. Xu (2011), 'A survey on iterative learning control for nonlinear systems', in International Journal of Control, vol. 84, no. 7, pp. 1275-1294.

[29] J. van de Wijdeven and O.H. Bosgra (2010), 'Using basis functions in iterative learning control: analysis and design theory', in International Journal of Control, vol. 83, no. 4, pp. 661-675.

[30] S. Gunnarsson and M. Norrlof (2001), 'On the design of ILC algorithms using optimization', in Automatica, vol. 37, pp. 2011-2016.

[31] K. Barton and A. Alleyne (2009), 'Norm optimal ILC with time-varying weighting matrices', in American Control Conference, pp. 264-270.

[32] J. Bolder, B. Lemmen, S. Koekebakker, T. Oomen, O. Bosgra, and M. Steinbuch (2012), 'Iterative learning control with basis functions for media positioning in scanning inkjet printers', in IEEE Multi-Conference on Systems and Control, pp. 1255-1260.

[33] J. Bolder and T.A.E. Oomen (2015), 'Rational basis functions in iterative learning control - With experimental verification on a motion system', in IEEE Transactions on Control Systems Technology, vol. 23, no. 2, pp. 722-729.

[34] S. Boyd and L. Vandenberghe (2004),'Convex optimization', in Cambridge University Press.

[35] E. Polak (1997), 'Optimization-algorithms and consistent approximations', in Applied mathematical sciences Springer, vol. 124.

[36] D.R. Jones, C.D. Pertunnen and B.E. Stuckman (1993), 'Lipschitzian optimization without the Lipschitz constant', in Journal of Optimization Theory and Applications, vol. 79, no. 1, pp. 157-181.

[37] B.O. Shubert (1972), 'A sequential method seeking the global maximum of a function', in SIAM Journal on Numerical Analysis, vol. 9, no. 3, pp. 379-388.

[38] M. Volckaert, M. Diehl and J. Swevers (2013), 'Generalization of norm optimal ILC for nonlinear systems with constraints', in Mechanical Systems and Signal Processing, vol. 39, pp. 280-296.

[39] M. Volckaert, A. van Mulders, J. Schoukens, M. Diehl and J. Swevers (2009), 'Model based nonlinear iterative learning control: A constrained Gauss-Newton approach', in 17th Mediterranean Conference on Control and Automation, pp. 718-723.

[40] C. Yin, J. Xu and Z. Hou (2010), 'A High-Order Internal Model Based Iterative Learning Control Scheme for Nonlinear Systems With Time-Iteration-Varying Parameters', in IEEE Transactions on Automatic Control, vol. 55, no. 11, pp. 2665-2670.

[41] C. Chien (2008), 'A Combined Adaptive Law for Fuzzy Iterative Learning Control of Nonlinear Systems With Varying Control Tasks', in IEEE Transactions on Fuzzy Systems, vol. 16, no. 1, pp. 40-51.

[42] M. Jun and M.G. Safonov (1999), 'Automatic PID tuning: An application of unfalsified control,' in Proc. IEEE Int. Symp. CACSD, Hawaii, 328-333.

[43] M. Saeki (2003), 'Unfalsified control approach to parameter space design of PID controllers', in Proceedings 42nd IEEE Conf. Decision and Control, pp. 786-791.

[44] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin (1998), 'Iterative feedback tuning: Theory and applications,' in IEEE Contr. Syst. Mag., vol. 18, no. 4, pp. 26-41.

[45] O. Lequin, E. Bosmans, and T. Triest (2003), 'Iterative feedback tuning of PID parameters: Comparision with classical tuning rules', Contr. Eng. Pract., vol. 11, no. 9, 1023-1033.

[46] N. J. Killingsworth and M. Krstić (2006), 'PID tuning using extremum seeking: online, model-free performance optimization', in IEEE Control Systems Magazine, vol. 26, no. 1, pp. 70-79.

[47] S.Z. Khong, Y. Tan, C. Manzie and D. Nešić (2015), 'Extremum seeking of dynamical systems via gradient descent and stochastic approximation methods', in Automatica, vol. 56, pp. 44-52.

[48] S.Z. Khong, D. Nešić and M. Krstić (2016), 'Iterative learning control based on extremum seeking', in Automatica, vol. 66, pp. 238-245.

[49] M. Krstić and H.H. Wang (2000), 'Stability of extremum seeking feedback for general nonlinear dynamic systems', in Automatica, vol. 36, no.4, pp. 595-601.

[50] D. Nešić (2009), 'Extremum Seeking Control: Convergence Analysis', in European Journal of Control, vol. 15, no. 4, pp. 331-347.

[51] S.Z. Khong, D. Nešić, Y. Tan and C. Manzie (2013), 'Unified frameworks for sampled-data extremum seeking control: Global optimisation and multi-unit systems', in Automatica, pp. 2720-2733.

[52] D. Nešić, Y. Tan, W.H. Moase and C. Manzie (2010), 'A unifying approach to extremum seeking: adaptive schemes based on estimation of derivatives', in Proceedings of IEEE conference on decision and control, pp. 4625-4630.

[53] A.R. Teel and D. Popovic (2001), 'Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming,' in Proceedings of American Control Conference, vol. 3, pp. 2394-2399.

[54] Z. Cao, H. Durr, C. Ebenbauer, F. Allgower and F. Gao (2017), 'Iterative Learning and Extremum Seeking for Repetitive Time-Varying Mappings', in IEEE Transactions on Automatic Control, vol. 62, no. 7, pp. 3339-3353.

[55] P.Y. Papalambros and D.J. Wilde (2000), 'Principles of optimal design: Modeling and computation', Cambridge University Press New York, NY, USA.

[56] Thermo Fisher Scientific. `https://www.fei.com/products/`.

[57] E. Baeyens, A. Herreros, J.R. Perán (2016), 'A Direct Search Algorithm for Global Optimization', in Algorithms, vol. 9, no. 2, pp. 40-61.

APPENDIX A

Appendix

A.1 Gradient projection method

The gradient projection method is used to solve constrained optimization problem. If the solution is not in the feasible set, then gradient projection method "projects" it onto the feasible set. The gradient projection method is explained with an illustrative example. Consider the following inequality constrained optimization problem [55]:

$$\begin{aligned}&\min f(\mathbf{x})\\&\text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{Sx} < \mathbf{0}\end{aligned}, \tag{A.1}$$

where the constrained function $h(x)$ is defined as follows:

$$h(x) = \begin{bmatrix} -A-\delta \\ A-\delta \end{bmatrix}, \tag{A.2}$$

Since $S = \nabla h$, equation A.2 can be rewritten as follows:

$$h(x) = \underbrace{\begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} A \\ \delta \end{bmatrix}}_{x} \tag{A.3}$$

In Figure A.1 an illustrative example of the gradient projection method is depicted. Consider a constraint set $\mathcal{X}$, starting from a initial point $x^{(k)} \in \mathcal{X}$, the gradient projection method iterates the following equation until a stopping condition is met:

$$\begin{aligned}x^{(k+1)} &= P_{\mathcal{X}}\left(y^{(k+1)}\right) \text{ with}\\ y^{(k+1)} &= x^{(k)} - t^{(k)}\nabla f(x^{(k)})\end{aligned}, \tag{A.4}$$

with $t^{(k)}$ the step size and $P_{\mathcal{X}}$ the projection matrix [55]:

$$P \triangleq I - S^T\left(SS^T\right)^{-1}S, \tag{A.5}$$

where I is the identity matrix. The idea of gradient projection method is simple: if the point $y^{(k+1)}$ is located outside the constraint set $\mathcal{X}$ (due to the gradient update), then the gradient projection method projects $y^{(k+1)}$ back on the constrained set $\mathcal{X}$.

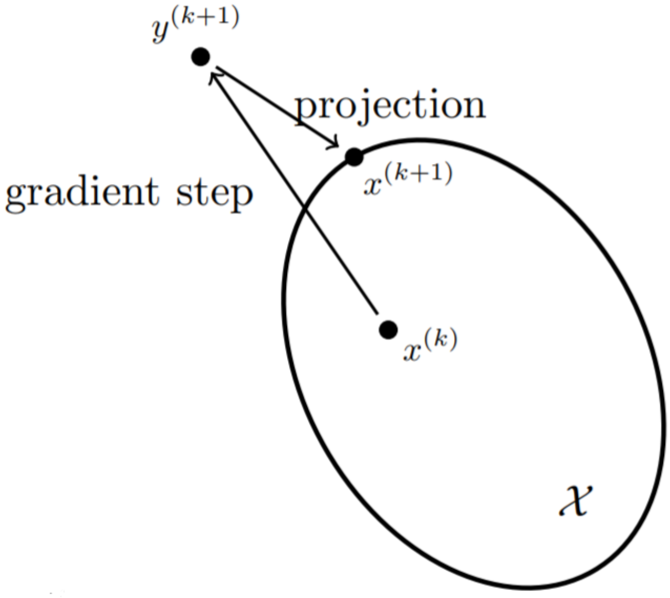

Figure A.1: The gradient projection method.

A.2 Input-output mapping

In Figures A.2 and A.3 the contour plots of the input-output mapping (3.1) for both friction characteristics are visualized where the time-varying integrator gain is parametrized by step-like basis functions. It is illustrated that there exists a set of optima (optimal area) and the classical gradient descent optimization algorithm (3.5) can be used to solve the optimization problem. In Figures A.2 and A.3, the evolution of the optimized parameters $u^{(2)}$ and $u^{(3)}$ by the learning controller, for different initial parameter vector u_0 are depicted (the magenta colored lines, labelled as $'$ESC$'$). It is presented that indeed the classical gradient descent optimization algorithm (3.5) is sufficient to solve the optimization problem at hand and that for several initial values of the parameter vector u, the minimum of $Q(u)$ is obtained in the set of optima (dark blue area in Figures A.2 and A.3).

The input-output mapping (3.1) for the Fourier basis function parametrization of $k_i(t)$ with respect to both friction characteristics are illustrated in Figures A.4 and A.5. It is illustrated that there exist regions of optima, which includes local and global optima (nonconvex mapping). The gradient descent algorithm (3.5) only finds local optima, so, for this case the gradient descent algorithm (3.5) is not suitable. However, it is presented Figures A.4 and A.5 that even for this case using the classical gradient descent method (3.5), a minimum of $Q(u)$ is obtained for several initial values of the parameter vector u, which still results in optimal desired performance. This is due to the fact that the iterative learning algorithm (ESC), independent of the initialization of the parameter vector u, still reaches an appropriate extremum (in the set of optima). The white area in Figures A.4 and A.5 depicts the constrained set. The constrained set is, that for $\delta/A > 1$ and $\delta/A <$ -1 there are no solutions of the Fourier basis function, and thus for the control force.

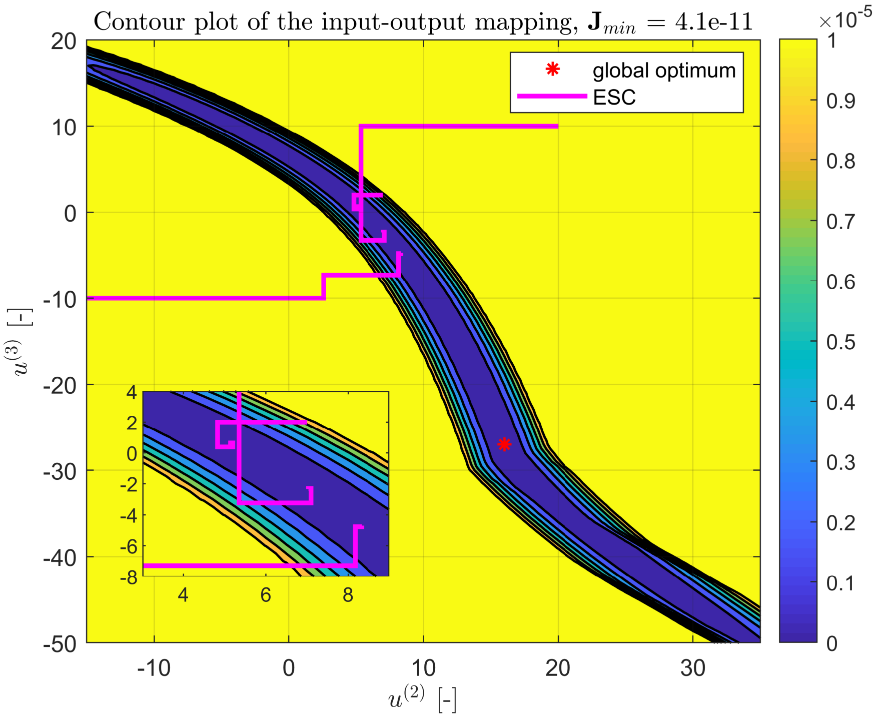

Figure A.2: Contour plot of the input-output mapping (3.1) for the step-like basis function parametrization of k_i(t) with respect to friction case a).

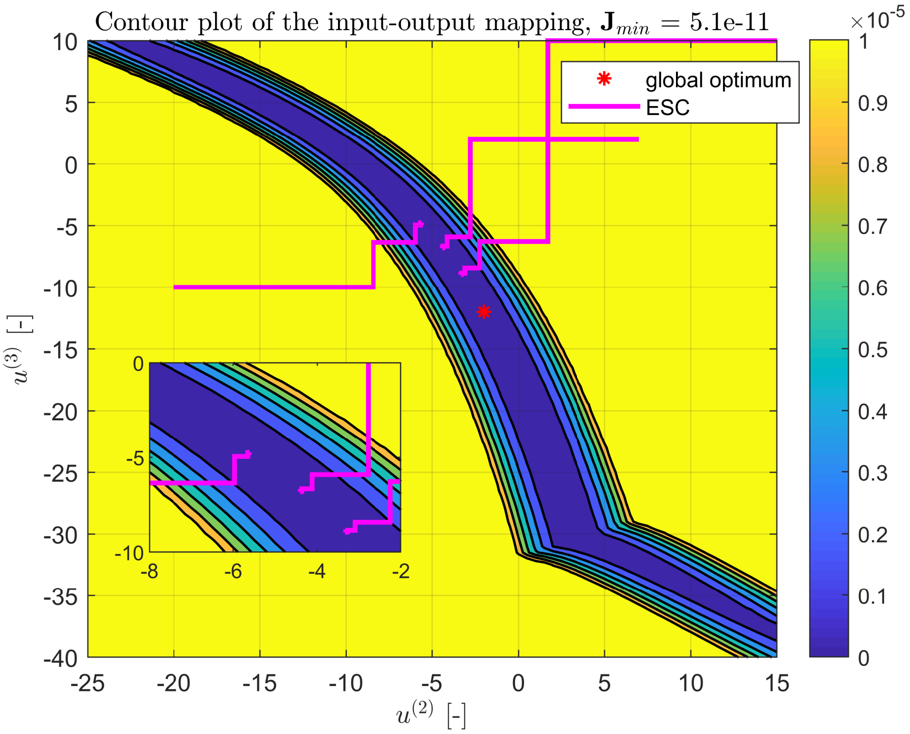

Figure A.3: Contour plot of the input-output mapping (3.1) for the step-like basis function parametrization of k_i(t) with respect to friction case b).

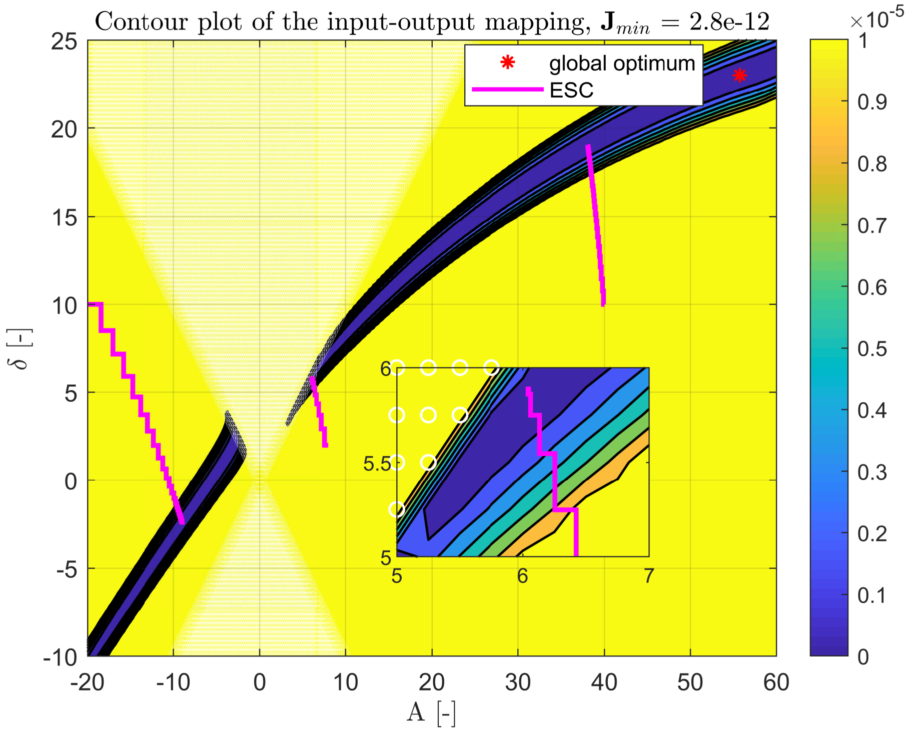

Figure A.4: Contour plot of the input-output mapping (3.1) for the Fourier basis function parametrization of k_i(t) with respect to friction case a).

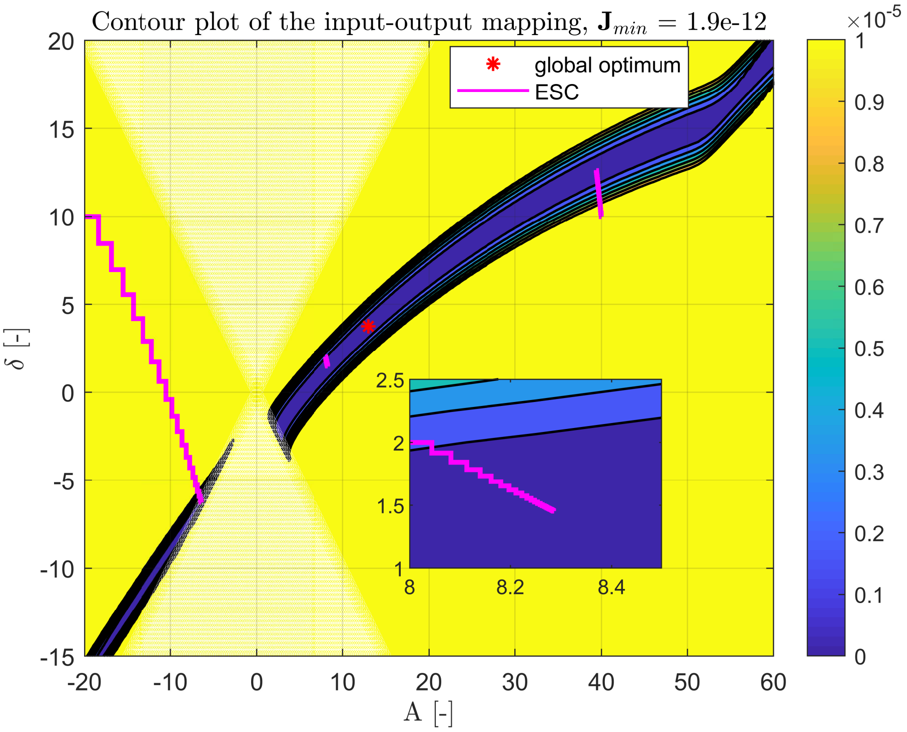

Figure A.5: Contour plot of the input-output mapping (3.1) for the Fourier basis function parametrization of k_i(t) with respect to friction case b).

A.3 Simulation results: Application of the time-varying learning integrator gain

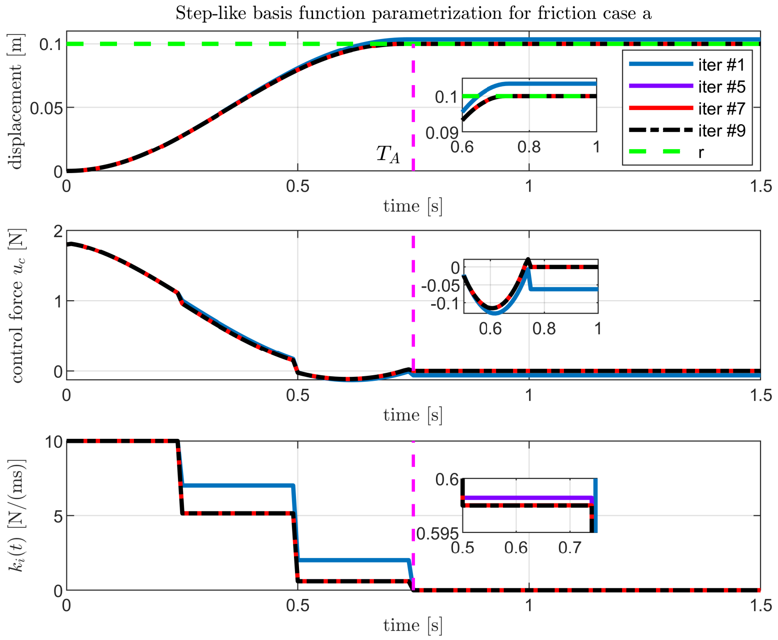

Figure A.6: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the step-like basis function parametrization of $k_i(t)$ with respect to friction case a).

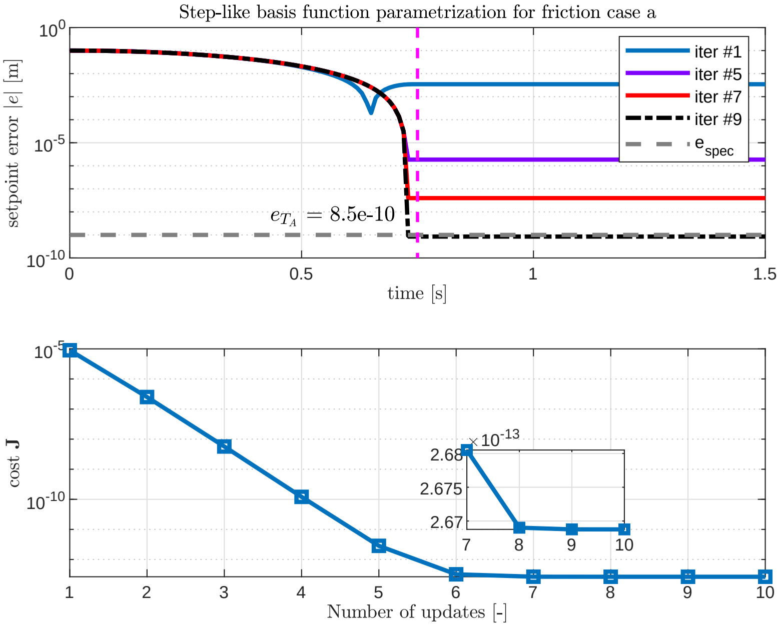

Figure A.7: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the step-like basis function parametrization of $k_i(t)$ with respect to friction case a).

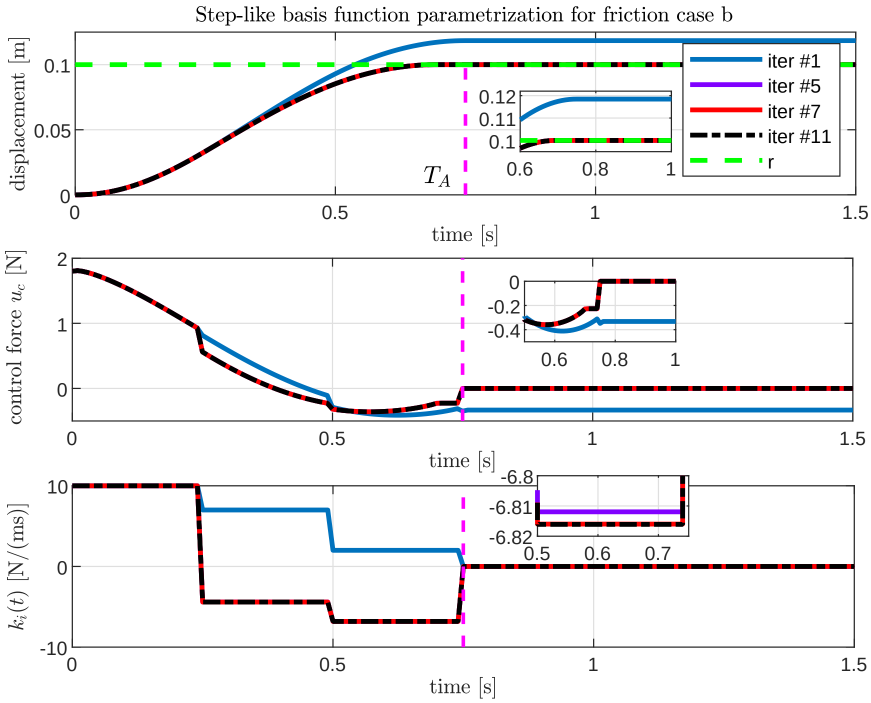

Figure A.8: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the step-like basis function parametrization of $k_i(t)$ with respect to friction case b).

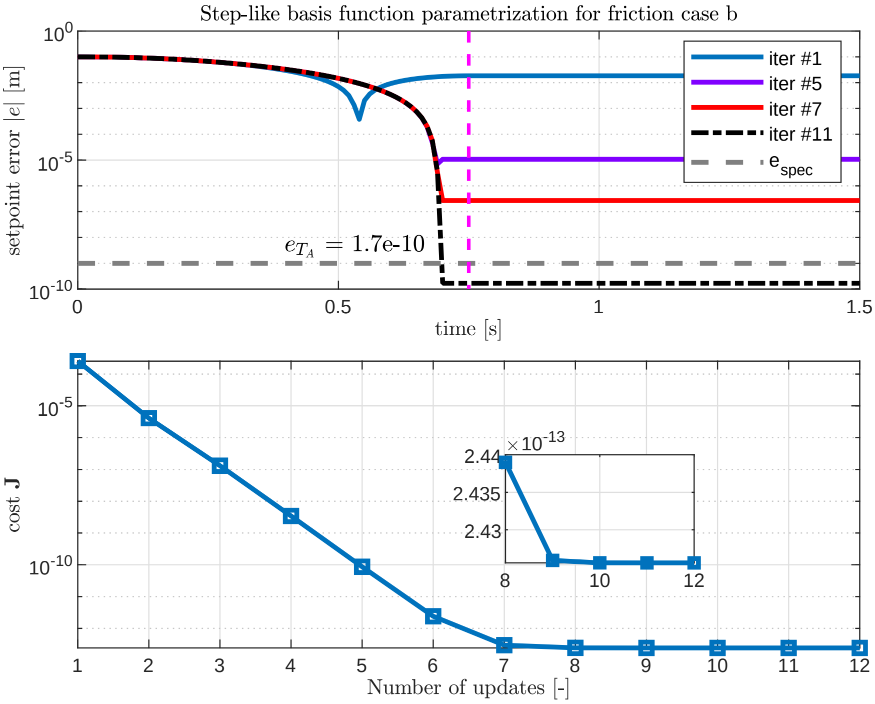

Figure A.9: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the step-like basis function parametrization of $k_i(t)$ with respect to friction case b).

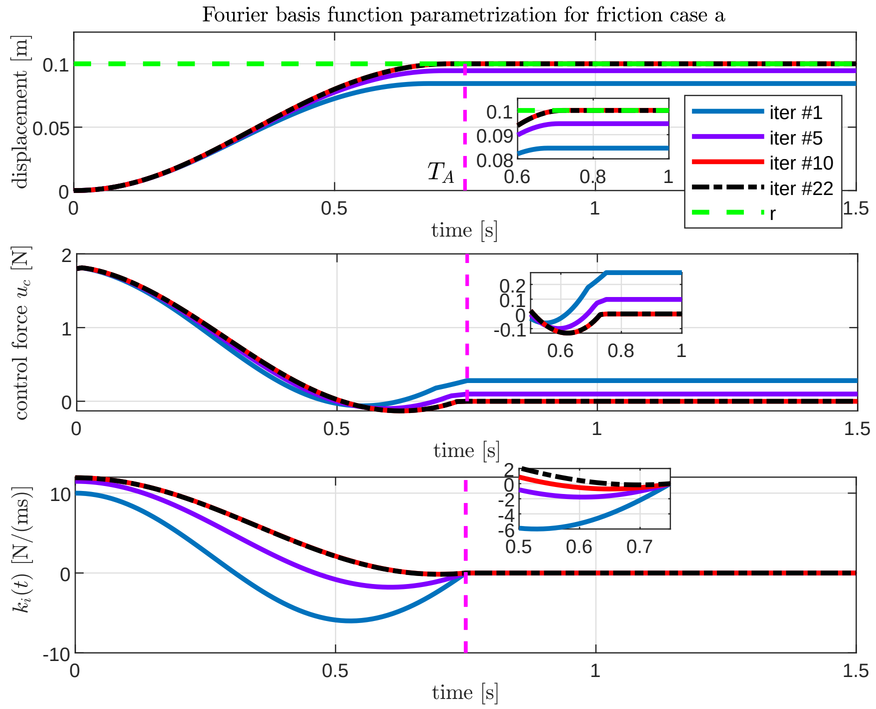

Figure A.10: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the Fourier basis function parametrization of $k_i(t)$ with respect to friction case a).

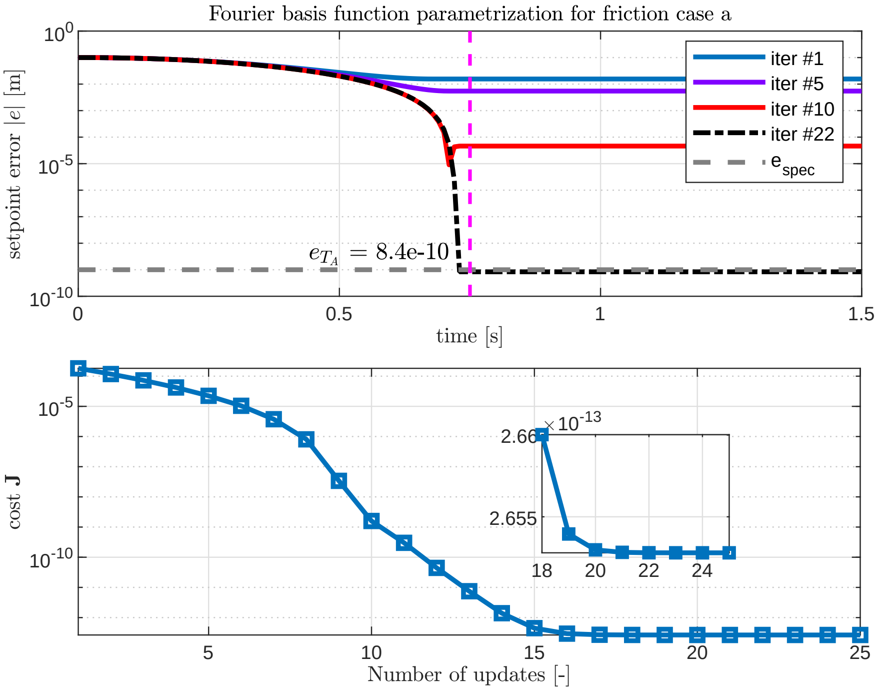

Figure A.11: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the Fourier basis function parametrization of $k_i(t)$ with respect to friction case a).

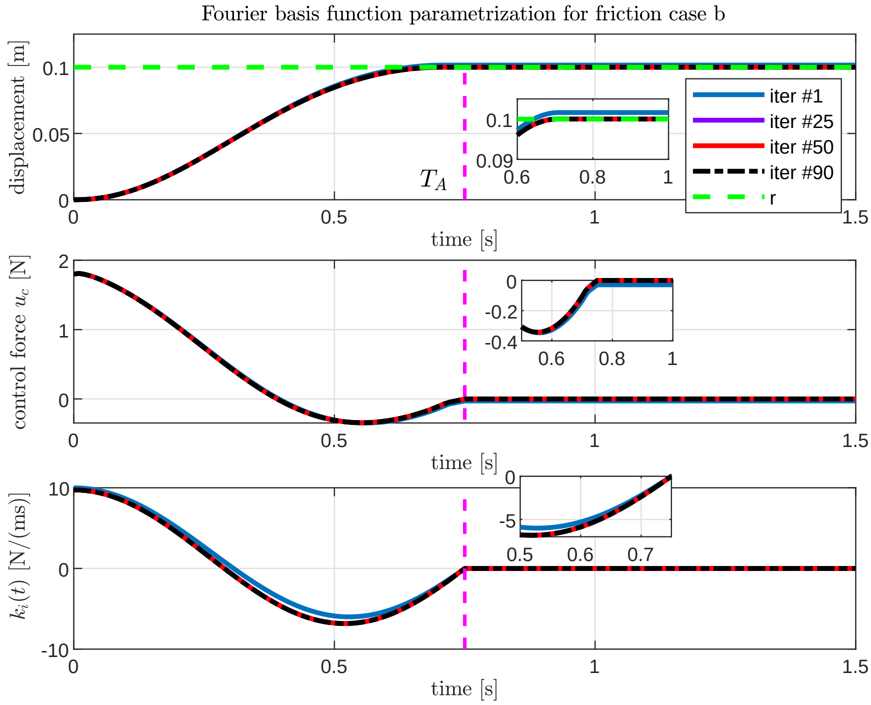

Figure A.12: The displacement x_1, control force u_c and time-varying integrator gain $k_i(t)$, for the Fourier basis function parametrization of $k_i(t)$ with respect to friction case b).

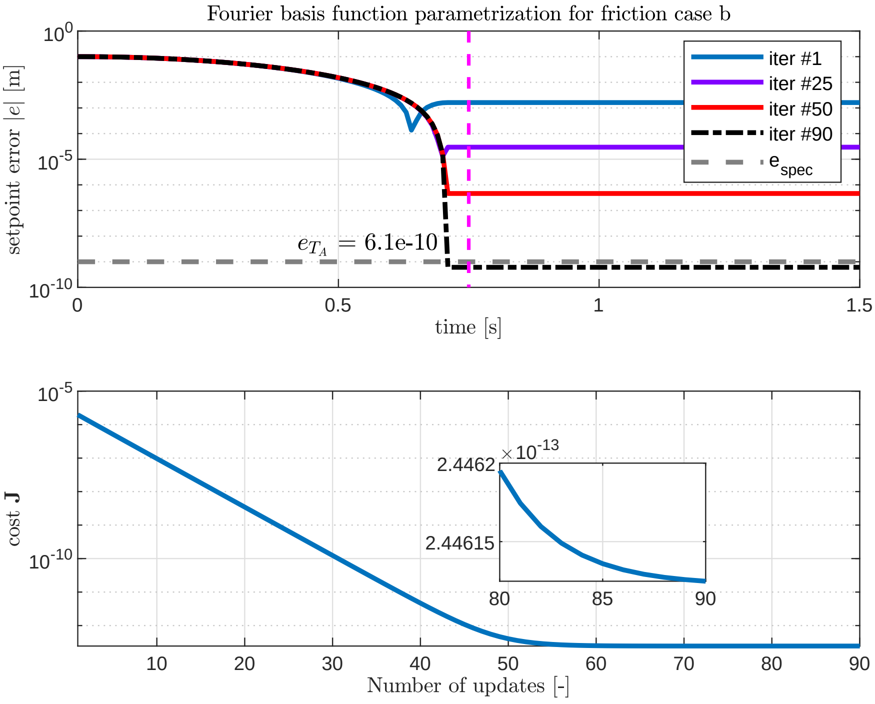

Figure A.13: The absolute error $|e|$ and the objective function $\mathbf{J}$, for the Fourier basis function parametrization of $k_i(t)$ with respect to friction case b).

A.4 Experimental case study results

In Figures A.14 and A.15, the time responses of an extremum seeking learning controller experiment for an initial parameter vector $u_0^T = [u^{(2)} \;\; u^{(3)}] = [12 \cdot 10^7 \;\; 10 \cdot 10^7]$ are depicted. For this experiment, the step size of the gradient estimator τ and optimizer gain are set to $5 \cdot 10^6$ and $\gamma = 6{\cdot}10^{23}$, respectively. The found optimal settings are $u_{opt}^T = [u^{(2)} \;\; u^{(3)}] = [8.86 \cdot 10^7 \;\; 7.59 \cdot 10^7]$.

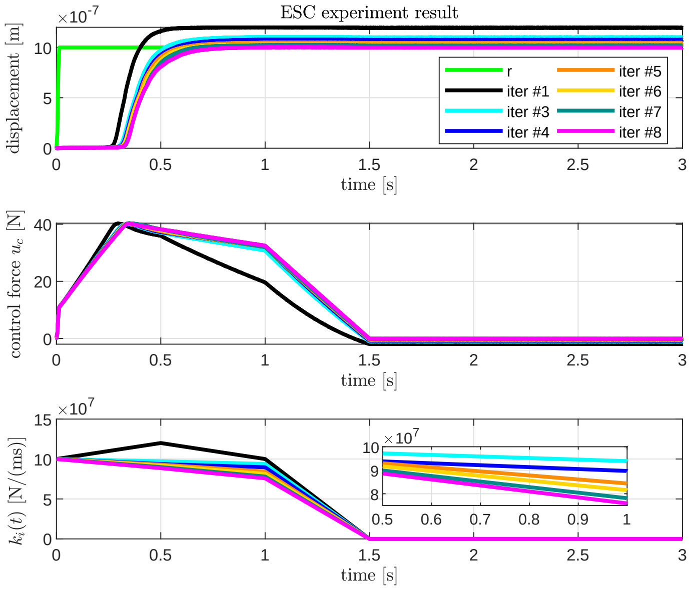

Figure A.14: Time responses of the displacement, control force u_c and $k_i(t)$ using ESC.

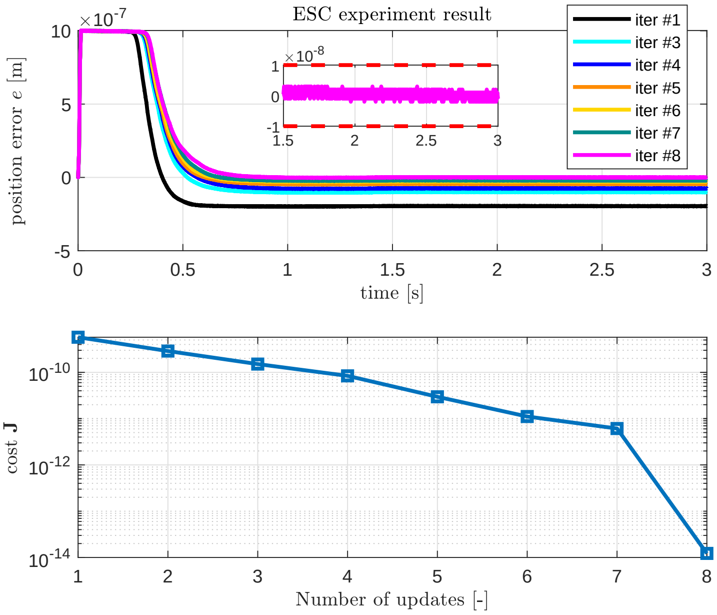

Figure A.15: Time responses of the error e and objective function **J** using ESC.

In Figures A.16 and A.17, the time responses of an extremum seeking learning controller experiment for an initial parameter vector $u_0^T = [u^{(2)} \quad u^{(3)}] = [8 \cdot 10^7 \quad 6 \cdot 10^7]$ are depicted. For this experiment, the step size of the gradient estimator τ and optimizer gain are set to $5 \cdot 10^6$ and $\gamma = 2{\cdot}10^{23}$, respectively. The found optimal settings are $u_{opt}^T = [u^{(2)} \quad u^{(3)}] = [6.65 \cdot 10^7 \quad 6.67 \cdot 10^7]$.

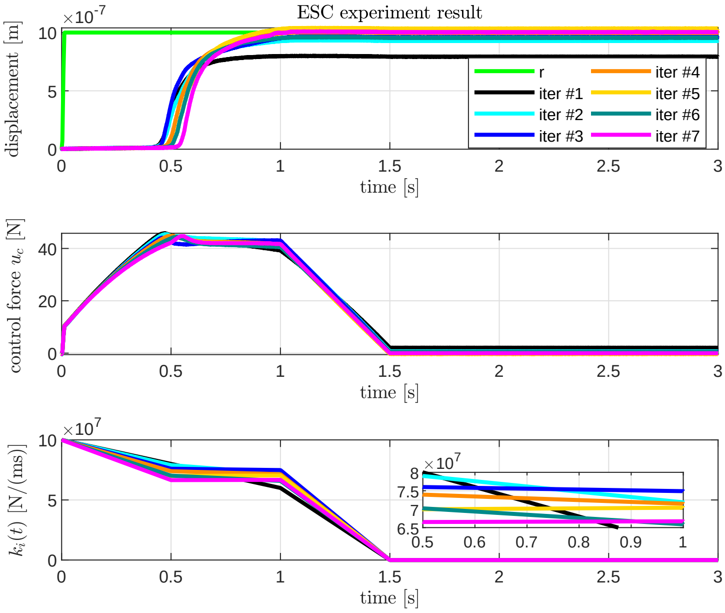

Figure A.16: Time responses of the displacement, control force u_c and $k_i(t)$ using ESC.

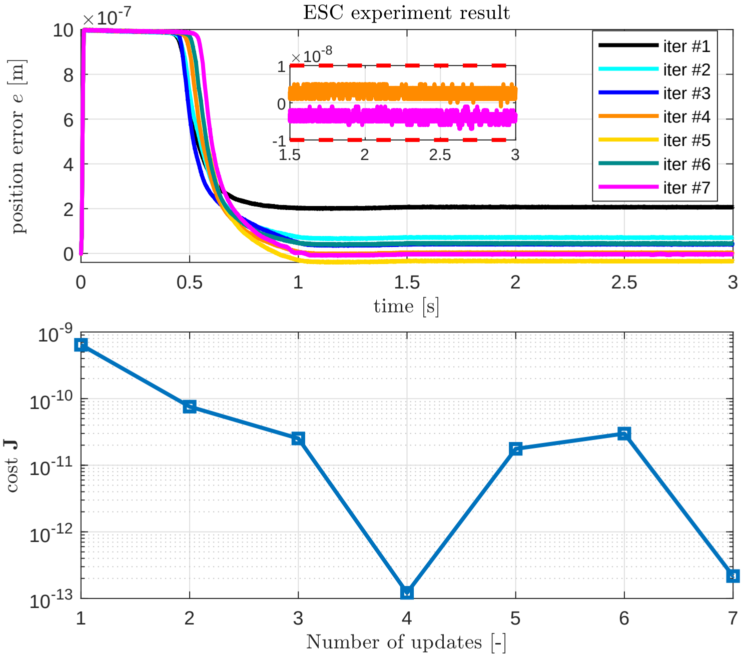

Figure A.17: Time responses of the error e and objective function **J** using ESC.

TU/e Technische Universiteit
Eindhoven
University of Technology

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

06-11-2019

Name

Hekmatullah Sekandari

ID-number

0821565

Signature

Submit the signed declaration to the student administration of your department.

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU