

A sampled-data extremum-seeking approach for accurate setpoint control of motion systems with friction^{*}

Leroy Hazeleger^{*} Ruud Beerens^{*} Nathan van de Wouw^{*,**}

^{*} Eindhoven University of Technology, Department of Mechanical Engineering, Eindhoven 5600 MB, The Netherlands

^{**} University of Minnesota, Department of Civil, Environmental and Geo-Engineering, Minneapolis MN 55455, USA
(e-mail: {l.hazeleger; r.beerens; n.v.d.wouw}@tue.nl)

Abstract: This paper presents an extremum-seeking approach for accurate setpoint control of motion systems with friction, performing a repetitive motion. The classical PID controller, often used in industry for frictional motion systems, suffers from severe performance limitations. In particular, friction-induced limit cycling (hunting) is observed when integral control is employed on systems with unknown Stribeck friction, thereby compromising stability. Moreover, even if stability is warranted, transient performance highly depends on the particular frictional characteristic, which is typically uncertain. To deal with such uncertainty and to warrant optimal setpoint performance for the actual frictional properties, we propose a PID-based learning controller that achieves improved transient performance. Hereto, we consider a PID-type controller with a time-varying integral controller gain, which is adaptively obtained by employing a sampled-data extremum-seeking approach, resembling iterative learning control. The proposed approach does not require any knowledge on the friction characteristic. The working principle is illustrated by means of a representative simulation example.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Setpoint stability, friction, sampled-data control, extremum-seeking

1. INTRODUCTION

Many industrial motion systems perform repetitive tasks, such as, e.g., repetitive motion profiles in pick-and-place machines, large-scale transferring of circuit topology to silicon wafers in lithography systems, and automated scanning procedures in electron microscopes. Due to hardware cost reduction in the design phase or wear in the operational phase, friction is commonly present in such high-precision positioning systems, thereby limiting the achievable positioning accuracy.

Various control solutions have been presented throughout the literature to cope with frictional effects. Model-based methods, such as, e.g., model-based friction compensation (see, e.g., Makkar (2007), Freidovich (2010)), exploit parametric models in the control loop to compensate for friction. However, as friction characteristics are commonly unknown, uncertain, and (slowly) time-varying, model-based methods are prone to modeling errors, ultimately compromising positioning performance. Non-model-based methods, e.g., impulsive control (see, e.g., van de Wouw (2012)) and sliding-mode control (see, e.g., Bartolini (2003)), can result in stabilization of the setpoint, but may be challenging to implement and to tune due to the lack of intuitive tooling for control practitioners.

Despite several applications of these control techniques in literature, the vast majority of the high-precision industry employs classical PID control due to its intuitive and easy-to-use design and tuning tools, and knowledge and experience of control practitioners. Moreover, it is well-known that integrator action in PID control is capable of com-

pensating for *unknown* static friction in motion systems. However, PID control is prone to performance limitations as well. For example, friction-induced limit cycling (i.e., *hunting*, see Hensen (2003)), is observed when integral control is employed on systems where the friction characteristic includes the velocity-weakening (Stribeck) effect, destabilizing the setpoint. Even if stability is warranted, transient performance depends on the particular friction characteristic, which is highly uncertain in practice.

In this work, we propose a PID-based learning controller to guarantee a high positioning accuracy. The controller effectively deals with unknown and uncertain frictional effects in motion systems that perform a repetitive motion. The learning controller consists of a PID term, and a learning mechanism that iteratively improves transient performance by adaptive tuning of a *time-varying integrator gain*. The learning mechanism resembles iterative learning control (ILC, see, e.g., Bristow (2006) and Wang (2009)). However, the optimization problem is formulated in terms of a model-free sampled-data extremum-seeking control problem (ESC, see, e.g., Teel (2001), Kvaternik (2011), Khong (2013), Khong (2016)) by using an appropriate set of basis functions to parameterize the time-varying integrator gain. Parametrization of the to-be-designed input signal is similar to the one in Khong (2016), where the input signal is parametrized by using a so-called de-multiplexer. However, in this work, the parametrization of inputs is not necessarily limited to step-like basis functions. Moreover, opposed to classical iterative learning control approaches, employing a sampled-data extremum-seeking approach is beneficial, as it is able to deal with unknown, uncertain, and time-varying dynamical systems, while optimizing system performance (see, e.g., Krstić (2000), Haring (2013), Cao (2017)).

^{*} This research is part of the research programme High Tech Systems and Material (HTSM), which is supported by NWO domain Applied and Engineering sciences and partly funded by the Ministry of Economic Affairs.

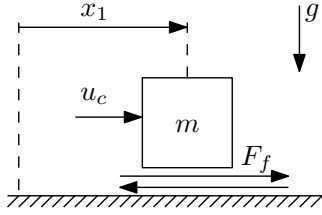


Fig. 1. Schematic representation of the motion system subject to friction.

The main contributions of this paper are as follows. First, we design a PID-based controller with a *time-varying* integrator gain, which facilitates a high setpoint accuracy and improved transient response, compared to classical P(I)D control, despite the presence of unknown friction. Second, we present a basis function parametrization of the time-varying integrator gain, which allows the setpoint control problem to be formulated as an extremum-seeking control problem. Third, we employ a sampled-data extremum seeking controller design, to iteratively find the values for the parameterized time-varying integrator gain that result in optimal setpoint accuracy and transient behavior.

The remainder of this paper is organized as follows. We formalize the control problem in Section 2, and we present the PID-based learning controller in Section 3. The working principle of the proposed controller is illustrated by means of a numerical example in Section 4, and conclusions are presented in Section 5.

Notation: $\text{Sign}(\cdot)$ (with an upper-case S) denotes the *set-valued* sign function, i.e., $\text{Sign}(y) := 1$ for $y > 0$, $\text{Sign}(y) := -1$ for $y < 0$, and $\text{Sign}(y) := [-1, 1]$ for $y = 0$.

2. SETPOINT CONTROL PROBLEM FORMULATION

Consider a single-degree-of-freedom motion system, consisting of a mass m sliding on a horizontal plane with position x_1 and control input u_c , as schematically depicted in Fig. 1. The goal is to control the system to the reference r repetitively, with the same initial conditions in each setpoint operation. We denote the measurable position as x_1 , and the velocity of the mass is denoted by x_2 . The mass is subject to a friction force, taking values according to the set-valued mapping of the velocity $x_2 \mapsto \Phi(x_2)$, resulting in dynamics governed by the following differential inclusion:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &\in \frac{1}{m} (\Phi(x_2) + u_c). \end{aligned} \quad (1)$$

The set-valued friction characteristic Φ consists of a Coulomb friction component with (unknown) static friction F_s , and a velocity-dependent friction component f , encompassing the Stribeck effect, and a viscous friction contribution, i.e.,

$$\Phi(x_2) := -F_s \text{Sign}(x_2) + f(x_2), \quad (2)$$

which we assume to satisfy $x_2 \Phi(x_2) \leq 0$, for all x_2 . Before presenting the proposed time-varying controller in Section 3, we first consider a classical PID controller for input u_c in (1), i.e.,

$$\begin{aligned} u_c &= k_p e + k_d \dot{e} + k_i x_3, \\ \dot{x}_3 &= e, \end{aligned} \quad (3)$$

where $e := r - x_1$ denotes the setpoint error, x_3 denotes the integrator state, and k_p , k_d , and k_i the proportional, derivative, and integral controller gains, respectively.

Integrator action allows the system to escape a stick phase, induced by the set-valued Coulomb friction effect, by inte-

grating the position error. The control force that is built up in this way eventually compensates for the *unknown* static friction. In the slip phase that follows, however, friction is overcompensated due to the velocity-weakening effect typically present in f . This process repeats and results in stick-slip limit cycling, compromising setpoint stability, see Fig. 2. Omission of the integrator action (i.e., only employing PD control) results in a non-zero steady-state position error, whose magnitude depends inversely on the proportional gain k_p (Putra (2007)), see Fig. 2. Respecting the popularity of PID control in industry, and taking into account the advantages of integrator action, in this paper we address the following setpoint control problem.

Problem 1. Design a PID-based controller for input u_c in (1), that achieves 1) optimal setpoint accuracy, and 2) optimal transient behavior, for a repetitive motion profile characterized by r , with respect to any *unknown* friction characteristic Φ in (2).

3. PID-BASED LEARNING CONTROLLER

In this section, we first present a PID-type controller with time-varying integrator gain $k_i(t)$ to solve Problem 1. We then formulate the design of $k_i(t)$ as a model-free extremum-seeking control problem. To this end, we formalize the objective to-be-optimized, and we propose a parametrization of $k_i(t)$ by means of step-like basis functions. Finally, we present a sampled-data extremum seeking approach that solves Problem 1 by adaptive tuning of the parameterized time-varying integrator gain $k_i(t)$.

3.1 PID controller with time-varying integrator gain

The limit-cycling present in the case of PID control, with *constant* integrator gain, is caused by the build-up of integrator action (during transients and the stick phase) in interplay with the friction characteristic. This observation motivates the design of a *time-varying* integrator gain $k_i(t)$, such that 1) the presence of integrator action still allows the system to escape a stick phase, and 2) overcompensation of friction is avoided, by altering k_i during the slip phase. Intuitively speaking, a relatively high integrator gain k_i is desired to quickly escape the stick phase, whereas a reduced (or even negative) k_i may be desired in the slip phase to avoid overcompensation of friction. Finally, zero integral action is desired at the setpoint to achieve

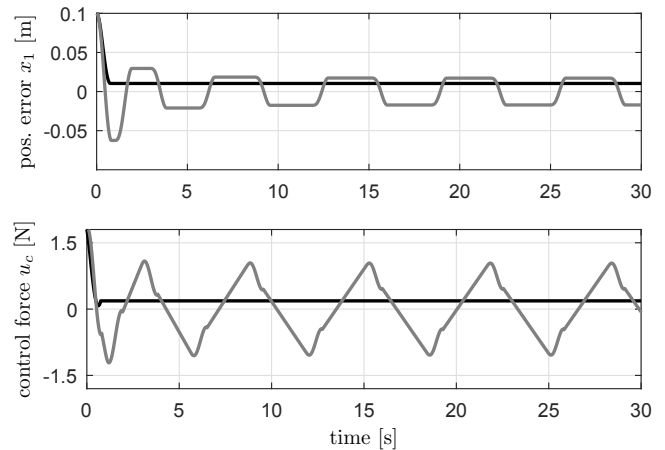


Fig. 2. Example of a position error response (top) and corresponding control force (bottom) for a PD controller (—), and a PID controller with fixed integrator gain (---). $k_p = 18$, $k_d = 2.5$, $k_i = 35$. Friction parameters in f and F_s are given in Section 4.

a standstill of the system. The resulting controller is then given by

$$\begin{aligned} u_c &= k_p e + k_d \dot{e} + k_i(t) x_3, \\ \dot{x}_3 &= e. \end{aligned} \quad (4)$$

Since the friction characteristic Φ in (1) is generally unknown or uncertain, the optimal model-based design of $k_i(t)$ is challenging, or even impossible. It is therefore determined by a learning algorithm to adaptively tune $k_i(t)$, as presented in Section 3.2 below. The proposed learning controller embeds the above engineering intuition on $k_i(t)$, and is formulated as a model-free, sampled-data extremum-seeking control problem to avoid the need of a plant model (which is generally unavailable due to the unknown friction characteristic).

Remark 1. We choose here to employ a time-varying integrator gain (and learn its characteristic), instead of an appropriate feedforward control signal in combination with a constant integrator gain (required to compensate for unknown static friction). With the former approach, we are able to create robustness close to the setpoint, by realizing $k_i = 0$ during the desired standstill of the system. Indeed, when $k_i = 0$, the system remains in standstill since build up of control force is prevented, so that the system cannot escape the stick phase any more. Moreover, robustness to other force disturbances is then obtained as the proportional action is low, compared to the static friction (due to the small position error close to the setpoint).

3.2 Model-free adaptive tuning mechanism

Setpoint performance metric: We consider a desired repetitive motion profile defined on the time interval $t \in [0, T]$, where the system starts and ends at rest. The time window $[0, T]$ can be separated into two particular parts, specified by the so-called *standstill time instant* T_B (i.e., the time instant where the mass is required to arrive at the setpoint) as follows:

- i) $t \in [0, T_B]$; the so-called *transient* during which the system moves from 0 to the constant setpoint r ;
- ii) $t \in [T_B, T]$, the window during which standstill is required with $e = 0$. The time interval $[T_B, T]$ is typically used by the machine, of which the motion system is part, to perform a certain operation, for which accurate positioning is necessary.

The desired performance, i.e., an optimal transient on $[0, T_B]$, and an optimal setpoint accuracy on $[T_B, T]$, can be captured by the following objective function L_p :

$$L_p(e) := w_1 \int_0^{T_B} |e(t)|^2 dt + w_2 \int_{T_B}^T |e(t)|^2 dt, \quad (5)$$

with $e := r - x_1$, and w_1 and w_2 suitable weighting factors, trading off the emphasis on transient performance versus setpoint accuracy. Other (transient) performance relevant variables such as, e.g., the control effort u_c generated by (4), or the velocity x_2 of the mass if accurate velocity measurements are available, can be taken into account in (5) as well. We will show in Section 4 below that the objective function in (5) indeed captures the desired performance.

Parametrization of time-varying integrator gain $k_i(t)$: We propose a design for $k_i(t)$, parameterized by a finite set of basis functions Ψ , and a parameter vector $u \in \mathbb{R}^p$ to be designed, as follows:

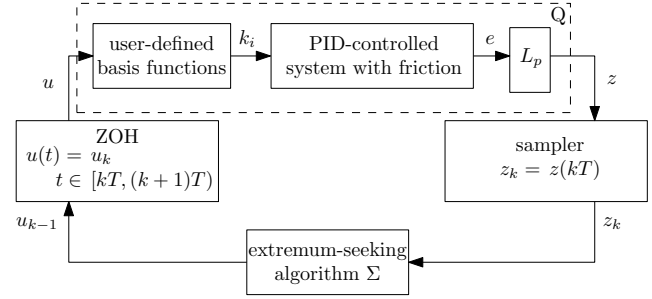


Fig. 3. The sampled-data extremum-seeking framework based on sampled-data control law with periodic sampling time T , and sampler and zero-order hold elements.

$$k_i(t) := \sum_{j=1}^p u^{(j)} \Psi^{(j)}(t), \quad (6)$$

where $u^{(j)}$ denotes the j th element of the vector u , and p are the number of elements in u . One possible choice for the basis functions are step basis functions, i.e.,

$$\Psi^{(j)}(t) := \begin{cases} 1, & t \in [(j-1)t_s, jt_s) \\ 0, & t \notin [(j-1)t_s, jt_s) \end{cases} \quad \text{for } j = 1, 2, \dots, p, \quad (7)$$

where t_s satisfies $T = pt_s$. Without loss of generality, this choice is adopted in the paper.

Remark 2. Other types of basis function designs can be adopted from the iterative learning control literature. For example, polynomial bases (see, e.g., van de Wijdeven (2010), van der Meulen (2008)) and rational bases (see, e.g., Bolder (2015)) can similarly be exploited, which can potentially reduce the number of parameters to be optimized.

The number of elements p is to be chosen by the user. Taking $p = 1$ results in a constant integrator gain k_i , see (6). Taking $p > 1$ allows for more freedom in the design of $k_i(t)$, facilitating its time-varying design. The parametrization in (6), (7), for example, allows to generate a relatively high integrator gain k_i in stick (to escape such a phase), and a reduced integrator gain k_i in the slip phase (to avoid overcompensation of friction). Increasing p , however, results in a larger vector u in (7) and increases the complexity of the controller, in the sense that an increased number of parameters in u need to be found. The precise realization of the vector u that minimizes (5) is typically unknown, especially due to the uncertainty in the friction characteristic. To deal with these complications, we present an adaptive mechanism to design the vector u .

Sampled-data extremum-seeking framework: We propose an iterative learning algorithm to adaptively tune u . Specifically, given the objective function in (5) and parametrization of the integrator gain in (6), we formulate the setpoint control problem as a model-free sampled-data extremum seeking problem (see, e.g., Khong (2013) and Khong (2016)). In particular, the cascade connection of the PID-controlled motion system given by (1)-(3), with k_i parameterized by (6), and L_p in (5), yields the following *unknown* input-output map $Q: \mathbb{R}^p \rightarrow \mathbb{R}$:

$$Q(u) := L_p(e) = w_1 \int_0^{T_B} |e(t)|^2 dt + w_2 \int_{T_B}^T |e(t)|^2 dt. \quad (8)$$

Based solely on output measurements, extremum-seeking

control is able to adaptively find a vector u that minimizes (8). Fig. 3 shows the sampled-data extremum-seeking framework, i.e., the interconnection of the dynamical system (1)-(3), (6), (7) and objective function L_p in (5) implemented as follows:

$$z(t) := L_p(e(t)) = w_1 \int_{t-T}^{t-t^*} |e(s)|^2 ds + w_2 \int_{t-t^*}^t |e(s)|^2 ds, \quad (9)$$

where $t^* = T - T_B$ and $e(s) = 0$ for $s \in [-T, 0)$, with a T -periodic sampler, a discrete-time extremum-seeking algorithm Σ , and a zero-order hold (ZOH) element. Let $\{u_k\}_{k=0}^\infty$ be a sequence of vectors generated by Σ based on collected measurements, and define the ZOH operation as follows:

$$u(t) := u_k \quad \forall t \in [kT, (k+1)T), \quad (10)$$

with $k = 0, 1, 2, \dots$, and sampling period $T > 0$ with $u_k \in \mathbb{R}^p$, for all k . Let us define the ideal periodic sampling operation:

$$z_k := z(kT), \quad k = 1, 2, \dots, \quad (11)$$

where $z_k = Q(u_{k-1})$ are the collected measurements as used by the extremum-seeking algorithm Σ , see Fig. 3. We care to stress that T is the sampling period of the extremum-seeking controller, which conforms to the period time of the motion profile, and T is *not* the sampling period of the underlying motion system. Moreover, it must be noted that periodic re-initialization of the states to fixed values is needed, i.e., $x(kT) = x_0$ for all $k = 1, 2, \dots$, for Q in (8) to be uniquely defined (see also Remark 4 below).

Remark 3. In most (sampled-data) extremum-seeking literature, Q reflects the *steady-state* behavior of the dynamical system. In those cases, the sampling period T , or so-called waiting time T , see, e.g., Teel (2001), Khong (2013), Kvaternik (2011), needs to be chosen sufficiently large by the user such that the closed-loop extremum-seeking scheme is robust against inexact measurements of the cost Q due to the *transient* behavior of the system. Here, Q in (8) actually incorporates the transient behavior of the system, which ultimately determines positioning accuracy. As such, the role of the waiting time T is different here, and is conveniently chosen equal to the period time T of the repetitive motion profile.

Remark 4. A common requirement in the extremum-seeking literature is that the input-output mapping Q is independent of initial conditions. Here, the transient behavior is partly determined by the initial conditions, and re-initialization after each setpoint operation is theoretically required for an input-output mapping Q as in (8) to be uniquely defined. Re-initialization for transient performance optimization is also a well-known requirement in the iterative learning control literature, see, e.g., Nörrolof (2002, Sec. 4) and Bristow (2006).

Let Σ be any algorithm that solves the optimization problem of finding the minimum of $Q(u)$:

$$z^* := \min_u Q(u). \quad (12)$$

Within this periodic sampled-data framework, and depending on the nature of the input-output mapping Q , many algorithms from the optimization literature can be employed to solve (12). For example, in Khong (2013) the so-called DIRECT and Shubert algorithms (see Jones (1993) and Shubert (1972), respectively) are employed to find the global extremum of Q . If finding a local minimum suffices or if Q possesses only a single (global) extremum, the classical gradient descent or Newton method can be used (see, e.g., Boyd (2004)).

Extremum-seeking controller design Without loss of generality, we employ here the following gradient descent algorithm to optimize the vector u in (6):

$$u_k = u_{k-1} - \lambda \nabla Q(u_{k-1}), \quad (13)$$

with λ the optimizer gain. Since Q is unknown, its gradient is unknown. As such, the gradient of Q will be estimated based on finite differences as follows:

$$\nabla Q(u) \approx \frac{1}{\tau} \begin{bmatrix} Q(u + \tau d_1) - Q(u) \\ \vdots \\ Q(u + \tau d_p) - Q(u) \end{bmatrix}, \quad (14)$$

where τ is the step size of the gradient estimator, and d_j with $j = 1, \dots, p$ are dither signals, i.e., vectors where the j th element is equal to one, and all other elements are zero. Dithering needs to be done in a sequential manner to acquire the elements in (14). In particular, the gradient descent algorithm in (13) and the sequence of dithers to obtain the approximate gradient in (14) is implemented through the following extremum-seeking algorithm:

$$u_k = \begin{cases} u_{k-n} + \tau d_n & \text{if } n \neq 0 \\ u_{k-(p+1)} - \lambda \nabla Q(u_{k-(p+1)}) & \text{if } n = 0 \end{cases}, \quad (15)$$

for all $k = 1, 2, \dots$, with $n = \text{mod}(k, p+1) \in \mathbb{N}$, initial input u_0 , and

$$\nabla Q(u_{k-(p+1)}) = \frac{1}{\tau} \begin{bmatrix} Q(u_{k-p}) - Q(u_{k-(p+1)}) \\ \vdots \\ Q(u_{k-1}) - Q(u_{k-(p+1)}) \end{bmatrix}. \quad (16)$$

Note that the case $n = 0$ in (15) implements an update of the control signal u .

4. SIMULATION EXAMPLE

In this section, the working principle and effectiveness of the proposed PID-based learning controller in Section 3 are illustrated by means of a numerical example.

In order to correctly deal with the set-valued dynamics, a numerical time-stepping scheme is used Acary (2008, Chap. 10). Consider a PID-controlled motion system given by (1)-(3), and illustrated by Fig. 1. The integrator gain k_i is either constant, or time-varying in accordance with the proposed learning controller. We adopt the following numerical values: $m = 1$ kg, $g = 9.81$ m/s², $k_p = 18$, and $k_d = 2.5$. The friction characteristic (2) satisfies $F_s = 0.981$ N, and the velocity-dependent part of the friction f contains the velocity-weakening (Stribeck) effect, and is given by

$$f(x_2) = ((F_s - F_c)\eta x_2)(1 + \eta|x_2|)^{-1} - \alpha x_2, \quad (17)$$

with F_c the Coulomb friction force, α the viscous friction coefficient, and η the Stribeck shape parameter. We illustrate the effectiveness of the controller by considering two cases in the simulation study, in the form of different friction characteristics (see also Fig. 4):

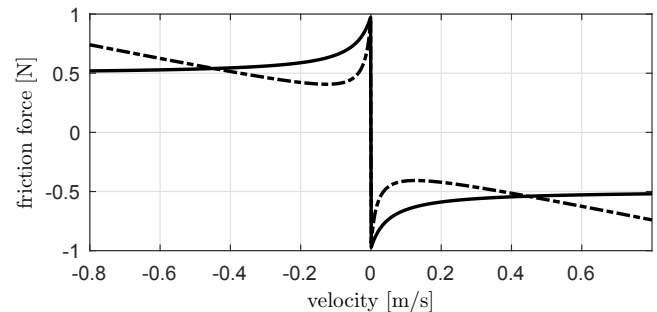


Fig. 4. Friction characteristics a) (—), and b) (---).

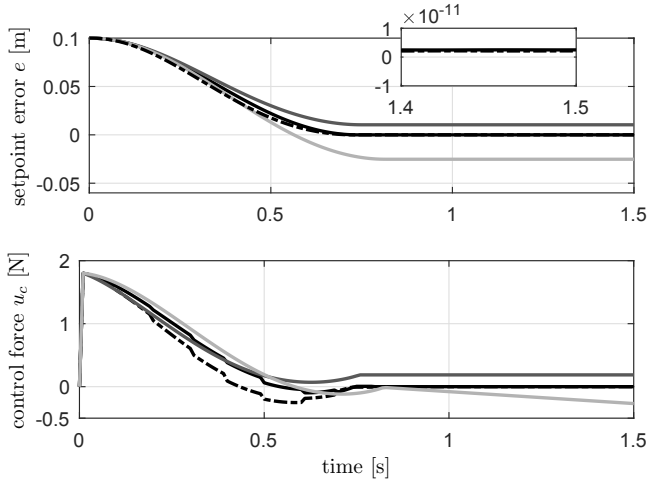


Fig. 5. Setpoint error e and control force u_c with a fixed $k_i = 10$ (—), and no integral action $k_i = 0$ (---) for case a), and for a time-varying $k_i(t)$ for case a) (—) and case b) (---).

case a) $F_c = F_s/2$, $\alpha = 0$ and $\eta = 20$;
case b) $F_c = F_s/3$, $\alpha = 0.5$, and $\eta = 60$.

We consider a desired repetitive motion profile on the interval $[0, T]$, with $T = 1.5$, and a constant setpoint reference $r = 0.1$ m. We require the system to arrive at the setpoint at $t = 0.75$ s, i.e., the standstill time instance $T_B = 0.75$. The integrator gain k_i is parameterized as in (6), and we employ step basis functions (cf. (7)) with $p = 15$. The optimizer gain is chosen as $\lambda = 8000$, and the step size of the gradient estimator in (14) yields $\tau = 10^{-8}$. Finally, the weighting factors for the objective function in (5) are $w_1 = 10^{-10}$ and $w_2 = 1$. As the results below illustrate, in this example a weight on the transient term in the objective function may be chosen very low, while still achieving minimization of overshoot.

4.1 Simulation results

Let us first discuss the simulation results with a fixed integrator gain k_i . Consider Fig. 5, showing the position error and control force for case a) (cf. Fig. 4), for $k_i = 0$ (i.e., a PD controller), and for $k_i = 10$. The former results in a constant steady-state error, and the latter results in overshoot of the setpoint and eventually in limit cycling, both compromising setpoint accuracy. A similar behavior holds for case b), but this case is omitted from the figure for the sake of brevity.

We now employ the learning controller presented in Section 3, to adaptively find a time-varying gain $k_i(t)$ to improve setpoint accuracy. We choose the initialization of $k_i(t)$ (i.e., the initial input vector u_0 of the gradient descent algorithm, see (10)), as $u_0^T := [10 \ 8 \ 6 \ 4 \ 2 \ 0 \ \dots \ 0]$. This particular choice for u_0 embeds the engineering intuition of integrator gain reduction to counteract the reduced friction with increasing velocity, as a result of the Stribeck effect. Consider Fig. 5. Application of the learning controller to both case a) and b) reveals that the resulting position error at the setpoint has decreased significantly, and overshoot is minimized, after convergence of the extremum seeking controller.

Let us now discuss the evolution of $k_i(t)$ and the system's response during the iterative learning process of the controller for friction characteristics a) and b) in more detail. The position error e is reduced for each iteration, see

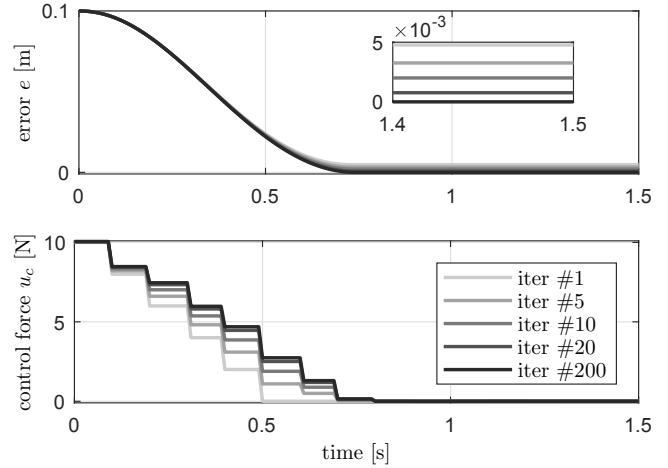


Fig. 6. The setpoint error e and time-varying integrator gain $k_i(t)$ at different iterations for case a).

Fig. 6 and 7 for cases a) and b), respectively. Overshoot is avoided in case a), and minimized in case b), despite the presence of a severe Stribeck effect. The cost function L_p in (5) is effectively minimized by the extremum seeking controller, see Fig. 8. The accuracy of the time-stepping simulator at the setpoint is reached after 180 iterations, but we emphasize that the setpoint accuracy already improved 92% and 86% after only 20 iterations, for cases a) and b), respectively, compared to the first trial.

Consider now Fig. 6 and 7 for cases a) and b), respectively. For case a), the initialization of $k_i(t)$ appears to be a good initial guess, since the optimized $k_i(t)$ takes a similar decaying shape, which results in a high setpoint accuracy. Case b), however, suffers from a significant Stribeck effect. The integrator gain $k_i(t)$ is reduced further during the transient phase, and even takes negative values in order to counteract the rapid reduction in friction force due to the severe Stribeck effect. Nonetheless, a high setpoint accuracy is reached for this case as well, illustrating the achievable performance benefits of the proposed learning controller. Moreover, note that $k_i(t)$ is zero at the setpoint, so that the total control force at this point is significantly lower than the level of static friction. As a result, robustness to force disturbances is automatically obtained.

Remark 5. The resulting $k_i(t)$, as determined by the learning algorithm, depends on the initialization of $k_i(t)$, since

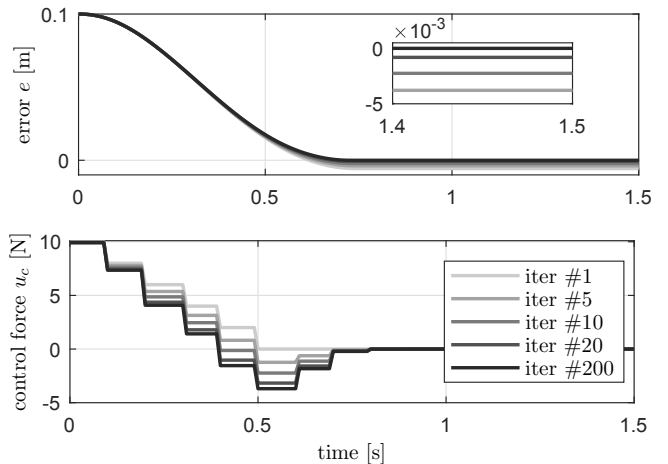


Fig. 7. The setpoint error e and time-varying integrator gain $k_i(t)$ at different iterations for case b).

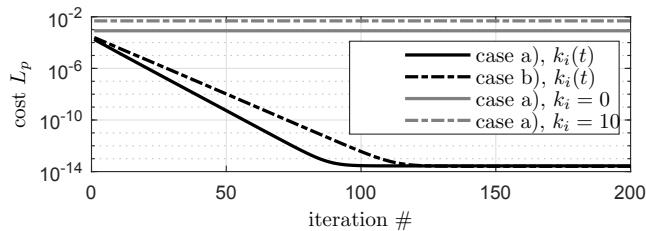


Fig. 8. The cost L_p at different iterations using the proposed controller, for friction characteristic a) (—) and b) (---), and cost L_p for case a) using a fixed $k_i = 10$ (---), and $k_i = 0$ (—).

the gradient descent algorithm only finds *local* optima. Although a high setpoint accuracy can be obtained (see Fig. 5), global optimization methods (e.g., DIRECT and Shubert, see Khong (2013)), are able to find the global optimum and may therefore improve performance even further.

Remark 6. Given $p = 15$ in this example, it is hard to infer the amount of minimizers that the unknown map Q possesses. Many simulations or experiments would be needed to construct Q and investigate its minimizers. Still, we opted for a gradient descent algorithm as opposed to a global optimization algorithm for i) ease of implementation, and ii) the fact that convergence speed of global optimization algorithms can be slow. If one chooses $p \leq 3$, significantly less simulations or experiments are required to construct Q , and it allows visualization of the map Q . This can be helpful to infer the amount of minimizers, and to decide on an appropriate extremum seeking algorithm.

5. CONCLUSION

We have presented a sampled-data extremum-seeking approach for setpoint control of motion systems with friction, performing a repetitive motion. In particular, a PID-based learning controller with a time-varying integrator gain is proposed to deal with unknown or uncertain friction during a setpoint control operation. The time-varying integrator gain is parameterized by using step basis functions, effectively formulating the setpoint control problem as a model-free sampled-data extremum seeking problem. The specific time-varying integrator gain of the learning controller is adaptively obtained by employing a sampled-data extremum-seeking approach. The proposed approach does not require any knowledge of the friction characteristic. The effectiveness of the proposed method is demonstrated by means of a numerical example. The results illustrate that the proposed controller achieves 1) a significantly improved setpoint accuracy for different friction characteristics, and 2) optimizes the transient response by minimizing overshoot. Future work includes a stability analysis of PID-controlled frictional motion systems with a time-varying integrator gain, and study different parametrizations of k_i with less parameters to achieve faster convergence.

REFERENCES

C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction. *Trans. Autom. Control*, 52(10):1988–1994, 2007.

L. Freidovich, A. Robertsson, A. Shiriaev, and R. Johansson, LuGre model-based friction compensation. *Trans. Control Syst. Technol.*, 18(1):194–200, 2010.

N. van de Wouw, R.I. Leine, Robust impulsive control of motion systems with uncertain friction. *Int. J. of Robust and Nonlinear Control*, 22:369–397, 2012.

G. Bartolini, A. Pisano, E. Punta, E. Usai, A survey of applications of second-order sliding mode control to mechanical systems. *Int. J. of Control*, 76:875–892, 2003.

Z. Cao, H-B. Dür, C. Ebenbauer, F. Allgöwer, F. Gao, Iterative learning and extremum-seeking for repetitive time-varying mappings. *Trans. Autom. Control*, 62(7):3339–3353, 2017.

R.H.A. Hensen, M.J.G. van de Molengraft, M. Steinbuch, Friction induced hunting limit cycles: A comparison between the LuGre and switch friction model. *Automatica*, 39(12):2131–2137, 2003.

D.A. Bristow, M. Tharayil, A.G. Alleyne, A survey of iterative learning control. *Control Systems Magazine*, 26:96–114, 2006.

M. Haring, N. van de Wouw, D. Nešić, Extremum-seeking control for nonlinear systems with periodic steady-state outputs. *Automatica*, 49:1883–1891, 2013.

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

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

M. Krstić, H-H. Wang, Stability of extremum-seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4):595–601, 2002.

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

Y. Wang, F. Gao, F.J. Doyle III, Survey on iterative learning control, repetitive control, and run-to-run control. *J. of Process Control*, 19:1589–1600, 2009.

D. Putra, H. Nijmeijer, N. van de Wouw, Analysis of undercompensation and overcompensation of friction in 1-DOF mechanical systems. *Automatica*, 43(8):1387–1394, 2007.

V. Acary, B. Brogliato, Numerical methods for nonsmooth dynamical systems. *Springer-Verlag*, Berlin, 2008.

J. van de Wijdeven, O. Bosgra, Using basis functions in iterative learning control: Analysis and design theory. *Int. J. of Control*, 83(4):661–675, 2010.

J. Bolder, T. Oomen, Rational basis functions in iterative learning control - with experimental verification on a motion system. *Trans. Control Syst. Technol.* 23(2):722–729, 2015.

S. van der Meulen, R.L. Tousain, O.H. Bosgra, Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and desktop printer. *J. Dyn. Syst.-T. ASME*, 130, 051006-1–16, 2008.

M. Nörrlof, S. Gunnarsson, Time and frequency domain convergence properties in iterative learning control. *Int. J. Control*, 75(14):1114–1126, 2002.

S. Boyd, L. Vandenberghe, Convex optimization. *Cambridge University Press*, 2004.

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

B.O. Shubert, A sequential method seeking the global maximum of a function. *Siam J. Numer. Anal.* Vol. 9(3):379–388, 1972.

K. Kvaternik, L. Pavel, Interconnection conditions for the stability of nonlinear sampled-data extremum-seeking schemes. *50th Conf. on Dec. Control and European Control Conf.*, 4448–4454, 2011.