Automatica 66 (2016) 238–245

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Iterative learning control based on extremum seeking[☆]

Sei Zhen Khong[a,1], Dragan Nešić[b], Miroslav Krstić[c]

[a] Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, MN 55455, USA

[b] Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC 3010, Australia

[c] Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093-0411, USA

ARTICLE INFO

Article history:
Received 30 April 2015
Received in revised form
29 October 2015
Accepted 11 November 2015
Available online 27 January 2016

Keywords:
Extremum seeking
Iterative learning control
Non-model based approach
Local and global optimisation

ABSTRACT

This paper proposes a non-model based approach to iterative learning control (ILC) via extremum seeking. Single-input–single-output discrete-time nonlinear systems are considered, where the objective is to recursively construct an input such that the corresponding system output tracks a prescribed reference trajectory as closely as possible on finite horizon. The problem is formulated in terms of extremum seeking control, which is amenable to a range of local and global optimisation methods. Contrary to the existing ILC literature, the formulation allows the initial condition of each iteration to be incorporated as an optimisation variable to improve tracking. Sufficient conditions for convergence to the reference trajectory are provided. The main feature of this approach is that it does not rely on knowledge about the system's model to perform iterative learning control, in contrast to most results in the literature.

1. Introduction

Iterative learning control (ILC) is a learning based method for tracking a prescribed trajectory. It carries out the same task multiple times with respect to recursively updated control inputs while improving the tracking performance by learning from previous executions (Moore, 1993; Moore, Dahleh, & Bhattacharyya, 1992; Xu & Tan, 2003). ILC is known to achieve good performance in the presence of repeating disturbances and certain model uncertainty due to its iteratively learning feature. Practically, ILC has been applied to a wide range of engineering applications, including robotics (Messner, Horowitz, Kao, & Boals, 1991), induction motors (Saab, 2004), rolling mills (Garimella & Srinivasan, 1998), stroke rehabilitation (Freeman, Rogers, Burridge, Hughes, & Meadmore, 2015; Freeman, Rogers, Hughes, Burridge, & Meadmore, 2012), and aluminium extruders (Pandit & Buchheit, 1999); see Ahn, Chen, and Moore (2007) for a classification of the ILC literature. It is also useful within the context of motion planning (Srinivasan & Ruina, 2006). Bristow, Tharayil, and Alleyne (2006) contains an excellent survey of a particular ILC algorithm by Moore (1993), where several topics in analysis (e.g. performance, transients, robustness) and design methods (e.g. plant inversion, quadratically optimal) are covered.

This paper proposes an *extremum-seeking* based framework within which to perform iterative learning control of discrete-time single-input–single-output time-varying nonlinear systems on finite horizon. It is noted here that multi-input–multi-output systems are addressable with the same approach. A key feature of extremum seeking is its ability to locate an optimum with respect to some measure without assuming knowledge about the underlying *models* governing the dynamics of the nonlinear systems (Ariyur & Krstić, 2003; Zhang & Ordóñez, 2011). Such knowledge may be unavailable due to the difficulty associated with modelling of complicated nonlinear systems. Extremum seeking has found applications in a wide array of problems, including biochemical reactors (Guay, Dochain, & Perrier, 2003; Wang, Krstić, & Bastin, 1999), gas-turbine combustors (Moase, Manzie, & Brear, 2010), power electronics (Scheinker, Bland, Krstić, & Audia, 2014), multi-agent source seeking (Khong, Tan, Manzie, & Nešić, 2014), and finite-horizon optimal control (Frihauf, Krstić, & Başar, 2013). Within the context of ILC, extremum seeking has been applied to pulse shaping in a double-pass laser amplifier (Ren, Frihauf, Rafac, & Krstić, 2012).

In this paper, we propose a unifying framework in which to apply optimisation-based extremum seeking algorithms to ILC in the spirit of Khong, Nešić, Tan, and Manzie (2013) and

☆ This work was supported by the Swedish Research Council through the LCCC Linnaeus Centre and the Australian Research Council. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Warren E. Dixon under the direction of Editor Andrew R. Teel.

E-mail addresses: seizhen@control.lth.se (S.Z. Khong), dnesic@unimelb.edu.au (D. Nešić), krstic@ucsd.edu (M. Krstić).

1 Tel.: +46 46 2220362; fax: +46 46 138118.

http://dx.doi.org/10.1016/j.automatica.2015.12.019
0005-1098/

Nešić, Tan, Moase, and Manzie (2010). In particular, the proposed framework is shown to be amenable to a broad range of local and global optimisation methods (Khong, Nešić, Manzie, & Tan, 2013; Pintér, 1996; Teel & Popović, 2001). This allows complexity of implementation and convergence speed of the algorithms to be taken into account in the control design stage. For instance, if large variations in the control input is undesirable but convergence to local optima is tolerable, local optimisation methods may be selected. Furthermore, Newton-based methods can be employed if a quadratic convergence rate is solicited. In the proposed framework, the cost function is defined as the distance between the system output and the reference trajectory. For local optimisation methods, ultimately bounded asymptotic stability of local minima is demonstrated. In the case of global optimisation, it is shown that the proposed ILC converges to a global minimum.

Several optimisation-based ILC methods can be found in the literature, but the vast majority of them rely on knowledge on the models. For instance, the updating control laws as well as convergence of the ILC methods in Gunnarsson and Norrlöf (2001) and Owens and Hätönen (2005) depend on the precise knowledge of the nominal model. Owens, Hatonen, and Daley (2009) proposes a robust monotone gradient-based scheme for ILC of linear time-invariant (LTI) systems, where the multiplicative modelling uncertainty is assumed to be bounded. The robustness analysis therein reminisces that performed in Bristow et al. (2006). Schoellig, Mueller, and D'Andrea (2012) considers the case where an LTI model is subject to noisy disturbances and proposes a combined model-based Kalman filter and convex optimisation approach to ILC. Mishra, Topcu, and Tomizuka (2011) proposes a primal barrier method to ILC of LTI systems contingent on the availability of knowledge about the gradient and Hessian of the quadratic cost function, which in turn is dependent on the models.

While the standard ILC literature considers learning controllers for systems that perform the same operation repetitively under the same initial conditions, we depart from such a setting and incorporate the initial conditions as parts of the optimisation variables, so that they may vary from one iteration to the next for improved tracking. Indeed, the former is subsumed by the latter by setting the initial conditions to be constant across all iterations. The formulation in this paper differs from that of repetitive control (Longman, 2000) and repetitive learning control (Sun, Ge, & Mareels, 2006), where the initial conditions of the current iteration are set to be the final conditions of the previous trial. It is also noteworthy that the proposed extremum seeking based ILC, which updates the control input signal, differs from iterative feedback tuning (Hjalmarsson, Gevers, Gunnarsson, & Lequin, 1998), where non-model based optimisation methods are exploited to iteratively tune controller's parameters in order to achieve tracking of an output trajectory given a fixed reference input.

The paper has the following structure. A formal definition of ILC and the class of nonlinear systems considered in this paper are stated in the next section. In Section 3, ILC is formulated in terms of an extremum seeking problem. Subsequently, local and global optimisation based extremum seeking approaches are discussed in Sections 4 and 5 respectively. Section 6 contains simulation examples illustrating the main results. Finally, some concluding remarks are provided in Section 7.

2. Iterative learning control

The problem of iterative learning control (ILC) is formulated in this section. The special case where the plant is linear time-invariant (LTI) and a commonly used ILC method are reviewed.

2.1. Nonlinear plants

Consider the following dynamical discrete-time time-varying nonlinear state-space system defined over a *finite* time interval/horizon $k = 0, 1, \ldots, T$:

$$x(k+1) = f(x(k), u(k), k) \quad x(0) = \bar{x}; \\ y(k) = h(x(k), u(k), k), \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R} \times \mathbb{T} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R} \times \mathbb{T} \to \mathbb{R}$ are locally Lipschitz functions in each argument and $\mathbb{T} := \{0, 1, \ldots, T\}$. Repeated disturbances that are present on both the state-update differential and state-to-output algebraic equations are accounted for by f and h being functions of the time unit k. The corresponding input–output operator for system (1) is denoted by Σ, whereby $y = \Sigma(\bar{x}, u)$. Also, given a $z : \mathbb{T} \to \mathbb{R}$, define the ℓ^2 norm by

$$\|z\|_2 := \sqrt{\sum_{t=0}^{T} z(k)^2}.$$

With a slight abuse of notation, $\|v\|_2$ is also used to denote the Euclidean norm for the vector $v \in \mathbb{R}^n$. Only discrete-time plants are considered in this paper. It is a natural formulation because ILC uses information from previous trials which needs to be stored on suitable digital media. By the same token, the dynamics of the plant are assumed to evolve along a *finite* horizon $[0, T]$.

Denote by $r : \mathbb{T} \to \mathbb{R}$ the reference trajectory. The *control objective* is to construct a u^* and an $\bar{x}^*$ such that the corresponding system output $y^* = \Sigma(\bar{x}^*, u^*)$ tracks r as accurately as possible. In other words,

$$(\bar{x}^*, u^*) := \arg\min_{u \in \mathcal{U}; \bar{x} \in \Omega} \|r - \Sigma(\bar{x}, u)\|_2,$$

where $\mathcal{U}$ is an appropriate compact subset of $\{u : \mathbb{T} \to \mathbb{R}\}$ and Ω a compact subset of $\mathbb{R}^n$. In general, any ℓ^p-norm may be employable when defining the distance above. Note that a reference r may not be realisable by the system, i.e. there exist no $\bar{x}^*$ and u^* such that $\Sigma(\bar{x}^*, u^*) = r$. In this case, the achievable minimum of the optimisation problem above is nonzero. When Σ is an LTI operator, realisability of references may be studied using the notions of controllability and observability. Characterising this when Σ is nonlinear is a lot harder, and may require knowledge about the solutions to (1).

When f and h are known precisely, a brute-force optimisation over $\bar{x}, u(1), \ldots, u(T)$ can be used to generate a y^* such that the error $e(k) := r(k) - y^*(t)$ is minimised. Alternatively, should this prove to be an infeasible approach, by introducing an additional *iteration-time domain* j, several model-based ILC algorithms in the literature (Moore, 1993; Xu & Tan, 2003) can be used to iteratively design u_j based on previous trials' outputs $y_i = \Sigma(\bar{x}, u_i)$ for $i < j$ such that $u_j \to u^*$ in the ℓ^2-norm for a fixed $\bar{x}$ across all iterations. By tuning the parameters of the ILC algorithms appropriately, the desired transient properties, such as monotone convergence, may be achieved.

Control design in ILC can be specified in the following form. If $\bar{x}_j$ is the initial condition and u_j is the input applied to the plant at trial $j = 0, 1, 2, \ldots$ and $e_j := r - y_j = r - \Sigma(\bar{x}_j, u_j)$ is the resulting tracking error, the control design involves constructing an iteratively updated control law expressed as a functional relationship typified by the equation

$$\bar{x}_{j+1} = g_1(e_j, \ldots, e_{j-s}, u_j, \ldots, u_{j-t}, \bar{x}_j, \ldots, \bar{x}_{j-t}) \\ u_{j+1} = g_2(e_j, \ldots, e_{j-s}, u_j, \ldots, u_{j-t}, \bar{x}_j, \ldots, \bar{x}_{j-t}),$$

where $s, t \leq j$. Ideally, the control law should have the property that $u_\infty := \lim_{j\to\infty} u_j = u^*$ and $\bar{x}_\infty := \lim_{j\to\infty} \bar{x}_j = \bar{x}^*$, or equivalently, $e_\infty := \lim_{j\to\infty} e_j = 0$. A looser requirement on this is that there exists some small $\epsilon > 0$ such that $\|u_\infty - u^*\|_2 < \epsilon$ and

$\|\bar{x}_\infty - \bar{x}^*\|_2 < \epsilon$, i.e. u_j (resp. $\bar{x}_j$) converges to an ϵ-neighbourhood of u^* (resp. $\bar{x}^*$).

3. An extremum seeking approach to iterative learning control

In this section, an extremum seeking approach to ILC of nonlinear system (1) without assuming knowledge on the model is proposed. Unlike the LTI ILC scheme in Bristow et al. (2006), Moore (1993) and Moore et al. (1992), the approach builds upon well-studied optimisation methods (Boyd & Vandenberghe, 2004; Jones, Pertunnen, & Stuckman, 1993), in which the significance of the convergence properties is better understood. This section sets up the framework in which to apply local and global optimisation methods to ILC in subsequent sections.

The object under study is the nonlinear time-varying plant Σ given in (1). For all iterations $j = 0, 1, \ldots$, the system output $y_j : \mathbb{T} \to \mathbb{R}$ is related to the system input $u_j : \mathbb{T} \to \mathbb{R}$ and initial condition $\bar{x}_j \in \mathbb{R}^n$ through the state $x_j : \mathbb{T} \to \mathbb{R}^n$ by

$$\begin{aligned} &x_j(k+1) = f(x_j(k), u_j(k), k) \quad x_j(0) = \bar{x}_j; \\ &y_j(k) = h(x_j(k), u_j(k), k). \end{aligned} \tag{2}$$

The tracking error $\bar{e}_j$ in each iteration is measured by the ℓ^2 distance:

$$\bar{e}_j := \|r - y_j\|_2.$$

Recall that the goal of ILC is to iteratively construct $\bar{x}_j$ and u_j so that $\bar{e}_j$ is minimised as $j \to \infty$. The update equations for $\bar{x}_{j+1}$ and u_{j+1} depend on the tracking errors, initial conditions, and inputs in past executions:

$$\begin{aligned} \bar{x}_{j+1} &= g_1(\bar{e}_j, \ldots, \bar{e}_{j-s}, u_j, \ldots, u_{j-s}, \bar{x}_j, \ldots, \bar{x}_{j-s}) \\ u_{j+1} &= g_2(\bar{e}_j, \ldots, \bar{e}_{j-s}, u_j, \ldots, u_{j-s}, \bar{x}_j, \ldots, \bar{x}_{j-s}), \end{aligned}$$

for some $s \le j$.

Remark 1. In order to optimise over the initial condition of the plant, it is first necessary to know its dimension. In the event that this is inaccessible, one may set the initial condition to be constant across all learning iterations as in standard ILC.

Remark 2. Since the ILC trial duration is finite, the state and output signals of (1) are always bounded in every iteration. In the absence of uniform asymptotic stability of (1), however, the transient response along the trials may be unsatisfactory. In this case, a stabilising controller may be synthesised for the plant (1) and the ILC designed for the resulting stable dynamics.

The formulation above can be transformed into a problem of static optimisation, as shown below. Denote by $\mathcal{U}$ the set of functions $u : \mathbb{T} \to \mathbb{R}$. Given a vector $v \in \mathbb{R}^{n+T+1}$, define the demultiplexer $D : \mathbb{R}^{n+T+1} \to \mathbb{R}^n \times \mathcal{U}$ by

$$\begin{aligned} &D(v) = (w, z) \\ &w = [v_1, \ldots, v_n]^T \\ &z(k) = v_{n+1+k} \quad k = 0, 1, \ldots, T. \end{aligned} \tag{3}$$

Similarly, given a $w \in \mathbb{R}^n$ and $z \in \mathcal{U}$, define the multiplexer $M : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}^{n+T+1}$ by

$$\begin{aligned} &v = M(w, z) \\ &v_i = w_i \quad i = 1, \ldots, n \\ &v_k = z(k) \quad k = n+1, n+2, \ldots, n+T+1. \end{aligned} \tag{4}$$

The demultiplexer and multiplexer are useful for analytically connecting the behaviour of the plant with the optimisation method, as will be demonstrated later.

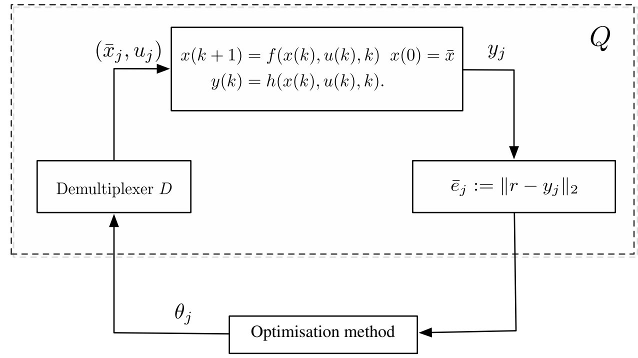


Fig. 1. An extremum seeking based framework for iterative learning control.

Now given $\bar{x}_j \in \mathbb{R}^n$, $u_j \in \mathcal{U}$, and $\theta_j := M(\bar{x}_j, u_j)$, let $Q : \mathbb{R}^{n+T+1} \to \mathbb{R}$ be defined as

$$\begin{aligned} Q(\theta_j) = \|r - \Sigma(D(\theta_j))\|_2 &= \|r - \Sigma(\bar{x}_j, u_j)\|_2 \\ &= \|r - y_j\|_2 \\ &= \bar{e}_j, \end{aligned} \tag{5}$$

where y_j is related to $\bar{x}_j$ and u_j via (2). Note that Q is locally Lipschitz continuous by hypothesis. Also, when Σ is LTI, Q is a convex function on $\mathbb{R}^{n+T+1}$. The process above transforms the problem of ILC to one of static optimisation, to which a broad array of local and global algorithms is applicable. In particular, θ_{j+1} can be designed iteratively based on some update relationship

$$\theta_{j+1} = \Gamma(\bar{e}_j, \ldots, \bar{e}_{j-s}, \theta_j, \ldots, \theta_{j-s})$$

and applied to Σ via the demultiplexer D. Fig. 1 illustrates the extremum seeking based approach to ILC. Notice that the demultiplexer accepts as its input a vector of real numbers from the optimisation method and outputs a corresponding initial condition and control input signal to the plant.

In the succeeding sections, both local (gradient based) and non-convex global (sampling based) optimisation methods are explored in the setup shown in Fig. 1.

4. Local optimisation

Local optimisation methods generally employ estimations about the (first or higher order) derivatives of the cost function in updating the inputs. This section demonstrates how they can be applied to performing ILC within the extremum seeking based framework introduced in the previous section.

First, some notation is introduced. A function $\gamma : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ is of class-$\mathcal{K}$ (denoted $\gamma \in \mathcal{K}$) if it is continuous, strictly increasing, and $\gamma(0) = 0$. If γ is also unbounded, then $\gamma \in \mathcal{K}_\infty$. A continuous function $\beta : \mathbb{R}_{\ge 0} \times \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ is of class-$\mathcal{KL}$ if for each fixed t, $\beta(\cdot, t) \in \mathcal{K}$ and for each fixed s, $\beta(s, \cdot)$ is decreasing to zero (Khalil, 2002). Given any subset $\mathcal{X}$ of $\mathbb{R}^m$ and a point $x \in \mathbb{R}^m$, define the distance of x from $\mathcal{X}$ as $\|x\|_{\mathcal{X}} := \inf_{a \in \mathcal{X}} \|x - a\|_2$.

Consider the following static function optimisation problem:

$$z^* := \min_{\theta \in \Omega} Q(\theta), \tag{6}$$

where $Q : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable and $\Omega \subset \mathbb{R}^m$ is compact. Assume that the Jacobian $\nabla Q = 0$ in a nonempty, compact set $\mathcal{C} \subset \mathbb{R}^m$, i.e. Q achieves its minimum on $\mathcal{C}$. A local optimisation method Γ may generate its update according to

$$\theta_{j+1} = \Gamma(\theta_j, \ldots, \theta_{j-1}, z_j, \ldots, z_{j-s}),$$

where $z_i = Q(\theta_i)$. The following assumption is satisfied by a broad class of local gradient-based optimisation algorithms.

Assumption 3. The optimisation method Γ, when applied to (6), satisfies the following conditions:

(i) Γ is time-invariant. Denote by $\{\theta_j\}_{j=0}^{\infty} \subset \Omega \subset \mathbb{R}^m$ the output sequence Γ generates based on inputs to Γ, $\{z_j\}_{j=1}^{\infty}$, where $z_j := Q(\theta_j)$. Γ is causal in the sense that the output at any time $N \in \mathbb{N}$, i.e. θ_N, is determined based only on θ_j and z_j for $j = 0, 1, \ldots, N-1$, i.e. the past probe values to Q and the corresponding measurements.

(ii) Denote by $\mathcal{S}(\theta_0)$ the set of all admissible output sequences of Γ with respect to the initial point θ_0. There exists a class-$\mathcal{KL}$ function β such that for any initial point $\theta_0 \in \Omega$, all outputs $\theta \in \mathcal{S}(\theta_0)$ satisfy for some $\delta \geq 0$

$$\|\theta_j(\theta_0)\|_{\mathcal{C}} \leq \beta(\|\theta_0\|_{\mathcal{C}}, j) + \delta \quad \forall j \geq 0. \tag{7}$$

Note that Assumption 3(ii) states that the sequence θ_j converges asymptotically to a δ-neighbourhood of $\mathcal{C}$.

Remark 4. The set of outputs $\mathcal{S}(\theta_0)$ in Assumption 3(ii) arises, for example, from modelling the optimisation algorithm with a difference inclusion involving a set-value 'state-update' map F by $\theta^+ \in F(\theta, Q(\theta))$; see Kellet and Teel (2005). In the simplest case, $\mathcal{S}(\theta_0)$ is a singleton, i.e. there is only one possible output sequence given a fixed initial condition. For example, one modelled by the difference equation $\theta^+ = F(\theta, Q(\theta))$.

Recall the demultiplexer D and multiplexer M defined in (3) and (4) respectively.

Theorem 5. *Given a nonlinear plant Σ in (1) and a reference trajectory $r \in \mathcal{U}$, the feedback interconnection shown in Fig. 1 with the optimisation method Γ satisfying Assumption 3 in which $m = n+T+1$ has the following convergence property: there exists a class-$\mathcal{KL}$ function β such that for any initial point $\theta_0 \in \Omega$ and for some $\delta \geq 0$,*

$$\|M(\bar{x}_j, u_j)\|_{\mathcal{C}} \leq \beta(\|M(\bar{x}_0, u_0)\|_{\mathcal{C}}, j) + \delta \quad \forall j \geq 0, \tag{8}$$

where $Q : \Omega \subset \mathbb{R}^m \to \mathbb{R}$ is as defined in (5), whose Jacobian is zero on $\mathcal{C} \subset \Omega$, the search domain of interests.

Proof. Note that by the setup of the feedback interconnection in Fig. 1, $(\bar{x}_j, u_j) = D(\theta_j)$ for all $j = 0, 1, \ldots$. By applying Γ to Q, it follows from Assumption 3 that there exists a class-$\mathcal{KL}$ function β such that for any initial point $\theta_0 \in \Omega$ and some $\delta \geq 0$,

$$\|\theta_j\|_{\mathcal{C}} \leq \beta(\|\theta_0\|_{\mathcal{C}}, j) + \delta \quad \forall j \geq 0.$$

This is equivalent, via the relationship $(\bar{x}_j, u_j) = D(\theta_j)$, to (8), as claimed. □

Theorem 5 contains sufficient conditions under which the iterations on $\bar{x}_j$ and u_j are convergent in the ℓ^2 sense to a δ-neighbourhood of local minima of the function Q defined in (5), which measures the ℓ^2-size of the tracking error.

In the following, we detail two of the most well-known methods (Boyd & Vandenberghe, 2004; Polak, 1997) in operations research which satisfy Assumption 3. They are (i) the *gradient descent method*:

$$\theta_{j+1} = \theta_j - \lambda_j \nabla Q(\theta_j), \tag{9}$$

where λ_i denotes the step size which can be computed by, say, the Armijo method (Polak, 1997, Alg. 1.3.3) and (ii) *Newton's method*:

$$\theta_{j+1} = \theta_j - \nabla^2 Q(\theta_j)^{-1} \nabla Q(\theta_j),$$

where $\nabla Q(\cdot)$ and $\nabla^2 Q(\cdot)$ denote, respectively, the Jacobian and Hessian of Q. It can be readily seen that the gradient and Newton methods satisfy the time-invariance and causality Assumption 3(i). The following result can be found in Boyd and Vandenberghe (2004) and Polak (1997).

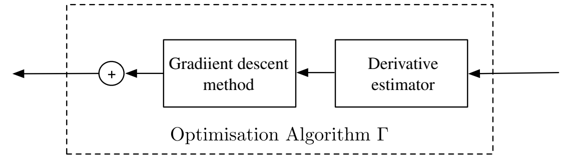


Fig. 2. A gradient-based extremum seeking controller paradigm.

Proposition 6. *Suppose $Q : \Omega \to \mathbb{R}$ is twice Lipschitz continuously differentiable and strictly convex on $\mathcal{S} \subset \Omega$, whereby there exist $\underline{M}, \bar{M} \in \mathbb{R}$ such that*

$$\underline{M}I \leq \nabla^2 Q(\theta) \leq \bar{M}I \quad \text{for all } \theta \in \mathcal{S}.$$

Furthermore, suppose there exists a minimiser $\theta^* \in \mathcal{S}$ such that $\nabla Q(\theta^*) = 0$. Let $\{\theta_j\}_{j=0}^{\infty}$ be the sequence generated by the gradient or Newton method when applied to minimising Q. Then there exists a class-$\mathcal{KL}$ function β such that for any $\theta_0 \in \mathcal{S}$,

$$\|\theta_j - \theta^*\|_2 \leq \beta(\|\theta_0 - \theta^*\|_2, j) \quad \forall j \geq 0. \tag{10}$$

Note that the rate of convergence for the gradient descent method is linear while that for Newton is quadratic, at least within a sufficiently small neighbourhood of the minimiser.

Proposition 6 states the convergence conditions for the gradient descent and Newton method when exact values of the Jacobian $\nabla Q(\theta_j)$ and Hessian $\nabla^2 Q(\theta_j)$. In practice, they need to be estimated from several past measurements. This can be achieved by using the Euler methods, trapezoidal method, or the more sophisticated Runge–Kutta methods (Press, Teukolsky, Vetterling, & Flannery, 2007); see Fig. 2.

To be more specific, let the initial output of the optimisation method be $\theta_0 := \tau_0$. As determined by the derivative estimator, the following length-p sequence of step commands $\{\theta_j\}_{j=0}^{p-1}$ can be used to probe Q along the desired directions:

$$(\tau_0 + d_1(\tau_0), \ldots, \tau_0 + d_p(\tau_0)), \tag{11}$$

where $d_i : \Omega \to \mathbb{R}^m$, $i = 1, \ldots, p$ denote the dither signals. The corresponding outputs of Q are then collected by the derivative estimator to numerically approximate the Jacobian $\nabla Q(\theta_0)$. Exploiting this information, the optimisation algorithm can then update its next probing point τ_1, and the series of steps described above repeats to generate $\{\theta_j\}_{j=p}^{2p-1}$.

Suppose the use of the derivative estimates (instead of their precise values) in Fig. 2 introduces a bounded additive error term in the update of the gradient and Newton methods:

$$\begin{aligned} &\theta_{j+1} = \theta_j - \lambda_j \nabla Q(\theta_j) + e_1(k, \theta_j) \quad \text{and} \\ &\theta_{j+1} = \theta_j - \nabla^2 Q(\theta_j)^{-1} \nabla Q(\theta_j) + e_2(k, \theta_j), \end{aligned} \tag{12}$$

where

$$\begin{aligned} &\|e_1(k, \theta_j)\|_2 \leq l_1 + q_1 \alpha(\|\theta_j\|_{\mathcal{C}}) \quad \text{and} \\ &\|e_2(k, \theta_j)\|_2 \leq l_2 + q_2 \alpha(\|\theta_j\|_{\mathcal{C}}), \end{aligned} \tag{13}$$

for some $l_1, l_2, q_1, q_2 \geq 0$. It follows from the non-vanishing perturbation results for discrete-time systems in Cruz-Hernández, Alvarez-Gallegos, and Castro-Linares (1999) that for sufficiently small l and q, the gradient/Newton-based extremum seeking controller in Fig. 2 satisfies the ultimately bounded asymptotic stability Assumption 3(ii). In particular, there exist a class-$\mathcal{K}$ function α and a class-$\mathcal{KL}$ function β such that

$$\|\theta_j\|_{\mathcal{C}} \leq \beta(\|\theta_0\|_{\mathcal{C}}, j) + \alpha(l) \quad \forall j = 0, 1, \ldots.$$

Assumption 7. There exists an $\alpha_d \in \mathcal{K}$ and $c > 0$ such that the dither signals in (11) satisfy for each $i = 1, \ldots, p$,

$$\|d_i(\theta)\|_2 \leq \alpha_d(\|\theta\|_{\mathcal{C}}) + c.$$

When Assumption 7 holds with $c = 0$, it follows that the step size used in estimating the derivatives converges to zero as θ_j tends to the minimising set $\mathcal{C}$. This implies by the definition of differentiation that the magnitudes of the error terms e_1 and e_2 in (13) tend to zero as $k \to \infty$, i.e. $l_1 = l_2 = 0$. In other words, the perturbations are vanishing and the extremum seeking controller is asymptotically stable as in Assumption 3(ii) with $\delta = 0$ (Cruz-Hernández et al., 1999).

Remark 8. When the plant Σ in Theorem 5 is LTI, the resulting Q in (5) is convex. It follows that either the gradient descent or Newton methods would result in convergence to a neighbourhood of a global minimum of Q. On the contrary, if Σ is nonlinear, convergence to a neighbourhood of a local minimum can be achieved.

Besides the gradient descent and Newton methods detailed above, other local optimisation methods satisfying Assumption 3 include the following difference inclusion form (Teel & Popović, 2001):

$$\theta^+ \in F(\theta, G(\theta)), \tag{14}$$

where F is an upper semi-continuous set-valued map (the update θ^+ can be any element of the set) and G is a function that carries information regarding the estimate of the gradient of Q around θ. In particular, F maps from $\mathbb{R}^m \times \mathbb{R}^p$ to subsets of $\mathbb{R}^m$,

$$G(\theta_j) := \begin{bmatrix} Q(\theta_j + d_1(\theta_j)) \\ \vdots \\ Q(\theta_j + d_p(\theta_j)) \end{bmatrix},$$

and $d_i : \Omega \to \mathbb{R}$, $i = 1, \ldots, p$ are dither/perturbation functions. See Teel (2000) for a class of Lyapunov-based non-smooth optimisation algorithms of the form described above which employ the notion of Clarke generalised gradient.

5. Global optimisation

In the case where large variations in the initial condition and control input can be tolerated from iteration to iteration, sampling-based global optimisation methods which do not exploit information about gradients of the cost function can be employed to locate global minima in the presence of local ones in ILC. Bounds on initial conditions and input saturation constraints are naturally accommodated by these methods. This section adapts such methods for ILC within the framework of extremum seeking control described in Section 3.

Consider the following bound-constrained optimisation problem:

$$z^* := \min_{\theta \in \Omega} Q(\theta), \tag{15}$$

where

$$\Omega := \left\{ \theta \in \mathbb{R}^m \mid \theta_i \in [a_i, b_i] \subset \mathbb{R}, i = 1, 2, \ldots, m \right\}. \tag{16}$$

Assume that $Q : \Omega \subset \mathbb{R}^m \to \mathbb{R}$ is a globally Lipschitz continuous function (Khalil, 2002), i.e. there exists a known $L > 0$ such that

$$|Q(\theta) - Q(\theta')| \le L\|\theta - \theta'\|_2$$

for all $\theta, \theta' \in \Omega$. Note that Q possesses a global minimum by the extreme value theorem (Rudin, 1976, Thm. 4.16) since it is (Lipschitz) continuous on a compact domain Ω. The compactness is motivated by the ubiquity of control input saturation constraints in physical systems (Khalil, 2002). It is assumed that Q achieves its global minimum on a nonempty set $\mathcal{C} \subset \Omega$.

A global optimisation method Γ may generate its update according to

$$\theta_{j+1} = \Gamma(\theta_j, \ldots, \theta_{j-1}, z_j, \ldots, z_{j-s}),$$

where $z_i = Q(\theta_i)$. Define

$$\hat{z}_k := \min_{i=0,1,\ldots,k} z_i.$$

In what follows we consider global optimisation algorithms that satisfy the next assumption.

Assumption 9. The global optimisation algorithm Γ, when applied to optimisation (15), has the convergence property that a subsequence of $\{\theta_j\}_{j=0}^{\infty}$ converges to $\mathcal{C}$, which implies that $\hat{z}_k \to z^*$.

Theorem 10. *Given a nonlinear plant Σ in (1) and a reference trajectory $r \in \mathcal{U}$, the feedback interconnection shown in Fig. 1 with the optimisation method Γ satisfying Assumption 9 in which $m = n + T + 1$ has the following convergence property: $\{M(\bar{x}_j, u_j)\}_{j=0}^{\infty}$ has a subsequence which converges to $\mathcal{C}$, where $Q : \Omega \subset \mathbb{R}^m \to \mathbb{R}$ as defined in (5) achieves its global minimum on a nonempty set $\mathcal{C} \subset \Omega$ and Ω is as in (16) for some $a_i, b_i \in \mathbb{R}$, $i = 1, \ldots, m$.*

Proof. It can be seen from the feedback interconnection in Fig. 1 that $(\bar{x}_j, u_j) = D(\theta_j)$ for all $j = 0, 1, \ldots$. By applying Γ to Q, it follows from Assumption 9 that a subsequence of $\{\theta_j\}_{j=0}^{\infty}$ converges to $\mathcal{C}$. This is equivalent, via the relationship $(\bar{x}_j, u_j) = D(\theta_j)$, to $\{M(\bar{x}_j, u_j)\}_{j=0}^{\infty}$ having a subsequence which converges to $\mathcal{C}$. □

While a wide range of global optimisation algorithms satisfying Assumption 9 are available in the literature (Pintér, 1996; Strongin & Sergeyev, 2000), we describe in the following a particular one called the DIRECT (DIviding RECTangles) optimisation method (Jones et al., 1993) to be used in the simulation example later.

5.1. DIRECT optimisation algorithm

Here, a brief review of the DIRECT optimisation method (Jones et al., 1993) is given. DIRECT is a deterministic sampling method which solves (15). The only assumption that DIRECT makes is the Lipschitz continuity of Q. DIRECT does not require knowledge of the function Q or the Lipschitz constant. It also makes no attempts at estimating derivatives of Q.

Algorithm 1. The DIRECT algorithm (Jones et al., 1993).

Given: A Lipschitz function $Q : \Omega \subset \mathbb{R}^m \to \mathbb{R}$.

Notation: q denotes the iteration number of DIRECT and k the total number of samples. Input samples taken by DIRECT are denoted $\theta_i \in \Omega$, $i = 0, 1, \ldots, k-1$ and the corresponding outputs $z_i := Q(\theta_i)$.

(i) Initialise $q := 1$ and $k := 0$.
(ii) Evaluate $Q(\theta_0)$, where $\theta_0 \in \mathbb{R}^m$ denotes the centre point of Ω. Set $\hat{z}_0 := Q(\theta_0)$ and increment k, i.e. $k^+ := k + 1$.
(iii) Identify the set of indices of potentially optimal hyper-rectangles S, i.e. all $j \in \{0, \ldots, k-1\}$ for which there exists a positive $\tilde{L} \in \mathbb{R}$ such that

$$\begin{aligned} &Q(\theta_j) - \tilde{L}d_j \le Q(\theta_i) - \tilde{L}d_i \quad \forall i = 0, \ldots, k-1 \\ &Q(\theta_j) - \tilde{L}d_j \le \hat{z}_k - \epsilon|\hat{z}_k| \quad \text{for some } \epsilon > 0, \end{aligned} \tag{17}$$

where d_i denotes the distance from the centre point to the vertices of the ith hyper-rectangle. As noted in Jones et al. (1993), the set of potentially optimal hyper-rectangles can be found using Graham's scan (Graham, 1972), which is an efficient algorithm for determining the convex hull of a finite set.

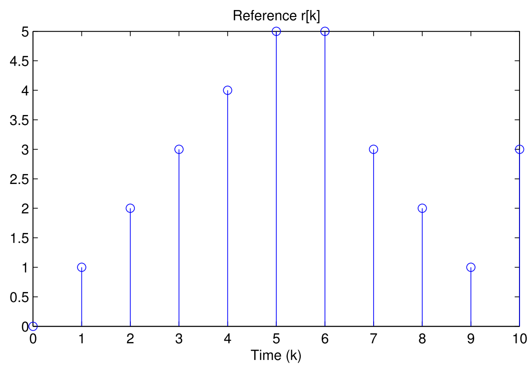


Fig. 3. Reference trajectory.

(iv) For every $j \in S$, subdivide the jth hyper-rectangle with centre θ_j according to the following rule. First identify the set I of dimensions $i \in \{1, \ldots, m\}$ in which the jth hyper-rectangle has the maximum side length and let δ be one-third of this value. Sample Q at the points $\theta_j \pm \delta e_i$ for all $i \in I$, where e_i denotes the ith unit vector in $\mathbb{R}^m$. Sequentially divide the jth hyper-rectangle into thirds along the dimension $i \in I$ in an ascending order of $\min\{Q(\theta_j + \delta e_i), Q(u_j - \delta e_i)\}$. Set $k^+ := k + \Delta k$, where Δk is the number of *new* points sampled during the qth iteration.

(v) Set $q^+ := q + 1$ and the minimal estimate

$$\hat{z}_q := \min_{i=0,1,\ldots,k} z_i. \tag{18}$$

(vi) Loop from (iii).

See Jones et al. (1993) for more details regarding the DIRECT method, including a pictorial illustration. Finkel (2004) contains a numerical implementation of the algorithm using MATLAB. Since DIRECT identifies all potentially optimal hyper-rectangles via (17) for subdivision, it is well-balanced between local and global searches. Indeed, once DIRECT locates the basin of convergence of a global optimum, the local part of the algorithm automatically exploits it to expedite the search (Jones et al., 1993). The following demonstrates that the DIRECT algorithm satisfies Assumption 9, whereby Theorem 10 holds with DIRECT being the optimisation method.

Proposition 11 (*Jones et al., 1993*). *As the number of iterations approaches infinity, the points sampled by DIRECT form a dense subset of Ω. Since Q is Lipschitz continuous, the estimate by DIRECT converges to z^* in (15). Mathematically, it holds that*

$$\lim_{q\to\infty} \hat{z}_q = z^* := \min_{\theta\in\Omega} Q(\theta).$$

The denseness in domain-sampling property of DIRECT mentioned in the proposition above is of critical importance in the proof of its convergence. It holds by the way the algorithm is set up, which subdivides all potentially optimal rectangles in the search space iteratively.

6. Illustrative examples

Consider the following forced pendulum equation modified from Khalil (2002, Example 4.4):

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\sin(x_1) - 0.2x_2 + u$$
$$y = x_1.$$

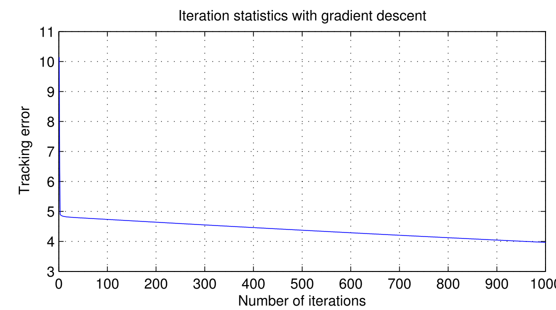


Fig. 4. Tracking error vs. the number of iterations of gradient descent.

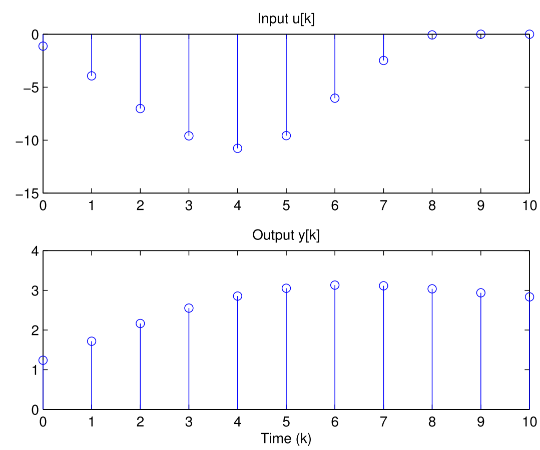


Fig. 5. Minimising input and the corresponding output from gradient descent.

Discretising the plant using Euler's method with respect to a sampling period of 0.1s yields

$$\begin{aligned} x_1(k+1) &= x_1(k) + 0.1x_2(k) \\ x_2(k+1) &= -0.1\sin(x_1(k)) + 0.98x_2(k) + 0.1u(k). \end{aligned} \tag{19}$$

Issues arising from the discretisation phase are not considered in this example; they can be dealt with using, for instance, the analysis in Nešić, Teel, and Kokotović (1999). Let $T = 10$. Fig. 3 illustrates a prescribed reference trajectory r to be tracked. It is noted here that as constrained by the dynamics of the plant (19), such a trajectory r is not realisable as an output. In what follows, both local and global optimisation methods will be applied to ILC of (19) with respect to the prescribed reference.

6.1. *Gradient descent method*

The extremum seeking based ILC scheme in Fig. 1 is employed for the plant (19) with the optimisation method being the gradient descent (9). The optimisation step size $\lambda_j := 0.5\ \forall j$ and first-order Euler approximation is used for estimating the derivative with a step size of 0.01, where 14 samples are needed to compute one estimation. In particular, the dither sequence is given by $d_1 = 0$ and $d_i = 0.01 * \epsilon_{i-1}$ for $i = 2, \ldots, 14$, where $\epsilon_i \in \mathbb{R}^{13}$ denotes the canonical vector containing only zero entries except the ith-entry, where it is equal to 1. Both the initial condition and starting input are set to 0.

A total of 1000 iterations of the gradient descent method are simulated, which result in 14 000 executions of the plants. The final ℓ^2 tracking error is 3.97. Fig. 4 illustrates the tracking error with respect to the number of iterations of the gradient descent method. The resulting initial condition is [1.24, 4.78]. Fig. 5 shows the resulting input and the corresponding output.

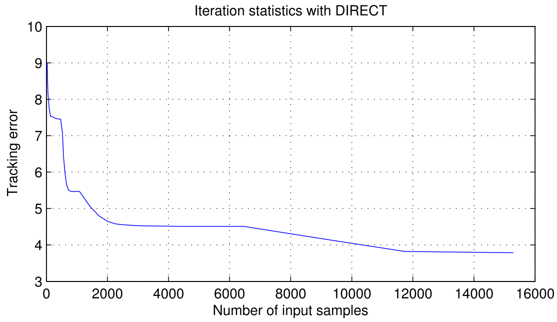


Fig. 6. Tracking error vs. the number of samples taken by DIRECT.

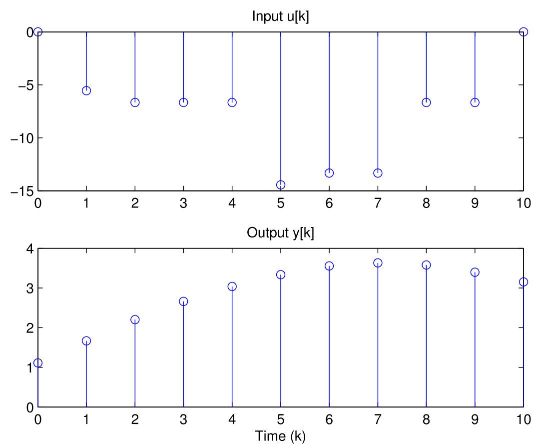


Fig. 7. Minimising input and the corresponding output from DIRECT.

6.2. DIRECT method

The same ILC scheme in Fig. 1 is applied to the plant (19), but with the optimisation method being the DIRECT Algorithm 1. Suppose that each coordinate of the initial condition and input lies within the compact interval $[-15, 15]$, i.e. $a_i = -15$ and $b_i = 15$ for $i = 1, \ldots, 13$ in (16).

After 34 iterations of the DIRECT algorithm and 15 293 runs of the plant, the final ℓ^2 tracking error achieved is 3.79. Fig. 6 illustrates the tracking error with respect to the number of runs of the plant. The resulting initial condition is $[1.11, 5.56]$. Fig. 7 shows the resulting input and the corresponding output.

7. Conclusions

An extremum seeking based framework has been proposed for iterative learning control of discrete-time nonlinear time-varying systems. It is demonstrated that the framework accommodates both local and global optimisation algorithms. Simulation examples are provided to illustrate the results. Future research directions may involve investigating iterative learning under noisy measurements and stochastic optimisation methods.

References

Ahn, H.-S., Chen, Y. Q., & Moore, K. L. (2007). Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *37*(6), 1099–1121.

Ariyur, K. B., & Krstić, M. (2003). *Real-time optimization by extremum seeking control*. Wiley-Interscience.

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

Bristow, D., Tharayil, M., & Alleyne, A. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, *26*(3), 96–114.

Cruz-Hernández, C., Alvarez-Gallegos, J., & Castro-Linares, R. (1999). Stability of discrete nonlinear systems under nonvanishing perturbations: application to a nonlinear model-matching problem. *IMA Journal of Mathematical Control and Information*, *16*, 23–41.

Finkel, D.E. (2004). DIRECT research and codes. http://www4.ncsu.edu/~ctk/Finkel_Direct/.

Freeman, C., Rogers, E., Burridge, J., Hughes, A.-M., & Meadmore, K. (2015). *Iterative learning control for electrical stimulation and stroke rehabilitation*. Springer.

Freeman, C., Rogers, E., Hughes, A.-M., Burridge, J., & Meadmore, K. (2012). Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation. *IEEE Control Systems*, *32*(1), 18–43.

Frihauf, P., Krstić, M., & Başar, T. (2013). Finite-horizon LQ control for unknown discrete-time linear systems via extremum seeking. *European Journal of Control*, *19*, 399–407.

Garimella, S., & Srinivasan, K. (1998). Application of iterative learning control to coil-to-coil control in rolling. *IEEE Transactions on Control Systems Technology*, *6*(2), 281–293.

Graham, R. L. (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, *1*, 132–133.

Guay, M., Dochain, D., & Perrier, M. (2003). Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. *Automatica*, *39*, 1283–1293.

Gunnarsson, S., & Norrlöf, M. (2001). On the design of ILC algorithms using optimization. *Automatica*, *37*, 2011–2016.

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

Jones, D. R., Pertunnen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, *79*(1), 157–181.

Kellet, C. M., & Teel, A. R. (2005). On the robustness of $\mathcal{KL}$-stability for diffrence inclusions: smooth discrete-time Lyapunov functions. *SIAM Journal on Control and Optimization*, *44*(3), 777–800.

Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Prentice Hall.

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

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

Khong, S. Z., Tan, Y., Manzie, C., & Nešić, D. (2014). Multi-agent source seeking via discrete-time extremum seeking control. *Automatica*, *50*, 2312–2320.

Longman, R. (2000). Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, *73*(10), 930–954.

Messner, W., Horowitz, R., Kao, W.-W., & Boals, M. (1991). A new adaptive learning rule. *IEEE Transactions on Automatic Control*, *36*(2), 245–251.

Mishra, S., Topcu, U., & Tomizuka, M. (2011). Optimization-based constrained iterative learning control. *IEEE Transactions on Control Systems Technology*, *19*(6), 1613–1621.

Moase, W. H., Manzie, C., & Brear, M. J. (2010). Newton-like extremum-seeking for the control of thermoacoustic instability. *IEEE Transactions on Automatic Control*, *55*(9), 2094–2105.

Moore, K. L. (1993). *Iterative learning control for deterministic systems*. Springer.

Moore, K. L., Dahleh, M., & Bhattacharyya, S. P. (1992). Iterative learning control: A survey and new results. *Journal of Intelligent and Robotic Systems*, *9*(5), 563–594.

Nešić, D., Tan, Y., Moase, W.H., & Manzie, C. (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).

Nešić, D., Teel, A. R., & Kokotović, P. V. (1999). Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, *38*, 259–270.

Owens, D. H., & Hätönen, J. (2005). Interative learning control—an optimization paradigm. *Annual Reviews in Control*, *29*, 57–70.

Owens, D. H., Hatonen, J. J., & Daley, S. (2009). Robust monotone gradient-based discrete-time iterative learning control. *International Journal of Robust and Nonlinear Control*, *19*, 634–661.

Pandit, M., & Buchheit, K.-H. (1999). Optimizing iterative learning control of cyclic production prprocess with application to extruders. *IEEE Transactions on Control Systems Technology*, *7*(3), 382–390.

Pintér, J. D. (1996). *Global optimization in action: continuous and Lipschitz optimization: algorithms, implementations and applications*. Kluwer Academic Publishers.

Polak, E. (1997). *Applied mathematical sciences*: *Vol. 124. Optimization—algorithms and consistent approximations*. Springer.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: the art of scientific computing* (3rd ed.). Cambridge University Press.

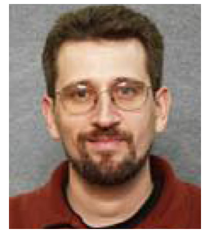
Ren, B., Frihauf, P., Rafac, R. J., & Krstić, M. (2012). Laser pulse shaping via extremum seeking. *Control Engineering Practice*, *20*, 674–683.

Rudin, W. (1976). *Principles of mathematical analysis* (3rd ed.). McGraw-Hill.

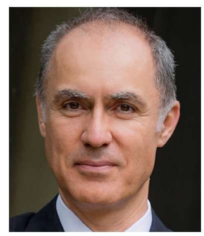
Saab, S. A. (2004). A stochastic iterative learning conrol algorithm with application to an induction motor. *International Journal of Control*, *77*(2), 144–163.

Scheinker, A., Bland, M., Krstić, M., & Audia, J. (2014). Extremum seeking-based optimization of high voltage converter modulator rise-time. *IEEE Transactions on Control Systems Technology*, *22*(1), 34–43.
Schoellig, A. P., Mueller, F. L., & D'Andrea, R. (2012). Optimization-based iterative learning for precise quadrocopter trajectory tracking. *Autonomous Robots*, *33*(1–2), 103–127.
Srinivasan, M., & Ruina, A. (2006). Computer optimization of a minimal biped model discovers walking and running. *Nature*, *439*, 72–75.
Strongin, R. G., & Sergeyev, Y. D. (2000). *Nonconvex optimization and its applications, Global optimization with non-convex constraints*. Kluwer Academic Publishers.
Sun, M., Ge, S. S., & Mareels, I. M. Y. (2006). Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning. *IEEE Transactions on Robotics*, *22*(3), 563–568.
Teel, A.R. (2000). Lyapunov methods in nonsmooth optimization, part I: Quasi-Newton algorithms for Lipschitz, regular functions and part II: persistently exciting finite differences. In *Proceedings of IEEE conference on decision and control*.
Teel, A.R., & Popović, D. (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).
Wang, H.-H., Krstić, M., & Bastin, G. (1999). Optimizing bioreactors by extremum seeking. *International Journal of Adaptive Control and Signal Processing*, *13*, 651–699.
Xu, J.-X., & Tan, Y. (2003). *Linear and nonlinear iterative learning control*. Springer.
Zhang, C., & Ordóñez, R. (2011). *Advances in industrial control, Extremum-seeking control and applications: a numerical optimization-based approach*. Springer.

Sei Zhen Khong received the Bachelor of Electrical Engineering degree (with first class honours) and the Ph.D. degree from The University of Melbourne, Australia, in 2008 and 2012, respectively. He held postdoctoral research fellowships in the Department of Electrical and Electronic Engineering, The University of Melbourne and the Department of Automatic Control, Lund University, Sweden. He is now a postdoctoral researcher at the Institute for Mathematics and its Applications at the University of Minnesota, USA. He received the 2011 Australian Control Conference Best Student Paper High Commendation Award. His research interests include distributed analysis of heterogeneous networks, robust control, linear systems theory, extremum seeking control, and sampled-data control.



Dragan Nešić is a Professor in the Department of Electrical and Electronic Engineering (DEEE) at The University of Melbourne, Australia. He received his BE degree in Mechanical Engineering from The University of Belgrade, Yugoslavia in 1990, and his Ph.D. degree from Systems Engineering, RSISE, Australian National University, Canberra, Australia in 1997. Since February 1999 he has been with The University of Melbourne. His research interests include networked control systems, discrete-time, sampled-data and continuous-time nonlinear control systems, input-to-state stability, extremum seeking control, applications of symbolic computation in control theory, hybrid control systems, and so on. He was awarded a Humboldt Research Fellowship (2003) by the Alexander von Humboldt Foundation, an Australian Professorial Fellowship (2004–2009) and Future Fellowship (2010–2014) by the Australian Research Council. He is a Fellow of IEEE and a Fellow of IEAust. He is currently a Distinguished Lecturer of CSS, IEEE (2008–). He served as an Associate Editor for the journals Automatica, IEEE Transactions on Automatic Control, Systems and Control Letters and European Journal of Control.



Miroslav Krstic holds the Alspach endowed chair and is the founding director of the Cymer Center for Control Systems and Dynamics at UC San Diego. He also serves as Associate Vice Chancellor for Research at UCSD. As a graduate student, Krstic won the UC Santa Barbara best dissertation award and student best paper awards at CDC and ACC. Krstic is a Fellow of IEEE, IFAC, ASME, SIAM, and IET (UK), and Associate Fellow of AIAA. He has received the PECASE, NSF Career, and ONR Young Investigator awards, the Axelby and Schuck paper prizes, the Chestnut textbook prize, the ASME Nyquist Lecture Prize, and the first UCSD Research Award given to an engineer. Krstic has also been awarded the Springer Visiting Professorship at UC Berkeley, the Distinguished Visiting Fellowship of the Royal Academy of Engineering, the Invitation Fellowship of the Japan Society for the Promotion of Science, and the Honorary Professorships from the Northeastern University (Shenyang) and the Chongqing University, China. He serves as Senior Editor in IEEE Transactions on Automatic Control and Automatica, as editor of two Springer book series, and has served as Vice President for Technical Activities of the IEEE Control Systems Society and as chair of the IEEE CSS Fellow Committee. Krstic has coauthored eleven books on adaptive, nonlinear, and stochastic control, extremum seeking, control of PDE systems including turbulent flows, and control of delay systems.