2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Data-Efficient Static Cost Optimization via Extremum-Seeking Control with Kernel-Based Function Approximation*

Wouter Weekers[1], Alessandro Saccon[1], and Nathan van de Wouw[1]

Abstract— **We present a novel type of sampled-data extremum-seeking control (ESC) aimed at speeding up convergence to the optimum and reducing the number of costly performance measurements in practical applications. The approach uses collected output measurements to construct online an approximation of the system's steady-state performance function using kernel-based function approximation. In regions where this approximation is detected to be sufficiently accurate, the proposed approach utilizes it to determine the search direction and compute a suitable optimizer gain for the update step. In regions where the approximation is not yet accurate, additional data is collected and employed in a 'standard' ESC update step, while also using it to refine the approximation of the performance function. By using the approximation of the performance function to determine the search direction and optimizer gain when possible, the number of required performance measurements and parameter update steps can be significantly reduced, e.g., with respectively 75% and 45% in our simulation study involving a static cost function.**

I. Introduction

Extremum-seeking control (ESC) is a data-based control technique aimed at optimizing a system's steady-state performance based on the assumption that a unique, time-invariant mapping between constant inputs and the steady-state system performance output exists. Typically, information about the gradient of this steady-state performance function is obtained by adding small perturbations (dither) to the system input and measuring the corresponding output to estimate the gradient of the performance function at a point of interest. This gradient information is used in gradient-based optimization schemes to steer the system input towards its performance-optimizing value. In general, ESC approaches are either of the continuous-time type [1]–[4] or the sampled-data type [5]–[8]. Here, we will focus on schemes of the latter type; in particular on standard gradient-based schemes as in [5].

While such gradient-based schemes have been used successfully in several applications, they also have a number of drawbacks. Firstly, convergence to the optimum is typically slow since 1) these approaches require a sufficiently long waiting time in between input perturbations and output measurements to guarantee that the system output is sufficiently close to its steady-state value, and 2) the optimizer gain of the gradient-based optimization scheme (sometimes also referred to as 'step size') should be chosen small to guarantee convergence to a close neighborhood of the optimum [8]. Secondly, due to the persistent use of dither to obtain gradient information of the performance function, one typically only achieves *practical* stability of the optimum, as opposed to more desirable *asymptotic* stability. Finally, evaluating the system output for many different constant inputs might be costly in practical applications, e.g., due to suboptimal inputs leading to an increase in the number of scrapped products.

To improve the speed of convergence to the optimum, observers have previously been employed in a continuous-time ESC setting to obtain improved gradient estimates via local approximations of the performance function [9]–[11]. Furthermore, in [12], local approximations were obtained using first-order least-squares fitting to estimate the gradient without the need for dither, resulting in faster convergence and asymptotic stability. However, as mentioned in [13], a drawback of these approaches is that they do not allow the use of available, previously collected datasets. Therefore, in [13], a non-local approximation of the performance function is constructed on the basis of both available datasets and online measurements. By constructing this non-local approximation and using its gradient in a gradient-based optimization scheme, the approach allows the use of previously collected datasets. Furthermore, in contrast to the local approximations that are not retained, the non-local approximation can be reused in case the optimizer returns to a region it has visited before. However, a drawback of the continuous-time approach in [13] is that it requires continuously perturbing the system and measuring the output.

In contrast to these continuous-time approaches, the goal of this work is therefore to improve the rate of convergence to the optimum and to reduce the number of different inputs that need to be applied to the system in a sampled-data setting. This goal is pursued by combining gradient-based sampled-data ESC with kernel-based function approximation. The three main contributions of this paper are as follows. Firstly, as a key stepping stone, we derive bounds on the gradient approximation error obtained by kernel-based approximation of the performance map under sufficient regularity conditions. These error bounds are inspired by the bounds on the function approximation error in [14]. Secondly, as a first step towards the general setting of systems with dynamics, we present a novel ESC scheme for optimizing static cost functions that extends sampled-data ESC as in [5] with kernel-based function approximation, in order to make more efficient use of previous output measurements. The proposed kernel-based ESC (KB-ESC) approach uses output measurements that have previously been collected, or

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

[1]Wouter Weekers, Alessandro Saccon, and Nathan van de Wouw are with the Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, {w.weekers, a.saccon, n.v.d.wouw}@tue.nl.

which are collected online by applying dither, to create online an approximation of the unknown performance function. In regions where this approximation is a sufficiently accurate description of the performance function, KB-ESC uses it to determine a suitable search direction for a gradient-based optimization step, instead of applying dither to estimate the gradient of the performance function. Moreover, it uses the approximation to determine a suitable optimizer gain, instead of using a (typically small) fixed optimizer gain. In contrast to the approach employed in [13], the proposed sampled-data approach only requires dither and measurements in regions where the approximation of the performance function is not sufficiently accurate. Measuring only in regions where the approximation is not sufficiently accurate has the benefit of reducing the number of possibly costly function evaluations (i.e., steady-state performance measurements) needed to reach the optimum, and leads to faster convergence. Furthermore, like the approach in [12], the KB-ESC approach enables asymptotic stability of the optimum, since dither is not required for gradient estimation once the approximation is sufficiently accurate around the optimum. Thirdly, we show the advantages of the proposed approach compared standard ESC on a static cost function.

The rest of this paper is organized as follows. In Section II, we present the problem formulation. In Section III, bounds on the gradient error in kernel-based function approximation are presented, and the proposed KB-ESC approach is introduced. A performance comparison of KB-ESC and sampled-data ESC as in [5] on a static cost function is given in Section IV. Finally, conclusions are presented in Section V.

II. Problem Formulation

Consider the general nonlinear system

$$\dot{\mathbf{x}}(t) = \mathbf{p}(\mathbf{x}(t), \boldsymbol{\theta}(t)), \tag{1a}$$

$$y(t) = q(\mathbf{x}(t), \boldsymbol{\theta}(t)), \tag{1b}$$

where $\mathbf{x} \in \mathbb{R}^{n_\mathbf{x}}$ denotes its state, $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^{n_{\boldsymbol{\theta}}}$ denotes its input, $y \in \mathbb{R}$ denotes its output, and $\mathbf{p} : \mathbb{R}^{n_\mathbf{x}} \times \boldsymbol{\Theta} \to \mathbb{R}^{n_\mathbf{x}}$ and $q : \mathbb{R}^{n_\mathbf{x}} \times \boldsymbol{\Theta} \to \mathbb{R}$ denote the state and output functions, respectively. Given system (1), the following assumption is typically adopted in extremum-seeking control [1].

Assumption 1: For every constant input $\bar{\boldsymbol{\theta}}$, (1) has a unique steady-state solution $\bar{\mathbf{x}} = \mathbf{l}(\bar{\boldsymbol{\theta}})$. That is, $\mathbf{p}(\mathbf{x}, \bar{\boldsymbol{\theta}}) = \mathbf{0}$ if and only if $\mathbf{x} = \bar{\mathbf{x}}$. Moreover, this steady-state solution is assumed to be globally asymptotically stable.

Under Assumption 1, there exists a unique steady-state map $f(\boldsymbol{\theta}) := \lim_{t\to\infty} q(\mathbf{x}(t), \boldsymbol{\theta}) = q(\mathbf{l}(\boldsymbol{\theta}), \boldsymbol{\theta})$ for every constant input $\boldsymbol{\theta}$. We adopt the following assumption on this map f.

Assumption 2: The steady-state map f has a minimizer $\boldsymbol{\theta}^*$, such that $f(\boldsymbol{\theta}^*) \leq f(\boldsymbol{\theta})\ \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$.

The goal in extremum-seeking control is to achieve the minimum steady-state output $y^* = f(\boldsymbol{\theta}^*)$ by steering the input $\boldsymbol{\theta}$ to the minimizer $\boldsymbol{\theta}^*$. To achieve this in a sampled-data context, let us define the zero-order hold (ZOH) operation

$$\boldsymbol{\theta}(t) := \boldsymbol{\theta}_i,\ \forall t \in [(i-1)T,\ iT) \tag{2}$$

with $i \in \{1, 2, \dots\}$ and waiting time $T > 0$, and the ideal T-periodic sampling operation

$$y_i := y(iT),\ \forall i \in \{1, 2, \dots\}. \tag{3}$$

Using these ZOH and sampling operations, a sampled-data extremum-seeking controller as considered in [5] collects $n_\mathbf{h}$ output measurements $y_i = q(\mathbf{x}(iT), \widehat{\boldsymbol{\theta}}_k + \mathbf{h}_{j(i)}) = f(\widehat{\boldsymbol{\theta}}_k + \mathbf{h}_{j(i)}) + \epsilon_i$, $i \in \{n_\mathbf{h}k + 1, \dots, n_\mathbf{h}(k+1)\}$, at each update step k. Herein, $\widehat{\boldsymbol{\theta}}_k$ is the current estimate of $\boldsymbol{\theta}^*$, $\mathbf{h}_{j(i)}$ is a perturbation vector called a dither signal, $j(i) := (i - 1 \mod n_\mathbf{h}) + 1$ is the dither index, i is the iteration index, $k := \lfloor (i-1)/n_\mathbf{h} \rfloor$ is the update index with $\lfloor \cdot \rfloor$ the floor function, and ϵ_i is a perturbation term resulting from the system not fully being in steady-state at the moment the output is measured. The output measurements are used to update $\widehat{\boldsymbol{\theta}}$ in an update step taking the following form:

$$\widehat{\boldsymbol{\theta}}_{k+1} \in \Sigma\left(\widehat{\boldsymbol{\theta}}_k, G(\widehat{\boldsymbol{\theta}}_k)\right). \tag{4}$$

Here, Σ is a (potentially set-valued) map and G is a function containing information about the gradient of f at $\widehat{\boldsymbol{\theta}}_k$, which is obtained by the output measurements y_i. An example of an optimization scheme taking the form of (4) is the central-difference-based update step

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k - \frac{\mu}{2h}\begin{bmatrix} y_{2n_{\boldsymbol{\theta}}k+2} - y_{2n_{\boldsymbol{\theta}}k+1} \\ \vdots \\ y_{2n_{\boldsymbol{\theta}}(k+1)} - y_{2n_{\boldsymbol{\theta}}(k+1)-1} \end{bmatrix} \tag{5}$$

with $y_{2n_{\boldsymbol{\theta}}k+l} = f(\widehat{\boldsymbol{\theta}}_k + \mathbf{h}_l) + \epsilon_{2n_{\boldsymbol{\theta}}k+l}$, $l \in \{1, 2, \dots, 2n_{\boldsymbol{\theta}}\}$, $\mathbf{h}_l = (-1)^l h \mathbf{e}_{\lceil l/2 \rceil}$ with $\lceil \cdot \rceil$ the ceiling function, h the dither magnitude, $\mathbf{e}_{\lceil l/2 \rceil}$ the $\lceil l/2 \rceil$-th unit vector in $\mathbb{R}^{n_{\boldsymbol{\theta}}}$, and $\mu > 0$ the optimizer gain. Note that for (5), $n_\mathbf{h} = 2n_{\boldsymbol{\theta}}$ evaluations of f are required for each update step, since $l \in \{1, \dots, 2n_{\boldsymbol{\theta}}\}$, which might be large in case $n_{\boldsymbol{\theta}}$ is large.

By requiring $n_\mathbf{h}$ output measurements for each update step, and discarding the collected output measurements afterwards instead of reusing them to build up knowledge on the performance map f, optimization schemes of the form (4) make inefficient use of collected data y_i. As mentioned in the introduction, it is desirable to make more efficient use of collected data to keep the number of required output measurements to a minimum. Additionally, update steps of the form (5) typically require the constant optimizer gain μ to be chosen sufficiently small to guarantee convergence of $\widehat{\boldsymbol{\theta}}$ to a (small) neighborhood of $\boldsymbol{\theta}^*$, which generally means that convergence is slow. To address these issues, we extend sampled-data ESC with kernel-based function approximation in the next section. As a key stepping stone, we consider systems without dynamics for simplicity. That is, we restrict our attention to the case where direct measurements of the performance function f are available (i.e., $\epsilon_i = 0$ for all $i = 1, 2, \dots$). We project that the extension to the dynamic case ($\epsilon_i \neq 0$) can be done in the same way as for other sampled-data ESC schemes.

III. ESC Using Function Approximations

In this section, we present our approach to make more efficient use of output measurements in ESC. To this end, we first present preliminaries of kernel-based function approximation. Next, we derive bounds on the gradient approximation error when using kernel-based function approximation to approximate the performance function f. Finally, we present a novel ESC technique that combines sampled-data ESC with kernel-based function approximation.

A. Preliminaries of kernel-based function approximation

Suppose a dataset $\mathcal{D} = \{(\boldsymbol{\theta}_i, y_i) \,|\, i = 1, \dots, N\}$ is available, where $\Omega = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N\}$ are N pairwise distinct inputs for which the system output has been measured, and $\begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^\intercal = \begin{bmatrix} f(\boldsymbol{\theta}_1) & \dots & f(\boldsymbol{\theta}_N) \end{bmatrix}^\intercal =: \mathbf{y}$ are the corresponding measured outputs. Our goal is to construct an approximation m of the performance function f based on the data $\mathcal{D}$. To this end, we will use kernel-based function approximation, where a kernel is considered to be a continuous, symmetric function $\kappa : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$ that is positive definite according to the following definition.

Definition 1: A function $\kappa : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$ is called positive definite if for any set of pairwise-distinct inputs $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N\}$ the $N \times N$ Gram matrix $\mathbf{K}$, whose elements are given by $\mathbf{K}_{i,j} = \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, is positive definite.

We assume κ is twice continuously differentiable and write

$$\nabla \mathbf{k}_\Omega(\cdot) := \begin{bmatrix} D^{(\mathbf{e}_1, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_1) & \dots & D^{(\mathbf{e}_1, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_N) \\ \vdots & \ddots & \vdots \\ D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}}, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_1) & \dots & D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}}, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_N) \end{bmatrix}$$

and

$$\nabla^2 \kappa(\cdot, \cdot) := \begin{bmatrix} D^{(\mathbf{e}_1, \mathbf{e}_1)}\kappa(\cdot, \cdot) & \dots & D^{(\mathbf{e}_1, \mathbf{e}_{n_{\boldsymbol{\theta}}})}\kappa(\cdot, \cdot) \\ \vdots & \ddots & \vdots \\ D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}}, \mathbf{e}_1)}\kappa(\cdot, \cdot) & \dots & D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}}, \mathbf{e}_{n_{\boldsymbol{\theta}}})}\kappa(\cdot, \cdot) \end{bmatrix},$$

where $\mathbf{0}$ denotes the zero vector in $\mathbb{R}^{n_{\boldsymbol{\theta}}}$, and

$$D^{(\mathbf{a}, \mathbf{b})}\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}') := \frac{\partial^{\, a_1 + \dots + a_{n_{\boldsymbol{\theta}}} + b_1 + \dots + b_{n_{\boldsymbol{\theta}}}}}{\partial \theta_1^{a_1} \, \dots \, \partial \theta_{n_{\boldsymbol{\theta}}}^{a_{n_{\boldsymbol{\theta}}}} \, \partial (\theta_1')^{b_1} \, \dots \, \partial (\theta_{n_{\boldsymbol{\theta}}}')^{b_{n_{\boldsymbol{\theta}}}}} \kappa(\boldsymbol{\theta}, \boldsymbol{\theta}').$$

By the Moore-Aronszajn theorem [15], any kernel κ satisfying Definition 1 has a uniquely determined reproducing kernel Hilbert space (RKHS) associated with it. This RKHS, which we denote by $\mathcal{H}$, is defined to be the completion of the class of functions of the form $g(\cdot) = \sum_i \alpha_i \kappa(\cdot, \boldsymbol{\theta}_i)$. It is equipped with the inner product $\langle g_1, g_2 \rangle_{\mathcal{H}} = \left\langle \sum_i \alpha_i \kappa(\cdot, \boldsymbol{\theta}_i), \sum_j \beta_j \kappa(\cdot, \boldsymbol{\theta}_j) \right\rangle_{\mathcal{H}} = \sum_i \sum_j \alpha_i \beta_j \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ and norm $\|g\|_{\mathcal{H}} := \sqrt{\langle g, g \rangle_{\mathcal{H}}}$.

Given an RKHS $\mathcal{H}$, we approximate f by searching an $\bar{m} \in \mathcal{H}$ such that the evaluations $\bar{m}(\boldsymbol{\theta}_i)$ match y_i for all data pairs $(\boldsymbol{\theta}_i, y_i)$ in $\mathcal{D}$, i.e., by solving

$$m = \operatorname*{argmin}_{\bar{m} \in \mathcal{H}} \|\bar{m}\|_{\mathcal{H}}^2 \tag{6a}$$

$$\text{s.t.}\, \bar{m}(\boldsymbol{\theta}_i) = y_i, \forall i = 1, \dots, N. \tag{6b}$$

By the representer theorem [16], (6) has a solution of the form $m(\cdot) = \sum_{i=1}^N \alpha_i \kappa(\cdot, \boldsymbol{\theta}_i)$. Enforcing the constraints (6b) thus results in the following set of linear equations

$$\mathbf{y} = \begin{bmatrix} \kappa(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1) & \dots & \kappa(\boldsymbol{\theta}_1, \boldsymbol{\theta}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{\theta}_N, \boldsymbol{\theta}_1) & \dots & \kappa(\boldsymbol{\theta}_N, \boldsymbol{\theta}_N) \end{bmatrix} \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{7}$$

where $\boldsymbol{\alpha} := \begin{bmatrix} \alpha_1 & \dots & \alpha_N \end{bmatrix}^\intercal$ is a vector of weights. By Definition 1, the symmetric matrix $\mathbf{K}$ is positive definite and hence invertible. The weights $\boldsymbol{\alpha}$ are thus obtained from (7) as $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}$, resulting in f being approximated by

$$m(\cdot) = \sum_{i=1}^N \alpha_i \kappa(\cdot, \boldsymbol{\theta}_i) = \mathbf{k}_\Omega(\cdot)\mathbf{K}^{-1}\mathbf{y} \tag{8}$$

with $\mathbf{k}_\Omega(\cdot) := \begin{bmatrix} \kappa(\cdot, \boldsymbol{\theta}_1) & \dots & \kappa(\cdot, \boldsymbol{\theta}_N) \end{bmatrix}$.

B. Novel error bound on the gradient approximation error

Since $\mathbf{K}$ and $\mathbf{y}$, and thus $\boldsymbol{\alpha}$, are constant given the dataset $\mathcal{D}$, and κ is assumed to be twice continuously differentiable, the gradient of m is given by

$$\nabla m(\cdot) = \sum_{i=1}^N \alpha_i \begin{bmatrix} D^{(\mathbf{e}_1, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_i) \\ \vdots \\ D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}}, \mathbf{0})}\kappa(\cdot, \boldsymbol{\theta}_i) \end{bmatrix} = \nabla \mathbf{k}_\Omega(\cdot)\mathbf{K}^{-1}\mathbf{y} \tag{9}$$

with $\nabla \mathbf{k}_\Omega(\cdot)$ as defined in Section III-A. To use m in (gradient-based) optimization schemes aimed at minimizing f, we are interested in quantifying the function approximation error $|f(\boldsymbol{\theta}) - m(\boldsymbol{\theta})|$ and the gradient approximation error $\|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\|$. Hereto, we note that error bounds on the function approximation error have been obtained in [14] under the following assumption.

Assumption 3 ([14, Assumption 3]): Given a kernel κ, we assume that f belongs to its corresponding RKHS $\mathcal{H}$, and that an upper bound on the norm $\|f\|_{\mathcal{H}} \leq \Gamma$ is known.

Under Assumption 3, the function approximation error for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ can be bounded by [14, Proposition 1]

$$|f(\boldsymbol{\theta}) - m(\boldsymbol{\theta})| \leq P_\Omega(\boldsymbol{\theta}) \sqrt{\Gamma^2 - \|m\|_{\mathcal{H}}^2}, \tag{10}$$

where $P_\Omega(\boldsymbol{\theta}) := \sqrt{\kappa(\boldsymbol{\theta}, \boldsymbol{\theta}) - \mathbf{k}_\Omega(\boldsymbol{\theta})\mathbf{K}^{-1}\left(\mathbf{k}_\Omega(\boldsymbol{\theta})\right)^\intercal}$ and

$$\|m\|_{\mathcal{H}}^2 = \left\langle \sum_{i=1}^N \alpha_i \kappa(\cdot, \boldsymbol{\theta}_i), \sum_{j=1}^N \alpha_j \kappa(\cdot, \boldsymbol{\theta}_j) \right\rangle_{\mathcal{H}} \tag{11}$$

$$= \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \kappa(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \boldsymbol{\alpha}^\intercal \mathbf{K} \boldsymbol{\alpha}. \tag{12}$$

Next, we will show that bounds of similar form can also be obtained for the gradient approximation error $\|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\|$, as given by the following theorem.

Theorem 1: Under Assumption 3, given an approximation m obtained by solving (6) on the basis of a dataset $\mathcal{D} = \{(\boldsymbol{\theta}_i, y_i) \,|\, i = 1, \dots, N\}$ with pairwise distinct inputs $\Omega = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N\}$ and corresponding function values $y_i = f(\boldsymbol{\theta}_i)$, $\|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\|$ for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ can be bounded as

$$\|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\| \leq \bar{\lambda}\left(\widetilde{P}_\Omega(\boldsymbol{\theta})\right) \sqrt{\Gamma^2 - \|m\|_{\mathcal{H}}^2}, \tag{13}$$

where $\widetilde{P}_\Omega(\boldsymbol{\theta}) := \left(\nabla^2\kappa(\boldsymbol{\theta},\boldsymbol{\theta}) - \nabla\mathbf{k}_\Omega(\boldsymbol{\theta})\mathbf{K}^{-1}\left(\nabla\mathbf{k}_\Omega(\boldsymbol{\theta})\right)^\intercal\right)^{\frac{1}{2}}$ and $\bar{\lambda}(\widetilde{P}_\Omega(\boldsymbol{\theta}))$ is its maximum eigenvalue.

Proof: The proof follows steps along the lines of the proof of [14, Proposition 1]. Let $\widetilde{\mathcal{D}} = \mathcal{D} \cup (\widetilde{\boldsymbol{\theta}}, \nabla f(\widetilde{\boldsymbol{\theta}}))$ be the dataset $\mathcal{D}$ with a single additional virtual gradient observation pair $(\widetilde{\boldsymbol{\theta}}, \nabla f(\widetilde{\boldsymbol{\theta}}))$. An approximation $\widetilde{m}$ of f interpolating all data pairs in $\widetilde{\mathcal{D}}$ can be obtained by solving (cf. (6))

$$\widetilde{m} = \operatorname*{argmin}_{\bar{m}\in\mathcal{H}} \|\bar{m}\|_{\mathcal{H}}^2 \tag{14a}$$
$$\text{s.t. } \bar{m}(\boldsymbol{\theta}_i) = y_i, \forall i = 1,\dots,N, \tag{14b}$$
$$\nabla\bar{m}(\widetilde{\boldsymbol{\theta}}) = \nabla f(\widetilde{\boldsymbol{\theta}}). \tag{14c}$$

By [17, Theorem 2], (14) has a solution of the form

$$\widetilde{m}(\cdot) = \sum_{i=1}^{N}\alpha_i\kappa(\cdot,\boldsymbol{\theta}_i) + \sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_{N+j}D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}}) \tag{15}$$
$$= \begin{bmatrix}\mathbf{k}_\Omega(\cdot) & \left(\nabla\mathbf{k}_{\widetilde{\boldsymbol{\theta}}}(\cdot)\right)^\intercal\end{bmatrix}\widetilde{\boldsymbol{\alpha}} \tag{16}$$

with $\nabla\mathbf{k}_{\widetilde{\boldsymbol{\theta}}}(\cdot) := \begin{bmatrix}D^{(\mathbf{0},\mathbf{e}_1)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}}) & \dots & D^{(\mathbf{0},\mathbf{e}_{n_{\boldsymbol{\theta}}})}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\end{bmatrix}^\intercal$, $\widetilde{\boldsymbol{\alpha}} := \begin{bmatrix}\alpha_1 & \dots & \alpha_{N+n_{\boldsymbol{\theta}}}\end{bmatrix}^\intercal$, and $\mathbf{k}_\Omega(\cdot)$ as defined in Section III-A. Similar to before, the gradient of $\widetilde{m}$ is given by

$$\nabla\widetilde{m}(\cdot) = \sum_{i=1}^{N}\alpha_i\begin{bmatrix}D^{(\mathbf{e}_1,\mathbf{0})}\kappa(\cdot,\boldsymbol{\theta}_i)\\ \vdots \\ D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}},\mathbf{0})}\kappa(\cdot,\boldsymbol{\theta}_i)\end{bmatrix} + \sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_{N+j}\begin{bmatrix}D^{(\mathbf{e}_1,\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\\ \vdots \\ D^{(\mathbf{e}_{n_{\boldsymbol{\theta}}},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\end{bmatrix} \tag{17}$$
$$= \begin{bmatrix}\nabla\mathbf{k}_\Omega(\cdot) & \nabla^2\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\end{bmatrix}\widetilde{\boldsymbol{\alpha}}. \tag{18}$$

Enforcing the constraints (14b) and (14c) in (16) and (18), respectively, and defining $\widetilde{\mathbf{y}} := \begin{bmatrix}\mathbf{y}^\intercal & \left(\nabla f(\widetilde{\boldsymbol{\theta}})\right)^\intercal\end{bmatrix}^\intercal$, we obtain the following set of linear equations

$$\widetilde{\mathbf{y}} = \begin{bmatrix}\widetilde{m}(\boldsymbol{\theta}_1)\\ \vdots \\ \widetilde{m}(\boldsymbol{\theta}_N)\\ \nabla\widetilde{m}(\widetilde{\boldsymbol{\theta}})\end{bmatrix} = \begin{bmatrix}\mathbf{k}_\Omega(\boldsymbol{\theta}_1) & \left(\nabla\mathbf{k}_{\widetilde{\boldsymbol{\theta}}}(\boldsymbol{\theta}_1)\right)^\intercal\\ \vdots & \vdots \\ \mathbf{k}_\Omega(\boldsymbol{\theta}_N) & \left(\nabla\mathbf{k}_{\widetilde{\boldsymbol{\theta}}}(\boldsymbol{\theta}_N)\right)^\intercal\\ \nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}}) & \nabla^2\kappa(\widetilde{\boldsymbol{\theta}},\widetilde{\boldsymbol{\theta}})\end{bmatrix}\widetilde{\boldsymbol{\alpha}} \tag{19}$$
$$= \begin{bmatrix}\mathbf{K} & \left(\nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}})\right)^\intercal\\ \nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}}) & \nabla^2\kappa(\widetilde{\boldsymbol{\theta}},\widetilde{\boldsymbol{\theta}})\end{bmatrix}\widetilde{\boldsymbol{\alpha}} =: \widetilde{\mathbf{K}}\widetilde{\boldsymbol{\alpha}}. \tag{20}$$

It follows from (16) and reproducing properties [17, Theorem 1] $D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\boldsymbol{\theta}_i,\widetilde{\boldsymbol{\theta}}) = \langle\kappa(\cdot,\boldsymbol{\theta}_i), D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\rangle_{\mathcal{H}}$, $D^{(\mathbf{e}_j,\mathbf{0})}\kappa(\widetilde{\boldsymbol{\theta}},\boldsymbol{\theta}_i) = \langle D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}}), \kappa(\cdot,\boldsymbol{\theta}_i)\rangle_{\mathcal{H}}$ and $D^{(\mathbf{e}_i,\mathbf{e}_j)}\kappa(\widetilde{\boldsymbol{\theta}},\widetilde{\boldsymbol{\theta}}) = \langle D^{(\mathbf{0},\mathbf{e}_i)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}}), D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\rangle_{\mathcal{H}}$ that

$$\|\widetilde{m}\|_{\mathcal{H}}^2 = \Bigg\langle\sum_{i=1}^{N}\alpha_i\kappa(\cdot,\boldsymbol{\theta}_i) + \sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_{N+j}D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}}),\dots \sum_{i=1}^{N}\alpha_i\kappa(\cdot,\boldsymbol{\theta}_i) + \sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_{N+j}D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\cdot,\widetilde{\boldsymbol{\theta}})\Bigg\rangle \tag{21}$$
$$= \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\kappa(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j) + \sum_{i=1}^{N}\sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_i\alpha_{N+j}D^{(\mathbf{0},\mathbf{e}_j)}\kappa(\boldsymbol{\theta}_i,\widetilde{\boldsymbol{\theta}}) + \sum_{i=1}^{N}\sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_i\alpha_{N+j}D^{(\mathbf{e}_j,\mathbf{0})}\kappa(\widetilde{\boldsymbol{\theta}},\boldsymbol{\theta}_i) + \sum_{i=1}^{n_{\boldsymbol{\theta}}}\sum_{j=1}^{n_{\boldsymbol{\theta}}}\alpha_{N+i}\alpha_{N+j}D^{(\mathbf{e}_i,\mathbf{e}_j)}\kappa(\widetilde{\boldsymbol{\theta}},\widetilde{\boldsymbol{\theta}}) \tag{22}$$
$$= \widetilde{\boldsymbol{\alpha}}^\intercal\widetilde{\mathbf{K}}\widetilde{\boldsymbol{\alpha}}. \tag{23}$$

Assuming $\widetilde{\mathbf{K}}$ is invertible, we use (20) and block matrix inversion to write (23) as

$$\|\widetilde{m}\|_{\mathcal{H}}^2 = \widetilde{\mathbf{y}}^\intercal\widetilde{\mathbf{K}}^{-1}\widetilde{\mathbf{y}} \tag{24}$$
$$= \widetilde{\mathbf{y}}^\intercal\begin{bmatrix}\mathbf{K}^{-1} + \mathbf{X}\widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\mathbf{X}^\intercal & -\mathbf{X}\widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\\ -\widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\mathbf{X}^\intercal & \widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\end{bmatrix}\widetilde{\mathbf{y}} \tag{25}$$
$$= \mathbf{y}^\intercal\mathbf{K}^{-1}\mathbf{y} + \widetilde{\mathbf{y}}^\intercal\begin{bmatrix}\mathbf{X}\\ -\mathbf{I}\end{bmatrix}\widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\begin{bmatrix}\mathbf{X}\\ -\mathbf{I}\end{bmatrix}^\intercal\widetilde{\mathbf{y}} \tag{26}$$
$$= \mathbf{y}^\intercal\mathbf{K}^{-1}\mathbf{y} + \left\|\widetilde{P}_\Omega^{-1}(\widetilde{\boldsymbol{\theta}})\left(\mathbf{X}^\intercal\mathbf{y} - \nabla f(\widetilde{\boldsymbol{\theta}})\right)\right\|^2, \tag{27}$$

where $\mathbf{X} := \mathbf{K}^{-1}(\nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}}))^\intercal$, and $\widetilde{P}_\Omega^2(\widetilde{\boldsymbol{\theta}}) := \nabla^2\kappa(\widetilde{\boldsymbol{\theta}},\widetilde{\boldsymbol{\theta}}) - \nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}})\mathbf{K}^{-1}(\nabla\mathbf{k}_\Omega(\widetilde{\boldsymbol{\theta}}))^\intercal$ is the (symmetric) Schur complement of $\mathbf{K}$ in $\widetilde{\mathbf{K}}$. Using (7), (9), (12) and (27), we obtain

$$\|\widetilde{m}\|_{\mathcal{H}}^2 = \|m\|_{\mathcal{H}}^2 + \left\|\widetilde{P}_\Omega^{-1}(\widetilde{\boldsymbol{\theta}})\left(\nabla m(\widetilde{\boldsymbol{\theta}}) - \nabla f(\widetilde{\boldsymbol{\theta}})\right)\right\|^2. \tag{28}$$

Denoting the minimum and maximum eigenvalue of a matrix $\mathbf{A}$ by $\underline{\lambda}(\mathbf{A})$ and $\bar{\lambda}(\mathbf{A})$, respectively, we note that $\underline{\lambda}\left(\mathbf{A}^{-1}\right) = \bar{\lambda}^{-1}(\mathbf{A})$ and $\bar{\lambda}\left(\mathbf{A}^2\right) = \bar{\lambda}^2(\mathbf{A})$. Furthermore, we note that since $\widetilde{P}_\Omega(\widetilde{\boldsymbol{\theta}})$ is symmetric, it holds that

$$\underline{\lambda}\left(\widetilde{P}_\Omega^{-2}(\widetilde{\boldsymbol{\theta}})\right)\left\|\nabla m(\widetilde{\boldsymbol{\theta}}) - \nabla f(\widetilde{\boldsymbol{\theta}})\right\|^2 \leq \left\|\widetilde{P}_\Omega^{-1}(\widetilde{\boldsymbol{\theta}})\left(\nabla m(\widetilde{\boldsymbol{\theta}}) - \nabla f(\widetilde{\boldsymbol{\theta}})\right)\right\|^2. \tag{29}$$

With these observations we obtain the following inequality from (28)

$$\|m\|_{\mathcal{H}}^2 + \bar{\lambda}^{-2}\left(\widetilde{P}_\Omega(\widetilde{\boldsymbol{\theta}})\right)\left\|\nabla f(\widetilde{\boldsymbol{\theta}}) - \nabla m(\widetilde{\boldsymbol{\theta}})\right\|^2 \leq \|\widetilde{m}\|_{\mathcal{H}}^2. \tag{30}$$

Finally, note that $\|\widetilde{m}\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} \leq \Gamma$, where the first inequality follows from the fact that $f \in \mathcal{H}$ and $\widetilde{m}$ is the function in $\mathcal{H}$ with the smallest norm that interpolates all data pairs in $\widetilde{\mathcal{D}}$, while the second follows from Assumption 3. Consequently, alternatively to (30), we can write

$$\|m\|_{\mathcal{H}}^2 + \bar{\lambda}^{-2}\left(\widetilde{P}_\Omega(\widetilde{\boldsymbol{\theta}})\right)\left\|\nabla f(\widetilde{\boldsymbol{\theta}}) - \nabla m(\widetilde{\boldsymbol{\theta}})\right\|^2 \leq \Gamma^2, \tag{31}$$

from which (13) follows by rewriting the inequality and noting that $\widetilde{\boldsymbol{\theta}}$ could be any input $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. ∎

C. Kernel-Based Extremum-Seeking Control

As mentioned in Section II, in ESC typically $n_{\mathbf{h}}$ output measurements are made to perform update steps (4) aimed at finding the minimizer $\boldsymbol{\theta}^*$. Instead of discarding these output measurements after each update step, as is commonly done in

sampled-data ESC, we collect those measurements and their corresponding inputs in a dataset $\mathcal{D}$ as defined in Section III-A. Each time a new data pair $(\boldsymbol{\theta}_i, y_i)$ is added to $\mathcal{D}$, we can obtain a new approximation m of the performance function f using the kernel-based function approximation technique described in Section III-A. If m is a sufficiently accurate approximation of f around $\widehat{\boldsymbol{\theta}}_k$ (in a sense that will be defined shortly below), we use update steps

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k - \hat{\mu}\nabla m(\widehat{\boldsymbol{\theta}}_k), \tag{32}$$

where $\hat{\mu} > 0$ is a suitable optimizer gain, instead of the classical update steps in the form (4). We say that m is a sufficiently accurate approximation of f around $\widehat{\boldsymbol{\theta}}_k$ if there exists a $\hat{\mu}$ that satisfies the Armijo condition [18, Chapter 3]

$$f(\widehat{\boldsymbol{\theta}}_{k+1}) \leq f(\widehat{\boldsymbol{\theta}}_k) - c\hat{\mu}\left(\nabla f(\widehat{\boldsymbol{\theta}}_k)\right)^{\intercal}\nabla m(\widehat{\boldsymbol{\theta}}_k). \tag{33}$$

Here, the control parameter $c \in (0, 1)$ is a given constant that is typically chosen small, and $\widehat{\boldsymbol{\theta}}_{k+1}$ is defined as in (32). Furthermore, we call any $\hat{\mu}$ for which (33) is satisfied a *suitable optimizer gain*. Using this definition for the accuracy of m, an update step (32) with a suitable optimizer gain $\hat{\mu}$ guarantees a sufficient decrease in f to aid convergence if $-\nabla m$ is a descent direction [18, Chapter 3]. The following lemma provides sufficient conditions for using m to check if (33) is satisfied and to check if $-\nabla m$ is a descent direction.

Lemma 1: Under the same conditions as Theorem 1, the Armijo condition (33) is guaranteed to be satisfied if

$$\begin{aligned} m(\widehat{\boldsymbol{\theta}}_k) - \delta_1(\widehat{\boldsymbol{\theta}}_k) - c\hat{\mu}(\|\nabla m(\widehat{\boldsymbol{\theta}}_k)\|^2 + \delta_2(\widehat{\boldsymbol{\theta}}_k)\|\nabla m(\widehat{\boldsymbol{\theta}}_k)\|) \\ \geq m(\widehat{\boldsymbol{\theta}}_k - \hat{\mu}\nabla m(\widehat{\boldsymbol{\theta}}_k)) + \delta_1(\widehat{\boldsymbol{\theta}}_k - \hat{\mu}\nabla m(\widehat{\boldsymbol{\theta}}_k)) \end{aligned} \tag{34}$$

with $\delta_1(\cdot) := P_\Omega(\cdot)\sqrt{\Gamma^2 - \|m\|_{\mathcal{H}}^2}$ the bound on the function approximation error from (10), and $\delta_2(\cdot) := \bar{\lambda}\left(\widetilde{P}_\Omega(\cdot)\right)\sqrt{\Gamma^2 - \|m\|_{\mathcal{H}}^2}$ the bound on the gradient approximation error from (13). Moreover, $-\nabla m(\boldsymbol{\theta})$ is guaranteed to be a descent direction if $\delta_2(\boldsymbol{\theta}) < \|\nabla m(\boldsymbol{\theta})\|$.

Proof: It follows from (10) that f can be bounded as

$$m(\boldsymbol{\theta}) - \delta_1(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta}) \leq m(\boldsymbol{\theta}) + \delta_1(\boldsymbol{\theta}). \tag{35}$$

Furthermore, from the Cauchy-Schwarz inequality and (13)

$$\begin{aligned} &\nabla f(\boldsymbol{\theta})^{\intercal}\nabla m(\boldsymbol{\theta}) \\ &\quad = \|\nabla m(\boldsymbol{\theta})\|^2 + (\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta}))^{\intercal}\nabla m(\boldsymbol{\theta}) \\ &\quad \leq \|\nabla m(\boldsymbol{\theta})\|^2 + \|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\|\,\|\nabla m(\boldsymbol{\theta})\| \\ &\quad \leq \|\nabla m(\boldsymbol{\theta})\|^2 + \delta_2(\boldsymbol{\theta})\,\|\nabla m(\boldsymbol{\theta})\| =: \gamma(\boldsymbol{\theta}). \end{aligned} \tag{36}$$

Combining the left inequality in (35) and (36), it follows that

$$m(\boldsymbol{\theta}) - \delta_1(\boldsymbol{\theta}) - c\hat{\mu}\gamma(\boldsymbol{\theta}) \leq f(\boldsymbol{\theta}) - c\hat{\mu}\nabla f(\boldsymbol{\theta})^{\intercal}\nabla m(\boldsymbol{\theta}). \tag{37}$$

Thus, if (34) is satisfied, then it follows from combining (32), the right inequality in (35), and (37) that

$$\begin{aligned} f(\widehat{\boldsymbol{\theta}}_{k+1}) &\leq m(\widehat{\boldsymbol{\theta}}_k - \hat{\mu}\nabla m(\widehat{\boldsymbol{\theta}}_k)) + \delta_1(\widehat{\boldsymbol{\theta}}_k - \hat{\mu}\nabla m(\widehat{\boldsymbol{\theta}}_k)) \\ &\leq m(\widehat{\boldsymbol{\theta}}_k) - \delta_1(\widehat{\boldsymbol{\theta}}_k) - c\hat{\mu}\gamma(\widehat{\boldsymbol{\theta}}_k) \\ &\leq f(\widehat{\boldsymbol{\theta}}_k) - c\hat{\mu}\nabla f(\widehat{\boldsymbol{\theta}}_k)^{\intercal}\nabla m(\widehat{\boldsymbol{\theta}}_k), \end{aligned} \tag{38}$$

which is the Armijo condition (33), concluding the proof of the first statement. The second statement follows from using similar steps as in (36) to show that $-\nabla f(\boldsymbol{\theta})^{\intercal}\nabla m(\boldsymbol{\theta}) < 0$ for $\nabla f(\boldsymbol{\theta}) \neq \mathbf{0}$ if $\delta_2(\boldsymbol{\theta}) < \|\nabla m(\boldsymbol{\theta})\|$. ■

Since evaluations of m are cheap compared to possibly costly measured evaluations of f, Lemma 1 allows to select a suitable optimizer gain $\hat{\mu}$ to be used in (32) that guarantees that (33) is satisfied solely on the basis of knowledge on the approximation m, its gradient, and their error bounds in (10) and (13), or to decide to use a classical ESC update step as in (4) if such $\hat{\mu}$ cannot be found. To this end, we use a backtracking line search that iteratively reduces $\hat{\mu}$ by a factor $\rho \in (0, 1)$, starting from a given maximum value μ_{max}. The search terminates when either a $\hat{\mu}$ satisfying (34) is found, or $\hat{\mu}$ becomes smaller than a given minimum value μ_{min}. In the latter case, we assume no suitable optimizer gain $\hat{\mu}$ exists and that the approximation m is not sufficiently accurate. In this case, we perform $n_{\mathbf{h}}$ additional output measurements as in standard ESC, and use them to perform an update step of the form (4). Additionally, we add the newly collected data pairs $(\boldsymbol{\theta}_i, y_i)$, consisting of the output measurements and the corresponding inputs, to the dataset $\mathcal{D}$ to improve the approximation m. This approach, which we will refer to as kernel-based ESC (KB-ESC), is summarized in Algorithm 1.

Note that update steps of the form (32) do not require any additional output measurements, as opposed to the $n_{\mathbf{h}}$ output measurements required for update steps of the form (4). Furthermore, the backtracking line search allows computation of a suitable optimizer gain $\hat{\mu}$ at each update step of the form (32), in contrast to requiring a (typically small) constant optimizer gain μ, e.g., in update steps of the form (5). By using update steps of the form (32) whenever m is a sufficiently accurate approximation of f, KB-ESC thus reduces the total number of output measurements by making more efficient use of previously collected data, and possibly allows for faster convergence by allowing computation of a suitable $\hat{\mu}$ at these update steps. Satisfaction of the Armijo condition (33), which by Lemma 1 is guaranteed by satisfaction of (34), ensures a sufficient decrease in the system output upon each update (32) to guarantee $\widehat{\boldsymbol{\theta}}$ converges to (a neighborhood of) the minimizer $\boldsymbol{\theta}^*$ over time.

Remark 1: In the above we considered a static system where the measurements $y_i = f(\boldsymbol{\theta}_i) + \epsilon_i$ give direct evaluations of the performance function f with all $\epsilon_i = 0$. In the dynamic setting, generally only perturbed measurements ($\epsilon_i \neq 0$) are available due to the system not fully being in steady-state when the output is measured after a finite waiting time. However, by Assumption 1, for every $\bar{\epsilon}$ there exists a waiting time $T_{\bar{\epsilon}}$ such that $|\epsilon_i| \leq \bar{\epsilon}, \forall i = 1, 2, \ldots$ (i.e., $|y(t) - f(\boldsymbol{\theta})| \leq \bar{\epsilon}$ for all $t \geq T_{\bar{\epsilon}}$). In case such $\bar{\epsilon}$ corresponding to the chosen waiting time T is known, or in case a waiting time $T_{\bar{\epsilon}}$ is known such that $\bar{\epsilon}$ is guaranteed to be smaller than a desired value, one could use for example ε-support vector regression to obtain an approximation m. Using such approach, steps similar to the proof of [14, Theorem 2] could be used to derive a bound on the gradient approximation error $\|\nabla f(\boldsymbol{\theta}) - \nabla m(\boldsymbol{\theta})\|$ of the

Algorithm 1 Kernel-Based Extremum-Seeking Control

Input: initial input $\widehat{\boldsymbol{\theta}}_0$, line search starting value μ_{max}, line search termination value μ_{min}, reduction factor ρ, dither $\mathbf{h}$, and maximum number of iterations k_{max}.
Output: estimate $\widehat{\boldsymbol{\theta}}$ of the minimizer $\boldsymbol{\theta}^*$
procedure KB-ESC($\widehat{\boldsymbol{\theta}}_0$, μ_{max}, μ_{min}, ρ, $\mathbf{h}$, k_{max})
  $k \leftarrow 0$, $\mathcal{D} \leftarrow \{\}$, $\widehat{\boldsymbol{\theta}} \leftarrow \widehat{\boldsymbol{\theta}}_0$
  Perform $n_{\mathbf{h}}$ output measurements with dither $\mathbf{h}$ as described in Section II.
  Update $\widehat{\boldsymbol{\theta}}$ according to (4).
  Update the dataset $\mathcal{D}$ with the collected data pairs $(\boldsymbol{\theta}_i, y_i)$, $i \in \{n_{\mathbf{h}}k+1, \ldots, n_{\mathbf{h}}(k+1)\}$. Use the updated $\mathcal{D}$ to determine the new m and ∇m according to (8) and (9).
  while $k < k_{max}$ **do**
    $k \leftarrow k+1$, $\hat{\mu} \leftarrow \mu_{max}$
    while $\hat{\mu} \geq \mu_{min}$ **and not** (34) **do**
      $\hat{\mu} \leftarrow \rho\hat{\mu}$
    end while
    if $\hat{\mu} \geq \mu_{min}$ **and** $\delta_2(\widehat{\boldsymbol{\theta}}) < \|\nabla m(\widehat{\boldsymbol{\theta}})\|$ **then**
      Update $\widehat{\boldsymbol{\theta}}$ according to (32).
    else
      Perform the steps described in lines 5-7.
    end if
  end while
end procedure

form (13) plus some additional perturbation induced by the mismatch of the measured performance output with respect to the steady-state performance function. This bound can be used to obtain a condition similar to (34) to check if the Armijo condition is guaranteed to be satisfied. Similar to [14, Theorem 2], the additional perturbation will decrease as $\bar{\epsilon}$ decreases (e.g., by choosing a longer waiting time), and (13) is recovered for $\bar{\epsilon} = 0$. We project that these insights can be used to extend the results in this paper to the scenario of KB-ESC for steady-state performance optimization of dynamical systems, but this will be addressed in a future investigation.

IV. Simulation Study

We compare the performance of KB-ESC to standard sampled-data ESC as considered in [5], by applying both approaches to the static cost function $f(\theta) = -\frac{1}{2}\kappa(\theta,-1) - 2\kappa(\theta,0) - \kappa(\theta,1) + \kappa(\theta,2) + \frac{1}{2}\kappa(\theta,3)$, where $\kappa(\theta,\theta') = \exp(-(\theta-\theta')^2/\sigma^2)$ is a squared exponential kernel with length scale $\sigma = 4$. Starting from the initial guess $\widehat{\theta}_0 = 5$, we use updates of the form (5) with dither amplitude $h = 0.1$ in both approaches when updating $\widehat{\theta}_k$ on the basis of output measurements. Furthermore, for KB-ESC, we use $c = 10^{-4}$ for the Armijo control parameter (cf. (34)), and the backtracking parameters $\rho = 0.9$, $\mu_{max} = 50$, and $\mu_{min} = 0.01$. Finally, we use $\Gamma = 3$ as a 20% higher conservative upper bound on the true value of $\|f\|_{\mathcal{H}} = 2.49$.

The performance of both approaches is compared for a low optimizer gain ($\mu = 0.1$ in (5)), an intermediate one ($\mu = 1$), and a high one ($\mu = 10$). In the low-gain case, changes in the estimate $\widehat{\theta}$ of the minimizer are small for standard ESC, resulting in a slow convergence rate, as shown in Fig. 1. Conversely, for KB-ESC, larger changes in $\widehat{\theta}$ can be observed due to a larger optimizer gain $\hat{\mu}$ being chosen

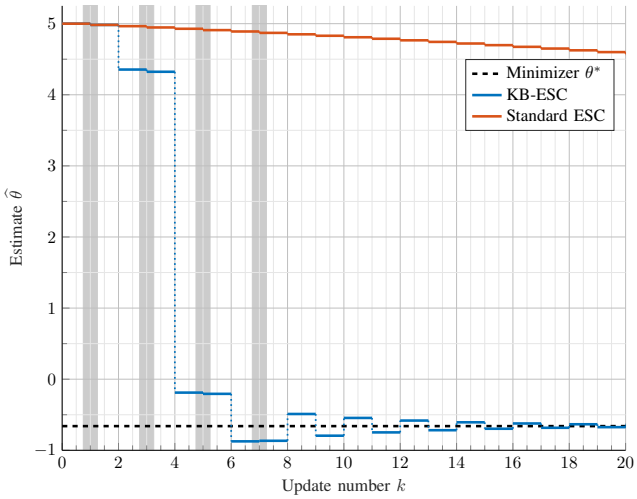

Fig. 1: Estimate $\widehat{\theta}$ at update k for the two ESC approaches in the low-gain case ($\mu = 0.1$). Shaded intervals indicate KB-ESC updates needing output measurements. Convergence to the minimizer θ^* is quicker for KB-ESC and requires only 8 output measurements (needed in 4 update steps).

whenever condition (34) from Lemma 1 can be satisfied. These large changes result in faster convergence of $\widehat{\theta}$ towards the minimizer $\theta^* = -0.66$. Moreover, for KB-ESC, only the 4 update steps indicated by the shaded intervals in Fig. 1 require output measurements. Since $n_{\mathbf{h}} = 2$ for these update steps (cf. (5)), KB-ESC only requires 8 output measurements to converge to the minimizer θ^*, whereas standard ESC used 40 measurements for the 20 updates shown in Fig. 1 alone.

Faster convergence to θ^* is obtained for standard ESC in the intermediate-gain case ($\mu = 1$), as can be seen by comparing Fig. 2 to Fig. 1. However, similar to the small-gain case, KB-ESC converges even faster due to its ability to use larger optimizer gains $\hat{\mu}$ at some update steps. As a result, KB-ESC requires 45% fewer update steps to achieve the same level of accuracy as standard ESC (9 instead of 20). Furthermore, KB-ESC only requires 5 update steps with output measurements, as illustrated by the shaded intervals in Fig. 2. Because of this, only 10 output measurements are made instead of 40, leading to a reduction of 75%.

Finally, in the high-gain case ($\mu = 10$), the optimizer gain is too large for $\widehat{\theta}$ to converge to a small neighborhood of θ^* using standard ESC, as shown in Fig. 3. In contrast, for KB-ESC, $\widehat{\theta}$ converges to θ^* as a result of a smaller optimizer gain $\hat{\mu}$ that guarantees a sufficient decrease in f being computed at updates of the form (32). The convergence is achieved using only 5 update steps in which output measurements are required, as indicated by the shaded intervals in Fig. 3, resulting in only 10 output measurements being made.

V. Conclusion

We presented a novel extremum-seeking control (ESC) approach that combines sampled-data ESC with kernel-based function approximation aimed at making more efficient use of previously collected data. This new approach,

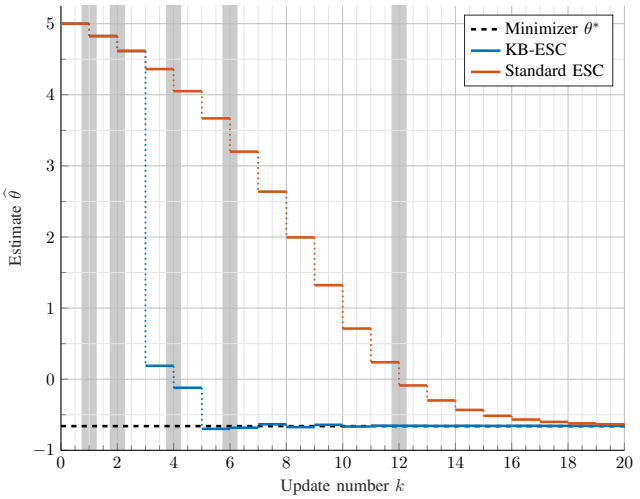

Fig. 2: Estimate $\widehat{\theta}$ at update k for the two ESC approaches in the intermediate-gain case ($\mu = 1$). Shaded intervals indicate KB-ESC updates needing output measurements. Convergence to the minimizer θ^* is quicker for KB-ESC and requires only 10 output measurements (needed in 5 updates).

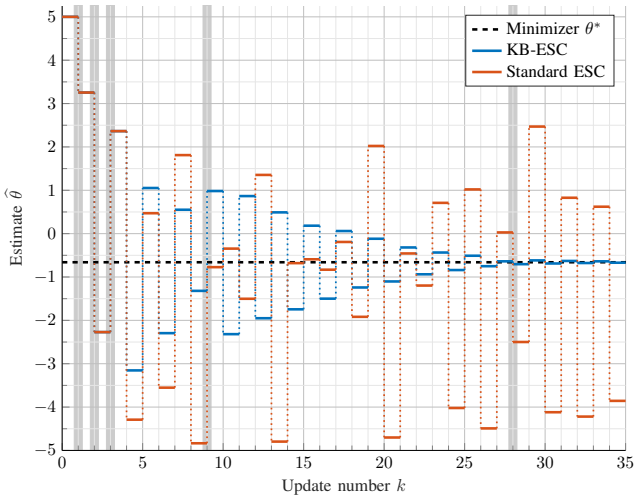

Fig. 3: Estimate $\widehat{\theta}$ at update k for the two ESC approaches in the high-gain case ($\mu = 10$). Shaded intervals indicate KB-ESC updates needing output measurements. KB-ESC achieves convergence to the minimizer θ^* with only 10 output measurements (needed in 5 updates), while standard ESC does not converge.

named kernel-based extremum-seeking control (KB-ESC), uses system output measurements and their corresponding inputs obtained during standard sampled-data ESC steps to construct a kernel-based approximation of the system's steady-state performance function. In regions where the approximation is sufficiently accurate, this approximation is used to determine a search direction and a suitable optimizer gain without performing additional output measurements. By using the approximation of the performance function to determine the search direction, the number of costly output measurements required for optimizing the steady-state performance is reduced. Additionally, the computation of a suitable optimizer gain based on the approximation of the performance function increases the convergence rate compared to standard gradient-based sampled-data ESC. These benefits have been shown in a simulation case study involving a static cost function, in which KB-ESC achieved reductions of 75% and 45% in the number of required output measurements and update steps, respectively, to converge as close to the minimizer as standard sampled-data ESC. Future work will consider extension of KB-ESC to the dynamic setting, and a formal analysis of its convergence properties.

References

[1] M. Krstić and H.-H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.

[2] Y. Tan, D. Nešić, and I. M. Y. Mareels, "On non-local stability properties of extremum seeking control," *Automatica*, vol. 42, pp. 889–903, 2006.

[3] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Y. Mareels, "Extremum seeking from 1922 to 2010," *Proceedings of the 29th Chinese Control Conference*, pp. 14–26, 2010.

[4] L. Hazeleger, M. A. M. Haring, and N. van de Wouw, "Extremum-seeking control for optimization of time-varying steady-state responses of nonlinear systems," *Automatica*, vol. 119, p. 109068, 2020.

[5] A. R. Teel and D. Popović, "Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming," *Proceedings of the 2001 American Control Conference*, pp. 2394–2399, 2001.

[6] K. Kvaternik and L. Pavel, "Interconnection conditions for the stability of nonlinear sampled-data extremum seeking schemes," *IEEE Conference on Decision and Control and European Control Conference*, pp. 4448–4454, 2011.

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

[8] L. Hazeleger, D. Nešić, and N. van de Wouw, "Sampled-data extremum-seeking framework for constrained optimization of nonlinear dynamical systems," *Automatica*, vol. 142, p. 110415, 2022.

[9] W. H. Moase, C. Manzie, and M. J. Brear, "Newton-like extremum-seeking part I: Theory," *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3839–3844, 2009.

[10] J. J. Ryan and J. L. Speyer, "Peak-seeking control using gradient and Hessian estimates," *Proceedings of the 2010 American Control Conference*, pp. 611–616, 2010.

[11] M. A. M. Haring and T. A. Johansen, "On the accuracy of gradient estimation in extremum-seeking control using small perturbations," *Automatica*, vol. 95, pp. 23–32, 2018.

[12] B. G. B. Hunnekens, M. A. M. Haring, N. van de Wouw, and H. Nijmeijer, "A dither-free extremum-seeking control approach using 1st-order least-squares fits for gradient estimation," *53rd IEEE Conference on Decision and Control*, pp. 2679–2684, 2014.

[13] J. I. Poveda, M. Benosman, and K. G. Vamvoudakis, "Data-enabled extremum seeking: A cooperative concurrent learning-based approach," *International Journal of Adaptive Control and Signal Processing*, vol. 35, pp. 1256–1284, 2021.

[14] E. T. Maddalena, P. Scharnhorst, and C. N. Jones, "Deterministic error bounds for kernel-based learning techniques under bounded noise," *Automatica*, vol. 134, p. 109896, 2021.

[15] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.

[16] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*, D. Helmbold and B. Williamson, Eds. Springer Berlin Heidelberg, 2001, pp. 416–426.

[17] D.-X. Zhou, "Derivative reproducing properties for kernel methods in learning theory," *Journal of Computational and Applied Mathematics*, vol. 220, pp. 456–463, 2008.

[18] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer New York, 2006.