# The State Following Approximation Method

Joel A. Rosenfeld, Rushikesh Kamalapurkar, *Senior Member, IEEE*, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*— A function approximation method is developed which aims to approximate a function in a small neighborhood of a state that travels within a compact set. The method provides a novel approximation strategy for the efficient approximation of nonlinear functions for real-time simulations and experiments. The development is based on the theory of universal reproducing kernel Hilbert spaces over the $n$-dimensional Euclidean space. Several theorems are introduced which support the development of this state following (StaF) method. In particular, it is shown that there is a bound on the number of kernel functions required for the maintenance of an accurate function approximation as a state moves through a compact set. In addition, a weight update law, based on gradient descent, is introduced where arbitrarily close accuracy can be achieved provided the weight update law is iterated at a sufficient frequency, as detailed in Theorem 4. An experience-based approximation method is presented which utilizes the samples of the estimations of the ideal weights to generate a global approximation of a function. The experience-based approximation interpolates the samples of the weight estimates using radial basis functions. To illustrate the StaF method, the method is utilized for derivative estimation, function approximation, and is applied to an adaptive dynamic programming problem where it is demonstrated that the stability is maintained with a reduced number of basis functions.**

***Index Terms*— Approximate dynamic programming, approximation theory, reproducing kernel Hilbert spaces, state following approximation.**

## I. Introduction

OFTEN in the theory of approximation, an accurate estimation of a function over a large compact set is sought [1], [2]. It is well known that the larger the compact set, a correspondingly larger number of basis functions are required to achieve an accurate function approximation. There is a large body of the literature concerned with methods for the reduction of the number of basis functions required to achieve such an approximation (see [3]–[5]).

In many control applications, function approximation is used to generate a stabilizing controller of a state in a dynamical system. For instance, in adaptive dynamic programming (ADP), an approximation of the optimal value function is leveraged to produce an approximate optimal controller [6]–[15]. Traditionally, the approximation is sought over a large compact set and requires many basis functions. The computational resources required to tune the weights of the basis functions renders real-time implementation of controllers based on ADP methods infeasible.

Motivated by problems in control theory, this paper introduces an approximation methodology that aims to establish and maintain an accurate approximation of a function in a neighborhood of a moving state in a dynamical system. The method, deemed the state following (StaF) method, reduces the number of basis functions required to achieve an accurate approximation by focusing on the approximation of a function over a small neighborhood by linear combinations of time and state varying basis functions. Therefore, even in cases where processing power of onboard CPUs is limited, an accurate approximation of a function can be maintained.

The particular basis functions that will be employed throughout this paper are derived from kernel functions corresponding to reproducing kernel Hilbert spaces (RKHSs). In particular, the centers are selected to be continuous functions of the state variable bounded by a predetermined value. That is, given a compact set $D \subset \mathbb{R}^n$, $\epsilon > 0$, $r > 0$, and $M \in \mathbb{N}$, $c_i(x) = x + d_i(x)$ where $d_i : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and $\sup_{x \in D} \|d_i(x)\| < r$ for $i = 1, \ldots, M$. The parameterization of a function $V : D \to \mathbb{R}$ in terms of StaF kernel functions is given by

$$\hat{V}(y; x(t), t) = \sum_{i=1}^{M} w_i(t) K(y, c_i(x(t)))$$

where $w_i(t)$ is a weight signal chosen to satisfy

$$\limsup_{t \to \infty} E_r(x(t), t) < \epsilon$$

where $E_r$ is a measure of the accuracy of an approximation in a neighborhood of $x(t)$, such as that of the supremum norm

$$E_r(x(t), t) = \sup_{y \in \overline{N_r(x(t))}} |V(y) - \hat{V}(y; x(t), t)|.$$

The goal of the StaF method is to establish and maintain an approximation of a function in a neighborhood of the state. The justification for this approach stems from the observation that an optimal controller only requires the value of the

Manuscript received February 23, 2017; revised November 9, 2017, March 9, 2018, and June 22, 2018; accepted September 2, 2018. Date of publication October 25, 2018; date of current version May 23, 2019. This work was supported in part by NSF under Grant 1509516 and in part by the Office of Naval Research under Grant N00014-13-1-0151. The work of J. A. Rosenfeld was supported by the Air Force Office of Scientific Research under Contract FA9550-15-1-0258, Contract FA9550-16-1-0246, and Contract FA9550-18-1-0122. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the sponsoring agencies. This paper was presented at the Proceedings of the 2015 American Control Conference [29]. *(Corresponding author: Joel A. Rosenfeld.)*

J. A. Rosenfeld is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA (e-mail: joelar@ufl.edu).

R. Kamalapurkar is with the Systems, Cognition, and Control Laboratory, Department of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078 USA.

W. E. Dixon is with the Nonlinear Controls and Robotics Laboratory, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2870040

estimation of the optimal value function to be accurate at the current system state. Thus, when computational resources are limited, computational efforts should be focused on improving the accuracy of approximations near the system state.

The advantage of using RKHSs for the purpose of local approximations is twofold. RKHSs have been found to be effective for nonlinear function approximation [16], and the use of RKHS can enable accurate estimations of a wide array of nonlinear functions. Also, the ideal weights corresponding to the Hilbert space norm provided by RKHSs change smoothly with respect to smooth changes in the centers, as demonstrated in Theorem 3, which allows the execution of weight update laws to achieve and maintain an accurate approximation. The ideal weights in the context of the StaF approximation method become a continuous function of the state and are investigated in Section V.

Previous efforts in the literature have performed nonlinear approximation through the adjustment of the centers of radial basis functions (see [17]–[19]) as a means to determine the optimal centers for global approximation. These efforts are more applicable when offline techniques can be used due to computational demands. For other applications where computational resources are limited, global approximations may not be feasible (especially as the dimension of the problem grows), nor is the optimal selection of parameters.

Other approaches that use only local approximation for function estimation, such as moving least-squares (MLS) approximation are similar to the StaF method [20]–[26]. In MLS approximation, weight functions are modified based on the state, and at each point a least-squares problem is solved to produce an approximation [21]. The StaF method leverages the theory of RKHSs to produce a local Hilbert space norm, and the approximation is estimated producing weights associated with real-time moving centers (or samples) that minimize the local Hilbert space norm. Instead of a fixed collection of samples used in MLS approximations, samples are continuously updated for real-time approximations with the StaF method. The StaF method may be implemented as an analog to the MLS method from a collection of samples by discretely (i.e., discontinuously) moving the StaF centers to the sample points when the neighborhood about the state encompasses the sample. The advantage gained by continuously moving the centers with the state is realized through the ability to use weight update laws to improve the estimation in real time. It should be noted that in MLS the term weight function refers to functions that determine the least-squares norm, whereas the term weight function in StaF refers to the ideal weights for approximation determined by the local Hilbert space norm. The two methods are contrasted in Section IX.

The purpose of this paper is to provide a mathematical justification for the StaF method as implemented in [27] and [28]. Their several assumptions are made about the existence and differentiability of ideal weight functions, which are critical for the stability analysis in [27] and [28]. The preliminary work for this paper was published in conference proceedings [29], where several weaker results were established such as Proposition 2 presented in this paper. The theoretical development of [29] does not provide convergence guarantees in terms of Hilbert space norms, which is the natural setting for kernel functions. This paper lays the mathematical foundation for the establishment and maintenance of a real-time moving local approximation of a continuous function and establishes results in a proper framework for RKHS. Moreover, this paper introduces a method of reconstruction of a function through the approximation of the ideal weight functions as a means of an experience-based approximation. In scenarios where a function is not directly sampled but weight update laws are used to improve an approximation, as in ADP and system identification, an experience-based approximation provides a means of constructing an approximation based on the estimates of the ideal weight function. The method is demonstrated for both derivative estimations as well as for approximating the value function.

Section III frames the particular approximation problem of the StaF method. Section IV demonstrates accurate approximation with a fixed number of moving basis functions. The accompanying supplementary materials provide a demonstration of an explicit bound on the number of required StaF basis functions for the case of the exponential kernel function. The ideal weight function arising from the StaF method is introduced and discussed in Section V, where the existence and smoothness of the ideal weight function are established. Section VI provides a proof of concept demonstrating the existence of weight update laws to maintain an accurate approximation of a function in a local neighborhood, ultimately establishing a uniform ultimate bounded (UUB) result. The remaining sections demonstrate the developed method through numerical experiments and discussions of applications. Specifically, Section VIII gives the results of a "gradient chase" algorithm. In Section VIII, the utility of StaF methods are demonstrated in an ADP application.

## II. Preliminaries

An RKHS, $H$, is a Hilbert space with inner product $\langle \cdot, \cdot \rangle_H$ of functions $f : X \to \mathbb{F}$ (where $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$) for which given any $x \in X$, the functional $E_x f := f(x)$ is bounded. By the Reisz representation theorem, for each $x \in X$ there is a unique function $k_x \in H$ for which $\langle f, k_x \rangle_H = f(x)$. Each function $k_x$ is called a reproducing kernel for the point $x \in X$. The function $K(x, y) = \langle k_y, k_x \rangle_H$ is called the kernel function for $H$ [30]. The norm corresponding to $H$ will be denoted as $\| \cdot \|_H$, and the subscript will be suppressed when the Hilbert space is understood. Kernel functions are dense in $H$ under the RKHS norm.

Kernel functions have the property that for each collection of points $\{x_1, \ldots, x_m\} \subset X$, the matrix $(K(x_i, x_j))_{i,j=1}^m$ is positive semidefinite. The Aronszajn–Moore theorem states that there is a one-to-one correspondence between kernel functions with this property and RKHSs. In fact, starting with a kernel function having the positive semidefinite property, there is an explicit construction for its RKHS. Generally, the norm for the RKHS is given by

$$\|f\|_H := \sup\{\|P_{c_1,\ldots,c_M} f\|_H : M \in \mathbb{N} \text{ and } c_1, \ldots, c_M \in X\} \tag{1}$$

where $P_{c_1,\ldots,c_m} f$ is the projection of $f$ onto the subspace of $H$ spanned by the kernel function $K(\cdot, c_i)$ for $i = 1, \ldots, M$. $P_{c_1,\ldots,c_M} f$ is computed by interpolating the points $(c_i, f(c_i))$ for $i = 1, \ldots, M$ with a function of the form $\sum_{i=1}^{M} w_i K(\cdot, c_i)$. The norm of the projection then becomes[1] $\|P_{c_1,\ldots,c_M} f\| = (\sum_{i,j=1}^{M} c_i \overline{c_j} K(c_j, c_i))^{1/2}$. In practice, the utility of computing the norm of $f$ as (1) is limited, and alternate forms of the norm are sought for specific RKHSs.

Unlike $L^2$ spaces, norm convergence in an RKHS implies pointwise convergence. This follows since if $f_n \to f$ in the RKHS norm, then

$$|f(x) - f_n(x)| = |\langle f - f_n, k_x \rangle| \leq \|f - f_n\| \|k_x\| = \|f - f_n\| \sqrt{K(x,x)}.$$

When $K$ is a continuous function of $X$, the term $\sqrt{K(x,x)}$ is bounded over compact sets, and thus, norm convergence implies uniform convergence over compact sets. Therefore, the problem of establishing an accurate approximation in the supremum norm of a function is often relaxed to determining an accurate approximation of a function in the RKHS norm.

Given an RKHS $H$ over a set $X$ and $Y \subset X$, the space $H_Y$ obtained by restricting each function $f \in H$ to the set $Y$ is itself an RKHS where the kernel function is given by restricting the original kernel function to the set $Y \times Y$. The resulting Hilbert space norm is given by

$$\|g\|_{H_Y} = \inf\{\|f\|_H : f \in H \text{ and } f|_Y = g\}.$$

Therefore, the map $f \mapsto f|_Y$ is norm decreasing from $H$ to $H_Y$ [30]. For the purposes of this paper, the norm obtained by restricting an RKHS $H$ over $\mathbb{R}^n$ to a closed neighborhood

$$\overline{N_r(x)} := \{y \in \mathbb{R}^n : \|x - y\|_2 \leq r\}$$

where $r > 0$ and $x \in \mathbb{R}^n$ will be denoted as $\|\cdot\|_{r,x}$.

## III. StaF Problem Statement

Given a continuous function $V : \mathbb{R}^n \to \mathbb{R}$, $r > 0$, an arbitrarily small $\epsilon > 0$, and a dynamical system $\dot{x} = f(x, u)$ (where $f$ is sufficiently regular for the system to be well defined), the goal of the StaF approximation method is to select state and time-varying basis functions $\sigma_i : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ for $i = 1, 2, \ldots, M$ and weight signals $w_i : \mathbb{R}_+ \to \mathbb{R}$ for $i = 1, 2, \ldots, M$ such that

$$\limsup_{t \to \infty} \sup_{y \in \overline{N_r(x(t))}} \left| V(y) - \sum_{i=1}^{M} w_i(t) \sigma_i(y; x(t), t) \right| < \epsilon. \quad (2)$$

In other words, the StaF approximation method aims to achieve an arbitrarily small steady-state error of order $\epsilon$ in a closed neighborhood of the state, $\overline{N_r(x(t))} = \{y \in \mathbb{R}^n : \|x(t) - y\|_2 \leq r\}$.

Central problems to the StaF method include determining the basis functions and the weight signals. When RKHSs are used for basis functions, (2) can be relaxed to where the supremum norm is replaced with the Hilbert space norm. Since the Hilbert space norm of an RKHS dominates the supremum norm, (2) with the supremum norm is simultaneously satisfied. Moreover, when using an RKHS, the basis functions can be selected to correspond to centers placed in a moving neighborhood of the state. In particular, given a kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ corresponding to a (universal) RKHS, $H$, and continuous center functions $c_i : \mathbb{R}^n \to \mathbb{R}^n$ for which $d_i(x) := c_i(x) - x$ is bounded by $r$, then the StaF problem becomes the determination of weight signals $w_i : \mathbb{R}_+ \to \mathbb{R}$ for $i = 1, \ldots, M$ such that

$$\limsup_{t \to \infty} \left\| V(\cdot) - \sum_{i=1}^{M} w_i(t) K(\cdot, c_i(x(t))) \right\|_{r,x(t)} < \epsilon \quad (3)$$

where $\|\cdot\|_{r,x(t)}$ is the norm of the RKHS obtained by restricting functions in $H$ to $N_r(x(t))$.

Since (3) implies (2), the focus of this paper is to demonstrate the feasibility of satisfying (3). Theorem 1 demonstrates that under a certain continuity assumption, a bound on the number of kernel functions necessary for the maintenance of an approximation throughout a compact set can be determined, and Theorem 3 shows that a collection of continuous ideal weight functions can be determined to satisfy (3). Theorem 3 justifies the use of weight update laws for the maintenance of an accurate function approximation, and this is demonstrated by Theorem 4 as well as the numerical results contained in Section VIII-A.

The choice of RKHS for Section VIII is that which corresponds to the exponential kernel $K(x, y) = \exp(x^T y)$ where $x, y \in \mathbb{R}^n$ and will be denoted by $F^2(\mathbb{R}^n)$ since it is closely connected to the Bargmann–Fock space [31]. The RKHS corresponding to the exponential kernel is a universal RKHS [30], [32], which means that given any compact set $D \subset \mathbb{R}^n$, $\epsilon > 0$ and continuous function $f : D \to \mathbb{R}$, there exists a function $\hat{f} \in F^2(\mathbb{R}^n)$ for which $\sup_{x \in D} |f(x) - \hat{f}(x)| < \epsilon$.

## IV. Feasibility of the StaF Approximation and the Ideal Weight Functions

The first theorem concerning the StaF method demonstrates that if the state variable is constrained to a compact subset of $\mathbb{R}^n$, then, there is a finite number of StaF basis functions required to establish the accuracy of an approximation.

*Theorem 1:* Suppose that $K : X \times X \to \mathbb{C}$ is a continuous kernel function corresponding to an RKHS, $H$, over a set $X$ equipped with a metric topology. If $V \in H$, $D$ is a compact set of $X$, $r > 0$, and $\|V\|_{r,x}$ is continuous with respect to $x$, then for all $\epsilon > 0$ there is a $M \in \mathbb{N}$ such that for each $x \in D$ there are centers $c_1, c_2, \ldots, c_M \in N_r(x)$ and weights $w_i \in \mathbb{C}$ such that

$$\left\| V(\cdot) - \sum_{i=1}^{M} w_i K(\cdot, c_i) \right\|_{r,x} < \epsilon.$$

*Proof:* Let $H_{x,r}$ be the RKHS obtained by restricting the functions of $H$ to the set $N_r(x)$. The span of the collection of kernel functions, $\{K(\cdot, y) : y \in N_r(x)\}$, is dense in $H_{x,r}$ [30]. Given $\epsilon > 0$, for each neighborhood $N_r(x)$ with $x \in D$, there

[1] For $z \in \mathbb{C}$, the quantity $\mathrm{Re}(z)$ is the real part of $z$, and $\overline{z}$ represents the complex conjugate of $z$.

exists a finite number of centers $c_1, \ldots, c_M \in N_r(x)$, and weights $w_1, \ldots, w_M \in \mathbb{C}$, such that

$$\left\| V(\cdot) - \sum_{i=1}^{M} w_i K(\cdot, c_i) \right\|_{r,x} < \epsilon.$$

Let $M_{x,\epsilon}$ be the minimum such number. The claim of the proposition is that the set $Q_\epsilon := \{M_{x,\epsilon} : x \in D\}$ is bounded. Assume by way of contradiction that $Q_\epsilon$ is unbounded. If $Q_\epsilon$ is unbounded, it follows that the set $D$ is not finite, since otherwise $Q_\epsilon$ must be bounded. For each $z \in D$, then by the unboundedness of $Q_\epsilon$, there is a $z' \in D$ for which $M_{z,\epsilon} + 1 < M_{z',\epsilon}$. Thus, there exists a sequence $\{x_n\} \subset D$ for which $\{M_{x_n,\epsilon}\}$ is unbounded. By [30, Lemma A.2.10], there exists a subsequence $\{x_{n_k}\}$ for which $x_{n_k}$ converges to some point $x$. Since $D$ is compact, it is closed. Therefore, $x \in D$. Without loss of generality, the sequence $\{x_n\}$ is henceforth identified with its convergent subsequence, $\{x_{n_k}\}$, to simplify the notation of the following argument. Let $c_1, \ldots, c_{M_{x,\epsilon/2}} \in N_r(x)$ and $w_1, \ldots, w_{M_{x,\epsilon/2}} \in \mathbb{C}$ be centers and weights for which

$$E(x) := \left\| V(\cdot) - \sum_{i=1}^{M_{x,\epsilon/2}} w_i K(\cdot, c_i) \right\|_{r,x} < \epsilon/2. \tag{4}$$

For convenience, let each $c_i \in N_r(x)$ be expressed as $x + d_i$ for $d_i \in N_r(0)$. The function $E(x)$ in (4) can be written as

$$\left( \|V\|_{r,x} - 2Re\left( \sum_{i=1}^{M_{x,\epsilon/2}} w_i V(x + d_i) \right) + \sum_{i,j=1}^{M_{x,\epsilon/2}} w_i \overline{w_j} K(x + d_i, x + d_j) \right)^{1/2}.$$

By the hypothesis, $K$ is continuous with respect to $x$, which implies that $V$ is continuous [1], and $\|V\|_{r,x}$ is continuous with respect to $x$. Hence, there exists $\eta > 0$ for which $|E(x) - E(x_n)| < \epsilon/2$ for all $x_n \in N_\eta(x)$. Thus, $E(x_n) < E(x) + \epsilon/2 < \epsilon$ for sufficiently large $n$. By minimality $M_{x_n,\epsilon} < M_{x,\epsilon/2}$ for sufficiently large $n$. This is a contradiction. ∎

The assumption of the continuity of $\|V\|_{r,x}$ in Theorem 1 is well founded. There are several examples where the assumption is known to hold. For instance, if the RKHS is a space of real entire functions, as it is for the exponential kernel, then $\|V\|_{r,x}$ is not only continuous but it is constant.

Using a similar argument as that in Theorem 1, the theorem can be shown to hold when the restricted Hilbert space norm is replaced by the supremum norm over $\overline{N_r(x)}$. The proof of the following theorem can be found in the preliminary work for this paper in [29].

*Proposition 2:* Let $D$ be a compact subset of $\mathbb{R}^n$, $V : \mathbb{R}^n \to \mathbb{R}$ be a continuous function, and $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a continuous and universal kernel function. For all $\epsilon, r > 0$, there exists $M \in \mathbb{N}$ such that for each $x \in D$, there is a collection of centers $c_1, \ldots, c_M \in N_r(x)$ and weights $w_1, \ldots, w_M \in \mathbb{R}$ such that

$$\sup_{y \in \overline{N_r(x)}} \left| V(y) - \sum_{i=1}^{M} K(y, c_i) \right| < \epsilon.$$

An example of a computable bound for the exponential kernel in the setting of the supremum norm can be found in the supplementary materials.

## V. Existence and Smoothness of the Ideal Weight Function

Theorem 1 and Proposition 2 establish that given a kernel function, a finite number of centers can be used to yield an arbitrarily accurate estimation of a function. The supplementary material provides an example, using the exponential kernel function, demonstrating that the number of centers required to achieve the desired approximation accuracy can be calculated in some cases. However, further investigation is required to understand the characteristics of the ideal weights that correspond to the moving centers. For example, in control applications involving function approximation or system identification, it is assumed that there is a collection of constant ideal weights, and much of the theory is in the demonstration of the convergence of approximate weights to the ideal weights. The subsequent Theorem 3 establishes that ideal weights, which are functions of the state-dependent centers, are $m$-times continuously differentiable. This property can then be used to develop weight update laws (see Section VI) that may be subsequently used to prove Lyupunov-based stability theorems (see [27] as well as Section VIII-A).

Since the ideal weights corresponding to a Hilbert space norm are unique, Theorem 3 is framed in the Hilbert space setting of (3). Theorem 3, together with Theorem 1, provides the StaF framework for RKHSs.

*Theorem 3:* Let $H$ be an RKHS over a set $X \subset \mathbb{R}^n$ with a strictly positive kernel $K : X \times X \to \mathbb{C}$ such that $K(\cdot, c) \in C^{m_0}(\mathbb{R}^n)$ for all $c \in X$. Suppose that $V \in H$. Let $C$ be an ordered collection of $M$ distinct centers, $C = (c_1, c_2, \ldots, c_M) \in X^M$, with the associated ideal weights

$$W(C) = \underset{(a_i)_{i=1}^M \in \mathbb{C}^M}{\arg\min} \left\| \sum_{i=1}^{M} a_i K(\cdot, c_i) - V(\cdot) \right\|_H. \tag{5}$$

The function $W$ is $m_0$-times continuously differentiable with respect to each component of $C$.

*Proof:* The determination of $W(C)$ is equivalent to computing the projection of $V$ onto the space $Y = \text{span}\{K(\cdot, c_i) : i = 1, \ldots, M\}$. To compute the projection, a Gram–Schmidt algorithm can be employed. The Gram–Schmidt algorithm is most easily expressed in its determinant form. Let $D_0 = 1$ and $D_m = \det(K(c_j, c_i))_{i,j=1}^m$, then for $m = 1, \ldots, M$ the functions

$$u_m(x) := \frac{1}{\sqrt{D_{m-1} D_m}} \times \det \begin{pmatrix} K(c_1, c_1) & \cdots & K(c_1, c_m) \\ \vdots & \ddots & \vdots \\ K(c_{m-1}, c_1) & \cdots & K(c_{m-1}, c_m) \\ K(x, c_1) & \cdots & K(x, c_m) \end{pmatrix}$$

constitute an orthonormal basis for $Y$. Since $K$ is strictly positive definite, $D_m$ is positive for each $m$ and every $C$. The coefficient for each $K(x, c_l)$ with $l = 1, \ldots, m$ in $u_m$ is a

sum of products of the terms $K(c_i, c_j)$ for $i, j = 1, \ldots m$. Each such coefficient is $m_0$-times differentiable with respect to each $c_i$, $i = 1, \ldots, M$. When $\langle V,\ u_m \rangle$ is computed for the projection, the result is a linear combination of evaluations of $V$ at each of the centers. Since $V \in H$ and the RKHS kernel function, $K$, is $m_0$-times differentiable, the function $V$ is $m_0$-times continuously differentiable by [30, Corollary 4.36]. Therefore, $\langle V,\ u_m \rangle$ is continuous with respect to the centers. Finally, each term in $W(C)$ is a linear combination of the coefficients determined by $u_m$ for $m = 1, \ldots, M$, and thus is $m_0$-times continuously differentiable with respect to each $c_i$ for $i = 1, \ldots, M$. ■

## VI. Gradient Chase Theorem

As mentioned before, control theory problems involving function approximation and system identification are centered around the concept of weight update laws. Weight update laws are a collection of rules that the approximating weights must obey to achieve convergence to the ideal weights. In the case of the StaF approximation framework, the ideal weights are replaced with ideal weight functions. Theorem 3 showed that the ideal weight functions will change smoothly with respect to smooth changes in the centers. Thus, weight update laws can be used to establish and maintain an accurate approximation of the ideal weight function.

Theorem 4 provides an example of such weight update laws that achieve a predetermined limiting error bound, called a UUB result in the engineering literature [33]. The theorem takes advantage of perfect samples of a function in the RKHS $H$ corresponding to a real-valued kernel function.

The proof of the theorem is similar to the standard proof for the convergence of the gradient descent algorithm for a quadratic programming problem [34]. The contribution of the proof is in a modification, where the mean value theorem is used to produce an extra term which yields a UUB result, and the continuity of the largest and smallest eigenvalues of a Gram matrix is used to get a uniform bound in tandem with the Kantorovich inequality. Following conventions of the optimization community, $x^k$ refers to the $k$th iteration of a variable in the following theorem.

*Theorem 4 (Gradient Chase Theorem):* Let $H$ be a real-valued RKHS over $\mathbb{R}^n$ with a continuously differentiable strictly positive definite kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Let $V \in H$, $D \subset \mathbb{R}^n$ be a compact set, and $x : \mathbb{R} \to \mathbb{R}^n$ a state variable for the dynamical system $\dot{x} = q(x, t)$, where $q : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$ is a bounded locally Lipschitz continuous function. Further suppose that $x(t) \in D$ for all $t > 0$. Let $c : \mathbb{R}^n \to \mathbb{R}^M$, where for each $i = 1, \ldots, M$, $c_i(x) = x + d_i(x)$ where $d_i \in C^1(\mathbb{R}^n)$, and let $a \in \mathbb{R}^M$. Consider the function

$$F(a, c) = \left\| V - \sum_{i=1}^{M} a_i K(\cdot, c_i(x)) \right\|_H^2.$$

At each time instance $t > 0$, there is a unique $W(t)$ for which

$$W(t) = \underset{a \in \mathbb{R}^M}{\arg\min}\, F(a, c(x(t))).$$

Given any $\epsilon > 0$ and initial value $a^0$, there is a frequency $\tau > 0$, where if the gradient descent algorithm (with respect to $a$) is iterated at time steps $\Delta t < \tau^{-1}$, then $F(a^k, c^k) - F(w^k, c^k)$ will approach a neighborhood of radius $\epsilon$ as $k \to \infty$.

*Proof:* Let $\bar{\epsilon} > 0$. By the Hilbert space structure of $H$

$$F(a, c) = \|V\|_H^2 - 2V(c)^T a + a^T K(c) a \tag{6}$$

where $V(c) = (V(c_1), \ldots, V(c_M))^T$ and $K(c) = (K(c_i, c_j))_{i,j=1}^M$ are the symmetric strictly positive kernel matrices corresponding to $c$. At each time iteration $t^k$, $k = 0, 1, 2, \ldots$, the corresponding centers and weights can be written as $c^k \in \mathbb{R}^{nM}$ and $a^k \in \mathbb{R}^M$, respectively. The ideal weights corresponding to $c^k$ will be denoted by $w^k$. It can be shown that $w^k = K(c^k)^{-1} V(c^k)$ and $F(w^k, c^k) = \|V\|_H^2 - V(c^k)^T K(c^k) V(c^k)$. Theorem 3 ensures that the ideal weights change continuously with respect to the centers which remain in a compact set $\tilde{D}^M$, where $\tilde{D} = \{x \in \mathbb{R}^M : \|x - D\| \leq \max_{i=1,\ldots,M}(\sup_{x \in D} |d_i(x)|)\}$, so the collection of ideal weights is bounded. Let $R > \bar{\epsilon}$ be large enough so that $N_R(0)$ contains both the initial value $a^0$ and the set of ideal weights. To facilitate the subsequent analysis, consider the constants

$$\begin{aligned}
R_0 &= \max_{x \in D, t > 0} |q(x, t)| \\
R_1 &= \max_{a \in \overline{N_r(0)}, c \in \tilde{D}} |\nabla_a F(a, c)| \\
R_2 &= \max_{c \in \tilde{D}} |\nabla_c F(w(c), c)| \\
R_3 &= \max_{c \in \tilde{D}} |\dot{d}_i(x(t)| \\
R_4 &= \max_{c \in \tilde{D}} \left\| \frac{d}{dc} w(c) \right\|
\end{aligned}$$

where $\nabla_a$ is the gradient with respect to $a$, and let $\Delta t < \tau^{-1} := \bar{\epsilon} \cdot (2(R_0 + R_3) \cdot (R_1 \cdot R_4 \cdot (R_0 + R_3) + R_2 + 1))^{-1}$. The proof aims to show that by using the gradient descent law for choosing $a^k$, the following inequality can be achieved:

$$\frac{F(a^{k+1}, c^{k+1}) - F(w^{k+1}, c^{k+1})}{F(a^k, c^k) - F(w^k, c^k)} < \delta + \frac{\bar{\epsilon}}{F(a^k, c^k) - F(w^k, c^k)}$$

for some $0 < \delta < 1$. Set

$$a^{k+1} = a^k + \lambda g \tag{7}$$

where $g = -\nabla_a F(a^k, c^k) = 2V(c^k) - 2K(c^k) a^k$ and $\lambda$ is selected so that the quantity $F(a^k + \lambda g, c^k)$ is minimized. $\lambda$ that minimizes this quantity is $\lambda = (g^T g / (2 g^T K(c^k) g))$ which yields $F(a^{k+1}, c^k) = F(a^k, c^k) - ((g^T g)^2)/(4 g^T K(c^k) g)$. Since $F(a^{k+1}, c^{k+1})$ is continuously differentiable in the second variable, we have $F(a^{k+1}, c^{k+1}) = F(a^{k+1}, c^k) + \nabla_c F(a^{k+1}, \eta) \cdot (c^{k+1} - c^k)$. Since $|\dot{c}(x(t))| < R_0 + R_3$, an application of the mean value theorem demonstrates that $\|c^{k+1} - c^k\| < (R_0 + R_3) \Delta t$. Thus,

$$F(a^{k+1}, c^{k+1}) = F(a^{k+1}, c^k) + \epsilon_1(t^k)$$

where $|\epsilon_1(t^k)| < \bar{\epsilon}/2$ for all $k$. The quantity $F(w^{k+1}, c^{k+1})$ is continuously differentiable in both variables. Thus, by the multivariable chain rule and another application of the mean value theorem

$$F(w^{k+1}, c^{k+1}) = F(w^k, c^k) + \epsilon_2(t^k)$$

for $|\epsilon_2(t^k)| < \bar{\epsilon}/2$ for all $k$. Therefore, the following is established:

$$\frac{F(a^{k+1}, c^{k+1}) - F(w^{k+1}, c^{k+1})}{F(a^k, c^k) - F(w^k, c^k)} = \frac{F(a^{k+1}, c^k) - F(w^k, c^k) + (\epsilon_1(t^k) - \epsilon_2(t^k))}{F(a^k, c^k) - F(w^k, c^k)} = 1 - \frac{(g^T g)^2}{(g^T K(c^k) g)(g^T K(c^k)^{-1} g)} + \frac{\epsilon_1(t^k) - \epsilon_2(t^k)}{F(a^k, c^k) - F(w^k, c^k)}.$$

The Kantorovich inequality [34] yields

$$1 - \frac{(g^T g)^2}{(g^T K(c^k) g)(g^T K(c^k)^{-1} g)} \leq \left(\frac{A_{c^k}/a_{c^k} - 1}{A_{c^k}/a_{c^k} + 1}\right)^2 \quad (8)$$

where $A_{c^k}$ is the largest eigenvalue of $K(c^k)$ and $a_{c^k}$ is the smallest eigenvalue of $K(c^k)$. The quantity on the right of (8) is continuous with respect to $A_{c^k}$ and $a_{c^k}$. In turn, $A_{c^k}$ and $a_{c^k}$ are continuous with respect to $K(c^k)$ (see [35, Exercise 4.1.6]) which is continuous with respect to $c^k$. Therefore, there is a largest value, $\delta$ that the right-hand side of (8) obtains on the compact set $\tilde{D}$ and this value is less than 1. Moreover, $\delta$ is independent of $\bar{\epsilon}$, so it may be declared that $\bar{\epsilon} = \epsilon(1 - \delta)$. Finally

$$\frac{F(a^{k+1}, c^{k+1}) - F(w^{k+1}, c^{k+1})}{F(a^k, c^k) - F(w^k, c^k)} \leq \delta + \frac{(\epsilon_1(t^k) - \epsilon_2(t^k))}{F(a^k, c^k) - F(w^k, c^k)}.$$

Therefore, setting $e(k) = F(a^k, c^k) - F(w^k, c^k)$, it can be shown that $e(k+1) \leq \delta e(k) + \epsilon(1 - \delta)$ and the conclusion of the theorem follows. ■

*Corollary 5:* In the framework of Theorem 4, if the gradient descent algorithm is iterated $\mu \in \mathbb{N}$ times per time step, the resulting error bound becomes

$$e(k+1) \leq \delta^\mu e(k) + \epsilon(1 - \delta^\mu)$$

where $k$ represents the number of time steps in the system.

## VII. Experience-Based Approximation

This section presents a method of using data recorded from the moving local approximation of the StaF method for the purpose of constructing a global approximation of a function. Instead of calculating an approximation of the function directly, continuity of the ideal weight function can be exploited to construct an approximation of the ideal weight function

$$W(x(t)) := \begin{pmatrix} w_1(x(t)) \\ \vdots \\ w_M(x(t)) \end{pmatrix} = \arg\min_{a \in \mathbb{R}^M} \left\| V(y) - \sum_{i=1}^{M} a_i K(y, c_i(x(t))) \right\|_{\overline{N_r(x(t))}} \quad (9)$$

over the entire compact domain. Approximation of the ideal weight function is then employed along with the StaF kernels to produce an approximation of the original function.

The approach uses estimates of ideal weights at the current state, produced by the weight update laws, as a sample of the ideal weight function. Since the ideal weight function is a continuous function of the state, these samples can be used to produce an approximation of the ideal weight function. Provided that the approximation of the ideal weight function is accurate enough, the approximation of the ideal weight function can be used in place of the ideal weight function to produce an accurate approximation of the original function.

### A. Collection of Sample Points for the Ideal Weight Function

Approximation of a function in this section occurs in two stages. The StaF algorithm is used as a system is running to generate ideal weight function estimates at points where the state visits. Estimated weight data are also collected and postprocessed offline to generate an approximation over the region the state has previously visited, henceforth the region of experience. These two stages are iterated until a sufficient amount of estimated weight data is collected to generate a global approximation over a compact set. For instance, suppose the region is $D = [-1, 1]^n \subset \mathbb{R}^n$. A grid of width $1/N$ for $N \in \mathbb{N}$ is laid over $D$. During a StaF approximation trial, the weight estimates are continuously updated as the state travels through $D$. When the state crosses a cell boundary, the weight estimate (the best estimate for the current cell) is recorded as a sample for later processing. The grid results in an almost uniform distribution of samples, which enables the use of approximation techniques such as [36]. The weight estimates will be labeled $\tilde{w}_i \subset \mathbb{R}^M$ for $i = 1, \ldots, N_0$ and correspond to the state $x_i \in D$ for $i = 1, \ldots, N_0$.

### B. Approximation via Scattered Data Interpolation

Scattered data interpolation via the Gaussian radial basis function (RBF), $K_G(x, y) = \exp(-((\|x - y\|^2)/\mu))$ can produce a continuously differentiable approximation (from $\mathbb{R}^n$ to $\mathbb{R}^M$) of (9) as

$$\hat{W}(x) = \begin{pmatrix} \hat{w}_1(x(t)) \\ \vdots \\ \hat{w}_M(x(t)) \end{pmatrix} := \left( \sum_{i=1}^{N_0} a_{i,j} K_G(x, x_i) \right)_{j=1}^{M}$$

where the matrix $\mathbf{a} = (a_{i,j}) \in \mathbb{R}^{N_0 \times M}$ satisfies the matrix equation: $G\mathbf{a} = \mathbf{w}$, where $G$ is the Gram matrix $(K_G(x_i, x_j))_{i,j=1}^{N_0}$ and $\mathbf{w} = (\tilde{w}_i) \in \mathbb{R}^{N_0 \times M}$ is the matrix of samples. The final approximation of the original function, $\hat{V}$, is given by

$$\hat{V}(x) = \hat{W}(x)^T (K(x, c_1(x)), \ldots, K(x, c_M(x)))^T = \sum_{j=1}^{M} \sum_{i=1}^{N_0} a_{i,j} K_G(x, x_i) K(x, c_j(x)). \quad (10)$$

The method and frequency by which the samples are collected effect the feasibility of approximations by scattered data interpolation. If the samples are too close, then the matrix $G$ becomes ill-conditioned, and it becomes difficult to determine $\mathbf{a}$. For this reason, the collection of samples will be uniformly distributed (see [36]), thereby mitigating the condition number problem by reducing clustering.

*Remark:* The resulting approximation may use a very large number of basis functions. The large number of basis functions

comes at the cost of efficient real-time estimation of a function by the StaF method. The weights for the global approximation are intended to be computed offline, when more computational resources are available, in contrast to the real-time approximation scheme afforded by the StaF method where computational power may be restricted.

## VIII. Examples and Numerical Experiments

The examples in this section illustrate the StaF method by establishing accurate function approximations with several weight update laws. Example 1 gives an example of establishing and maintaining the approximation of a function using StaF basis functions and a gradient descent update law as in (7) of Theorem 4. To improve the transient performance of the StaF method, the update law in (7) is iterated 10 times per time step and utilizes Corollary 5 to justify convergence. Example 2 utilizes the StaF approximation method to obtain an accurate estimation of a function's derivative. Derivative estimation requires greater accuracy, and as such, a faster weight update law was employed through the Nasterov accelerated gradient descent method (see [37, Section 3.7.1]) applied to the objective function $F_c(a) := F(a, c)$ as in (6). Since the centers $c$ follow the state variable, the optimum value of $F_c(a)$ is a moving target as it is in Theorem 4, and the Nasterov method is iterated to establish and maintain an accurate estimate. A control theoretic application is explored in Example 3, where the weight update laws are obtained from the Hamilton–Jacobi–Bellman (HJB) equation in an optimal control setting.

### Example 1 - Approximation Through the Gradient Chase Theorem

To demonstrate the effectiveness of the Gradient Chase theorem, a simulation performed on a 2-D linear system is presented in the following. The system dynamics are given by

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}$$

which is the dynamical system corresponding to a circular trajectory. The function to be approximated is

$$V(x_1, x_2) = x_1^2 + 5x_2^2 + \tanh(x_1 \cdot x_2)$$

and the kernel function to be used for function approximation is the exponential kernels, $K(x, y) = \exp(x^T y)$. The centers are arranged in an equilateral triangle centered about the state. In particular, each center resides on a circle of radius 0.1 centered at the state

$$c_i(x) = x + 0.1\begin{pmatrix}\sin((i-1)2\pi/3)\\ \cos((i-1)2\pi/3)\end{pmatrix}$$

for $i = 1, 2, 3$.

The initial values selected for the weights are $a^0 = [0\ 0\ 0]^T$. Theorem 4 provides guarantees on the accuracy of the estimation based on the ultimate velocity of the system state. The slower the system state, the more accurate is the ultimate result. To accelerate and improve the resulting estimation, this example iterates the gradient descent algorithm 10 times per 0.01-s time step. Thus, by Corollary 5, the error bound becomes $e(k+1) \leq \delta^{10} e(k) + \epsilon(1 - \delta^{10})$, where $k$ represents each time step. Fig. 1 presents the results of the simulation.

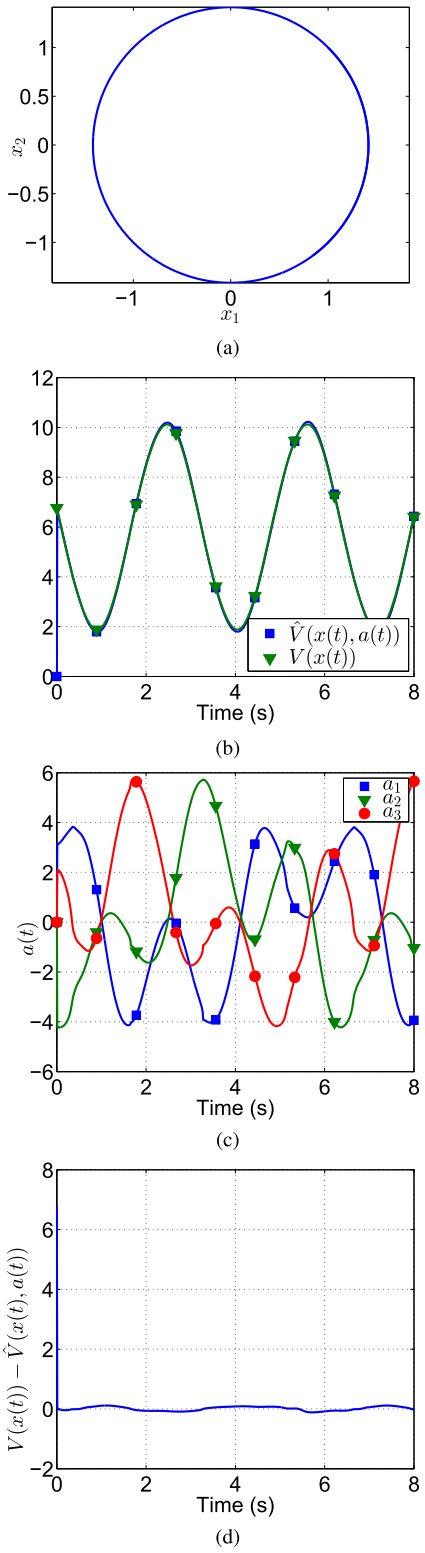


Fig. 1. Results of the numerical experiment demonstrating the gradient chase algorithm. (a) Trajectory of the state vector. (b) Comparison of $V$ and the approximation $\hat{V}$. (c) Values of the weight function estimates. (d) Error committed by the approximation at the current state.

Fig. 1(d) demonstrates that the function approximation error is regulated to a small neighborhood of zero as the gradient chase theorem is implemented and validates the claim of the

UUB result of Theorem 4. In Fig. 1(c), approximations of the ideal weight function can be seen to be periodic as well as smooth. The smoothness of the ideal weight function itself is given in Theorem 3, and the periodicity of the approximation follows from the periodicity of the selected dynamical system, as illustrated in Fig. 1(a). Fig. 1(b) presents a comparison of $V$ evaluated at the current state to that of the approximation evaluated at the current state. Approximation of the function is maintained as the system state moves through its domain as anticipated.

### EXAMPLE 2 - DERIVATIVE ESTIMATION

In this example, a scheme for derivative estimation is presented. In lieu of other derivative estimation methods, such as finite difference methods, a kernelized version of derivative estimation utilizing the StaF method is developed here instead. The motivation for this method stems from the exponential function's role as the eigenfunction of differentiation. Thus, if a differentiable function, $f : \mathbb{R} \to \mathbb{C}$, (together with some small $h > 0$) with values $f(t_0) = y_0, f(t_0 - h) = y_1$, and $f(t_0 - 2h) = y_2$ is interpolated by the function

$$\hat{f}(t) = w_0\ e^{t \cdot t_0} + w_1\ e^{t \cdot (t_0 - h)} + w_2\ e^{t \cdot (t_0 - 2h)}$$

then an estimate of $f$'s derivative at $t_0$ is given by

$$\hat{f}'(t_0) = w_0\ t_0\ e^{t_0 \cdot t_0} + w_1(t_0 - h)e^{t_0 \cdot (t_0 - h)} + w_2(t_0 - 2h)e^{t_0 \cdot (t_0 - 2h)}.$$

The universality of the exponential kernel and [30, Corollary 4.36] guarantees that $\hat{f}'(t_0)$ is an accurate estimation given a sufficient number of interpolation points are selected and $f$ is sufficiently regular. Theorem 1 guarantees a bound for the number of centers required for accurate function estimation. Moreover, for the exponential kernel in particular, a sufficiently dense collection of samples is sufficient to gain an accurate approximation of a function. Therefore, given a small enough approximation neighborhood, three centers is sufficient to give an accurate estimation of a function in that neighborhood. For more information concerning kernel-based numerical differentiation (refer to [38] and references contained therein).

The StaF method views the current time $t$ as the system state in this example. The StaF centers then become $c_1(t) = t$, $c_2(t) = t - h$, $c_3(t) = t - 2h$, and so on. For each time instance, interpolating $f$ involves solving $K(c(t))\mathbf{w}(t) = \mathbf{f}(t)$ where $K(c(t))$ is the matrix from Theorem 4, and $\mathbf{f}(t) = (f(c_1(t), f(c_2(t)), \ldots, f(c_M(t)))^T$. For accurate estimation of a function's derivative, $h$ must be selected to be small, which can result in a poorly conditioned $K(c(t))$. Thus, $K(c(t))$ can be difficult to invert. The StaF method then takes advantage of the continuity of $w(t)$ and can update $w(t)$ by either a gradient descent method (Theorem 4 for example) or by seeding a preconditioning technique with the most recent estimation of the weights.

In this example, an approximation of the derivative of $f(t) = \sin(2\pi t)$ was sought over the interval $[0.1, 2]$ using the StaF method. The interval $[0, 0.1]$ was used for initializing the method. In particular, the weight approximation was initialized by interpolating $f$ at the times $1/30$, $2/30$, and $3/30$ using the inverse operation provided by MATLAB. The values of the function $f$ were used for the initialization to help facilitate convergence to the steady-state approximation. However, other initial values may be selected for the weights and will result in varying transient performance, whereas the steady-state approximation will be unaffected. The simulation employed three StaF basis functions of exponential type, as did the previous example. Since the problem is 1-D, the centers were arranged to trail behind the current time at $c_1(t) = t$, $c_2(t) = t - 1/30$, and $c_3(t) = t - 2/30$. Nasterov's accelerated gradient descent method (see [37]) was employed as a weight update law. Nasterov's method provided a sharper estimation of the function's derivative than did performing the standard inverse operation in MATLAB. The Nasterov method was chosen since it can achieve a sufficient accuracy, and it demonstrates how the number of iterations of the gradient descent law can affect the overall accuracy. It should be noted that other iterative methods, such as conjugate gradient descent, may be used to produce sufficient accuracy with fewer iterations.

The simulation presented in Fig. 2 was incremented at timesteps of $\Delta t = 1/100$, and the Nasterov accelerated gradient descent method was employed at 10, 100, 1000, and 10 000 iterations per time step as shown in Fig. 2(a). Since the example demanded not only approximation of a function but also of its derivative, a much greater accuracy of the estimation of the ideal weights was required. The demand for greater accuracy led to a larger number of iterations of the gradient chase scheme.

*Experience-Based Approximation of the Derivative:* To employ the methods described in Section VII, a sample of the weight estimates was taken at intervals of $\Delta t' = 1/10$. The samples were interpolated by Gaussian RBFs of the form $K_G(x, y) = \exp(-((x - y)^2)/(1/2))$ with centers at $1/10,\ 2/10, \ldots, 19/10$. Interpolation of the sampled weight estimates (corresponding to the simulation with 10 000 iterations of gradient descent) are shown in Fig. 2(c). The resulting approximation is shown in Fig. 2(b).

Fig. 2(b) suggests that the experience-based approximation method is sensitive to errors in the estimations of the ideal weights. Notice that for both the 1000 and 10 000 iteration simulations, the corresponding StaF-based estimations of the derivative are both accurate in Fig. 2(a). However, Fig. 2(b) demonstrates that the resulting experience-based approximations of the derivative differ greatly near the endpoints of the approximation interval.

### EXAMPLE 3 - APPLICATION TO ADAPTIVE DYNAMIC PROGRAMMING

The application of approximation theory to the theory of optimal control arises through the approximation of the optimal value function, which is the solution to the HJB equation. Efficient methods for the approximation of the optimal value function are essential since an increase in dimension can lead to an exponential increase in the number of required basis functions necessary to achieve an accurate approximation, the so-called "curse of dimensionality."

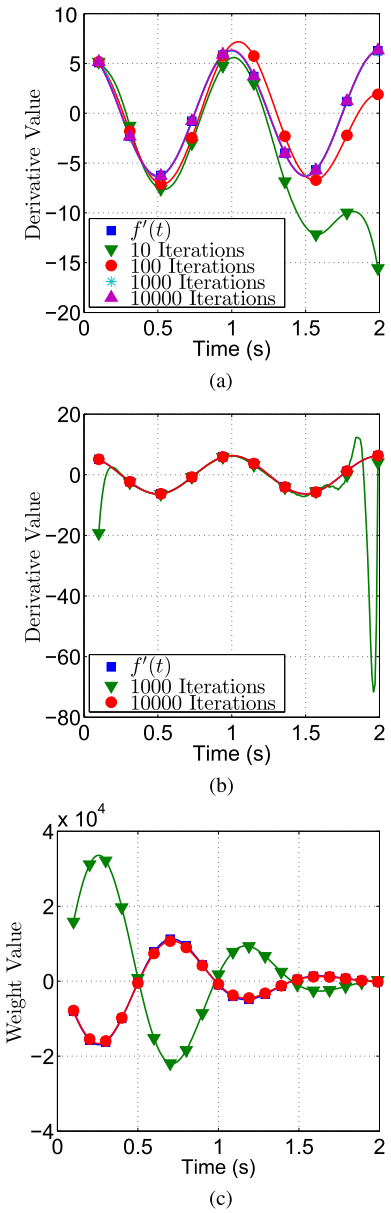


Fig. 2. Results of the numerical experiment demonstrating derivative estimation with StaF. (a) Approximation of the derivative of $f(t)$ with a varying number of iterations of the gradient descent algorithm per time step. Note that with increased iterations, the estimation improves. In particular, the curves representing 1000 and 10 000 time steps are indistinguishable from the derivative in this figure. (b) Experience-based approximation of the derivative utilizing the data collected from the 1000 and 10 000 iteration simulations in Fig. 2(a). (c) Interpolation of the weight samples via Gaussian radial basis functions for the simulation with 10 000 iterations of gradient descent. The samples correspond to the marks on the curves.

The optimal value function corresponds to the infinite horizon optimal regulator problem, where the cost function

$$J(x,u)=\int_0^\infty r(x(t),u(t))\,dt$$

is to be minimized subject to the dynamics

$$\dot{x}(t)=f(x(t))+g(x(t))u(t) \tag{11}$$

where $r(x,u)\triangleq x^TQx+u^TRu$, $x:\mathbb{R}_+\to\mathbb{R}^n$, $u:\mathbb{R}_+\to\mathbb{R}^m$, $Q\in\mathbb{R}^{n\times n}$, $R\in\mathbb{R}^{m\times m}$, with $Q$ and $R$ positive definite, $f:\mathbb{R}^n\to\mathbb{R}^n$, $g:\mathbb{R}^n\to\mathbb{R}^{n\times m}$. Moreover, $f$ and $g$ are assumed to be locally Lipschitz. The optimal value function $V^*:\mathbb{R}^n\to\mathbb{R}_+$ can be expressed as

$$V^*(x)\triangleq\inf_{u\in\mathcal{U}}\int_t^\infty r(\phi^u(\tau;t,x),u(\tau))\,d\tau$$

where $\mathcal{U}$ is the set of admissible controllers and $\phi^u(\tau;t,x)$ denotes the trajectory of the system in (11), evaluated at the time instance $\tau$, under the control signal $u$, with the initial condition $x\in\mathbb{R}^n$ and initial time $t\in\mathbb{R}_+$. When the optimal value function is continuously differentiable and an optimal controller, $u_*\in\mathcal{U}$ exists, the optimal value function is the unique solution to the HJB equation

$$0=\min_{u\in U}(r(x,u)+\nabla V(x)(f(x)+g(x)u) \tag{12}$$

where $U\subset\mathbb{R}^m$ is the action space.

Once the optimal value function is determined, the optimal policy takes the form

$$u^*(x)=-\frac{1}{2}R^{-1}g(x)^T\nabla V^*(x)^T \tag{13}$$

and the optimal controller is given by $u_*(t)=u^*(x(t))$, where $V^*$ denotes the unique solution to (12). In many applications, an approximation of the optimal controller is used in real time to yield autonomous behavior in a dynamic environment.

For some problems, such as the linear quadratic regulator (LQR) problem, the optimal value function takes a particular form which simplifies the choice of basis functions. In the case of LQR, the optimal value function is of the form $\sum_{i,j=1}^n w_{i,j}x_jx_i$ (see [39], [40]), so basis functions of the form $\sigma_{i,j}=x_jx_i$ will provide an accurate estimation of the optimal value function provided the weights, $w_{i,j}\in\mathbb{R}$, are tuned properly. However, in most cases, the form of the optimal value function is unknown, and generic basis functions have been proposed to parameterize the problem.

ADP replaces $V^*$ with a parametrization, $\hat{V}(x,W_c)=\sum_{i=1}^M w_{i,c}\sigma_i(x)$, with $W_c=(w_{1,c},\ldots,w_{M,c})\in\mathbb{R}^M$, and $u^*$ with a parametrization $\hat{u}(x,W_a)=-\frac{1}{2}R^{-1}g(x)^T\nabla_xV(x,W_a)^T$ where $W_a\in\mathbb{R}^M$. The actor and critic weights, $W_a$ and $W_c$, respectively, are tuned to minimize the residual Bellman error (BE)

$$\delta(x,W_a,W_c)=x^TQx+\hat{u}(x,W_a)^TR\hat{u}(x,W_a)+\nabla_x\hat{V}(x,W_c)(f(x)+g(x)\hat{u}(x,W_a))$$

overall $x$ in some compact set $D$ in real time. The BE is used to motivate weight update laws for $W_a$ and $W_c$ to achieve a real-time minimization.

Traditional ADP methods aim to achieve a global estimation of the value function. If generic basis functions such as sigmoids, RBFs, and polynomials are used, then approximation of a function over a larger volume typically requires a large number of basis functions. As a result, unless a smaller application-specific set of basis functions can be constructed using domain knowledge, ADP methods become computationally intractable, especially as the dimension of the state increases. Motivated by the observation that the optimal policy only requires information concerning the value function at the current state, the StaF approach to ADP establishes and maintains an approximation of the value function in a moving

neighborhood of the state. As a result, the computational effort can be concentrated to establish an estimation of the value function over an area that is immediately relevant for the generation of an optimal controller and not wasted to approximate the value function over areas of the state space where the state may never enter.

An immediate limitation of the local approach is that after successfully implementing a StaF-based online ADP controller, a global approximation of the value function cannot be extracted from the weight functions. To mitigate this limitation, an example is presented where the experience-based approximation method is utilized to construct a global estimate of the value function from the results of several simulations.

In this setting, the StaF problem becomes

$$\limsup_{t\to\infty} \sup_{x\in\overline{N_r(x)}} |\delta(x, W_a(t), W_c(t))| < \epsilon.$$

Section VIII-A provides more information concerning the application of the StaF method to ADP by presenting the results of companion papers [27], [28].

### *A. Details for the Application to Adaptive Dynamic Programming*

To demonstrate the effectiveness of the StaF technique in the context of optimal control, the simulation results from a companion paper are duplicated here. Analysis of StaF-based ADP appears in a preliminary form in [27] and in a detailed form in [28]. The dynamical system in question is of the form $\dot{x} = f(x) + g(x)u$, where $x = (x_1, x_2)^T \in \mathbb{R}^2$

$$f(x) = \begin{pmatrix} x_2 - x_1 \\ -\frac{1}{2}\, x_1 - \frac{1}{2}\, x_2(\cos(2x_1) + 2)^2 \end{pmatrix}$$

and

$$g(x) = \begin{pmatrix} 0 \\ \cos(2x_1) + 2 \end{pmatrix}. \tag{14}$$

Associated with this dynamical system is the cost functional

$$J(x, u) = \int_0^\infty (x^T(\tau)x(\tau) + u(\tau)^2)\, d\tau. \tag{15}$$

In the infinite horizon regulation problem, the goal is to determine an optimal control law $u^* : \mathbb{R}^2 \to \mathbb{R}$ (assuming an optimal control law exists) that satisfies

$$u^*(x_0) = \arg\min_{u\in\mathcal{U}} \int_0^\infty (x^T(\tau)x(\tau) + u(\tau)^2)\, d\tau$$

where $\mathcal{U}$ is the collection of admissible controllers and $x(0) = x_0$ inside the integrand. The optimal value function is given by

$$V(x_0) = \min_{u\in\mathcal{U}} \int_0^\infty (x^T(\tau)x(\tau) + u(\tau)^2)\, d\tau$$

when such a minimum exists, and the optimal value function satisfies the HJB equation (12). If $V^*$ satisfies the HJB equation and is also continuously differentiable, then it is the unique solution to (12). Furthermore, $u^*$ can be determined from $V^*$ by $u^*(x) = (1/2)g^T(x)\nabla V^*(x)^T$.

In most cases, the optimal value function cannot be determined analytically, and approximate solutions are used instead. However, for the system presented in this section, the optimal value function is known. In particular, for the infinite horizon optimal regulator problem with dynamics given by (14) with cost functional (15), the optimal value function is given by $V^*(x) = (1/2)x_1^2 + x_2^2$ and the associated optimal control law is given by $u^*(x) = -(\cos(2x_1) + 2)x_2$. More details can be found in [12].

In this example, the infinite horizon optimal regulator problem is solved in real time. The function $V^*$ is approximated by a function of the form

$$\hat{V}(x, \hat{W}_c) = \sum_{i=1}^{3} \hat{W}_{c,i}(\exp(x^T c_i(x)) - 1)$$

where $\hat{W}_c \in \mathbb{R}^3$ are weights to be adjusted in real time, and $c_i(x) = x + d_i(x)$ where

$$d_i = 0.7 \cdot \left(\frac{x^T x + 0.01}{1 + x^T x}\right) \begin{pmatrix} \sin((i-1)\cdot 2\pi/3) \\ \cos((i-1)\cdot 2\pi/3) \end{pmatrix} \tag{16}$$

for $i = 1, 2, 3$. Three centers were selected based on the heuristic that three centers can provide sufficient accuracy if the designated neighborhood is sufficiently small. Thereby, selecting three centers assists with the computational demand to establish the approximation. Furthermore, the quantity 0.01 was added to the numerator of (16) to assure that the centers remain distinct as $x^T x \to 0$. The approximation of the optimal control law is given by

$$\hat{u}(x, \hat{W}_a) = -\frac{1}{2}\, g^T(x)\nabla_x \hat{V}(x, \hat{W}_a)^T$$

where $\hat{W}_a \in \mathbb{R}^3$ are weights to be adjusted in real time. In the framework of ADP, the functions $V^*$ and $u^*$ are replaced by their approximations $\hat{V}$ and $\hat{u}$, respectively, in the HJB equation, yielding a residual nonzero error, called the BE. The goal is to minimize the BE by adjustments of the weights, $\hat{W}_a$ and $\hat{W}_c$. If the BE is identically zero after the adjustment of the weights, then the optimal value function and the approximation of the optimal value function coincide. For nonzero BE, the BE is used as a heuristic measure of the distance between $\hat{V}$ and $V^*$, as well as the distance between $\hat{u}$ and $u^*$. The weight update laws and subsequent convergence analysis can be found in [27].

The results of the numerical experiment are presented in Figs. 3 and 4. Fig. 3(a) indicates that the state is regulated to the origin when using the ADP algorithm combined with the StaF methodology. Fig. 3(b) shows that the weight vector $\hat{W}_a$ converged as well. In typical StaF implementations, the weights are not expected to converge. However, since the optimal control problem is a regulator problem, the state and the centers ultimately occupy a fixed neighborhood of the origin, and the weights converge to the ideal weights corresponding to a small neighborhood of the origin.

When the weights converge, it is expected that $\hat{W}_a$ and $\hat{W}_c$ converge to the same values. The convergence is demonstrated by comparing Fig. 3(b) and (c). The approximate controller and the optimal controller converge as well, as shown in Fig. 4(a), and the value function estimation error, given in Fig. 4(b), vanishes rapidly.

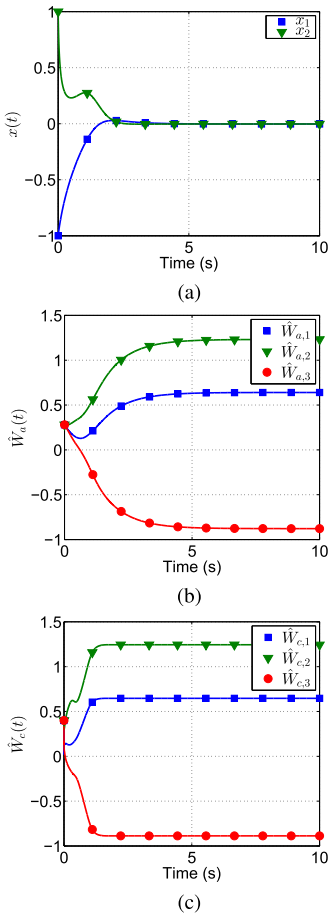


Fig. 3. State and weight trajectories demonstrating the convergence for the StaF ADP method [27]. (a) Trajectory of the state vector. (b) Trajectory of the actor weights. (c) Values of the critic weights.

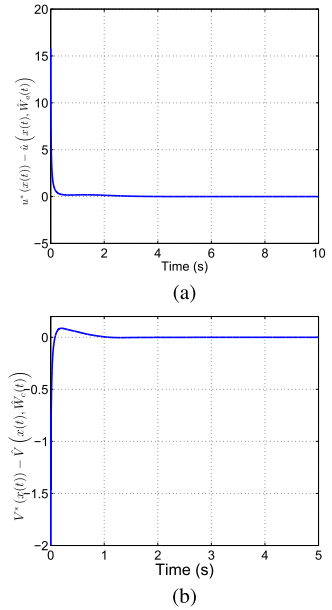


Fig. 4. Estimation errors demonstrating the convergence for the StaF ADP method [27]. (a) Error committed by approximate policy. (b) Error of the estimation of the value function at the current state.

### *B. Approximating the Value Function via Experience-Based Approximations*

The StaF approximation method is effective for producing a stabilizing, approximately optimal controller. For what follows, the same parameters are used as in Section VIII-A to execute the simulations, and the data collected from these simulations are used to determine an approximation of the value function over $D = [-1, 1]^2$.

The system selected in Section VIII-A is of the form (11) with $x : \mathbb{R}_+ \to \mathbb{R}^2$ given by

$$f(x) = \begin{pmatrix} -x_1 + x_2 \\ -\frac{1}{2}x_1 - \frac{1}{2}x_2(1 - (\cos(2x_2) + 2)^2) \end{pmatrix}$$

and

$$g(x) = \begin{pmatrix} 0 \\ \cos(2x_1) + 2 \end{pmatrix}.$$

The cost function was selected as

$$\int_0^\infty x^T(\tau)x(\tau) + u^2(\tau)\, d\tau.$$

The dynamical system was selected because the value function for this system is known, $V^*(x) = (1/2)x_1^2 + x_2^2$ (see [12]), thus the performance of the method developed in this paper can be evaluated.

For the purposes of data collection, the simulation was run 100 times, while moving $x_0$ along a circle of radius 2 centered at the origin. The radius was selected to give a circle larger than the domain of approximation. In this way, the weight update laws have an opportunity to approach the values of the ideal weight function.

As discussed in Section VII, a $10 \times 10$ grid was laid over $D$, and the approximate samples of the ideal weight function were recorded as the state passed over grid lines. The process resulted in 93 samples of the estimation of the ideal weight function. The data, $((x_{i1}, x_{i2}), (w_{i1}, w_{i2}, w_{i3}))_{i=1}^{93}$, were interpolated by kernels of the form $K_G(x, y) = \exp(-\|x-y\|^2/\mu)$ where $\mu = 100 \cdot \ln(10)$.

The StaF kernel functions in Section VIII-A were selected as $K(x, c_i(x)) = \exp(-x^T c_i(x)) - 1$, where $c_i(x) = x + d_i(x)$ with $d_i$ given by (16). Both the kernel functions and the value function satisfy $V(0) = 0$ and $K(0, c_i(x)) = 0$, so the approximation near the origin is expected to be very accurate.

Fig. 5(a)–(c) presents the results of the numerical experiment. The function approximate is presented in Fig. 5(a). Near the corners of the approximation domain, the approximate value function is much larger than the optimal value function, which takes a maximum value of 1.5. However, the plot of the error over the approximation region in Fig. 5(b) demonstrates that in a neighborhood near the origin, the value function approximation is precise. Fig. 5(c) shows that this region extends beyond a small neighborhood of the origin, and at $(-1, 0.6)$ and $(-1, -0.4)$ the accuracy of the approximation has been maintained.

Accuracy of the approximation is poor near the boundary of $D$ and the accuracy is especially bad at the corners. The large errors primarily stem from the amount of time the weight update laws had to perform their estimation. The corners of the domain are the closest points in the domain to the circle of radius 2 where the state was initialized. Therefore, this is where the weight update laws had the least amount of time to correct the approximation, and thus the approximation was still in its transient stage of poor performance when the weights were sampled.

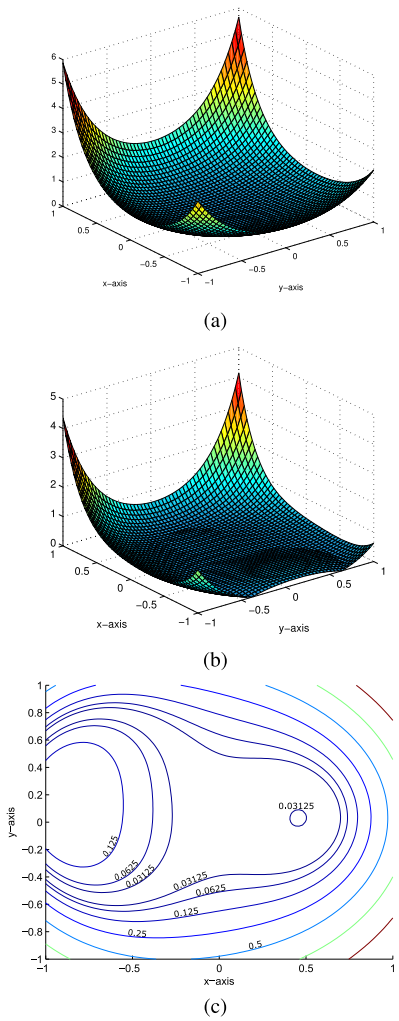


Fig. 5. Results of the experience-based approximation of the value function. (a) Plot of the approximation of the value function. (b) Error committed by the approximation. (c) Contour map of the error function.

On the other hand, in the interior of the region $D$, the accuracy of the approximation is improved. In this region, the weight update laws have had the most time to tune the values of the approximations of the ideal weight function, and thus they have achieved greater accuracy. Thus, when these values are used to approximate the value function, the performance of the approximation is improved.

Fig. 5(a)–(c) demonstrates both the strengths and the weaknesses of the StaF method. Having fast and accurate weight update laws are essential to guarantee an accurate estimation of the value function. When the approximate weights are sampled from areas where the weight update laws are given more time, the resulting approximation is more accurate. In principle, better choices of update laws and gains will yield better approximations when the experience-based approximation method is used.

## IX. Discussion Concerning Similarities Shared With Other Kernel Methods

As demonstrated in this paper, the StaF kernel method aims to establish a moving approximation using local samples of a function. Central to the StaF method is the establishment of continuous or smooth ideal weight functions so that weight update laws given either as differential equations or through gradient descent updates can be leveraged to establish and maintain an accurate estimation of a function. Other kernel methods have used state-dependent weight functions for approximation schemes in different ways. In particular, the MLS method achieves an approximation of a function by adjusting weights emphasizing different samples of a function based on the position of the state variable.

The weighted inner product for MLS takes the form

$$\langle f, g\rangle_{Q(x)} = \sum_{i=1}^{N} f(x_i)g(x_i)Q_i(x)$$

where $f$ and $g$ are functions in a Hilbert space, $Q(x) := (Q_1(x), \ldots, Q_N(x))$ is a vector of nonnegative functions, and $x_1, \ldots, x_n \in \mathbb{R}^n$ are predefined sample points [21]. This yields the state-dependent norm

$$\|f - g\|^2_{\mathrm{MLS},x} := \sum_{i=1}^{N} (f(x_i) - g(x_i))^2 \; Q_i(x).$$

Thus, the approximation of $f$ with the basis functions $u_j$ for $j = 1, \ldots, M$, given the samples $((x_i, f(x_i))_{i=1}^N$, takes the form $\hat{f}(x) = \sum_{i=1}^{M} b_i(x)u_j(x)$, where $(b_1(x), \ldots, b_M(x))$ minimizes

$$\min_{(a_i(x))_{i=1}^M} \left\| f - \sum_{j=1}^{M} a_j(x)u_j \right\|_{\mathrm{MLS},x}.$$

As with the StaF method, the weight functions $b_i$ may depend smoothly on the state variable, depending on the choice of $Q$. Also, like the StaF method, weight functions must be recalculated for each state. The main difference between the MLS method and the StaF method is that the MLS method is an approximation method that relies on a collection of a priori sampled points, whereas the StaF method is implemented through continuously sampling a function online in a fixed grid. In particular, the StaF method does not require a set of samples to be available before implementation, which is ideal in the setting of dynamical systems, where the ultimate trajectory of a system may be unknown. In addition, in contrast to the MLS method, the StaF approximation method utilizes basis functions that change with the state variable and the change in the ideal weight functions come from adjustments in the basis functions rather than the inner product.

Each of the two methods provides an approximation of a function using local information in different settings. Where the MLS method to be implemented in the StaF setting, a very large number of sample points would be required to be stored in advance, which would become computationally intractable.

However, the StaF method can be modified to work in an MLS setting, if we allow for the state-dependent centers to change discretely. Given a collection of samples $\{(x_i, f(x_i))\}_{i=1}^N$, let $C(x) = (c_1(x), \ldots, c_M(x))$ to be the collection of the closest $M$ points of $\{x_1, \ldots, x_N\}$ to the state variable $x$. Each of the centers $c_i(x)$ and the associated ideal weights are piecewise constant for $i = 1, \ldots, M$. Some care may be required if there were more than $M$ points that qualify

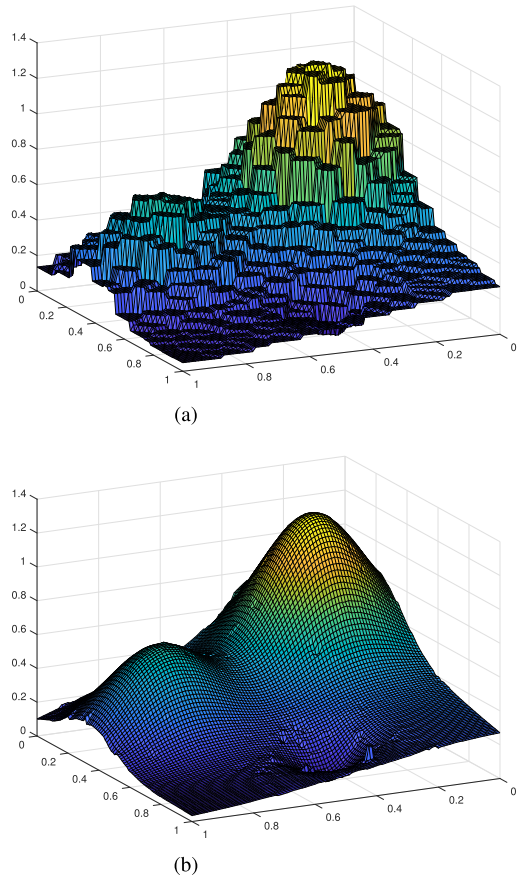

Fig. 6. StaF method is employed to approximate the Franke function given a collection of presampled points. This differs from the usage of the StaF method in the rest of this paper, where the centers changed continuously. Here, the centers are adjusted discretely to the nearest neighbors of the state variable. (a) Estimation of the Franke function obtained when using only one kernel function with the center determined by the nearest neighbor to the state. (b) Estimation of the Franke function obtained when using only 10 kernel functions with the centers determined by the 10 nearest neighbors to the state.

as being the closest to $x$, such as when $M = 1$, $x_1 = -1$, $x_2 = 1$, and $x = 0$. However, this issue is only expected to arise for isolated state values.

Fig. 6 shows the result of using the StaF method in the setting where the centers are selected as the nearest neighbors. The approximated function is the Franke function, a common benchmark for interpolation problems [21]. The sampled points are 256 points in $[0, 1]^2$ which were generated from the Halton sequence [21]. The particular kernel used in this example is the Gaussian kernel function $K(x, y) = \exp(-((\|x - y\|_2^2)/5))$. Fig. 6(a) presents the results when using one nearest neighbor, and Fig. 6(b) presents the results when using 10 nearest neighbors. Since the ideal weight functions are piecewise constant, the ideal weights were calculated at each state by inverting the Gram matrix for interpolation. Table I gives the rms errors for different choices of nearest neighbors. Note that the error is governed not only by the number of kernels employed but also by the spacing of the initial sample set $\{x_i\}_{i=1}^N$.

To establish a comparison with the MLS method, the indicator function of the ball with center $x$ and radius 0.1 was

TABLE I

HERE, THE ERRORS COMMITTED BY THE STAF METHOD IN A TYPICAL APPROXIMATION REGIME. HERE, THE CENTERS ARE DETERMINED BY NEAREST POINTS $\{x_i\}_{i=1}^N$ TO THE STATE $x$, WHICH RESULTS IN PIECEWISE CONSTANT CENTERS AND IDEAL WEIGHT FUNCTIONS. PRESENTED IS THE ESTIMATION ERROR CORRESPONDING TO DIFFERENT NUMBERS OF NEIGHBORS CHOSEN. TWO FACTORS DETERMINE THE RMS ERROR, THE SPACING OF THE POINTS $\{x_i\}_{i=1}^N$ AND THE NUMBER OF KERNELS EMPLOYED FOR THE APPROXIMATION (WHICH CORRESPONDS TO THE NUMBER OF NEAREST NEIGHBORS SELECTED)

| Number of Neighbors | RMS Error | Duration |
|---|---|---|
| 1 | 3.257E-2 | 0.9 sec |
| 5 | 5.113E-3 | 1.2 sec |
| 10 | 5.577E-3 | 1.6 sec |
| 15 | 1.269E-3 | 2.5 sec |

TABLE II

ERRORS COMMITTED WHEN USING MLS WITH A POLYNOMIAL BASIS. IT CAN BE SEEN THAT A CUBIC POLYNOMIAL BASIS (WHICH CONSISTS OF 10 FUNCTIONS) APPROACHES THE ORDER OF ACCURACY OF THE STAF METHOD WITH FIVE BASIS FUNCTIONS

| Support Radius | Polynomial Degree | RMS Error | Duration |
|---|---|---|---|
| 0.1 | 1 | 1.760E-2 | 8.6 sec |
| 0.1 | 2 | 6.995E-3 | 12.4 sec |
| 0.1 | 3 | 5.759E-3 | 18.8 sec |
| 0.1 | 4 | 4.511E-3 | 33.0 sec |
| 0.1 | 5 | 4.565E-3 | 50.2 sec |
| 0.2 | 1 | 5.763E-2 | 8.6 sec |
| 0.2 | 2 | 9.883E-3 | 11.7 sec |
| 0.2 | 3 | 1.031E-2 | 18.1 sec |
| 0.2 | 4 | 2.308E-3 | 32.4 sec |
| 0.2 | 5 | 1.953E-3 | 63.4 sec |

selected as

$$Q_i(x) = \begin{cases} 1, & \|x - x_i\|_2 \leq 0.1 \\ 0, & \|x - x_i\|_2 > 0.1 \end{cases}$$

as well as the basis of monomials of degree at most $M$, $\{x^p y^q\}_{p+q \leq M}$. The results of the MLS implementation are shown in Table II. The numerical accuracy of the two methods is comparable, with the StaF method performing slightly better. The accuracy of the MLS method may be improved with a better selection of $Q_i$. For instance, when $Q_i$ is the indicator of a ball with radius 0.2, the accuracy improves to $1.953E - 3$. However, the MLS method has a significantly longer computation time the larger the support of $Q_i$. The duration of the simulation using a support radius of 0.2 and a fifth-degree polynomial basis was 63.4 s to achieve a lower accuracy than the StaF method with 15 centers executed in 2.5 s. The longer computation time is likely due to the repeated computation of the weighted inner products.

## X. CONCLUSION

A new StaF kernel method is introduced in this paper for the purpose of function approximation. The development in this paper establishes that by using the StaF method a local approximation of a function can be maintained in real time as a state moves through a compact domain. Heuristically,

much fewer kernel functions are required in comparison to more traditional function approximation schemes since the approximation is maintained in a smaller region. For the exponential kernels, a new theorem in this paper established that an explicit bound on the number of kernel functions required can be calculated. Two applications of this methodology were presented. In Section VI, a "gradient chase" algorithm was developed. There, it was seen that a function may be well approximated provided that the algorithm was applied with a high enough frequency. Simulations results provided in Section VIII demonstrated the performance of the gradient chase algorithm, the gradient chase algorithm was also implemented for a derivative estimation problem in Section VIII, and an application to ADP is provided in Section VIII-A for an infinite horizon optimal regulation problem.

The strength of the StaF methodology is the reduction of the computational requirements for real-time implementation of function approximation, through the reduction in the number of basis functions. The reduction in the number of basis functions was demonstrated in Section VIII-A, where only three basis functions were required to achieve a stabilizing approximate optimal controller for a 2-D system. However, since the StaF method aims at maintaining an accurate local approximation of the value function only in a local neighborhood of the current system state, the StaF kernel method lacks memory, in the sense that the information about the ideal weights over a region of interest is lost when the state leaves the region of interest. Thus, unlike existing techniques, the StaF method generates an approximation that is valid only in a local region.

A so-called experience-based approximation method is presented to address this limitation of the StaF method. An example in Section VIII is presented where the ideal weight functions were approximated to produce an approximation of the derivative of $\sin(2\pi t)$ over a compact interval. In Section VIII-B, the method was applied an ADP problem, where moderate success was achieved in the approximation of the optimal value function. The experience-based approximation is more suited to applications in settings such as ADP, where the weight update laws govern the approximation, and direct sampling of the value function is not available.

## REFERENCES


[1] A. Christmann and I. Steinwart, "Universal kernels on non-standard input spaces," in *Proc. Adv. Neural Inf. Process.*, 2010, pp. 406–414.

[2] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006.

[3] A. K. Noor and J. M. Peters, "Reduced basis technique for nonlinear analysis of structures," *Amer. Inst. Aeronaut. Astronaut. J.*, vol. 18, no. 2, pp. 455–462, 1980.

[4] E. Balmès, "Parametric families of reduced finite element models. Theory and applications," *Mech. Syst. Signal Process.*, vol. 10, no. 4, pp. 381–394, 1996.

[5] C. Prud'homme *et al.*, "Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods," *AMSE J. Fluids Eng.*, vol. 124, no. 1, pp. 70–80, 2001.

[6] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[7] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and E. W. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.

[8] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Netw.*, vol. 22, nos. 5–6, pp. 851–860, 2009.

[9] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.

[10] P. Mehta and S. Meyn, "Q-learning and Pontryagin's minimum principle," in *Proc. IEEE Conf. Decision Control*, Dec. 2009, pp. 3598–3605.

[11] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

[12] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[13] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[14] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[15] H. Zhang, D. Liu, Y. Luo, and D. Wang, "Adaptive dynamic programming for control algorithms and stability," in *Communications and Control Engineering*. London, U.K.: Springer-Verlag, 2013.

[16] R. A. DeVore, "Nonlinear approximation," *Acta Numer.*, vol. 7, pp. 51–150, Jan. 1998.

[17] M. Gaggero, G. Gnecco, and M. Sanguineti, "Dynamic programming and value-function approximation in sequential decision problems: Error analysis and numerical results," *J. Optim. Theory Appl.*, vol. 156, no. 2, pp. 380–416, 2013.

[18] M. Gaggero, G. Gnecco, and M. Sanguineti, "Approximate dynamic programming for stochastic n-stage optimization with application to optimal consumption under uncertainty," *Comput. Optim. Appl.*, vol. 58, no. 1, pp. 31–85, 2014.

[19] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and extended Ritz method for the solution of functional optimization problems," *J. Optim. Theory Appl.*, vol. 112, no. 2, pp. 403–440, 2002.

[20] G. E. Fasshauer, *Matrix-Free Multilevel Moving Least-Squares Methods, in Approximation Theory X: Splines* in *Wavelets, and Applications*. Nashville, TN, USA: Vanderbilt Univ. Press, 2002, pp. 271–281.

[21] G. E. Fasshauer, "Meshfree approximation methods with MATLAB," in *Interdisciplinary Mathematical Sciences*, vol. 6. Hackensack, NJ, USA: World Scientific, 2007.

[22] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *Math. Comput.*, vol. 37, no. 155, pp. 141–158, 1981.

[23] D. Levin, "The approximation power of moving least-squares," *Math. Comput. Amer. Math. Soc.*, vol. 67, no. 224, pp. 1517–1531, 1998.

[24] S. Li and W. K. Liu, "Meshfree and particle methods and their applications," *Appl. Mech. Rev.*, vol. 55, no. 1, pp. 1–34, 2002.

[25] W.-K. Liu, S. Li, and T. Belytschko, "Moving least-square reproducing kernel methods (I) methodology and convergence," *Comput. Methods Appl. Mech. Eng.*, vol. 143, nos. 1–2, pp. 113–154, 1997.

[26] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. 23rd ACM Nat. Conf. (ACM)*, 1968, pp. 517–524.

[27] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "State following (StaF) kernel functions for function approximation Part II: Adaptive dynamic programming," in *Proc. Amer. Control Conf.*, 2015, pp. 521–526.

[28] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.

[29] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "State following (StaF) kernel functions for function approximation Part I: Theory and motivation," in *Proc. Amer. Control Conf.*, 2015, pp. 1217–1222.

[30] I. Steinwart and A. Christmann, *Support Vector Machines* (Information Science and Statistics). New York, NY, USA: Springer, 2008.

[31] K. Zhu, *Analysis on Fock Spaces* (Graduate Texts in Mathematics), vol. 263. New York, NY, USA: Springer, 2012.

[32] A. Pinkus, "Strictly positive definite functions on a real inner product space," *Adv. Comput. Math.*, vol. 20, no. 4, pp. 263–271, 2004.
[33] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[34] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.
[35] G. K. Pedersen, *Analysis Now* (Graduate Texts in Mathematics), vol. 118. New York, NY, USA: Springer-Verlag, 1989.
[36] Z. Wu, X. Sun, and L. Ma, "Sampling scattered data with bernstein polynomials: Stochastic and deterministic error estimates," *Adv. Comput. Math.*, vol. 38, no. 1, pp. 187–205, 2011.
[37] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, nos. 3–4, pp. 231–357, 2015.
[38] O. Davydov and R. Schaback, "Error bounds for kernel-based numerical differentiation," *Numerische Mathematik*, vol. 132, no. 2, pp. 243–269, 2013.
[39] D. Kirk, *Optimal Control Theory: An Introduction*. Mineola, NY, USA: Dover, 2004.
[40] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

**Joel A. Rosenfeld** received the Ph.D. degree from the Department of Mathematics, University of Florida, Gainesville, FL, USA, in 2013.

Subsequently, he spent four years as a Post-Doctoral Researcher with the Department of Mechanical and Aerospace Engineering, University of Florida, under the supervision of Dr. W. E. Dixon, where he was involved in problems relating to machine learning an control theory, and later he spent a year as a Post-Doctoral Researcher with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA, under the supervision of Dr. T. T. Johnson. He is currently a Senior Research Scientist Engineer with the Department of Electrical Engineering and Computer Science, Vanderbilt University, where he is studying machine learning methods as they pertain to formal methods in computing.

**Rushikesh Kamalapurkar** (M'13–SM'17) received the M.S. and Ph.D. degrees from the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA, in 2011 and 2014, respectively.

After working for a year as a Post-Doctoral Researcher with Dr. W. E. Dixon, he was appointed as the 2015–2016 MAE Post-Doctoral Teaching Fellow. In 2016, he joined the School of Mechanical and Aerospace Engineering, Oklahoma State University at Stillwater, Stillwater, OK, USA, as an Assistant Professor. He has published a book, multiple book chapters, over 20 peer-reviewed journal papers and over 20 peer-reviewed conference papers. His current research interests include intelligent, learning-based optimal control of uncertain nonlinear dynamical systems.

Dr. Kamalapurkar's work has been recognized by the 2014 University of Florida Department of Mechanical and Aerospace Engineering Outstanding Graduate Research Award and the 2015 University of Florida Department of Mechanical and Aerospace Engineering Best Dissertation Award.

**Warren E. Dixon** (M'94–F'16) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA, in 2000.

He was a Research Staff Member and a Eugene P. Wigner Fellow at Oak Ridge National Laboratory, Oak Ridge, TN, USA, until 2004, when he joined the Mechanical and Aerospace Engineering Department, University of Florida, Gainesville, FL, USA. His current research interests include the development and application of Lyapunov-based control techniques for uncertain nonlinear systems.

Dr. Dixon is an ASME Fellow. His work has been recognized by a number of best paper, early career, and mentoring awards. He received the Air Force Commander's Public Service Award in 2016 for his contributions to the U.S. Air Force Science Advisory Board. In addition to actively or formerly serving as an Associate Editor for various journals, such as the *Journal of Dynamic Systems, Measurement and Control* (ASME Journal), *Automatica*, the IEEE CONTROL SYSTEMS, the IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS: PART B CYBERNETICS, and the *International Journal of Robust and Nonlinear Control*, his service contributions include previous roles as an IEEE Control Systems Society (CSS) Distinguished Lecturer and Director of Operations for the Executive Committee of the IEEE CSS Board of Governors from 2012 to 2015.