2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

A Reproducing Kernel Hilbert Space Approach for the Online Update of Radial Bases in Neuro-Adaptive Control

Hassan A. Kingravi, Girish Chowdhary, Patricio A. Vela, and Eric N. Johnson

Abstract— **Classical gradient based adaptive laws in model reference adaptive control for uncertain nonlinear dynamical systems with a Radial Basis Function (RBF) neural networks adaptive element do not guarantee that the network weights stay bounded in a compact neighborhood of the ideal weights without Persistently Exciting (PE) system signals or a-priori known bounds on ideal weights. Recent work has shown, however, that an adaptive controller using specifically recorded data concurrently with instantaneous data can guarantee such boundedness without requiring PE signals. However, in this work, the assumption has been that the RBF network centers are fixed, which requires some domain knowledge of the uncertainty. We employ a Reproducing Kernel Hilbert Space theory motivated online algorithm for updating the RBF centers to remove this assumption. Along with showing the boundedness of the resulting neuro-adaptive controller, a connection is also made between PE signals and kernel methods. Simulation results show improved performance.**

I. INTRODUCTION

Model Reference Adaptive Control (MRAC) has been widely studied for broad classes of uncertain nonlinear dynamical systems with significant modeling uncertainties (see [1, 8, 16, 22] and the references therein). In MRAC, the system uncertainty is approximated using a weighted combination of basis functions, with the numerical values of the weights adapted online to minimize the tracking error. When the structure of the uncertainty is unknown, a neuro-adaptive approach is often employed, in which a Neural Network (NN), with its weights adapted online, is used to capture the uncertainty. A popular example of such NN is the Radial Basis Function (RBF) NN for which the universal approximation property is known to hold [18]. Classical MRAC methods however, do not guarantee that the weights of the NN approach and stay bounded around their ideal values without requiring a condition on Persistency of Excitation (PE) in the system states [2]. These conditions are overly restrictive and often infeasible to implement or monitor online. Hence, authors have introduced various modifications to the adaptive law to ensure that the weights stay bounded around an *a-priori* determined value (usually set to 0). Examples of these include Ioannou's σ-mod [9], Narendra's e-mod, and the use of a projection operator to bound the weights [22]. However, recent work in concurrent learning [5] has shown that if carefully selected and recorded data is used concurrently with current data for adaptation, then the stored information can be used to guarantee that the weights remain bounded within a compact neighborhood of the ideal weights, without requiring PE [6].

In order to approximate the uncertainty, a set of RBF centers must be chosen over the domain of the uncertainty. In previous work in neuro-adaptive control, it is either assumed that the centers for the RBF network are fixed [6, 20, 23], or that the centers are moved to minimize the tracking error e [14]. In both these cases, the system designer needs some domain knowledge about the uncertainty to determine how the centers should be selected. In this work, we use methods from the theory of Reproducing Kernel Hilbert Spaces (RKHSs) to remove this assumption. The algorithm we propose, called Budgeted Kernel Restructuring (BKR) is motivated by a connection between PE signals and RKHSs. If it is augmented with previous work in concurrent learning, the resulting algorithm (BKR-CL) selects the most appropriate set of centers during operation to allow us to control the system effectively even if all the RBF centers are initialized to the same value in the state space (for example $0 \in \mathbb{R}^n$). In addition to removing the assumption on fixed RBF centers, we also show that given a fixed budget (maximum number of allowable centers), the presented method outperforms existing methods that uniformly distribute the centers over an expected domain.

The organization of this paper is as follows; in Section II, we outline some preliminaries, including the definition of PE and the relevant ideas from RKHS theory. Section III poses the MRAC problem for nonlinear systems, along with the concurrent learning method. Section IV shows a connection between PE and RKHS theory, and outlines a kernel linear independence method for selecting the centers that utilizes this connection. Section V establishes the boundedness of the weights for the centers using Lyapunov analysis. Section VI presents the results of an exemplary simulation study. Section VII concludes the paper.

II. PRELIMINARIES

Partly due to the success of Support Vector Machines, there has been great interest in recent years in kernel methods, a class of algorithms that exploit the properties of Reproducing Kernel Hilbert Spaces [3]. A *kernel function* on $\mathbb{R}^n$ is any continuous, symmetric positive-semidefinite function of the form $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Mercer's Theorem implies that there exists some Hilbert space $\mathcal{H}$ (of functions)

H. Kingravi and Asst. Prof. P. A. Vela are with the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta GA. G. Chowdhary and Assoc. Prof. E. N. Johnson are with the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology, Atlanta GA, kingravi@gatech.edu, pvela@gatech.edu, Girish.Chowdhary@gatech.edu, Eric.Johnson@ae.gatech.edu

978-1-61284-799-3/11/$26.00 ©2011 IEEE

and a mapping $\psi : \mathbb{R}^n \to \mathcal{H}$ s.t.

$$k(x, y) = \langle \psi(x), \psi(y) \rangle_{\mathcal{H}}, \tag{1}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is an inner product on $\mathcal{H}$. For a given kernel function, the mapping $\psi(x) \in \mathcal{H}$ does not have to be unique, and is often unknown. However, since ψ is implicit, it does not need to be known for most machine learning algorithms, which exploit the nonlinearity of the mapping to create nonlinear algorithms from linear ones [7]. For the purpose of this paper, it is sufficient to note that the Gaussian function used in RBF networks is an example of a bounded kernel function, which generates an infinite dimensional RKHS $\mathcal{H}$ (see for example [19]). In this work, the Gaussian function is used in the RBF network. Given a point $x \in \mathbb{R}^n$, the mapping $\psi(x) \to \mathcal{H}$ can be thought of as a point in the Hilbert space $\mathcal{H}$; however, in $\mathbb{R}^n$, it is a Gaussian function centered at x, and is thus given by $k_x(y) = e^{(-\|x-y\|^2/2\mu^2)}$, where μ is the bandwidth. An alternative description is $k(x, \cdot) = \langle \psi(x), \psi(\cdot) \rangle_{\mathcal{H}}$, to emphasize that the function is an inner product in $\mathcal{H}$.

Fixing a dataset $C = \{c_1, \ldots, c_n\}$, where $c_i \in \mathbb{R}^n$, we define by

$$\mathcal{F}_C := \left\{ \sum_{i=1}^{n} \alpha_i k(c_i, \cdot) : \alpha_i \in \mathbb{R} \right\} \tag{2}$$

the *linear subspace generated by C in $\mathcal{H}$*. Note that this is in fact a class of functions, and that any RBF network with a fixed bandwidth can be defined as an element in this subspace. To see this, let $\sigma(x) = [k(x, c_1) \ldots, k(x, c_n)]^T$, and let $W = [w_1, \ldots, w_n]$ where $w_i \in \mathbb{R}$. Then $W^T \sigma(x)$ is the output of a standard RBF network. In the machine learning literature, $\sigma(x)$ is sometimes known as the *empirical kernel map*. Two different datasets C_1 and C_2 generate two different families of functions F_{C_1} and F_{C_2}, and two different empirical kernel maps $\sigma_{C_1}(x)$ and $\sigma_{C_2}(x)$.

In this work we will use Tao's definition of a PE signal [22]:

Definition 1 A bounded vector signal $x(t)$ is persistently exciting if for all $t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that

$$\int_t^{t+T} \Phi(\tau)\Phi^T(\tau) d\tau \geq \gamma I. \tag{3}$$

III. Model Reference Adaptive Control and Concurrent Learning

This section outlines the formulation of Model Reference Adaptive Control using approximate model inversion [5, 10, 12–14]. Let $D_x \in \mathbb{R}^n$ be compact, let $x(t) \in D_x$ be the known state vector, let $\delta \in \mathbb{R}^k$ denote the control input, and consider the following dynamical system:

$$\dot{x} = f(x(t), \delta(t)), \tag{4}$$

where the function f is assumed to be continuously differentiable in $x \in D_x$, and control input δ is assumed to be bounded and piecewise continuous. The conditions for the existence and the uniqueness of the solution to (4) are assumed to be met. Since the exact model (4) is usually not available or not invertible, an approximate inversion model $\hat{f}(x, \delta)$ is introduced which can be used to determine the control input δ:

$$\delta = \hat{f}^{-1}(x, \nu), \tag{5}$$

where ν is the pseudo control input, which represents the desired model output $\dot{x}$ and is expected to be approximately achieved by δ. Hence, the pseudo control input is the output of the approximate inversion model:

$$\nu = \hat{f}(x, \delta). \tag{6}$$

This approximation results in a model error of the form:

$$\dot{x} = \nu(x, \delta) + \Delta(x, \delta) \tag{7}$$

where the model error $\Delta : \mathbb{R}^{n+k} \to \mathbb{R}^n$ is given by:

$$\Delta(x, \delta) = f(x, \delta) - \hat{f}(x, \delta). \tag{8}$$

A reference model can be chosen to characterizes the desired response of the system:

$$\dot{x}_{rm}(t) = f_{rm}(x_{rm}(t), r(t)), \tag{9}$$

where $f_{rm}(x_{rm}(t), r(t))$ denotes the reference model dynamics which are assumed to be continuously differentiable in x for all $x \in D_x \subset \mathbb{R}^n$. The command $r(t)$ is assumed to be bounded and piecewise continuous. Furthermore, it is assumed that all requirements for guaranteeing the existence of a unique solution to (9) are satisfied. It is also assumed that the reference model states remain bounded for a bounded reference input.

A tracking control law consisting of a linear feedback part $u_{pd} = Kx$, a linear feedforward part $u_{crm} = \dot{x}_{rm}$, and an adaptive part $u_{ad}(x)$ can be chosen to have the following form [4, 6, 11]:

$$u = u_{crm} + u_{pd} - u_{ad}. \tag{10}$$

Define the tracking error e as $e(t) = x_{rm}(t) - x(t)$, then, letting $A = -K$ the tracking error dynamics are found to be:

$$\dot{e} = Ae + [u_{ad}(x, \delta) - \Delta(x, \delta)]. \tag{11}$$

The baseline full state feedback controller $u_{pd} = Kx$ is assumed to be designed such that A is a Hurwitz matrix. Hence for any positive definite matrix $Q \in \mathbb{R}^{n \times n}$, a positive definite solution $P \in \mathbb{R}^{n \times n}$ exists to the Lyapunov equation:

$$A^T P + PA + Q = 0. \tag{12}$$

Let $\bar{x} = [x, \delta] \in \mathbb{R}^{n+k}$. Generally, two cases for characterizing the uncertainty $\Delta(\bar{x})$ are considered. In *structured uncertainty*, the mapping $\Phi(\bar{x})$ is known, whereas in *unstructured uncertainty*, it is unknown. We focus on the latter in this paper.

Assume that it is only known that the uncertainty $\Delta(\bar{x})$ is continuous and defined over a compact domain $D \subset \mathbb{R}^{n+k}$. A Radial Basis Function (RBF) Neural Network (NN) can be used as the adaptive element. In this case the adaptive element takes the following form

$$u_{ad}(\bar{x}) = W^T \sigma(\bar{x}). \tag{13}$$

where $W \in \mathbb{R}^{n \times l}$ and $\sigma(\bar{x}) = [1, \sigma_2(\bar{x}), \sigma_3(\bar{x}), ..., \sigma_l(\bar{x})]^T$ is a vector of known radial basis functions. For $i = 2, 3..., l$ let c_i denote the RBF centroid and μ_i denote the RBF width. Then for each i, the radial basis functions are given as

$$\sigma_i(x) = e^{-\|\bar{x}-c_i\|^2/2\mu^2}, \tag{14}$$

which can also be written as $k(c_i, \cdot)$ according to the notation introduced earlier in the paper. Appealing to the universal approximation property of RBF NN [18, 21], we have that given a fixed number of radial basis functions l there exists ideal weights $W^* \in \mathbb{R}^{n \times l}$ and a real number $\tilde{\epsilon}(\bar{x})$ such that the following approximation holds for all $x \in D$ where D is compact:

$$\Delta(x) = {W^*}^T \sigma(\bar{x}) + \tilde{\epsilon}(\bar{x}), \tag{15}$$

and $\bar{\epsilon} = \sup_{\bar{x} \in D} \|\tilde{\epsilon}(\bar{x})\|$ can be made arbitrarily small when a sufficient number of radial basis functions are used.

A commonly used update law, which will be referred to as the baseline adaptive law, is given as [1, 16, 22]

$$\dot{\hat{W}} = -\Gamma_W \sigma(\bar{x}) e^T PB, \tag{16}$$

which only guarantees that the weights approach their ideal values (W^*) if and only if the signal $\sigma(\bar{x})$ is PE. In absence of PE, without additional modifications such as σ mod[8], e mod[15], or without projection based adaptation [22], this adaptive law does not guarantee that the weights W remain bounded.

The work in [5] shows that if specifically selected recorded data is used concurrently with instantaneous measurements, then the weights approach and stay bounded in a compact neighborhood of the ideal weights subject to a sufficient condition on the linear independence of the recorded data; PE is not needed. This is captured in the following theorem:

Theorem 1 Consider the system in (4), the control law of (10), $\bar{x}(0) \in D$ where D is compact, and the case of unstructured uncertainty. For the j^{th} recorded data point let $\epsilon_j(t) = W^T(t)\sigma(\bar{x}_j) - \Delta(\bar{x}_j)$. Also, let p be the number of recorded data points $\sigma(\bar{x}_j)$ in the matrix $Z = [\sigma(\bar{x}_1), ..., \sigma(\bar{x}_p)]$, such that $\mathrm{rank}(Z) = l$. Then, the following weight update law

$$\dot{W} = -\Gamma_W \sigma(\bar{x}) e^T PB - \Gamma_W \sum_{j=1}^{p} \sigma(\bar{x}_j) \epsilon_j^T, \tag{17}$$

renders the tracking error e and the RBF NN weight errors $\tilde{W}$ uniformly ultimately bounded. Furthermore, the adaptive weights $W(t)$ will approach and remain bounded in a compact neighborhood of the ideal weights W^*.

IV. The Kernel Linear Independence Method

A. PE Signals and the RKHS

We consider a general system

$$\dot{x}(t) = f(x) \tag{18}$$

In this section, we leverage RKHS theory to relate PE of $x(t)$ to PE of $\sigma(x(t))$. To make a connection to kernel methods, let $G = \sigma(x(t))\sigma(x(t))^T$. Then

$$\begin{aligned} G &= \begin{pmatrix} k(x, c_1) \\ \vdots \\ k(x, c_n) \end{pmatrix} \begin{pmatrix} k(x, c_1) & \cdots & k(x, c_n) \end{pmatrix} \\ &= \begin{pmatrix} \langle \psi_{x,c_1} \rangle \langle \psi_{x,c_1} \rangle & \cdots & \langle \psi_{x,c_1} \rangle \langle \psi_{x,c_n} \rangle \\ \vdots & \ddots & \vdots \\ \langle \psi_{x,c_n} \rangle \langle \psi_{x,c_1} \rangle & \cdots & \langle \psi_{x,c_n} \rangle \langle \psi_{x,c_n} \rangle \end{pmatrix} \end{aligned}$$

where $\langle \psi_{x,c_i} \rangle$ is shorthand for $\langle \psi(x), \psi(c_i) \rangle$. A matrix G is positive definite if and only if $v^T G v > 0 \ \forall v \in \mathbb{R}^n$. In the above, this translates to

$$\begin{aligned} v^T G v &= \sum_{i,j=1}^{n} v_i v_j G_{i,j} \\ &= \sum_{i,j=1}^{n} v_i v_j \langle \psi(x), \psi(c_i) \rangle \langle \psi(x), \psi(c_j) \rangle \\ &= \left\langle \psi(x), \sum_{i=1}^{n} v_i \psi(c_i) \right\rangle^2 . \end{aligned}$$

From basic results on the Gaussian kernel, it is known that if $c_i \neq c_j$, then $\psi(c_i)$ and $\psi(c_j)$ are linearly independent in $\mathcal{H}$. Further, if the trajectory $x(t) \in \mathbb{R}^n$ is bounded for all time, then $k(x(t), c) \neq 0$. From this, it follows trivially that the signal $\int_t^{t+T} G(\tau) d\tau$ is bounded. Further, the next two theorems follow immediately.

Theorem 2 Suppose $x(t)$ evolves in the state space according to Equation (18). Then if there exists some time $t_f \in \mathbb{R}_+$ s.t. the mapping $\psi(x(t)) \to \mathcal{H}$ for $t > t_f$ is orthogonal to the linear subspace $\mathcal{F}_C \subset \mathcal{H}$ for all time, the signal $\sigma(x(t))$ is not persistently exciting.

Theorem 3 Suppose $x(t)$ evolves in the state space according to Equation (18). If there exists some state $x_f \in \mathbb{R}^n$ and some $t_f \in \mathbb{R}_+$ s.t. $x(t) = x_f \ \forall t > t_f$, $\sigma(x(t))$ is not persistently exciting.

Thus PE of $\sigma(x(t))$ follows only if neither of the above conditions are met. Figure 1 shows an example of the mapping ψ in $\mathcal{H}$. Figure 2 depicts a geometric description of a non PE signal in the Hilbert space. This implies that $\psi(x(t))$ starts becoming orthogonal to the centers $C = \{c_1, \ldots, c_n\}$ in $\mathcal{H}$ if $x(t)$ is extremely far away from them in $\mathbb{R}^n$. Therefore, even though orthogonality is desired in the state space $\mathbb{R}^n$ for guaranteeing PE, orthogonality in $\mathcal{H}$ is detrimental to PE of $\sigma(x(t))$.

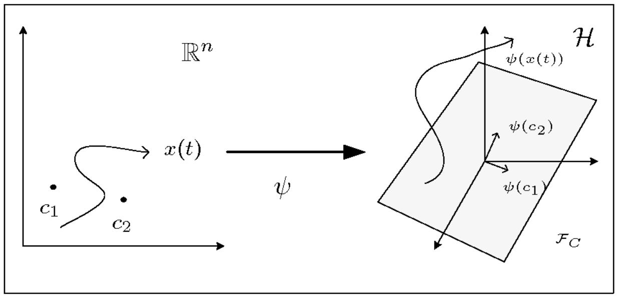


Fig. 1. An example Hilbert space mapping. The trajectory $x(t)$ evolving in the state space $\mathbb{R}^n$ becomes $\psi(x(t))$ in $\mathcal{H}$. If c_1 and c_2 are the centers for the RBFs, then they generate the linear subspace $\mathcal{F}_C \subset \mathcal{H}$, which is a family of linearly parameterized functions, via the mappings $\psi(c_1)$ and $\psi(c_2)$.

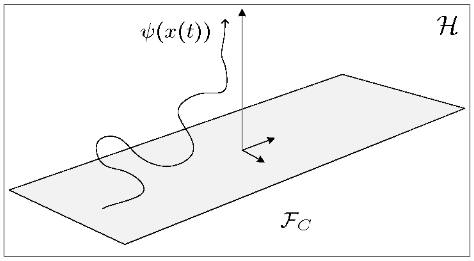


Fig. 2. An example of a non PE signal: Any trajectory $x(t) \in \mathbb{R}^n$ that induces a trajectory $\psi(x(t)) \in \mathcal{H}$ that is orthogonal to $\mathcal{F}_C$ after some time t_f renders $\sigma(x(t))$ non-PE.

The work in [5] shows that if states are recorded in a history stack, the law in (17) suffices to ensure convergence to a compact neighborhood of the ideal weight vector W^*. We can reinterpret the concurrent learning gradient descent in our RKHS framework. The law is given by

$$\dot{W} = -\Gamma_W \sigma(\bar{x}) e^T P B - \Gamma_W \sum_{j=1}^{p} \sigma(\bar{x}_j) \epsilon_j^T, \qquad (19)$$

where $\epsilon_j(t) \to \tilde{\epsilon}$ as $W(t) \to W^*$. By the above analysis, $\sigma(\bar{x}_j)$ is the projection of $\psi(\bar{x}_j)$ onto $\mathcal{F}_C$. If $\psi(x(t))$ is orthogonal to $\mathcal{F}_C$ in $\mathcal{H}$, the first term in Equation (19) (which is also the baseline adaptive law of (16) vanishes, causing the evolution of weights to stop with respect to the tracking error e. The condition that $\sum_p \sigma(\bar{x}_j)\sigma(\bar{x}_j)^T$ has the same rank as the number of centers in the RBF network is the statement that one has a collection of vectors $\{\bar{x}_j\}_{j=1}^p$ where the projections of $\{\psi(\bar{x}_j)\}_{j=1}^p$ onto $\mathcal{F}_C$ do not vanish, which is the same as saying that the centers stay close to $x(t)$ in the state space. At the same time, we also do not want to throw out all of the earlier centers either, especially if the system is expected to revisit that region in the state space again. The next section outlines a strategy that attempts to address both of these concerns.

B. Linear Independence

We use the scheme introduced in [17] to ensure that the RBF centers reflect the current domain of operation. At any point in time, our algorithm maintains a 'dictionary' of centers $C_l = \{c_i\}_{i=1}^l$, where l is the current size of the dictionary, and N_D is the upper limit on the number of points (the budget). To test whether a new center c_{l+1} should be inserted into the dictionary, we check whether it can or cannot be approximated in $\mathcal{H}$ by the current set of centers. This test is performed using

$$\gamma = \left\| \sum_{i=1}^{l} a_i \psi(c_i) - \psi(c_{l+1}) \right\|_{\mathcal{H}}^2, \qquad (20)$$

where the a_i denote the coefficients of the linear independence. Unraveling the above equation in terms of (1) shows that the coefficients a_i can be determined by minimizing γ, which yields the optimal coefficient vector $\hat{a}_l = K_l^{-1}\hat{k}_l$, where $K_l = k(C_l, C_l)$ is the kernel matrix (a kernel distance matrix evaluated pairwise between points) for the dictionary dataset C_l, and where $\hat{k}_l = k(C_l, c_{l+1})$ is the kernel vector. Note that this implies that knowledge of the mapping ψ is unnecessary. After substituting the optimal value $\hat{a}_l$ into Equation (20), we get

$$\gamma = K(c_{l+1}, c_{l+1}) - \hat{k}_l^T \hat{a}_l. \qquad (21)$$

The algorithm is summarized in Algorithm 1. For more details see [17]; in particular, an efficient way to implement the updates is given there, which we do not get into here. Note that due to the nature of the linear independence test, every time a state $x(t)$ is encountered that cannot be approximated within tolerance η by the current dictionary C_l, it is added to the dictionary. Further, since the dictionary is designed to keep the most varied basis possible, this is the best one can do with a greedy strategy on a budget. Therefore, in the final algorithm, all the centers are initialized to 0, and we periodically run the linear independence test to check whether or not to add a new center to the RBF network. This is fundamentally different from update laws of the kind given in [14], which attempt to move the centers to minimize the tracking error e. Since such update laws are rank-1, if all the centers are initialized to 0, they will all

Algorithm 1 Kernel Linear Independence Test

```
**Input:** New point $c_{l+1}, \eta, N_D$.
Compute
  $\hat{a}_l = K_l^{-1}\hat{k}_l$
  $k_l = k(C_l, c_{l+1})$
Compute $\gamma$ as in Equation (21)
if $\gamma > \eta$ then
  if $l < N_D$ then
    Update dictionary by storing the new point $c_{l+1}$, and
    recalculating $\gamma$'s for each of the points
  else
    Update dictionary by deleting the point with the
    minimal $\gamma$, and then recalculate $\gamma$'s for each of the
    points
  end if
end if
```

move in the same direction together. Furthermore, from the above analysis, it is clear that the amount of excitation is maximized when the centers are previous states themselves.

Summarizing, the following are the advantages of picking centers online with the linear independence test (21):

1) In light of Theorem 2, Algorithm 1 ensures inserted excitation in the system does not disappear by selecting centers that ensure $\mathcal{F}_C$ is not orthogonal to current states. In less formal terms, it ensures that at least some centers are "sufficiently" close to the current state.
2) Algorithm 1 enables the design of adaptive controllers without any prior knowledge of the domain. For example, this method would allow one to initialize all centers to zero, with appropriate centers then selected by Algorithm 1 online.
3) On a budget, selecting centers with Algorithm 1 is better than evenly spacing centers, since the centers are selected along the path of the system in the state space. This results in a somewhat optimal distribution of centers without any prior knowledge of the uncertainty.

If the centers for the system are picked using the kernel linear independence test, and the weight update law is given by the standard baseline law (16), we call the resulting algorithm Budgeted Kernel Restructuring (BKR). If BKR is augmented with a concurrent learning update law, the resulting algorithm is denoted by BKR-CL.

V. Lyapunov Analysis

In this section, we prove the boundedness of the closed loop signals when using BKR-CL. Note that Algorithm 1 picks the centers discretely; therefore, there always exists an interval $[t_k, t_{k+1}]$ where $k \in \mathbb{N}$ where the centers are fixed. The discrete update of the centers introduces switching in the closed loop system. Let $\sigma^k(x)$ denote the value of σ given by this particular set of centers, and denote by W^{k^*} the ideal set of weights for these centers and by $\sigma^k(\bar{x})$ the radial basis function for these centers. Then, over each interval, the tracking error dynamics are given by the following switching system

$$\dot{e} = Ae + [W^T\sigma^k(\bar{x}) - \Delta(\bar{x})]. \tag{22}$$

The NN approximation error of (15) for the k^{th} system can be rewritten as

$$\Delta(\bar{x}) = W^{k^{*^T}}\sigma^k(\bar{x}) + \tilde{\epsilon}^k(\bar{x}), \tag{23}$$

with $\bar{\epsilon}^k = \sup_{\bar{x}\in D}\|\tilde{\epsilon}^k(\bar{x})\|$. Now, we prove the following equivalent of Theorem 1 for the following switching weight update law

$$\dot{W} = -\Gamma_W\sigma^k(\bar{x})e^TPB - \Gamma_W\sum_{j=1}^{p}\sigma^k(\bar{x}_j)\epsilon_j^{k^T}. \tag{24}$$

Theorem 4 Consider the system in (4), the control law of (10), $\bar{x}(0) \in D$ where D is compact, and the case of unstructured uncertainty. For the j^{th} recorded data point let $\epsilon_j^k(t) = W^T(t)\sigma^k(\bar{x}_j) - \Delta(\bar{x}_j)$, let p be the number of recorded data points $\sigma_j^k := \sigma^k(\bar{x}_j)$ in the matrix $Z = [\sigma_1^k, ..., \sigma_p^k]$, such that $\operatorname{rank}(Z) = l$, and assume that the RBF centers are updated using Algorithm 1. Then, the weight update law in (24) ensures that the tracking error e of (22) and the RBF NN weight errors $\tilde{W}^k$ of (23) are bounded.

Proof: Consider the tracking error dynamics given by (22) and the update law of (24). Note that since $\nu_{ad} = W^T(\sigma^k(x))$, so $\epsilon_j = W^T\sigma^k(\bar{x}) - \Delta(\bar{x})$. The NN approximation error is now given by (23). With $\tilde{W}^k = W - W^{k^*}$, this implies that

$$\epsilon_j^k(x) = \tilde{W}^k\sigma^k(x) - \tilde{\epsilon}^k(x).$$

Therefore over $[t_k, t_{k+1}]$, the weight dynamics are given by the following switching system

$$\begin{aligned}\dot{\tilde{W}}^k(t) = -\Gamma\Bigg[&\left(\sum_{j=1}^{p}\sigma^k(\bar{x}_j)\sigma^{k^T}(\bar{x}_j)\right)\tilde{W}^k(t)\\ &+\sum_{j=1}^{p}\sigma^k(\bar{x}_j)\tilde{\epsilon}^{k^T}(z_j) - \sigma^k(\bar{x}(t))e^T(t)PB\Bigg].\end{aligned} \tag{25}$$

Consider the family of positive definite functions $V^k = \frac{1}{2}e^TPe + \frac{1}{2}\operatorname{Tr}(\tilde{W}^{k^T}\Gamma_W^{-1}\tilde{W}^k)$, where $\operatorname{Tr}(\cdot)$ denotes the trace operator. Note that $V^k \in \mathcal{C}^1$, $V^k(0) = 0$ and $V^k(e, \tilde{W}^k) > 0\ \forall e, \tilde{W}^k \neq 0$, and define $\Omega^k := \sum_j \sigma_j^k s_j^{kT}$. Then

$$\begin{aligned}\dot{V}^k(e,\tilde{W}^k) &= -\frac{1}{2}e^TQe - \operatorname{Tr}\Bigg(\tilde{W}^{k^T}\Bigg[\sum_j\sigma_j^k\sigma_j^{kT}\tilde{W}^k\\ &\qquad + e^TPB\tilde{\epsilon} - \sum_j\sigma_j\tilde{\epsilon}_j^k\Bigg]\Bigg)\\ &\leq -\frac{1}{2}\lambda_{\min}(Q)e^Te - \lambda_{\min}(\Omega^k)\tilde{W}^{k^T}\tilde{W}^k\\ &\qquad + \|e^TPB\bar{\epsilon}\| - \|\tilde{W}^{k^T}\sum_i\sigma_i^k\tilde{\epsilon}^k\|\\ &\leq \|e\|\left(C_1^k - \frac{1}{2}\lambda_{\min}(Q)\|e\|\right)\\ &\qquad + \|\tilde{W^k}\|(C_2^k - \lambda_{\min}(\Omega^k)\|\tilde{W}^k\|),\end{aligned}$$

where $C_1^k = \|PB\|\tilde{\epsilon}^k$ and $C_2 = p\tilde{\epsilon}^k\sqrt{q}$, ($q$ being the number of RBFs in the network). Hence if $\|e\| > 2C_1/\lambda_{\min}(Q)$ and $\|\tilde{W}^k\| > C_3/\lambda_{\min}(\Omega^k)$, we have $\dot{V}(e,\tilde{W}^k) < 0$, which means the set $\Pi^k = \{(e,\tilde{W}^k) : \|e\| + \|\tilde{W}^k\| \leq 2C_1/\lambda_{\min}(Q) + C_2/\lambda_{\min}(\Omega^k)\}$ is positively invariant for the k^{th} system.

Let $\mathcal{S} = \{(t_1, 1), (t_2, 2), \ldots\}$ be an arbitrary switching sequence with finite switches in finite time (note that this is always guaranteed due to the discrete nature of Algorithm 1). The sequence denotes that a system S_k was active between t_k and t_{k+1}. Suppose at time t_{k+1}, the system switches from S_k to $S_k + 1$. Then $e(t_k) = e(t_{k+1})$ and $\tilde{W}^{k+1} = \tilde{W}^k + \Delta W^{k^*}$, where $\Delta W^{k^*} = W^{k^*} - W^{(k+1)^*}$. Since $\dot{V}^k(e,\tilde{W}^k)$ is guaranteed to be negative definite outside of a compact interval, it follows that $e(t_k)$ and $\tilde{W}^k(t_k)$ are guaranteed to be bounded. Therefore, $e(t_{k+1})$ and $\tilde{W}^{k+1}(t_{k+1})$ are also bounded and since $\dot{V}^{k+1}(e,\tilde{W}^{k+1})$ is guaranteed to be negative definite outside of a compact set $e(t)$ and $\tilde{W}^{k+1}(t)$ are also bounded. Furthermore, over every interval $[t_k, t_{k+1}]$, they will approach the positively invariant set Π^{k+1} or stay bounded within Π^{k+1} if inside.

■

VI. Simulation Results

In this section we present simulation results of the above scheme for adaptive control of a nonlinear system. Let θ denote the angular position and δ denote the control input; then the unstable dynamics under consideration are given by

$$\ddot{\theta} = \delta + \sin(\theta) - |\dot{\theta}|\dot{\theta} + 0.5e^{\theta\dot{\theta}}. \qquad (26)$$

A second order reference model with natural frequency and damping ratio of 1 is used, the linear controller is given by $K = [1.5, 1.3]$, and the learning rate is set to $\Gamma_W = 3.5$. The initial conditions are set to $x(0) = [\theta(0), \dot{\theta}(0)] = [1, 1]$. The model uncertainty is given by $y = W^{*T}\Phi(x)$ with $W^* = [-1, 1, 0.5]$ and $\Phi(x) = [\sin(\theta), |\dot{\theta}|\dot{\theta}, e^{\theta\dot{\theta}}]$. A step in position ($\theta_c = 1$) is commanded at $t = 20\,\text{sec}$.

The comparisons were made between Budgeted Kernel Restructuring without concurrent learning (BKR), concurrent learning without BKR (CL) and BKR with concurrent learning (BKR-CL). In both cases, the number of centers was chosen to be 35. In BKR, the centers were all initialized to $0 \in \mathbb{R}^2$ at $t = 0$, while in CL they were spaced evenly across the domain of uncertainty (where the prior knowledge of the domain was determined through experiments conducted beforehand). Figure 3(a) shows the tracking of the reference model by CL, BKR and BKR-CL. Figure 3(b) shows the tracking error for the same. As can be seen, BKR-CL is the most effective scheme. Figure 4 shows an example of how concurrent learning affects the evolution of weights; it has a tendency to spread the weights across a broader spectrum of values, leading the algorithm to choose from a richer class of functions in $\mathcal{F}_C$. This is the reason BKR-CL does better than BKR alone. Figure 5 shows the final set of centers that are used; as can be seen, the centers follow the path of the system in the state space. Figure 6 shows how our scheme is better at uncertainty tracking than evenly spaced centers on a budget; Figure 6(a) shows uncertainty tracking with CL, while Figure 6(b) shows the same with BKR-CL.

VII. Conclusion

In this work, we established a connection between kernel methods and Persistently Exciting (PE) signals using Reproducing Kernel Hilbert Space theory. Particularly, we showed that in order to ensure PE, not only do the system states have to PE, but the centers need to be also selected in such a way that the mapping from the state space to the underlying RKHS is never orthogonal to the linear subspace generated by the Radial Basis Function (RBF) centers. This ensures that the output of the radial basis function does not vanish. We used this connection to motivate an algorithm called Budgeted Kernel Restructuring (BKR) that updates the RBF network in a way that ensures any inserted excitation is retained. This enabled us to design adaptive controllers without assuming any prior knowledge about the domain of the uncertainty. Furthermore, it was shown through simulation that on a budget (limitation on the maximum number of RBFs used), the method is better at capturing the uncertainty than evenly spaced centers, because the centers are selected along the path of the system through the state space. We augmented BKR with Concurrent Learning (CL), a method that concurrently uses specifically recorded and instantaneous data for adaptation, to create the BKR-CL adaptive control algorithm. It was shown through Lyapunov-like analysis that the weights for the centers picked by BKR-CL are bounded without needing PE, simulations showed improved tracking performance.

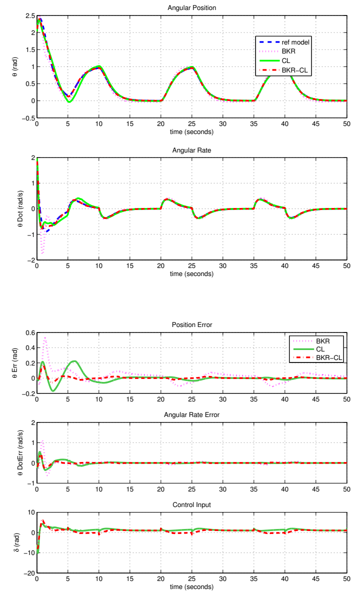


Fig. 3. Tracking with BKR and no concurrent learning (BKR), with no BKR and concurrent learning (CL), and with both BKR and concurrent learning (BKR-CL). As can be seen, BKR-CL is the most effective scheme in terms of tracking.

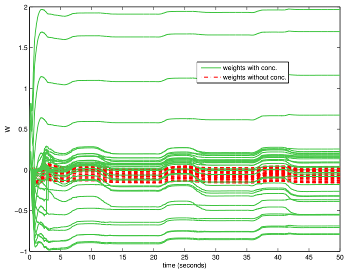


Fig. 4. Weight evolution with BKR-CL and with BKR.

VIII. Acknowledgments

This work was supported in part by NSF ECS-0238993, NSF ECS-0846750, and NASA Cooperative Agreement NNX08AD06A. We thank Maximilian Mühlegg for his valuable help in revising this paper and in the simulations.

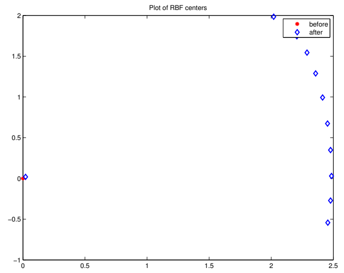
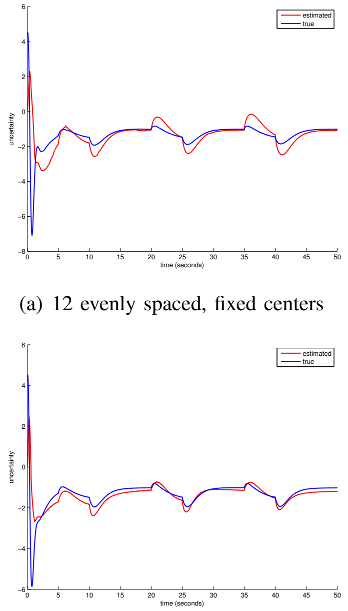


Fig. 5. The final selected centers in the state space using BKR.



(a) 12 evenly spaced, fixed centers

(b) 12 centers with BKR

Fig. 6. Uncertainty tracking with 12 centers. Uncertainty tracking with BKR is clearly more accurate on a budget, mainly due to the centers being picked on the path of the system in the state space.

References

[1] K. J. Aström and B. Wittenmark. *Adaptive Control*. Addison-Weseley, Readings, 2nd edition, 1995.

[2] S. Boyd and S. Sastry. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica*, 22(6):629–639, 1986.

[3] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[4] Anthony J. Calise, Manu Sharma, and Seungjae Lee. Adaptive autopilot design for guided munitions. *AIAA Journal of Guidance, Control, and Dynamics*, 23(5), 2000.

[5] G. Chowdhary. *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[6] Girish Chowdhary and Eric N. Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control*, 2010.

[7] N. Cristianini and J. Shawe-Taylor. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[8] P. A. Ioannou and J. Sun. *Robust Adaptive Control*. Prentice-Hall, Upper Saddle River, 1996.

[9] Petros A. Ioannou and Peter V. Kokotovic. *Adaptive Systems with Reduced Models*. Springer Verlag, Secaucus, NJ, 1983.

[10] E. N. Johnson. *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2000.

[11] Eric N. Johnson and Anthony J. Calise. Limited authority adaptive flight control for reusable launch vehicles. Journal of Guidance, Control and Dynamics, 2001.

[12] S. Kannan. *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2005.

[13] N. Kim. *Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2003.

[14] N. Nardi. *Neural Network based Adaptive Algorithms for Nonlinear Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, nov 2000.

[15] K. Narendra and A. Annaswamy. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134–145, February 1987.

[16] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*. Prentice-Hall, Englewood Cliffs, 1989.

[17] D. Nguyen-Tuong, B. Schölkopf, and J. Peters. Sparse online model learning for robot control with support vector regression. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[18] J. Park and I.W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.

[19] I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. *IEEE Transactions on Information Theory*, 52(10), 2006.

[20] N. Sundararajan, P. Saratchandran, and L. Yan. *Fully Tuned Radial Basis Function Neural Networks for Flight Control*. Springer, 2002.

[21] J.A.K. Suykens, J.P.L. Vandewalle, and B.L.R. De Moor. *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. Kluwer, Norwell, 1996.

[22] G. Tao. *Adaptive Control Design and Analysis*. Wiley, New York, 2003.

[23] Kostyantyn Y. Volyanskyy, Wassim M. Haddad, and Anthony J. Calise. A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond σ and e-modifications. *IEEE Transactions on Neural Networks*, 20(11):1707–1723, Nov 2009.