Enhanced PID: Adaptive Feedforward RBF Neural Network Control of Robot manipulators with an Optimal Distribution of Hidden Nodes

Qiong Liu, Dongyu Li, Shuzhi Sam Ge, Zhong Ouyang, and Wei He

Abstract—This paper focus on three inherent demerits of adaptive feedback RBFNN control with lattice distribution of hidden nodes: 1) The approximation area of adaptive RBFNN is difficult to be obtained in priori; 2) Only partial persistence of excitation (PE) can be guaranteed; 3) The number of hidden nodes is the exponential growth with the increase of the dimension of the input vectors and the polynomial growth with the increase of the number of the hidden nodes in each channel which is huge especially for the high dimension inputs of the RBFNN. Adaptive feedforward RBFNN control with lattice distribution of hidden node can improve solve the demerits 1) but just improve demerits 2) and 3) slightly. This paper proposes an adaptive feedforward RBFNN control strategy with an optimal distribution of hidden nodes. It solves the demerits 2) and 3) that the standard PE can be guaranteed and the number of hidden nodes is linear increase with the complexity of the desired state trajectory rather than the exponential growth with the increase of the dimension of the input vectors. In addition, we articulate that PID is the special case of adaptive feedforward RBFNN control for the set points tracking problem and we named the controller is enhanced PID. It is very easy tuning our algorithm which just more complex than PID slightly and the tuning experience of PID can be easily transferred to our scheme. In the case of the controller implemented by digital equipment, the control performance can equal or even better than it in model-based schemes such as computed torque control and feedforward nonlinear control after enough time to learn. Simulations results demonstrate the excellent performance of our scheme. The paper is a significant extension of deterministic learning theory.

Index Terms—Adaptive RBFNN control, deterministic learning, feedback compensation, K-means, robot manipulator, enhanced PID.

I. INTRODUCTION

Adaptive RBF neural network (RBFNN) control is an effective way to handle the uncertainties of the dynamic of systems when both the structures and parameters are unknown [1]-[4]. Compared with other non-linearly parametrized networks, the local response plays an effective role in both learning process and approximation process is the unique characteristic of RBFNNs with deterministic hidden nodes. The local response means that there is only a small part of the hidden nodes activated for each input vector. In the approximation process, the approximated values are in linear correlation with the values of the activated hidden nodes. In the learning process, the weights are linear adjusted according to the local activated values. It leads to the most important advantage that is the faster learning speed than that of both multilayer neural networks and RBFNNs with adjustable hidden nodes. Because of this, Wang Cong also named this kind of RBFNN as deterministic learning.

Generally, there are two structures to accomplish adaptive RBFNN control which are adaptive feedback RBFNN control and adaptive feedforward RBFNN control. Adaptive feedback RBFNN control is original from composite adaptive control which use both tracking errors in the joint motion and the prediction errors in the predicted filtered torque or power to drive the parameter adaptation [5]. Adaptive feedforward RBFNN control is derived from feedforward-plus-PD control which use adaptive RBFNN to approximate the feedforward dynamics. The same control structure also be utilized in adaptive control [6].

Adaptive feedback RBFNN control is widely used in the adaptive RBFNN community because it is beautiful and rigorous from the view of theory and simpler applied in the stability proof. It has many other merits, and this paper only focuses on the below three demerits:

1) To guarantee the partial PE, we need to construct the hidden node as lattice distribution and the inputs of RBFNN must stay within the regular lattice. However, the domain of the inputs of adaptive RBFNN is difficult to know in priori. The approximation must cover the domain of the inputs. If the states leave out the approximation area, the value of RBFs would vanish to zeros and the outputs of the RBFNN also vanish to zeros, which lead to the failure of the approximation [1], [7].

Generally, the trajectory errors cannot be known quantitatively in priori. The authors in [8], [9] estimate the domain of the trajectory errors by analyzing the initial state and the transition performance carefully and choosing the control parameters cautiously, but the transition performance is difficult to be decided quantitatively because of the unknown system. So the approximation area is designed roughly according to the desired states and the tracking errors of the states. The control performance is guaranteed by excessive simulations and tuning control parameters carefully [10]-[13]. Sliding mode control is also utilized to construct the framework of the adaptive RBFNN control in [7], [19].
When the states leave the approximation domain, the slide control scheme can push the states to the domain again. However, it needs extra information about the systems which is not easy to obtain.

The controller based on the Barrier Lyapunov function is designed to constrain the state errors in the designed intervals actively, and the method demands the design of the approximation area including the initial states \([20]–[22]\). It is effective in theory, but it complicates the controller design and needs the hardware with high-sampling to guarantee the small barrier in the real application.

2) Only the partial persistence of excitation (PE) of adaptive RBFNN is satisfied for adaptive feedback RBFNN control with deterministic hidden nodes \([23]\). This is meaning that, for a periodic reference orbit, only the localized RBFNN can satisfy PE condition, and the respected weights can convergence to its ideal value. For other hidden nodes whose respect RBF does not satisfy PE, the respected weight can not be guaranteed to convergence to its ideal value. It would degrade its robustness ability. In addition, partial PE needs strict requirements to be guaranteed and not easy to achieve. It should rigorously achieve the following three control process in sequence: 1) An adaptive feedback RBFNN controller is utilized to guarantee that the states errors of the closed-loop system can converge to the small intervals which can be arbitrary small in a finite time by increasing the control gains so that the closed-loop states become recurrent as the desired recurrent states; 2) The recurrent states lead to partial PE of the regression sub-vector of the localized RBFNN whose along the recurrent desired state trajectories; and 3) under the partial PE condition, the states error and the approximation error exponentially converge to the small intervals and the weights of the localized RBFNN converge to its optimal values. \([23], [24]\). We can see the process is complex and it needs high control gains to achieve the recurrent state trajectory in step (2). This is why most of the literature in the adaptive RBFNN control field just proof the closed-loop is semi-globally ultimate uniformly bounded rather than exponential stability as \([23]\).

3) With the lattice distribution of hidden node, the number of hidden nodes is \(m^p\), where \(m\) is the number of the hidden nodes in each channel and \(p\) is the dimension of the input vectors, which is the exponential growth with the increase of the dimension of the input vectors and the polynomial growth with the increase of the number of the hidden nodes in each channel. The large number is unaccepted especially in the high dimension of inputs because of the limited computation cost. The dimension of the input vectors is decided by the control structure and the degree of the controlled system. In adaptive feedback RBNN control, the dimension of input is \(5n\) \([25]\) where \(n\) is the degree of freedom of the system, and it can be reduced to \(4n\) further by introducing a virtual control variable to combine the states and the errors \([9], [14]\). In addition, to achieve better approximation performance and tracking performance, the number of hidden nodes in each channel should be increased, but we should trade off the control performance and the exponential growth of the number of hidden nodes \([1], [23]\). The huge number of hidden nodes need a huge computation cost to calculate it which limits its application in electrical equipment.

Compared with the population of the adaptive feedback RBN control, adaptive feedback RBFNN control with lattice distribution of hidden node is seldom utilized. The possible reason may be as following: The feedforward nonlinear control has the inherent drawback that the control gains should be large enough to suppress the residual errors between the feedback dynamics and feedforward dynamic; It degrades the beauty of the process of proof from the view of theory, but, from the view of practical application, the control performance is equal to computer torque control \([26]\) or even better at the exit of the noise or the imprecise dynamics \([27], [28]\). However, the controller can solve the demerits 1) and improve demerits 2) and 3) further.

1) Only the desired states rather than the real states which contain the desired states and the error of the states as the inputs of the adaptive RBFNN. Then the approximation area can be unknown in priori and the lattice distribution of the hidden node can be designed according to the approximation area. \([24], [29]–[31]\). In addition, the needed approximation area is smaller than it in the adaptive feedback RBFNN scheme. It means that the required hidden nodes for filling the area also can be reduced.

2) The strict requirements of the plants’ state are recurrent in step (1) of \([23]\) is not needed. The desired states rather than real states are utilized as the inputs of adaptive RBFNN guarantee the adaptive feedforward RBFNN satisfying the partial PE.

3) The dimension is reduced to \(3n\) further. It can reduce the control structure and the number of hidden nodes.

Based on the above analysis, this paper proposed the most effective, simplest adaptive RBFNN control scheme hitherto. Compared with the adaptive feedforward RBFNN control with lattice distribution of hidden node \([24], [29]–[31]\), the major difference of our controller is the hidden node is the optimal distribution for the desired states which is achieved by K-means algorithm before the begin of the control process. Compared with the above all literature, it leads to the qualitative improvement in the case of demerit 2) and 3) and in other aspects further:

1) The main contribution of our scheme is that it guarantees the standard PE rather than partial PE in \([23], [24], [32]–[35]\). Thanks to the PD-plus-feedforward control structure, the inputs of RBNN can be known in priori. All hidden nodes of the adaptive feedforward RBNN are optimally distributed that along the desired state trajectories rather than the lattice distribution in the approximation area which is decided by the bounded of the desired states. It achieves by the K-means algorithm. From the view of standard PE condition, the optimal distribution has two aspect meanings: 1) For each period, all hidden nodes are experienced the most activation periodically
along the desired state trajectory in sequence and each hidden node at least experience one time of the most activation; 2) The number of times of the most activation of each hidden node is similar. This is why our scheme can achieve the standard PE condition. Correspondingly, in lattice distribution of hidden nodes which just achieve the partial PE condition, 1) only the part of hidden nodes are experienced the most activation periodically along the desired state trajectory in sequence and there are still many of the hidden nodes even not experience any times of the most activation, those hidden nodes whose the never experience the periodically highest activation does not contribute any to the adaptive feedforward RBFNN control and it even leads to robustness problem. The bad behavior becomes worse with the increase of the dimension of inputs. 2) The number of times of the most activation of each hidden node maybe not similar because the distribution of hidden node is a lattice. It is inflexible to the state trajectories. This is the reason why our scheme is the most effective hitherto.

2) Our scheme can achieve the best approximation performance and control performance but with the fewest number of hidden nodes for the adaptive RBFNN among the adaptive RBFNN control field. This is means that our scheme can sharply decrease implementation cost in terms of hardware selection. The number of hidden nodes is linear increase with the complexity of the desired state trajectory rather than the exponential growth with the increase of the dimension of the input vectors and the exponential growth with the increase of the number of the hidden nodes in each channel. This is also brought by the optimal distribution of hidden nodes for the corresponding trajectory. It has two meanings from the view of the approximation: 1) all the hidden node is effective and no hidden node is useless. It can avoid all the uselessness of the hidden node of the lattice distribution of hidden nodes ad the number of hidden nodes is reduced tremendously especially for the high dimension of input of adaptive RBFNN; 2) each hidden node can nearly achieve it best approximation ability. The scheme proposed in this paper just more complex than PID slightly but the control performance can equal to model-based schemes such as computed torque control and feedforward nonlinear control after enough time to learn. It can be verified by the nearly perfect simulation result that the tracking errors are less than $2 \times 10^{-5}$ with the step size being 0.01s and 20 hidden nodes.

3) It is very easy tuning our algorithm which just complex than PID slightly and the tuning experience of PID can be easily transferred to our scheme. In the case of the period reference orbit, if the system can be controlled by PD or PID controller, even the control performance is not good, the adaptive feedforward RBFNN can be added with slight modification and improve the control performance tremendously. Moreover, the tuning experience of the learning rate is similar to the control gain of the integral term of PID can easily be transferred to our scheme. In addition, our paper articulate an interesting but wired problem that PID is the simplest case of adaptive feedforward RBFNN control which is just for set point tracking problem. Adaptive feedforward RBFNN control is an enhanced PID and the excellent approximation ability is the key reason for adaptive RBFNN control over PID.

4) In the case of the controller implemented by digital equipment, with the help of the gradient learning algorithm which utilizes the feedback error to train RBFNN, our method even can approximation the error of dynamics which is led by sample time. It makes it have the potential to achieve better control performance than model-based schemes such as computed torque control and feedforward nonlinear control after enough time to learn.

We have to remind that the optimal distribution of hidden nodes is calculated by the K-means algorithm according to the desired state trajectory in priori. It is just optimal to the inputs of the RBFNN and not optimal to the target functions because the target functions are unknown in priori and the property of target functions cannot be considered.

The rest of this paper is organized as follows. The control problem is formulated in Section II including the dynamics description, function approximation of RBFNNs, K-means algorithm, and PE condition of RBFNNs. The main results are given in Sections III. In Section IV, we articulate that PID is the special case of adaptive feedforward RBFNN control for the set points tracking problem and we named the controller is enhanced PID. Simulations are presented in Section V. Finally, conclusions are given in Section VI.
Remark 1. It is worth noticing that NN control is just suited to deal with recurrent trajectory orbits problems because NNs only can memory a similar problem which it has seen beforehand. In this paper, we only consider the circumstance that the trajectories are periodic.

B. Function Approximation

The activation functions of adaptive RBFNN are Gaussian function and the form are:
\[ S_{ij}(Z) = \exp \left[ -\frac{(Z - \mu_j)^T(Z - \mu_j)}{\sigma_i^2} \right], \quad j = 1, 2, \ldots, m, \quad (3) \]
where \( \mu_j = [\mu_{j1}, \mu_{j2}, \ldots, \mu_{jp}]^T \) is the position of the hidden node, \( Z = [Z_1, Z_2, \ldots, Z_p]^T \in \Omega_Z \subseteq \mathbb{R}^q \) is the input vector, and \( \sigma_i \) is the width of Gaussian functions of the \( i \)th outputs.

The output of the MIMO functions can be approximated as:
\[ F_i(Z) = W_i^T S_i(Z) + \varepsilon_i(Z), \quad \forall Z \in \Omega_Z. \quad (4) \]

The ideal weights of the RBFNNs are defined as:
\[ W_i^* := \arg \min_{W_i} \left\{ \sup_{Z \in \Omega_Z} |F_i - \hat{W}_i^T S_i(Z)| \right\}. \quad (5) \]

The continuous function can be approximated by the RBFNNs which is achieved through:
\[ F(Z) = \hat{W}^T \bullet S(Z), \quad \forall Z \in \Omega_Z, \quad (6) \]
where \( \bullet \) means GL product defined in \([10]\), \( W^T \bullet S(Z) := [W_1^T S_1(Z), W_2^T S_2(Z), \ldots, W_n^T S_n(Z)]^T \), and \( F = [F_1, F_2, \ldots, F_n]^T \).

C. K-means Algorithm

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the clustering problem, which finds a globally optimal partition of a given data into a specified number of clusters. The method aims at minimizing the objective function:
\[ J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_{ij} - c_j \right\|^2, \quad (7) \]
where \( \left| x_{ij} - c_j \right|^2 \) is a chosen distance measure between a data point \( x_{ij} \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centres.

There are many methods to achieve this aim, and in this paper, we utilize the K-means++ method \([37]\) to find the optimal distribution of hidden nodes of the RBFNN.

D. Persistence Excitation for RBFNN with an Optimal Distribution of Hidden Node

Persistent excitation is one of the great importance in adaptive control and identification problem, which decide whether the parameters can convergence to its real parameters.

Definition 1. \([23]\): A piecewise-continuous, uniformly-bounded, vector-valued function \( S : [0, \infty) \to \mathbb{R}^m \) is said to satisfy the persistent excitation condition, if there exist positive constants \( \alpha_1, \alpha_2, \) and \( T_0 \) such that:
\[ \alpha_1 I \geq \int_{t_0}^{t_0+T_0} S(\tau)S(\tau)^T d\tau \geq \alpha_2 I \quad \forall t_0 \geq 0, \]
where \( I \in \mathbb{R}^{m \times m} \) is the identity matrix.

According to this definition, the PE condition requires that the integral of the semidefinite matrix \( S(\tau)S(\tau)^T \) be uniformly positive definite over an interval of length \( T_0 \). It is noted that if is persistently exciting for the time interval \([t_0, t_0 + T_0]\), it is PE for any interval of length \( T_1 > T_0 \) \([38]\).

Lemma 1. Consider any continuous periodic orbit \( Z(t) : \mathbb{R}^+ \to \Omega_x \) with period \( T_0 \). For the approximation area which is decided by the \( \epsilon \)-neighborhood of the hidden nodes of the RBFNN \( W^T S(Z) \) is large enough to cover the periodic orbit \( Z(t) \), the regressor \( S(Z) \), is standard PE rather than partial PE in \([23], [39], [40]\). The time of PE is \( T > T_0 \), where \( T_0 \) is period of the desired state trajectory.

K-means algorithm guarantees that every input visits the specified neighborhood of the neuron center of the RBFNN and each neuron center can be visited at least once. The \( \epsilon \)-neighborhood of the hidden nodes calculated by the K-means algorithm is the local region along the trajectory, which is equal to the localized RBF networks \( S_i(Z) \) defined in \([23]\). The proof of Lemma \( [1] \) is also the same as that of Theorem 2 in \([23]\).

III. CONTROL DESIGN

Defining the tracking errors as follows:
\[ e_1 = q_d - q, \]
\[ \dot{e}_1 = \dot{q}_d - \dot{q}. \quad (8) \]
A virtual control term is introduced as:
\[ e_2 = \dot{e}_1 + K_1 e_1, \quad (9) \]
where \( K_1 = \text{diag}(K_{11}, K_{12}, \ldots, K_{1n}) \) is diagonal positive definite matrix.

Thus, we have:
\[ \dot{q}_r = \dot{q} + e_2 = \dot{q}_d + K_1 e_1, \]
\[ \ddot{q}_r = \ddot{q} + \ddot{e}_2 = \ddot{q}_d + K_1 \ddot{e}_1. \]

The model based PD-plus-feedforward controller is recast as:
\[ \tau = K_2 e_2 + M(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + G(q_d), \quad (10) \]
where \( K_2 = \text{diag}(K_{21}, K_{22}, \ldots, K_{2n}) \) is diagonal positive definite matrix. \( K_2 e_2 = K_1 K_2 e + K_2 \dot{e}_1 \) is a PD term.

In adaptive feedback RBFNN controller, the feedback dynamics term \( M(q) \ddot{q}_r + C(q, \dot{q}) \dot{q} + G(q) \) is approximated by RBFNN, then the controller is transferred into:
\[ \tau = K_2 e_2 + \hat{W}^T \bullet S(Z), \quad (11) \]
where \( Z = [q^T, \dot{q}^T, \ddot{q}^T, \dot{q}_r^T, \ddot{q}_r^T]^T \).
In adaptive feedforward RBFNN controller, the RBFNN $W^T \bullet S(Z)$ are utilized to approximate the feedforward dynamics as follows:

$$W^* T \bullet S(Z) + \epsilon(Z) = M(q_a) \tilde{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + G(q_a) \tag{12}$$

where $Z = [\tilde{q}_d, \dot{q}_d, \ddot{q}_d]^T$ is the input of adaptive RBFNN.

The controller is formulated to:

$$\tau = K_2 e_2 + \dot{W}^T \bullet S(Z_a), \tag{13}$$

where $Z = [\tilde{q}_d, \dot{q}_d, \ddot{q}_d]^T$.

A residual error is led by replacing the feedback dynamics with the feedforward dynamic:

$$\bar{H} = (M(q) \tilde{q}_r + C(q, \dot{q}) \dot{q} + G(q)) - (M(q_a) \tilde{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + G(q_a)). \tag{14}$$

In a same manner as Remark 3 of [41], since $M(q), C(q, \dot{q})$ and $G(q)$ are of class $C^1$, $\forall q, \dot{q}, \ddot{q} \in \mathbb{R}^n$. Defining $e = \coth(e_1, e_2)$. The Mean Value Theorem can be applied to $\bar{H}$ to obtain:

$$\|ar{H}\| \leq \rho(||e||)||e||, \tag{15}$$

in which $\rho_1 : \mathbb{R}^+ \to \mathbb{R}^+$ is a certain function that is globally invertible and strictly increasing [42].

**Remark 2.** We can also represent the $\bar{H}$ as [43]:

$$-e^T_2 \bar{H} \leq e^T_2 \left(\lambda_{\max}(K_1) \lambda_M + b_1 I\right) e_2 + e^T_2 \left(-\lambda^2_{\min}(K_1) \lambda_M + b_2 I\right) e_1 + b_3 \left(\|e_2\|^2 \|e_1\| + K_1 \|e_2\| \|e_1\|^2\right), \tag{16}$$

In this paper, we do use the format [15] rather than [16] for simple representation.

The gradient method with the discontinuous switching $\delta$-modification is utilized to train the RBFNNs:

$$\dot{W}_i = \Gamma_i \left(S_i(Z) e_{2i} - \delta_i \bar{W}_i\right), \tag{17}$$

where the discontinuous switching $\delta$ is

$$\delta_i = \begin{cases} 0 & \text{if } \|\bar{W}_i\| < W_{0i} \\ \delta_{i0} & \text{if } \|\bar{W}_i\| \geq W_{0i} \end{cases}. \tag{18}$$

**Remark 3.** Utilizing $\delta$-modification to avoid the behavior of parameter drift in [1], [10], [12], [23] lead a drawback that it adds pure damping to weights, which inhibit the adaptation process and the weight cannot convergence to its optimal value even in the ideal circumstance of without external disturbance and the PE satisfied. The ideal form of $\delta$-modification is $\delta(W^* - \bar{W})$, but $W^*$ is cannot be known in priori, so the common practices is choosing $W^* = 0$. This modification can guarantee the boundedness of $\bar{W}$, but it brings a key drawback that it adds pure damping to weights, which inhibit the adaptation process. It limits the potential performance of the controller. The discontinuous switching $\delta$-modification can avoid the drawbacks by setting $\delta = 0$ when the weight $\bar{W}_i$ is bounded by the designed value $W_{0i}$. Theorem 1. For the Euler-Lagrange system [1] under Property [1], [10], the controller [10] and the learning algorithms [17], the tracking error $e_1$, $e_2$, and approximation error $W$ will exponential convergence to the small intervals. The interval can be arbitrary small by increase the control gain $K_1$ and $K_2$ and decreasing $\epsilon$ by improving the approximation performance of the RBFNNs.

1) Proving the boundedness of the state error $e_1$ and $e_2$ and the weight error $\bar{W}_i$.

Considering the following Lyapunov function candidiate:

$$V = \frac{1}{2} e_1^T e_1 + \frac{1}{2} e_2^T M(q) e_2 + \frac{1}{2} \sum_{i=1}^{n} \bar{W}_i^T \Gamma_i^{-1} \bar{W}_i. \tag{19}$$

The derivative of $V$ is:

$$\dot{V} = -e_1^T K_1 e_1 + e_2^T M(q) e_2 + \frac{1}{2} \sum_{i=1}^{n} \bar{W}_i^T \Gamma_i^{-1} \bar{W}_i, \tag{20}$$

where $\bar{W}_i = W^*_i - \bar{W}_i$.

Let us solve the error equation $M(q) \dot{e}_2$ firstly. The RBFNN can be reformulated as:

$$\dot{W}^T \bullet S(Z_a) = M(q) \tilde{q}_r + C(q, \dot{q}) \dot{q} + G(q) - \bar{H} - \epsilon(Z) - \dot{W}^T \bullet S(Z_d). \tag{21}$$

Applying the aforementioned result [21] into the controller [13], we have:

$$\tau_i = K_2 e_2 + M(q) \tilde{q}_r + C(q, \dot{q}) \dot{q} + G(q) - \epsilon(Z) - \dot{W}^T \bullet S(Z_d) - \bar{H}. \tag{22}$$

Substituting the controller [22] into the closed-loop system [1], the error equation of the closed-loop system can be obtained:

$$M(q) \dot{e}_2 + C(q, \dot{q}) e_2 = -K_2 e_2 + \bar{H} + \epsilon(Z) + \dot{W}^T S(Z_d). \tag{23}$$

Substituting the error equation [23] into the derivative of the Lyapunov function [20] and using Property [3], further yields:

$$\dot{V} = -e_1^T K_1 e_1 + e_2^T e_2 + \frac{1}{2} \bar{W}_2^T M\dot{e}_2 + e_2^T \left(-K_2 e_2 + \bar{H} + \epsilon(Z) + \dot{W}^T \bullet S(Z_d)\right) - \sum_{i=1}^{n} \bar{W}_i^T \Gamma_i^{-1} \bar{W}_i \leq -\lambda_{\min}(K_1) \frac{1}{2} \|e_1\|^2 - \lambda_{\min}(K_2 - 1) \|e_2\|^2 + e_2^T \bar{H} + \frac{1}{2} e_2^2 + \sum_{i=1}^{n} \bar{W}_i^T (S_i(Z) e_{2i} - \Gamma_i^{-1} \bar{W}_i). \tag{24}$$

It easy to obtain $\|e_2\| \leq \|e\|$. Noting $\bar{H}$ in [15], we have $e_2^T \bar{H} \leq \|e_2\| \rho(||e||) \leq \|e\|^2 \rho(||e||)$. Substituting the above inequality, we have:
\[
\dot{V}_i \leq -\lambda_{\min}(K_1 - \frac{1}{2}I)\|e_i\|^2
\]
\[
- \lambda_{\min}(K_2 - C(q_0, q_0) - 1)\|e_2\|^2
\]
\[
+ \|e\|^2 \rho(\|e\|) + \frac{1}{2} \dot{\varepsilon}^2 + \sum_{i} \delta_i \dot{W}_i^T \dot{W}_i
\]
\[
\leq - (K_\ast - \rho(\|e\|)) \|e\|^2 + \frac{1}{2} \dot{\varepsilon}^2 + \sum_{i} \delta_i \dot{W}_i^T \dot{W}_i,
\]
with \(K_\ast = \lambda_{\min}(\lambda_{\min}(K_1 - \frac{1}{2}I), \lambda_{\min}(K_2 - 1)).\)

In the case of \(\delta_i \dot{W}_i^T \dot{W}_i\), when \(\|W_i\| \leq M_0, \delta_i = 0\) and \(\|W_i\| < 2M_0 \Rightarrow \delta_i = 2W_i^2 - \|W_i\|^2 \geq 0 \Rightarrow \delta_i \dot{W}_i^T \dot{W}_i = 0 \leq \frac{1}{2}(2W_i^2 - \|W_i\|^2)\): When \(\|W_i\| \leq M_0, \delta_i = \delta_0\) \(\Rightarrow \delta_i \dot{W}_i^T \dot{W}_i = \delta_0 \dot{W}_i^T \dot{W}_i = -\delta_0 \dot{W}_i^T \dot{W}_i + \delta_0 \dot{W}_i^T \dot{W}_i \leq -\frac{\delta_0}{2} \||W_i\|^2 + 0.\)

Then, we obtain:
\[
\dot{V}_i \leq - (K_\ast - \rho(\|e\|)) \|e\|^2 + \frac{1}{2} \dot{\varepsilon}^2 + \sum_{i} \delta_i \dot{W}_i^2.
\]
(26)

The domain of \(K_\ast - \rho(\|e\|) > 0\) is estimated by:
\[
\Omega_{\text{er}} := \{e \mid \|e\| < \rho^{-1}(K_\ast)\}.
\]
(27)

\(\dot{V}\) can be recast as the standard format:
\[
\dot{V} \leq -c_1 V + c_2 \text{ for } \|e\| < \rho^{-1}(K_\ast)
\]
(28)

where \(c_1 = \min\left(\frac{\lambda_{\min}(K_2 - \rho(\|e\|))}{\lambda_{\max}(1,M)}, \frac{\sigma_0}{\lambda_{\max}(1)}\right)\) and \(C_0 = \frac{1}{2} \dot{\varepsilon}^2 + \sum_{i} \delta_i \dot{W}_i^2\).

Integrating (28), we have:
\[
V \leq (V(0) - \frac{c_2}{c_1}) \exp^{-c_1 t} + \frac{c_2}{c_1} \leq V(0) + \frac{c_2}{c_1}
\]
(29)

Define \(D := 2(V(0) + \frac{c_2}{c_1})\). Then the tracking errors \(\|e\|\) and the weights error \(\dot{W}_i\) is bounded by:
\[
\|e_i\| \leq \sqrt{D}
\]
\[
\|e_2\| \leq \sqrt{D \lambda_{\min}(M)}
\]
\[
\|\dot{W}\| \leq \sqrt{D \lambda_{\min}(1^-)}
\]
(30)

To guarantee the stability of the closed loop system, \(\Omega_{\text{er}}\) can be arbitrarily enlarged by the increase of the control gains \(K_1\) and \(K_2\) to include the bounded of \(|e_1|\) and \(|e_2|\).

2) Proving the state error \(e_1\) and \(e_2\) and weight error \(\dot{W}\) exponentially converging to the residual intervals under the regressor \(S(Z_d)\) being PE.

The closed loop system can be expressed as:
\[
\begin{align*}
\dot{e}_1 &= -\frac{1}{K_1}e_1 + \frac{1}{K_1}Z_2 \\
\dot{e}_2 &= M(q)\dot{e}_2 + C_{e2} = -K_2\dot{e}_2 + \dot{H} + \epsilon(Z) \\
&+ \dot{W}^T S(Z_d) \\
\dot{W} &= -\dot{W} = -\Gamma S(Z_d)\dot{e}_2 + \sigma \Gamma \dot{W}.
\end{align*}
\]
(31)

We can recast the closed loop system (31) as
\[
\begin{bmatrix}
\dot{e}_1 \\
M \dot{e}_2
\end{bmatrix} =
\begin{bmatrix}
-K_1 \frac{1}{M} & 0 \\
0 & -\frac{MS(Z_d)}{M}
\end{bmatrix}
\begin{bmatrix}
e_1 \\
M \dot{e}_2
\end{bmatrix}
+ \begin{bmatrix}
\dot{H} + \epsilon \\
\sigma \Gamma \dot{W}
\end{bmatrix}.
\]
(32)

For the sake of simple analysis, the system is also recast as:
\[
\begin{bmatrix}
\dot{E} \\
\dot{W}
\end{bmatrix} =
\begin{bmatrix}
-A & bS(Z_d) \frac{1}{M} \\
-bT S(Z_d) & 0
\end{bmatrix}
\begin{bmatrix}
E \\
W
\end{bmatrix}
+ \begin{bmatrix}
b(\dot{H} + \epsilon) \\
\sigma \Gamma \dot{W}
\end{bmatrix}
\]
(33)

where \(b = [0^n, 1^n]^T\) and \(c_0 = [1^n, \frac{1}{M}]^T\). The matrix \(A\) can satisfies \(A + AT \leq -Q \leq 0\) by increasing the control gain \(K_2\) and \(K_2\), then \((A, b)\) is controllable. The system (33) is perturbed system and \(H\) is a vanishing perturbation and \(\epsilon\) is the the RBFNN approximation error bounded by \(\dot{e}\) which is a nonvanishing perturbation. \(W\) is bounded according to the above analysis and \(\sigma = 0\) when \(\dot{W}\) decrease below \(W_0\).

Now, since \(S(Z_d)\) is PE and \((A, b)\) is controllable. In the case of the vanishing perturbation, according to Lemma 9.1 in [44], there exist suitably large control gains \(K_1\) and \(K_2\) to make the system is exponentially convergence. According to the exponential convergence results given in [23], [24], [45]–[47], the tracking error \(e_1\), \(M \dot{e}_2\), and approximation error \(\dot{W}\) will exponential convergence to the small intervals. The interval can be arbitrary small by increase the control gain \(K_1\) and \(K_2\) and decrease \(\epsilon\) by improving the approximation performance of the RBFNNs.

IV. DISCUSSIONS: WHY AFF-RBFNN CONTROL IS AN ENHANCED PID

We analyze the AFF-RBFNN controller from the view of the approximation ability and compare them with the PID controller for the sake of understanding. To simplify the analysis, we just consider the function of the bias in the controller at the circumstance \(\|W_i\| \leq c_i\) for simplifying. The AFF-RBFNN controller (13) can be reshaped by combining the adaptive law (17) as the integral format respectively
\[
\tau_i = K_{2i} r_i + W_i^T S_i(Z_d)
\]
\[
= K_{2i} r_i + \Gamma_i S_i(Z_d)^T \int S_i(Z_d) r_i dt.
\]
(34)

PID can be seen as a special case of AFF-RBFNN, we can understand it from two views:

1) When \(\sigma \to \infty\), \(S(Z_d) \to 0\). The property of the local response of RBFNNs vanishes to zero and the controller (34) is degraded to the PID controller and the
approximation ability of RBFNNs degrades to just can approximate constants or horizon line. With the decrease of the value of $\sigma$, the local response is increasing, and the approximation performance is also improved. The controller transit from PID to the adaptive RBFNNs controller.

2 Considering the simplest circumstance in which the desired state $q_d = c$, $\dot{q}_d = 0$, $\ddot{q}_d = 0$, there is only need one hidden node to achieve the approximation and the position of the hidden node is set in the position of the desired state $Z_d$. In this circumstance $S(Z_d) = 1$ and the simplest AFF-RBFNN is the same as PID. We can also get an interesting but a little weird conclusion that the integral term in PID can achieve the same PE condition as the AFF-RBFNN controller. The time $T$ of the persistent excitation in the (1) is arbitrarily small, which is indirectly proved by the exponential stability in [48], [49], and the integral term can approximate the targeted dynamics.

V. SIMULATION

Three simulations which are PID, model-based feedforward control (MBFF) control, and adaptive feedforward neural network control have been carried out on the 2-DOF robot manipulator adopting from Section 3.6.1 of [2] to verify the effectiveness of the proposed controller. The initial states are $q_1 = q_2 = 0$ and $\dot{q}_1 = \dot{q}_2 = 0$. The desired trajectories are $q_{d1} = \sin(t)$ and $q_{d2} = \cos(t)$. The control gains are $K_1 = [10, 0; 0, 6]$ and $K_2 = [3, 0; 0, 1.8]$. The step size of those simulations is constant which is 0.01s. The above parameters are the same in the three simulations.
In the case of PID, we recast the format as \( \tau = K_2r + K_I \int r \). The control gain \( K_I = [0.05, 0.05] \). MBFC is recast as \( \tau = K_2r + M(q_d)q_d + C(q_d, q_d)q_d + G(q_d) \). As a baseline controller, the dynamics parameters is known precisely.

In terms of AFF-RBFNN controller, the number of hidden nodes is 20. The step size is 0.01s, so the input vectors of RBFNN (the desired states) can be decided in priori. Then the distribution of hidden nodes is decided by the K-means algorithm. In term of the adaptation law, \( \Gamma_1 = 2.4, \Gamma_2 = 2.4 \). The widths of RBFs are \( \sigma_1 = 1.1 \) and \( \sigma_2 = 1.2 \).

**Remark 4.** In this simulation, the number of hidden nodes is reduced to 20 by introducing the feedback control structure and utilizing the K-means algorithm to distribute the hidden node optimally.

PID and MBFC are selected as the baseline. The tracking performances of the manipulator system in the three controllers are given in Figs. [1] and [2]. From the simulation result of \( 0 - 100s \), we can see that the PID can achieve the basic tracking performance, the tracking errors of MBFC converge to the small intervals quickly. The convergence speed of AFNNC is slower than its in MBFC and the tracking errors of AFNNC controller is worse than its in MBFC in the time of \( 0 - 100s \); however, after enough time learning, the tracking errors of AFF-RBFNN controller can achieve better tracking performance than it in MBFC.

The approximation performances of the AFF-RBFNN controller can be shown in Figs. [3]. The approximation errors converge to the small intervals. we also have met a problem why the AFF-RBFNN controller can achieve better tracking performance.
performance than MBFC? To explain this problem, we should consider the closed-loop system carefully. The controller is discrete, and the robot can be seen as a continuous system, so there are also exiting error leading by sampling time. In MBFC, the error is not considered but the error can be approximated by the AFF-RBFNN controller. This is the reason why the AFF-RBFNN controller can over the MBFFC controller. This also bring a problem how to represent the approximation error of RBFNN, $WS - M(q_1)\dot{q}_1 + C(q_1, \dot{q}_1)\dot{q}_1 + G(q_1)$ or $K_2r^r$. The closed-loop can be recast as $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = K_2r^r + WS(q_1, \dot{q}_1, \ddot{q}_1)$. The prefect circumstance is impossible to achieve for discrete controller. $K_2^r$ can reflect the approximation error for the system. From the Figs. 2 we can seen that the tracking error represented by $K_2^r$ is smaller than its represented by $WS - M(q_1)\dot{q}_1 + C(q_1, \dot{q}_1)\dot{q}_1 + G(q_1)$ or $K_2r^r$. This is means that, in fact $K_2^r$ is more suitable to represent the approximation error of the closed-loop system rather than $WS - M(q_1)\dot{q}_1 + C(q_1, \dot{q}_1)\dot{q}_1 + G(q_1)$.

Remark 5. The value $M_{AE_1}$ of MBFFC in Table I is not zeros because even though the precise dynamics is utilized in the controller, the controller is discrete. There are still having errors by discrete errors. We also have other engineering tools to reduce the discrete errors, but this paper just uses a simple way because we aim to show the excellent approximation performance of our scheme.

A qualitative comparison of performance indices of the stable stage (19000 s – 20000 s) among the three controllers is given in Table I. Both the control performance and approximation of PID in the stable stage are worst considerably among the three methods. Compared with MBFFC, the approximation error of the AFFNCC is improved about 4 times for link 1 and the tracking error is improved about 4.5 times, respectively. In the case of link 2, compared with MBFFC, the approximation error of the AFFNCC is improved about 1.9 times for and the tracking error is improved about 2.5 times, respectively. It indicates that the improved approximation performances are linear correction with the tracking performances. However, it is not rigorous linear because the tracking performance also is influenced by control gains.

VI. CONCLUSION

This paper solved the demerits 2) and 3) that the standard PE can be guaranteed and the number of hidden nodes is linear increase with the complexity of the desired state trajectory rather than the exponential growth with the increase of the dimension of the input vectors. In addition, we articulated that PID is the special case of adaptive feedforward RBFNN control for the set points tracking problem and we named the controller is enhanced PID. It is very easy tuning our algorithm which just more complex than PID slightly and the tuning experience of PID can be easily transferred to our scheme. In the case of the controller implemented by digital equipment, the RBFNN even can remedy the sampling error which is brought by the sampling frequency. Simulations results demonstrated the excellent performance, in which the tracking performance is better than it in model-based nonlinear feedforward control after enough time to learn. The paper is a significant extension of the deterministic learning theory.

### Table I: Comparisons of performance indices for three adaptive RNNFs controllers from 90 s – 100 s

| Controller | $M_{AE_1}$ | $M_{AE_2}$ | $M_{AE_1}$ | $M_{AE_2}$ |
|------------|------------|------------|------------|------------|
| PID        | 1.32       | 0.0432     | 0.374      | 0.0327     |
| MBFFC      | 0.00209    | 0.0000685  | 0.000581   | 0.000513   |
| AFFNCC     | 0.000522   | 0.0000153  | 0.003010   | 0.000208   |

| 1,2 $M_{AE_1}$ and $M_{AE_2}$: the maximum absolute approximate error with respect to links 1 and 2, respectively. 2,3 $M_{AE_1}$ and $M_{AE_2}$: the maximum absolute tracking error with respect to links 1 and 2, respectively. |

### References

[1] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, Stable adaptive neural network control. Springer Science & Business Media, 2002, vol. 13.

[2] S. S. Ge and C. J. Harris, Adaptive Neural Network Control of Robotic Manipulators. World Scientific, 1998.

[3] E. Arabi, T. Yuceelen, B. C. Gorenwold, M. Fravolini, S. Balakrishnan, and N. T. Nguyen, “A neuroadaptive architecture for model reference control of uncertain dynamical systems with performance guarantees,” Systems & Control Letters, vol. 125, pp. 37–44, 2019.

[4] W. He and Y. Dong, “Adaptive fuzzy neural network control for a constrained robot using impedance learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 4, pp. 1174–1186, 2018.

[5] J. E. Slotine and W. Li, “Composite adaptive control of robot manipulators,” Automatica, vol. 25, no. 4, pp. 509–519, 1989.

[6] ——, “On the adaptive control of robot manipulators,” The International Journal of Robotics Research, vol. 6, no. 3, pp. 49–59, 1987.

[7] R. M. Sanner and J. E. Slotine, “Gaussian networks for direct adaptive control,” IEEE Transactions on Neural Networks, vol. 3, no. 6, pp. 837–863, Nov 1992.

[8] S. Ge, C. Hang, and T. Zhang, “A direct method for robust adaptive nonlinear control with guaranteed transient performance,” Systems & control letters, vol. 37, no. 5, pp. 275–284, 1999.

[9] S. S. Ge and Cong Wang, “Adaptive neural control of uncertain mimo nonlinear systems,” IEEE Transactions on Neural Networks, vol. 15, no. 3, pp. 674–692, May 2004.

[10] S. S. Ge and C. Wang, “Adaptive neural control of uncertain mimo nonlinear systems,” IEEE Transactions on Neural Networks, vol. 15, no. 3, pp. 674–692, May 2004.

[11] D. Li, W. Zhang, W. He, C. Li, and S. S. Ge, “Two-layer distributed formation- containment control of multiple euler–lager systems by output feedback,” IEEE transactions on cybernetics, vol. 49, no. 2, pp. 675–687, 2018.

[12] M. M. Polycarpou, “Stable adaptive neural control scheme for nonlinear systems,” IEEE Transactions on Automatic Control, vol. 41, no. 3, pp. 447–451, March 1996.

[13] J. Yu, P. Shi, W. Dong, B. Chen, and C. Lin, “Neural network-based adaptive dynamic surface control for permanent magnet synchronous motors,” IEEE transactions on neural networks and learning systems, vol. 26, no. 3, pp. 640–645, 2014.

[14] W. He, S. S. Ge, Y. Li, E. Chew, and Y. S. Ng, “Neural network control of a rehabilitation robot by state and output feedback,” Journal of Intelligent & Robotic Systems, vol. 80, no. 1, pp. 15–31, 2015.

[15] S. G. Shuzhi, C. C. Hang, and L. Woon, “Adaptive neural network control of robot manipulators in task space,” IEEE transactions on industrial electronics, vol. 44, no. 6, pp. 746–752, 1997.

[16] H. Du, H. Shao, and P. Yao, “Adaptive neural network control for a class of low-triangular-structured nonlinear systems,” IEEE Transactions on Neural Networks, vol. 17, no. 2, pp. 510–514, 2006.

[17] J. Fei and H. Ding, “Adaptive sliding mode control of dynamic system using rbf neural network,” Nonlinear Dynamics, vol. 70, no. 2, pp. 1563–1573, 2012.
[18] S. Zhang, P. Yang, L. Kong, W. Chen, Q. Fu, and K. Peng, “Neural networks-based fault tolerant control of a robot via fast terminal sliding mode,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1–11, 2019.

[19] Y. Zhao and J. A. Farrell, “Locally weighted online approximation-based control for nonlinear systems,” IEEE Transactions on Neural Networks, vol. 18, no. 6, pp. 1709–1724, 2007.

[20] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, “Adaptive neural control for output feedback nonlinear systems using a barrier lyapunov function,” IEEE Transactions on Neural Networks, vol. 21, no. 8, pp. 1339–1345, 2010.

[21] K. P. Tee, S. S. Ge, and E. H. Tay, “Barrier lyapunov functions for the control of output-constrained nonlinear systems,” Automatica, vol. 45, no. 4, pp. 918–927, 2009.

[22] Y.-J. Liu and S. Tong, “Barrier lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints,” Automatica, vol. 64, pp. 70–75, 2016.

[23] Cong Wang and D. J. Hill, “Learning from neural control,” IEEE Transactions on Neural Networks, vol. 17, no. 1, pp. 130–146, Jan 2006.

[24] Y. Pan and H. Yu, “Biomimetic hybrid feedback feedforward neural-network learning control,” IEEE transactions on neural networks and learning systems, vol. 28, no. 6, pp. 1481–1487, 2016.

[25] F. L. Lewis, “Neural network control of robot manipulators,” IEEE Expert, vol. 11, no. 3, pp. 64–75, 1996.

[26] Chae An, C. Atkeson, J. Griffiths, and J. Hollerbach, “Experimental evaluation of feedback and computed torque control,” in Proceedings. 1987 IEEE International Conference on Robotics and Automation, vol. 4, Mar. 1987, pp. 165–168.

[27] P. K. Khosla and T. Kanade, “Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators,” The International Journal of Robotics Research, vol. 7, no. 1, pp. 18–28, Feb. 1988.

[28] F. Reyes and R. Kelly, “Experimental evaluation of model-based controllers on a direct-drive robot arm,” Mechatronics, vol. 11, no. 3, pp. 267–282, 2001.

[29] X. Ren, F. L. Lewis, and J. Zhang, “Brief paper: Neural network compensation control for mechanical systems with disturbances,” Automatica, vol. 45, no. 5, pp. 1221–1226, May 2009.

[30] W. Chen, L. Jiao, and J. Wu, “Globally stable adaptive robust tracking control using rbf neural networks as feedforward compensators,” Neural Computing and Applications, vol. 21, no. 2, pp. 351–363, 2012.

[31] Y. Pan, Y. Liu, B. Xu, and H. Yu, “Hybrid feedback feedforward: An efficient design of adaptive neural network control,” Neural Networks, vol. 36, pp. 122–134, 2012.

[32] X. Dong, C. Wang, and W. Si, “Identification of distributed parameter system with recurrent trajectory via deterministic learning,” in 2017 36th Chinese Control Conference (CCC), Jul 2017, pp. 1557–1562.

[33] T. Zheng and C. Wang, “Relationship between persistent excitation levels and rbf network structures, with application to performance analysis of deterministic learning,” IEEE Transactions on Cybernetics, vol. 47, no. 10, pp. 3380–3392, Oct. 2017.

[34] M. Wang, C. Wang, P. Shi, and X. Liu, “Dynamic learning from neural control for strict-feedback systems with guaranteed predefined performance,” IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 12, pp. 2564–2576, Dec. 2016.

[35] M. Wang and C. Wang, “Learning from adaptive neural dynamic surface control of strict-feedback systems,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 6, pp. 1247–1259, Jun. 2015.

[36] R. Ortega, J. A. L. Perez, P. J. Nicklasson, and H. J. Sira-Ramirez, Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications. Springer Science & Business Media, 2013.

[37] S. Agarwal, S. Yadav, and K. Singh, “Notice of violation of ieee publication principles: K-means versus k-means ++ clustering technique,” in 2012 Students Conference on Engineering and Systems, Mar 2012, pp. 1–6.

[38] A. J. Kurdila, F. J. Narcowich, and J. D. Ward, “Persistency of excitation in identification using radial basis function approximants,” SIAM J. Control Optim., vol. 33, no. 2, p. 625642, Mar. 1995. [Online]. Available: https://doi.org/10.1137/03330292232555

[39] D. Gorinevsky, “On the persistency of excitation in radial basis function network identification of nonlinear systems,” IEEE Transactions on Neural Networks, vol. 6, no. 5, pp. 1237–1244, Sep 1995.

[40] Songwu Lu and T. Basar, “Robust nonlinear system identification using neural-network models,” IEEE Transactions on Neural Networks, vol. 9, no. 3, pp. 407–429, May 1998.

[41] B. Xian, D. M. Dawson, M. S. de Queiroz, and J. J. Chen, “A continuous asymptotic tracking control strategy for uncertain nonlinear systems,” IEEE Transactions on Automatic Control, vol. 49, pp. 1206–1211, Jul 2004.

[42] M. S. De Queiroz, Jun Hu, D. M. Dawson, T. Burg, and S. R. Donepudi, “Adaptive position/force control of robot manipulators without velocity measurements: theory and experiment,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 27, no. 5, pp. 796–809, Sep 1997.

[43] N. Sadegh and R. Horowitz, “Stability and robustness analysis of a class of adaptive controllers for robotic manipulators,” The International Journal of Robotics Research, vol. 9, no. 3, pp. 74–92, Jun 1990.

[44] H. Khalil, Nonlinear Systems, ser. Pearson Education. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[45] J. A. Farrell, “Stability and approximator convergence in nonparametric nonlinear adaptive control,” IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 1008–1020, Sep. 1998.

[46] K. S. Narendra and A. M. Annaswamy, Stable adaptive systems. Englewood Cliffs, N.J: Prentice Hall, 1989.

[47] S. Sastry and M. Bodson, Adaptive control: stability, convergence, and robustness. Englewood Cliffs, N.J: Prentice Hall, 1989.

[48] P. Rocco. “Stability of pid control for industrial robot arms,” IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 606–614, Aug 1996.

[49] J. Alvarez-Ramirez, J. Cervantes, and R. Kelly, “Pfd regulation of robot manipulators: stability and performance,” Systems & Control Letters, vol. 41, no. 2, pp. 73 – 83, Oct 2000.