Nonparametric Control Algorithms for Nonlinear Fading Memory Systems

S. Sandilya and S. R. Kulkarni

***Abstract*—We develop an algorithm to control an unknown nonlinear fading memory discrete-time system. Our approach is based on nonparametric regression techniques rather than traditional feedback control. We discuss a procedure that, given a desired periodic output and a tolerance, produces an acceptable output from any system in a wide class. The algorithm uses as data past inputs (which are selected by the algorithm) and corresponding (possibly noisy) output observations and needs no extra parametric knowledge or restrictions on the system. We also present an algorithm that produces an output that converges to the desired output when stricter conditions are imposed on the system. Our approach, in its current form, however, cannot be used to control open-loop unstable systems.**

***Index Terms*—Control, discrete time, estimation, fading memory, learning, nonparametric estimation, system identification.**

II. Introduction

We address the problem of controlling an unknown discrete time nonlinear system that satisfies only generic conditions like continuity, and asymptotic time invariance and causality. Given this unknown system, a desired periodic output and a tolerance, we wish to arrive at an input that will produce an output within the desired range. We are free to choose the input sequence and may observe the output sequence. We propose an algorithm that achieves the above objective for any system within a wide class. We also present modifications of the above algorithm that work in situations wherein the output observations are corrupted by additive noise. These algorithms require no knowledge of the system other than generic regularity conditions. Given more knowledge of the system, we show that it is possible to obtain asymptotically good control, meaning that we can get the output of the system to converge to the desired periodic waveform.

There has been significant interest in the application of ideas from nonparametric statistics to the fields of system identification and control. Connections between the fields of PAC learning and system identification are explored in [1], [2]. The identification algorithms in [1] are based on the canonical smooth estimators developed in [3], [4]. The procedures we present here are reminiscent of techniques used in nonparametric regression estimation. These ideas have been applied to output prediction [5] and system identification [6]–[11]. Reference [5] deals with a system that is driven by a predetermined input, and the problem is to estimate the current output of the system upon observing all the past inputs and outputs and the current input. A similar approach for predicting time series is considered in [12]. In the setup of [5], one is not interested in identifying the system as the output needs to be predicted only for one input sequence. In order to achieve the objective of system identification, the system must be fed a sufficiently rich input sequence that can excite all the modes of the system. Complexity issues in identification from an estimation perspective are considered in [13]. A comprehensive survey of identification can be found in [14]. The topic of this note is perhaps most closely related to two approaches from this literature. References [15], [6]–[8], [16], [17], [9], [10], [18], and [11] deal with identifying a system from a class when the input is a stationary or i.i.d. random process. These papers consider the identification of restricted classes of systems, typically cascade models like Hammerstein or Wiener systems. There has also been interest in the theoretical limits of identification algorithms in a worst case scenario [19], [20]. The problem we consider in this note is in between the above two cases of prediction and identification. It is not sufficient for us to be able to predict the output of a single input sequence, but it is sufficient to predict the output for a set of input sequences such that one member of this set produces a desirable output.

Another body of work that is closely related is that of model (in)validation with an objective. Control based model invalidation [21]–[23], for instance, deals with the design of robust controllers that achieve a prespecified objective for a class of systems that is believed to contain an adequate description of the given system. If the controller designed does not achieve this objective, the model class is discarded (invalidated) and if the controller achieves the desired performance, it is employed. So the interest in identification is only as far as it helps design controllers. A similar problem is considered with smoothing as the performance criterion in [24] and a chapter in [25].

Most previous work on controlling unknown systems has focused on using feedback control. In most situations, system identification techniques are used wherein it is assumed that the system belongs to a class that is completely determined by a set of parameters that are estimated and these estimates are used to construct controllers. Here, we are interested in a nonparametric approach to the problem. We assume no prior model on the system, except that it is defined by a mapping that is continuous and asymptotically causal and time invariant. Our focus is on exploring the connections between nonparametric regression and control, and on achievability results in this general setting. Our algorithms are not a practical substitute for traditional feedback control owing to the high computational and space complexities involved. The complexity of our algorithms grows unboundedly as the number of training intervals increases. Computing even the expected complexity of our procedure may be intractable as the number of training intervals depends on the distribution of the additive noise. As the main focus of this note is not efficient algorithms, a detailed analysis of our algorithm may not be worthwhile in the current context. The approach we adopt does not alter the input–output relationship of the system, and hence, unlike feedback control, cannot be used in its present form to control open-loop unstable systems. Reference [26] studies limits on the performance of nonparametric estimation based feedback schemes to control such systems.

For the sake of simplicity of presentation, we only consider the case of single input single output systems with the outputs being real numbers and the inputs restricted to a compact subset of the reals. The approach we present, however, can easily be extended to the case wherein the system has multiple outputs and the inputs lie in any compact metric space.

III. Formulation

We consider the problem of producing a desired periodic output from a given unknown discrete time system. We denote the system under consideration by a map (that represents its input–output relationship) $H\colon [-B, B]^\infty \to R^\infty$ [where for a set A, $A^\infty = \{(u_0, u_1, \ldots)\colon u_i \in A, i \geq 0\}$]. The input–output relationship is given by

$$y = H(u)$$

$$z = y + e$$

Manuscript received November 22, 1999; revised August 25, 2000, September 29, 2000, and January 15, 2001. Recommended by Associate Editor X. Zhou. This work was supported in part by the National Science Foundation under NYI Grants IRI-9457645 and ECS-9873451.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA.

Publisher Item Identifier S 0018-9286(01)06609-0.

where

y output produced by the system;

z output observation;

e additive noise $(u,\, y,\, e \in R^{\infty})$.

We define the projection operator for a set $S \subseteq N$, $P_S\colon R^{\infty} \to R^{|S|}$ in the natural way.

$$P_S(u) = (u_{i_1},\, u_{i_2},\, \ldots) \quad \text{where } i_1 < i_2 < \cdots \text{ and } \{i_1,\, i_2,\, \ldots\} = S.$$

If $S = \{n\}$, we denote P_S by P_n, and if $S = \{i \text{ such that } m \leq i \leq n\}$, then we denote P_S by $P_{[m,\,n]}$.

We consider situations wherein the output observations are noiseless $(e = 0)$ and when they are corrupted by i.i.d. additive noise.

We impose the following conditions on the system H:

1) continuity: $\forall\, \epsilon \exists \gamma(\epsilon)$ such that $\|u - v\|_{\infty} < \gamma(\epsilon) \Rightarrow \|H(u) - H(v)\|_{\infty} < \epsilon$;
2) fading memory: $\forall\, \epsilon \exists L = L(\epsilon)$ and $T = T(\epsilon)$ such that $m,\, n \geq T$ and $P_{[m-L+1,\,m]}v = P_{[n-L+1,\,n]}u \Rightarrow |P_n H(u) - P_m H(v)| \leq \epsilon$.

Condition 1) is necessary in order to make any inferences about the input–output relationship of the system based on the output observations for a specific set of inputs. Condition 2) guarantees that the system is eventually causal and time invariant, forgets initial conditions, etc. It encompasses several commonly studied classes of systems such as causal asymptotically stable LTI systems, and Wiener and Hammerstein systems.

IV. The Noiseless Scenario

In this section, we look at the case of the problem wherein the output observations are uncorrupted by noise. Though this case is not of as much interest as the noisy case, it merits attention as it lays out the intuition behind our approach to the problem. The basic idea behind our algorithms is control via prediction. We construct an output predictor for the given system that is very similar to the predictor studied in [5]. In the prediction problem [5], the aim is to predict the output just for a single input sequence. However, in the control problem, we need to be able to predict the output for a class of input sequences that is rich enough so that at least one sequence from this class would produce an acceptable output (one that is within the range specified). We partition time into learning intervals wherein a predetermined input is fed to the system and the outputs observed are used to make inferences about the system's input–output relationship, and control intervals wherein the knowledge gained in the learning intervals is used to drive the output to the desired output. In order to derive a predictor that is good enough, we ensure that a sufficiently rich class of inputs is fed to the system during training/learning intervals. The outputs observed during these intervals are then used to construct a predictor. We pick the input sequence in the above class such that its predicted output is the closest to the target. If the predicted output satisfies the given output constraints, then we feed the chosen sequence as input to the system during what we term as the control intervals. If at any stage in these control intervals, the output falls outside the specified range, we then choose a richer class and redo the training procedure, predictor construction, etc. The following lemma, which shows that to track periodic output sequences with a strictly positive tolerance, the class of periodic input sequences is rich enough, greatly simplifies the procedure for picking the sequence of increasingly richer classes in which the candidate inputs lie.

Lemma 1: Given a periodic sequence y^r with period Q in the output space of the system such that there exists an input sequence u^* in the domain of the system with $\lim_{n\to\infty} |P_n H(u^*) - y_n^r| = 0,\, \forall\, \epsilon > 0$, there is a periodic input u^{p*} such that $\limsup_{n\to\infty} |P_n H(u^{p*}) - y_n^r| < \epsilon$.

Proof: There exists an input u^* such that $|P_n H(u^*) - y_n^r| \to 0$. Pick γ such that

$$\|u - v\| < \gamma \Rightarrow \|H(u) - H(v)\| < \epsilon/3$$

and the pair $(L,\, T_1)$ such that for all $m,\, n > T_1$

$$P_{[m-L+1,\,m]}v = P_{[n-L+1,\,n]}u \Rightarrow |P_n H(u) - P_m H(v)| \leq \epsilon/3.$$

Pick T_2 such that for all $n > T_2 |P_n H(u^*) - y_n^r| < \epsilon/3$. Let $T = \max\{T_1,\, T_2\}$. Construct a γ-cover for $[-B,\, B]$ denoted $C_\gamma[-B,\, B]$. Let $N_C(\gamma)$ denote the number of elements in the γ-cover $C_\gamma[-B,\, B]$. We may assume that $C_\gamma[-B,\, B]$ is minimal, so that $N_C(\gamma)$ is the γ-covering number of $[-B,\, B]$ For each $u_k^* k \geq T$, pick the closest member of $C_\gamma[-B,\, B]$ and assign it to $u_k^{\gamma *}$. Let

$$\begin{aligned} m = \min \big\{n > T \text{ such that } & P_{[n-L+1,\,n-1]}(u^{\gamma *}) \\ &= P_{[n-qQ-L+1,\,n-qQ-1]}(u^{\gamma *}) \\ &\text{for some } q > 0,\, n - qQ > T\big\}. \end{aligned}$$

We have that m is finite because $(C_\gamma[-B,\, B])^L$ is a finite set. Let

$$\begin{aligned} s = \min \big\{q > 0 \text{ such that } & P_{[m-L+1,\,m-1]}(u^{\gamma *}) \\ &= P_{[m-qQ-L+1,\,m-qQ-1]}(u^{\gamma *})\big\}. \end{aligned}$$

Note that $s \leq (N_C(\gamma))^L$. Let $u_k^{p*} = u_k^{\gamma *}$ for $m - sQ + 1 \leq k \leq m$ and let u^{p*} be a periodic extension of the above assignment with period sQ. We claim that u^{p*} is an input that produces an output in the required range asymptotically.

$$\begin{aligned} &|P_n H(u^{p*}) - y_n^r| \\ &\quad \leq |P_n H(u^{p*}) - P_k H(u^{\gamma *})| + |P_k H(u^{\gamma *}) - P_k H(u^*)| \\ &\qquad + |P_k H(u^*) - y_n^r|. \end{aligned} \tag{1}$$

We know that $\forall\, n > T,\, \exists k \geq T,\, k = n - qQ$ such that $P_{[n-L+1,\,n]}(u^{p*}) = P_{[k-L+1,\,k]}(u^{\gamma *})$ and $|P_k H(u^{\gamma *}) - P_k H(u^{p*})| < \epsilon/3$.

Hence, as y is periodic with period Q, $y_k^r = y_n^r$, so each of the terms on the right hand side of (1) is less than $\epsilon/3$ and, therefore, $|P_n(H(u^{p*})) - y_n^r| < \epsilon$. □

We established in the above lemma that there exists a γ such that in order to get within ϵ of the target output, it is sufficient to search among all periodic inputs whose value at each time instant lies in a γ cover of $[-B,\, B]$. As the system is unknown, we do not know the values of γ or L that guarantee the required performance. However, we know that any smaller γ and larger L than those chosen in the proof of the lemma will provide an input that is "good." Hence, we define increasingly rich classes over which we conduct our search. We define classes parameterized by k that become richer as k increases. In particular, we pick monotonic sequences $\gamma_k \to 0$ and $L_k \to \infty$ corresponding to which, the sequence of input classes is

$$\mathcal{U}_k = \Big\{u \in (C_{\gamma k}[-B,\, B])^{\infty}: \; u \text{ is periodic with period at most } Q N_C(\gamma_k)^{L_k}\Big\}.$$

We know that for some $(\gamma,\, L)$, there exists a periodic input that works and for sufficiently large k, $\gamma_k < \gamma$ and $L_k > L$ and hence there exists a periodic input in $\mathcal{U}_k$ that keeps the output in the required range. When a particular k is found unsatisfactory, we increment it. We denote by k_n the k in use at time n and by T_k the least n for which $k_n = k$. As the candidate inputs we consider lie only in $\mathcal{U}_k$, we construct a predictor whose domain is vectors of length L_k in $C_{\gamma k}[-B,\, B]$. The set of input vectors for which we wish to predict the input is finite and we get to observe the output uncorrupted by noise. Hence, in order to construct such a predictor, we just ensure that every vector in this set is input to the system and use the observed output thereof as the prediction. We then pick from among those inputs whose value at each time instant lies in $C_{\gamma k}[-B,\, B]^{L_k}$ the best one, i.e., that input whose predicted output is closest to the desired.

In the noiseless case, we predict the output of a sequence as the observation at the first time the vector was input during the last training interval. As the observations are not corrupted, the predictor gets more accurate with the passage of time and increasing k (with increasing k, the quantization of the input space gets finer and the predictor has a longer memory) . The predictor is given by the following equation:

$$p_n^k(x_1, \ldots, x_{L_k}) = y_m \tag{2}$$

where $m = \inf\{n \geq T_k + L_k$ such that $(u_{n-L_k+1}, \ldots, u_n) = (x_1, \ldots, x_{L_k})\}$.

A. Tracking Algorithm

1) Pick monotonic sequences $\gamma_k \in R_+$, and $L_k \in N$ such that $\gamma_k \to 0$ and $L_k \to \infty$. Set $k_0 = 0$, $T_0 = 0$.
2) Increment k, set $T_k = n$.
3) Construct a γ_k-cover for $[-B, B]$ denoted $C_{\gamma_k}[-B, B]$.
4) Input to the system a string u^{γ_k} such that $\forall x \in (C_{\gamma_k}[-B, B])^{L_k}$, $\exists m \geq T_k$ s.t. $P_{[m, m+L_k-1]}(u^{\gamma_k}) = x$. This ensures that the predictor is a well-defined function with domain $(C_{\gamma_k}[-B, B])_k^L$.
5) Let $u^c = \arg\min_{\{u \in \mathcal{U}_k\}} \sup_m |p_n^k(u_{m-L_k+1}, \ldots, u_m) - y_m^r|$.
6) If $\sup_n |p_m^k(u_{m-L_k+1}^c, \ldots, u_m^c) - y_m^r| > \epsilon$, then jump to step 2).
7) Input to the system u^c. If for any n, $|y_n - y_n^r| > \epsilon$, then jump to step 2).

Proposition 1: For any $\epsilon > 0$, the above algorithm converges to an input u^p such that

$$\limsup_{n\to\infty} |P_n H(u^p) - y_n^r| \leq \epsilon.$$

Proof: If the output ever strays from the desired range, the algorithm conducts the training procedure for the next richer input class. So the algorithm fails only if the training intervals are infinitely many. From the proof of Lemma 1, it directly follows that once $n > T(\epsilon/3)$, $L_k > L(\epsilon/3)$, and $\gamma_k < \gamma(\epsilon/3)$, $|P_n H(u^c) - y_n^r| \leq \epsilon$, and so the output for input u^c stays in the required range for all n after that. $L_k \to \infty$ and $\epsilon_k \to 0$ imply that the above requirements are met in a finite amount of time, after which the training procedure is never undertaken. □

V. The Noisy Scenario

This section deals with strategies for the more interesting situation in which the output observations are corrupted by noise that is potentially large but has some structural properties that can be exploited. We are concerned with the scenario in which the noise is "good" in the sense that though it can be large infinitely often, it is small on the average. Hence, a natural extension of the noiseless algorithm would be to enlarge the learning intervals and apply the same inputs many times so that we could take advantage of this "good" behavior of the noise on the average.

Unlike the noiseless case, we cannot simply use one output observation as the predicted output. Here, we look at all the times in the past that the vector of inputs has occurred since the beginning of the last training interval, and use the average of these outputs as the prediction. As the magnitude of the noise samples is not consistently large, the averaging smooths it out and produces an acceptable prediction. The predictor is given by

$$p_n^k(x_1, \ldots, x_{L_k}) = \frac{\sum_{i=T_k+L_k}^{n} z_i 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}. \tag{3}$$

A. Tracking Algorithm

1) Pick monotonic sequences $\gamma_k \in R_+$, and $L_k, N_k \in N$ such that $\gamma_k \to 0$ and $L_k, N_k \to \infty$. Set $k_0 = 0$, $T_0 = 0$.
2) Increment k, set $T_k = n$.
3) Construct a γ_k-cover for $[-B, B]$ denoted $C_{\gamma_k}[-B, B]$.
4) Input to the system a string u_{γ_k} such that $\forall x \in (C_{\gamma_k}[-B, B])^{L_k}$, $\exists m \geq T_k$ s.t. $P_{[m, m+L_k-1]}(u_{\gamma_k}) = x$. Repeat this N_k times.
5) Let $u^c = \arg\min_{\{u \in \mathcal{U}\}} \sup_m |p_n^k(u_{m-L_k+1}, \ldots, u_m) - y_m^r|$.
6) If $\sup_m |p_n^k(u_{m-L_k+1}^c, \ldots, u_m^c) - y_m^r| > 2\epsilon/3$, then jump to step 2).
7) Input to the system u^c. If for any n, $|p_n^k(u_{n-L_k+1}^c, \ldots, u_n^c) - y_n^r| > 2\epsilon/3$, then jump to step 2).

The tracking algorithm is almost the same as the noiseless case. The only changes are in the length of the learning intervals (to incorporate the averaging in the predictor) and in the criterion for accepting and rejecting candidate inputs. We accept an input only if it produces a predicted output within $2\epsilon/3$ of the desired output. For the algorithm to accept a "bad" input (one that produces an output more than ϵ away from the desired), the noise would have to corrupt the output prediction by at least $\epsilon/3$. This allows the algorithm to tolerate noise that has a mean as large as $\epsilon/3$. By changing the acceptance criterion, we can tolerate noise that has a mean arbitrarily close to $\epsilon/2$.

Theorem 1: If the noise is i.i.d with absolute value of mean less than $\epsilon/3$, then there exists a sequence N_k such that almost surely the training procedure stops in a finite amount of time and the input settles on a periodic sequence whose nominal output is within ϵ of the target.

Proof: Define

$$\mathcal{U}_k^1 = \{u \text{ such that } P_{[n,\infty)} u \in \mathcal{U}_k: |P_m H(u) - y_n^r| < \epsilon/3 \ \forall m > n\}$$

$$\mathcal{U}_k^2 = \left\{u \text{ such that } P_{[n,\infty)} u \in \mathcal{U}_k: \limsup_m |P_m H(u) - y_n^r| > \epsilon\right\}.$$

When $\gamma_k < \gamma(\epsilon/9)$, $n > T(\epsilon/9)$ and $L_k > L(\epsilon/9)$, $\mathcal{U}_k^1 \neq \emptyset$.

The algorithm fails only if

1) $k_n \to \infty$;
2) it settles on $u \in \mathcal{U}_k^2$ for some $k = \lim_n k_n$.

We show that for judiciously chosen N_k, each of these events has probability zero. Hence, almost surely the complement of the above event occurs i.e., the algorithm succeeds and settles on a good input.

1) Consider k, n large enough as discussed above.

$$\begin{aligned} P[k_n = k+1 \text{ for some } n] &= P[T_{k+1} < \infty] \\ &\leq P\left[\cup_{u \in \mathcal{U}_k^1} \{|p_n^k(u_{n-L_k+1}, \ldots, u_n) - y_n^r| > 2\epsilon/3\}\right] \end{aligned}$$

$$\begin{aligned} p_n^k(x_1, \ldots, x_{L_k}) &= \frac{\sum_{i=T_k+L_k}^{n} z_i 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}} \\ &= \frac{\sum_{i=T_k+L_k}^{n} y_i 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}} \\ &\quad + \frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i) = (x_1, \ldots, x_{L_k})\}}}. \end{aligned} \tag{4}$$

Hence, for $u \in \mathcal{U}_k^1$,

$$|p_n^k(u_{n-L_k+1}, \ldots, u_n) - y_n^r| < \epsilon/3 + \frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}}.$$

Substituting the above in (4), we have

$$P[T_{k+1} < \infty] \le P\left[\bigcup_{u \in \mathcal{U}_k^1} \left\{ \left| \frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}} \right| > \epsilon/3 \right\}\right]$$

$$\le \sum_{x \in (C_{\gamma_k}[-B, B])^{L_k}} \cdot P\left[\left\{ \left| \frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(x_1, \ldots, x_{L_k})\}}} \right| > \epsilon/3 \right\}\right]$$

$$\le \sum_{x \in (C_{\gamma_k}[-B, B])^{L_k}} \exp(-\rho m(x))$$

$$\le \exp(-\rho N_k)(N_C(\gamma_k))^{L_k}.$$

As $\{T_{k+1} < \infty \; \forall k > K\} \subseteq \limsup_k \{T_{k+1} < \infty\}$, we have

$$\sum_{k=K}^{\infty} P[T_{k+1} < \infty] \le \sum_{k=K}^{\infty} \exp(-\rho N_k)(N_C(\gamma_k))^{L_k} \triangleq S.$$

If $S < \infty$ then almost surely $T_{k+1} = \infty$ for some k (by the Borel–Cantelli Lemma). This may be ensured, for instance, by choosing $N_k = (L_k \ln N_C(\gamma_k) + 2 \ln k)/\rho$.

2) Suppose $k = \lim_n k_n$ and the chosen input $u \in \mathcal{U}_k^2$. By the strong law of large numbers, we have that almost surely

$$\frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(x_1, \ldots, x_{L_k})\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(x_1, \ldots, x_{L_k})\}}} \to E(e_0) < \epsilon/3.$$

For $u \in \mathcal{U}_k^2$

$$|p_n^k(u_{n-L_k+1}, \ldots, u_n) - y_n^r| > \epsilon - \frac{\sum_{i=T_k+L_k}^{n} e_i 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}}{\sum_{i=T_k+L_k}^{n} 1_{\{(u_{i-L_k+1}, \ldots, u_i)=(u_{n-L_k+1}, \ldots, u_n)\}}}$$

$$\Rightarrow \liminf_n |p_n^k(u_{n-L_k+1}, \ldots, u_n) - y_n^r| > \epsilon - \epsilon/3 = 2\epsilon/3.$$

Hence, $P[T_{k+1} = \infty \,\&\, u \in \mathcal{U}_k^2] = 0$.

$$P[\text{Algorithm fails}] \le P[k_n \to \infty] + P[\cup_k \{T_{k+1} = \infty \;\&\; u \in \mathcal{U}_k^2\}] \le P[k_n \to \infty] + \sum_k P[T_{k+1} = \infty \;\&\; u \in \mathcal{U}_k^2] = 0.$$

□

VI. Asymptotically Good Control

In the previous sections, we outlined a strategy that eventually achieved the objective of controlling the given system to any prespecified tolerance ϵ. Now we focus on the problem of asymptotically good control wherein we desire that the nominal output converge to the desired output. Let the set of good inputs be denoted A: good inputs are those whose outputs converge to the desired i.e., $u \in A$ if $|P_n H(u) - y_n^r| \to 0$. In order that the algorithm ensure convergence, we need to restrict our attention to progressively smaller neighborhoods of the inputs in this set. Previously, it was sufficient to pick the input in a γ-cover such that its output as predicted from past observations best fitted the desired output. We showed that the input chosen by this algorithm eventually ensured control with an ϵ-tolerance. However, it may not be true that a sufficiently small neighborhood of this input contains an input from A. Hence, it may not be possible to restrict our attention to such neighborhoods and achieve our goal.

The procedure we adopt here is, hence, slightly different. We assume that there exists a periodic input in A. Moreover, we assume greater knowledge of the system in terms of the (ϵ, δ) relationship in the continuity condition and the $(\epsilon, (L, T))$ relationship in the fading memory condition. As the periodic input in A must have a period that is a multiple of Q, we search for "good" inputs among those with periods that are progressively larger multiples of Q. For any particular input period lQ, the class of inputs over which the search is conducted is $([-B, B])^{lQ}$ which is a totally bounded metric space.

As before, our algorithm centers around construction of finer and finer covers for this set. However, due to the fact that we use a different search procedure, we arrange them hierarchically. First, we construct a γ_1 cover for $([-B, B])^{lQ}$ Then, for each k, a γ_k cover is constructed for the γ_{k-1} neighborhood of each entry in the γ_{k-1}-cover. Evidently, for each k, the union of all these covers forms a γ_k-cover for the entire input space $([-B, B])^{lQ}$. We now organize these inputs into a tree wherein an input sequence x in the γ_k-cover is connected to another input sequence x' in the γ_{k-1}-cover if x is in the γ_k-cover constructed for the neighborhood of x'. We label the edges emanating from each node by distinct positive integers so that we may now label any node in this tree at depth k by a string of integers of length k signifying the labels of the edges that lead from the root node to the desired node (such a path is unique as this is a tree, but there may be multiple nodes representing the same input). We denote the input corresponding to the node $(i_1, \ldots, i_k)$ by $x^{(i_1, \ldots, i_k)}$. Following standard terminology, we say that node $(i_1, \ldots, i_k)$ is the parent of node $(i_1, \ldots, i_k, i_{k+1})$ and the latter is a child of the former. We define descendants and ancestors of a node as the natural extension of the parent–child relationship.

A. Algorithm

The algorithm just consists of conducting a depth first search on the tree constructed as described above. The input corresponding to a node is rejected if it produces an output such that we believe that no good input could be in its neighborhood. If an input is not rejected, then we proceed to examine the inputs corresponding to its children (which form a cover for a neighborhood of the current input). Here we present only the algorithm used to control the system in the presence of observation noise.

We assume that the following functions are known:

$$\epsilon'(\gamma) = \sup_{\|x-y\|<\gamma} \|H(x) - H(y)\|$$

$$\epsilon''(L, T) = \sup_{n>T} \sup_{x} \sup_{y \in K(x)} |P_n H(x) - P_n H(y)|$$

$$K(x) = \left\{y: P_{[n-L+1, n]}(x) = P_{[n-L+1, n]}(y)\right\}.$$

The knowledge of these functions allows us to determine at the present γ, L, and T, how close the algorithm should get to the desired output, and hence, on observing a deviant output, we can declare that the entire γ-neighborhood of the current input has no "good" inputs and focus our attention on other input sequences. The predictor we use here is almost the same as that in the algorithm for ϵ-control. If the current node is $\underline{i}$, we use as the prediction the average of the output over time instants (after node $\underline{i}$ was reached in the current tree) that have the same L_k input history

$$p_n^{\underline{i}}(x_1, \ldots, x_{L_k}) = \frac{\sum_{j=T_{\underline{i}}+L_k}^{n} z_j 1_{\{(u_{j-L_k+1}, \ldots, u_j)=(x_1, \ldots, x_{L_k})\}}}{\sum_{j=T_{\underline{i}}+L_k}^{n} 1_{\{(u_{j-L_k+1}, \ldots, u_j)=(x_1, \ldots, x_{L_k})\}}}.$$

The procedure is as follows.

1) Pick sequences γ_k and L_k as before. Pick a sequence $N_{k,l}$ such that $\lim_k N_{k,l} = \infty$ and $\lim_l N_{k,l} = \infty$.
2) Set $l = 1$.
3) Set $k = 0$, $\underline{i} = ()$.
4) Construct the tree for input space $([-B, B])^{lQ}$.
5) If the current node has not been rejected, pick the edge out of the current node with the smallest label that has not been rejected, say a. Set $\underline{i} = (\underline{i}, a)$. Set $T_{\underline{i}} = n$. Increment k. Else, pick the closest ancestor of the current node that has not been rejected and set $\underline{i}$ to be its label. Jump back to step 5). If the root node has been rejected, jump to step 9).
6) Input to the system $x^{\underline{i}}$ for $L_k + N_{k,l} lQ$ time instants.
7) The node $\underline{i}$ is rejected if the predicted value of the output corresponding to $x^{\underline{i}}$ is more than $3\epsilon'(\gamma_k) + 2\epsilon''(L_k, T_{\underline{i}})$ away from y^r.
8) A node is rejected if any of its ancestors has been rejected or if all its children have been rejected.
9) If the root node is rejected, then increment l and goto step 3), else jump to step 5).

Using arguments very similar to the previous case, we can show that the above algorithm achieves the asymptotically good control objective.

Theorem 2: If there exists a periodic input u^* such that $|P_n H(u^*) - y_n^r| \to 0$, the functions ϵ' and ϵ'' are known, and if the additive noise is i.i.d. with zero mean and finite variance, then there exists a sequence $N_{k,l}$ such that the algorithm above results in asymptotically good control.

REFERENCES

[1] M. C. Campi and P. R. Kumar, "Learning dynamic systems in a stationary environment," *Syst. Control Lett.*, pp. 125–32, 1998.

[2] L. Ljung, "PAC learning and a esymptotic identification theory," in *Proc. 35th IEEE Conf. Decision Control*, 1996, pp. 2303–2307.

[3] K. L. Buescher and P. R. Kumar, "Learning and canonical smooth estimation—Part I: Simultaneous estimation," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 545–556, Apr. 1996.

[4] ——, "Learning and coninical smooth estimation—Part II: Learning and choice of model complexity," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 557–69, Apr. 1996.

[5] S. R. Kulkarni and S. E. Posner, "Nonparametric output prediction for nonlinear fading memory systems," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 29–37, Jan. 1996.

[6] W. Greblicki, "Nonparametric identification of wiener systems by orthogonal series," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 2077–86, Oct. 1994.

[7] W. Greblicki and M. Pawlak, "Dynamic system identification with order statistics," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1474–1489, Sept. 1994.

[8] ——, "Cascade nonlinear system identification by a nonparametric method," *Int. J. Syst. Sci.*, pp. 129–153, 1994.

[9] A. Krzyzak, "On estimation of a class of nonlinear systems by the kernel regression estimate," *IEEE Trans. Inform. Theory*, vol. 36, pp. 141–152, Jan. 1990.

[10] ——, "Identification of nonlinear systems by recursive kernel regression estimates," *Int. J. Syst. Sci.*, vol. 24, pp. 577–598, Mar. 1993.

[11] L. Rutkowski, "On system identification by nonparametric function fitting," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 225–227, Feb. 1982.

[12] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Physical Review Letters*, pp. 845–848, 1987.

[13] G. Zames and J. G. Owen, "A note on metric dimension and feedback in discrete time," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 664–667, Apr. 1993.

[14] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice-Hall, 1987.

[15] A. A. Georgiev, "A nonparametric algorithm for identification of linear dynamic SISO systems of unknown order," *Syst. Control Lett.*, pp. 273–80, 1984.

[16] I. W. Hunter and M. J. Korenberg, "The identification of nonlinear biological systems: Wiener and Hammerstein cascade models," *Biol. Cybern.*, pp. 135–44, 1986.

[17] M. J. Korenberg and I. W. Hunter, "The identification of nonlinear biological systems: LNL cascade models," *Biol. Cybern.*, pp. 125–34, 1986.

[18] M. Pawlak, "On the series expansion approach to the identification of Hammerstein systems," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 763–767, June 1991.

[19] M. A. Dahleh, E. D. Sontag, D. N. C. Tse, and J. N. Tsitsiklis, "Worst-case identification of nonlinear fading memory systems," *Automatica*, pp. 503–508, 1995.

[20] D. N. C. Tse, M. A. Dahleh, and J. N. Tsitsiklis, "Optimal asymptotic identification under bounded disturbances," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1176–1190, Aug. 1993.

[21] M. A. Dahleh and J. C. Doyle, "From data to control," in *The Modeling of Uncertainty in Control Systems*. New York: Springer-Verlag, 1994.

[22] M. A. Dahleh and M. M. Livstone, "A unified framework for identification and control," in *Feedback Control, Nonlinear Systems, and Complexity*. New York: Springer-Verlag, 1995.

[23] M. M. Livstone, M. A. Dahleh, and J. A. Farrell, "A framework for robust control based model invalidation," in *Proc. Amer. Control Conf.*, 1994, pp. 3017–3020.

[24] P. L. Bartlett and S. R. Kulkarni, "The complexity of model classes and smoothing noisy data," *Syst. Control Lett.*, pp. 133–140, 1998.

[25] M. Vidyasagar, *A Theory of Learning and Generalization*. New York: Springer-Verlag, 1997.

[26] L. L. Xie and L. Guo, "On the capability of feedback for controlling uncertain nonlinear systems," in *Proc. IEEE Conf. Decision Control*, 1999, pp. 2833–2838.