Recursive lazy learning for modeling and control

Gianluca Bontempi, Mauro Birattari, and Hugues Bersini

Iridia - CP 194/6
Universitè Libre de Bruxelles
50, av. Franklin Roosevelt
1050 Bruxelles - Belgium
email: gbonte,mbiro,bersini@ulb.ac.be
http://iridia.ulb.ac.be

Abstract. This paper presents a local method for modeling and control of non-linear dynamical systems, when only a limited amount of input-output data is available. The proposed methodology couples a local model identification inspired by the lazy learning technique, with a control strategy based on linear optimal control theory. The local modeling procedure uses a query-based approach to select the best model configuration by assessing and comparing different alternatives. A new recursive technique for local model identification and validation is presented, together with an enhanced statistical method for model selection. The control method combines the linearization provided by the local learning techniques with optimal linear control theory, to control non linear systems in far from equilibrium configurations. Simulation results of the identification of a non-linear benchmark model and of the control of a complex non-linear system (the bioreactor) are presented.

1 Introduction

In this paper we present a local method to model and control an unknown dynamical system starting from a limited amount of input-output data. Local techniques are an old idea in time series prediction (Farmer & Sidorowich, 1987), classification (Cover & Hart, 1967) and regression (Cleveland, 1979). The idea of local approximators as alternative to global models originated in non-parametric statistics (Epanechnikov, 1969), (Benedetti, 1977) to be later rediscovered and developed in the machine learning field (Aha, 1989), (Bottou & Vapnik, 1992). Recent work on lazy learning (a.k.a memory-based or instance-based learning) gave a new impetus to the adoption of local techniques for modeling (Atkeson *et al.*, 1997a) and control problem (Atkeson *et al.*, 1997b).

Our approach extends the idea of local learning in several directions. First, we propose a model identification methodology based on the use of an iterative optimization procedure to select the best local model among a set of different candidates. Modeling a non-linear mapping using a limited number of observations requires the data analyst to make several choices involving the set of relevant variables and observations, the model structure, the learning algorithm, and the validation protocol. Our method defers all of these decisions until a prediction or a local description is requested (query-based approach). In classical methods the several options of a local model are designed according to heuristic criteria and a priori assumptions. Here we propose an automatic tuning procedure which searches for the optimal model configuration, by returning for each candidate model its parameters and a statistical description of its generalization properties.

The second contribution of the paper is the introduction of a new algorithm to estimate in a recursive way the model performance in cross-validation. Myers (1990) introduced the PRESS statistic which is a simple, well-founded and economical way to perform leave-one-out cross validation (Efron & Tibshirani, 1993) and to assess the performance in generalization of local linear models. Here we propose a technique based on recursive least squares methods to compute the PRESS in an incremental way. Moreover, a powerful and well-founded statistical test is used to compare the performance of two alternative candidates on the basis of their cross-validation error sampling distributions.

The third contribution of the paper is a non-linear control design technique, which uses extensively analysis and design tools imported from linear control. The idea of employing linear techniques in a non-linear setting is not new in control literature but had recently a new popularity thanks to methods for combining multiple estimators and controllers in different operating regimes of the system (Murray-Smith & Johansen, 1997). Gain scheduling (Shamma & Athans, 1992), fuzzy inference systems (Takagy & Sugeno, 1985), local model networks (Johansen & Foss, 1993) are well-known examples of control techniques for non linear systems inspired to linear control. However, two strong assumptions underlie linearization control methods: an analytical description of the equilibrium points locus is available and the system is supposed to evolve in a sufficiently restricted neighborhood of the desired regime. Here we propose an indirect control method for performing finite-time horizon control which requires only a limited amount of input-output data from the observed system behavior. The controller is designed with optimal control techniques parameterized with the values returned by the linear local estimator. The idea is that a combination of a local estimator with a time varying optimal control can take into account the non-linearity of a system over a wider range than conventional linearized quadratic regulators (LQR).

The remainder of the paper is organized as follows. In section 2 we will introduce our modeling technique based on an iterative selection procedure. In section 2.1 we present the recursive algorithm for PRESS statistic computation. In section 2.2 the method for statistical comparison between models assessed by the recursive PRESS evaluation is introduced. Details on the algorithm for local optimal control can be found in section 3. Finally, in section 4 examples of identification and of control of two simulated systems are given.

2 Local modeling as an optimization problem

Modeling from data involves integrating human insight with learning techniques. In many real cases, the analyst faces a situation where a limited set of data is available and an accurate prediction is required. Often, information about the order, the structure or the set of relevant variable is missing or not reliable. The process of learning consists of a trial and error procedure during which the model is properly tuned on the available data. In the lazy learning approach, the estimation of the value of the unknown function is solved giving the whole attention to the region surrounding the point where the estimation is required. The classical non-adaptive memory-based procedure essentially consists of these steps:

- for each query point x_q, defining a set of neighbors, each weighted according to some relevance criterion (e.g. the distance)
- choosing a regression function f in a restricted family of parametric functions estimating the local weighted regression
- computing the regression value $f(x_q)$.

The data analyst who adopts a local regression approach, has to take a set of decisions related to the model (e.g. the number of neighbors, the weight function, the parametric family, the fitting criterion to estimate the parameters). We extend the classical approach with a method that automatically selects the adequate configuration. To this aim, we simply import tools and techniques from the field of linear statistical analysis. The most important of these tools is the PRESS statistic (Myers, 1990), which is a simple, well-founded and economical way to perform *leave-one-out* cross validation (Efron & Tibshirani, 1993) and therefore to assess the performance in generalization of local linear models. Due to its short computation time which allows its intensive use, it is the key element of our approach to modeling data. In fact, if PRESS can assign a quantitative performance to each linear model, alternative models with different configurations can be tested and compared in order to select the best one. This same selection strategy is indeed exploited to select the training subset among the neighbors, as well as various structural aspects like the features to treat and the degree of the polynomial used as a local approximator. The general ideas of the approach can be summarized in the following way.

1. The task of learning an input output mapping is decomposed in a series of linear estimation problems
2. Each single estimation is treated as an optimization problem in the space of alternative model configurations
3. The estimation ability of each alternative model is assessed by the cross-validation performance computed using the PRESS statistic.

The core operation of the algorithm consists in assessing and comparing local models having different configurations. Each assessment requires the PRESS computation and the comparison with the other candidate models. In order to make these operations more effective we propose two innovative algorithms in the lazy learning method:

- a recursive algorithm for the parametric estimation and the cross-validation of each local model. This method avoids having to restart each model evaluation from scratch and decreases noticeably the computational cost
- a more rigorous statistical test to compare the performance of two alternative candidate models. The test does not consider only the average values of the cross-validation errors but also their sampling distributions.

In the next two sections we will discuss these algorithms in detail.

2.1 The PRESS statistic and the recursive method

To illustrate the local regression procedure, we will first define some notation. Let us consider an unknown input-output mapping $f\colon \mathbb{R}^d \rightarrow \mathbb{R}$ of which we are given a set of N samples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$. These examples can be collected in an input matrix X of dimensionality $[N \times d]$, and in an output vector y of dimensionality $[N \times 1]$.

Given a specific query point x_q, the prediction of the value $y_q = f(x_q)$ is computed as follows. First, for each sample (x_i, y_i), a weight w_i is computed as a function of the distance $d(x_i, x_q)$ from the query point x_q to the point x_i. Each row of X and y is then multiplied by the corresponding weight creating the variables $Z = WX$ and $v = Wy$, with W diagonal matrix having diagonal elements $W_{ii} = w_i$. Finally, a local model is fitted solving the equation $(Z^T Z)\beta = Z^T v$ and the prediction of the value $f(x_q)$ is obtained evaluating such a model in the query point $\hat{y}_q = x_q^T (Z^T Z)^{-1} Z^T v$.

An important aspect of local learning is that along with the prediction and with the model parameters, an assessment of the performance can be easily computed. We will focus on the *leave-one-out* cross-validation procedure (Efron & Tibshirani, 1993), which returns a reliable estimation of the prediction error in x_q. We define the i-th *leave-one-out* error $e^{\mathrm{cv}}(i)$ as the difference between y_i and the prediction given by the local model centered in x_q and fitted using all the examples available but the i-th. Hence, an estimation of the prediction error in x_q is given by the average of the errors $e^{\mathrm{cv}}(i)$ each weighted according to the respective distance $d(x_i, x_q)$. When considering local linear model, the *leave-one-out* cross-validation can be performed without recalculating the regression parameter for each excluded example thanks to the local version of the PRESS statistic (Atkeson *et al.*, 1997a):

$$\begin{aligned} \mathrm{MSE}^{\mathrm{cv}}(x_q) &= \frac{1}{\sum_i w_i^2} \sum_i \left(\frac{v_i - z_i^T (Z^T Z)^{-1} Z^T v}{1 - z_i^T (Z^T Z)^{-1} z_i} \right)^2 \\ &= \frac{1}{\sum_i w_i^2} \sum_i \left(w_i \, e^{\mathrm{cv}}(i) \right)^2 \ . \end{aligned} \tag{1}$$

In our modeling procedure the performance of a model in cross validation is the criterion adopted to choose the best local model configuration. One of the most important parameters to be tuned in a local model configuration is the size of the region surrounding x_q, in which the function f can be conveniently approximated by a linear local model. Such a parameter can be defined by the number of training examples which fall into the region of linearity. Consequently the task of identifying the region of linearity can be reconducted to the task of finding, among the examples available, the number n of neighbors of x_q to be used in the local regression fit. Thus, we consider different models, each fitted on a different number of examples, and we use the *leave-one-out* cross-validation to compare them and to select the one for which the predicted error is smaller.

To make the procedure faster and to avoid repeating for each model the parameter and the PRESS computation we adopt an incremental approach, based on recursive linear techniques. Recursive algorithms have been developed in model identification and adaptive control literature (Goodwin & Sin, 1984) to identify a linear model when data are not available from the beginning but are observed sequentially. Here we employ these methods to obtain the parameters of the model fitted on n nearest neighbors by updating the parameters of the model with $n-1$ examples. Also, the *leave-one-out* errors $e^{\mathrm{cv}}(i)$ are obtained exploiting partial results from the least square method and do not require additional computational overload. Once adopted as the weighting kernel the indicator function which assigns $w_i = 1$ to the examples used to fit the model, the recursive lazy algorithm is described by the following equations:

$$\begin{cases} P(n+1) = P(n) - \dfrac{P(n)x(n+1)x(n+1)^T P(n)}{1 + x(n+1)^T P(n)x(n+1)} \\ \gamma(n+1) = P(n+1)x(n+1) \\ e(n+1) = y(n+1) - x(n+1)^T\beta(n) \\ \beta(n+1) = \beta(n) + \gamma(n+1)e(n+1) \end{cases} \tag{2}$$

$$e^{\mathrm{cv}}_{n+1}(i) = \frac{y(i) - x(i)^T\beta(n+1)}{1 + x(i)^T P(n+1)x(i)}$$

where, $\big(x(n+1), y(n+1)\big)$ is the $n+1$-th nearest neighbor of the query point, $P(n)$ is the recursive approximation of the matrix $(Z^T Z)^{-1}$, $\beta(n)$ denotes the optimal least squares parameters of the model fitted on the n nearest neighbor, and $e^{\mathrm{cv}}_n(i)$, with $1 \le i \le n$, is the vector E^{cv}_n of *leave-one-out* errors. Once this vector is available, the formula (1) is easily computed. This value is a weighted average of the cross-validated errors and is the simplest statistic that can be used to describe the performance of the model defined by n neighbors. However, the problem of assessing the right dimension of the linearity region using a limited number of samples affected by noise requires a more powerful statistical procedure. In the following section, we will discuss in detail the method used in our model selection procedure.

2.2 The statistical test for model selection

The recursive method described in the previous section returns for each size n of the neighborhood a vector E^{cv}_n of *leave-one-out* errors. In order to select the best model, our procedure (described in detail in Fig. 1) consists in increasing the number of neighbors considered when identifying the local model, until the model performance deteriorates and a departure from the region of local linearity is detected. This requires a statistical test to evaluate when the enlarged model is significantly worse than those already considered. In terms of hypothesis testing,

1. $n = 0$. Initialize the parameter of the local model (e.g. conventional initialization: $\beta = 0$).
2. Add the example $\big((x(n+1), y(n+1)\big)$. The parameter β is updated and the vector E_n^{cv} is evaluated using Eq. (2).
3. Check for a departure from the local linear region comparing the new model with the one previously accepted. If the new model is significantly worse goto 4, else accept the the model and goto 2 to consider adding further examples.
4. Stop the recursive identification procedure and return the parameters of the last model accepted.

Fig. 1. The procedure adopted to identify recursively local linear model and to define the largest region of linearity.

we formulate the null hypothesis H_0 that E_n^{cv} and E_{n+1}^{cv} belong to the same distribution. To evaluate this hypothesis we use a permutation test (Siegel & N.J. Castellan, 1988) which does not require any assumptions about normality, homogeneity of variance, or about the shape of the underlying distribution. We adopt a paired version of the permutation algorithm because of the correlation between the two error vectors.

In simple words, the procedure consists in computing for each pair of errors the difference $d_i = e_n^{\mathrm{cv}}(i) - e_{n+1}^{\mathrm{cv}}(i)$, and computing the value $D = \sum d_i$ which is assumed to be an instance of the random variable D^*. The sampling distribution of D^* is found by a randomization procedure (Cohen, 1995), a computer-intensive statistical method to derive the sampling distribution of a statistic by simulating the process of sample extraction. In the permutation test, this is done by creating a high number of pseudo-samples D^b, with $b = 1, \ldots, B$, derived from the actual sample D by substituting randomly a difference d_i with $-d_i$. Ones the sampling distribution of D^* is generated, a one-tailed test determines whether the null hypothesis has to be rejected.

The randomization test shows one of the main advantage of a local modeling procedure: with low computational effort it is possible to return along with the prediction and the linear local parameters also a statistical description of the uncertainty affecting these results. This property can result useful both for prediction and for control problems.

3 Lazy learning optimal control

Although non-linearity characterizes most real control problems, methods for analysis and control design are considerably more powerful and theoretically founded for linear systems than for non-linear ones. Here we propose a hybrid architecture for the indirect control of non linear discrete time plants from their observed input-output behavior. This approach combines the local learning identification procedure described in the previous section with control techniques borrowed from conventional linear optimal control

Consider a class of discrete time dynamic systems whose equations of motion can be expressed in the form

$$\begin{aligned} y(k) = f\big(y(k-1), \ldots, y(k-ny),& \\ u(k-d), \ldots, u(k-d-nu),& \\ e(k-1), \ldots, e(k-ne)\big) + e(k)& \ , \end{aligned} \tag{3}$$

where $y(k)$ is the system output, $u(k)$ the input, $e(k)$ is a zero-mean disturbance term, d is the relative degree and $f(\cdot)$ is some non linear function. This model is known as the NARMAX model (Leontaritis & Billings, 1985). Let us assume we have no physical description of the function f but a limited amount of pairs $\big[u(k), y(k)\big]$ from the observed input-output behavior. Defining the information vector

$$\begin{aligned} \varphi(k-1) = \big[y(k-1), \ldots, y(k-ny),& \\ u(k-d), \ldots, u(k-d-nu),& \\ e(k-1), \ldots, e(k-ne)\big]& \ , \end{aligned} \tag{4}$$

the system (3) can be written in the input-output form $y(k) = f\big(\varphi(k-1)\big) + e(k)$.

Consider the optimal control problem of a non linear system over a finite horizon time. Using a quadratic cost function, the solution to an optimal control problem is the control sequence U that minimizes

$$J = \frac{1}{2} y(t_f)^T P(t_f) y(t_f) + \frac{1}{2} \sum_k \left| \, y(k)^T u(k)^T \, \right| \begin{vmatrix} Q_k & M_k \\ M_k^T & R_k \end{vmatrix} \left| \, y(k) u(k) \, \right| \tag{5}$$

with Q_k, M_k, R_k, P_f weighting terms designed a priori. While analytic results are not available for a generic non linear configuration, optimal control theory (Stengel, 1986) provides the solution for the linear case. Hence, we will now present the non linear problem in a linear time varying setting.

Consider the trajectory of the dynamical system once forced by an input sequence $U = \big[u(1), u(2), \ldots, u(t_f)\big]$. Assume that the system can be linearized about each state of the trajectory. Neglecting the residual errors due to the first order Taylor series approximation, the behavior of the linear system along a generic trajectory is the behavior of a linear time varying system whose state equations can be written in the form

$$\begin{aligned} y(k+1) &= A\big(\varphi(k)\big)\,y(k) + B\big(\varphi(k)\big)\,u(k) + K\big(\varphi(k)\big) \\ &= A_k y(k) + B_k u(k) + K_k \end{aligned} \tag{6}$$

with A_k, B_k, K_k parameters of the system linearized about the query point $\varphi(k)$. K_k is an offset term that equals zero in equilibrium points. This term requires a slight modification in the linear controller formulation. However, in order to simplify the notation, in the following we will neglect the constant term.

Optimal control theory provides the solution for the linear time varying system (6). At each time step the optimal control action is

$$u(k) = -(R_k + B_k^T P_{k+1} B_k)^{-1}(M_k^T + B_k^T P_{k+1} A_k)y(k) \tag{7}$$

where Pk is the solution to the backward Riccati equation.

$$\begin{aligned} P_k &= Q_k + A_k^T P_{k+1} A_k \\ &\quad - (M_k + A_k^T P_{k+1} B_k)(R_k + B_k^T P_k + 1 B_k)^{-1}(M_k^T + B_k^T P_{k+1} A_k) \end{aligned} \tag{8}$$

having as final condition

$$P(t_f) = P_f \tag{9}$$

The piecewise-constant optimal solution is obtained by solving the Euler-Lagrange equations, the three necessary and sufficient conditions for optimality when the final time is fixed.

$$0 = \frac{\partial H_k}{\partial u_k} = y_k^T M_k + u_k^T R_k + \lambda_{k+1}^T B_k \tag{10}$$

$$\lambda_k^T = \frac{\partial H_k}{\partial y_k} = y_k^T Q_k + u_k^T M_k^T + \lambda_{k+1}^T A_k \tag{11}$$

$$\lambda_f^T = y_f^T P_f \tag{12}$$

with $\lambda_k = P_k y_k$ adjoint term in the augmented cost function (Hamiltonian)

$$H_k = J + \lambda_{k+1}^T\big(A_k y(k) + B_k u(k)\big) \tag{13}$$

The Euler-Lagrange equations do not hold for non linear systems. Anyway, if the system can be represented in the form (6), formula (10) can be used to compute the derivative of the cost function (5) with respect to a control sequence U. This requires at each time k the matrices A_k, B_k that can be obtained by linearizing the system dynamics along the trajectory forced by the input sequence.

As discussed in section 2, our modeling procedure performs system linearization with minimum effort, no a priori knowledge and only a reduced amount of data. Hence, we propose an algorithm for non linear optimal control, formulated as a gradient based optimization problem and based on the local system linearization.

The algorithm searches for the sequence of input actions

$$U^{opt} = arg \min_{U^i} J(U^i) \tag{14}$$

that minimizes the finite-horizon cost function (5) along the future t_f steps. The cost function $J(U^i)$ for a generic sequence U^i is computed simulating forward for t_f steps the model identified by the local learning method. The gradient of $J(U^i)$ with respect to U^i is returned by (10).

These are the basic operations of the optimization procedure (described in detail in Fig. 2) executed each time a control action is required:

- forward simulation of the lazy model forced by a finite control sequence U^i of dimension t_f
- linearization of the simulated system about the resulting trajectory
- computation of the resulting finite cost function $J(U^i)$
- computation of the gradient of the cost function with respect to simulated sequence
- updating of the sequence with a gradient based algorithm.

Once the search algorithm returns U^{opt}, the first action of the sequence is applied to the real system (receding horizon control strategy (Clarke, 1994)). Let us remark how the lazy learning model has a twofold role in the algorithm

1. Initialization of the algorithm with a random sequence of actions U_s^0.
2. Forward simulation of the system forced by the sequence $U_s^i = \left[u_s^i(k), u_s^i(k+1), \dots, u_s^i(k+t_f)\right]$ where $u_s^i(j)$ denotes the action applied to the simulated system at time j. The system behavior is predicted using the model identified by the local learning method.
3. Formulation of the non linear system in the time varying form. The parameters A_j, B_j, K_j $j = k, \dots, k+t_f$ are returned by the local model identification.
4. Backward resolution of the discrete-time Riccati equation (8) for the resulting time varying system.
5. Computation of the cost function (5).
6. Computation of the gradient vector $\frac{\partial J}{\partial U_s^i} = \left[\frac{\partial J}{\partial u_s^i(k)}, \frac{\partial J}{\partial u_s^i(k+1)}, \dots, \frac{\partial J}{\partial u_s^i(t_f)}\right]$ by using formula (10).
7. Updating of the control sequence $U_s^i \rightarrow U_s^{i+1}$. The optimization step is performed by a constrained gradient based algorithm implemented in the Matlab function `constr` (Grace, 1994).
8. If the minimum has been reached ($U_s^i = U_s^{i+1}$) goto 9 else goto 2.
9. Control action execution. The first action $u_s^{opt}(k)$ of the sequence U_s^{opt} is applied to the real system.

Fig. 2. The lazy learning optimal control algorithm

in Fig. 2: (i) at step 2 it behaves as an approximator which predicts the behavior of the system once forced with a generic input sequence (ii) at step 3 it returns a linear approximation to the system dynamics.

Atkeson et al. (1997b) and Tanaka (1995) applied infinite-time LQR regulator to non linear systems linearized with lazy learning and neuro-fuzzy models. The drawback of these approaches is that an equilibrium point or a reference trajectory is required. Also, they make the strong assumption that the state of the system will remain indefinitely in a neighborhood of the linearization point. As discussed above, the advantage of the proposed approach is that these requirements do not need to be satisfied. First, lazy learning is able to linearize a system in points far from equilibrium. Secondly, the time varying approach makes possible the use of a linear control strategy even though the system operates within different linear regimes.

Remark: we make the assumption that the parameters returned by the local models are a real description of the local behavior (certainty equivalence principle). This is a restricting assumption which requires a sufficient degree of accuracy in the approximation. However, we see in the optimal control theory a possible solution to this limitation. In fact, stochastic optimal control theory provides a formal solution to the problem of parameter uncertainty in control systems (dual control (Fel'dbaum, 1965)). Further, our modeling procedure can return at no additional cost a statistical description of the estimated parameters (see Section 2.2). Hence, future work will focus on the extension of the technique to the stochastic control case.

4 Experiments

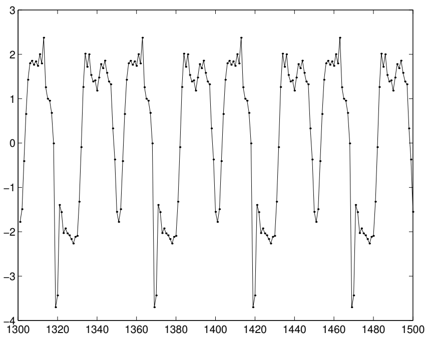
4.1 The identification of a non linear discrete time system

The approach has been applied to the identification of a complex non linear benchmark proposed by (Narendra & Li, 1996). The discrete time equations of the system are:

$$\begin{cases} x_1(k+1) &= \left(\dfrac{x_1(k)}{1+x_1^2(k)}+1\right)\sin\big(x_2(k)\big) \\ x_2(k+1) &= x_2(k)\cos\big(x_2(k)\big)+x_1(k)\,\mathrm{e}^{-\frac{x_1^2(k)+x_2^2(k)}{8}} \\ & \quad +\dfrac{u^3(k)}{1+u^2(k)+0.5\cos\big(x_1(k)+x_2(k)\big)} \end{cases} \tag{15}$$

$$y(k)=\frac{x_1(k)}{1+0.5\sin\big(x_2(k)\big)}+\frac{x_2(k)}{1+0.5\sin\big(x_1(k)\big)}$$

where (x_1, x_2) is the non observable state and only the input u and the output y are accessible. We model the non observable system in the input-output form $y(k+1) = f\big(y(k), y(k-1), y(k-2), y(k-3), u(k)\big)$. We use an initial empty database which is updated all along the identification. We perform the identification for 1500 time steps with a test input $u(k) = \sin\left(\frac{2\pi k}{10}\right) + \sin\left(\frac{2\pi k}{25}\right)$. The plot in Fig. 3a) shows the model and the system output in the last 200 points, while the plot in Fig. 3b) shows the identification error.



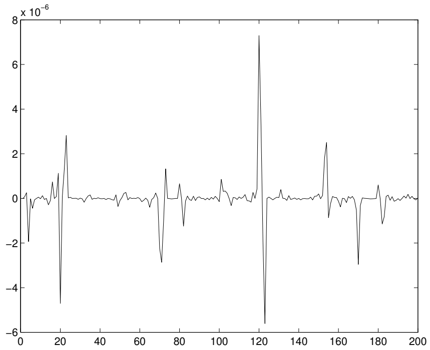


Fig. 3. Non linear system identification results: a) system (solid) and model (dotted) outputs b) identification error

We obtain a good performance in modeling this complex non observable system. These results outperform those obtained by (Narendra & Li, 1996) with a much wider training set of 500,000 points and a complex architecture (4-layer feed-forward neural network).

4.2 The control of the bioreactor

Consider, as second example, the bioreactor system, a well-known benchmark in non linear control (MillerIII *et al.*, 1990). The bioreactor is a tank containing water, nutrients, and biological cells. Nutrients and cells are introduced into the tank where the cells mix with the nutrients. The state of this process is characterized by the number of cells ($c1$) and the amount of nutrients ($c2$). Bioreactor equations of motion are the following:

$$\begin{cases} \frac{dc_1}{dt} & = -c_1 u + c_1(1-c_2)\,\mathrm{e}^{\frac{c_2}{\gamma}} \\ \frac{dc_2}{dt} & = -c_2 u + c_1(1-c_2)\,\mathrm{e}^{\frac{c_2}{\gamma}} \left(\dfrac{1+\beta}{1+\beta-c_2}\right) \end{cases} \tag{16}$$

with $\beta = 0.02$ and $\gamma = 0.48$. In our experiment the goal was to stabilize the multivariable system about the unstable state $(c_1^*, c_2^*) = (0.2107, 0.726)$ by performing a control action each 0.5 seconds.

We use the control algorithm described in section 4. The system is modeled in the input-output form (3) having the orders $ny = 2, nu = 1, ne = 0, d = 1$. The horizon of the control algorithm is fixed to $t_f = 5$. The initial state conditions are set by to the random initialization procedure defined in (MillerIII *et al.*, 1990). We initialize the lazy learning database with a set of 1000 points collected by preliminarly exciting the system with a random uniform input. The database is then updated on-line each time a new input-output pairs is returned by the simulated system. The plot in Fig. 4a) shows the output of the two controlled state variables, while the plot in Fig. 4b)below the control action.

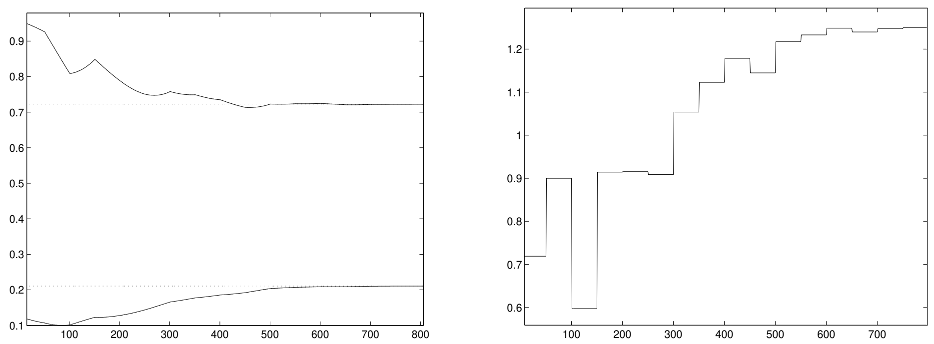


Fig. 4. Control results a) controlled variables (solid) and references (dotted) b) control action

The bioreactor is considered as a challenging problem for its non linearity and because small changes in parameters value can cause the bioreactor to become unstable. These results show how using local techniques it is possible to control complex systems on a wide non linear range, with only a limited amount of points and no a priori knowledge about the underlying dynamics.

5 Conclusions and future developments

Local modeling techniques are a powerful technique for learning from limited amount of data by providing at the same time an useful insight on the local behavior of the modeled system. Furthermore, together with the required prediction and/or parametric description, they return a statistical distribution of the uncertainty affecting this information. We proposed an innovative algorithm to improve the performance of local modeling techniques which is a based on a recursive version of the cross-validation and a statistical model selection.

In control literature local controllers have generally a restricted range of operating conditions. Here, we proposed a controller, which although making extensive use of local techniques, works on an extended range of operating conditions. These characteristics makes of it a promising tool for intelligent control systems, inspired to traditional engineering methods but able to deal with complex non linear systems. Successful applications of the method to simulated identification and control problems were presented. Future developments will concern the combination of the local modeling technique with other certainty equivalence controllers (e.g. minimum variance controller, pole placement) and the extension of the method to stochastic dual control.

References

[Aha, 1989] Aha D.W. 1989. Incremental, instance-based learning of independent and graded concept descriptions. *Pages 387–391 of: Sixth International Machine Learning Workshop.* San Mateo, CA: Morgan Kaufmann.

[Atkeson *et al.*, 1997a] Atkeson C.G. , Moore A.W. & Schaal S. 1997a. Locally weighted learning. *Artificial Intelligence Review*, **11**(1–5), 11–73.

[Atkeson *et al.*, 1997b] Atkeson C.G. , Moore A.W. & Schaal S. 1997b. Locally weighted learning for control. *Artificial Intelligence Review*, **11**(1–5), 75–113.

[Benedetti, 1977] Benedetti J.K. 1977. On the non parametric estimation of regression functions. *Journal of the Royal Statistical Society, Series B*, 248–253.

[Bottou & Vapnik, 1992] Bottou L. & Vapnik V.N. 1992. Local learning algorithms. *Neural Computation*, **4**(6), 888–900.

[Clarke, 1994] Clarke D.W. 1994. *Advances in Model-Based Predictive Control.* Oxford University Press.

[Cleveland, 1979] Cleveland W.S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, **74**, 829–836.

[Cohen, 1995] Cohen P.R. 1995. *Empirical Methods for Artificial Intelligence.* Cambridge, Massachusetts: The MIT Press.

[Cover & Hart, 1967] Cover T. & Hart P. 1967. Nearest neighbor pattern classification. *Proc. IEEE Trans. Inform. Theory*, 21–27.

[Efron & Tibshirani, 1993] Efron B. & Tibshirani R.J. 1993. *An Introduction to the Bootstrap.* New York: Chapman and Hall.

[Epanechnikov, 1969] Epanechnikov V.A. 1969. Non parametric estimation of a multivariate probability density. *Theory of Probability and Its Applications*, 153–158.

[Farmer & Sidorowich, 1987] Farmer J.D. & Sidorowich J.J. 1987. Predicting chaotic time series. *Physical Review Letters*, **8**(59), 845–848.

[Fel'dbaum, 1965] Fel'dbaum A.A. 1965. *Optimal control systems.* New York: Academic Press.

[Goodwin & Sin, 1984] Goodwin G.C. & Sin K. S. 1984. *Adaptive Filtering Prediction and Control.* Prentice-Hall.

[Grace, 1994] Grace Andrew . 1994 (October). *Optimization Toolbox ForUse with MATLAB.* The MATHWORKS Inc.

[Johansen & Foss, 1993] Johansen T.A. & Foss B.A. 1993. Constructing NARMAX models using ARMAX models. *International Journal of Control*, **58**, 1125–1153.

[Leontaritis & Billings, 1985] Leontaritis I.J. & Billings S.A. 1985. Input-output parametric models for non-linear systems. *International Journal of Control*, **41**(2), 303–344.

[MillerIII *et al.*, 1990] MillerIII W.T. , Sutton R.S. & Werbos P.J. (eds) 1990. *Neural Networks for Control.* The MIT Press.

[Murray-Smith & Johansen, 1997] Murray-Smith R. & Johansen T.A. (eds) 1997. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis.

[Myers, 1990] Myers R.H. 1990. *Classical and Modern Regression with Applications*. Boston, MA: PWS-KENT.

[Narendra & Li, 1996] Narendra K.S. & Li S.M. 1996. Mathematical Perspectives on Neural Networks. Lawrence Erlbaum Associates.

[Shamma & Athans, 1992] Shamma J.S. & Athans M. 1992. Gain scheduling: Potential hazards and possible remedies. *IEEE Control Systems Magazine*, June, 101–107.

[Siegel & N.J. Castellan, 1988] Siegel S. & N.J. Castellan Jr. 1988. *Non Parametric Statistics for the Behavioral Sciences*. 2nd edn. McGraw-Hill International.

[Stengel, 1986] Stengel Robert F. 1986. *Stochastic optimal control: theory and application*. New York: John Wiley and Sons.

[Takagy & Sugeno, 1985] Takagy T. & Sugeno M. 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on System, Man and Cybernetics*, **15**(1), 116–132.

[Tanaka, 1995] Tanaka K. 1995. Stability and Stabilizability of Fuzzy-Neural-Linear Control Systems. *IEEE Transactions on Fuzzy Systems*, **3**(4), 438–447.