ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

Lazy learning for control design

Gianluca Bontempi, Mauro Birattari, Hugues Bersini

Iridia - CP 194/6
Université Libre de Bruxelles
1050 Bruxelles - Belgium
email: {gbonte, mbiro, bersini}@ulb.ac.be

Abstract. This paper presents two local methods for the control of discrete-time unknown nonlinear dynamical systems, when only a limited amount of input-output data is available. The modeling procedure adopts *lazy learning*, a query-based approach for local modeling inspired to memory-based approximators. In the first method the *lazy* technique returns the forward and inverse models of the system which are used to compute the control action to take. The second is an indirect method inspired to adaptive control where the self-tuning identification module is replaced by a *lazy* approximator. Simulation examples of control of nonlinear systems starting from observed data are given.

1. Introduction

The idea of local memory-based approximators as alternative to global models originated in non-parametric statistics to be later rediscovered and developed in machine learning (Bottou & Vapnik, 1992). Recent work on *lazy learning* (a.k.a just-in-time learning) gave a new impetus to the adoption of local techniques for modeling (Atkeson *et al.*, 1997a), and control problems (Atkeson *et al.*, 1997b). In the *lazy learning approach*, the value of an unknown mapping is estimated giving the whole attention to the region surrounding the point $\boldsymbol{\varphi}_q$ where the estimation itself is required. The procedure essentially consists of these steps: i) for each query point $\boldsymbol{\varphi}_q$, define a set of neighbors, each weighted according to some relevance criterion (e.g. the distance) ii) choose a local regression function $f(\cdot)$ in a restricted family of parametric functions iii) compute the regression value $\hat{f}(\boldsymbol{\varphi}_q)$. In (Bontempi *et al.*, 1997) we extend the classical memory-based approach with a method that automatically selects the local model configuration. To this aim, we apply tools and techniques from linear statistical analysis to nonlinear modeling problems. The most relevant one is the PRESS statistic (Myers, 1990), a simple, well-founded and economical way to perform *leave-one-out* cross validation (Efron & Tibshirani, 1993) and

The work of Gianluca Bontempi was supported by the European Union TMR Grant FMBICT960692. The work of Mauro Birattari was supported by the F.I.R.S.T. program of the Région Wallonne, Belgium.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

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 this paper we present and compare two methods for discrete-time control based on the local linear description returned by the *lazy learning* model. The idea of employing linear techniques in a nonlinear 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) and local model networks (Johansen & Foss, 1993), are well-known examples of control techniques for nonlinear systems inspired to linear control.

The first method we propose is an example of gradient-based control system which is inspired to neural controllers and combines an inverse with a forward lazy model to select the control action. The algorithm has been introduced by (Atkeson *et al.*, 1997b) and applied to a static control task. Here we test its stability properties on two different dynamic tasks. The second controller is based on the self-tuning regulator (STR) architecture (Astrom & Wittenmark, 1990) and combines discrete-time conventional control (e.g. generalized minimum variance, pole placement) with the lazy local modeling. The adoption of linear control techniques allows the local analysis of the closed-loop systems in terms of stability properties.

In the next section, we will introduce the two control algorithms. In section 3. we will present some simulation results of the control of two nonlinear systems. Some final considerations are summarised in section 4.

2. Lazy learning for control design

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

$$y(k) = f\big(y(k-1), \ldots, u(k-d), \ldots, e(k-1), \ldots\big) + e(k), \qquad (1)$$

where k denotes the time, $y(k)$ is the system output, $u(k)$ the input, $e(k)$ is a zero-mean disturbance term, $d > 0$ is the relative degree and $f(\cdot)$ is some nonlinear function. Let us assume we have no physical description of the function $f(\cdot)$ but a limited amount of pairs $\big[u(k), y(k)\big]$ is available.

2.1. The lazy learning gradient-based controller

The idea of the *lazy* gradient-based controller is to solve a one time-step horizon control problem as an optimization problem. Suppose that the system (1), is required to reach at the next time time step a reference value y_{ref} (for clarity, we assume $d = 1$). The *lazy* model can be used to predict the response of the system to the control action u^i:

$$\hat{y}_{u^i}(k) = \hat{f}\big(y(k-1), \ldots, u^i, u(k-2), \ldots, e(k-1), \ldots\big). \qquad (2)$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

In addition, the linearization returned by the local description provides an estimate of the gradient of the system output $\frac{\mathrm{d}\hat{y}_{u^i}(k)}{\mathrm{d}u^i}$ with respect to the control action (Atkeson *et al.*, 1997a). The control problem can therefore be formulated as a constrained gradient based optimization problem:

$$u^{\mathrm{opt}} = arg \min_{u^i} J(u^i) = arg \min_{u^i} (y_{ref} - \hat{y}_{u^i}(k))^2. \tag{3}$$

In order to speed up the optimization resolution, the algorithm can be initialized with the value returned from the model of the inverse dynamics:

$$u^0 = \hat{f}_{inv}\big(y_{ref}, y(k-1), \ldots, u(k-2), \ldots, e(k-1), \ldots\big). \tag{4}$$

The *lazy* gradient-based control algorithm can be described by the following steps to be repeated at each sampling period:

1. Initialization of the algorithm with the value u^0 provided by the inverse mapping (4).
2. Prediction of the outcome $\hat{y}_{u^i}$ of the system forced by the input u^i and computation of the gradient vector $\frac{\mathrm{d}J(u^i)}{\mathrm{d}u^i}$.
3. Updating of the control sequence $u^i \rightarrow u^{i+1}$. The optimization step is performed by a constrained gradient based algorithm.
4. If the minimum has been reached ($u^i = u^{i+1}$) goto 5 else goto 2.
5. Control action execution.
6. Updating of the database.

2.2. The Lazy learning self-tuning controller

This approach combines the lazy learning identification procedure with control techniques from adaptive linear control (Astrom & Wittenmark, 1990), e.g. minimum-variance and pole-placement. Let us consider a linear discrete-time process described in input-output form by the equation:

$$A(z)y(k) = z^{-d}B(z)u(k) + C(z)e(k), \tag{5}$$

and suppose we want to regulate it to $y_{ref} = 0$. The MV control problem can be stated as finding the control law which minimizes the variance of the output. The MV controlled closed loop system is stable only if B has all of its roots inside the unit circle (minimum phase). However, more complex formulations are available in the case of a tracking problem or in the case of non minimum-phase systems (Generalized MV or GMV). Pole placement design is an alternative technique to deal with non minimum-phase configurations. The procedure requires first to choose the desired closed loop pole positions and then to calculate the appropriate controller. Both these design techniques require a model linearization in the form (5). In our approach this is returned at each time step by the lazy learning model. In detail the control algorithm can be described by the following steps to be repeated at each sampling period:

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

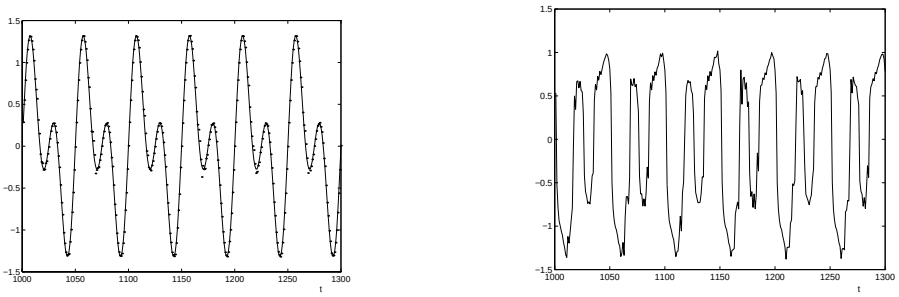

Figure 1: Gradient-based control: a) reference (solid) and system (dotted) outputs b) control action

1. Linearization of the function $f(\cdot)$ computed by the *lazy learning* algorithm.
2. Derivation of the polynomials A, B, C of (5) from the linearized model.
3. Design of a MVG/PP controller for (5) which satisfies the required properties (stability, accuracy, speed ...) of the closed loop behavior.
4. Computation and execution of the control signal.
5. Updating of the database.

3. Simulation studies

3.1. A lazy gradient-based control example

In this simulation we consider the control of the plant described by the example 11.2 in (Narendra & Li, 1996). We use the control algorithm described in section 2.2.. The system is represented in the minimum-phase 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. The system is controlled for 1300 time steps. The plot in Fig. 1a shows the model and the system output in the last 300 points, while the plot in Fig. 1b shows the control action. These results outperform those obtained by (Narendra & Li, 1996) after 2,000,000 steps of on-line adjustments and a complex architecture (4-layer feed-forward neural network).

3.2. A lazy self-tuning control example

In this simulation we consider the control of the nonlinear SISO system described by the difference equation:

$$y(k+1) = \frac{y(k)y(k-1)y(k-2)\big(y(k-2)-1\big)u(k-1)+u(k)}{1+y^2(k-1)+y^2(k-2)}. \qquad (6)$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

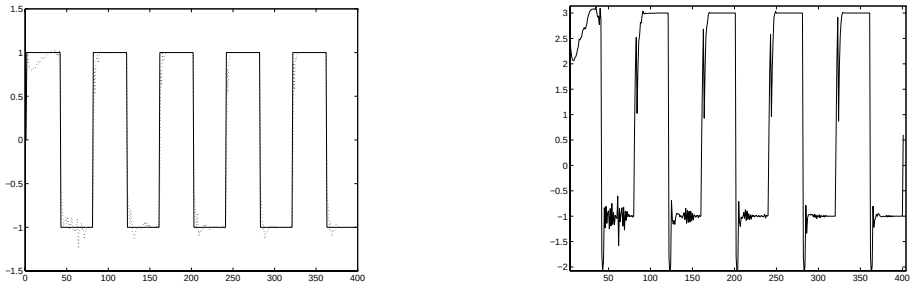

Figure 2: Self-tuning control: a) reference (solid) and system (dotted) outputs b) control action.

The system is represented in the input-output form $y(k+1) = f\big(y(k), y(k-1), y(k-2), u(k), u(k-1)\big)$. The reference output $y_{ref}(k)$ is given by a periodic square wave. The *lazy* nonlinear algorithm is not able to control the system over this reference trajectory. On the contrary, a self-tuning regulator based on a pole placement algorithm is able to track the trajectory. We initialize the *lazy learning* database with a set of 5000 points collected by preliminarly exciting the system with a random uniform input. The database is then updated online each time a new input-output pairs is returned by the simulated system. The plot in Fig. 2a shows the reference and the system output , while the plot in Fig. 2b shows the control action. From the local analysis of the identified system we have that the system is non minimum-phase (i.e. absolute value of the zero greater than one) when y is in the neighborhood of $y = -1$ (e.g. see the time interval $50 - 100$). This is indeed the region where the gradient-based controller fails to control the system by making the feedback loop unstable.

4. Final considerations

We illustrated and tested two control systems, which make an extensive use of local modeling. The *lazy* gradient-based control system, inspired by neural control, makes use of forward and inverse approximations of the system dynamics to select the control action. Like in neuro-control, properties of stability cannot be guaranteed in a general case. However, we showed how this approach can obtain performances more accurate than neural networks even using a smaller set of training examples. The *lazy* self-tuning architecture adopts a linear control technique which eases the analysis in terms of stability properties and provides a useful insight in the dynamic properties of the nonlinear system.

It is worthy noting how in both the approaches we made the assumption of *certainty equivalence*. Future developments will focus on how to deal with parameter uncertainty in local control.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 73-78

References

Astrom K.J. & Wittenmark B. 1990. *Computer-controlled Systems: Theory and Design*. Prentice-Hall International Editions.

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

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

Bersini H. , Birattari M. & Bontempi G. . 1997. *Adaptive memory-based regression methods*. submitted to 1998 IEEE International Joint Conference on Neural Networks.

Bontempi G. , Birattari M. & Bersini H. . 1997. *Recursive lazy learning for modeling and control*. submitted to Tenth European Conference On Machine Learning (ECML-98).

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

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

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

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

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

Narendra K.S. & Li S.M. 1996. Neural Networks in Control Systems. *Chap. 11, pages 347–394 of:* Paul Smolensky M.C. Mozer & Rumelhart D.E. (eds), *Mathematical Perspectives on Neural Networks*. Lawrence Erlbaum Associates.

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

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.