

ADVANCED PREDICTIVE CONTROL FOR COMPLEX PROCESSES

Nicolae Constantin

*Automatic Control and System Engineering Dept.
Politehnica University of Bucharest
Spl. Independentei, 313, 77206, Bucharest, Romania.
Tel. 021-42029974, email: nicu@cib.pub.ro*

Abstract: In this paper a new algorithm for local estimation approximating a function by means of local models is introduced. The instant or in-time estimation provides an attractive alternative for nonlinear identification, since it requires less structural decisions to be made by the user. It represents a hybrid between local and global modeling. This estimation procedure is combined with predictive algorithm with a modification cost function for efficient control. Simulation results are provided to prove the results of the proposed approach.

Keywords: nonlinear predictive control, in-time estimation, multimodels, optimization.

1. INTRODUCTION

Model Predictive Control (MPC) using local models has been explored in the process control literature in the context of fuzzy modeling, artificial neural networks, and other interpolation techniques (Constantin and Dumitrache, 2002). Fuzzy modeling approaches require the user to decide on how to partition the regressor space for good performance at local linear models. For most applications, this is a highly iterative process although a few automated methods have recently emerged.

Neural network based approaches can be tedious since the user must judiciously select good initial parameter estimates. If the initial parameters are successful such that the neural network converges, the user must still decide whether the neural network structure is adequate and repeat the initial parameter/structure selection problem all over again.

The instant estimation provides an attractive alternative for nonlinear identification, since it requires less structural decisions to be made by the user. It represents a hybrid between local and global modeling (Braun, et.al, 2001).

It adaptively select the number of data in the regressor space to build a local model. It continually

constructs new local models as the current operating point moves about the regressor space. As a result, the initialization, optimization and structure selection issues of global semiphysical and neural network modeling are avoided.

The user is not faced with the decisions of how to partition the regressor space, commonly found with other local modeling techniques. In an MPC framework it provides natural local model transition through the regressor space. Other MPC controllers which use local linear models, must decide how to transition between the local linear models. The user need only to determine the Nonlinear Auto Regressive with Exogeneous inputs (NARX) regressor structure, and a lower bound on the number of data to be used in the local models. The best fit global linear ARX structure often provides a good choice for the NARX structure and the lower bound on the number of data to be used for the local model can be determined through validation in a few number of iterations.

The remainder of the paper is organized as follows. In section 2 the multimodel estimation technique based on an iterative selection procedure is presented. Algorithmic details and theoretical analysis for model predictive control based on the

proposed identification method can be found in section 3. Finally, in section 4 a simulation example of an continuous stirred time reactor control is given.

2. MULTIMODEL ESTIMATION

2.1. Global versus local modeling

Global modeling. A structure to be used for the entire database it is first assumed. The structure can be based on linear or nonlinear assumptions, or on first-principles knowledge of the system. In linear regression the parameters of the model have been fit by minimizing the sum of square errors over the entire dataset. Dynamical models as the AutoRegressive with Exogeneous inputs (ARX) which has a linear structure are considered as global models. The idea of global modeling is to compress all the information contained within the data into a few parameters. This produces a simple predictor, at the expense of a biased estimate.

Local modeling. One way to estimate the true value y at the current operating point x is to take the average of the response variables produced by a small neighborhood around the current operating point x . In order to improve the estimate it is possible to give more weight to data closer to the operating point and less weight to data that are farther away. These weights can be chosen based on a kernel function, which explicitly defines the shape of the weights. The shape of these windows and the number of data points resides in the neighborhood have a great influence about the performance of the estimate. The estimation of the response variable from a small portion of the dataset is known as local modeling.

2.2. In-time modeling

The two main issues in local modeling are: deciding how much data to include when a local estimate is made what weighting to give the data in that subset. In-time modeling extends the local modeling in some way. IT estimator performs the data selection and weighting on-line at each current point. As the process moves away from the current operation point, a subset of the database is selected, and a new dynamical model is built.

This estimator has the following advantages:

- the predictions are data-driven rather than structure dependent;
- the predictions are less likely to get trapped by local minima in the optimization step as compared to global modeling;
- new data can be added and old data can be disregarded from the database.

Consider a SISO process with nonlinear ARX structure

$$y(k) = f(\varphi(k)) + e(k), \quad k = 1, \dots, N \quad (1)$$

where $f(\cdot)$ is an unknown nonlinear mapping and $e(k)$

is an error term modelled as i.i.d. random variables with zero mean and known variance.

The IM predictor attempts to estimate a value \hat{y} based on a local neighborhood of the regressor space. The regressor vector is of the form

$$\varphi(t) = [y(t-1) \dots y(t-n_a) \dots u(t-d) \dots u(t-d-n_b)]^T \quad (2)$$

where n_a , n_b , and d denote the number of previous outputs, inputs, and delay in the model.

A local estimate \hat{y} can be obtained from the solution of the weighted regression problem

$$\hat{\theta} = \arg \min_{\theta} \sum_{k \in \Omega_k} l(y(k) - f(\varphi(k), \theta)) \cdot K\left(\frac{d(\varphi(k), \varphi(t))}{h}\right) \quad (3)$$

where $l(\cdot)$ is a scalar-valued and positive norm function, $K(\cdot)$ is a window function (referred to as a kernel), d is a scaled distance function and h is the bandwidth. Here, a tricube kernel function that smoothly descends to zero and has continuous derivative is used

$$K(z) = \begin{cases} (1-z^3)^3 & , |z| \leq 1 \\ 0 & , |z| > 1 \end{cases} \quad (4)$$

The distance function d is designed to determine how far away are the data from the current operating point. This function decides how the data will be weighted by the kernel function.

It can be described as

$$d(X_i, x) = (X_i - x)^T D (X_i - x) \quad (5)$$

D is equal to the inverse covariance of the regressors. The algorithm should select enough data in the local neighborhood, such that the least squares problem is not ill-conditioned. Then the neighborhood size can be expanded in order to achieve an accepted bias and variance errors tradeoff. The variance error of the estimate decreases, while the bias error of the estimate increases with increasing bandwidth.

The choice of the number of neighbors to be used in the local regression fit is related to the problem of finding the proper bias/variance trade-off. Adding points reduces the variances and increases the bias; on the other hand reducing the number of neighbors makes the bias smaller at the cost of a greater variance of the estimator. Thus, different models are considered, each fitted on a different number of examples, and the leave-one-out cross-validation procedure is used to compare them and to select the one for which the predicted error is smaller.

Localized automatic bandwidth selectors are used to estimate the quality of fit and a proper balance between bias and variance error.

Assuming a local linear model structure,

$$f(\varphi(k), \theta) = \theta_0 + \theta_1^T (\varphi(k) - \varphi(t)), \quad (6)$$

a quadratic norm, $l(u) = u^2$ is used and the model is then linear in the unknown parameters. The estimate can be easily computed using the least squares methods. A one step ahead prediction is given by

$$\hat{y}(t) = f(\varphi(t), \hat{\theta}) \quad (7)$$

Each local regression problem produces a single prediction $\hat{y}(t)$ corresponding to the current regression vector $\varphi(t)$. To obtain predictions at other locations in the regressor space, the weights change and new optimization problems have to be solved. This is in contrast to the global modeling approach where the model is fitted to data only once and then discarded. However, in a neighborhood around $\varphi(t)$, the local linear model provides an input-output linearization

$$A(q^{-1})y(t) = B(q^{-1})u(t-d) + \alpha \quad (8)$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the backward time-shift operator q^{-1} obtained from the components of $\hat{\theta}$, and

$$\alpha = \hat{\theta}_0 - \hat{\theta}_1^T \varphi(t) \quad (9)$$

is an offset term.

The bandwidth h controls the neighborhood size and has a critical impact on the resulting estimate since it governs a trade-off between bias and variance errors of the estimate. Traditional bandwidth selectors produce a global bandwidth. In IT estimation, bandwidth is computed adaptively at each prediction. The method used here is a localized Akaike Information Criterion (AIC). Use of entire data set for computing a model and evaluating the goodness-of-fit for the successive addition of one datapoint to the database would require a great amount of time.

In (Braun, et.al, 2001) an exponential search scheme with the following steps is proposed:

1. fit at a very small bandwidth h_0 , for which the estimation problem is well-posed.

2. increase bandwidth exponentially according to

$$h_i = (1 + 0.3/dx)h_{i-1}$$

dx represent the dimension of the regressor space. Compute the corresponding fits and proceed until the goodness-of-fit falls below a predefined level.

3. use the model of the bandwidth with lowest goodness-of-fit cost from the previous step.

It can be set a range for the number of data points used by the algorithm $[k_{\min}, k_{\max}]$ in which the exponential update will search. The value of k_{optim} used for each data point in the prediction is evaluated from the instabilities perspective due to poor data support. Based on this analysis the value k_{\min} can be increased. There is no real limit on k_{\max} other than computation time and database length. k_{\max} can be set beyond the highest value for k_{optim} observed for the validation dataset. Choosing k_{\min} , and k_{\max}

properly and determine a reasonable regressor structure are the two critical decisions.

One critical decision associated to model estimation is the selection of lagged inputs and outputs to be used as regressors for the model. The regressor space that provides the best fit for the global linear ARX modeling approach provides the best fit for this approach as well. The global linear ARX model is in fact a special case of ITMPC predictor. By using a boxcar window instead of a tricube window and forcing the predictor to use the entire estimation database, a global linear ARX model would be created at each step. Therefore, regressors which work well for the global linear ARX case are likely to be a good starting point for the more general case of IT estimation.

3. IT-MODEL PREDICTIVE CONTROL

MPC represents a class of discrete-time control systems which make explicit use of a model to predict the process output at future instants. A receding horizon strategy is used so that at each instant the horizon is displaced towards the future. At each instant the controller computes a control sequence that minimizes a given objective function which penalizes the control error and the control energy. Only the first control move calculated is fed to the plant, and then the horizon is advanced and a new control problem is solved.

To incorporate IT-estimation into MPC controller, the MPC objective function is chosen as in (Meadows and Rawlings, 1997). Given the model description and the knowledge of the current system state, a control is computed that minimizes the objective function

$$J = \sum_{k=0}^{N_p-1} Q_e(k) [r(t+k+1) - \hat{y}(t+k+1)]^2 + \sum_{k=0}^{N_p-1} Q_u(k) u^2(t+k) + \sum_{k=0}^{N_p-1} Q_{\Delta u}(k) \Delta u^2(t+k) \quad (10)$$

where $Q_e(k)$, $Q_u(k)$, $Q_{\Delta u}(k)$ represent penalty weights on the control error, control signal, and control move size respectively. r represents the setpoint, u the control signal, and \hat{y} the estimated output.

As new measurements become available, a new optimization problem is formulated whose solution provides the next control action. This referred to as the receding horizon principle. A special feature of the formulation is the presence of the move size in the objective. In some process control applications the move size is restricted due to actuator constraints. The main advantage of MPC is that hard constraints can be specified by the user on the input magnitude, move size or outputs.

Since optimization can be computationally intensive for large prediction horizons often only few control increments are solve, with the implicit assumption that the control action will be held constant for the remaining moves. The practice of using smaller horizons has the effect of producing less aggressive

controllers and providing stable control for non-minimum phase systems (Meadows, 1997).

Since the offset term changes stepwise, a CARIMA model is chosen such that the controller has integral action.

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t) \quad (11)$$

To express the output prediction at time $t + k$ as a function of future controls, introduce the following identity (Astrom and Wittenmark, 1995)

$$1 = A(q^{-1})F_k(q^{-1})\Delta = q^{-k}G_k(q^{-1}) \quad (12)$$

i.e.

$$\hat{y}(t+k) = B(q^{-1})F_k(q^{-1})\Delta u(t+k-1) + G_k(q^{-1})y(t) \quad (13)$$

By partitioning $B(q^{-1})F_k(q^{-1})$ as

$$B(q^{-1})F_k(q^{-1}) = S_k(q^{-1}) + q^{-k}\tilde{S}_k(q^{-1}) \quad (14)$$

where $\deg S_k(q^{-1}) = k-1$ and $\deg \tilde{S}_k(q^{-1}) = n_b-2$, the output prediction above can be rewritten as

$$\hat{y}(t+k) = S_k(q^{-1})\Delta u(t+k-1) + \bar{y}(t+k) \quad (15)$$

The first term depends on future control actions whereas the remaining terms depend on measured quantities only. By introducing

$$\begin{aligned} \hat{y} &= (\hat{y}(t+1) \dots \hat{y}(t+N))^T \\ \Delta \tilde{u} &= (\Delta u(t) \dots \Delta u(t+N-1))^T \\ \bar{y} &= (\bar{y}(t+1) \dots \bar{y}(t+N))^T \end{aligned} \quad (16)$$

$$S = \begin{pmatrix} s_0 & 0 & \dots & 0 \\ s_1 & s_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-1} & s_{N-2} & \dots & s_0 \end{pmatrix}$$

where s_i are the coefficients of $S_k(q^{-1})$ and

$$\hat{y} = \bar{y} + \tilde{S}\Delta \tilde{u} \quad (17)$$

Therefore $\hat{y} = \bar{y} + S\Delta u$ where

$$S = \tilde{S}\Lambda, \Lambda = \begin{pmatrix} I \\ 0 \end{pmatrix} \quad (18)$$

The control moves can also be expressed in vector form

$$u = T\Delta u + \bar{u} \quad (19)$$

where T is a matrix with 1's in the lower triangle and diagonal, and u is a vector containing only $u(t-1)$'s.

The objective can then be expressed as

$$J = \|r - y - S\Delta u\|_{Q_e}^2 + \|T\Delta u + \bar{u}\|_{Q_u}^2 + \|\Delta u\|_{Q_m}^2 \quad (20)$$

where r denotes the desired reference trajectory. For the *unconstrained* case, the minimizing control sequence is obtained explicitly by ordinary least squares theory. For the *constrained* case, the constraints can be re-formulated in matrix/vector form and the problem is solved efficiently using standard numerical optimization algorithms.

The full MPC formulation proposed in this paper includes the terms for the constrained case in order to allow for handling of infeasible output constraint situations. Vectors of slack variables e_{min} and e_{max} , penalize predicted output constraint violation at each instant along the prediction horizon. The following terms are added in the objective cost $\|\varepsilon\|_{Q_y}^2 + Q_y \varepsilon$. This formulation incorporates a weighted endpoint condition for ensuring stability, as in (Meadows and Rawlings, 1997). The number of endpoint conditions can be adjusted to cover the order of the system and force the states of the system to zero, while meeting setpoint at the end of the prediction horizon.

4. SIMULATION RESULTS

Continuous stirred tank reactors (CSTRs) are widely applied in chemical and biochemical engineering for the conversion of certain reactants into products. Usually, for irreversible exothermic reactions, the adiabatic mode leads to large production rates, so that in order to keep the reactor temperature within a given limit, the heat generated has to be removed by cooling. The vessel is assumed to be perfectly mixed, and a single first-order exothermic, irreversible reaction, takes place.

The mass and energy balance of an irreversible, exothermic, first order reaction can generally be described as:

$$\begin{aligned} \frac{dC_A}{dt} &= \frac{F}{V}(C_{Af} - C_A) - rC_A \\ \frac{dT}{dt} &= \frac{F}{V}(T_f - T) - \frac{\Delta H}{c_p \rho} rC_A - \frac{UA}{c_p \rho V}(T - T_j) \end{aligned} \quad (21)$$

$$r = k_0 \exp\left(\frac{-E}{RT}\right)_A$$

with the reaction rate function r which is dependent on the concentration of the reactant and the temperature in the reactor.

The inlet stream is fed at a constant rate F with constant concentration C_{Af} into the vessel. The final concentration of the reactant C_A is the controlled variable and the jacket temperature T_j is manipulated to keep the exit stream concentration C_A at the setpoint.

The exiting stream leaves at a rate F and since it is assumed the vessel is perfectly mixed, the exiting

concentration and vessel concentration are assumed to be the same. The in-out-flow rate F/V can be considered as a constant process parameter, as well as the area A , the coefficient U of the heat transfer, the fluid density, the fluid specific heat capacity c_p , and the reaction heat ($-\Delta H$), this all for a certain geometry of the reactor and within a certain selected reaction range. Non-adiabatic CSTR reactor parameters are presented in table I.

The equations have been solved in Simulink environment, using a fixed step-size $T=0.1$ hrs and a 4th Runge-Kutta method. The ARX structure that had the best fit was determined by examining values less than 10 for the regression orders and delay values. The k_{min} value was chosen 55 and the maximum value was equal to the length of the estimation database. The control evaluation focuses on setpoint tracking ability of the IT-MPC controller with and without input constraints. The control parameters has the following common values : $Q_e = 1$, $Q_u = 0$, $Q_{\Delta u} = 0.001$, $N_p = 5$, $N_u = 3$.

In the figure 1 it is shown the comparison between two responses with constrained input in the interval [265 350] and [280 340]. The root mean square values for these cases were $rms = 0.3687$ and $rms = 0.40662$ respectively. In the figure 2 are presented two responses which differ on the $Q_{\Delta u}$ penalty value which was chosen as 0.1 and 0,001. The rms values was obtained 0.3687 and 0.29651. In the figure 3 was presented the situation with and without endpoints constraints.

Table 1

Parameter	Units	Value
F/V	hr^{-1}	1
k_0	hr^{-1}	9703*3600
$(-\Delta H)$	Kcal/kgmol	5960
E	Kcal/kgmol	11843
$C_p \rho$	Kcal/(m ³ °C)	500
T_f	°C	25
C_{Af}	Kgmol/m ³	10
UA/V	Kcal/m ³ °Chr)	150
T_j	°C	25

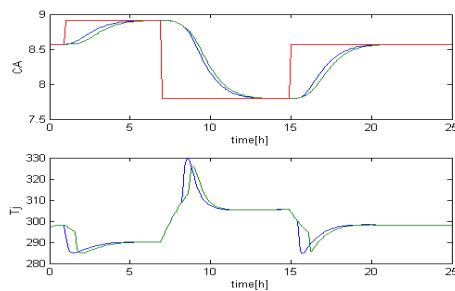


Fig. 1 Tracking with different input constraints

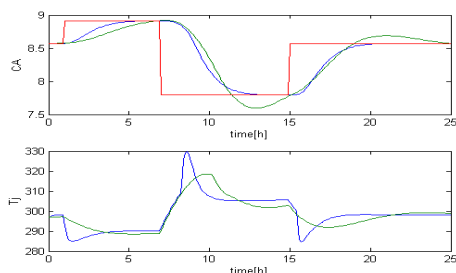


Fig. 2 Tracking with penalties on increment input

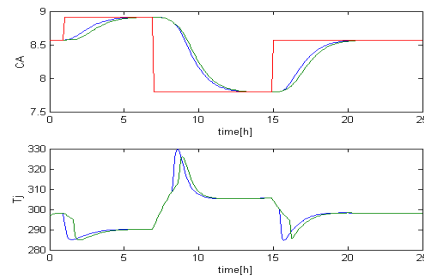


Fig. 3 Tracking with and without endpoint condition

5. CONCLUSIONS

The methodology provided in this paper allows the user to proceed through a framework of decision making which can reduce the length and cost of the nonlinear identification and control design process. By incorporating an endpoint condition for ensuring stability, predictive control can handle nonlinear processes in an efficient way. These performances benefits have been demonstrated on a continuous stirred tank reactor simulations.

REFERENCES

- Braun, M.W., D. E. Rivera and A. Stenman (2001). A 'Model-on-Demand' identification methodology for non-linear process systems. *Int. Journal of Control*, Vol.74, No. 18, 1708--1717.
- Constantin, N. and I. Dumitrache (2002). Multiple models for robust nonlinear control. *Proc. 15thIFAC World Congress*, CDRom, Barcelona, Spain.
- Demircioglu, H. and D. W. Clarke (1993). Generalised predictive control with end-point state weighting. *IEE Proceedings-D*, Vol. 140, No. 4, 275--282.
- Hurvich, C.M. and J. S. Simonoff (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society B*, Vol 60, 271-293.
- Meadows, E.S. and J.B. Rawlings (1997). Model Predictive Control. In: *Nonlinear Process Control*, Eds. M. A. Henson and D. E. Seborg, Prentice Hall, NJ.
- Ljung, L. (1997), *MATLAB System Identification Toolbox: User's Guide*. The Mathworks, Inc., Nantucket, MA.
- Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall Ptr, NJ.
- Scokaert, P.O. and J. B. Rawlings (1999). Feasibility Issues in Linear Model Predictive Control. *AIChE Journal*, Vol. 45, No. 8, 1649--1659.
- Stenman, A., F. Gustafsson and L. Ljung (1996). Just-in-time models for dynamical systems. *Proc. of the IEEE Conference on Decision and Control*, Kobe, Japan, 1115--1120