Estimation of Nonlinear Systems Using Linear Multiple Models

A. Banerjee and Y. Arkun
School of Chemical Engineering, Georgia Institute of Technology, Atlanta, GA 30332

B. Ogunnaike and R. Pearson
E. I. DuPont de Nemours & Co., Wilmington, DE 19880

This article addresses the problem of estimating the states of a nonlinear plant that operates in multiple regimes and makes transitions between them. It is often difficult to obtain a single nonlinear model that accurately describes the plant in all regimes. Even if a global model is available, it sometimes cannot be used conveniently in an estimator. An alternative approach is presented where local linear models are identified at each different operating point, and estimation is performed by tracking the transitions from one regime to another. This is done by first estimating the validity of the local models on-line and then constructing a time-varying global model by interpolating between the local linear models. State and parameter estimation is then performed using this global model in a moving horizon estimator. To demonstrate the effectiveness of this method, it is applied to three test systems: a simple numerical example, a CSTR, and a copolymerization reactor.

Introduction

In this work we describe a method of state estimation for nonlinear systems that are subject to multiple operating regimes. Different operating conditions are usually initiated by external factors such as changes in product specifications (e.g., polymer product and grade changeovers) or persistent plant disturbances (e.g., variations in feed conditions).

Poor transition control leads to long periods of transient operation, usually accompanied by the production of off-specification material and significant economic loss. However, in order to design a controller, it is necessary to obtain a model that accurately describes the plant at each of the different operating points, as well as during transition. Often a first-principles model that includes the underlying physics and chemistry of the process is difficult to obtain. Even if such a model is available, it may be inappropriate for state estimation and controller design. The alternative is to identify a single empirical model using plant input/output data. However, since chemical processes are nonlinear, unmodeled dynamics that are negligible at one operating point may be dominant at another. Therefore in order to uncover all the necessary plant dynamics, the identification algorithm might require inputs with large amplitude and/or large frequency, which may not be practically implementable. This would make it difficult to identify an empirical model that is valid for all operating conditions of interest.

The approach followed in this article is to use a local linear model for each of the different plant operating regimes. These multiple models may either be empirical models identified from local input/output data or linearizations of some global first-principles model, if available. Each model should satisfactorily describe the plant in a region around the point that it was identified, but may perform poorly when the plant moves out of that domain. Therefore this article describes ways of combining the information contained in the local models into a global description of the plant, and then carrying out state estimation by tracking transition on-line. Furthermore, a method of estimating the uncertainty around the global model is discussed, in order to study the effect of adding and removing local models.

The multiple-model paradigm has been used often in recent years for modeling and control of nonlinear systems. Progress in this active research area can be found in a recent book by Murray-Smith and Johansen (1997).

Local Models

Without any loss of generality it will henceforth be assumed that an estimator has to be designed for a single-

Correspondence concerning this article should be addressed to Y. Arkun.

input/single-output (SISO) nonlinear plant of the form

$$\dot{x} = f(x,u)$$
$$y = g(x,u), \tag{1}$$

where x is the state, y is the output, and u is the input.

Next let there be N different operating points:

$$(x_{s,i}, u_{s,i})$$

and assume that a linear state space model has been identified at each operating point:

$$\{M_i, i = 1, \ldots, N\}.$$

Therefore M_i consists of the state-space matrices $[A_i, B_i, C_i, D_i]$, obtained at the steady state $(x_{s,i}, u_{s,i})$. These local models may either have been obtained through identification or by linearizing a first-principles global model. It is also assumed that the true plant and all the models are observable.

The local models may be either discrete or continuous. In the remainder of this article results are presented for the continuous case. However, with minor modifications, the results readily apply to the discrete case also.

There are two broad approaches to combining the local models into a global model. The first method is to estimate at every sampling time the probability that each of the local models is a valid representation of the plant. This can be done by comparing the plant measurements with the outputs of each of the models. Then at any moment the plant states can be estimated by weighting the state estimates from each of the local models against the probability that the model is correct. The second approach is to construct a global model by interpolating between the local models using time-varying weights. These weights then need to be identified on-line so that the outputs of the global model match those of the plant. Therefore while the first approach considers the performance of each of the local models separately, the second approach looks only at the performance of the global model.

Therefore state estimates will be obtained from the global model, either by mixing the states of the local models or by mixing the models themselves. There will be a mismatch between the global model and the actual plant, which arises from two sources. First, there is modeling error in the local models around the different operating points. Second, any transition will take the plant through unmodeled regions that are only being approximated by the global model. Therefore errors in this approximation will also contribute to the mismatch. The mismatch will be treated as an uncertainty and its size will be estimated. Finally, the theory will be applied to three test systems: a simple numerical example, a continuously stirred-tank reactor (CSTR), and a copolymerization reactor.

Bayesian Estimator

This section describes how to use the adaptive estimation technique of Lainiotis and coworkers (Hilborn and Lainiotis, 1969; Lainiotis, 1971; Deshpande et al., 1973; Lainiotis, 1976) to choose the best model at any time, and to perform state estimation. This is done by calculating model probabilities, that is, $p(j)$ is the probability of the jth model being a valid representation of the plant.

Let the plant measurements be denoted by $y(k)$, and the measurement history by $Y(k) = [y(k), y(k-1)\cdots]^T$. Let $p(j \mid Y(k))$ denote the probability that model j best describes the plant given the measurement history until time k. Then applying Bayes' theorem,

$$\begin{aligned} p(j \mid Y(k)) &= p[j \mid y(k), Y(k-1)] \\ &= \frac{f[y(k) \mid j, Y(k-1)]\, p[j \mid Y(k-1)]}{p(y(k))} \\ &= \frac{f[y(k) \mid j, Y(k-1)]\, p[j \mid Y(k-1)]}{\Sigma_i f[y(k) \mid i, Y(k-1)]\, p[i \mid Y(k-1)]}, \end{aligned} \tag{2}$$

where $f[y(k) \mid j, Y(k-1)]$ is the probability distribution function (PDF) of the outputs of the jth model at time k given the measurement history $Y(k-1)$.

Equation 2 describes how an incoming plant measurement changes the belief about model validity, that is, how the measurement relates the *a posteriori* probability to the *a priori* probability. The approach followed in Hilborn and Lainiotis (1969), Lainiotis (1971), Deshpande et al. (1973), and Lainiotis (1976) to compute the distribution function, is to design a Kalman filter for each model. Then the assumptions underlying the jth Kalman filter are that the jth model exactly matches the plant and any mismatch is due to Gaussian noise entering the system. If these assumptions are satisfied, the model residuals ϵ_j will be zero-mean, and their covariance will be given by $\Omega_j = C_j P_j C_j^T + R_j$, where P_j is the state error covariance of the Kalman filter and R_j is the covariance of the measurement noise in the jth regime. Therefore, assuming stationarity,

$$\begin{aligned} f[y(k) \mid j, Y(k-1)] &= f[y(k) \mid j] \\ &= f(\epsilon_j(k)) \\ &= \frac{\exp\left\{-\frac{1}{2}\epsilon_j(k)\Omega_j^{-1}(k)\epsilon_j(k)^T\right\}}{\left[(2\pi)^N \det(\Omega_j(k))\right]^{1/2}}. \end{aligned} \tag{3}$$

Equation 3 can be substituted into Eq. 2 to obtain an algorithm for the recursive estimation of model validity. The algorithm must be initialized by the designer with an estimate for the starting probabilities $p[j \mid Y(0)]$.

The estimation of the model probabilities is strongly dependent on the covariance matrices of the model residuals, which in turn depend on the assumptions on the process and measurement noise covariance matrices. Figure 1 shows two scenarios, both for the case where there are two SISO models. In both situations the probability distribution functions of the models are centered around their estimates, and the spread of the distributions are given by Ω_1 and Ω_2. In the first scenario $\Omega_1 < \Omega_2$, and in the second $\Omega_1 = \Omega_2$. For that reason $f(\epsilon_1) < f(\epsilon_2)$ in the first graph even though model 1 has a smaller residual. However, in the second graph $f(\epsilon_1) > f(\epsilon_2)$ for the same residuals. Since the recursive form of Eq.

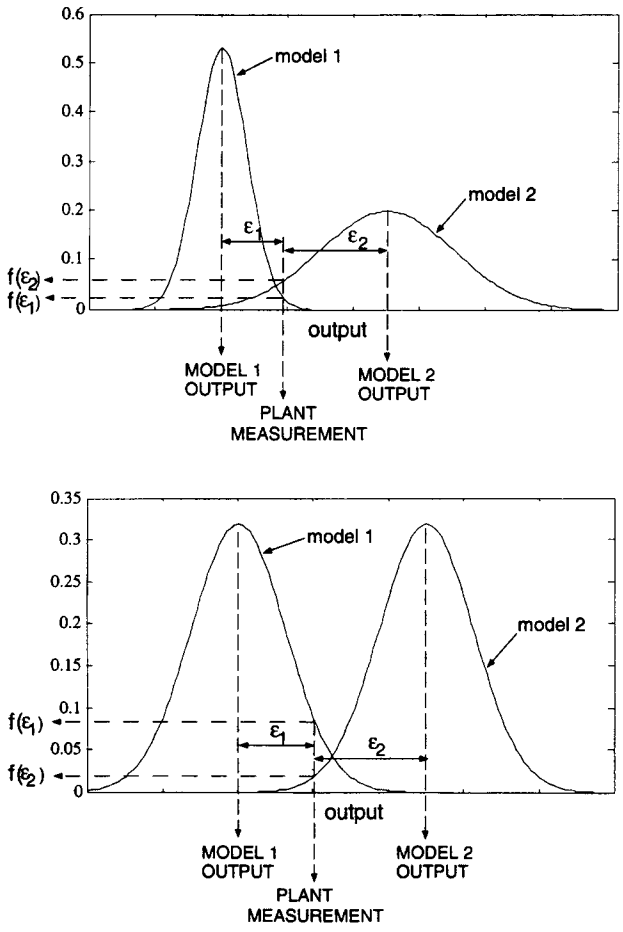


Figure 1. Probability distribution functions of model residuals.

2 causes the probability of the model with the largest $f(\epsilon)$ to eventually converge to 1, choosing the correct Ω_1 and Ω_2 is clearly very important, which means that the noise covariance matrices must be tuned carefully. This matter is illustrated further in the examples.

The partition theorem proposed by Lainiotis (1971), states that for the measurement history $Y(k)$, the optimal mean square estimate of the states is given by

$$\hat{x}(k \mid k) = \sum_{i=1}^{N} \hat{x}(k \mid k,i)p[i \mid Y(k)], \tag{4}$$

where $x(k \mid k,i)$ denotes the state estimate of the ith model at sampling time k, based on the measurement history $Y(k)$.

However, this approach was developed for the problem when the unknown plant matches one of a set of models exactly, and the algorithm has to determine the correct model. It is assumed that the actual plant does not switch from one region to another, and so the probabilities are being computed on the basis of a measurement history $Y(k)$ that includes all data until the present time. Therefore once the algorithm locks onto one model, the probability of that model approaches one and those of the other models approach zero. If the plant at any later time makes a transition to another regime, it may take a very long time for the probability of that model to change from zero because of the multiplicative recursion of Eq. 2. One ad hoc "fix" for this "slow response" problem (e.g., Willsky et al., 1980) is to specify a lower bound on the probabilities away from zero. However, this compromises the optimality guarantee of Eq. 4. As a more reasonable and intuitive approach, we propose a moving-horizon Bayesian estimator (MHBE) that calculates model probabilities based only on a past window of input/output data. This proposal is explained in the next section.

Moving-Horizon Bayesian Estimator

The Bayesian estimator described in the previous section was formulated recursively based on a measurement history that contained all measurements until the present time. The moving-horizon Bayesian estimator (MHBE) uses the same formulation, but makes use of only the plant measurements collected over the window $[k - n_e, \ldots, k]$.

Define a measurement history over the horizon given by

$$Y(k,k-n_e) = [y(k), y(k-1), \ldots, y(k-n_e)]^T$$

Then Eq. 2 can be rewritten as

$$p[j \mid Y(k,k-n_e)] = \tag{5}$$

$$\frac{f[y(k) \mid j,Y(k-1,k-n_e)]p[j \mid Y(k-1,k-n_e)]}{\Sigma_i f[y(k) \mid i,Y(k-1,k-n_e)]p[i \mid Y(k-1,k-n_e)]}, \tag{6}$$

where $f(y(k) \mid j,Y[k-1,k-n_e]$ is still given by Eq. 3.

As in the previous section it is assumed that all noise is Gaussian, and a bank of Kalman filters is used to estimate states using each of the local models. In addition the estimator stores the residuals from each of the filters collected over the moving window. Since the estimator is not making use of any data collected before the beginning of the window, it should not be biased initially. Therefore the probabilities are initialized at the beginning of the horizon by choosing

$$p[j \mid Y(k-n_e,k-n_e)] = 1/N,$$

where N is the number of models. Then $p[j \mid Y(k,k-n_e)]$ is calculated by using Eq. 5 recursively over the window as shown in Figure 2 for the case of two models. At the next sampling time, the window is shifted forward by one step and the process is repeated. This eliminates the problem of slow response of the probabilities because the estimator never uses data from before time step $k - n_e$, and reinitializes the probabilities at the beginning of the window.

The length of the window must be chosen long enough so that enough data are available to the estimator for updating the probabilities. However, if the length of the window is too long, then the estimator will be slow in responding to transitions.

This section has described how to use a moving horizon to improve the traditional Bayesian estimator. A comparison of the two Bayesian techniques will be shown later using physical examples.

Moving-Horizon State and Parameter Estimator

The focus of both Bayesian estimators just discussed is to

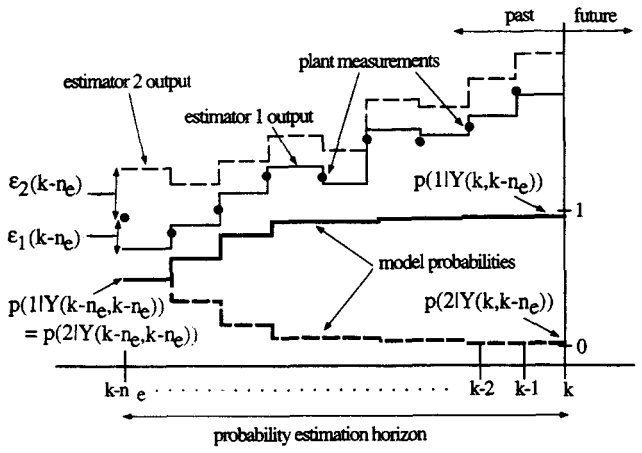


Figure 2. Moving-horizon Bayesian estimator.

select the best model out of a group of models. Since this article is motivated by the problem of state estimation during plant transitions, it should be remembered that the plant smoothly changes from matching one model to matching another during a transition. Therefore instead of simply choosing the best single model, a better approach would be to construct a global model that interpolates between the local models as the plant transforms itself. This is the method adopted in this section.

Constructing a global model

A first-order Taylor expansion of the plant about any steady-state point (x_s, u_s) is given by

$$\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{x_s,u_s}(x - x_s) + \left.\frac{\partial f}{\partial u}\right|_{x_s,u_s}(u - u_s)$$

$$y = \left.\frac{\partial g}{\partial x}\right|_{x_s,u_s}(x - x_s) + \left.\frac{\partial g}{\partial u}\right|_{x_s,u_s}(u - u_s). \tag{7}$$

It is known that at the point where the ith linear model was obtained,

$$\left.\frac{\partial f}{\partial x}\right|_{x_s,u_s} = A_i, \qquad \left.\frac{\partial f}{\partial u}\right|_{x_s,u_s} = B_i,$$

$$\left.\frac{\partial g}{\partial x}\right|_{x_s,u_s} = C_i, \qquad \left.\frac{\partial g}{\partial u}\right|_{x_s,u_s} = D_i,$$

$$(x_s, u_s) = (x_{s,i}, u_{s,i}),$$

and so during transition from the ith regime to the jth one, the Jacobians change from $[A_i, B_i, C_i, D_i]$ to $[A_j, B_j, C_j, D_j]$, and the local steady state changes from $(x_{s,i}, u_{s,i})$ to $(x_{s,j}, u_{s,j})$. However, in general the nonlinear state equations will not be known, so in order to use Eq. 7 to approximate the plant during transition, two things need to be done:

1. The Jacobians need to be estimated on-line as they change. One way of doing this would be to keep interpolating between the state-space matrices of the local models at each time step. This is similar in principle to linearizing the plant at discrete time steps during transition, except that linearization is replaced by interpolation between a finite number of models.

2. At any point in time a suitable (x_s, u_s) has to be chosen at which to apply Eq. 7. This can be done by estimating the steady-state curve of the nonlinear plant from the steady-state curves of the local models, and then choosing an appropriate point on this approximate curve.

A method of choosing an approximate steady-state point is outlined in the next section. Then a method of interpolating the state-space matrices is described. Next these two methods are combined to give a state estimator for the nonlinear plant.

Estimating the steady-state curve

If Eq. 7 is to be used to approximate the plant during transition, an appropriate steady-state curve from $(x_{s,i}, u_{s,i})$ to $(x_{s,j}, u_{s,j})$ has to be estimated on-line. Then at each sampling time, the closest steady-state point on this curve has to be found, and the deviations used in Eq. 7 have to be calculated from that.

For single-input systems, $f(x, u) = 0$ is the true steady-state curve for the nonlinear plant (Eq. 1). For multiple-input systems, this represents a steady-state manifold. If the nonlinear state equations given by Eq. 1 were known, then at any time the estimated states and inputs could simply be projected onto the steady-state manifold to get the nearest steady-state point. However, in the rest of this section it will be assumed that the state equations are unknown, and the steady-state manifold has to be approximated from the local models.

The steady-state equation for the ith local model is

$$A_i(x - x_{s,i}) + B_i(u - u_{s,i}) = 0.$$

Assuming that A_i is invertible, solving this equation gives the hypersurface

$$x = x_{s,i} - A_i^{-1}B_i(u - u_{s,i}), \tag{8}$$

and converting states to outputs gives

$$y = y_{s,i} - C_iA_i^{-1}B_i(u - u_{s,i}), \tag{9}$$

where $y_{s,i}$ is the vector of steady-state outputs. It follows from Eq. 7 that this is a tangent space to the actual unknown steady-state manifold at $(y_{s,i}, u_{s,i})$. Therefore each of the N models gives a tangent space that is a local approximation to the steady-state manifold at its own operating point.

This is shown in Figure 3 with two local models for an SISO system. Tangents obtained by applying Eq. 9 to the two models are used to approximate the steady-state curve. It is clear from Figure 3 that only a part of the tangent surface of each model is used to approximate the steady-state curve. The input space has been subdivided into a domain for each model: $u \le u_{\text{int}}$ and $u > u_{\text{int}}$, where u_{int} is the point of intersection of the two tangents. For each model only that part of the tangent is used that corresponds to inputs in its own domain. In general there are therefore N local approximations to the steady-state curve, but there is only one global approximation, and it is constructed by choosing that subspace from

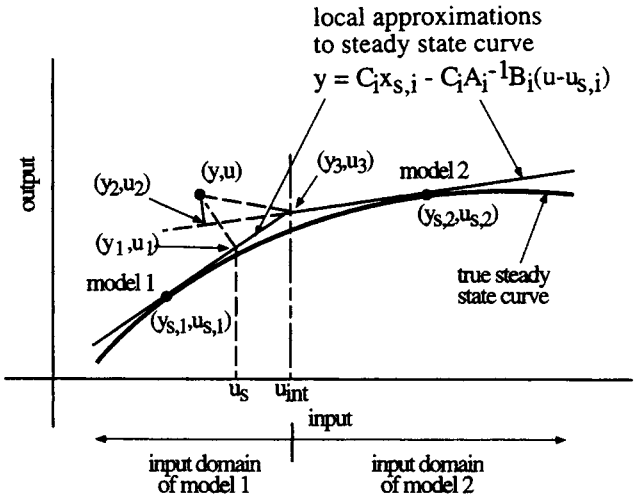


Figure 3. Estimating the nearest steady-state point for a SISO system.

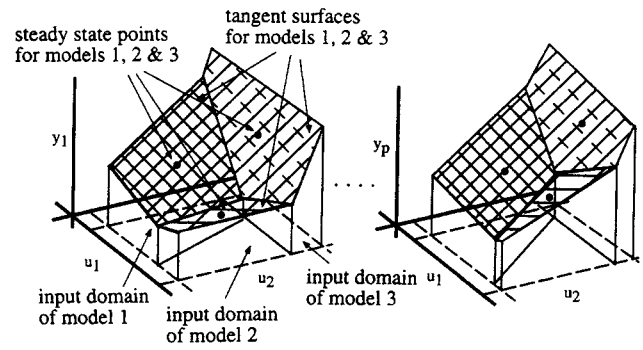


Figure 4. Estimating the nearest steady-state point for a MIMO system.

each of the tangents that corresponds to the input domain of the corresponding local model.

Then for any input/output pair (u, y), an estimate of the nearest steady-state point (y_s, u_s) is obtained by projecting the point onto the approximate global steady-state curve. However, it is not clear which tangent space to project (y, u) onto, because each model provides one. The easiest way to resolve this problem is to project the point onto each of the different tangent spaces to get N choices for (y_s, u_s), and then simply choose the geometrically closest of them. However, Figure 3 shows that this would not be correct. The point (y,u) is projected onto both tangents to give (y_1, u_1) and (y_2, u_2), respectively. While (y_2, u_2) is closer to (y,u) than (y_1, u_1), it is clear that (y_2, u_2) is outside the input domain of model 2, and therefore model 2 is no longer valid there. Therefore (y_1, u_1) should be compared with (y_3, u_3) because that is the closest point on the tangent of model 2 that is also within its input domain. Since (y_1, u_1) is closer to (y, u) than (y_3, u_3), it is chosen as the nearest steady-state point,

$$y_s = y_1 \qquad u_s = u_1;$$

x_s can then be found by the substitution of u_s into Eq. 8 for model 1.

This procedure as described earlier applies only to SISO systems. Now consider a multiple-input/multiple-output (MIMO) system. With multiple inputs and outputs, the true unknown steady-state manifold is now a hypersurface. Each linear model provides a tangent to this surface corresponding to Eq. 9, which is a local approximation to the steady-state manifold. Figure 4 shows these tangents by projecting them onto p reduced-order subspaces, each corresponding to a single output. For graphical simplicity the procedure is illustrated on a system with two inputs and three local models, but it can be easily extended to larger systems.

As in the SISO case only part of each tangent is included in the global approximation. The subset of the tangent surface chosen for inclusion consists of those points that lie in its input domain. For example, the figure shows how the input domain for model 1 is bounded by the intersections of its tangent with the tangents of every other model. Points within the input domain satisfy the following conditions:

1. They satisfy Eq. 9 for the ith model, that is, they lie on its tangent surface and satisfy

$$y = y_{s,i} - C_i A_i^{-1} B_i (u - u_{s,i}). \tag{10}$$

2. For each of the p outputs, they lie on the same side of all the other tangent surfaces as does $(y_{s,i}, u_{s,i})$. Therefore they must satisfy

$$\begin{aligned} &y - \left[y_{s,j} - C_j A_j^{-1} B_j (u - u_{s,j}) \right] \\ &\quad > 0 \text{ if } y_{s,i} > y_{s,j} - C_j A_j^{-1} B_j (u_{s,i} - u_{s,j}) \\ &\quad < 0 \text{ if } y_{s,i} < y_{s,j} - C_j A_j^{-1} B_j (u_{s,i} - u_{s,j}) \\ &\qquad\qquad (j = 1, \ldots, N, j \neq i). \end{aligned} \tag{11}$$

To better understand this, consider the SISO case of Figure 3. Applying this condition to the tangent of model 2, it chooses only that part of the tangent that falls to the right of the tangent of model 1. This corresponds to the input range $u > u_{\text{int}}$, which is the input domain of the second model.

Therefore these constraints define the subsets of the local tangent surfaces that are to be included in the global approximation of the steady-state manifold. This approximate steady-state manifold is used on-line to determine at any given time an appropriate steady-state point, (x_s, u_s), to be used in a model of the form of Eq. 7. Given a plant data point (y, u), the following procedure is implemented:

1. For each model calculate the closest point on its tangent to (y, u), which is also within the model's input domain. For the ith model this point is denoted by (y_i, u_i), and calculated by solving

$$\min_{y_i, u_i} \| y - y_i \|_2^2 + \| u - u_i \|_2^2, \tag{12}$$

subject to the conditions of Eqs. 10 and 11. This is a quadratic programming (QP) problem that is easily solved by any standard optimization package without much computational burden, even for large systems. However, it should be carried out only after scaling the inputs and outputs by the maximum values of $x_{s,i}$ and $u_{s,i}$ to eliminate ill-conditioning. Otherwise

any output or input with a large magnitude will bias the optimization in its favor.

2. Then $(y_s,u_i)=(y_i,u_i)$, where the ith model is that with the smallest value of $\| y-y_i \|_2^2 + \| u-u_i \|_2^2$.

3. u_s can then be substituted back into Eq. 8 for model i to get x_s.

If the local models are discrete, then the preceding procedure will remain unchanged, except that the ith local approximation to the steady-state surface will be given by

$$y = y_{s,i} + C_i(I - A_i)^{-1} B_i(u - u_{s,i}) \tag{13}$$

instead of Eq. 10, and this will get substituted into Eq. 11.

In the following sections, this operation of finding the nearest (x_s, y_s, u_s) on the appropriate steady-state hypersurface will denoted by

$$(x_s, y_s, u_s) = \text{near}(y, u).$$

Interpolating the state-space matrices

A method of estimating an appropriate steady-state point on-line was presented in the previous section. A method of approximating the Jacobians will now be described, where they are parametrized by estimates of model validity functions that are updated on-line.

Model validity functions are estimates of the validity, or trustworthiness, of each of the local models. The functions make up a vector

$$p(t) = [p_1(t), \ldots, p_N(t)]^T \in R^N$$

and may be defined so that $p_i(t)$ maps the plant's outputs and inputs onto a measure of the validity of the ith model:

$$p_i(t):(y(t),u(t)) \rightarrow [0,1], \tag{14}$$

where

$p_i(t) \rightarrow 1$ when the ith model is valid,
$p_i(t) \rightarrow 0$ otherwise,

and

$$\sum_i^N p_i(t) = 1. \tag{15}$$

Equation 15 implies that as the plant moves into a region where one of the models becomes more trustworthy than the others, the other models lose their validity.

One way of tracking transitions between the local models, is to parametrize the state-space matrices of the *nominal* global model M by the model validity functions

$$\{A[p(t)], B[p(t)], C[p(t)], D[p(t)]\},$$

where the specific form of the matrices is yet to be determined. These matrices are therefore time-varying estimates of the Jacobian of the actual plant. It is assumed that nonlinearities of the true plant are captured by the functional dependence of the system state-space matrices on $p(t)$.

The nominal global model is then defined by a linear parameter-varying (LPV) system, given by

$$\dot{x} = A[p(t)]x + B[p(t)]u \tag{16}$$

$$y = C[p(t)]x + D[p(t)]u. \tag{17}$$

Traditionally LPV systems have been interpreted as linearizations of nonlinear systems along parameter trajectories (Apkarian et al., 1994; Ravi et al., 1991). In this work the model validity functions constitute the parameters.

Now a specific form of the functional dependence of the state-space matrices on $p(t)$ must be decided. An implicit restriction on the global model is that if $p_i = 1$, it should reduce to M_i, the ith linear state-space model:

$$\left.\begin{array}{ll} A(p)=A_i & B(p)=B_i \\ C(p)=C_i & D(p)=D_i \end{array}\right|_{p_i=1},$$

which has already been locally validated. One way of imposing this is to choose the following matrices for the global model:

$$A(p(t)) = \Sigma_i^N p_i(t) A_i \qquad B[p(t)] = \Sigma_i^N p_i(t) B_i \tag{18}$$

$$C[p(t)] = \Sigma_i^N p_i(t) C_i \qquad D[p(t)] = \Sigma_i^N p_i(t) D_i. \tag{19}$$

This structure attempts to interpolate between the state-space matrices of the local models. If the p_i are chosen in an optimal manner, then it should be expected that this approach has to be at least as good as the Bayesian approach where each model is considered separately, and no mixing of models takes place. The preceding global modeling is similar to that presented in Johansen and Foss (1992) and (1995), with the very important difference that this work addresses the on-line estimation of the validity functions and presents some new results in this direction.

Therefore the structure that we have chosen to implement as $M(p)$, the nominal global model, is

$$\dot{x} = \left[\sum_i^N p_i(t) A_i\right][x - x_s(t)] + \left[\sum_i^N p_i(t) B_i\right][u - u_s(t)]$$

$$y = y_s(t) + \left[\sum_i^N p_i(t) C_i\right][x - x_s(t)] + \left[\sum_i^N p_i(t) D_i\right][u - u_s(t)] \tag{20}$$

$$(x_s, y_s, u_s) = \text{near}(y, u).$$

Note that the model validity functions are very similar to the model probabilities of the Bayesian techniques—only the interpretation is different. The model probabilities were computed based on the residuals of each of the local models. However, the model validity functions parametrize the global model and need to be calculated so that the global model matches the plant measurements.

The advantages of using this particular LPV structure as the global model are:

1. The global model satisfies the linearization property, that is, each local model is recovered exactly from the global model at the operating point at which it was obtained, provided the $p_i(t)$ are estimated properly.
2. Recent work in Apkarian et al. (1994) presents methods to design robust linear parameter-varying controllers for plants of this structure. Based on this, the design of transition controllers for nonlinear plants is discussed in Banerjee et al. (1995).

However some questions still need to be answered: How does one assign the model validity functions? How well does the global model in Eq. 20 match the actual plant, and how many local linear models are needed?

Moving-horizon estimation of model validity functions

This section describes a method of estimating the values of the model validity functions on-line. The approach used here is to consider the model validity functions p_i to be parameters of the global model, and treat their time variation as random walk processes. State and parameter estimation is performed simultaneously, but subject to the constraint that $\Sigma_i p_i = 1$. A discrete moving-horizon-based estimation approach similar to that proposed in Robertson and Lee (1995) can be used for this purpose, the difference being that in Robertson and Lee it was assumed that the nonlinear state-space function of Eq. 1 is known. Since this is usually not the case here, that approach is modified to fit the present problem.

For notational convenience in this section, time will be denoted in the subscript. Therefore x_k, $p_{i,k}$, and p_k denote the plant state, the ith model validity function, and the vector of validity functions, respectively, at time step k. Also for any vector z, $z_{j|k}$ denotes the best estimate of z at time step j based on measurements up to, and including, time step k.

Let the process and measurement noise at time step k be denoted by w_k^x and v_k, respectively. If the local models are continuous, then the time-varying state-space matrices of the local models must be discretized to get

$$\{A_d[p(k)], B_d[p(k)], C[p(k)], D[p(k)]\}.$$

If T is the sampling time, these matrices are given by

$$\begin{aligned} A_d[p(k)] &= e^{A[p(kT)]T} \\ B_d[p(k)] &= -[I - e^{A[p(kT)]T}]A^{-1}[p(kT)]B[p(kT)] \\ C[p(k)] &= C[p(kT)] \\ D[p(k)] &= D[p(kT)]. \end{aligned} \tag{21}$$

Next define

$$\begin{aligned} F(x_k, u_k, p_k) &= x_{s,k} + A_d[p(k)](x_k - x_{s,k}) \\ &\quad + B_d[p(k)](u_k - u_{s,k}) \\ G(x_k, p_k) &= y_{s,k} + C[p(k)](x_k - x_{s,k}) + D[p(k)](u_k - u_{s,k}), \end{aligned}$$

where

$$(x_{s,k}\, y_{s,k}\, u_{s,k}) = \text{near}(y_k, u_k).$$

Let the vector of validity functions be treated as a random walk process:

$$p_{k+1} = p_k + w_k^p,$$

where w_k^p is modeled as white noise. Then the state-space equations to be used by the estimator are

$$\begin{bmatrix} x_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} F(x_k, u_k, p_k) \\ p_k \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^p \end{bmatrix} \tag{22}$$

$$y_k = G(x_k, p_k) + v_k.$$

At any instant, state and parameter estimation is performed by examining plant input/output data over the preceding m time steps. As shown in Figure 5 this window has been termed the estimation horizon, and the global model is simulated using the inputs given to the plant over that interval. The estimator then attempts to select values of the parameters that minimize the mismatch between actual plant measurements and the outputs of the global model. More specifically at time step k, $w_{k-m+1}^x \cdots w_{k-1}^x$, $w_{k-m+1}^p \cdots w_{k-1}^p$, and the initial states and validity functions are chosen so that the outputs of Eq. 22 from time $k-m+1$ to k match the true plant as closely as possible.

The initial estimate of the states and validity functions at time $k-m+1$ is chosen to be

$$\begin{bmatrix} x_{k-m+1|k-m} \\ p_{k-m+1|k-m} \end{bmatrix} + X_{k-m}^e,$$

where X_{k-m}^e is an error vector. At time step l let y_l denote the actual plant measurements, and v_l denote the mismatch between the plant measurements and model outputs. Also let the noise in the states and parameters be represented by

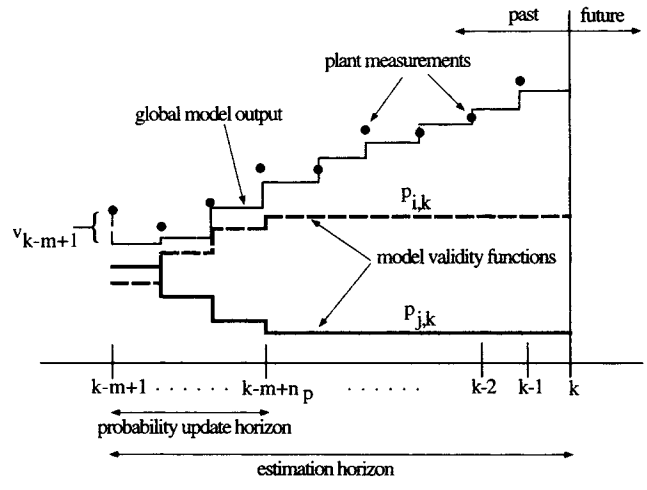


Figure 5. Moving horizon estimation.

$$w_l = \begin{bmatrix} w_l^x \\ w_l^p \end{bmatrix}.$$

Then in order for the model outputs to match the plant, the following optimization must be carried out:

$$\min_{\substack{X_{k-m}^e \\ w_{k-m+1} \cdots w_{k-1}}} \quad (X_{k-m}^e)^T P^{-1} X_{k-m}^e + \Sigma_{l=k-m+1}^{k} v_l^T R^{-1} v_l + \Sigma_{l=k-m+1}^{k-1} w_l^T Q^{-1} w_l + \lambda \Sigma_{l=k-m+1}^{k} d_l^T p_l \quad (23)$$

s.t.

$$v_l = y_l - g(x_l, p_l)$$

$$x_l = F(x_{l-1}, u_{l-1}, p_{l-1}) + w_{l-1}^x$$

$$p_l = p_{l-1} + w_{l-1}^p$$

$$d_l = [\, \| y - y_{s,1} \|^2 \cdots \| y - y_{s,N} \|^2 \,]$$

$$0 \le p_{i,k} \le 1$$

$$\Sigma_{i=1}^{N} p_{i,k} = 1,$$

where P^{-1}, Q^{-1}, and R^{-1} are positive definite weighting matrices that are chosen to reflect the user's confidence in the initial state estimate, the weighted state equations, and the weighted output equations. In addition, extra constraints of the form

$$| p_{i,k} - p_{i,k-1} | \le \alpha_p \quad (24)$$

can also be added, where α_p represents a bound on the rate of change of the validity functions. This can be used to prevent the estimator from overreacting to noise.

The results of a typical optimization are shown in Figure 5. The profiles of the global model output and the validity functions are based on data collected from time steps $k-m+1$ to k. It should be noted that at the next time step, the data window will be moved forward and the profiles will be recomputed over the new horizon.

The first three terms of the objective function are similar to those proposed in Roberston and Lee (1995), and are responsible for making the model output match that of the plant without being overly sensitive to noise. The matrices P^{-1} and Q^{-1} are usually chosen diagonal, and can be partitioned into the following structure:

$$P^{-1} = \begin{bmatrix} P_x^{-1} & 0 \\ 0 & P_p^{-1} \end{bmatrix} \qquad Q^{-1} = \begin{bmatrix} Q_x^{-1} & 0 \\ 0 & Q_p^{-1} \end{bmatrix}. \quad (25)$$

Here P_x^{-1} and Q_x^{-1} are diagonal matrices of size equal to the number of states, which weight the initial mismatch and process noise added to x_k. Similarly P_p^{-1} and Q_p^{-1} are diagonal matrices of size equal to the number of models, and weight the initial mismatch and process noise added to p_k. If all the states are of the same order of magnitude, then P_x^{-1} and Q_x^{-1} may simply be constant-times-identity matrices with suitably chosen constants. Otherwise these matrices should include scaling factors to ensure that the effect of the larger states do not dominate those of the smaller ones in the objective function. P_p^{-1} and Q_p^{-1} can in general be chosen to be constant-times-identity matrices. As for R^{-1}, it too should include scaling factors if the outputs of the models are of different orders of magnitude.

The fourth term in the objective function has been included to "anchor" the nominal global model to the local models at the points where they were obtained. It is essentially a weighted sum of the model validity functions, and the weights at any time step are the distances between the model output and the points where the models were identified. Therefore if the model output is far from the operating point where the ith model was obtained, the weight on its validity function is high. Therefore this term penalizes the estimator for assigning high validity functions to models at points far away from where they were identified. It also smoothes out the effect of noise on the estimation of probabilities. For example, if the plant remains close to the steady-state operating point of one of the models, the fourth term will keep the probability of that model close to 1, and prevent the estimator from changing the validity functions as a result of noise. This smoothing cannot be accomplished fully with the P_p^{-1} and Q_p^{-1}, because high values of these terms are needed to prevent the $p(t)$ from reacting to noise. However, this would then slow down the changing of $p(t)$ during transitions. The variable λ in the objective function is a tuning parameter that can be used to increase or decrease the weight given to the penalty.

One way of reducing the computational burden of this optimization is to compute the validity functions for only the first n_p time steps, and hold $w_{k-m+n_p+1}^p = \cdots = w_{k-1}^p = 0$. This is shown in Figure 5, where n_p has been termed the probability update horizon. This assumes that the validity functions remain constant in the remainder of the horizon. However, it should be noted that when the optimization is repeated at the next time step, the validity functions are updated again.

This moving-horizon parameter estimator does have one other advantage. Usually the states are physical quantities that remains bounded—for example, the concentration of a reactant can never become negative. Therefore constraints can be added to the optimization in Eq. 23 in order to ensure that the estimated states do not violate these bounds. However, neither the Bayesian estimators nor the traditional extended Kalman filter (EKF) can include constraints in this manner. This is discussed further in the examples.

Estimating Uncertainty

The true unknown plant is represented by the nominal global model $M(p)$, plus any mismatch that may be attributed to uncertainty, disturbances, and noise. The uncertainty incorporates the effects of:

1. The errors and uncertainties in the local models—these represent the plant-model mismatches around the points where each of the models was identified.

2. The errors and uncertainties in the global nominal model—these represent the plant-model mismatches during transition between the regimes and arise mainly due to approximating Eq. 1 by Eq. 20.

Furthermore the uncertainty in the global model, which may be large during transitions, reduces to the uncertainty in the ith local model when the plant is within the domain of the ith model. Therefore the uncertainty will be time-varying and will be represented henceforth by $\Delta(t)$.

A method for estimating the size of the uncertainty based on plant input/output data is given in this section. This is useful for two reasons:

1. It provides a measure of how good the nominal global model is. This is useful in deciding if additional models need to be added, or if some existing models can be removed.
2. It provides a measure of how robust the controller design needs to be.

Let u and y represent the true plant inputs and outputs, and u_s and y_s be the time-varying estimate of the closest plant steady-state point obtained at each sampling time using the procedure outlined earlier. Next define the deviation variables $\hat{u} = u - u_s$ and $\hat{y} = y - y_s$. Then Figure 6 shows the relationship between the true plant and the nominal global model, along with disturbances, d, and noise, n. Therefore the output of the true plant is given by

$$\begin{aligned} y &= y_s + \hat{y} \\ &= y_s + M[p(t)]\hat{u} + \Delta(t)\hat{u} + n + d, \end{aligned} \tag{26}$$

where $M[p(t)]$ is the LPV nominal global model (Eq. 20). The objective of this section is to use plant input/output data to estimate the smallest size of Δ that is consistent with the data.

It is assumed that the sequences $u = \{u_0, u_1, \ldots, u_{l-1} \in R^m\}$ and $y = \{y_0, y_1, \ldots, y_{l-1} \in R^p\}$ are available that contain the sampled measurements of the plant inputs and outputs, respectively. Next the moving-horizon estimator described in the preceding section can be run off-line on these data to produce the sequence of model outputs $\tilde{y} = \{\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{l-1}\}$, $y_s = \{y_{s,0}, y_{s,1}, \ldots, y_{s,l-1}\}$, and $u_s = \{u_{s,0}, u_{s,1}, \ldots, u_{s,l-1}\}$. This can be represented by

$$\begin{aligned} \tilde{y} &= y_s + M[p(t)](u - u_s) \\ &= y_s + M[p(t)]\hat{u}. \end{aligned} \tag{27}$$

However, the moving horizon parameter estimator described earlier estimates the states by trying to determine both the probabilities and noise required to make the model outputs match the plant. Equation 27 does not include the effects of output disturbances, d, and noise, n. However, the state estimator adds noise to the states in order to reduce mismatch due to model uncertainty. Therefore, when using the estimator to produce model outputs for estimating uncertainty, it becomes necessary to turn off the process noise estimation.

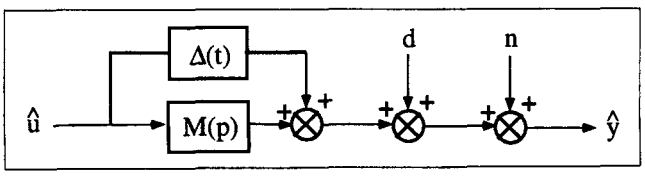


Figure 6. Nominal model with uncertainty.

This means setting $w_k^x = 0$ in Eq. 22, so that it is dropped from the list of decision variables in Eq. 23.

Then combining Eqs. 26 and 27, the plant-model mismatch can be expressed as

$$e = y - \tilde{y} = \Delta(t)\hat{u} + n + d, \tag{28}$$

where

$$\begin{aligned} e &= \{y_0 - \tilde{y}_0, y_1 - \tilde{y}_1, \ldots, y_{l-1} - \tilde{y}_{l-1}\} \\ \hat{u} &= \{u_0 - u_{s,0}, u_1 - u_{s,1}, \ldots, u_{l-1} - u_{s,l-1}\}. \end{aligned}$$

The problem then reduces to finding the smallest amount of uncertainty that can explain the plant-model mismatch in the presence of noise and disturbances. Before tackling this problem, it is easier to first consider the case when there is no noise nor disturbances. To solve this we make use of the following theorem (reproduced from Poolla et al., 1994, with a change in notation).

Theorem 1. Let the induced 2-norm of an operator Δ be defined by

$$\| \Delta \|_{i2} = \sup_{u \neq 0} \frac{\| \Delta u \|_2}{\| u \|_2}.$$

Then, given sequences $\hat{u} = \{\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{l-1} \in R^m\}$ and $e = \{e_0, e_1, \ldots, e_{l-1} \in R^p\}$, there exists a stable, causal, linear, time-varying operator Δ with

$$\| \Delta \|_{i2} \leq \gamma$$

such that

$$\Delta(\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{l-1}, *, *, \ldots) = (e_0, e_1, \ldots, e_{l-1}, *, *, \ldots)$$

if and only if

$$\| \pi_k e \|_2 \leq \gamma \| \pi_k \hat{u} \|_2 \qquad \text{for } k = 1, \ldots, l, \tag{29}$$

where π_k is the k-step truncation operator

$$\pi_k : (\hat{u}_0, \hat{u}_1, \ldots) \rightarrow (\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{k-1}, 0, 0, \ldots).$$

It follows from Eq. 29 that the smallest value of γ that satisfies the conditions of the theorem is

$$\gamma = \max_k \frac{\| \pi_k e \|_2}{\| \pi_k \hat{u} \|_2} \qquad \text{for } k = 1, \ldots, l, \tag{30}$$

which provides an estimate of the smallest uncertainty that explains the data if there are no noise or disturbances present. If the inputs to the plant do not excite every mode of plant behavior, γ will be smaller than the actual value of $\| \Delta \|_{i2}$. Therefore the plant input/output data being used should cover as much of the different operating regions as possible. However, even if the calculated value of γ is smaller than the actual value of $\| \Delta \|_{i2}$, it is still useful, as it pro-

vides a lower bound on the uncertainty that the controller must be able to tolerate.

Next, consider the same problem in the presence of noise and disturbances. It is clear from Eq. 28 that just by looking at values of e and u it is impossible to say how much of the mismatch is due to Δ and how much is a result of n and d. Therefore estimating the size of Δ from data is possible only if n and d can be measured, or if they are bounded in some way. In such a case, the main result follows from Theorem 1.

Theorem 2. Suppose the disturbance and noise sequences are unknown, but are known to satisfy the system of constraints $f_C(d,n) \le 0$. Let the space of noise and disturbance sequences that satisfy this constraint be denoted by

$$\Omega = \{d, n \mid f_C(d,n) \le 0\}. \tag{31}$$

Then given the sequences $\hat{u} = \{\hat{u}_0, \ldots, \hat{u}_{l-1}\}$ and $e = \{e_0, \ldots, e_{l-1}\}$,

$$\gamma = \min_{d,n \in \Omega} \frac{\max_k \| \pi_k(e-d-n) \|_2}{\| \pi_k \hat{u} \|_2} \qquad \text{for } k = 1, \ldots, l \tag{32}$$

is the smallest value of γ for which there exist

$$\Delta(t), \qquad d = \{d_0, \ldots, d_{l-1}\}, \qquad n = \{n_0, \ldots, n_{l-1}\}$$

such that

$$\| \Delta \|_{i2} = \sup_{\hat{u} \neq 0} \frac{\| \Delta \hat{u} \|_2}{\| \hat{u} \|_2} \le \gamma,$$

$$(d,n) \in \Omega$$

and

$$(e_0, \ldots, e_{l-1}, *, *, \ldots) = \Delta(\hat{u}_0, \ldots, \hat{u}_{l-1}, *, *, \ldots) + (d_0, \ldots, d_{l-1}, *, *, \ldots) + (n_0, \ldots, n_{l-1}, *, *, \ldots).$$

Proof. If d and n were known, then

$$e - d - n = \Delta \hat{u}. \tag{33}$$

Substituting this into Eq. 30 gives

$$\gamma = \frac{\max_k \| \pi_k(e-d-n) \|_2}{\| \pi_k \hat{u} \|_2} \qquad \text{for } k = 1, \ldots, l. \tag{34}$$

However if the only information known about d and n is that they belong to the set 31, then for each of these sequences a value of γ can be calculated. From these, the smallest value of γ for which there exists d and n sequences that belong to the set Ω is given by Eq. 32.

As example, suppose that n and d are unknown but bounded, that is, n_b and d_b are known such that

$$-d_b \le d \le d_b \qquad -n_b \le n \le n_b, \tag{35}$$

and for notational convenience assume that the system is SISO. Then this simple noise model can be used to further simplify the optimizations in Eq. 32. The expression for γ reduces to

$$\gamma = \max_k \frac{\| \pi_k w \|_2}{\| \pi_k \hat{u} \|_2} \qquad \text{for } k = 1, \ldots, l, \tag{36}$$

where w is defined as

$$w = \begin{cases} |e| - d_b - n_b & \text{if } |e| \ge (d_b + n_b) \\ 0 & \text{if } |e| < (d_b + n_b). \end{cases} \tag{37}$$

Calculating the value of γ helps decide how many local models are needed for the global model to accurately represent the plant. Clearly, the larger the number of local models that are available, the better will Eq. 20 match Eq. 1. However, after a certain number of models, the incremental benefit of adding more might not be significant. This can be checked by using plant data to see how the value of γ reduces by adding more models. When the number of local models is fixed, then the estimate of the uncertainty provides a lower bound on the robustness needed from the controller. If no controller can be found that is robust enough, then the number of local models is insufficient.

The quantity γ can also be considered a measure of nonlinearity of the system. If it changes considerably with the number of models being used, then the system is very nonlinear. However, if γ is relatively independent of how many models are used, then the incremental effect of adding models is small, and it implies that the system is almost linear.

Examples

The theory developed earlier was applied to three test systems. The first of these is a simple numerical example to demonstrate the differences between the different approaches discussed before. The other examples are reactor systems that are chosen to illustrate the application of the theory to typical real-life processes.

Numerical example

The nonlinear system was chosen to be the following:

$$\dot{x}_1 = -1 - 5x_1 - 10x_1^3 + x_2 + u + 3u^2 \tag{38}$$

$$\dot{x}_2 = -x_2 + (u - 0.6)^2$$
$$y = x_1. \tag{39}$$

Two local models were then obtained by linearizing the state equations at the following two steady-state points:

$$x_{s,1} = \begin{bmatrix} 0.2159 \\ 0.01 \end{bmatrix}, u_{s,1} = 0.7 \qquad x_{s,2} = \begin{bmatrix} 0.4499 \\ 0.16 \end{bmatrix}, u_{s,2} = 1.$$

The models were then discretized with a sampling time of one unit to give the following discrete state-space matrices:

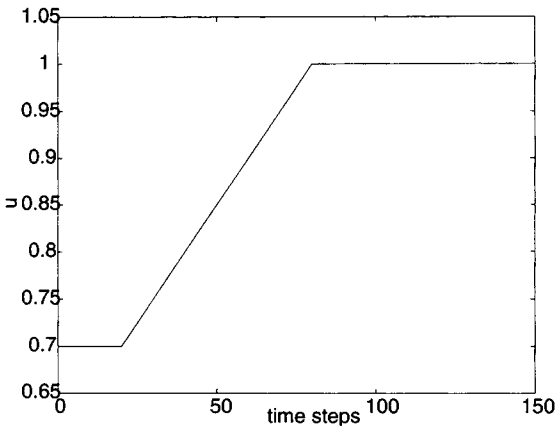


Figure 7. Input given to nonlinear system in Example 1.

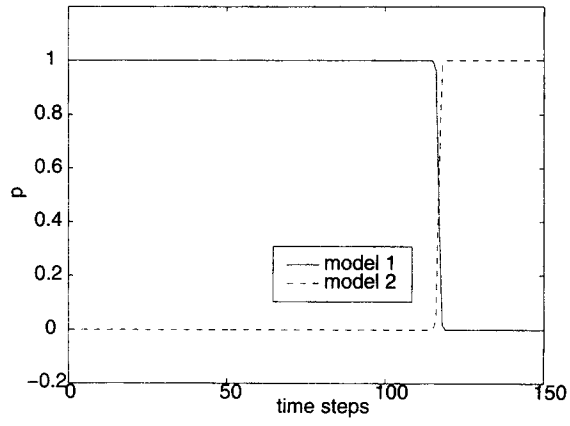


Figure 9. Model probabilities for Example 1 using BE.

$$A_1 = \begin{bmatrix} 1.66\times10^{-3} & 0.0678 \\ 0 & 0.3679 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1.55\times10^{-5} & 0.0365 \\ 0 & 0.3679 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.83 \\ 0.13 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.68 \\ 0.51 \end{bmatrix}$$

$$C_1 = C_2 = [1 \quad 0]$$

$$D_1 = D_2 = 0.$$

Although the second state is not measured, both local models are observable.

The nonlinear system was simulated using a smooth linear ramp in input from $u_{s,1}$ to $u_{s,2}$, as shown in Figure 7. Measurements of the first state were sent to a Bayesian estimator, a moving-horizon Bayesian estimator, and a moving-horizon parameter estimator (MHPE). The aim was to see how well these estimators tracked the transition and estimated the second state. This is not an easy task as the gains of the unmeasured state x_2 with respect to the input changes by a factor of 4 during the transition.

noise covariance matrices

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.005 \end{bmatrix} \qquad R = 2.5\times10^{-5}. \tag{40}$$

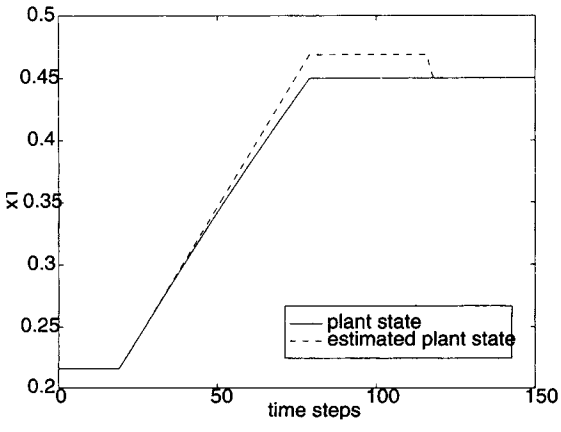
Even though no noise was added to the plant states or measurements in this example, this value of R was chosen so that Ω, the variance of the residuals, was roughly the same for each model:

$$\Omega_1 = 4.94\times10^{-5} \qquad \Omega_2 = 4.91\times10^{-5}.$$

This eliminates any bias in model selection with the Bayesian estimator when using Eqs. 2 and 3.

Figure 8 compares the estimated states to the true plant states during the transition using the Bayesian estimator, and it is clear that the estimator does not perform well, especially for the unmeasured state x_2. Figure 9 shows that the model probabilities change only about $t = 120$, which is well after the transition is complete. As a result, the estimated states do not match the plant states until this time. The reason for this delayed response is the slow response of the model probabilities, as discussed earlier. Even though the second Kalman filter starts outperforming the first one midway through the transition, this effect does not show up in the estimated states, because the probabilities take such a long time to switch.

It is to eliminate this slow response that the MHBE was proposed. The covariance matrices given in Eq. 40 were again used to design the two local Kalman filters, and these were



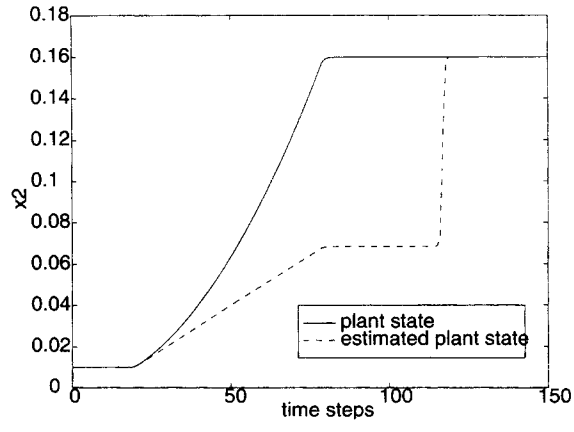


Figure 8. Estimated states for Example 1 using BE.

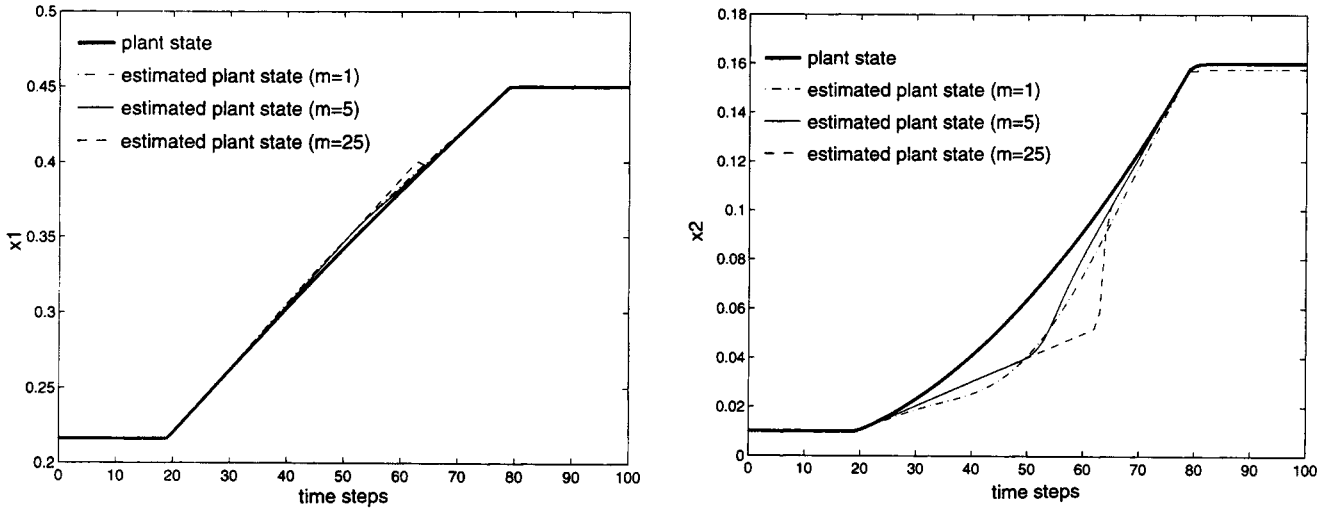


Figure 10. Estimated states for Example 1 using the MHBE with different estimation horizons.

used with a moving horizon of $m = 5$. The state estimates appear in Figure 10 and the probabilities are given in Figure 11. It is clear that the estimator does a much better job as the probabilities switch smoothly during the transition, but there is still some mismatch in the unmeasured state x_2.

Next the effect of changing the window size was studied, and Figures 10 and 11 show how the state estimates and model probabilities varied when the horizon was changed to $m = 1$, and then $m = 25$. For both of these cases, there is mismatch in the estimation of x_2, and it is more than the case when m was chosen to be 5. While the difference between $m = 1$ and $m = 5$ is not much, there is considerable deterioration when m is increased to 25. This trend is shown more clearly in Figure 12, where the 2-norm of the error in the estimate of x_2 is plotted as a function of window size. It is clear that when using the MHBE, there is an optimal window size around 5. The reason for this trend is that for smaller horizons, not enough data are being used to determine the probabilities, and for larger horizons the effect of the slow response is increasingly pronounced.

In order to reduce the mismatch in x_2, a MHPE was used with different horizon sizes. The parameters used in the optimization were

$$P^{-1} = \begin{bmatrix} P_x^{-1} & 0 \\ 0 & P_p^{-1} \end{bmatrix} = Q^{-1} = \begin{bmatrix} Q_x^{-1} & 0 \\ 0 & Q_p^{-1} \end{bmatrix} = \begin{bmatrix} 1{,}000 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \tag{41}$$

$$R^{-1} = 4 \times 10^4 \qquad \lambda = 0. \tag{42}$$

It should be noticed that P_x^{-1} and Q_x^{-1} have been chosen equal to the inverse of the process noise covariance matrix Q used in the Kalman filters. Similarly, R^{-1} has also been chosen as the inverse of the measurement noise covariance matrix R. This was done in order to make a fair comparison

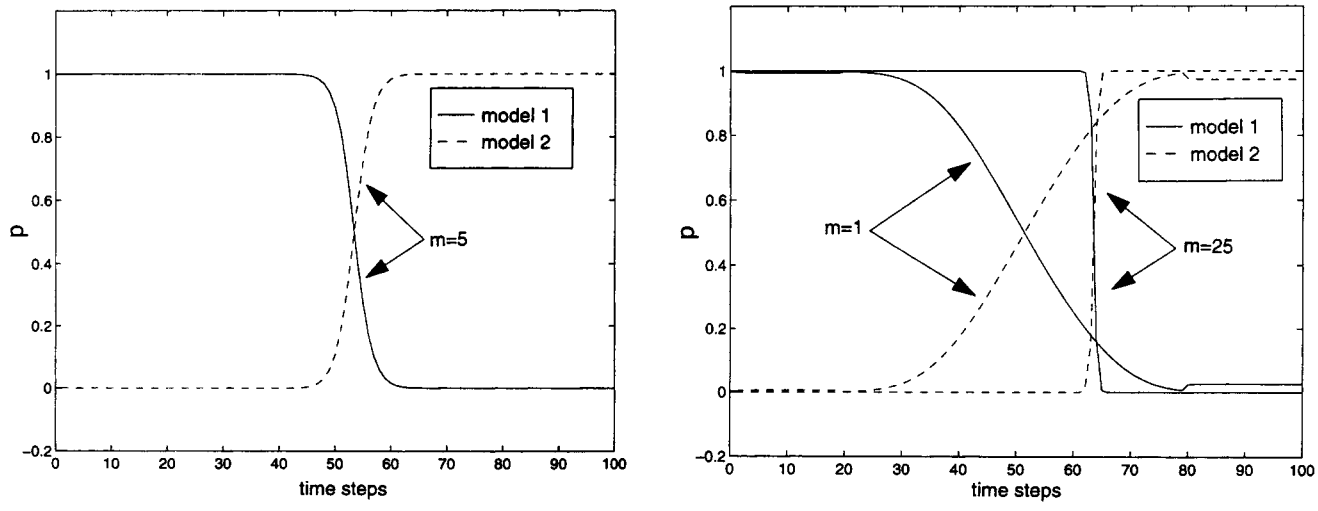


Figure 11. Model probabilities for Example 1 using the MHBE with different estimation horizons.

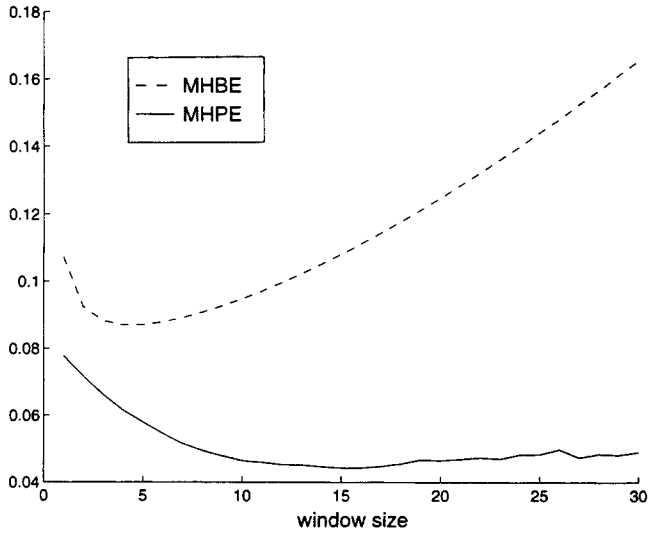


Figure 12. $\| e \|_2$, the 2-norm of the estimation error in the second state, shown as a function of estimation horizon.

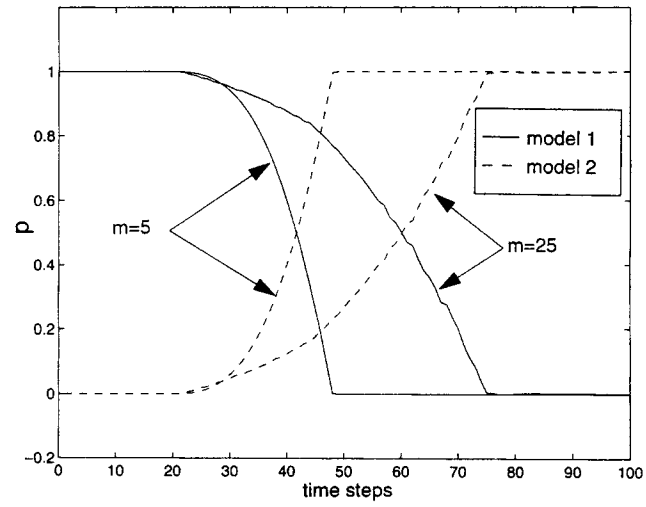


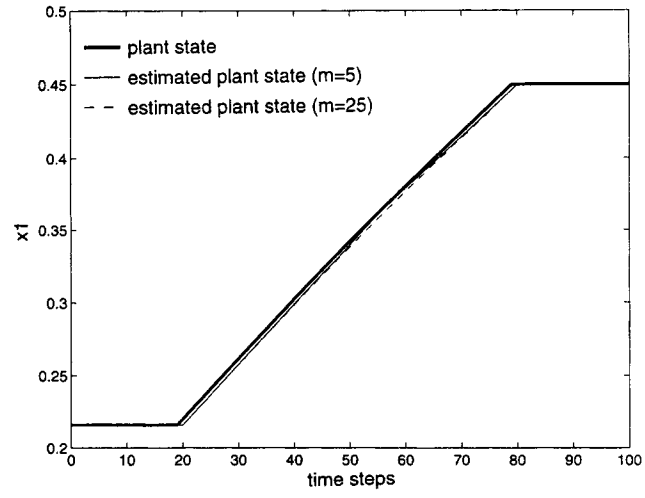
Figure 14. Model probabilities for Example 1 using the MHPE with different estimation horizons.

between the two methods. The tuning parameters P_p^{-1} and Q_p^{-1} were chosen to be $3I_2$ so as to get a good response during the transition. However, since there was no measurement or process noise, the estimation was not very sensitive to this parameter. For the same reason λ was chosen to be zero because it was not necessary to smooth out the effect of noise on the validity functions.

Figures 13 and 14 show the performance of the MHPE for two different window sizes, and it is clear that the MHPE outperforms the MHBE, as the estimation error is smaller. With a horizon of five time steps, the estimation of x_2 has improved only marginally over the MHPE, as there is still mismatch. However, when the horizon was increased from 5 to 25 time steps, the MHPE was able to track x_2 with a marked reduction in mismatch.

Figure 12 shows how the estimation error in x_2 varied as a function of window size when an MHPE was used. Like the MHBE the error decreases initially as window size reduces, but then unlike the MHBE the estimation error does not start significantly increasing again as window size is increased. This is not surprising, as the MHPE does not suffer from the slow-response problem as the horizon becomes larger—it simply adjusts the profile of the validity functions in order to match the larger amount of data available. However, the leveling off of the curve indicates that there is some estimation error that cannot be reduced by changing the window size. This reflects the amount of information about the process contained in the local models.

For all the simulations presented for this example, performance of the estimator is directly related to how the probabilities/validity functions change during the transition. For the BE, Figure 9 shows that the probabilities switch very fast, and that too after the transition. Since at any given time, only one of the models is being selected, during the period when neither model matches the plant, the estimator performs poorly. When a MHBE is used with large horizons, the probabilities converge and stay at either one or zero for longer times and then switch abruptly toward the end of the transi-



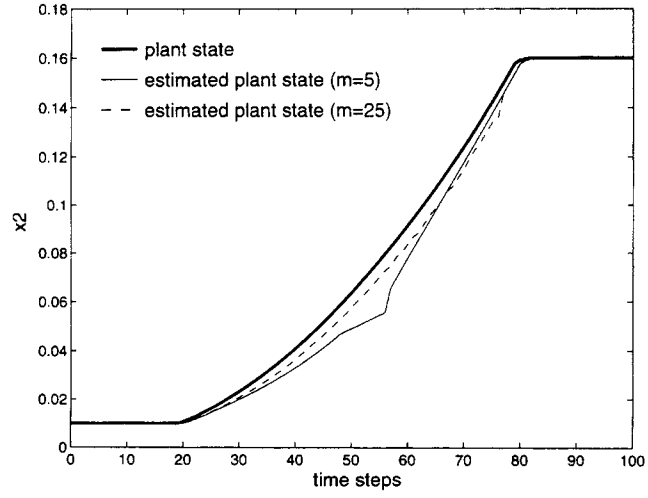


Figure 13. Estimated states for Example 1 using the MHPE with different estimation horizons.

tion period. Consequently, the estimated states of the local models are hardly mixed (see Eq. 4) during transition, and the estimation error grows. This is observed for $m = 25$ in Figures 10, 11 and 12. When the horizon is reduced to 5, the probabilities gradually change during the transition, the state estimates are mixed, and the estimator performs well. A further decrease in horizon length to one time step causes the response of the probabilities to become sluggish, and that increases the estimation error again. Another drawback of using a very small estimation horizon is that it would make the MHBE more sensitive to measurement noise.

When using the MHPE, increasing the horizon makes more data available to the estimator, which makes an attempt to mix models rather than select one of them. Therefore, using a horizon of 25, allows the MHPE to change the validity functions more gradually, and in a more optimal manner. This occurs as the plant moves out of the domain of model 1, and gradually enters the domain of model 2. As a result, the estimation improves.

This simple example was chosen to illustrate the fundamental differences between the three approaches presented in the theory. The next two examples simulate systems that may be commonly found in a chemical plant.

CSTR example

The second test system to be studied is a CSTR with a first-order exothermic reaction. The dynamic behavior is described by the following equations (Uppal et al., 1974):

$$\frac{dx_1}{dt} = -x_1 + D_a(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right)$$

$$\frac{dx_2}{dt} = -x_2 + BD_a(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right) + \beta(u-x_2)$$

$$y = x_2,$$

where the states x_1 and x_2 are the dimensionless concentration and reactor temperature, respectively, and t is dimensionless time. The input u is the dimensionless temperature of the cooling jacket surrounding the reactor. The constants are $D_a = 0.072$, $\gamma = 20$, $B = 8$, and $\beta = 0.3$. This system was chosen because it exhibits output multiplicity, as can be seen from the steady-state curve given in Figure 15. Furthermore the middle branch of the steady-state curve is unstable, though the upper and lower branches are stable. Therefore the system has different dynamics in the regions surrounding the three branches, and so is well suited for application of the theory.

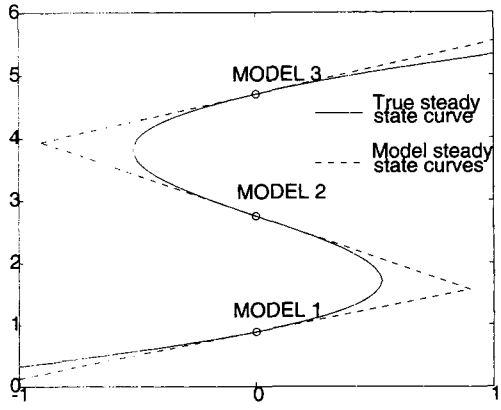


Figure 15. Steady-state curve for CSTR, showing the model locations and the approximate model steady-state curves.

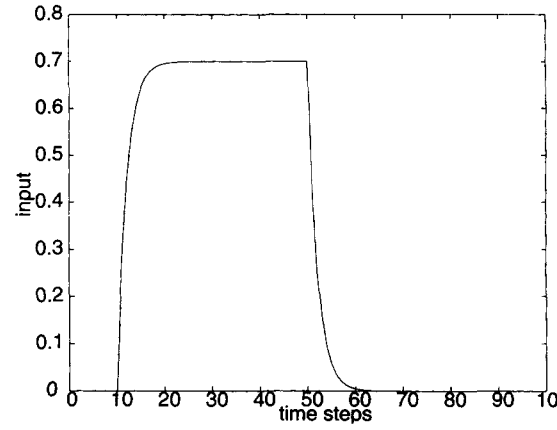


Figure 16. Input to CSTR.

An open-loop simulation was carried out using three local models, obtained by linearizing the nonlinear state equations on each of the three branches, as shown in Figure 15. The corresponding local steady-state curves obtained using Eq. 8 are also shown, and they make up the global approximate steady-state curve. The first and third models are stable, and the second model is unstable.

The plant is initially at the center of the lower branch, which is where the first model was obtained. The input given to the CSTR is shown in Figure 16. When the input increases, the plant moves off the lower branch and goes to the upper branch. Once the plant has reached the upper branch, the input is again reduced to zero, and the plant moves to the center of the upper branch. Figure 17 shows the plant output as a result of this input change. The measurements have been corrupted by Gaussian noise with a standard deviation of 0.05, and the sampling time used is one dimensionless time unit.

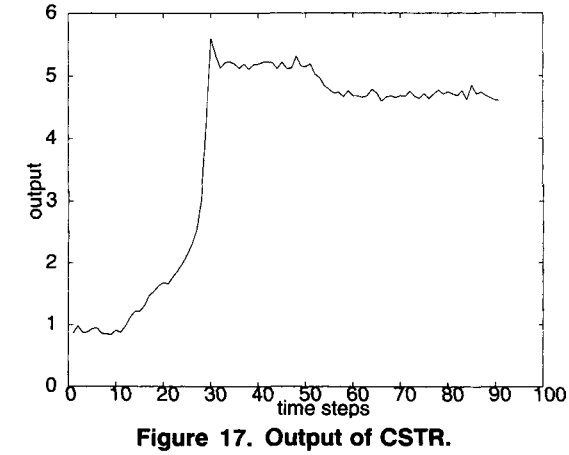


Figure 17. Output of CSTR.

Estimation was carried out using the BE, MHBE, and the MHPE. In each case the design parameters are as follows:

1. BE: A Kalman filter was designed for each of the models, and the noise characteristics of each of the filters was chosen to be

$$Q_i = (0.1X_{s,i})^2 \qquad R_i = (0.05Y_{s,i})^2,$$

where $X_{s,i}$ and $Y_{s,i}$ are diagonal matrices containing the states and outputs at the ith operating point.

2. MHBE: A Kalman filter was designed for each of the models with the same noise characteristics as for the Bayesian estimator. In addition a window size of five sampling times was used to estimate model probabilities.

3. MHPE: An estimation horizon of $m = 5$ was found to be sufficient. The rate of change of the validity functions in Eq. 24 was chosen to be $\alpha_p = 0.05$. The weights used in the objective function were

$$P^{-1} = \begin{bmatrix} 10(X_{s,1})^{-1} & 0 \\ 0 & 10I_{3\times 3} \end{bmatrix}$$

$$Q^{-1} = \begin{bmatrix} 10(X_{s,1})^{-1} & 0 \\ 0 & 10I_{3\times 3} \end{bmatrix}^2$$

$$R^{-1} = \left[100(Y_{s,1})^{-1}\right]^2 \qquad \lambda = 20.$$

The matrices $X_{s,1}$ and $Y_{s,1}$ are included in the definition of P^{-1}, Q^{-1}, and R^{-1} in order to scale the states and inputs. The scalar constants in these matrices were chosen to give the maximum weightage in the objective function to the mismatch between the measured and modeled output. That explains why the constant in R^{-1} is larger than the constants in the other matrices.

The results are shown in Figures 18 and 19. All three estimators initially choose the first model ($p_1 = 1$, $p_2 = 0$, $p_3 = 0$), as the CSTR is being operated around the lower branch of the steady-state curve. Then when the input is increased, the plant moves to an operating point on the upper branch. Finally, when the input is stepped down again the plant moves to the center of the upper branch, which is where the third model was obtained. That is why the probabilities eventually converge to ($p_1 = 0$, $p_2 = 0$, $p_3 = 1$), which means that the third model is being selected. However, while the plant is making the transition from the lower to the upper branch, it moves through a region close to the middle branch, which is why the probability and validity function of the second model approach one for a short while. It is clear that the BE takes longer to react to the input changes than do the other two moving-horizon estimators. This affects the state estimation, which for the BE is not as good as the moving-horizon estimators (see Figure 19). Comparing the two moving-horizon estimators, it is clear that the MHPE outperforms the MHBE, as the estimated states are closer to the true plant states. This is because the MHBE probabilities converge to the best local model at each sampling time. Since this is rather fast, the effect of mixing the states is lost. So the estimated states from the MHBE are usually those of the Kalman filter based on the best local model at that time. However, the MHPE

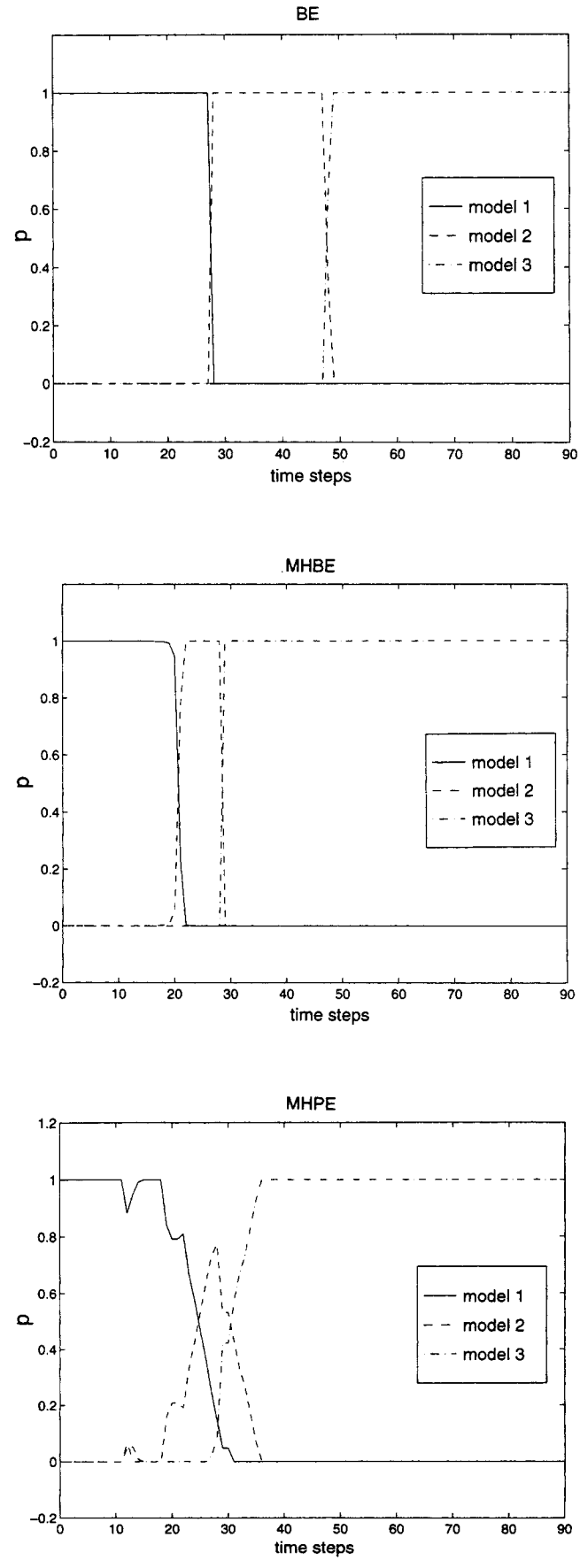
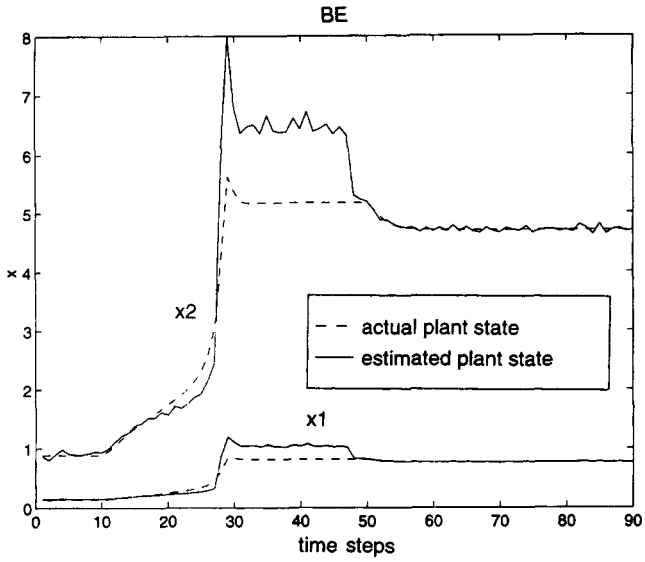
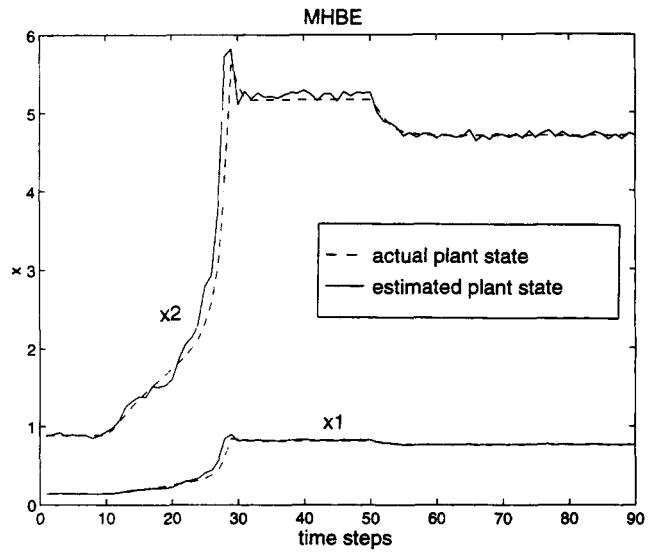


Figure 18. Model probabilities for the CSTR example.





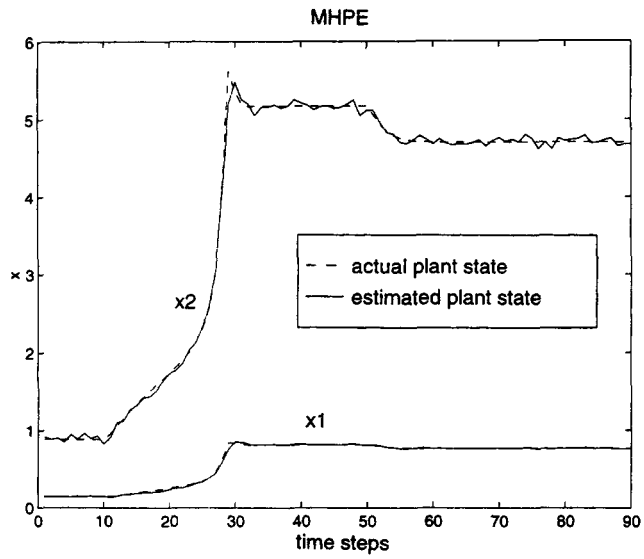


Figure 19. Estimated states for the CSTR example using three models.

Table 1. Average % Errors in the Estimated States Using the Three Different Types of Estimators

Estimator	Average % Error
BE	8.54
MHBE	2.53
MHPE	1.62

attempts to mix models in an optimal manner, and so should be able to find a mixed model that is at least as good as the best of the local models. A comparison of the average error in the states from using the three estimators is given in Table 1.

An intermediate result shown in Figure 20 illustrates how the nearest steady-state point, (x_s, u_s) was estimated during the simulation. Initially, the estimated point remains on the tangent of the first model. Then as the plant moves away from the lower branch, the steady-state point switches to the tangent of the second model. Finally when the plant gets close to the upper branch, the steady-state point moves over to the third tangent.

In order to study the effect of changing the number of models, two further simulations were carried out with the MHPE, using two models and five models. These models were obtained by linearizing the nonlinear state equations at the different locations circled on the steady-state curve in Figure 21. The parameters of the MHPE were not changed, except that P_p^{-1} and Q_p^{-1} were changed to $10I_{2\times 2}$ when two models were used and $10I_{5\times 5}$ when five models were used. The estimated states are compared to the true plant states in Figure 22. It appears that the simulation with five models is very similar to that with three models. However, when two models are used in the estimator, the states are estimated poorly during the transition from the lower branch to the upper branch. This is not surprising, as the plant is unstable around the central branch, whereas the two models being used have been obtained on the stable branches. Therefore the models do not match the plant dynamics in between the upper and lower branches.

The results from Theorem 2 was applied to each of the three simulations in order to assess the effect of changing the number of models on the magnitude of the uncertainty bound

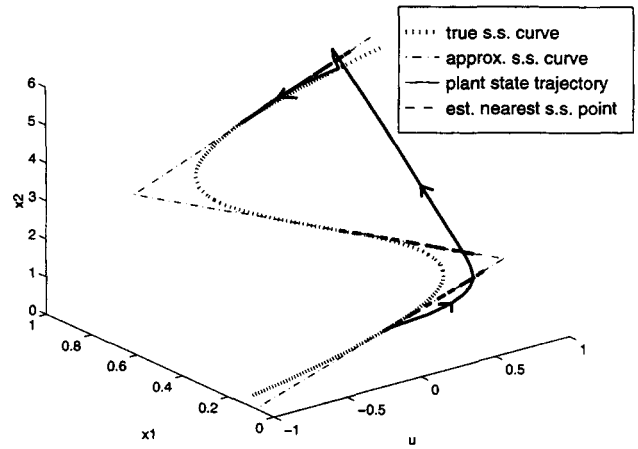


Figure 20. Trajectory of the estimated nearest steady state point for the CSTR example.

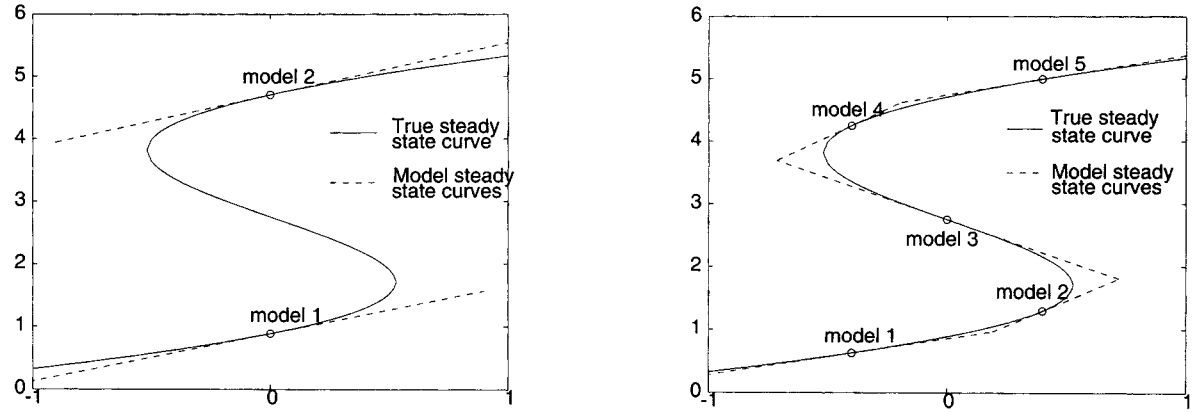


Figure 21. Location of models when two models and five models are used.

γ. The value of γ was calculated using the MHPE with two, three, and five models. In each case the uncertainty was estimated both in the presence and absence of measurement noise. The γ values for these simulations are given in Table 2.

The table demonstrates how measurement noise can affect the calculation of γ. For these simulations it was observed that measurement noise did not exceed a value of 0.13, and there were no disturbances. Therefore Eqs. 35–37 were used to calculate γ, using the values $n_b = 0.13$ and $d_b = 0$. When using noise-free measurements, both d_b and n_b were taken to be zero. The estimated value of γ decreases slightly in the presence of noise, because it cannot be exactly determined how much plant-model mismatch is due to noise and how much is a result of uncertainty.

More importantly Table 2 also shows that while there is hardly any benefit from having more than three models, the uncertainty increases threefold if the central model is dropped, since the local models on the lower and upper branches are not adequate to describe the nonlinear transition dynamics. This increased uncertainty may significantly degrade closed-loop performance if the estimator is used with a controller. This is borne out by the results of Banerjee et al. (1995) where a multiple-model-based controller is used with this CSTR. It is shown there that if the central model is dropped, the controller can no longer reach setpoints around the middle branch.

Copolymerization reactor example

The third example is a copolymerization reactor (Congalidis et al., 1989) whose flow diagram appears in Figure 23. A proportional integral derivative (PID) controller manipulates the rate of coolant flow in order to obtain any desired jacket temperature. It is assumed that the dynamics of this controller are much faster than those of the reactor and have consequently been neglected. Therefore, in the subsequent discussion, the reactor jacket temperature will be treated as a manipulated variable, whereas in reality it is the setpoint of the PID controller.

Monomers A and B are added continuously to the CSTR, along with initiator, solvent, and chain transfer agent. There is also the possibility of an inhibitor entering the system. Table 3 lists the chemical names of each of the components.

In this work, the two monomer feeds, the initiator, and the chain transfer agent along with the reactor jacket tempera-

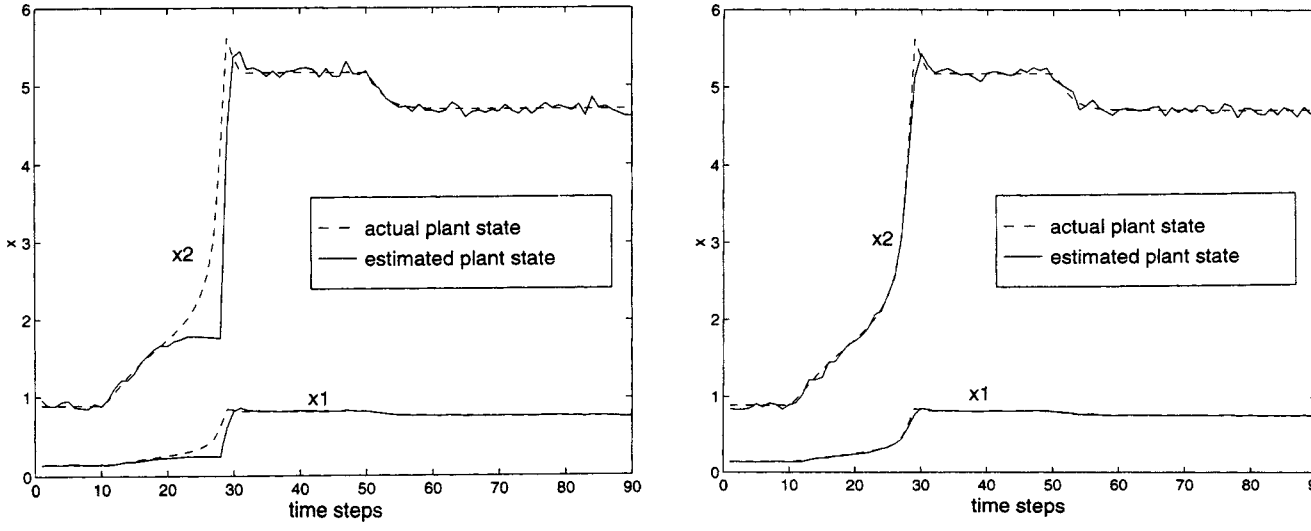


Figure 22. Estimated states for the CSTR using two models and five models.

Table 2. Change in γ with Number of Models

No. of Models	γ Noisy Data	γ Noise-free Data
2	2.9155	3.1581
3	1.0126	1.3211
5	1.0627	1.0645

Table 3. Chemicals Used in the Plant

Monomer A	Methyl methacrylate
Monomer B	Vinyl Acetate
Solvent	Benzene
Initiator	Azobisisobutyronitrile
Chain transfer agent	Acetaldehyde
Inhibitor	*m*-Dinitrobenzene

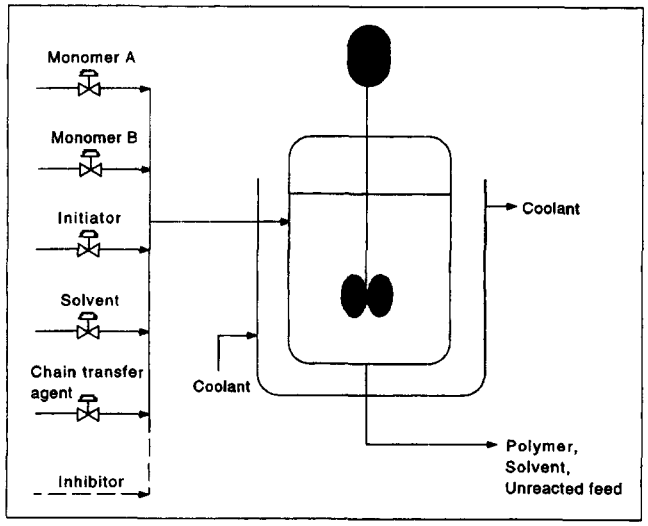


Figure 23. Copolymerization reactor.

ture have been used as manipulated variables. The solvent feed rate and the reactor feed temperature have been treated as measured disturbances, and the inhibitor feed rate has been treated as an unmeasured disturbance. The measured outputs are the polymer production rate, the mole fraction of A in polymer, the weight average molecular weight, and the reactor temperature.

The reactor is run in three different operating regions in order to produce three different grades of polymer. The three grades have weight average molecular weights of about 34,000, 54,000 and 100,000; and the mole fractions of A in the copolymer are 66%, 54%, and 39%, respectively. Table 4 shows the steady-state operating conditions for each of the three modes of operation.

A nonlinear first principles model with 12 states has been taken from Congalidis et al. (1989) to simulate the plant. This is based on a free-radical mechanism with 27 separate reactions. In addition three linear models have been obtained by numerically linearizing the nonlinear model about the three different operating points. The models were then discretized with a sampling time of 1,000 s.

Next, since the inhibitor is being treated as an unmeasured disturbance, the state representing its concentration has been dropped. Therefore, each of the models has only 11 states, which are listed in Table 5. This causes the effect of the inhibitor to become an unmodeled disturbance, and therefore if the flow rate of the inhibitor is increased, the plant moves into a region not covered by any of the local models. This reduction in the model states was intentionally done in order to check the performance of the estimator in such an adverse situation.

The simulations were carried out using the nonlinear model as the plant. The inputs to the reactor are shown in Figure 24. Initially the plant is at steady state in the first mode. At the tenth time step the inputs were ramped up to those of the second mode, and were held there from sampling times of 100 to 200. Then the inputs were ramped to the operating point of the third mode, the transition lasting from time 200 to 800. The inputs were then kept at this operating point until the 1,000th time step, after which they were returned to their original values in the first operating region. This sequence of inputs represents three open-loop transitions between the different grades of polymer, and are shown in the first five graphs of Figure 24. However, an additional disturbance in the inhibitor flow rate has been made at the 900th sampling time. At this time the inhibitor flow rate is stepped up from 0 kg/h to 0.05 kg/h, and this is shown in the sixth graph of Figure 24. Since this is an unmeasured input, its value is not available to the estimator.

Figure 25 shows the plant's response to these input changes. The measurement noise was assumed to be Gaussian with a standard deviation of 1% of the steady-state outputs of the first mode.

Table 4. Steady-State Operating Conditions

Inputs	Grade 1	Grade 2	Grade 3
Monomer A feed rate	18 kg/h	11.5 kg/h	5 kg/h
Monomer B feed rate	90 kg/h	195 kg/h	300 kg/h
Initiator feed rate	0.18 kg/h	0.11 kg/h	0.04 kg/h
Solvent feed rate	36 kg/h	36 kg/h	36 kg/h
Chain transfer agent feed rate	2.7 kg/h	1.35 kg/h	0 kg/h
Inhibitor feed rate	0 kg/h	0 kg/h	0 kg/h
Reactor jacket temperature	336 K	336 K	335 K
Reactor feed temperature	353 K	353 K	353 K
Outputs			
Polymer production rate	20 kg/h	14 kg/h	8 kg/h
Mole fraction of A in polymer	66%	54%	39%
Weight average molecular weight	33,800	53,600	100,000
Reactor temperature	350 K	348 K	346 K

Table 5. States of the Local Linear Models

1	Concentration of monomer A (mol/m^3)
2	Concentration of monomer B (mol/m^3)
3	Concentration of initiator (mol/m^3)
4	Concentration of solvent (mol/m^3)
5	Concentration of chain transfer agent (mol/m^3)
6	Reactor temperature (°C)
7	Molar concentration of A in dead polymer
8	Molar concentration of B in dead polymer
9	Zeroth moment of the molecular weight distribution
10	First moment of the molecular weight distribution
11	Second moment of the molecular weight distribution

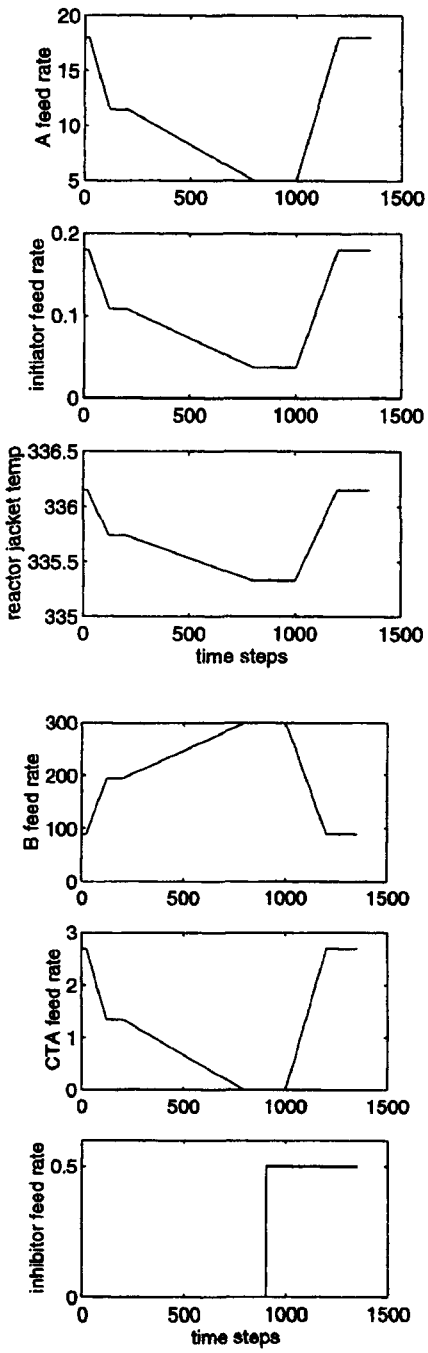


Figure 24. Inputs to the reactor: five manipulated variables and one unmeasured disturbance.

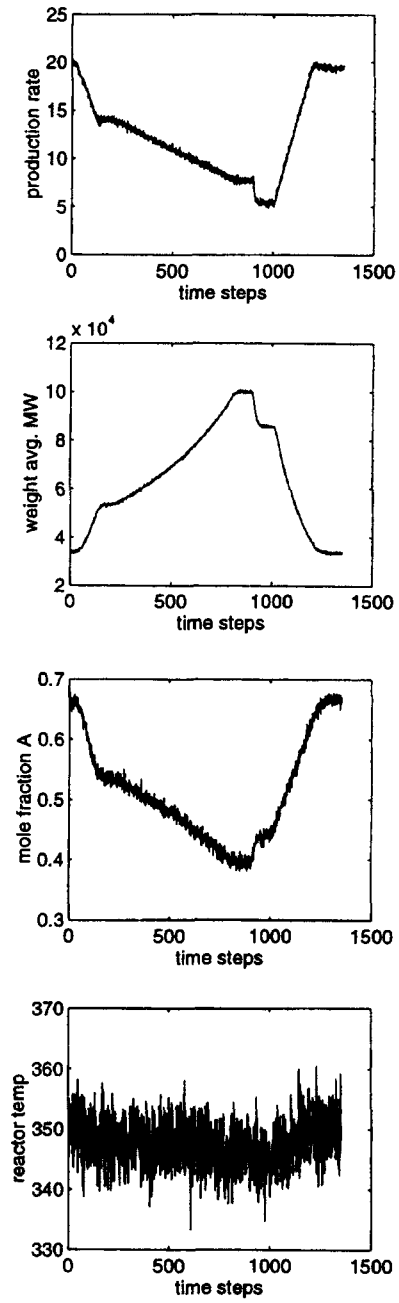


Figure 25. Reactor measurements.

First a BE was designed. The noise characteristics for each of the Kalman filters were chosen to be

$$Q = (0.01X_{s,1})^2 \qquad R = (0.01Y_{s,1})^2,$$

where $X_{s,1}$ and $Y_{s,1}$ are diagonal matrices containing the steady-state values of the states and outputs at the first operating point.

In the second simulation, a MHBE was used. The three Kalman filters used were the same as for the Bayesian estimator. The only additional parameter chosen was a window size of five sampling times in which to estimate the model probabilities.

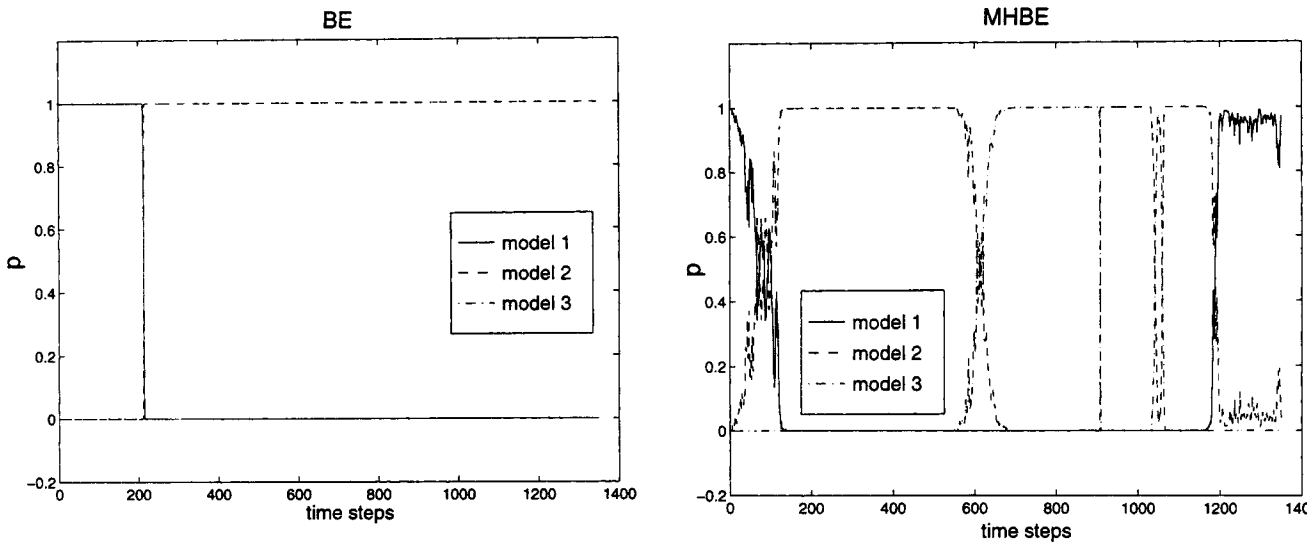


Figure 26. Model probabilities for the copolymerization reactor example using the BE and MHBE.

The MHPE was used in the third simulation with the following weights in the objective function:

$$P^{-1} = \begin{bmatrix} (0.01X_{s,1})^{-2} & 0 \\ 0 & 100I_{3\times 3} \end{bmatrix}$$

$$Q^{-1} = \begin{bmatrix} (0.01X_{s,1})^{-2} & 0 \\ 0 & 100I_{3\times 3} \end{bmatrix}$$

$$R^{-1} = (0.01Y_{s,1})^{-2} \qquad \lambda = 50.$$

Figures 26 and 27 show how the model probabilities and validity functions varied during the simulations. It is clear that the Bayesian estimator failed to detect the second and third transitions. This is due to the problem mentioned earlier in that the probabilities do not switch fast enough. If the plant had been operated much longer in each of the modes, then the Bayesian probabilities would eventually have reflected the transition. It is also shown that the MHBE improves matters considerably. The estimator does detect the transitions, and this shows up in the switching of the model probabilities. At the beginning model 1 has a high probability, but after the transition to the second mode the probability of model 2 approaches 1. Then the transition to the third operating point causes the probability of the third model to approach 1. However, the step disturbance in inhibitor flow rate causes a switch in probability from model 3 to model 2 at the 900th sampling time.

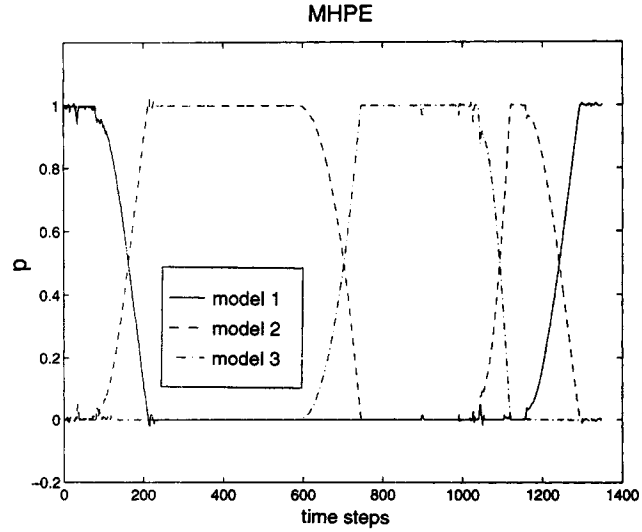


Figure 27. Model probabilities for the copolymerization reactor example using the MHPE.

This must be examined more carefully. The inhibitor flow rate is an unmodeled disturbance, and stepping it up takes the plant into a region not covered by any of the local models. If the manipulated inputs are kept at the mode 3 steady-state values when the inhibitor feed is stepped up, Tables 4 and 6 show that the mole fraction of A and the molecular weights fall in between the steady-state values of the second and third modes. Therefore it is reasonable to expect that the dynamic behavior of the plant will also lie in between these modes. What the MHBE tells us is that the Kalman filter based on the second model outperforms the one based on the third model in this region.

The final transition is also picked up by the estimator despite the disturbance. It turns out that if the plant is in the first operating region, this step increase in inhibitor flow rate does not have much effect on the steady-state outputs, as can seen from Tables 4 and 6. Therefore the MHBE picks the first model eventually. However, in order to reach the first

Table 6. Steady-State Operating Conditions with the Inputs of Modes 1 and 3, with an Inhibitor Feed Rate of 0.05 kg/h

Outputs	Mode 1 with Inhibitor	Mode 3 with Inhibitor
Polymer production rate	19.4 kg/h	5.4 kg/h
Mole fraction of A in polymer	67%	44%
Weight average molecular weight	33,500	85,000
Reactor temperature	350 K	345 K

region, the plant must emerge from a region that is in between the second and third models. Figure 26 shows that after the inputs start to ramp at $t = 1{,}000$, the plant switches between the second and third Kalman filters before eventually picking the first model.

The MHPE also picks up the transitions easily. From Figure 27 it can be seen that until the time of the disturbance at $t = 900$, the behavior of the validity functions is very similar to that of the probabilities calculated by the MHBE. However, at $t = 900$, the validity functions do not switch over to the second model and, unlike the MHBE, the MHPE still picks the third model. This is due to the difference in formulation between the two approaches. The MHPE minimizes the objective function given in Eq. 23. The fourth term of the objective function depends on the distance of the plant output from the points where each of the local models was obtained. Comparing Tables 4 and 6, it is clear that the steady-state outputs are closer to those of the third model than the second one. Therefore the fourth term of the objective function causes the third model to have a higher validity function even after the inhibitor feed has been increased.

The estimated states are shown in Figures 28 and 29, and match the true plant states fairly well. In order to compare the three estimators, the average error in the states was calculated after scaling by the steady-state values of the first region. These values were an average error of 7% for the BE, 2.4% for the MHBE, and 1.8% for the MHPE.

A point to be noted is that the effect of the inhibitor would adversely affect any estimation scheme simply because it is an unmodeled disturbance. Therefore any estimator, linear or nonlinear, would give rise to an offset when the inhibitor feed is stepped up. It is not a weakness of the multiple-model approaches that the states are not exactly matched when the inhibitor is increased. To eliminate this problem the effects of the inhibitor should be added to the local models, or another local model should be added that was identified from data collected when the inhibitor was entering the plant.

Figures 28 and 29 also show how constraints can be handled by the MHPE. The estimated values of four states become negative using the Bayesian approaches. Physically this is not possible since these represent concentrations, and the zeroth and first moments of the molecular weight distribu-

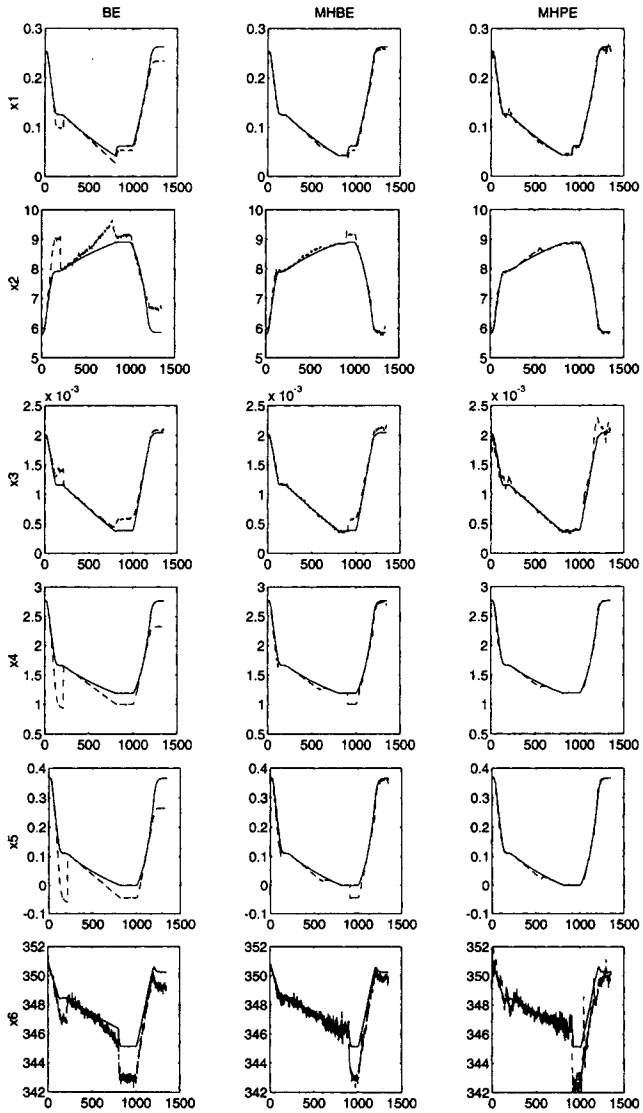


Figure 28. Estimated states—the solid line is the true plant state, and the dashed line is the estimated state.

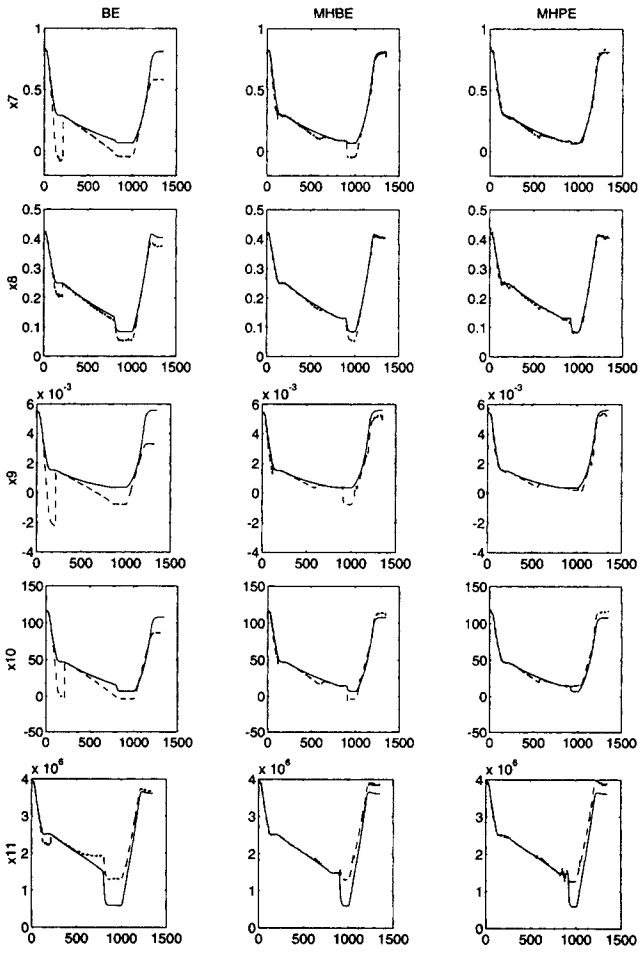


Figure 29. Estimated states—the solid line is the true plant state, and the dashed line is the estimated state.

tion. However, when the unmodeled disturbance starts at $t = 900$, all the local models become inaccurate, and so the Kalman filters based on them give wrong predictions, which is this case are negative. Since constraints cannot be incorporated into the Bayesian approach, this situation cannot be alleviated. The MHPE estimator, on the other hand, does not suffer from this weakness, as constraints on the estimated states can be explicitly included in the optimization.

Conclusions

Three multiple-model-based estimators have been described for processes that operate in multiple regimes. Such estimators are useful when it is difficult to produce an accurate nonlinear model of the plant that is also convenient to use. This article has outlined approaches that use multiple local linear models of the plant instead, each valid in some particular operating region of interest.

A method of quantifying the mismatch between the unknown nonlinear plant and the global model constructed from the local ones has also been discussed. This is based on an estimate of the minimum uncertainty that would explain plant input/output data and not invalidate the model. This measure of uncertainty can also be used to decide how many local models are required to describe the nonlinear plant. Therefore it may also be considered a measure of the nonlinearity of the process.

The estimation procedures outlined in this article have been shown to work on three test systems. These examples were chosen because they are nonlinear systems being operated in different regions, and the dynamics in each of the operating regions is different. Also each of the examples was selected in order to illustrate different aspects of the behavior of the estimators. The first system, which was a simple numerical example, illustrated the behavior of the estimators in an idealized setting without measurement noise or disturbances. The probabilities were shown to switch during transition, and the effect of changing the estimation horizon was studied. Next, in the CSTR example, the stability characteristics of the local models change depending on where the CSTR is operated. The effect of varying the number of models was examined, and the amount of uncertainty in the system was estimated. Finally, the copolymerization reactor was used to apply the estimators to a large-scale system, with noise and unmodeled disturbances, which take the plant into regions not covered by any of the local models.

In each of the examples, the MHPE outperformed the MHBE. Furthermore the MHPE offers the flexibility of incorporating constraints on the estimated states. However, these advantages come with the increased computational burden of having to perform a constrained optimization at each time step. Since the MHBE does not have an explicit optimization, it is a lot quicker than the MHPE. Therefore there is a trade-off between accuracy and computing resources needed. For the two industrial examples presented here, the CSTR and the copolymerization reactor, it is likely that the state estimates from the MHBE are close enough, and the additional burden of using the MHPE is not warranted. However, if a more precise knowledge of the states is required, as in the first test system, for example, the MHPE must be used.

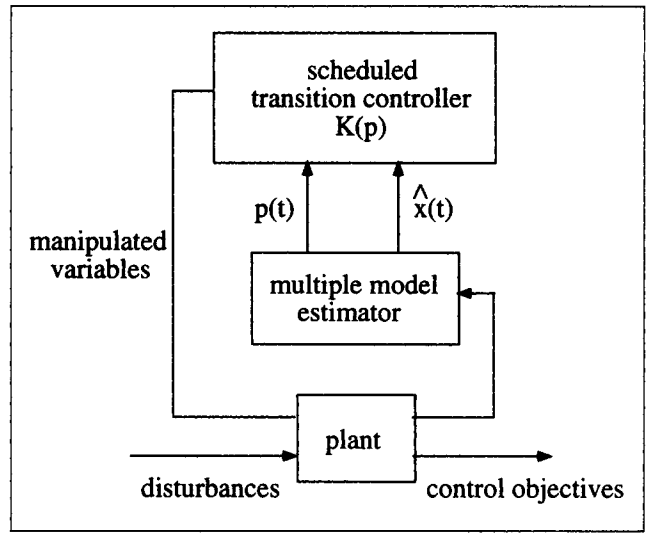


Figure 30. Transition controller using a multiple-model estimator.

The method outlined in this article are not meant to be an end in themselves, but instead provide a basis for controller design for plants that undergo transitions. It is expected that such transition controllers will make use of the model probabilities or validity functions to modify their behavior, depending upon which regime the plant is being operated in. This is shown in Figure 30, where a scheduled transition controller is being used. A method of transition control that takes advantage of the structure of the multiple-model-based estimator has been presented in Banerjee et al. (1995).

Acknowledgments

The first two authors gratefully acknowledge the financial support of E. I. DuPont de Nemours & Co., Inc., and the National Science Foundation through Grant CTS-9522564.

Literature Cited

Apkarian, P., P. Gahinet, and G. Becker, "Self-scheduled H_∞ Control of Linear Parameter-varying Systems," *Proc. Amer. Control Conf.*, p. 856 (1994).

Banerjee, A., Y. Arkun, B. Ogunnaike, and R. Pearson, "H_∞ Control of Nonlinear Processes Using Multiple Linear Models," *Proc. Eur. Control Conf.*, p. 2671 (1995).

Congalidis, J. P., J. R. Richards, and W. H. Ray, "Feedforward and Feedback Control of a Solution Copolymerization Reactor," *AIChE J.*, **35**, 891 (1989).

Deshpande, J. G., T. N. Upadhyay, and D. G. Lainiotis, "Adaptive Control of Linear Stochastic Systems," *Automatica*, **9**, 107 (1973).

Hilborn, C. G., and D. G. Lainiotis, "Optimal Estimation in the Presence of Unknown Parameters," *IEEE Trans. Syst. Sci. Cybern.*, **SSC-5**(1), (1969).

Johansen, T. A., and B. A. Foss, "Representing and Learning Unmodeled Dynamics with Neural Network Memories," *Proc. Amer. Control. Conf.*, p. 3037 (1992).

Johansen, T. A., and B. A. Foss, "Identification of Nonlinear System Structure and Parameters Using Regime Decomposition," *Automatica*, **31**(2), 321 (1995).

Lainiotis, D. G., "Optimal Adaptive Estimation: Structure and Parameter Adaptation," *IEEE Trans. Automat. Contr.*, **AC-16**(2), (1971).

Lainiotis, D. G., "Partitioning: A Unifying Framework for Adaptive Systems. I. Estimation," *Proc. IEEE*, 1126 (1976).
Murray-Smith, R., and T. A. Johansen, eds., *Multiple Model Approaches to Nonlinear Modeling and Control*, Taylor & Frances, London (1997).
Poolla, K., P. Khargonekar, A. Tikku, J. Krause, and K. Nagpal, "A Time-Domain Approach to Model Validation," *IEEE Trans. Automat. Contr.*, **AC-39**(5), 951 (1994).
Ravi, R., R. M. Nagpaul, and P. Khargonekar, "H_∞ Control of Linear Time-Varying Systems: A State-Space Approach," *SIAM J. Control. Optimiz.*, **29**, 1394 (1991).
Robertson, D. G., and J. H. Lee, "A Least Squares Formulation for State Estimation," *J. Process Control*, **5**(4), p. 291 (1995).
Uppal, A., W. H. Ray, and A. B. Poore, "On the Dynamic Behavior of Continuous Stirred Tank Reactors," *Chem. Eng. Sci.*, **29**, 967 (1974).
Willsky, A. S., E. Y. Chow, S. B. Gershwin, C. S. Greene, P. K. Houpt, and A. L. Kurkjian, "Dynamic Model-Based Techniques for the Detection of Incidents on Freeways," *IEEE Trans. Automat. Contr.*, **A-25**(3), 347 (1980).

Manuscript received May 30, 1996, and revision received Dec. 26, 1996.