MASTER

Multi data set parameter estimation applied on a spray dryer process

van Gessel, J.G.

Award date:
1991

DEPARTMENT OF ELECTRICAL ENGINEERING

EINDHOVEN UNIVERSITY OF TECHNOLOGY

group measurement & control

Multi Data Set Parameter Estimation

applied on a
spray dryer process

by: J.G. van Gessel

This report is submitted in fulfilment of the requirements of the degree of electrical engineer (M.Sc.) at the Eindhoven University of Technology. The work was carried out from december 1990 to august 1991 at IPCOS b.v. 's-Hertogenbosch, in charge of:

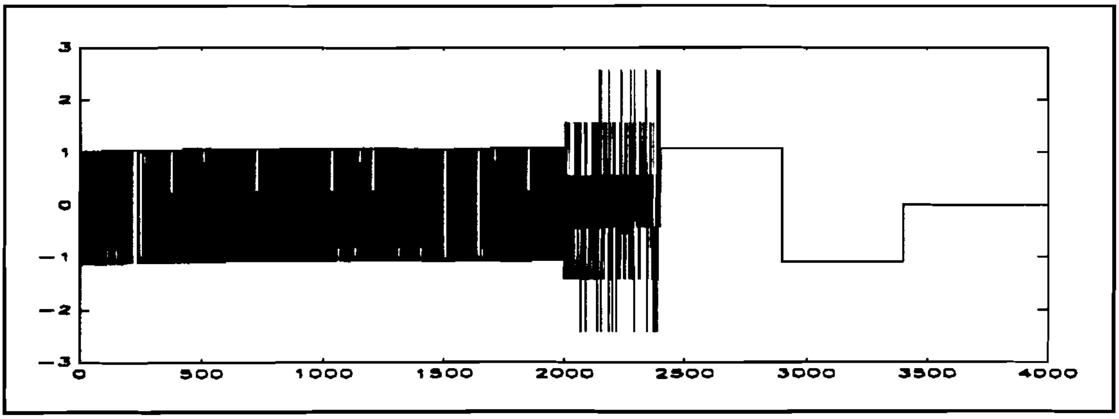
Prof.Dr.Ir. A.C.P.M. Backx

under supervision of:

Dr.Ir. Y.C. Zhu
Ir. J. Ludlage

Multi Data Set Parameter Estimation

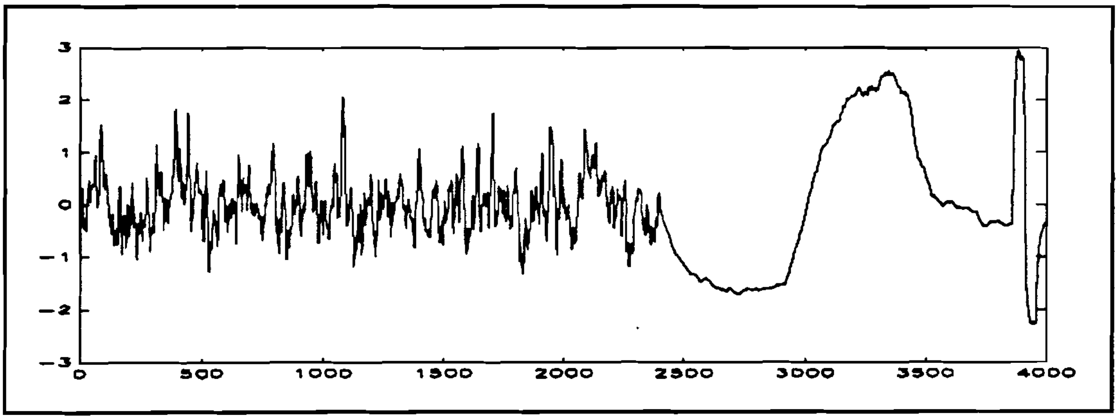
applied on a spray dryer process



author: J.G. van Gessel
date: august 1991

Multi Data Set Parameter Estimation

applied on a spray dryer process



Master science thesis
Department of electrical engineering
Eindhoven University of Technology
the Netherlands
august 1991
author: J.G. van Gessel (232794)
supervising: ir. J. Ludlage
dr.ir. Y.C. Zhu
prof.dr.ir. A.C.P.M. Backx

project performed at IPCOS b.v. 's-Hertogenbosch
the Netherlands

Preface

The work described in this report was carried out at IPCOS b.v. in 's-Hertogenbosch from december 1990 to august 1991. This report describes the modelling of the milk spray dryer tower with some new techniques. In a parallel research project Pieter van Gelder studied H-infinity controllers for the spray dryer tower. I thank him for providing the controllers for my models of the spray dryer and for all thinking, support and pleasure.

I would like to thank IPCOS group for providing the opportunity to gain experience with respect to this subject in an industrial environment. I considered it to be important to work in industry rather than on university during the last part of my studies.

In the IPCOS group I especially would like to thank my mentors Jobert Ludlage and Yu-Cai Zhu for their help on this subject. Also Michiel van Wijck is thanked for providing all the information on spray dryer processes. Of course all other members of IPCOS group are thanked hereby for their services.

I thank professor Backx for providing the subject at IPCOS group and his supervising.

I would like to use this opportunity to thank my family for their support during the time of my studies in electrical engineering.

Abstract

The work described in this report was carried out at IPCOS b.v. in 's-Hertogenbosch. IPCOS group works out modern model based multivariable process control for companies in varying process areas. For advanced controller design detailed and accurate models of the processes are needed. IPCOS group developed some model estimation algorithm which can be applied to various processes. In this model estimation scheme various data sets are measured from the process to identify the model. These data sets are:

- step response data sets (step input signals applied on the process)
- a fast PRBNS data set (fast PRBNS input signals applied on the process)
- a (final) PRBNS data set (PRBNS input signals applied on the process)

From these data sets one builts up knowledge on the process. For the final model estimation however only the last (final) PRBNS data set is used. So the question arises whether the data sets measured previous to the final PRBNS data set can be used in the final model estimation. One hopes that those data sets which contain information about specific frequency areas might lead to some improvement on the final process model.

In a parallel research project a student in mathematics studied the application of H-infinity control on a milk spray dryer process (a simulation pilot plant). For this controller design models of the milk spray dryer process are needed. Therefore I tried modelling the spray dryer process when using more data sets in estimation.

I tried coupling the data sets when using the estimation algorithm of IPCOS group. Results show that using a modified fast PRBNS data set and the final PRBNS data set gives some model improvement at high frequencies as expected.

Results show that using a low pass filtered PRBNS data set and the final PRBNS data set gives some improvement on the model at low frequencies.

When using a step response data set, a modified PRBNS data set and the final PRBNS data set (which was the main goal) as an estimation data set a problem with the estimation algorithm of IPCOS group, which is based on FIR model estimation, occurred due to the nature of the process. Therefore another estimation algorithm is developed which can handle processes with strongly varying time constants. The developed estimation technique, based on ARX model estimation, with all data sets incorporated in the estimation data set improves the model for all frequencies; especially the DC-gain. This technique might work for other processes (like the spray dryer process) with strongly varying time constants as well.

For a standard model of the milk spray dryer, estimated with the IPCOS estimation scheme, and a model, estimated with the newly developed multi data set estimation technique, pole placement (L_2 norm) and H-infinity norm controllers were designed. Results show an improvement in disturbance variance of the main output of about 40% for the multi data set model in comparison with the standard model when the system is controlled by a pole placement controller. Also change-over speed and decoupling are improved. For the H-infintity controlled system the improvement in disturbance variance is about 75% and change-over speed is increased also. This gives reason to believe that model estimation from all data sets is usefull for modern multivariable process control and model estimation.

Contents

1. Introduction.

Automatic process control is of increasing importance in industry due to growing demands on the quality of products and processes. Classical control techniques prove to be insufficient for controlling MIMO systems, especially towards changeovers. These techniques are useful for most SISO control loops in setpoint controllers, but cannot be used for the overall control of the critical process units.
Therefore, several advanced control strategies have been developed (LQG, IMC, adaptive and robust control). For advanced controller design detailed and accurate models of the process are needed. So first a good identification procedure must take place, because the performance obtainable for the controlled process is directly related to the accuracy of the underlying process model. We here follow the identification scheme for simulation models proposed by Backx and Damen [1].
This scheme contains several steps in which various experiments must take place before obtaining the final model. Each experiment yields a data set with information on a specific aspect of the process. In the final estimation of the process model only the last data set is used (final PRBNS data set). This final PRBNS input signal is a compromise signal with respect to power content and frequency range. A drawback of using only this signal is that previously recorded data sets are not used, although they provide information about the process characteristics at specific frequencies. A fast PRBNS signal could give some information about high frequency response and the step response could give additional information about low frequency (static) response of the process. In this report the possible use of data sets collected previous to the final PRBNS data set for the estimation of the final process model is considered. We might expect the final model to resemble the actual process closer than with the approach described before.
The general aim of using more data sets is to provide a better model with a smaller model uncertainty which leads to a better (robust) controller.
This project was carried out at the IPCOS b.v. (Integrated Production Control Systems) in 's-Hertogenbosch. This unit, managed by prof.dr.ir. A.C.P.M. Backx, works out modern process integrated control for companies in varying process areas. Due to prof. Backx the IPCOS group has close bonds with the Eindhoven University of Technology Measurement & Control Group.

2. Identification of industrial processes.

2.1 Introduction.

As stated in chapter one we need a good identification scheme to obtain a qualified model of the industrial process. Such a scheme was proposed by Backx and Damen [1] and will be used as a reference. This identification scheme was chosen for its stepwise execution of experiments which builds up knowledge about the process. New ideas can be implemented quite easily in this scheme.

2.2 The identification scheme.

The identification scheme [1] used consists of various steps in time:

operation:	product:
1. study of process talks with operators	
2. noise measurements at output (all inputs kept constant)	characterization of output noise frequency range
3. staircase inputs with duration and amplitude	largest time constant linearity, static gains
4. PRBNS input signal with high clock frequency and short duration for selected inputs	S/N ratio for each output small time constants choice of inputs/outputs
5. PRBNS input signal with clock frequency, sampling time and duration (adjusted to known process characteristics)	final raw data set
6. signal processing: - peak shaving - trend correction - offset correction - scaling - filtering - compensating time delays - sample rate reduction	final data set
7. FIR model estimation of proper length	high order model
8. model reduction to MPSSM	initial estimate MPSSM
9. adjustment of MPSSM to final estimation data set	final MPSSM model
10. model reduction	final low order model
11. validation	

2.3 State space descriptions of linear systems.

The state space description is mainly used for modelling of sampled processes in modern system and control theory. The state relationships are given by:

$\mathbf{x(k+1) = Ax(k) + Bu(k) + v(k)}$	state equation
$\mathbf{y(k+1) = Cx(k) + Du(k) + n(k)}$	output equation
$x(0)$	initial condition

The matrices are:

A = system matrix
B = distribution matrix
C = output or measurement matrix
D = input-output matrix (or direct feed through)

The disturbances are:

v(k) = process noise
n(k) = observation noise

Schematically the state space representation is given in fig. 2.1 [10].

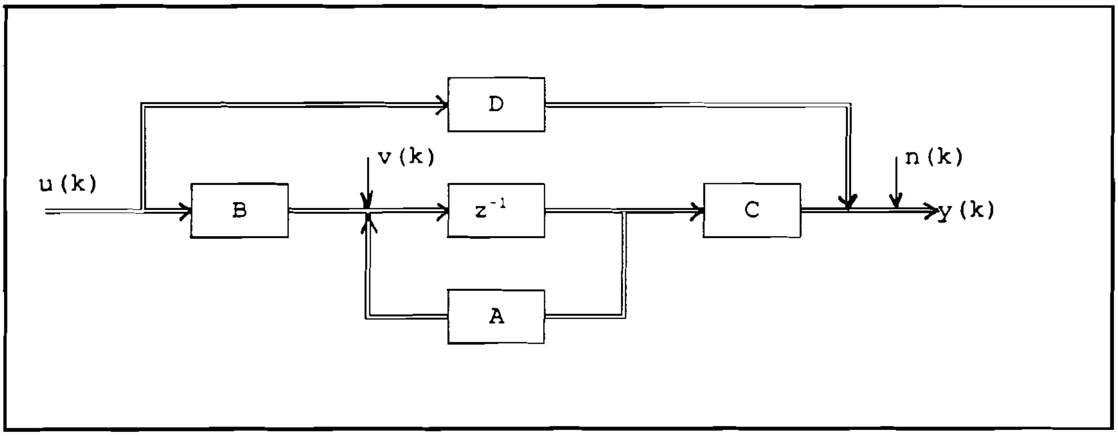


figure 2.1: *the state space description schematically [10].*

In the following sections the estimating scheme of section 2.2 is illustrated briefly. For detailed information see [8].

2.4 FIR model estimation.

FIR models are Finite Impulse Response models. In the identification scheme a FIR model is estimated first, because this model is a linear fit from the FIR parameters on the process data and that is easy to compute. The FIR model will be used as an initial model for the direct estimation of the state space model (see section 2.5).
With the final PRBNS a FIR model with M parameters is estimated [2]. In case of a SISO system this works as follows:

model: $y(t) = h_0u(t) + h_1u(t-1) + .. + h_{M-1}u(t-M+1)$

define: $\phi(t) = (u(t) \;\; u(t-1) \;\; .. \;\; u(t-M+1))^T$

$\theta = (h_0 \;\; .. \;\; h_{M-1})$

In practice M being of the order 20-50 for an accurate description of the systems dynamics.

In matrix notation:

$Y = \theta{\cdot}\Phi$

where:

$Y = [y(1) \; y(2) \; ... \; y(N)]$ (1xN)

$\Phi = [\phi(1) \; \phi(2) \; ... \; \phi(N)]$ (MxN)

introduce the error $\varepsilon(t) = y(t) - \theta{\cdot}\phi(t)$
so:

$\varepsilon = [e(1) \; e(2) \; ... \; e(N)]$ (1xN)

Now the least squares estimate of θ is defined as $\hat{\theta}$ that minimizes the loss function

$$V(\theta) = \varepsilon\varepsilon^T = \| \varepsilon \|^2$$

Then $V(\theta)$ has a unique minimum point given by:

$$\hat{\theta} = Y\Phi^T(\Phi\Phi^T)^{-1}$$

The corresponding minimal value of $V(\theta)$ is

$$\min V(\theta) = V(\hat{\theta}) = [YY^T - Y\Phi^T(\Phi\Phi^T)^{-1}\Phi Y^T]$$

This FIR model is estimated with an output error least squares method.

2.5 MPSSM model estimation.

The transition from a FIR model with a large number of parameters to a state space model without the need for detailed information about the Kronecker indices is performed by creating a Markov parameter model abbreviated by MPSSM (Minimal Polynomial and Start Sequence of Markov parameter model) [8].

Definition of a MPSSM model set:
Let the minimal polynomial of a discrete time causal system be given by:

$$s^r + a_1 s^{r-1} + \ldots + a_{r-1} s + a_r = 0$$

If the first r Markov parameters are given by M_j (j=1..r) and the direct feed through by D then the output y of this system generated by input u is given by the following convolution:

$$y_k = \sum_{j=0}^{\infty} F_j u_{k-j}$$

with F_j equals:

$$F_j = \begin{cases} D & \text{if } j=0 \\ M_j & \text{for } j=1..r \\ -\sum_{i=1}^{r} a_i F_{j-i} & \text{for } j > r \end{cases}$$

This MPSSM model will be fitted on the data in an output error least squares criterium and after that the model will be transformed into a state space model with some model reduction technique. This can be done while the MPSSM model has a considerably larger McMillan degree than the degree of the minimal state space representation. The extra degrees of freedom are the multiplicity of each pole. This multiplicity is removed afterwards by the model reduction technique.
In order to speed up the the minimization process a good initial estimate of the parameters is necessary. This initial estimate is obtained by fitting the MPSSM model to the FIR model by the Gerth method.

The degree r of the minimal polynomial is based on the singular values of the block Hankel matrix:

$$H = \begin{vmatrix} vec(M_1) & vec(M_2) & \dots & vec(Mj_r) \\ vec(M_2) & vec(M_3) & \dots & vec(M_{j+1}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ vec(M_i) & vec(M_{i+1}) & \dots & vec(M_L) \end{vmatrix}$$

With M_j are the Markov parameters of the estimated FIR model (j=1..L) and $vec(M_i)$ indicates that all columns of M_i are put below one another into one vector.
A sharp decrease of the singular values gives us the candidate r.

With this r the fit of the MPSSM model to the FIR model with the Gerth method can be accomplished. The Gerth method will give us the estimated MPSSM parameters a_i and thus the complete initial model. Gerths method consists of:

$$G\,a = v$$

with:

$$G = \begin{vmatrix} vec(M_1) & vec(M_2) & \dots & vec(M_r) \\ vec(M_2) & vec(M_3) & \dots & vec(M_{r+1}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ vec(M_{L-r}) & vec(M_{L-r+1}) & \dots & vec(M_{L-1}) \end{vmatrix}$$

$$v^T = [vec(M_{r+1})^T\ vec(M_{r+2})^T\ \dots\ vec(M_L)^T]^T$$

$$a^T = -[a_r\ a_{r-1}\ \dots\ a_1]^T$$

The solution $a = (G^TG)^{-1}G^Tv$ provides an estimate for the minimal polynomial coefficients.

3. Multi Data Set Parameter Estimation.

3.1 Introduction.

In the scheme of section 2.2 we can see that several data sets are generated during the estimation. These data sets are:

- freerun data set (inputs constant)
- step responses
- fast PRBNS data set
- final PRBNS data set

For the final model estimation however, only the last PRBNS data set is used. In this chapter a method is proposed to use other data sets (step responses and fast PRBNS data) as well for the final model estimation. One hopes to achieve a better model without doing extra measurements this way.

3.2 The upper bound of the modelling error.

If we could derive an upper bound for the modelling error, we might be able to prove what happens if we also use other data sets than the final PRBNS data set. An upper bound of the modelling error for a SISO FIR model was derived by Zhu [3].
Consider a linear time-invariant discrete time process.
Define the modelling error in the frequency domain as:

$$\Delta G(\exp(i\omega)) = G_o(\exp(i\omega)) - \hat{G}(\exp(i\omega))$$

where $G_o(\exp(i\omega))$ is the true transfer function of the process
and $\hat{G}(\exp(i\omega))$ is the nominal model of $G_o(\exp(i\omega))$

$\Delta G(\exp(i\omega))$ is the sum of a bias part and a random (variance) part:

$$\Delta G(\exp(i\omega)) = E\{\Delta G(\exp(i\omega))\} + [\Delta G(\exp(i\omega)) - E\{\Delta G(\exp(i\omega))\}]$$

The variance part is proven to be asymptotically normal [3], with variance $\sigma(\omega)$ and zero mean value under the following conditions (let n = number of FIR parameters and N = number of samples).
Here y(t) and u(t) are the output and input signal respectively at time t. v(t) is a stationary process of random variables, with zero mean values.
The formal assumptions are [9]:

(C1) the real process is given by:

$$G_0(\exp(i\omega)) = \sum_{k=1}^{\infty} g_0(k)\ \exp(-ik\omega), \qquad -\pi \le \omega \le \pi$$

this is a time-invariant **linear** model!
with:

$$\sum_{k=1}^{\infty} k \cdot |g_0(k)| < \infty \qquad \text{i.e. assymptotic stable.}$$

(C2)

$$E[v(t)v(t+\tau)] = \gamma_v(\tau), \qquad \sum_{\tau=-\infty}^{\infty} |\tau\gamma_v(\tau)| < \infty$$

(C3)

$$\Phi_v(\omega) = \sum_{\tau=-\infty}^{\infty} \gamma_v(\tau)\exp(-i\tau\omega), \qquad \Phi_v(\omega) < x \qquad \forall\omega$$

(C4)

$$n(N) \to \infty \quad as \quad N \to \infty$$

(C5)

$$\frac{n^2}{N} \to 0 \quad as \quad N \to \infty$$

(C6)

$u(t)$ *is independent of* $v(t)$ *and* $|u(t)| \le C_1, \ \forall t$

Let

$$\gamma_u^N(\tau) = \frac{1}{N}\sum_{t=1}^{N} u(t)u(t+\tau)$$

$$\Phi_u^N(\omega,n) = \sum_{\tau=-n}^{n} \gamma_u^N(\tau)\exp(-i\tau\omega)$$

(C7)

$$\lim_{N\to\infty} inf\ \Phi_u^N(\omega,n(N)) \ge \delta, \qquad \delta > 0, \qquad \forall\omega$$

(C8)

$$\Phi_u^N(\omega, n(N)) \rightarrow \Phi_u(\omega) \quad as \quad N \rightarrow \infty$$

$$where \quad \Phi_u(\omega) = \sum_{\tau=-\infty}^{\infty} E[u(t)u(T+\tau)]\exp(-i\tau\omega)$$

(C9)

$$C_2 \geq \Phi_u(\omega) \geq \delta > 0, \quad \forall\omega$$

(C10)

$$\frac{1}{\sqrt{n(N)}} \sum_{\tau=-2n(N)}^{2n(N)} |\tau\gamma_u^N(\tau)| \rightarrow 0 \quad as \quad N \rightarrow \infty$$

(C11)

$$\sum_{k=1}^{\infty} \left(\frac{n(k^2)}{k}\right)^2 < \infty$$

(C12)

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k)$$

For the spraydryer experiments the assumption that does **not** hold is:

(C1): The spraydryer process is nonlinear. One tries to linearize the process by estimating the model around one operating point with small input signals.

Then for a normal distribution:

$|\Delta G(\exp(i\omega)) - E\{\Delta G(\exp(i\omega))\}| \leq 3\sigma(\omega)$ w.p. 99.7%

Define $UB(\omega)$ as an upper bound of $\Delta G(\exp(i\omega))$ such that

$|\Delta G(\exp(i\omega))| \leq UB(\omega), \ -\pi \leq \omega \leq \pi$

So

$UB(\omega) = |E\{\Delta G(\exp(i\omega))\}| + 3\sigma(\omega)$

suppose that N data samples have been collected from the process:

Z^N : y(1) u(1),..., y(N) u(N)

Then the variance $\sigma^2(w)$ is proved [3] to be

$$var\ \hat{G}_N^n(\exp(i\omega)) = \frac{n\Phi_v(\omega)}{N\Phi_u(\omega)}\ \ if\ N\rightarrow\infty$$

where n = order of FIR estimation (M in section 3.2)
$\Phi_v(\omega)$ = power spectrum of the process disturbances
$\Phi_u(\omega)$ = power spectrum of the process input signal
N = number of samples

In practice the upper bound $UB(\omega)$ is calculated as

$$UB(\omega) = |\hat{G}_N^n(\exp(i\omega)) - \hat{G}_N^l(\exp(i\omega))| + 3\sqrt{\frac{n\ \Phi_v(\omega)}{N\ \Phi_u(\omega)}}$$

in which $\hat{G}_N^l(\exp(i\omega))$ is the low order estimate of the process.

The upper bound $UB(\omega)$ can also be derived for MIMO processes and models. This leads to the following results ([4] and [5])

Let $\hat{G}_i(\exp(i\omega)) = [\hat{g}_{i1}(\exp(i\omega)) \ .... \ \hat{g}_{im}(\exp(i\omega))]^T$

where $\hat{G}_i(\exp(i\omega))$ is a column of the estimated transfer matrix
$\hat{g}_{ij}(\exp(i\omega))$ is element i,j of the estimated transfer matrix

Then

$$var\ \hat{g}_{ij}^{n}(\exp(i\omega)) = \frac{n\ \Phi_{vi}(\omega)}{N\ [\Phi(\omega)]_{jj}} = \sigma_{ij}^{2}(\omega) \quad [3.1]$$

and

$$UB_{ij}(\omega) = |\hat{g}_{ij}^{n}(\exp(i\omega)) - g_{ij}^{l}(\exp(i\omega))| + 3\sigma_{ij}^{2}(\omega) \quad [3.2]$$

Algorithms have been designed to estimate this upper bound [5].

3.3 Practical use of multi data set parameter estimation.

With the multi data set parameter estimation several goals can be achieved:

1. in case of a regular process identification one might now be able to use data sets for identification that were collected previously to the final PRBNS data set. A fast PRBNS data set might give more information about the process characteristics at higher frequencies. A step response can put weight to low frequency characteristics.
2. when process identification is performed and a controller is designed one can calculate at which frequencies the controller mainly operates by Fourier transformation of the steering signal. A new identification experiment can now be performed by designing an input signal (of filtered white noise) which has power content in that specific frequency band. The results of that new experiment can be used with the old data sets for a new (improved) identification which may lead to an improved process model and controller.
3. Specific experiments can be done with input signal power content at interesting frequency areas. For instance in the frequency area where the noise power (estimated from the freerun experiment!) is high. To achieve a reasonable performance in this area the model error in that area must be small. An experiment can now be done with extra power content in that frequency band.
4. When starting up a (nonlinear) process we go through several operating points. For each operating point a model must be estimated and a controller must be designed. When the process goes to another operating point another controller is used. With the multi data set parameter estimation a sort of general startup process model can be estimated from all data sets. Only one robust controller has to be designed now for the startup procedure.

With this we might encounter a specific problem when coupling the data sets. On the coupling points we will lack the presence of switching characteristics between the two data sets. This because generally the data sets are measured independent from one antother. The total signal is not stationary. We might however expect that the lack of that switching information will not be of significant importance when the data sets are large enough compared to the largest time constant of the process.

3.4 Estimation from more than one data set: MDSPE_1.

Suppose we have a process (either SISO or MIMO) with known characteristics (for instance a FIR model with M parameters).

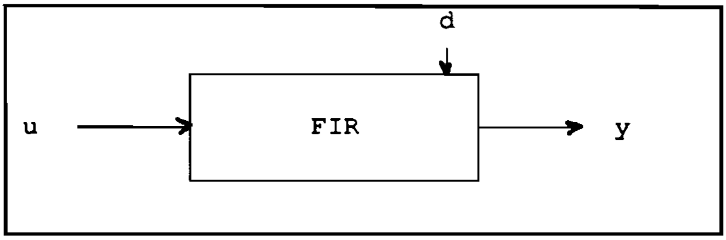


figure 3.1: *process.*

Now (in case of two data sets) two input signal matrices $u_{N1}(t)$ and $v_{N2}(t)$ will excite the process and will generate the open loop output signal matrices $y_{N1}(t)$ resp. $z_{N2}(t)$ (N1,N2 number of data samples).
Suppose $v_{N2}(t)$ is the final PRBNS input signal and $u_{N1}(t)$ is an earlier used version. In the present situation only $v_{N2}(t)$ and $z_{N2}(t)$ were used for parameter estimation leading to an upper bound for the estimation as in formula [3.2].

The trivial solution to reduce the variance term [3.1] would be to:

- increase N (=number of samples)
- increase the input power spectrum $\Phi_u(\omega)$

Both solutions have disadvantages:

Increasing N means that more data must be generated, which is expensive (more product will be lost; more time is needed).

Increasing the input power is not always possible due to nonlinearity in the process. A PRBNS input signal also has a specific power spectrum which depends upon the clock speed [10]. By choosing the clock speed for the final PRBNS signal so that the whole process frequency band is covered the power spectrum cannot be changed. The final PRBNS signal therefore is a compromise signal with equal power content at all frequencies.

I therefore propose two methods for making use of more than one data set, each with its own advantages and disadvantages.
The experiment in which the final and fast PRBNS data sets are joined before estimation takes place is referred to as MDSPE_1. The fast PRBNS data set provides more information at the higher frequencies about the process. For just those higher frequencies the controller gain will be high (for speeding up the process) and modelling errors then will influence the stability of the controlled system. By decreasing the modelling errors by incorporating the fast PRBNS data set robustness of the controlled system at high frequencies will be higher and the performance will be improved.

3.4.1 Method one.

A way of increasing N and Φ_{input} is to make use of other data sets collected besides the final PRBNS data set. The model would be better estimated and the model error in the frequency domain would be reduced.

To start with 2 data sets with input signals (scaled around zero): $u_{N1}(t)$ and $v_{N2}(t)$.

$u_{N1}(t)$ has a power spectrum $\Phi_u(\omega)$
$v_{N2}(t)$ has a power spectrum $\Phi_v(\omega)$.

The power spectrum $\Phi_u(\omega)$ is defined by (process with m inputs)

$$\textit{covariance}:\ R_u(\tau) = \frac{1}{N_1}\sum_{t=1}^{N_1} u(t)\bullet u^T(t+\tau) \qquad \{\text{mxm matrix}\}$$

$$\textit{power spectrum}:\ \Phi_u(\omega) = \sum_{\tau=-n}^{n} R_u(\tau)\ \exp(-i\tau\omega) \qquad \{\text{mxm matrix}\}$$

(A similar expression is found for the power spectrum $\Phi_v(\omega)$ with m inputs)

If m=1:
Now $u_{N1}(t) = [u(1) \ldots u(N1)]$ (N1x1 vector)
and $v_{N2}(t) = [v(1) \ldots v(N2)]$ (N2x1 vector)

Make
$u_{N1}(t) = [u(1) \ldots u(N1)\ 0 \ldots 0]$ ((N1+N2)x1 vector)
$v_{N2}(t) = [0 \ldots 0\ v(1) \ldots v(N2)]$ ((N1+N2)x1 vector)

Now define $w_{N1+N2}(t) = u_{N1}(t) + v_{N2}(t)$
$= [u(1) \ldots u(N1)\ v(1) \ldots v(N2)]$

So the length of w is (N1+N2).

In the variance term [3.1] N can now be replaced by (N1+N2) which reduces the term as we wanted.
The problem arises when we look at the power spectrum of the composed signal $w_{N1+N2}(t)$. By combining $u_{N1}(t)$ and $v_{N2}(t)$ we created in most cases a non-stationary signal $w_{N1+N2}(t)$. Therefore we cannot just add the spectra $\Phi_u(\omega)$ and $\Phi_v(\omega)$ for obtaining $\Phi_w(\omega)$. We see that as we take a closer look at the covariance functions:

$$R_u(\tau) = \frac{1}{N_1} \sum_{t=1}^{N_1} u(t) \cdot u^T(t+\tau)$$

$$R_v(\tau) = \frac{1}{N_2} \sum_{t=1}^{N_2} v(t) \cdot v^T(t+\tau)$$

$$R_w(\tau) = \frac{1}{N_1+N_2} \sum_{t=1}^{N_1+N_2} w(t) \cdot w^T(t+\tau)$$

Suppose that if t > N_1 : u(t) = 0 and if t > N2 : v(t) = 0.
Then (for m=1):

$$R_w(\tau) = \frac{N_1}{N_1+N_2} R_u(\tau) + \frac{N_2}{N_1+N_2} R_v(\tau) + \frac{1}{N_1+N_2} \sum_{k=1}^{\tau} v_k \, u_{N_1-\tau+k}$$

From this last formula we see crossterms appear which won't be nil if $u_{N1}(t)$ and $v_{N2}(t)$ are correlated. But for N_1 and N_2 very large we expect the crossterms to go to nil in the case of (filtered) white noise sets and we then can replace $\Phi_u(\omega)$ by $\Phi_w(\omega)$ in [3.1]:

$$\Phi_w(\tau) = \frac{N_1}{N_1+N_2} \Phi_u(\tau) + \frac{N_2}{N_1+N_2} \Phi_v(\tau)$$

$$var\ \hat{g}_{ij}^{n}(\exp(i\omega)) = \frac{n\ \Phi_{v_i}(\omega)}{[N_1 \Phi_u(\omega) + N_2 \Phi_v(\omega)]_{jj}} = \sigma_{ij}^2(\omega) \quad if\ N_1, N_2 \to \infty \qquad [3.3]$$

Since the theory of paragraph 3.2 is only valid for linear systems we might expect some differences between theory and practice. But in practice we often have nonlinear systems and we hope to avoid problems by identifying the process around a specific operating point and using small input signals.

3.4.2 Method two.

Another way of increasing the input power spectrum $\Phi_u(\omega)$ in the variance term of [3.1] is to simply add the data sets.
With the data sets (scaled around zero) $u_N(t)$, $v_N(t)$, $y_N(t)$ and $z_N(t)$ we create:

$input_N(t) = u_N(t) + \alpha\, v_N(t)$ (α is weighing factor)

$output_N(t) = y_N(t) + \alpha\, z_N(t)$

Adding the output sets can only be done if the process is linear!

From this follows [6]:

$$E\{input(t)\ input(t-\tau)\} = E\{u(t)\ u(t-\tau)\} + E\{v(t)\ v(t-\tau)\} + E\{u(t)\ v(t-\tau)\} + E\{v(t)\ u(t-\tau)\}$$

Here also crossterms appear so only if N1 and N2 go to infinity and in case $u_N(t)$ and $v_N(t)$ are uncorrelated we can state that:

$\Phi_{input}(\omega) = \Phi_u(\omega) + \Phi_v(\omega)$ if $N1, N2 \rightarrow \infty$

We can use this last expression in our variance term but N is not clearly defined when N1 is not equal to N2.
We will concentrate on the first method further on.

3.5 Variance reduction factor.

When adding a second data set v_{N2} to a data set u_{N1} as described in 3.4.1 we can introduce a weighing factor to dataset v_{N2}. This weighing factor tells us how strong v_{N2} is used in identification.
We now add (in case of a single-input-process):

$u_{N1}(t) = [u(1) .... u(N1)]$ (N1x1 vector)
$v_{N2}(t) = [v(1) .... v(N2)]$ (N2x1 vector)

Suppose that u and v are normed to 1; this means that the original vectors have been divided by their variance σ^2.
This gives the normed vectors:

$u_{n,N1}(t) = [u(1) .... u(N1)]/\sigma_u^2$ (N1x1 vector)
$v_{n,N2}(t) = [v(1) .... v(N2)]/\sigma_v^2$ (N2x1 vector)

Now a weighing factor α can be introduced to dataset $v_{n,N2}$.
This leads to the combined vector $w_{N1+N2}(t)$:

$$w_{N1+N2}(t) = [u_{n,N1}(t)\ \alpha \cdot v_{n,N2}(t)] = [u(1) .... u(N1)\ \alpha \cdot v(1) .... \alpha \cdot v(N2)]$$

Now the combined covariance function yields:

$$R_w(\tau) = \frac{N_1}{N_1+N_2} R_u(\tau) + \frac{\alpha^2 N_2}{N_1+N_2} R_v(\tau) + Residue$$

In case of uncorrelated data sets the residue will be 0. This leads with equation [3.1] to the new variance:

$$var\ \hat{g}^n_{ij}(\exp(i\omega)) = \frac{n\ \Phi_{vi}(\omega)}{[N_1 \Phi_u(\omega) + \alpha^2 N_2 \Phi_v(\omega)]_{jj}} = \sigma^2_{new}(\omega) \quad if\ N_1, N_2 \to \infty$$

Together with the old variance:

$$\sigma^2_{old} = \frac{n\ \Phi_{vi}}{N_1\ \Phi_u}$$

We now obtain a variance reduction factor (VRF):

$$\frac{\sigma^2_{new}}{\sigma^2_{old}} = \frac{N_1 \Phi_u}{N_1 \Phi_u + \alpha^2 N_2 \Phi_v} = \frac{1}{1 + \alpha^2 \frac{N_2}{N_1} \frac{\Phi_u}{\Phi_v}} \qquad with\ \frac{\Phi_u}{\Phi_v} = 1$$

$$\frac{\sigma^2_{new}}{\sigma^2_{old}} = \frac{1}{1 + \alpha^2 \frac{N_2}{N_1}} = \textit{variance reduction factor (VRF)}$$

3.6 Estimation from more than one data set: MDSPE_2.

When a fast PRBNS experiment and a freerun experiment (all inputs constant) have been performed information about the process bandwidth and the output disturbance spectra is present. In general the disturbance spectra are low-pass. Also the process spectrum is low-pass (see figure 3.2).

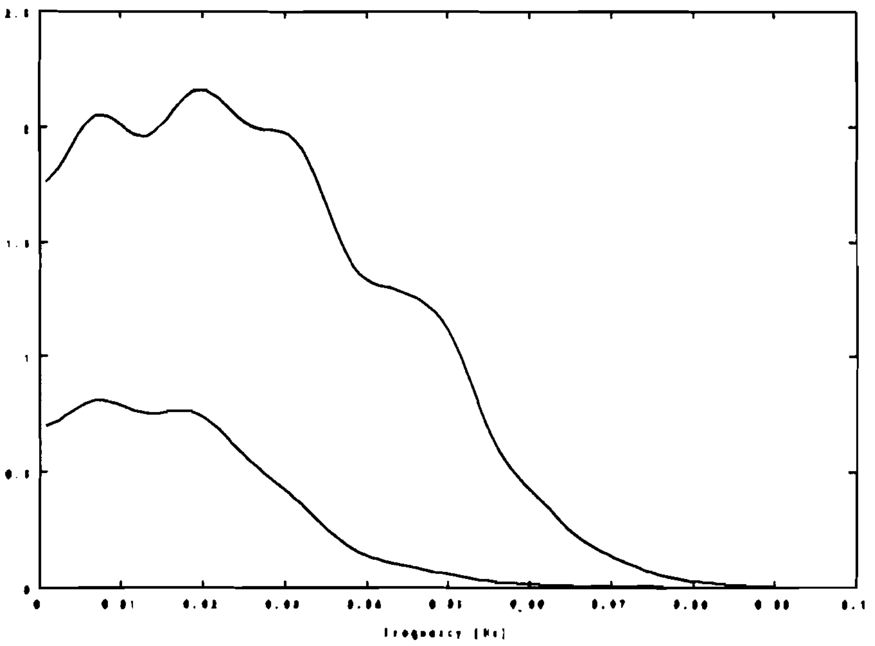


figure 3.2: *example of an output noise spectrum and a process spectrum.*

However, a (final) PRBNS input signal has a white noise spectrum, this means the spectrum is flat for all frequencies. To reduce the modelling error at higher frequencies the fast PRBNS data set can be used in identification according to experiment MDSPE_1 (section 3.4). This gives the controller more robustness at higher frequencies because of a smaller modelling error.
In order to improve the disturbance rejection which is mainly low frequent here, model accuracy for the interesting low frequency part of the spectrum must be improved. The process controller will mainly give steering signals in the low frequency part of the spectrum because the disturbances act on the process in just that frequency part. The better the model is estimated for low frequencies the better the disturbance rejection will be.

Besides the model errors in the frequency domain are high at low frequencies when performing a standard (white) PRBNS experiment (see results; chapter 5).
I therefore suggest to perform a new experiment (called MDSPE_2) where the input signal is chosen such that for high frequencies (higher than the process bandwidth) the input spectrum is lower than the PRBNS spectrum (not equal zero to maintain numerical stability) and for the plant bandwidth the spectrum lies much higher than the PRBNS spectrum. Such an input signal can be designed by filtering white noise (low pass filter with bandwidth equal to plant bandwidth) and adding white noise to the filtered noise afterwards.
In case of the spray dryer process (sample frequency 0.2 Hz) the PRBNS spectrum runs up to 0.1 Hz. The plant bandwidth however is about 0.01 Hz and even so is the noise bandwidth (why the sample frequency cannot be reduced is explained in chapter 5). When designing input signals according to the method above for low frequencies the new input signal spectrum about 8 times (= 18dB) higher than with normal PRBNS input signals. This is done when keeping the total signal power constant.

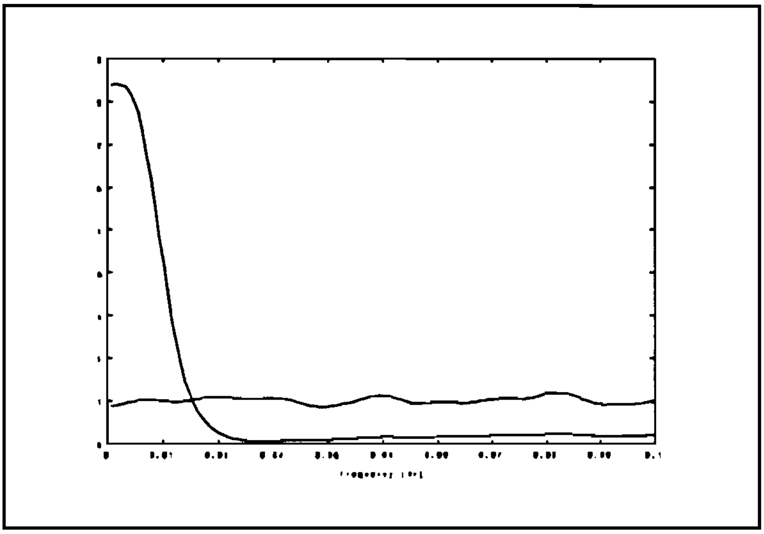


figure 3.3: *spectra of a white PRBNS signal and a coloured noise input signal with equal signal power.*

When an experiment is done with these input signals a model can be estimated with this new data set. Also this data set will be coupled to the final PRBNS data set and a model from that large data set will be estimated. Another possibility is to create a data set immediately with the first part white PRBNS input signals and the second part low pass filtered white PRBNS input signals. Here two independent data sets are coupled (that's the worst case) because in general one needs the results of the previously measured data.
Model errors in time and frequency domain will be verified for both models in section 6.3. The spectrum of the steering signal of the controller can be used afterwards to create a new input signal for the process. This data set may than be used in a comparable new identification step as the MDSPE_2 experiment. Because this will merely give the same results only the described MDSPE_2 experiment will be performed.

3.7 Estimation from more than one data set: MDSPE_3.

When having performed the experiments of section 3.6 (experiment name MDSPE_2) we obtain two models:

1. a model from the final PRBNS experiment
2. a model from the coloured noise experiment (MDSPE_2)

Both models are high order state space models for which the theory of section 3.2 is valid. So for both models we can give a two or three sigma upperbound for the model error in the frequency domain according to section 3.2. In experiment MDSPE_2 of section 3.7 one tries to reduce the model error by coupling both generated data sets and estimating a new (improved) model from this coupled data set.
Another way of estimating a model which fits well on both already estimated models is by generating new data (which can have nearly infinite length) with both models and their model errors in the frequency domain and estimating a new (improved) model (called MDSPE_3) on this data. This idea is illustrated in figure 3.4.

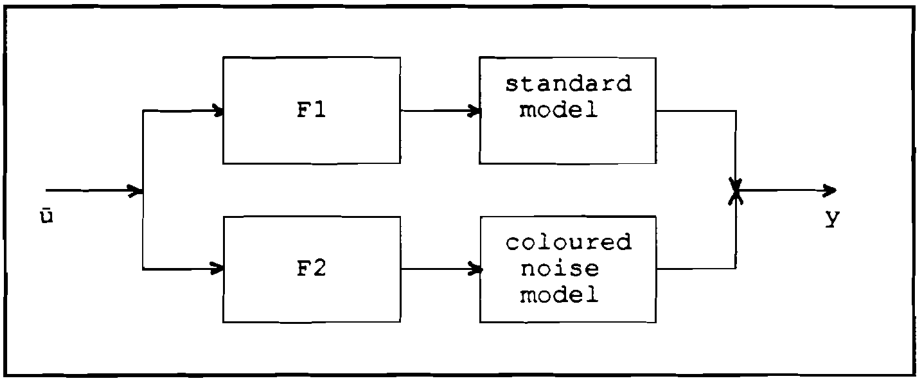


figure 3.4: *the MDSPE_3 experiment.*

In figure 3.4 the filters F1 and F2 can be designed with the 3-sigma model upperbounds. Each filter thus has four components (f11 f12 f21 f22) in case of a 2x2 MIMO model. For simplicity (and based on the results of the upperbounds) all filters f11 f12 f21 f22 are the same. That means we only deal with two different filters F1 and F2.
The results of the upperbound for the modelling errors show that the standard model performs well on high frequencies, so filter F1 will be high pass. On the other hand the coloured noise model performs better for low frequencies. Therefore the filter F2 will be chosen low pass.
Both filter shapes are plotted in figure 3.5. The sum of the given filter amplitudes is nearly one for all frequencies.
The input signals are PRBNS signals (white noise). The output signals can be added in case of the appropriate filter amplitudes (in case of only one filter pro model filtering can also be performed on the output signals with linear models; the advantage is then that the DC term disappears after filtering with F1).
Both models are high order state space models of which the signals are scaled back to their original physical values.

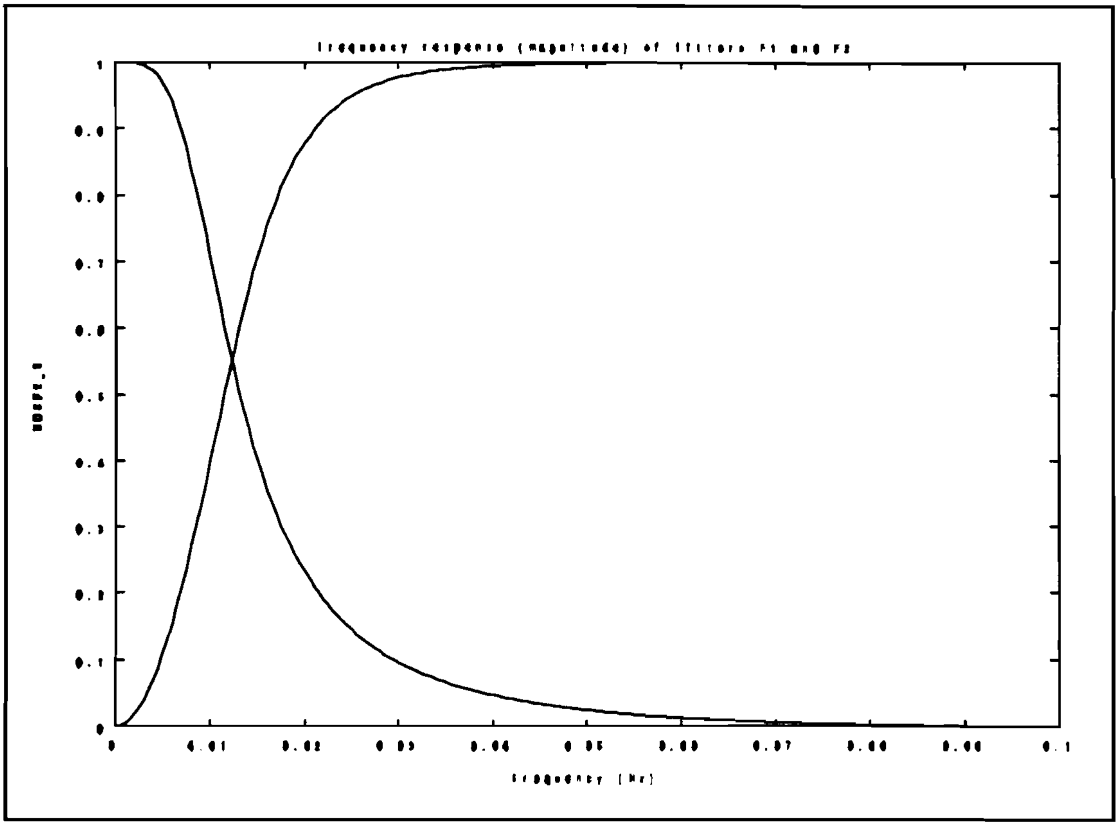


figure 3.5: *the filters F1 and F2.*

3.8 ARX model estimation.

In case of the spray dryer tower modelling the transfer of input u1 to both outputs has a time constant which is 10 times the time constant of input u2 to both outputs. A drawback of using the standard estimation procedure of Backx [1] is the enormous amount of FIR parameters necessary to estimate the second transfer function while still estimating the first transfer function with a normal amount of parameters. Therefore an ARX model will be estimated for this process.
The advantage of using ARX models is that this model has infinite impulse response and so a long time can be described with less than the FIR parameters.

The disadvantage of using ARX models is:

- equation error minimization in stead of output error minimization (one step ahead prediction)
- MISO model estimation; multiplicity of poles estimated in the MISO models.

One hopes to reduce the multiplicity of poles in de MISO models by model reduction on the MIMO model which is made out of the MISO models.

The ARX model structure is given by:

$$A(q)\ y(t) = B(q)\ u(t) + e(t)$$

Note that FIR model estimation is also MISO!: $A(q) = I$.
in which:

$$A(q) = \mathrm{diag}[A_{11}(q)\ A_{22}(q)]$$

$$B(q) = \begin{vmatrix} B_{11}(q) & B_{12}(q) \\ B_{21}(q) & B_{22}(q) \end{vmatrix}$$

$A_{..}(q)$ and $B_{..}(q)$ are polynomials of order n and n+1 respectively (n+1 is chosen for the transfer to state space representation; see further on)

$$A_{11}(q) = 1 + a_{11}q^{-1} + .. + a_{1n}q^{-n}$$

$$B_{11}(q) = b_{11} + b_{12}q^{-1} + ... + b_{1,n+1}q^{-n-1}$$

$$y(t) = [y_1(t)\ y_2(t)]^T$$

$$u(t) = [u_1(t)\ u_2(t)]^T$$

$$e(t) = [e_1(t)\ e_2(t)]^T$$

Note that an ARX model consists of two MISO systems for this 2x2 multivariable process.

Now this ARX model must be transformed into a MIMO state space model.This can be done with the observer canonical form (state space representation) for each MISO subsystem.
For MISO the state space observer canonical form is given as:

$$y(t) = b_{1n}u_1(t-n)+ .. +b_{11}u_1(t-1) + b_{2n}u_2(t-n)+ .. +b_{21}u_2(t-1) - a_1y(t-1)+ .. +a_ny(t-n) + e(t)$$

or:

$$y(t)\ (1+a_1z^{-1}+ .. +a_nz^{-n}) = u(t)\ (b_{11}z^{-1}+ .. +b_{1n}z^{-n} + b_{21}z^{-1}+ .. +b_{2n}z^{-n}) + e(t)$$

For the state space expression we have:

$$A = \begin{vmatrix} -a_1 & 1 & 0 & 0 & . \\ . & 0 & 1 & 0 & . \\ . & . & . & . & . \\ -a_n & 0 & . & 0 & 1 \end{vmatrix}$$

$$B = \begin{vmatrix} b_{11} & b_{21} \\ . & . \\ . & . \\ b_{1,n+1} & b_{2,n+1} \end{vmatrix}$$

$$C = [1\ 0\ ...\ 0]$$

$$D = [d_1\ d_2]$$

From the ARX estimation the parameters of the state space representation can be derived:

$[a_1\ ...\ a_n] = A_{11}(q)$

$[b_{11}\ ...\ b_{1,n}] = B_{11}(q)$ (parameter 2 .. n+1 !!!)

$[b_{21}\ ...\ b_{2,n}] = B_{21}(q)$ (parameter 2 .. n+1 !!!)

$[d_1\ d_2] = [B_{11}$(parameter 1) B_{21}(parameter 1)]

With this method we create a state space representation for both MISO models (in case of a 2x2 MIMO process). Now we merge both models into one MIMO state space model of order 2n:

$$A = \begin{vmatrix} A1 & 0 \\ 0 & A2 \end{vmatrix}$$

$$B = \begin{vmatrix} B1 \\ B2 \end{vmatrix}$$

$$C = \begin{vmatrix} C1 & 0 \\ 0 & C2 \end{vmatrix}$$

$$D = \begin{vmatrix} D1 \\ D2 \end{vmatrix}$$

This high order state space model can now be reduced in order.
This method will be tried out at the main goal: combining the final PRBNS, the fast PRBNS and the stepresponse data sets. This method is added because standard identification according to [1] fails on this data set (as results will show).

4. The spray dryer process.

4.1 Introduction.

The spray dryer process is widely used in food and chemical industry to remove liquids, mainly water, from solutions and suspensions. Applications in food industry products: e.g. milk products, fruit and vegetable extracts. We will take a closer look at a milk spray dryer tower.
The installation usually consists of a cylindrically shaped chamber with a conical bottom, see fig. 4.1. Feed is sprayed from the top of the tower through a rotary or pressure nozzle atomizer. Also heated air enters from the top in such a way that a rapid and efficient mixing between feed and air takes place. The dried powder and air leave the tower separately at the bottom.

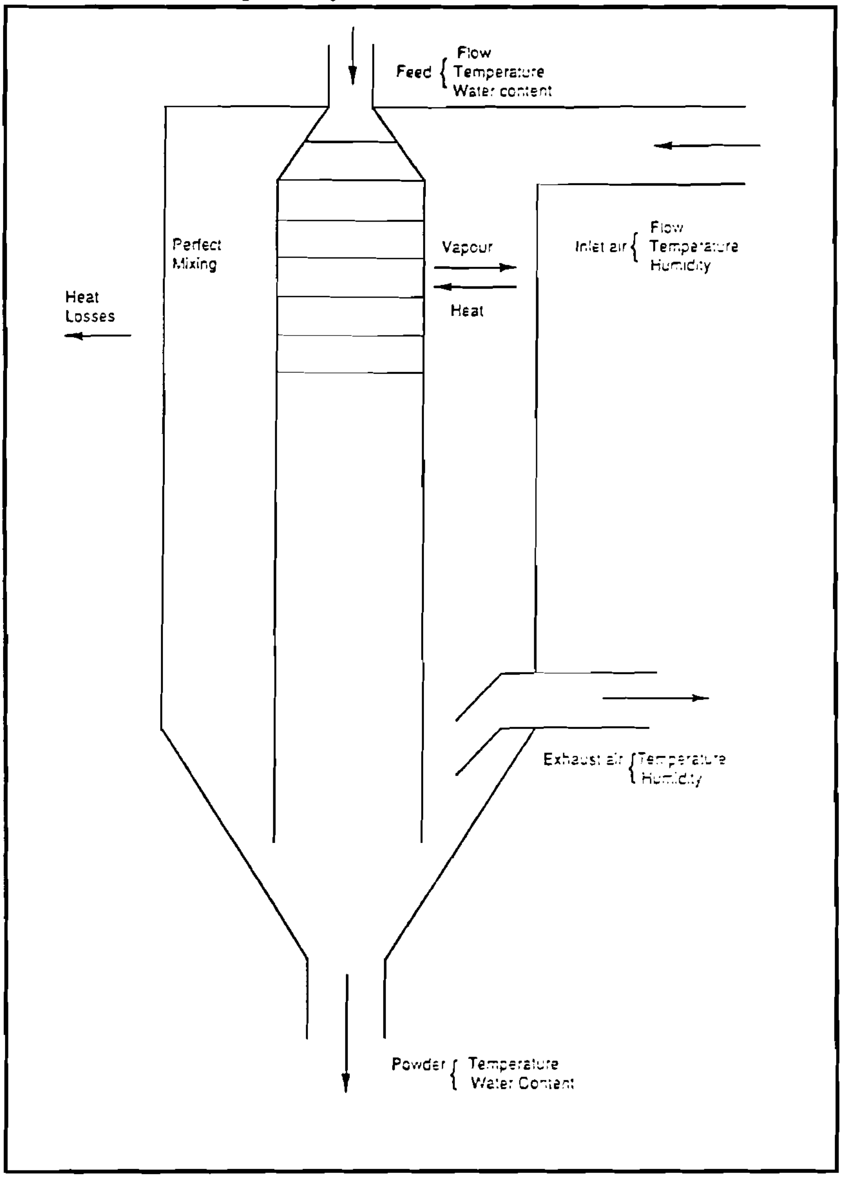


Figure 4.1: *lay out of a spray dryer tower [7].*

Due to the strong interaction between inputs and outputs and the presence of transport delays that are long compared to the system time constants, the performance of classical control systems is very limited. Modern multivariable control performs much better on such processes. Those modern control design methods rely on an accurate model of the process dynamics in state space representation. System identification can provide such a model from measured input and output signals of the real process. With this process the proposed Multi Data Set Parameter Estimation (MDSPE) will be used for identification.
To demonstrate these techniques, a complex simulation model of a milk spray dryer based on physical laws was derived in which all relevant characteristics of a real spray dryer are incorporated. This model, consisting of several nonlinear equations, was implemented on a microVax computer and can be used to generate data.

4.2 Modelling the spray dryer process.

The identification procedure developed by Backx [1,8] was used for identification of the spray dryer process.
The spray dryer process has 7 inputs and 5 outputs, which all can be simulated on the computer.

The inputs are:

u1: airflow
u2: watercontents of air
u3: setpoint for air inlet temperature
u4: environment temperature
u5: feed flow of product
u6: watercontents of feed
u7: temperature of feed

The outputs are:

y1: watercontents of powder
y2: powder temperature
y3: watercontents of air
y4: exhaust air temperature
y5: walltemperature

Inputs and outputs used by a controller are:

inputs: u1: air inlet temperature
u2: feed flow of product

These inputs were chosen for physical reasons: when varying these inputs the residence time of the milk powder remains constant.

outputs: y1: watercontent of powder
y2: exhaust air temperature

These outputs are chosen because:

- y1 must be held constant
- y2 is a direct measure for the maximum powder temperature and is more easy to measure than the powder temperature.

A control objective is to keep the powder temperature below 70 degrees. Because the exhaust air temperature is about the same as the powder temperature but is much easier to measure it is allowed to take the exhaust air temperature as process output.

Beforehand we have the following knowledge about this process:

- inputs and outputs are nonlinear related; this means that we cannot use very large input signals; we also have to estimate a model for every process setpoint.
- strong static couplings exist between input / output transfers.
- both outputs react fast on changes in input u2 (feed flow); time constants about 50 seconds.
- both outputs respond slow on changes in input u1 (air inlet temperature setpoint), due to the large time constant of about 500 seconds of the heater.
- delay times are long; the air temperature is measured 10 seconds after leaving the dryer and the powder moisture concentration after 25 seconds; the heater delay time is about 50 seconds; delays cannot be shortened.
- sanitary reasons give that a maximum percentage of water content is allowed in milk powder; when going under this maximum energy (=money) is waisted (efficiency).
- the temperature is critical as certain proteins in milk denature at about 70°C; the highest temperature is reached at the end of its residence time as the evaporation rate reduces; therefore the air flow is parallel to the feed flow.
- the temperature of the feed must be held as high as possible because the drying to the air then is at most and this will give the highest throughput.
- disturbances on the process are: the water content of the input air flow, the water content of the input feed flow and the temperature of the feed.

Identification can now be performed using the identification scheme of section 2, which is implemented in the programm IPCOS that runs under MATLAB on a 386-AT Personal Computer.

4.3 Operating points of the milk spray dryer.

The milk spray dryer will be identified in the following normal operating point.

Input:	Quantity:	
u1:	20 m^3/s	(airflow)
u2:	0.01 kg water/kg air = mass%	(watercontents of air)
u3:	200°C	(air temperature)
u4:	25°C	(environment temperature)
u5:	0.001 m^3/s	(feed flow of product)
u6:	65%	(watercontents of feed)
u7:	60°C	(temperature of feed)

These input values dry the feed to about 8% watercontent.

4.4 Modelling input disturbances on the milk spray dryer.

There are three kinds of input disturbances which will be added when identifying the process. These disturbances are in practice not measured. The disturbances are:

1. watercontent of air is not a constant. When using heated air from outside the watercontents may vary with the weather condition. The watercontents of outside air is expected to vary between 0 and 0.01 mass% (very wet). Changes are normally not within one hour. Therefore we simulate this type of disturbance by filtering white noise with a first order filter with a time constant of one hour. We put this disturbance as input to input u2. The amplitude is between 0 and 0.01%. We use a first order filter because then higher frequencies are not cleared completely.
2. watercontent of feed is not a constant. Normally feed is sprayed into the tower out of a buffer tank. This tank is filled now and then so the watercontent of the feed that comes out will not be constant. When making the tank large enough we suppress this disturbance. With our feed flow of 0.001m^3/s a normal (chosen) tank of 3.6m^3 will be empty in one hour. A buffer tank is typically a first order process so we can simulate this disturbance by filtering white noise with a first order filter with a time constant of one hour. We add this disturbance to our watercontent_feed setpoint (u6) of 65%. An often used value in articles for the amplitude of the disturbance part is +/- 2%.
3. temperature of feed is not a constant. The temperature of the feed sprayed into the tower will also not be constant due to the influence of the buffer tank. Variations of +/- 5% of the value of 60°C are typical. This disturbance is generated as white noise filtered by a first order filter with a time constant of one hour (see point 2) and is added to the temperature feed setpoint (input u7).

4.5 simulating the spray dryer process.

The milk spray dryer tower is simulated on a microVax computer under MatrixX.
Schematically the spray dryer process is installed as in figure 4.2.
The numbers of the inputs and outputs correspond with those in section 4.2.
The delay times are given in table 4.1.

Delays [sec]		
	Output y1	Output y4
Input u3	75	60
Input u5	25	10

Table 4.1.: *Delays (real)*

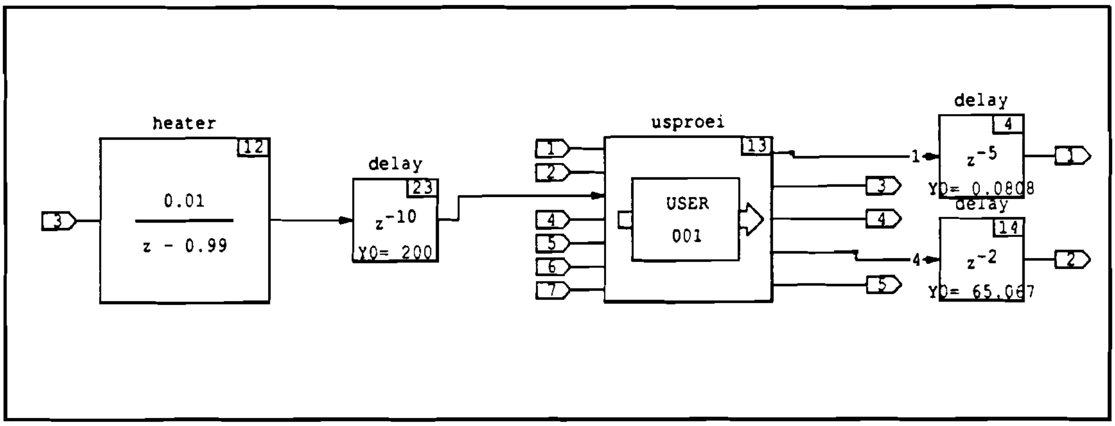


figure 4.2: *the milk spray dryer process in MatrixX.*

5. Standard identification of the spraydryer.

5.1 introduction.

In this chapter the standard identification procedure according to section 2.2 will be used for identification of the spraydryer process. At first a model will be estimated from noise-free data sets that are generated. This will lead to a so called noise-free model (named 'longzon' in the plots) which we will assume to resemble the actual process very closely. We will use this high order state space model for comparison with the latter obtained models which were estimated with the new identification techniques.
These models are estimated from data sets that are corrupted with noise so they will give a practical impression of the performance of the identification techniques presented here.
The sample time is 5 seconds in each experiment except for the fast PRBNS experiment in which the sampling time is 1 second. The reason for these values is given in section 5.4.

5.2 Staircase experiment.

To check the linearity of the outputs of the process, we put a staircase input signal on the inputs. First on input u1, then on input u2. From the responses we can check the static linearity. Inputs and responses are given in figure 5.1a-5.1h.
From these experiments a significant nonlinerarity showed up in both outputs when varying input u1 more than 5°C. Also for input u2 when varying this input more than 0.00005 m^3/s. Delays, time constants and gains can be estimated from this experiment according to table 5.1 to 5.3. Note that these experiments have been performed without noise. Step experiments have also been performed with additional noise and are given in figure 5.1i and 5.1j. Note that from the step responses with noise time delays cannot be estimated. Therefore in all model estimations time delays will be estimated by the model itself (it's better to underestimate the time delays than to overestimate them because than the model is not causal).

Delays [sec]		
	Output y1	Output y2
Input u1	75	60
Input u2	25	10

table 5.1.: *delays (of staircase).*

Static gains		
	Output y1	Output
Input u1	-6.7 E-4	0.31
Input u2	320	-3.0 E+4

table 5.2.: *static gains (of staircase).*

Largest time constants [sec]		
	Output y1	Output y2
Input u1	< 500	< 500
Input u2	< 100	< 100

table 5.3.: *largest time constants (of staircase).*

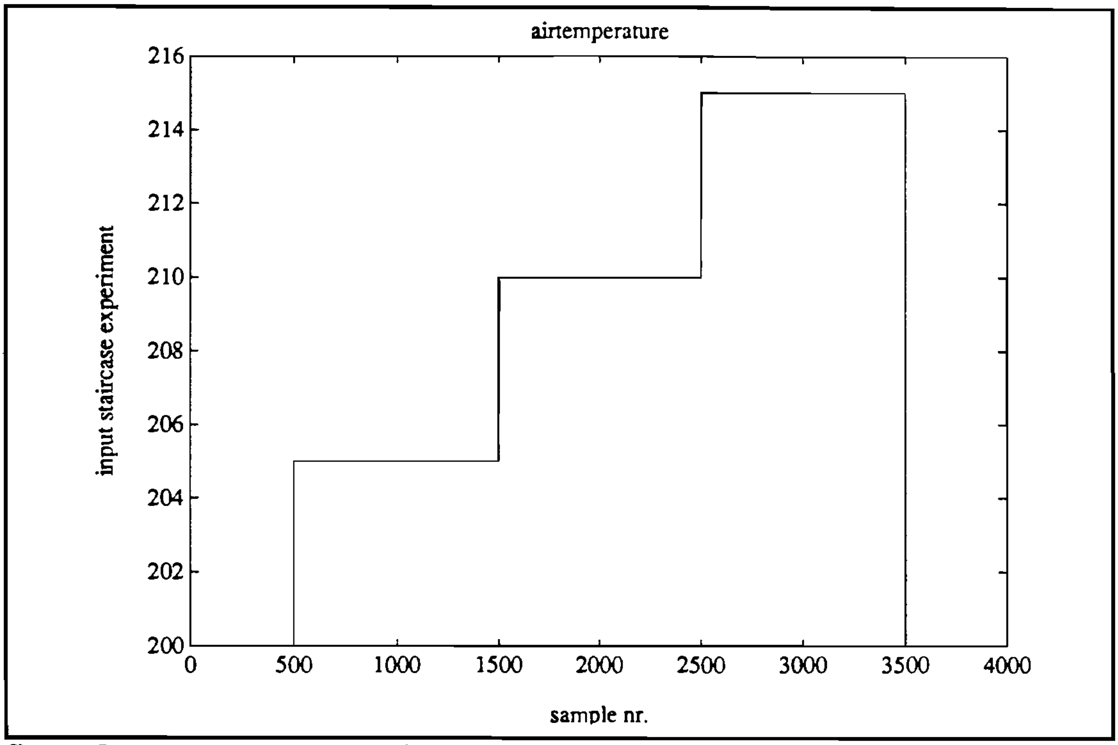


figure 5.1a.: *staircase input u1.*

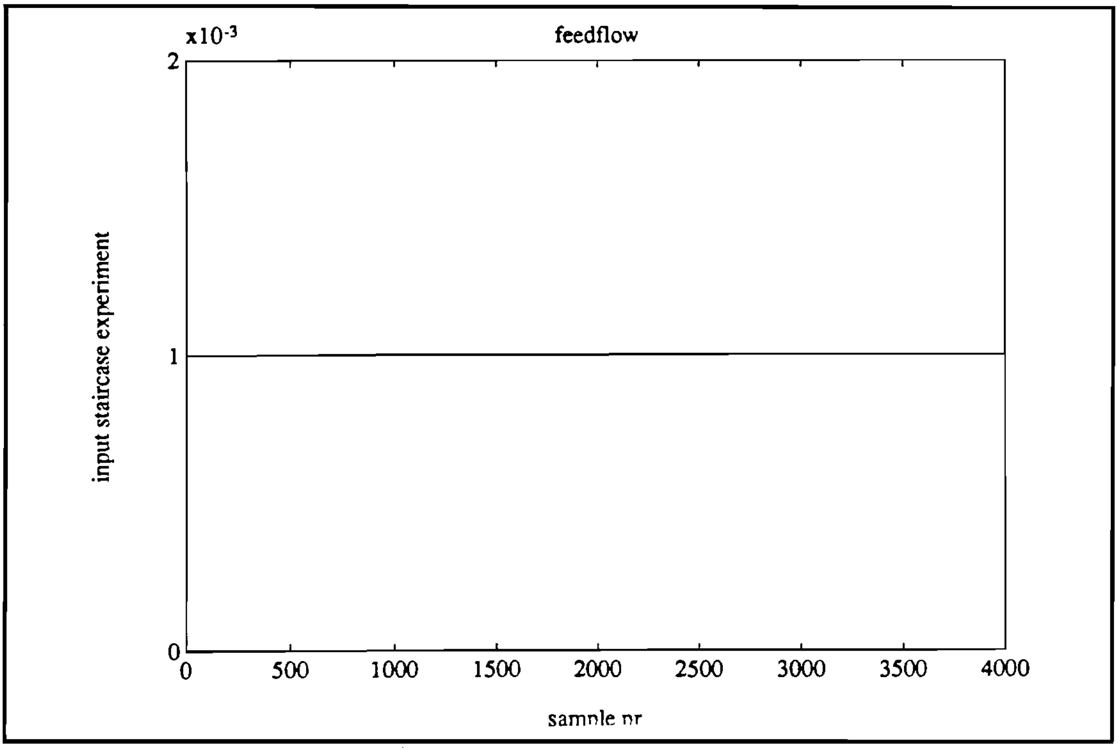


figure 5.1b.: *staircase input u2.*

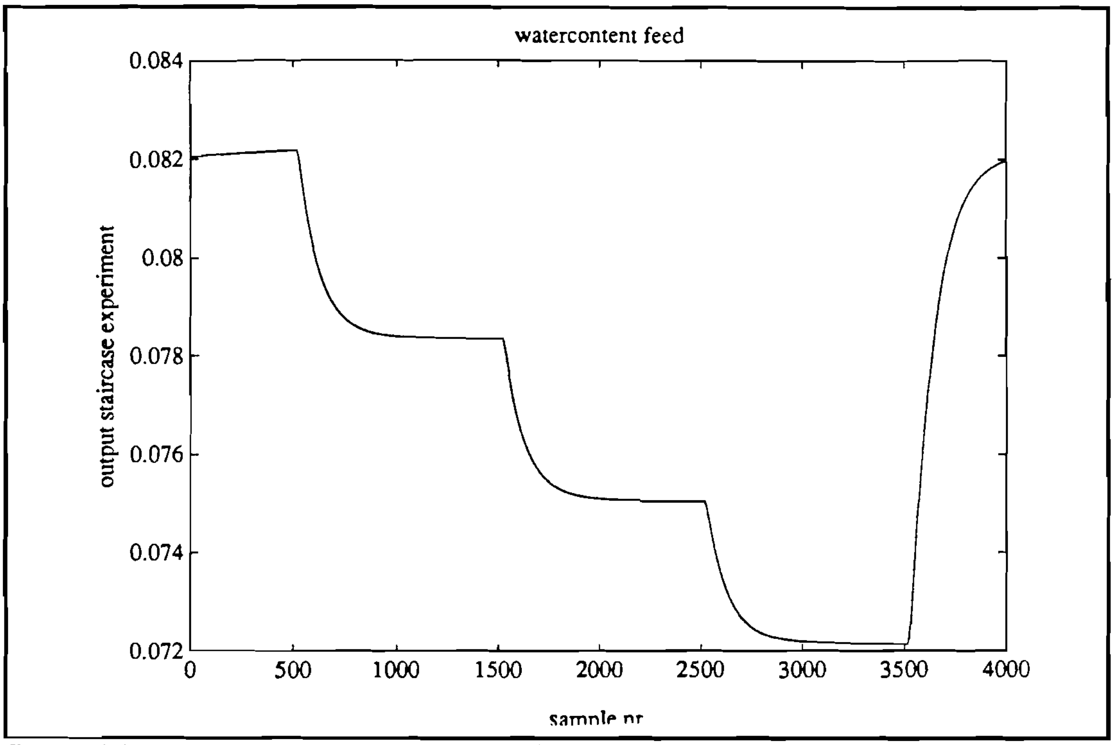


figure 5.1c.: *staircase response output y1.*

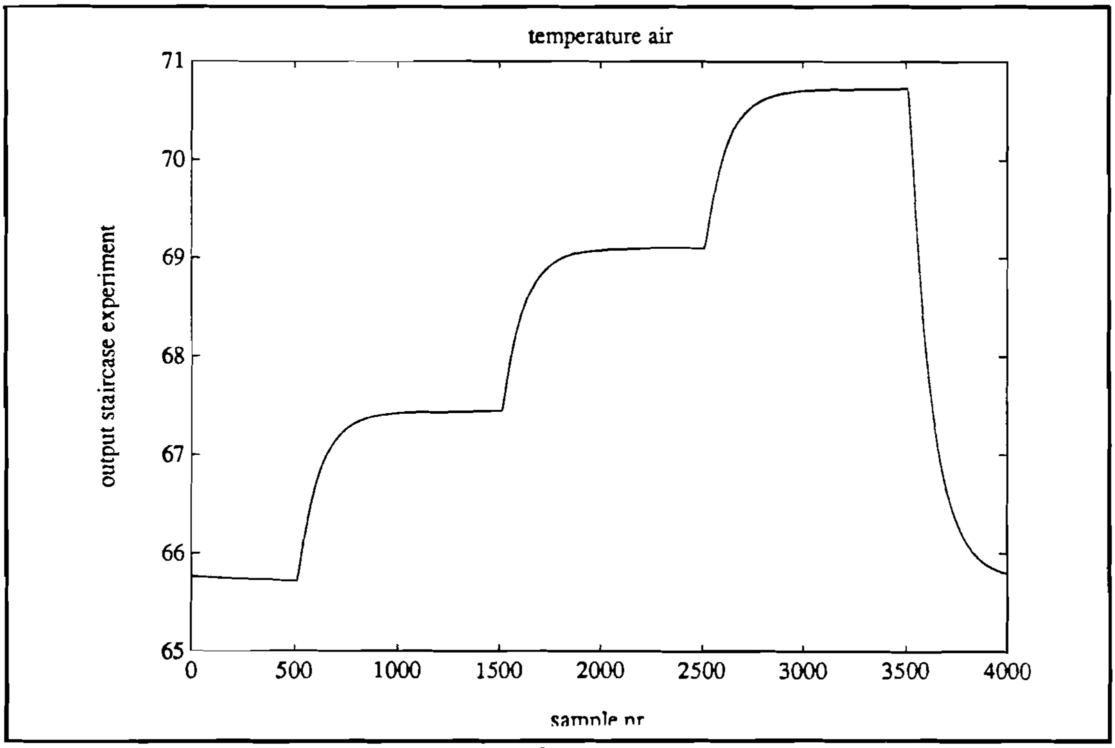


figure 5.1d.: *staircase response output y2.*

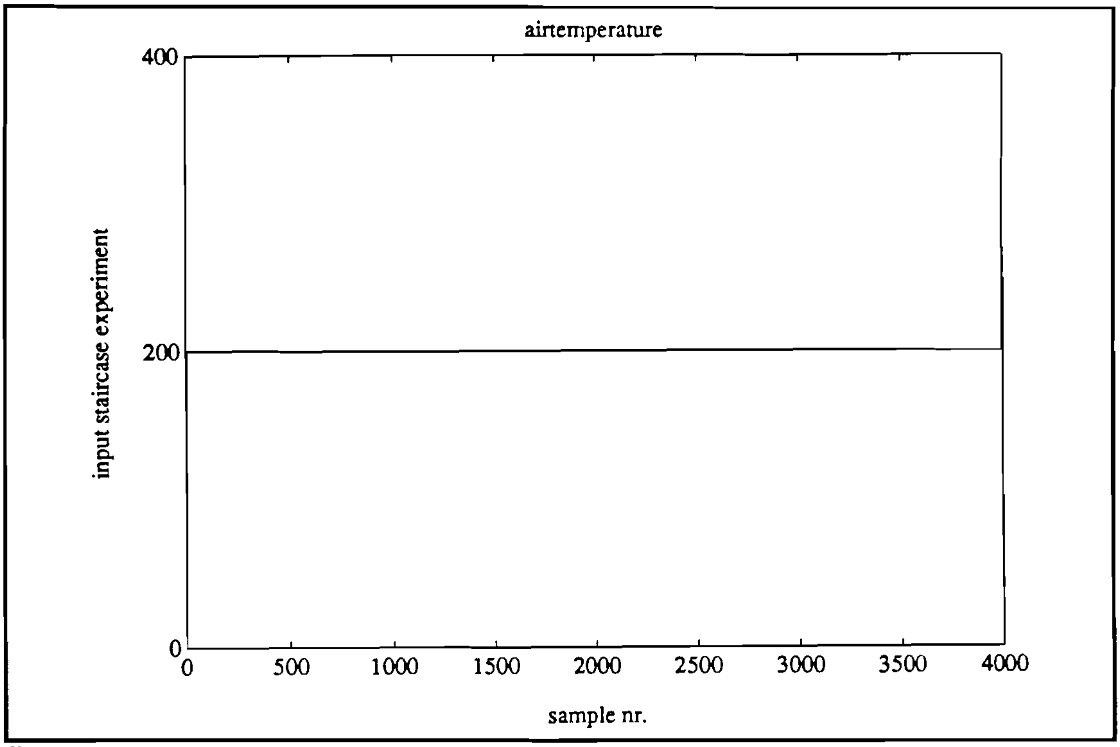


figure 5.1e.: *staircase input u1.*

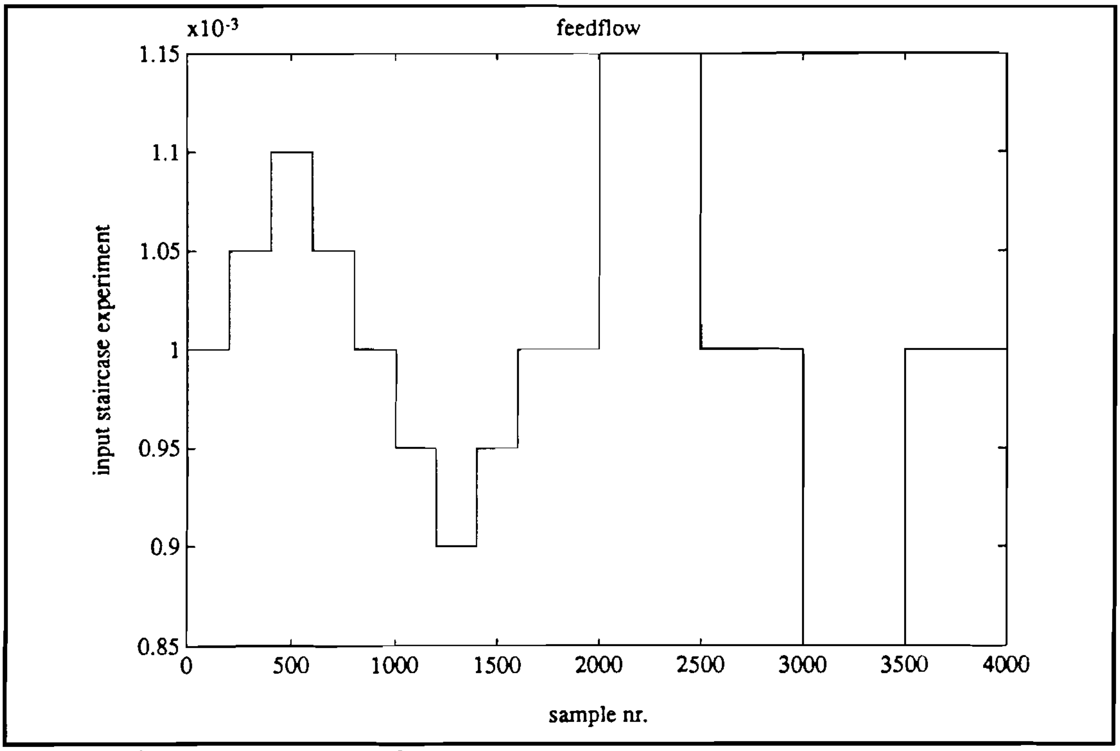


figure 5.1f.: *staircase input u2.*

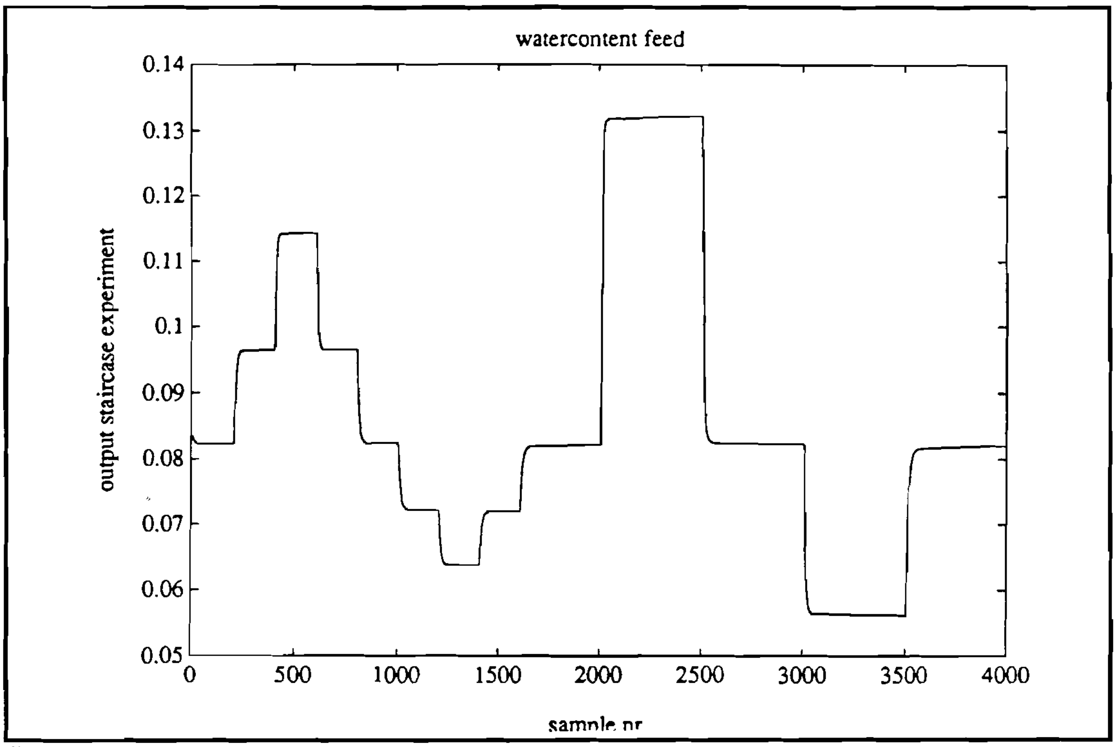


figure 5.1g.: *staircase response output y1.*

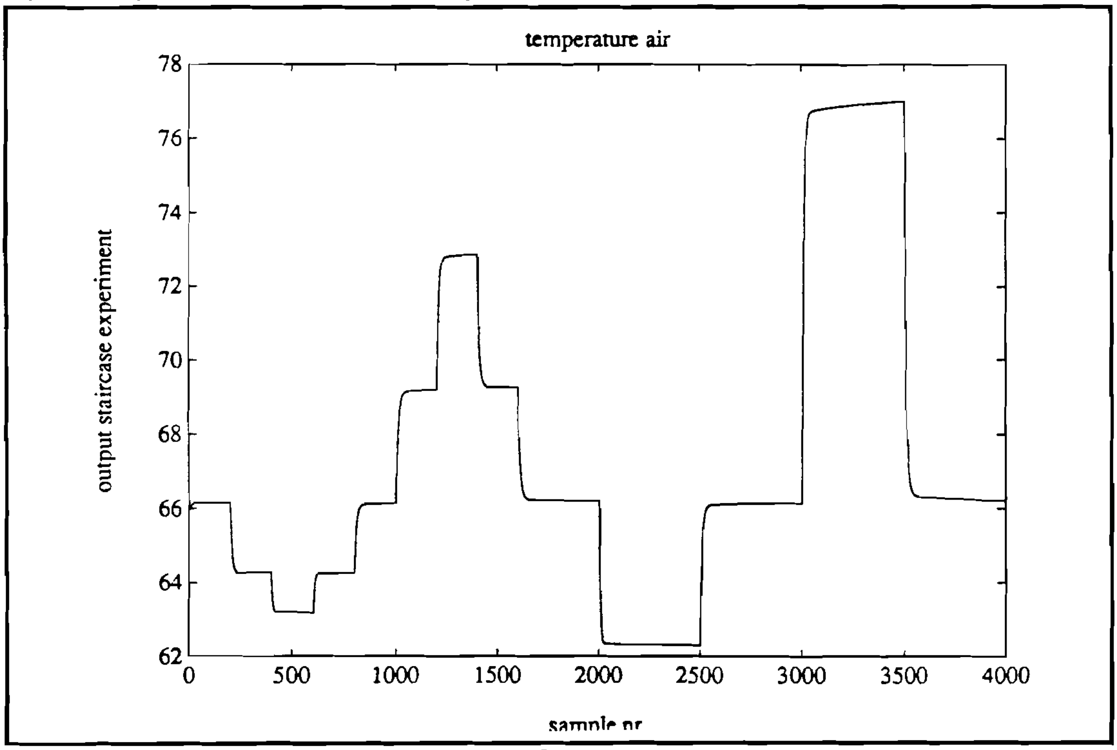


figure 5.1h.: *staircase response output y2.*

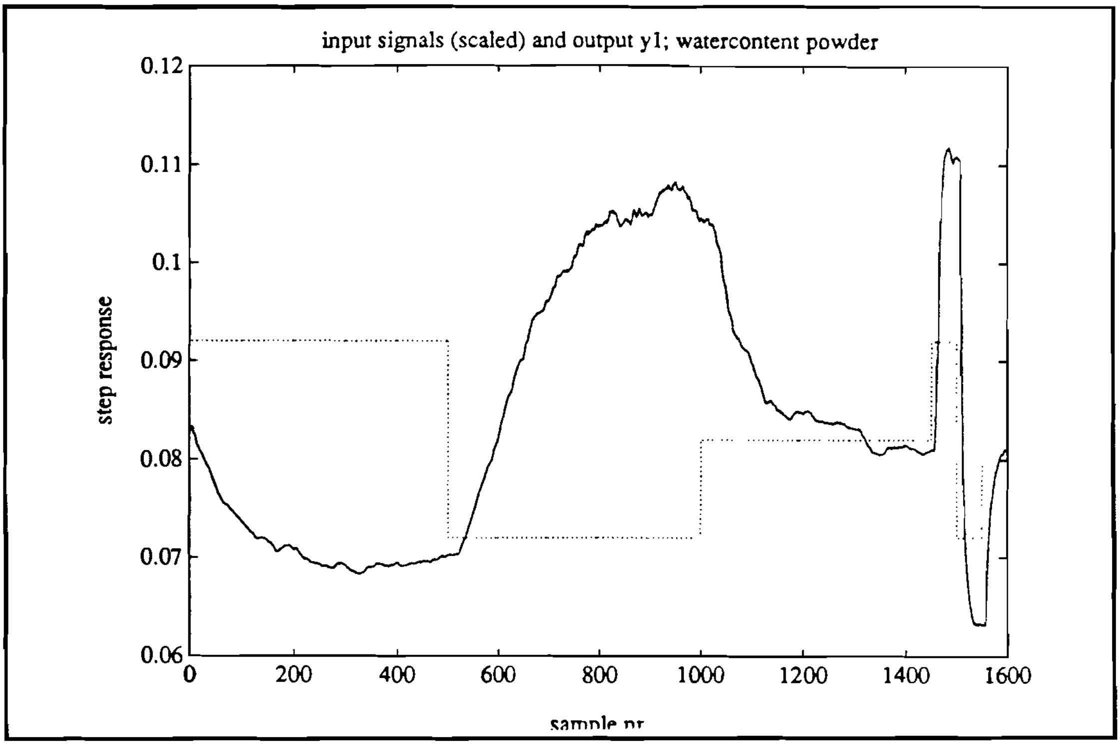


figure 5.1i.: *step response process output y1 with additional noise.*

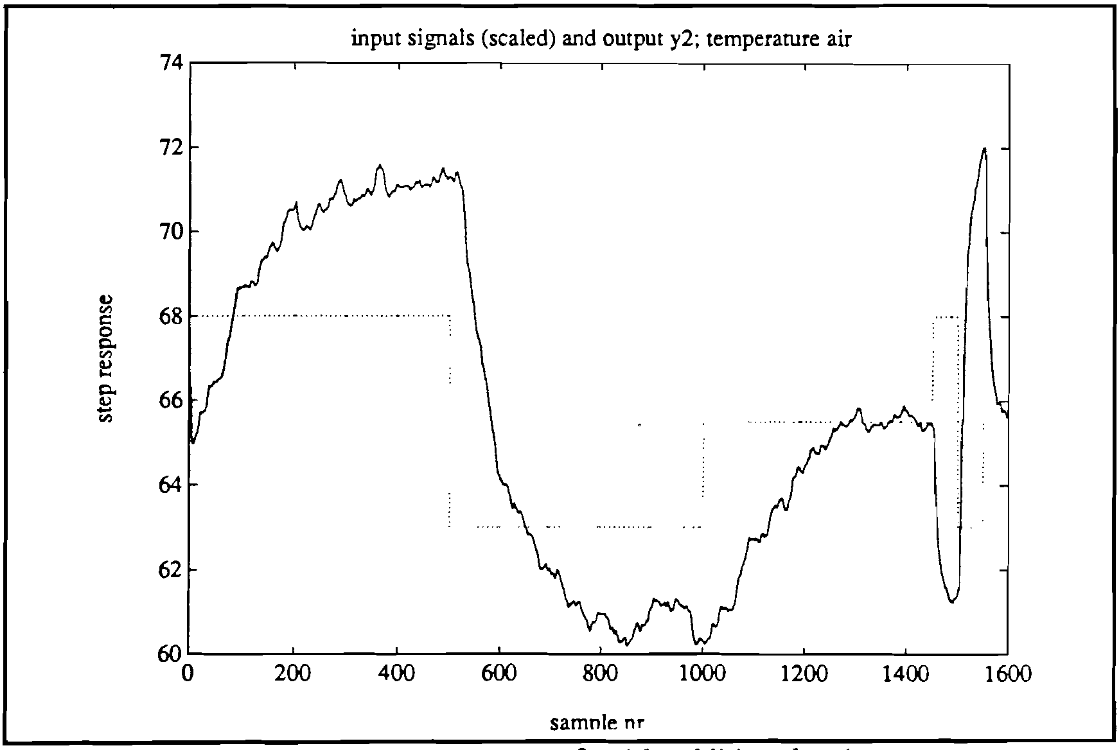


figure 5.1j.: *step response process output y2 with additional noise.*

5.3 Free run experiment.

A free run experiment is an experiment in which both manipulable inputs are kept constant. The outputs will then give an indication of the output levels in steady state and the dynamics of the disturbances. Outputs are given in figure 5.2a and 5.2b.
They give the following result:
mean value of output y1: 8.2%
peak-amplitude of noise on y1: 0.2%
mean value of output y2: 65.5°C
peak-amplitude of noise on y2: 1°C
amplitudes of the spectra have only relative meaning (not absolute)
For spectra of the outputs see figure 5.2c and 5.2d.

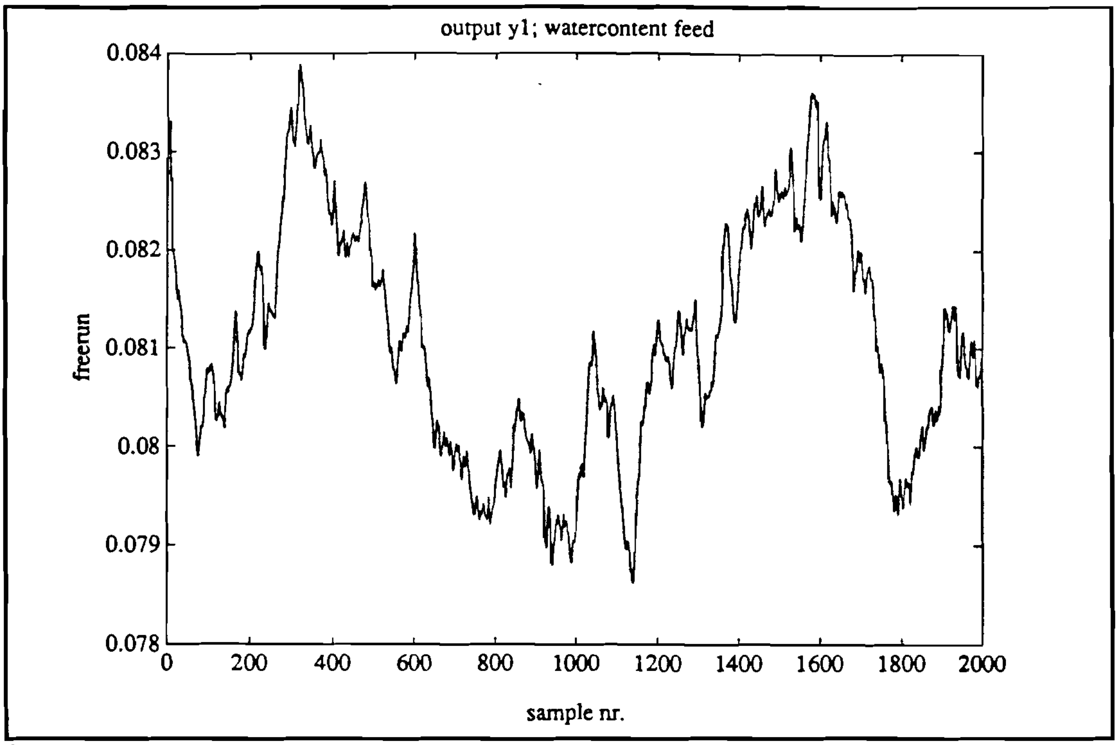


figure 5.2a.: *output y1 in free run experiment.*

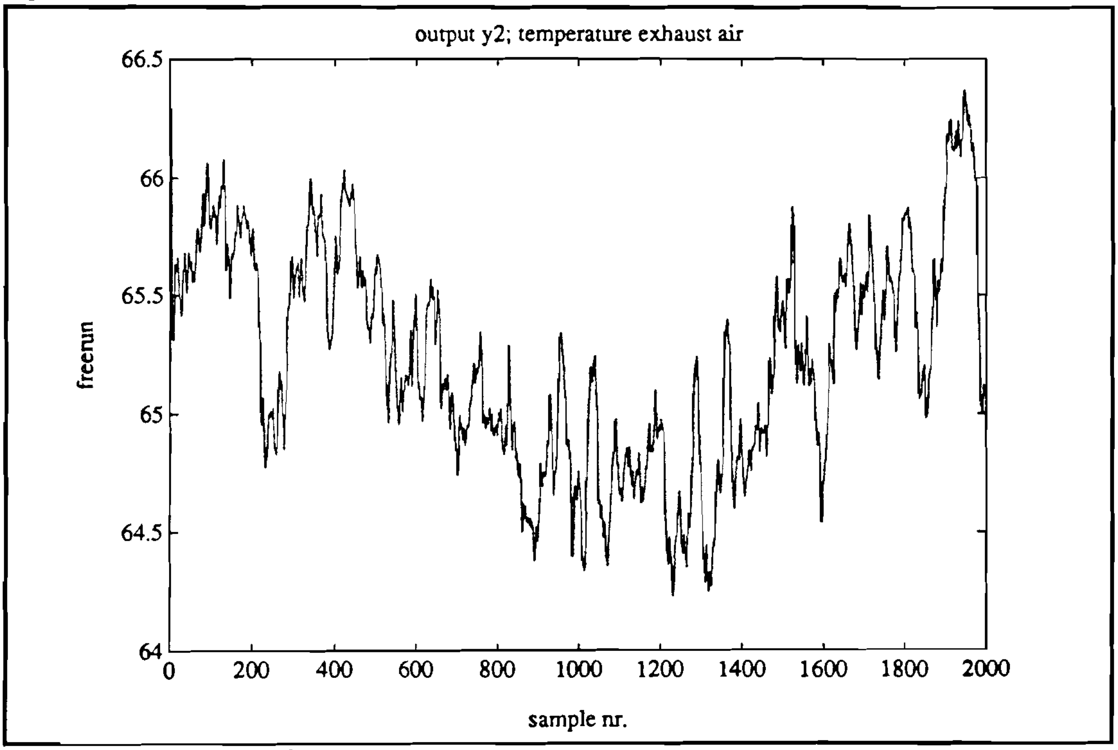


figure 5.2b.: *output y2 in free run experiment.*

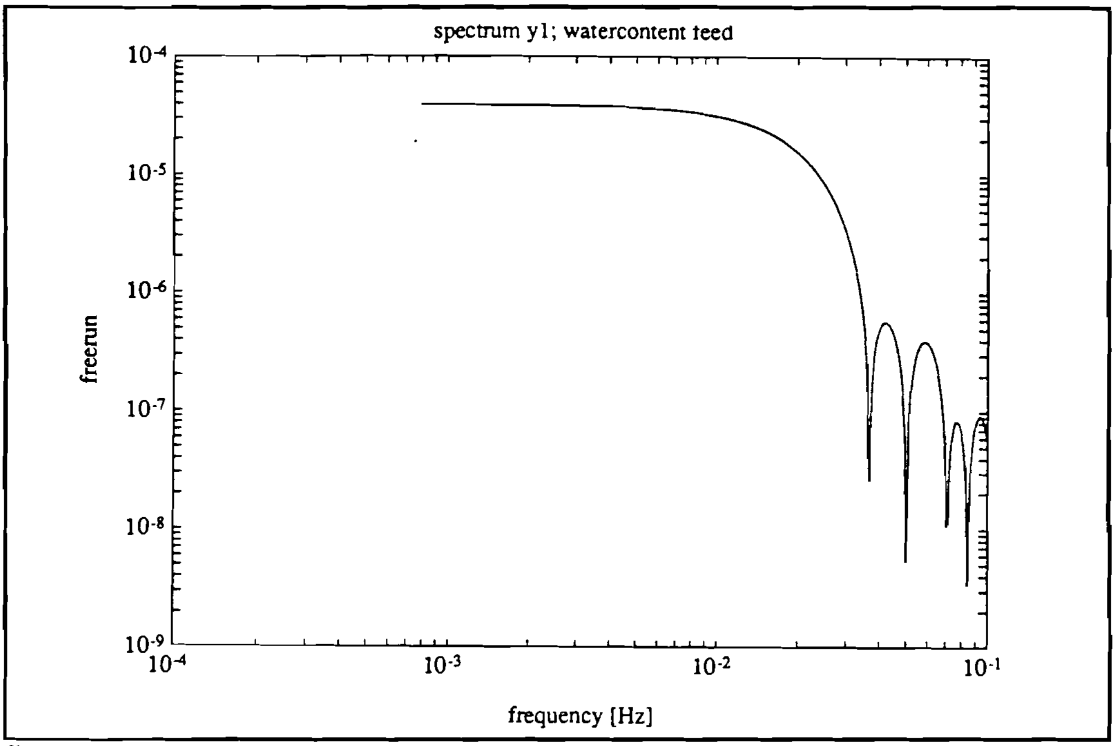


figure 5.2c.: *power spectrum of output y1.*

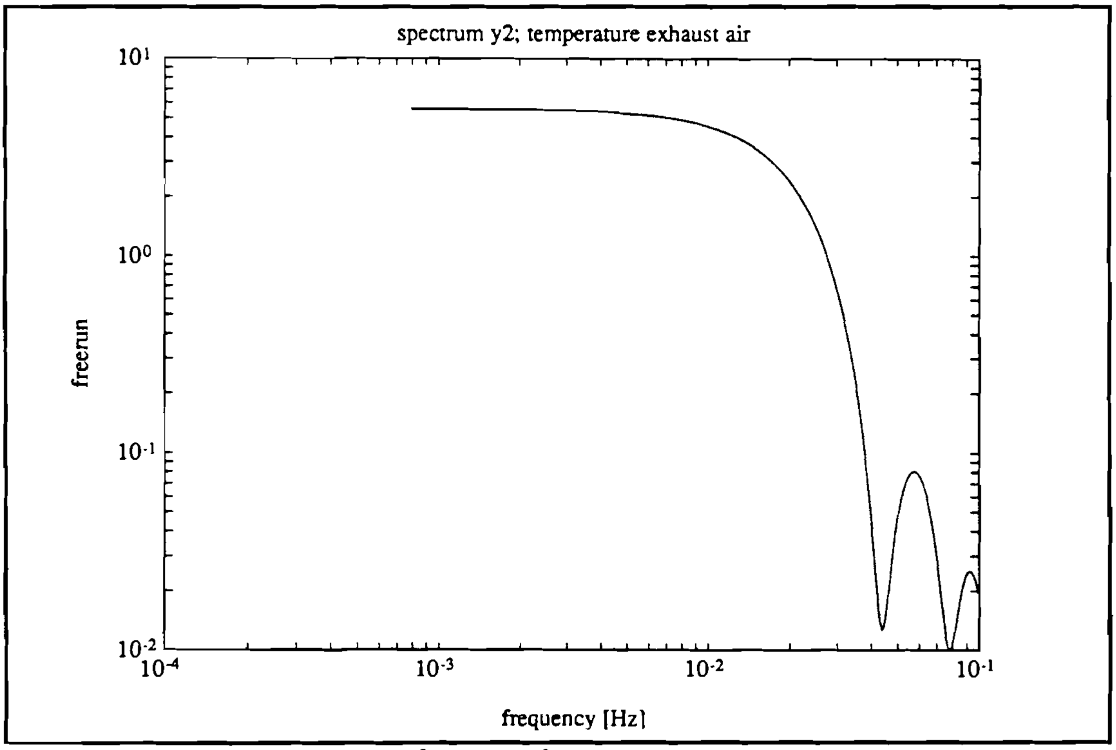


figure 5.2d.: *power spectrum of output y2.*

5.4 Fast PRBNS experiment.

To get an indication of the process bandwidth we perform a fast PRBNS experiment. In this experiment we choose as input a PRBNS noise sequence with a high sample frequency, so the input signal spectrum is flat in a large frequency band. The used sample frequency of 1 Hz gives a flat spectrum to at least 0.5 Hz [10].
The results indicate a maximum process bandwidth of approximately 0.01 Hz. This means that the final PRBNS experiment can have a maximum sample time of 25 seconds. When estimating the FIR model however we encounter a problem: because the length of the impulse responses from input 2 (feedflow) to both outputs is about 100 seconds this impulse response would be estimated by only 4 parameters in case of a sample time of 25 seconds. Therefore a sample time of 5 seconds was chosen which gives about 20 parameters for the fast characteristics (input 2 to both outputs) and unfortunately needs more than 100 FIR parameters for the slow characteristics (input 1 to both outputs). For practical reasons the number of FIR parameters is limited to 100, which describes all process impulse responses sufficiently.
The resulting plots of the fast PRBNS experiment are given if figure 5.3a-5.3b.

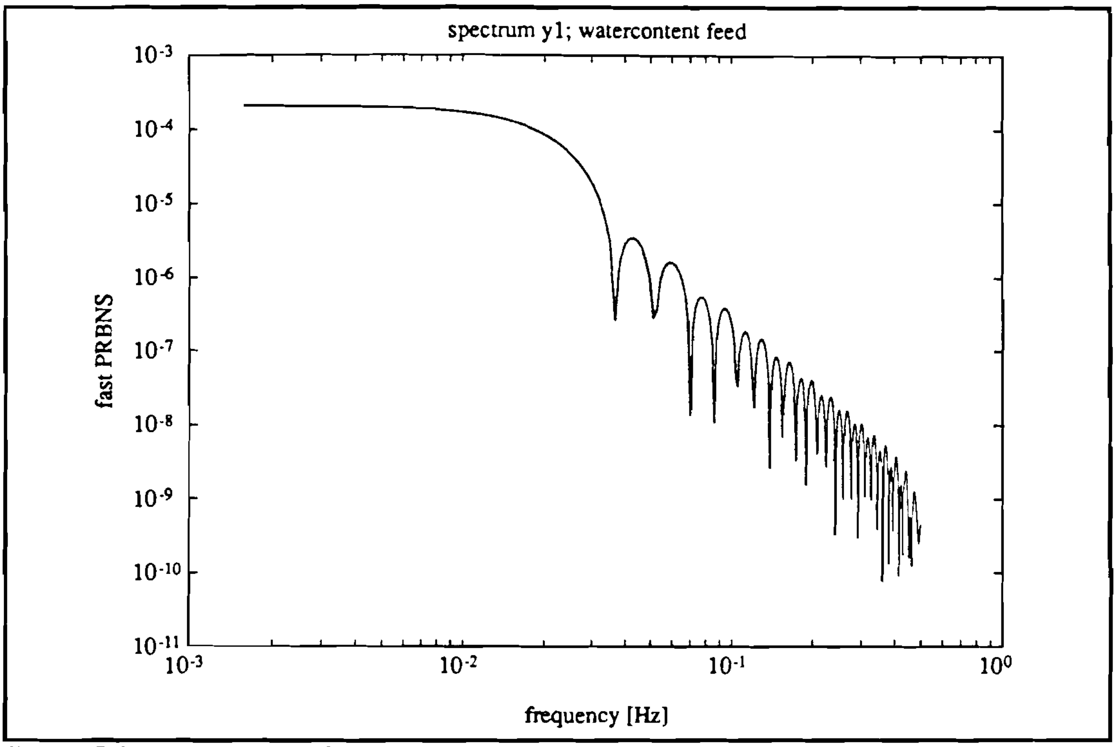


figure 5.3a.: *spectrum of y1.*

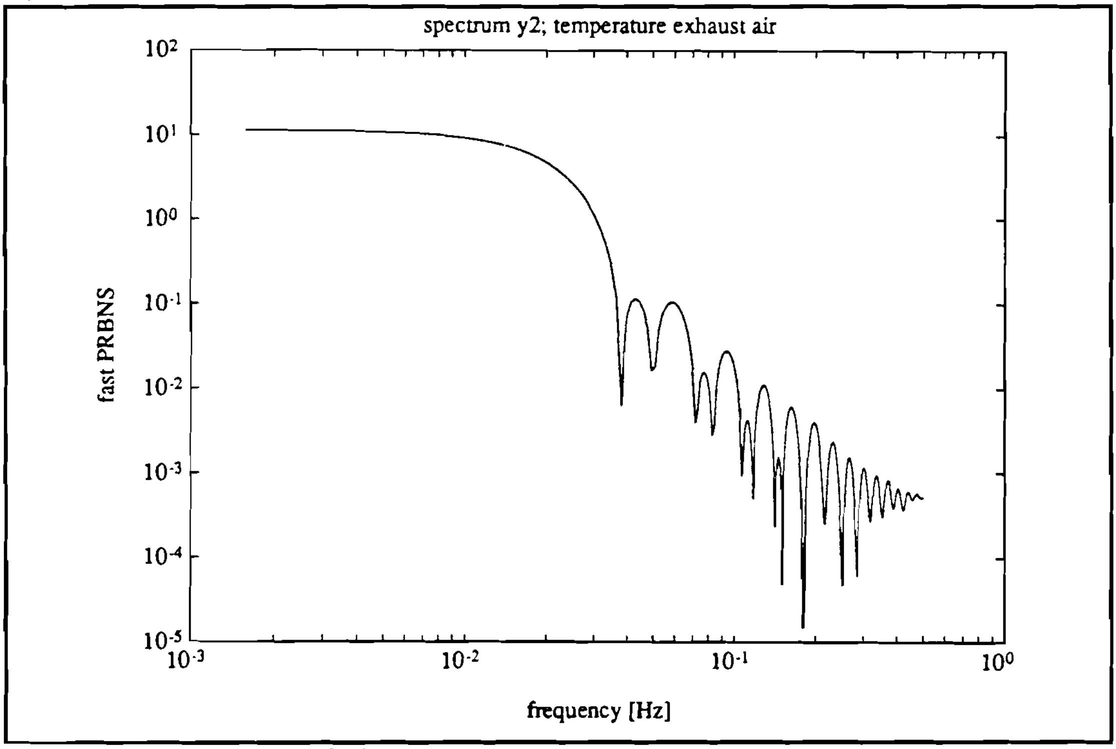


figure 5.3b.: *spectrum of y2.*

5.5 Final PRBNS experiments.

In the final PRBNS experiments both inputs were PRBNS signals with a sampling time of 5 seconds.
Two experiments were performed;

1. for obtaining a reference model the first data set was produced without noise (experiment called 'longzon'). Number of samples for estimation was 4000.
2. for obtaining the regular process model by standard identification the second data set was made with the additional noise described in section 4.4 (experiment called 'spraymet'). Number of samples for estimation was 2000.

The input signals for experiment 2 ('spraymet') are given in figure 5.4a-5.4g.

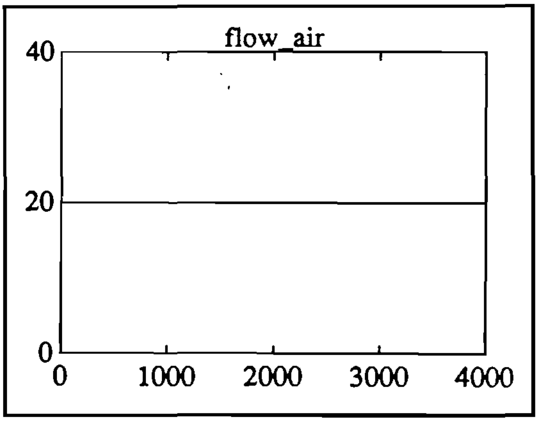
Cross correlations are made between input and output signals for estimating the delays. From the cross correlations the delay times were not clear, so they were not used for compensation.

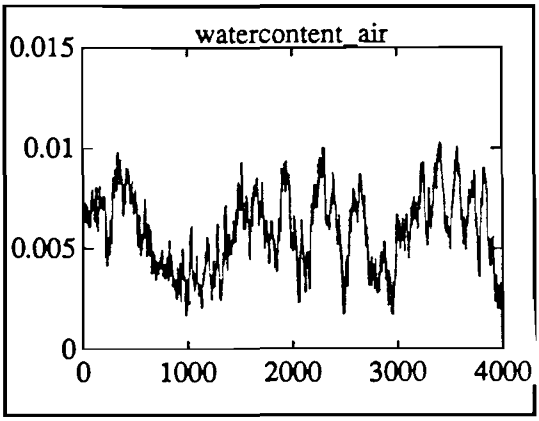
With the first dataset a FIR model estimation is performed according to paragraph 3.2. One hundred parameters are used for estimating the model. The FIR impulsresponses of the noise-free model are given in figure 5.5a-5.5d. The FIR model is transformed into an initial MPSSM model of order 8. Adjusting the initial MPSSM model to the dataset leads to the direct impulse responses of figure 5.6a-5.6d. Comparison of those results with the initial FIR impulse responses estimation shows a remarkably good resemblance.
Model reduction to a state-space model can now take place (to order 16 and 6).

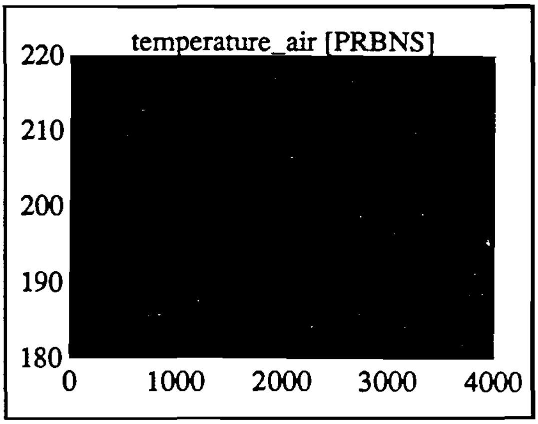
Also with the second dataset a FIR model estimation is performed according to paragraph 3.2. One hundred parameters are used for estimating the model. The FIR impulsresponses of the model with noise are given in figure 5.7a-5.7d. The FIR model is transformed into an initial MPSSM model of order 6. Adjusting the initial MPSSM model to the dataset leads to the direct impulse responses of figure 5.8a-5.8d. Comparison of those results to the initial FIR impulse responses estimation show a good resemblance.

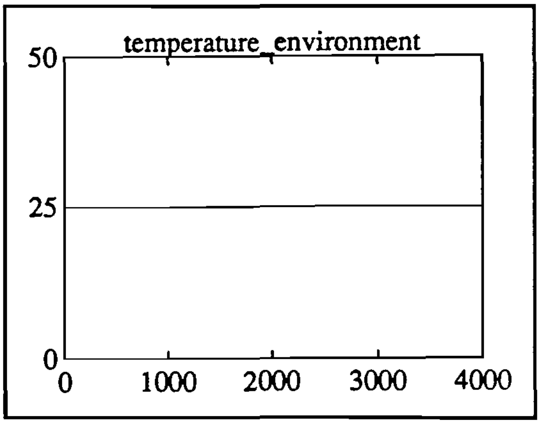
Model reduction to a state-space model is performed (to order 12 and 6).

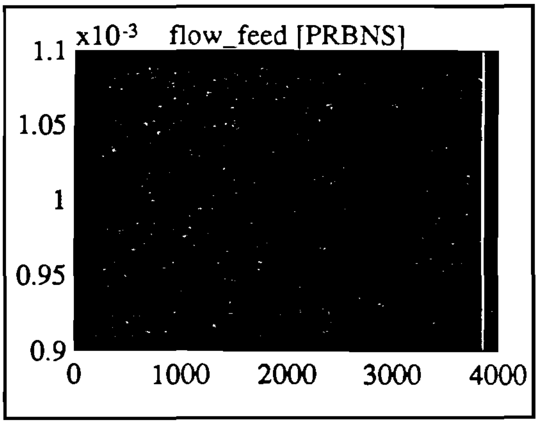
Validation of both models (noise-free and with noise) is performed and compared to the MDSPE experiment results in chapter 6.

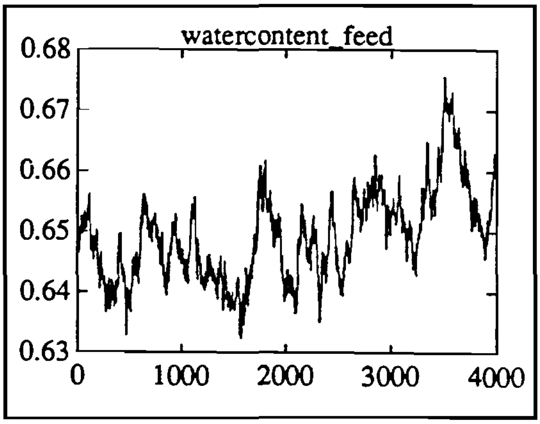












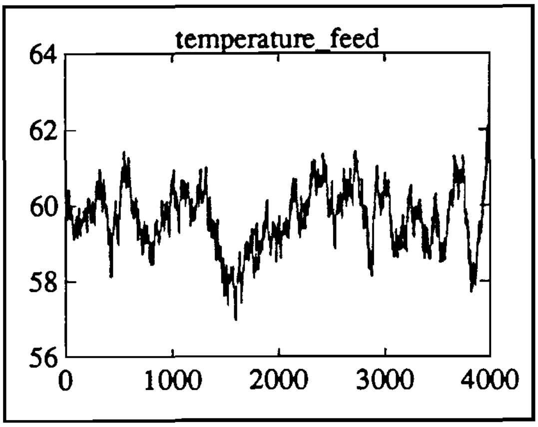


figure 5.4a-g: *input signals for experiment 2 (spraymet).*

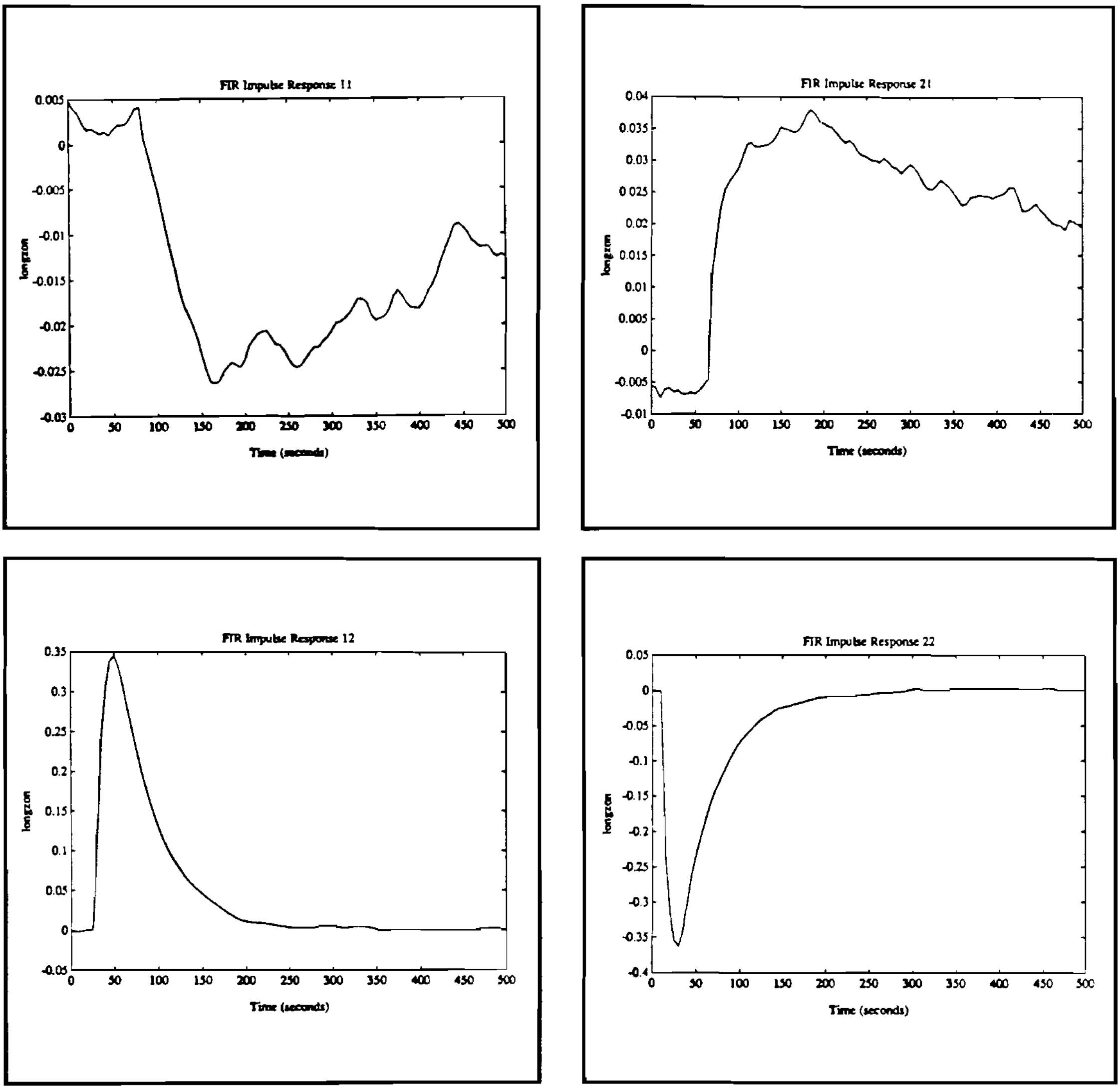


figure 5.5a-d: *FIR impulse responses of the noise free model.*

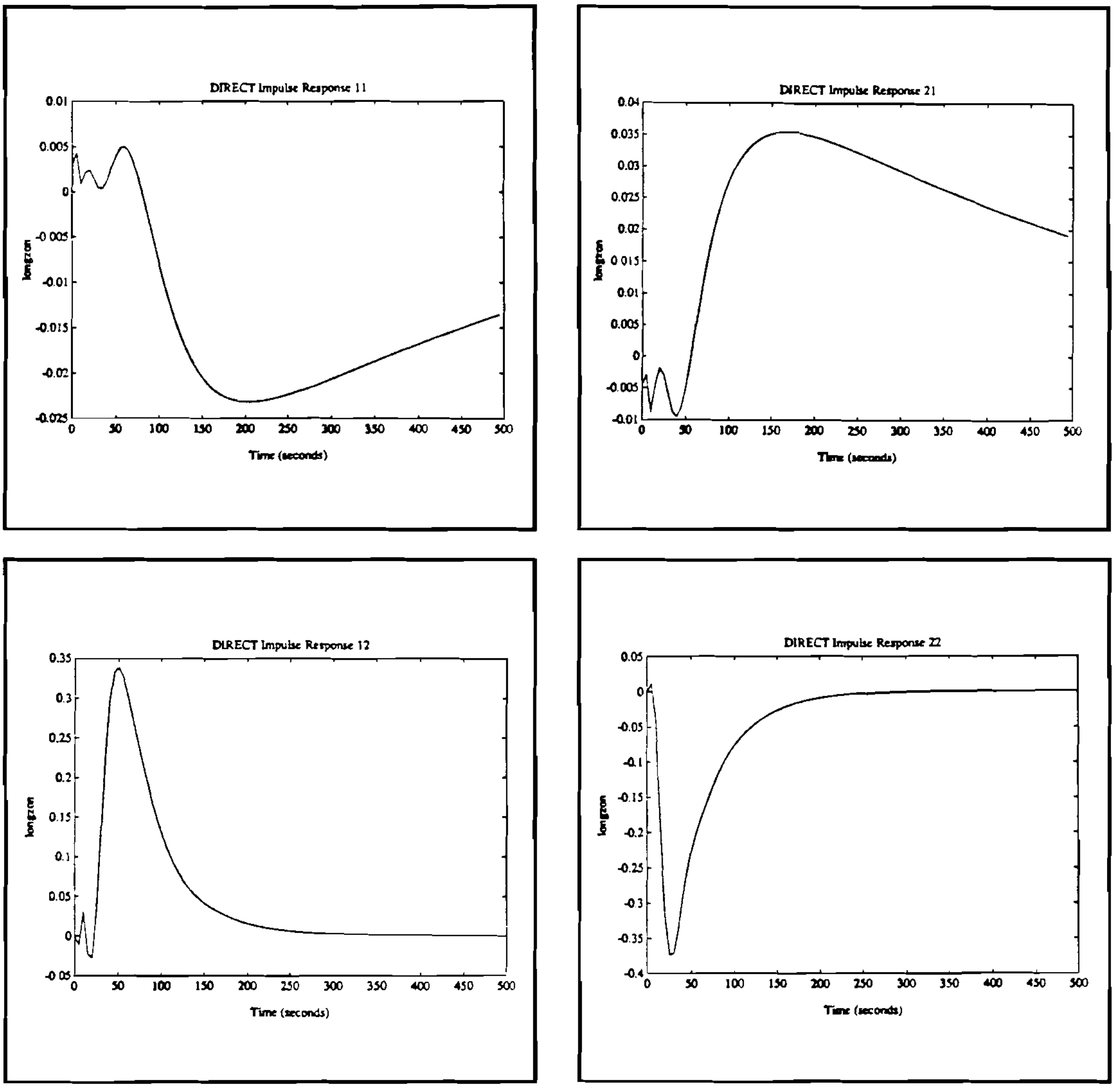


figure 5.6a-d: *Direct impulse responses of the noise free model.*

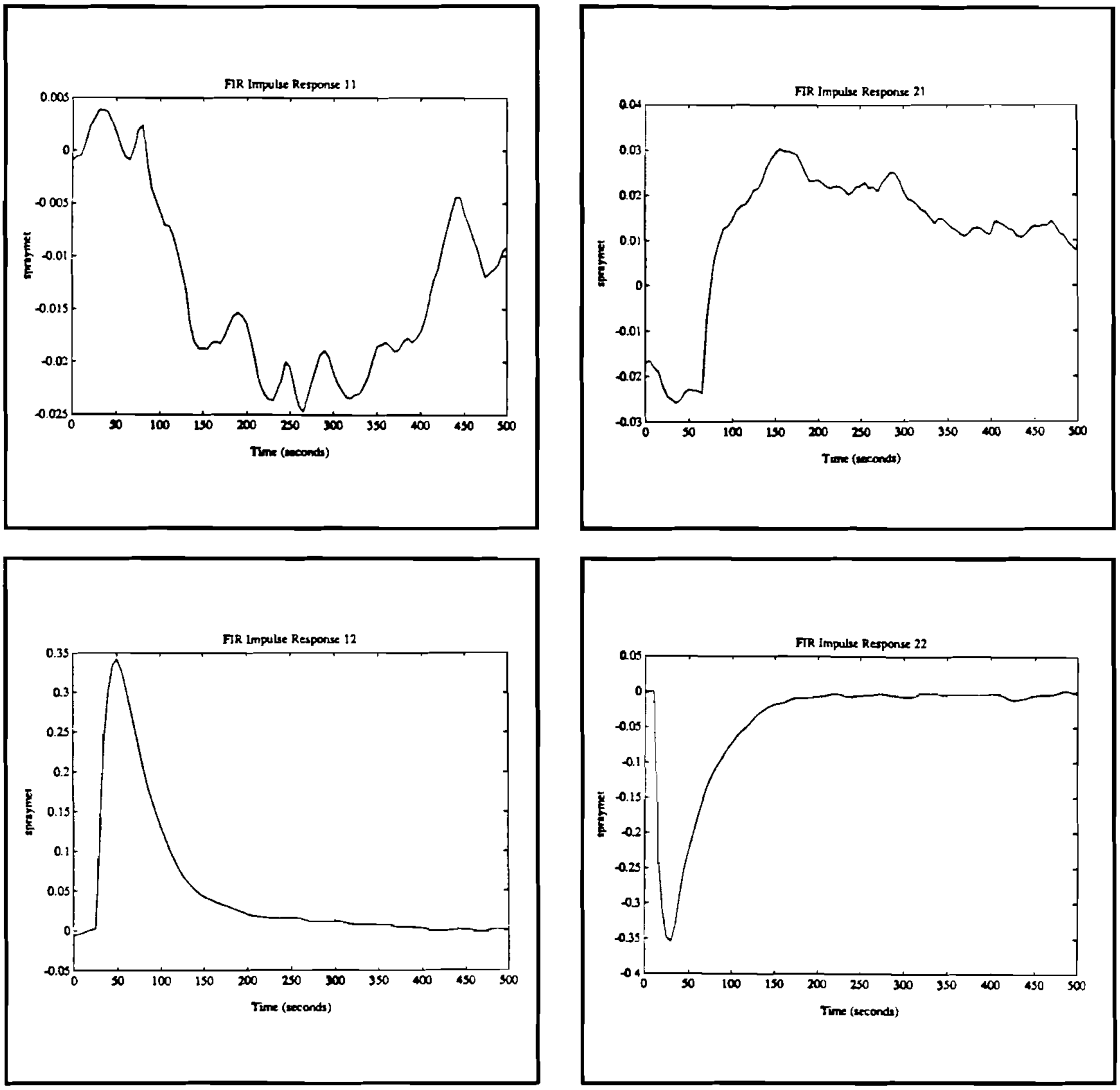


figure 5.7a-d: *FIR impulse responses of the model with noise (spraymet).*

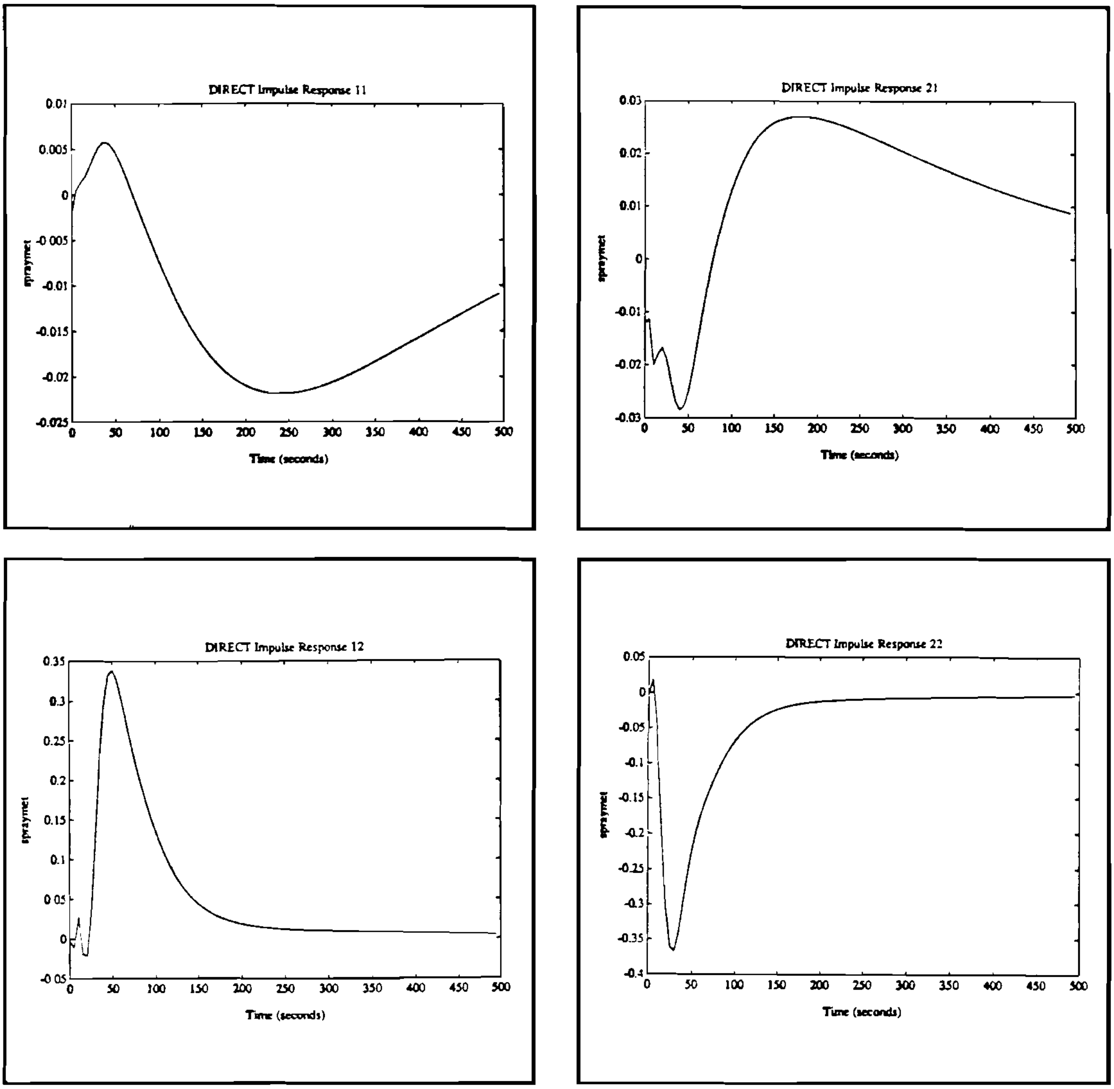


figure 5.8a-d: *Direct impulse responses of the model with noise (spraymet).*

6. MDSPE experiment results.

6.1 Modelling errors.

In the next sections two modelling error criteria are used.
The first modelling error criterion is the output error or tracking signal to noise ratio. This signal to noise ratio is calculated with formula 6.1.

$$J_{oe,i} = 10 * {}^{10}\log \left(\sum_{k=1}^{N} \frac{e(k)^2}{y(k)^2} \right) \qquad [6.1]$$

in which:

oe = output error = model output-process output
i = output number i
e(k) = output error = model output(sample k) - process output(sample k)
y(k) = process output (sample k)

The signal to noise ratio gives a direct indication of how well a model predicts the outputs of the process over a longer period of time.
The spectrum of the output error gives an indication of the distribution of the error over the frequency domain.

The second modelling error is given by the upperbound of the modelling error in the frequency domain as described in section 3.2. In case of estimated high order state space models the upperbound equals the three-sigma-bound. This is given by (for a MIMO system):

$$3\sigma_{ij} = 3\sqrt{\frac{n}{N}\Phi^{-1}_{u_j}(\omega)\Phi_{v_i}(\omega)} \qquad [6.2]$$

in which:

N = number of samples in model estimation
n = number of FIR parameters in initial estimation
i = output i
j = input j
Φ_u = input spectrum
Φ_v = output spectrum

6.2 MDSPE_1 experiment results.

From the fast PRBNS experiment and the final PRBNS experiment we obtain two data sets which can be coupled for estimating a better model. The fast PRBNS experiment had a sampling time of 1 second; the final PRBNS experiment however had a sampling time of 5 seconds. When coupling the data sets the sampling time must be made 5 seconds for both the data sets. Therefore the fast PRBNS data set (of 2000 samples) is modified. That is done by taking the mean value of every 5 samples. This also reduces the noise. Now both data sets are coupled; plots are given in figure 6.1a and 6.1b. This leads to a number of samples of 2000+2000/5 = 2400.
The model is estimated as a FIR output error least squares fit. This FIR model is transformed into an MPSSM start model. This initial model is again fitted on the data set and leads to a high order state space model. All these steps are according to chapter 2.
Validation results are given in table 6.1.

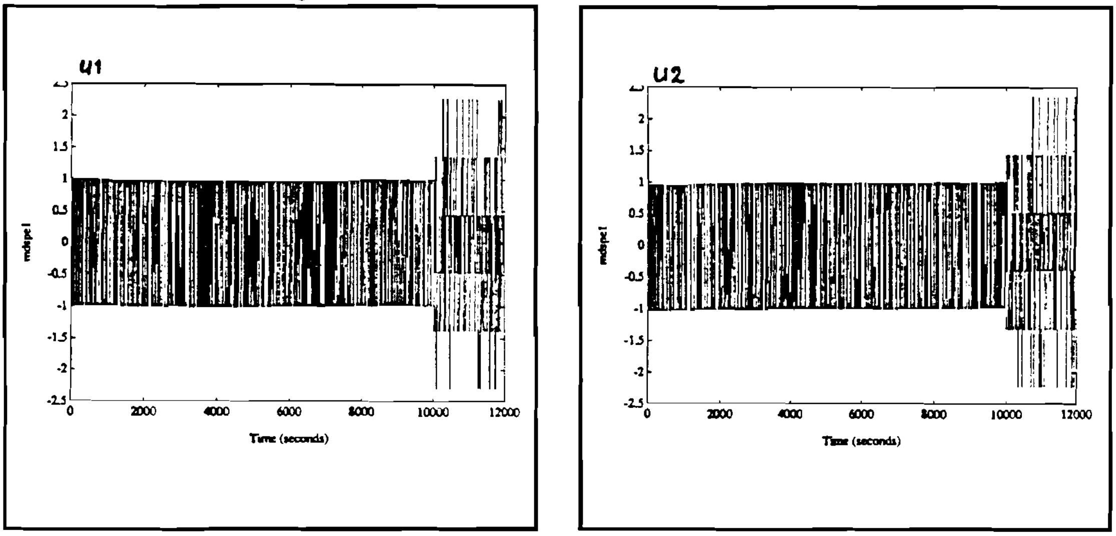


figure 6.1: *input signals for the MDSPE_1 experiment.*

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-10.3	-8.4
model with noise	12	2000	-9.6	-8.1
MDSPE_1 model	16	2400	-10.0	-7.7

table 6.1: *model validation on white PRBNS input signals.*

The validation is done on a data set not used in estimating the models (a so called cross validation). In table 6.1 three models are validated:

1. the noise free model which will be assumed to resemble the actual process closely.
2. the model estimated by the standard procedure with additional noise. This is the reference model to which the improvements should be made.
3. the MDSPE_1 model estimated on the coupled data sets of the final PRBNS data set (used in estimating model no. 2) and the modified fast PRBNS data set.

Looking at the validation results we see that no improvement is made by estimating the model by coupling the final and modified fast data set.
The spectra of the output error signals (model output minus real proces output) are given in figure 6.2a for the MDSPE_1 model. These can be compared to the spectra of the standard model with noise of figure 6.2b.
Comparison shows that the MDSPE_1 model gives less power content for the error signals at frequencies between 0.04 and 0.1 Hz. So although the error criterion function J shows no improvement the spectra show an improvement at high frequencies. This also shows the drawback of the criterion function J.
MDSPE_1 gives us a slightly better model at high frequencies.
When evaluating this result the conclusion can be made that if a second data set is added to the final PRBNS experiment this second data set should provide new process information in an interesting frequency part of the process spectrum. Therefore the experiment of section 3.6 was designed. Looking at the estimated upperbounds for the modelling errors in the frequency domain of the noise model in figure 6.3 gives a reason for a better estimation at low frequencies, because the modelling error is high at low frequencies.

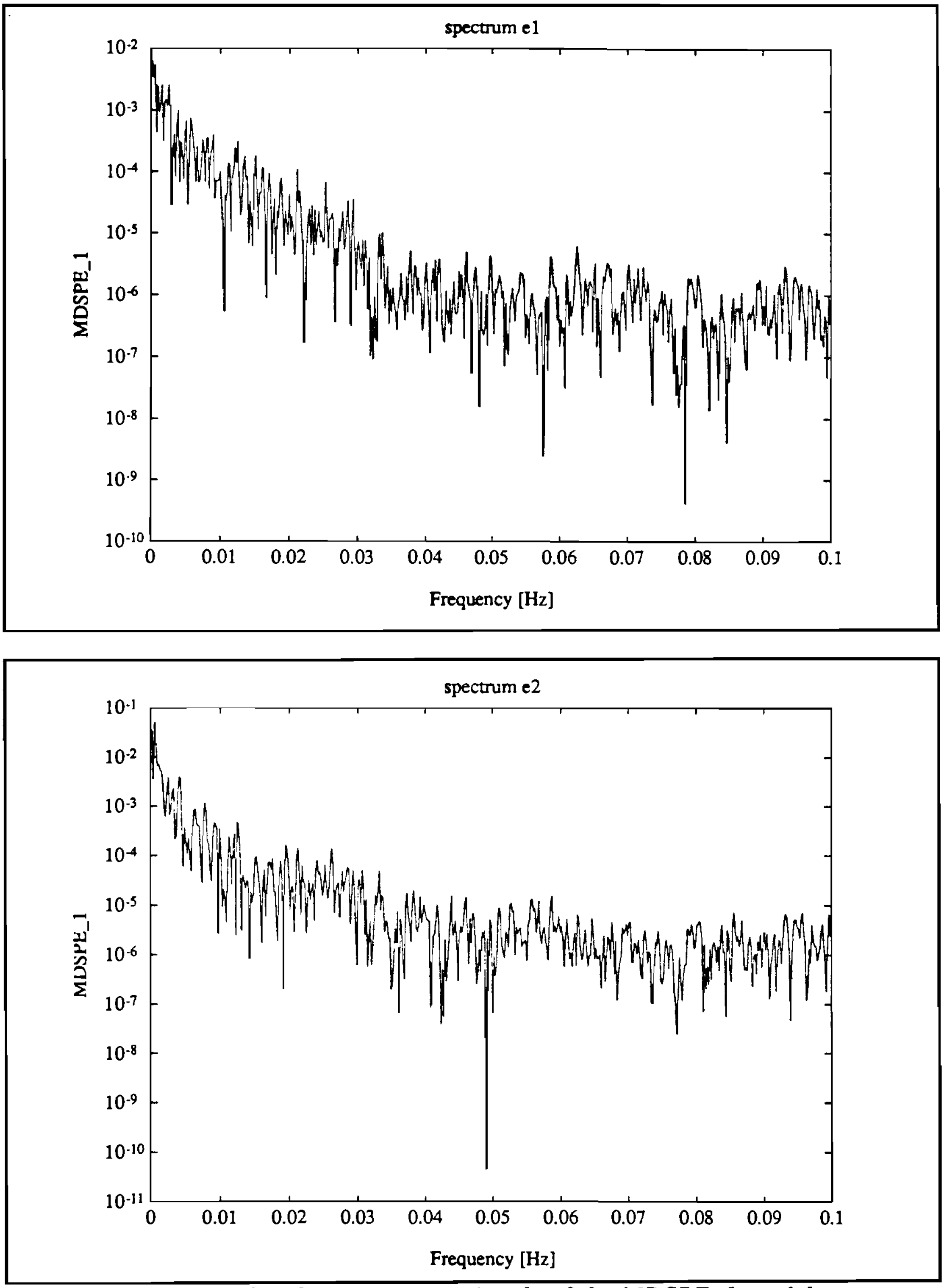
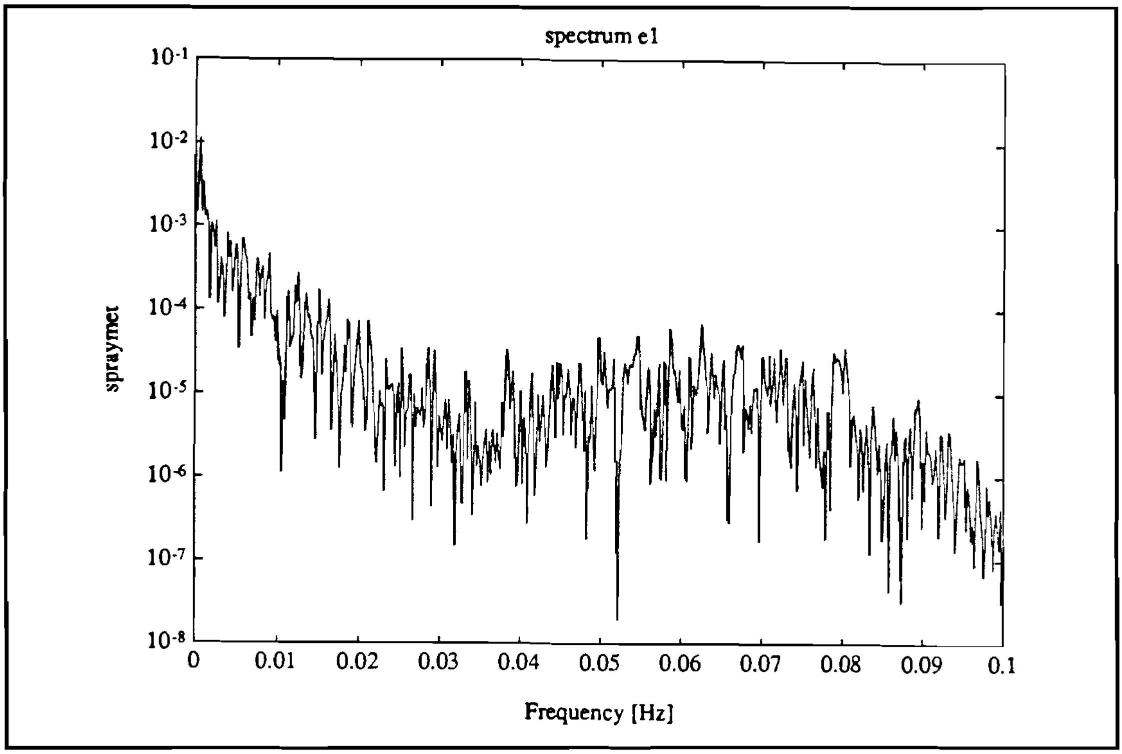


figure 6.2a: *spectra of both output error signals of the MDSPE_1 model.*



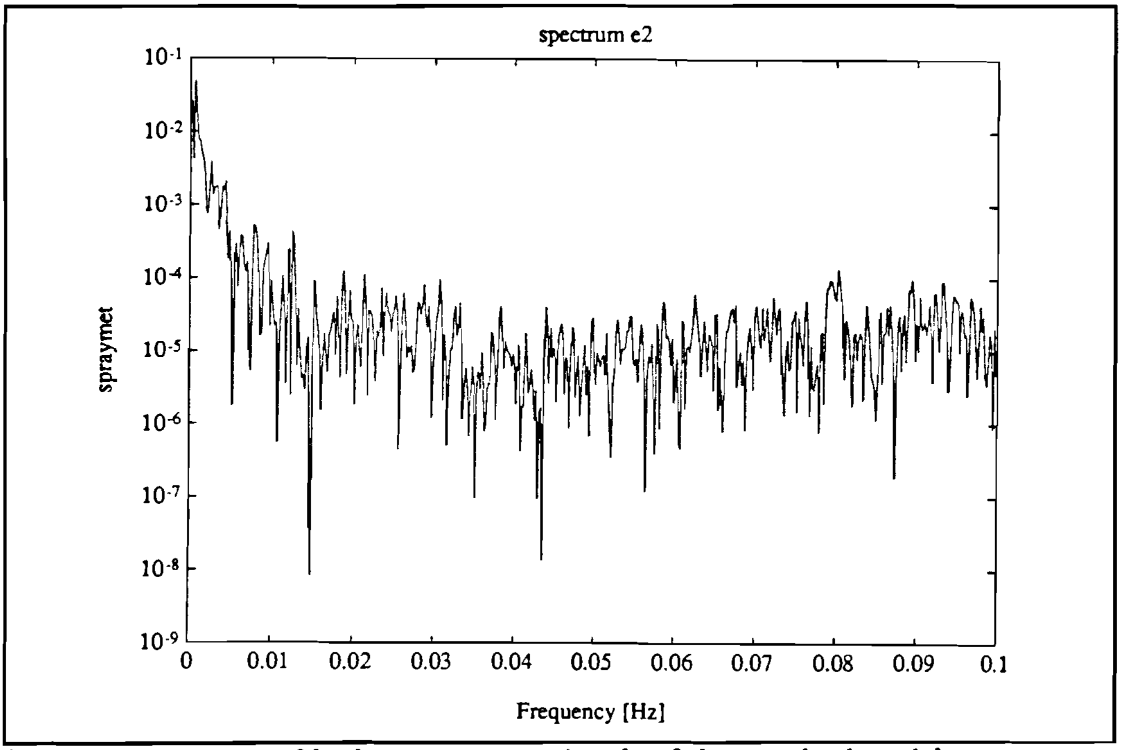


figure 6.2b: *spectra of both output error signals of the standard model.*

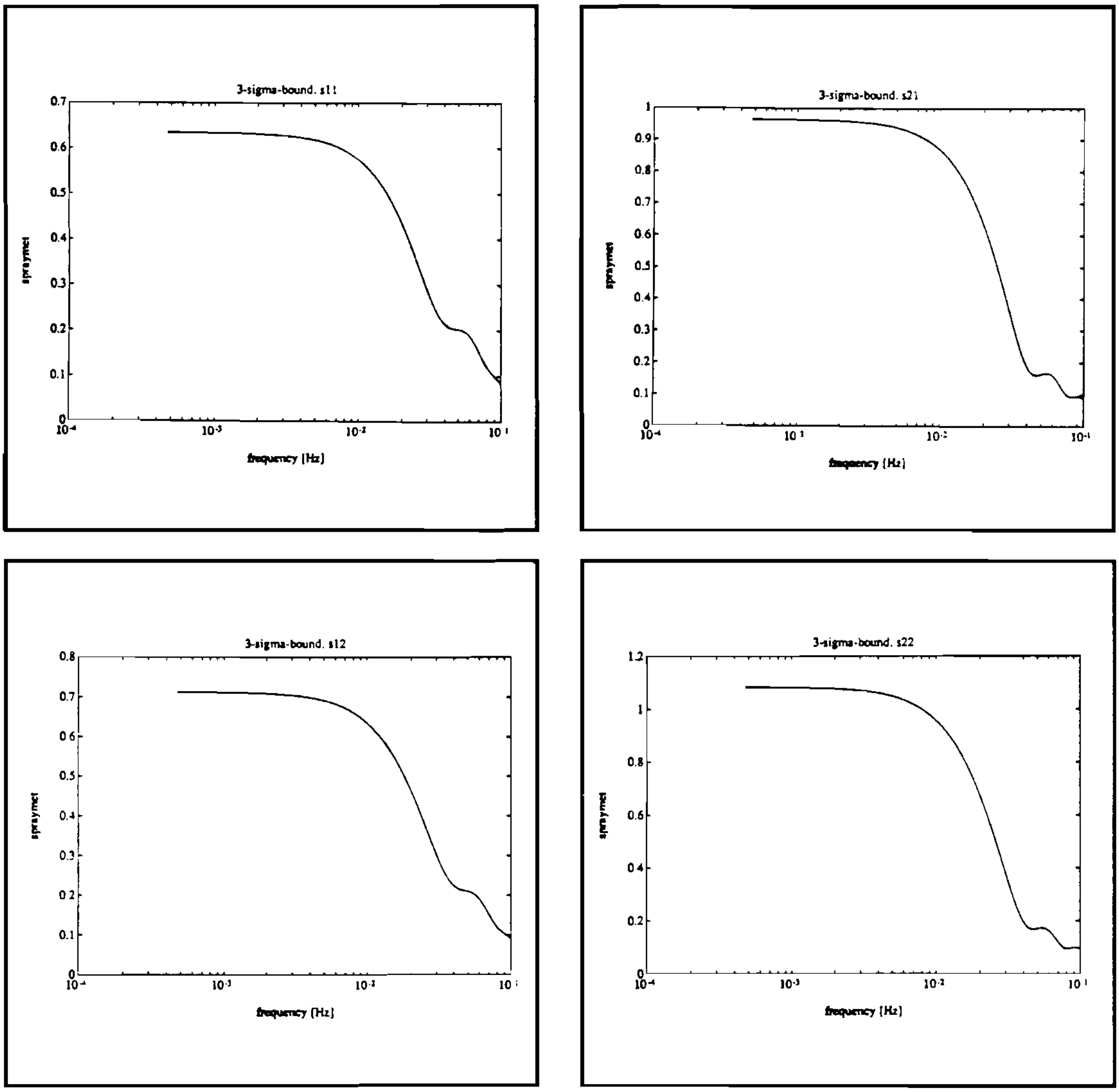


figure 6.3: *upperbound of the modelling errors in the frequency domain.*

6.3 MDSPE_2 experiment results.

A new experiment (called MDSPE_2) according to section 3.6 was performed. An input signal composed from low pass filtered (filtered with a second order Butterworth filter with cut-off frequency 0.01 Hz) PRBNS signals added with low power white noise (for numerical stability; see section 3.6). The total power content of the input signals remained constant. These input signals were applied to the spray dryer simulator and are given in figure 6.4a. Spectra of the input signals are given in figure 6.4b.

On this data set a model was estimated according the standard procedure of chapter 2.

Next this data set was coupled to the final PRBNS data set. The joined data set was used to estimate a new model (called MDSPE_2) according to the standard procedure. Note that the number of samples doubles to 4000. These newly estimated models have been validated on both white PRBNS input signals and on low pass filtered PRBNS input signals. Validations are given in table 6.2 and 6.3.

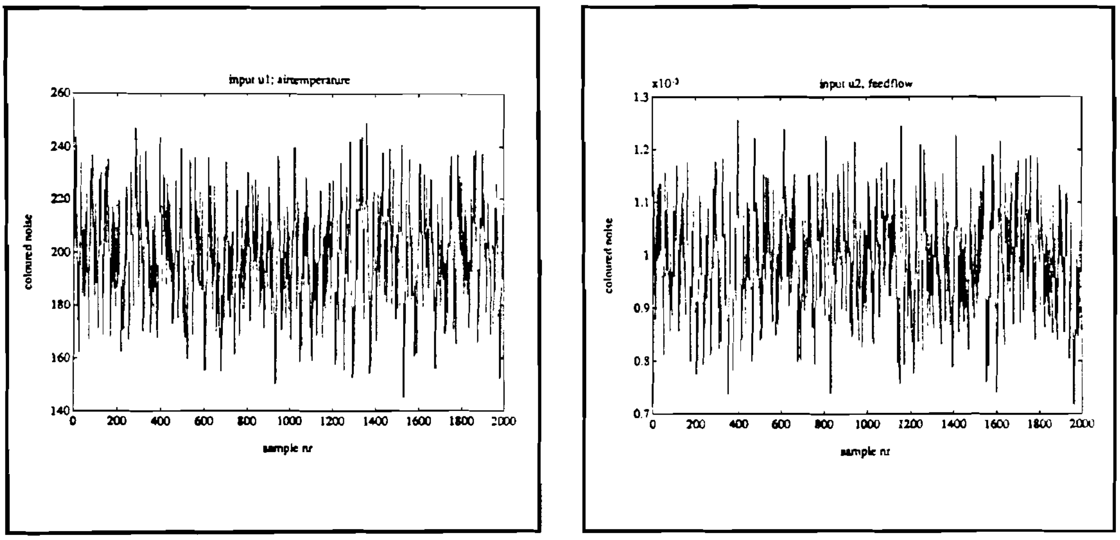


figure 6.4a: *low pass filtered input signals u1 and u2.*

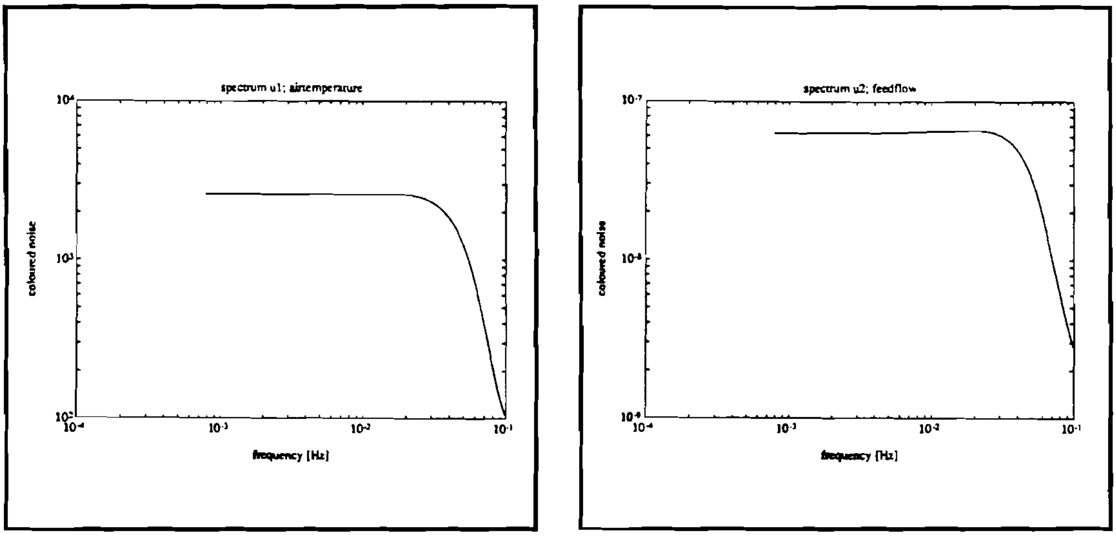


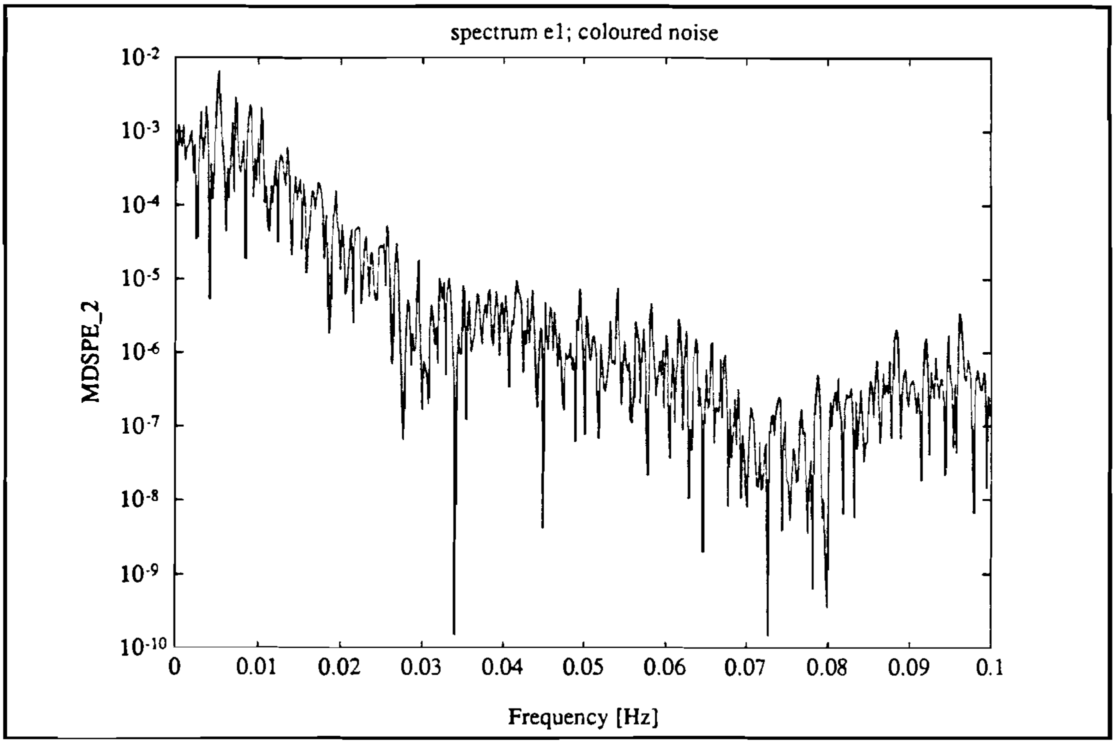
figure 6.4b: *spectra of the coloured noise input signals u1 and u2.*

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-11.4	-12.0
model with noise	12	2000	-10.8	-8.8
coloured noise model	12	2000	-12.2	-13.1
MDSPE_2 model	12	4000	-11.7	-13.1

table 6.2: *model validation on coloured noise input signals.*

When looking at the model validation on coloured noise input signals in table 6.2 first quite an improvement is made with only estimating a model on coloured noise inputs in comparison to the standard identification. For good understanding: the noise power at output y1 has decreased by 27% and the noise power at output y2 has decreased by 63%!
With the coupling of the data sets no further improvement is realized in comparison to the coloured noise model with this error criterion. The spectra of the error signals for the coloured noise model and the MDSPE_2 model are given in figure 6.5 and 6.6 respectively.
Looking at the spectra of figure 6.5 and 6.6 and comparing them with the output error spectra of the previously estimated models (for coloured noise input signals) shown

in figure 6.7 and 6.8 shows improvement is found for model MDSPE_2 with respect to the other models. The amplitude spectra are somewhat lower at low frequencies. The spectra of model MDSPE_1 show smaller amplitudes at high frequencies but lower than for all other models. Therefore at this point the model MDSPE_2 is considered to be the best general model for all frequencies with these input signals. The shape of the spectra is caused by the shape of the input signals, the noise spectra and the transfer function. (Note that comparison is a bit difficult due to the logarithmic y-axis but normal representation is impossible due to the large differences in amplitude).



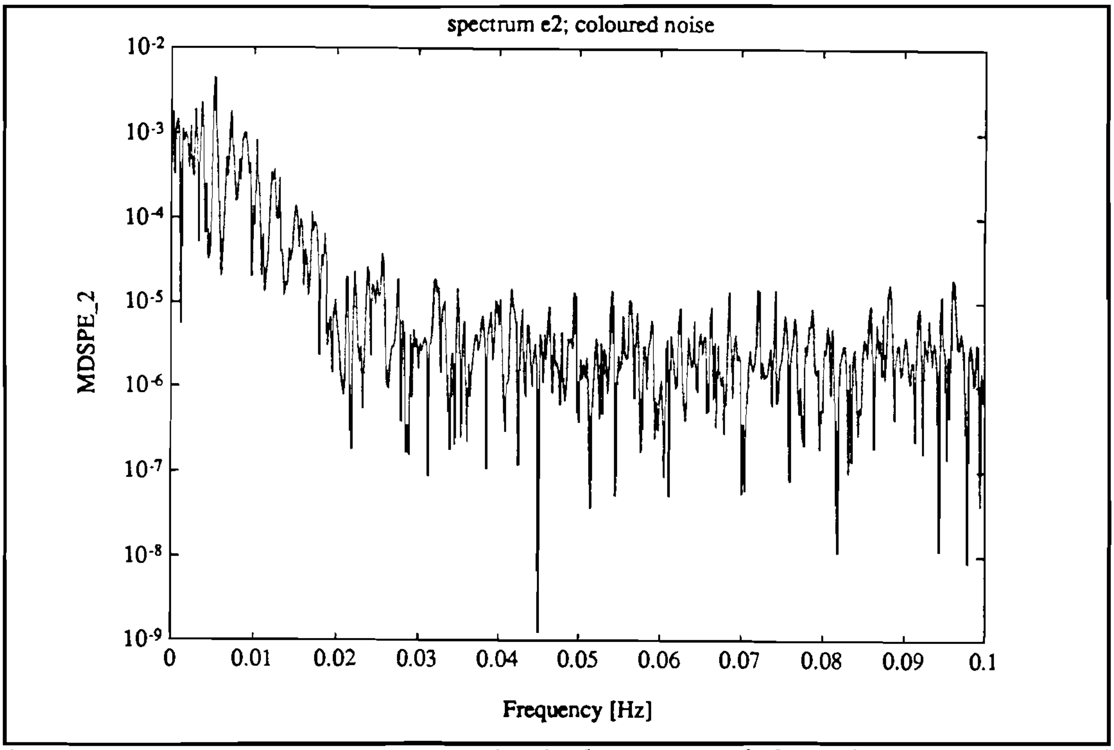
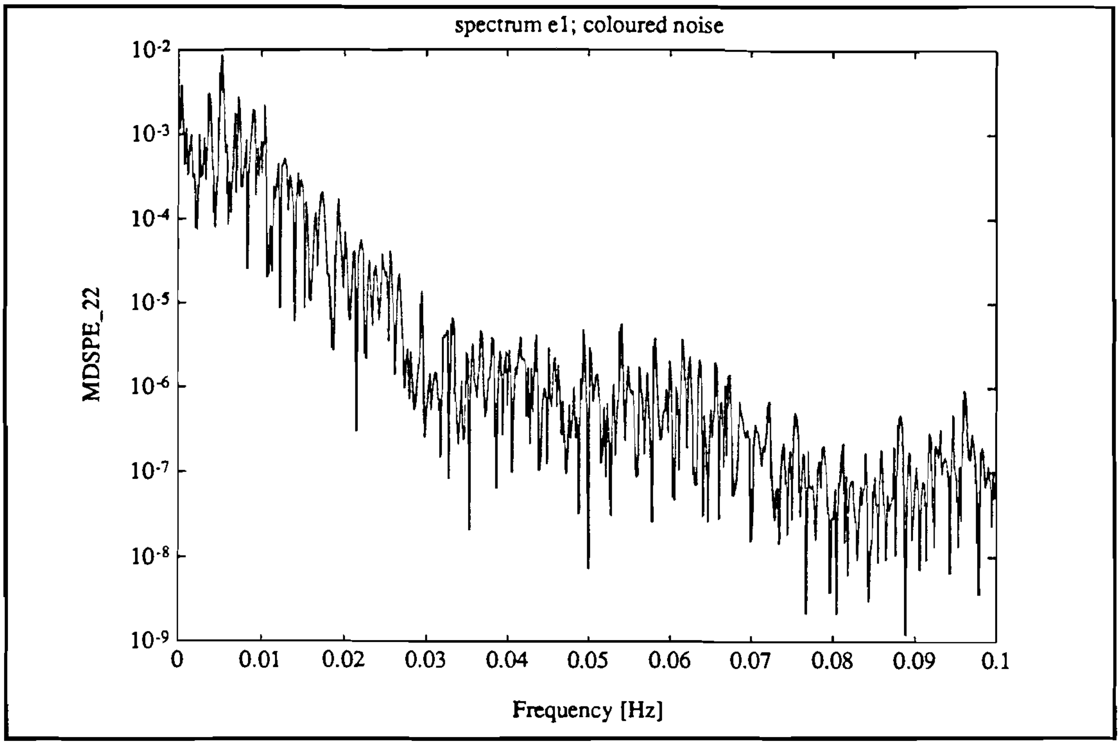


figure 6.5: *output error spectra for both outputs of the coloured noise model (coloured noise input signals).*



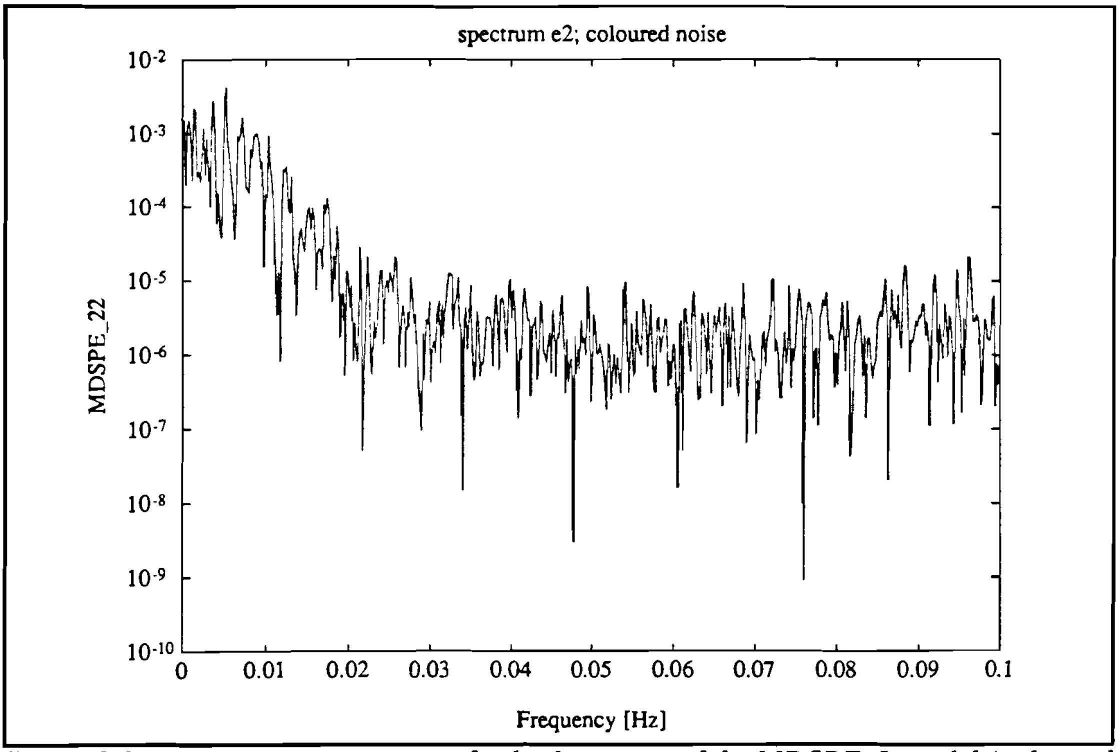
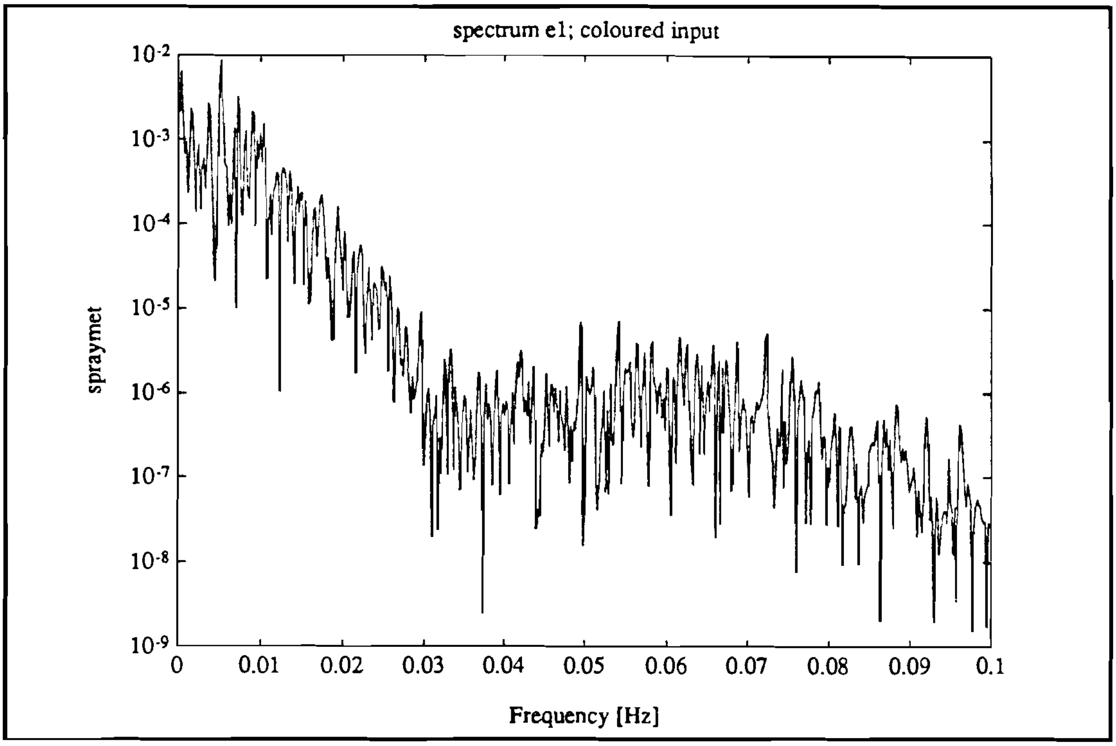


figure 6.6: *output error spectra for both outputs of the MDSPE_2 model (coloured noise input signals).*



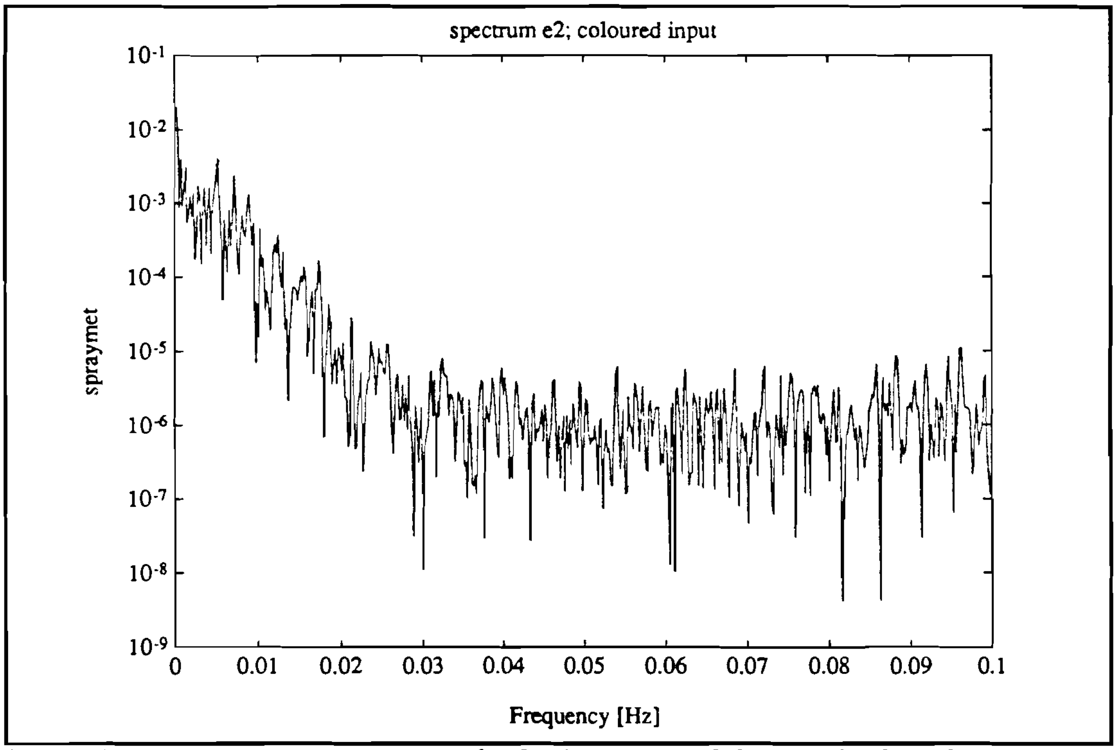


figure 6.7: *output error spectra for both outputs of the standard model (coloured noise input signals).*

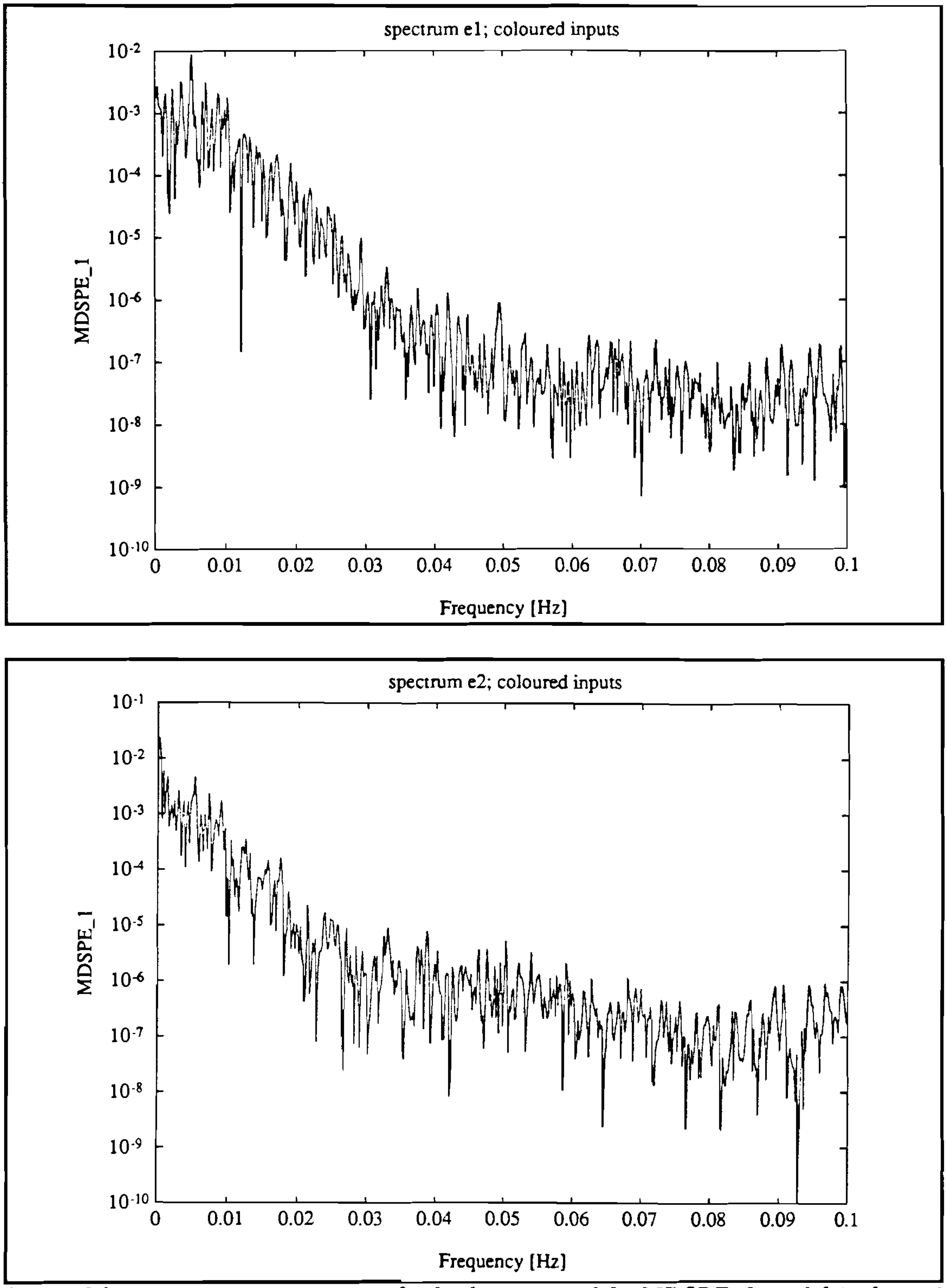


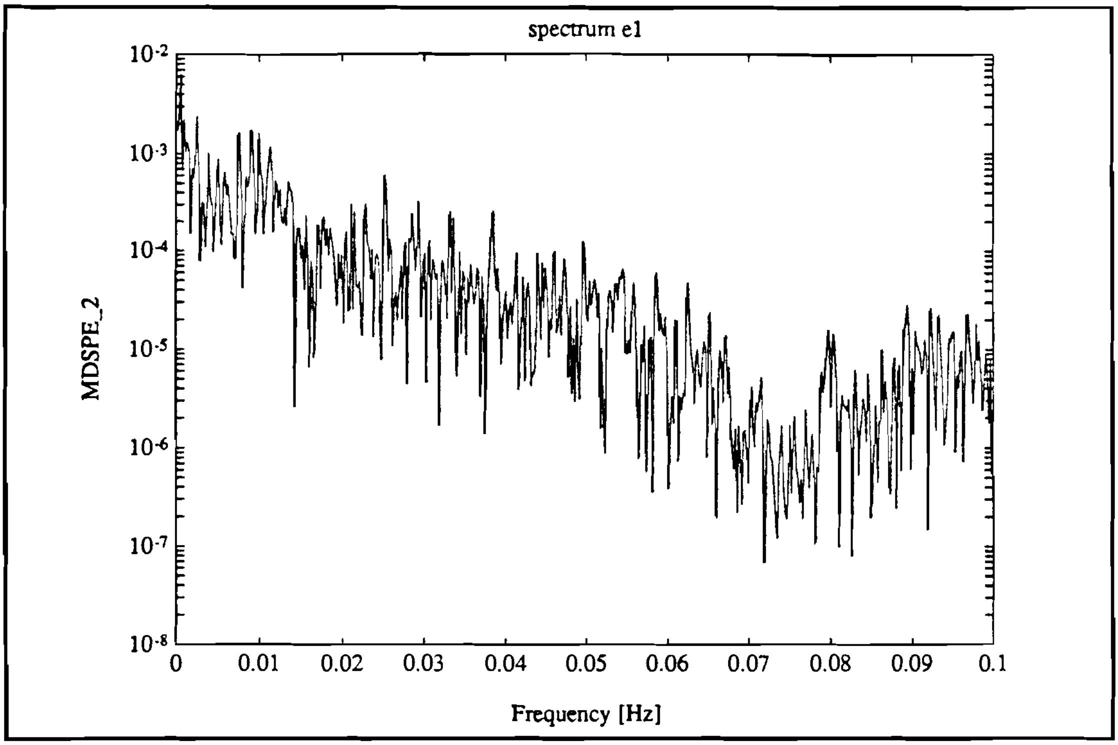
figure 6.8: *output error spectra for both outputs of the MDSPE_1 model (coloured noise input signals).*

The MDSPE_2 model should perform better on white noise inputs than the coloured noise model because more information about high frequencies is incorporated in the model. Validation is done on white noise PRBNS input signals. Results are given in table 6.3.

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-10.3	-8.4
model with noise	12	2000	-9.6	-8.1
coloured noise model	12	2000	-9.8	-7.5
MDSPE_2 model	12	4000	-9.9	-7.9

table 6.3: *model validation on white PRBNS input signals.*

Looking at the results in table 6.3 some improvement is made with the MDSPE_2 model with respect to the coloured noise model (as expected). Therefore the conclusion can be that the MDSPE_2 model performs better for all frequencies. More important however is the fact that both coloured noise models perform as well as the standard model (model with noise) on white PRBNS input signals but perform much better on low pass filtered PRBNS inputs on which the controller mainly operates. Therefore in case of the spray dryer and likely for many more processes the standard identification procedure with white noise PRBNS inputs can be improved by using low pass filtered PRBNS input signals (with cut-off frequency equal to the process bandwidth) with equal signal power and using both data sets according to the MDSPE_2 procedure (the modified fast data set can also be incorporated in the model estimation).
Spectra of the output error signals when applying white noise PRBNS input signals for the coloured noise model and the MDSPE_2 model are given in figure 6.9 and 6.10 respectively.
When estimating the MDSPE_2 model the coupling point of the two data sets can cause an estimation problem because switching information between the two data sets is not available (worst case approach). This problem can be solved by using two Hamming windows on the data set which take out the switch in and off data (the complete data set is multiplied by the Hamming values). The Hamming windows are plotted in figure 6.11.
Looking at the validation results in table 6.4 and 6.5 and comparing them with the results of the MDSPE_2 model shows that the Hamming windowed model is



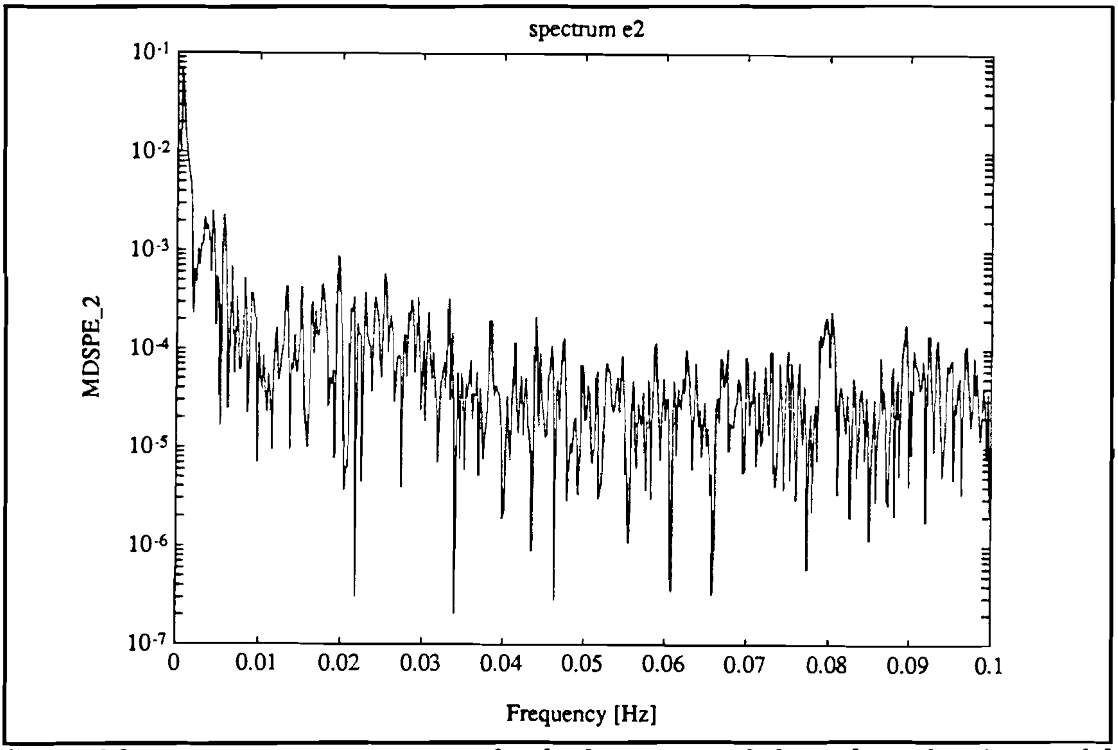
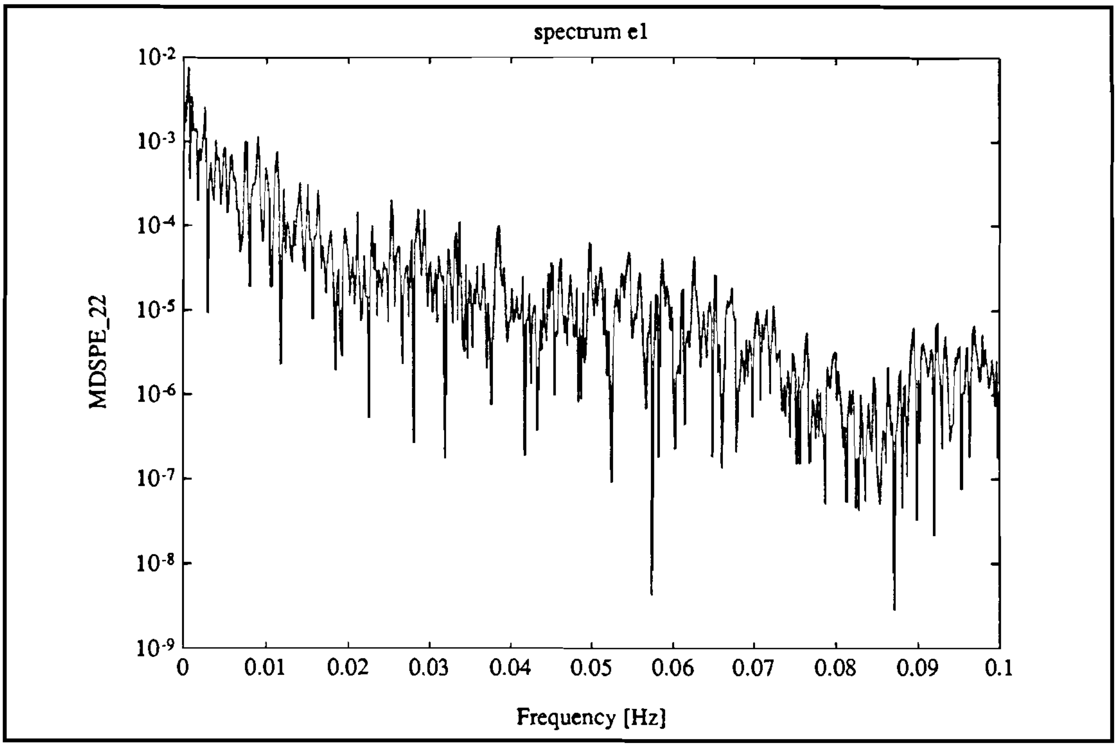


figure 6.9: *output error spectra for both outputs of the coloured noise model (white noise input signals).*



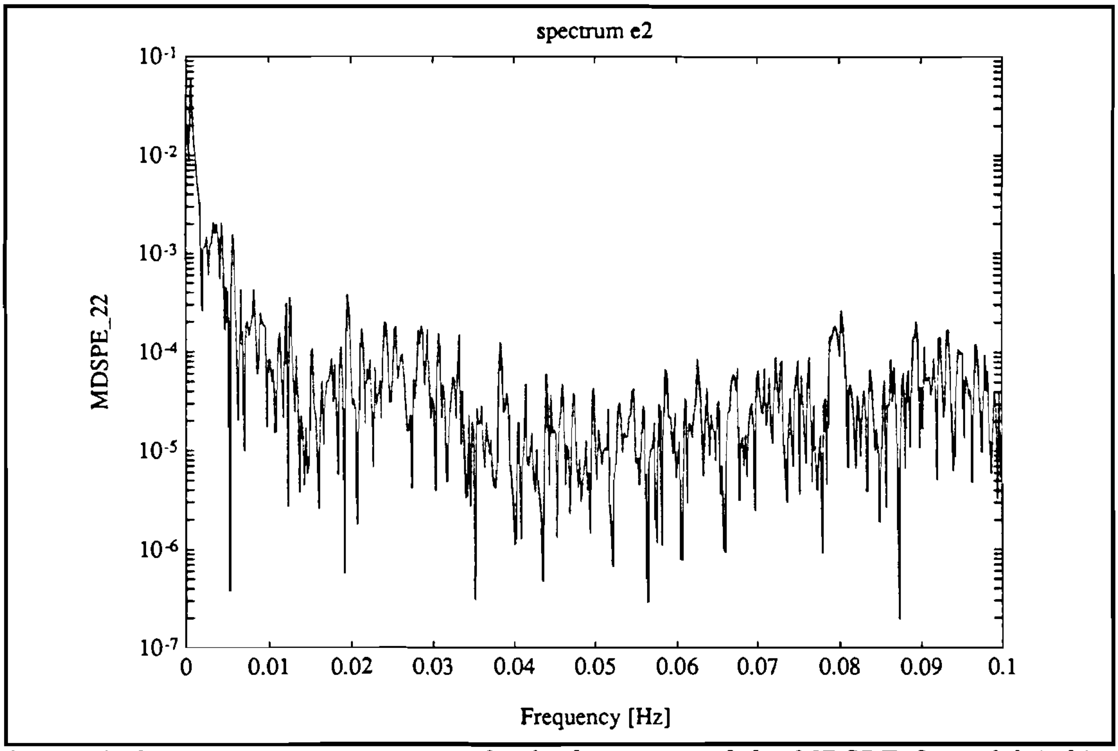


figure 6.10: *output error spectra for both outputs of the MDSPE_2 model (white noise input signals).*

somewhat better for coloured noise but worse for white noise. This effect can be explained as follows: the Hamming operation filters certain high frequency characteristics out of the data set. A conclusion might be that application of Hamming windows does not improve the model; therefore the best model until now is the MDSPE_2 model.

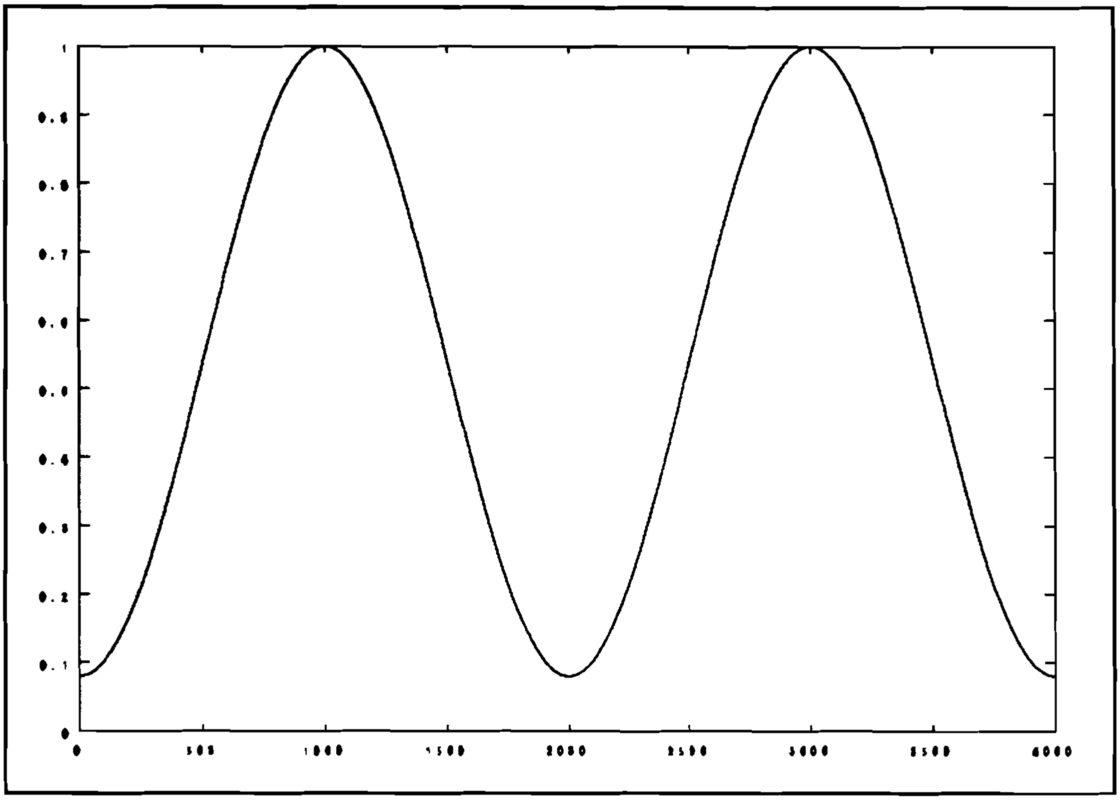


figure 6.11: *the applied Hamming window on data set MDSPE_2.*

	order	N	J_{oe1}	J_{oe2}
MDSPE_2 model Hamming windowed	12	4000	-9.4	-7.7

table 6.4: *Hamming windowed MDSPE_2 results; validation on white noise.*

	order	N	J_{oe1}	J_{oe2}
MDSPE_2 model Hamming windowed	12	4000	-12.0	-13.5

table 6.5: *Hamming windowed MDSPE_2 results; validation on coloured noise.*

6.4 MDSPE_3 experiment results.

When having performed the experiments of section 3.6 (experiment name MDSPE_2) we obtain two models:

1. a model from the final PRBNS experiment
2. a model from the coloured noise experiment (MDSPE_2)

Both models are high order state space models for which the theory of section 3.2 is valid. So for both models we can give a two or three sigma upperbound for the model error in the frequency domain according to section 3.2 (for the standard model with noise these three sigma upperbounds are given in figure 6.3). In experiment MDSPE_2 of section 3.7 one tries to reduce the model error by coupling both generated data sets and estimating a new (improved) model from this coupled data set. Another way of estimating a model which fits well on both already estimated models is by generating new data (which can have nearly infinite length) with both models and their model errors in the frequency domain and estimating a new (improved) model (called MDSPE_3) on this data. The scheme is given in figure 3.4.
Input signals u1 and u2 are PRBNS signals. Both outputs are filtered by a filter (it doesn't matter whether the inputs or outputs are filtered, because linear models are considered):

- the outputs of the MDSPE_2 model are filtered by a second order Butterworth filter with cut-off frequency 0.01 Hz. That because the MDSPE_2 model performs better for those frequencies than the standard model (see model errors of the standard model in figure 6.3).
- the outputs of the standard model are filtered by a second order high pass Butterworth filter with cut-off frequency 0.015 Hz. The cut-off frequency is chosen 0.015, because then the summed filter characteristics are nearly 1 for all frequencies (magnitude).

Before filtering the outputs of both models are scaled back to their physical values. This is necessary because the model estimation program IPCOS divides input and output signals by their variance (which are mostly different for different models). After filtering (the high pass filter filters out all bias) the corresponding outputs of both models are added and with the input signals a new model estimation is performed: this leads to model MDSPE_3.
A validation experiment with white noise input signals and low pass filtered white noise input signals is performed according to the previous experiments for comparison. The results for model MDSPE_3 are given in table 6.6 and 6.7.
The results show a dramatic loss of performance. This means that this type of model estimation will not work for these models.
An explanation for this misfit is that $F_1(j\omega) + F_2(j\omega) \neq 1$ for the phase in case of Butterworth filters. But even when using filters with phase 0 (anticausal filtering for instance) the transfer function will be influenced by the filter characteristics and the data set will not represent the process anymore.
Results give reason for not using this method in estimating a model.

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-10.3	-8.4
model with noise	12	2000	-9.6	-8.1
MDSPE_3 model	12	4000	-7.9	-5.5
MDSPE_2 model	12	4000	-9.9	-7.9

table 6.6: *model validation on white PRBNS input signals.*

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-11.4	-12.0
model with noise	12	2000	-10.8	-8.8
MDSPE_3 model	12	4000	-3.7	-5.6
MDSPE_2 model	12	4000	-11.7	-13.1

table 6.7: *model validation on coloured noise input signals.*

6.5 MDSPE_4 experiment results.

In experiment MDSPE_1 we accomplished to get a better performance of the model at high frequencies by using the fast PRBNS data set in the estimation process as well. In experiment MDSPE_2 we accomplished to estimate a better model at low frequencies by using a low pass coloured noise data set. This data set however has to be generated on the actual process which increases our number of measurements. Therefore a new experiment is performed in which we coupled the final PRBNS data set, the modified fast PRBNS data set and the stepresponses (which are allways measured so no new data has to be generated) for the low frequencies. This data set was used in a model estimation in which we hoped to achieve a model that is valid for all frequencies.
The input and output signals are given in figure 6.12.
The IPCOS model estimation program did not provide a good model on this data set. The FIR impulse responses of input u2 (feedflow) to both outputs (fast time constants) looked good but the FIR impulse responses of input u1 to both outputs were not estimated well. The estimation algorithm failed on this data.
An explanation might be that step responses put much weight to the FIR estimation. The DC gain must be estimated well. With the amount of 100 parameters this is not possible for the long impulse response. May be with much more parameters this would be possible. This was not possible for practical reasons. Therefore for model estimation at low frequencies low pass coloured PRBNS input signals must be used as in experiment MDSPE_2 with IPCOS Shell.
With the data set of figure 6.12 we try to perform the ARX model estimation of chapter 3.8.

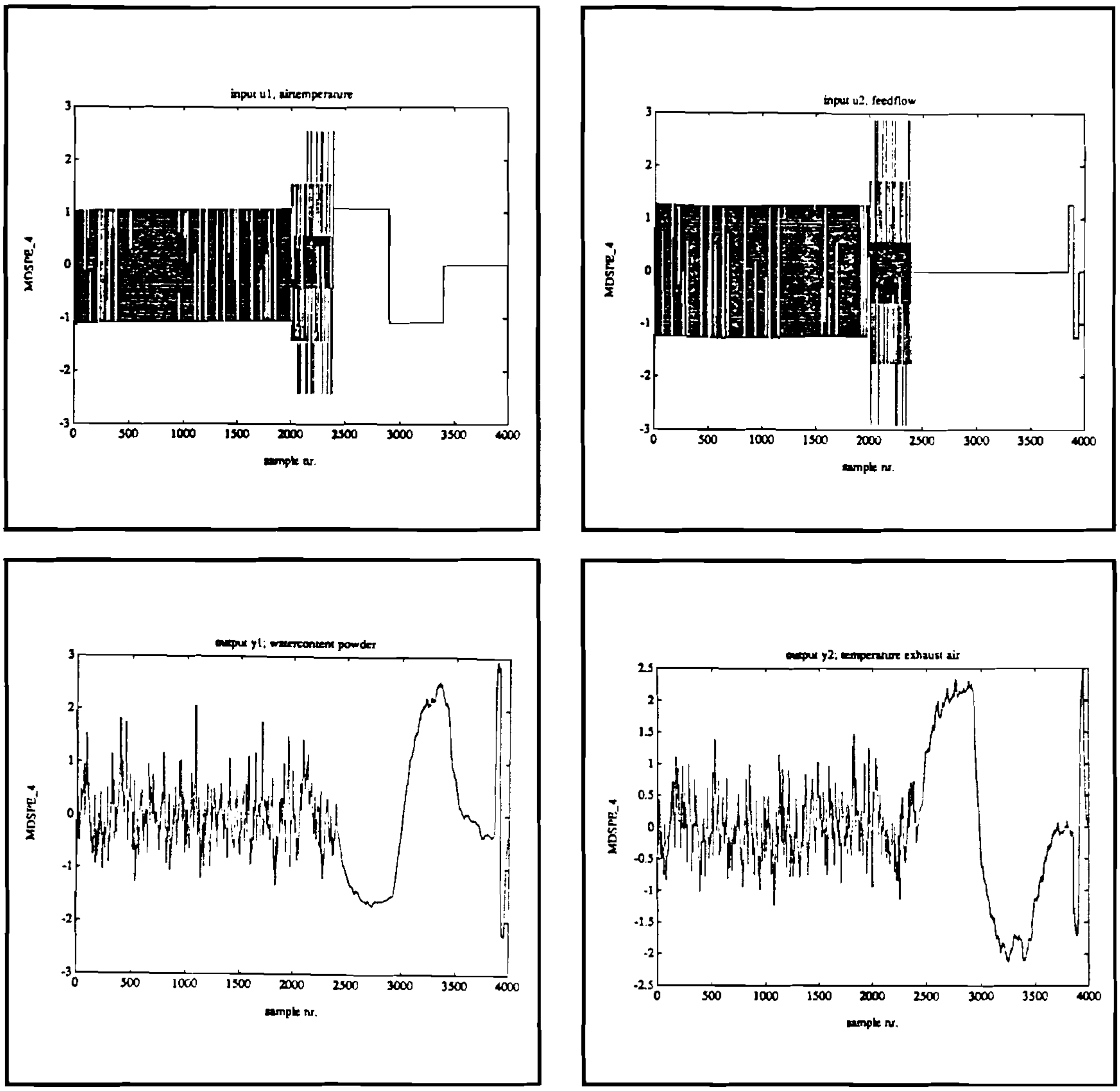


figure 6.12: *input and output signals for experiment MDSPE_4.*

6.6 ARX experiment results.

The time constants of input u1 (air temperature) to both outputs are about 500 seconds. With a sampling time of 5 seconds and 100 estimated FIR parameters the whole impulse response length cannot be covered. Although the FIR model is only used as an initial model for the latter direct estimation the question arises whether this initial model is good enough. In experiment MDSPE_4 of chapter 6.5 this was not the case. So with this data set of section 6.5 first an ARX model of high order (n = 30) is fitted on the data set; this yields two MISO ARX models with infinite impulse response. This explains the difference with the FIR estimation which has finite impulse response. See section 6.5.
Both models are transformed into two MISO state space models according to the technique described in chapter 3.8. Then a MIMO state space representation is made. This MIMO representation is reduced with a Schur model reduction method (appendix C). Validation on white and coloured data sets are given in table 6.8 and 6.9.
Rather impressive are the step responses of this model. Step responses of all interesting models are given in appendix B. Because the step responses are incorporated in the data set, the ARX based reduced order state space model gives the best response in comparison with the process output.

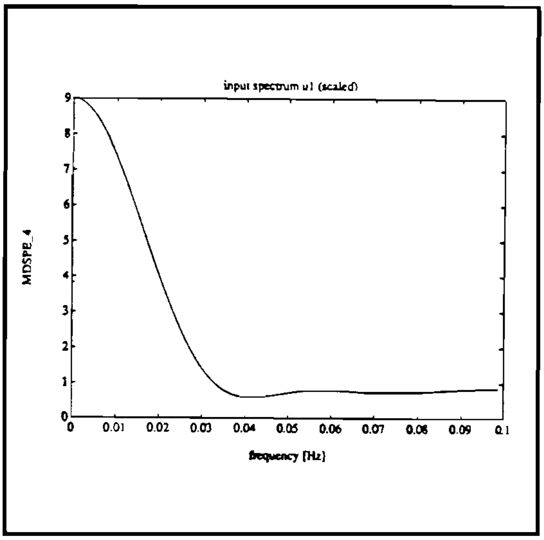
	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-10.3	-8.4
model with noise	12	2000	-9.6	-8.1
ARX-SS model	12	4000	-10.1	-7.9

table 6.8: *model validation on white PRBNS input signals.*

	order	N	J_{oe1}	J_{oe2}
noise free model	12	4000	-11.4	-12.0
model with noise	12	2000	-10.8	-8.8
ARX model	12	2000	-10.6	-12.3

table 6.9: *model validation on coloured noise input signals.*

The input spectra of this data set of figure 6.12 are given in figure 6.13. These spectra look very much like the input spectra of the coloured noise experiment (MDSPE_2). This results in a validation which is nearly the same for both models (MDSPE_2 and ARX) but in a better estimated step response for the ARX based state space model (see appendix B). This shows that input design is very important in parameter estimation and this can greatly improve the estimated models. The ARX estimation method is written in Matlab files which are given in appendix C.



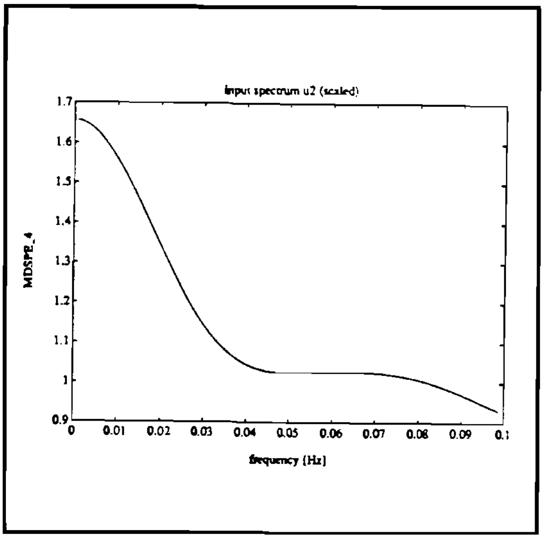


figure 6.13: *input spectra for the ARX data set of figure 6.12.*

6.7 Results: an overview.

The state space models are given in appendix A.
The step responses of various interesting models are given in appendix B.

			white noise		coloured noise	
model	order	N	J_{oe1}	J_{oe2}	J_{oe1}	J_{oe2}
noise free model	12	4000	-10.3	-8.4	-11.4	-12.0
model with noise	12	2000	-9.6	-8.1	-10.8	-8.8
MDSPE_1 model	16	2400	-10.0	-7.7	-11.5	-7.8
coloured noise model	12	2000	-9.8	-7.5	-12.2	-13.1
MDSPE_2 model	12	4000	-9.9	-7.9	-11.7	-13.1
MDSPE_2 Hamming windowed	12	4000	-9.9	-7.7	-12.0	-13.5
MDSPE_3 model	12	4000	-7.9	-5.5	-3.7	-5.6
ARX-SS model	12	4000	-10.1	-7.9	-10.6	-12.3

table 6.10: *all validation results of the MDSPE experiments (high order ss models).*

			white noise		coloured noise	
model	order	N	J_{oe1}	J_{oe2}	J_{oe1}	J_{oe2}
noise free model	6	4000	-10.3	-8.9	-11.3	-12.0
model with noise	6	2000	-9.6	-8.1	-10.9	-9.0
MDSPE_1 model	6	2400	-9.6	-7.6	-11.2	-8.1
coloured noise model	6	2000	-10.0	-7.5	-12.1	-12.8
MDSPE_2 model	6	4000	-10.0	-8.0	-11.7	-13.0

table 6.11: *all validation results of the MDSPE experiments (low order ss models).*

6.7 conclusions.

From the experiment results the following conclusions may be derived:

Positive conclusions:

- Adding the modified fast PRBNS data set to the standard final PRBNS data set reduces the modelling error at higher frequencies in case of the spray dryer process.

- Adding a data set with low pass filtered PRBNS input signals reduces the modelling error at low frequencies.

- Hamming windowing is not necessary.

- The estimated model from the coloured noise data set and the standard final PRBNS data set reduces the error signal power at output y1 with 27% and at output y2 with 63% compared to the standard model (validation on coloured noise) and gives a normal (equal) result (compared to the standard model) validating on white noise.

- Estimating a model on all data sets (final PRBNS, modified PRBNS and stepresponses provides a very good model for all frequencies and especially for step responses without doing DC-correction. This can be done with the developed ARX model estimation and reduction to a low order state space model with the tools of appendix C.

Negative conclusions:

- The standard identification procedure [1] implemented in the program IPCOS-Shell does not work well on this process with strongly varying time constants and on data sets with step responses incorporated. As an alternative the ARX based state space model estimation of appendix C is developed. One could research if this conclusion holds for all processes with strongly varying time constants.

The standard model estimated from the final PRBNS experiment and the ARX model are now used for H-infinity controller design. The ARX model based controller is expected to perform better on the frequencies were the modelling error of this model is lower. Controller design for the milk spray dryer tower with H-infinity technique is studied by Pieter van Gelder as a parallel Master Science project at Datex IPCOS group. More information about the H-infinity controller design for the spray dryer process given in chapter 7 can be found in his Master Science thesis [12].

7. Controller design

7.1 Introduction.

In order to test the standard model (spraymet) and the newly estimated ARX based model which performs better for all frequencies we estimated controllers for both models. Controllers are designed with the standard IPCOS procedure (IMC control: chapter 7.2) and with the H-infinity technique. This chapter will briefly include the results of a parallel research project by Pieter van Gelder [12].
Conclusions can be made about the models also based upon the performance of the controllers.

7.2 IMC control.

For both models a controller was designed with the standard IMC procedure of IPCOS shell. The schedule of IMC control is given in figure 7.1, in which:

- G is the actual process
- G_0 is the estimated low order state space model
- C is the designed controller
- r is the setpoint
- y are the process outputs
- e is the error signal
- u are the process inputs given by the controller
- d are disturbances (modelled as output noise here)

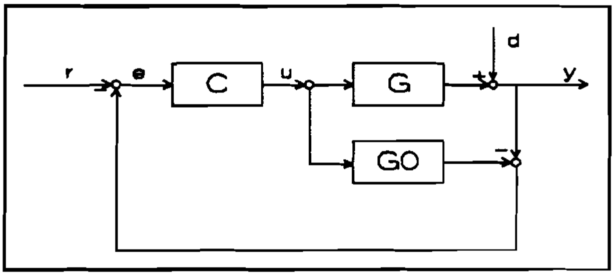


figure 7.1: *IMC schedule.*

The IMC controller is a normal state feedback controller. IPCOS shell provides the state feedback matrix F and the static decoupling matrix Hi. The state feedback matrix F is a L_2 minimization of the difference between the desired transfer function and the controlled system transfer function (a form of pole placement). The IMC schedule of figure 7.1 however is **not** a state feedback scheme. Controller C consists of two parts:

$$C = C1*C2$$

in which C1 is a lowpass filter for stability
in which C2 is the actual controller given by:

$$C2 = \left| \begin{array}{ll} A-BF & BHi \\ -F & Hi \end{array} \right|$$

For details on the IMC controller design I refer to [12].
For the standard model (DC corrected) of order 10 and for the ARX based model of order 12 an IMC controller is designed. Poles of the controller for both outputs of the desired transfer function are 0.9. These are chosen this slow because decreasing their value gives an unstable controlled system.
For both these controllers and models three experiments have been performed with the scheme in figure 7.2.

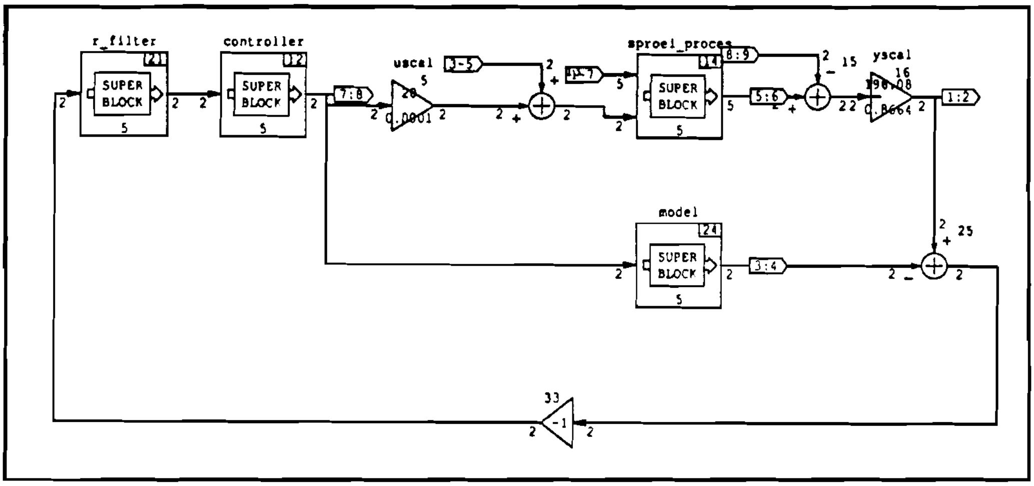


Figure 7.2: *IMC controller scheme in MatrixX.*

In figure 7.2 the outputs y1 and y2 are given by 5:6. A new setpoint can be given by entering the new setpoint to input 8:9.
First let us repeat the main goals for the controlled system:

-1- static decoupling of inputs and outputs
-2- fast changeovers for output y1
-3- output y2 must be held below 70°C but as high as possible
-4- disturbance rejection

The IMC controller according to IPCOS shell provides full decoupling. Therefore compromises must be made to the other goals. The controlled system cannot be made as fast as we wanted because of stability reasons.
To test the controlled system with the different models experiments are performed:

-1- a step response experiment in which the output y1 (watercontent powder) changes from 8.08% to 9%. This gives us the changeover speed.
-2- a step response experiment in which the output y2 (temperature of the exhaust air) is changed from 65.1°C to 68°C. This gives the changeover speed for output y2.

-3- an experiment in which both setpoints are held constant. This gives the disturbance rejection.

Note that all experiments are performed with additional noise **which is the same for all experiments.**
Expectations towards these experiments are on forehand:

-1- a better disturbance rejection of the controlled system than for the openloop system
-2- faster setpoint changes of the controlled system that for the openloop system
-3- decoupling
-4- better performance for point 1 to 3 for the ARX based model; a model closer to the actual process would then give a better controlled system.

Results of the controlled system for both models and for the described experiments are given in section 7.3.

7.3 Experiment results of the IMC controlled system.

First the IMC controlled system with the standard model is considered. Three experiments according to section 7.2 are performed. Results are given in figures 7.3, 7.4 and 7.5.

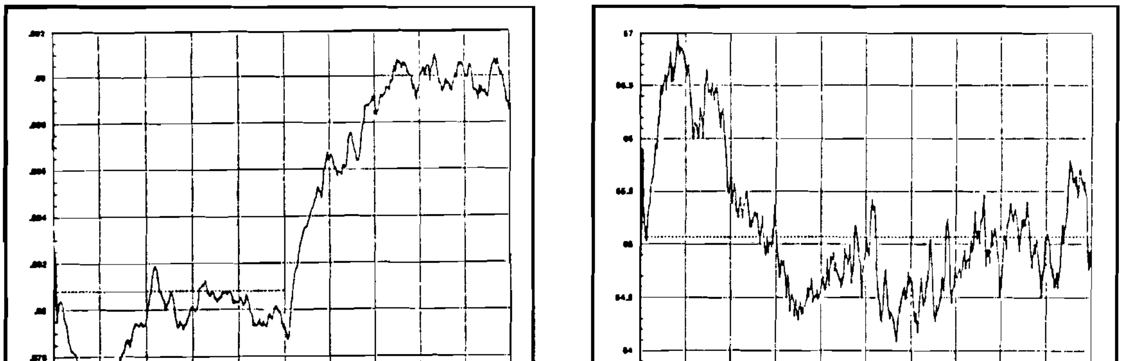


figure 7.3: *Step on output y1 and reaction on both outputs (standard model).*

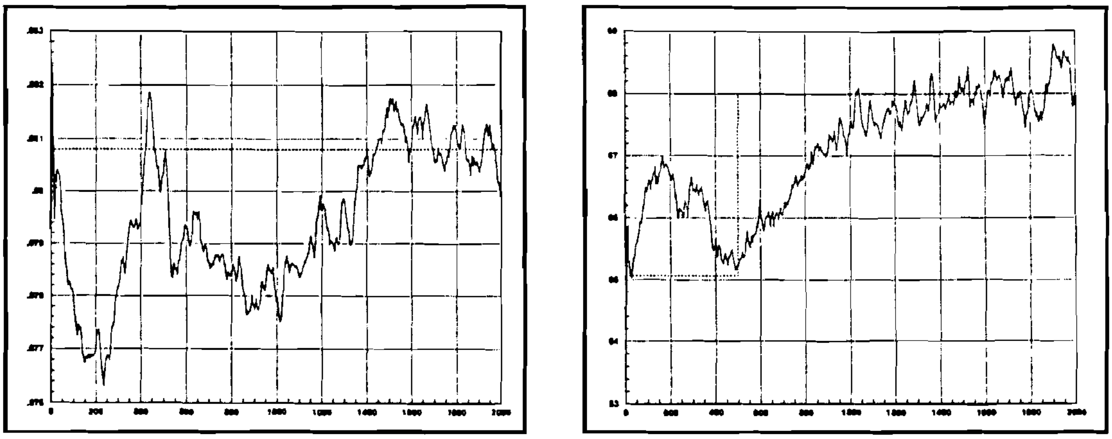


figure 7.4: *Step on output y2 and reaction on both outputs (standard model).*

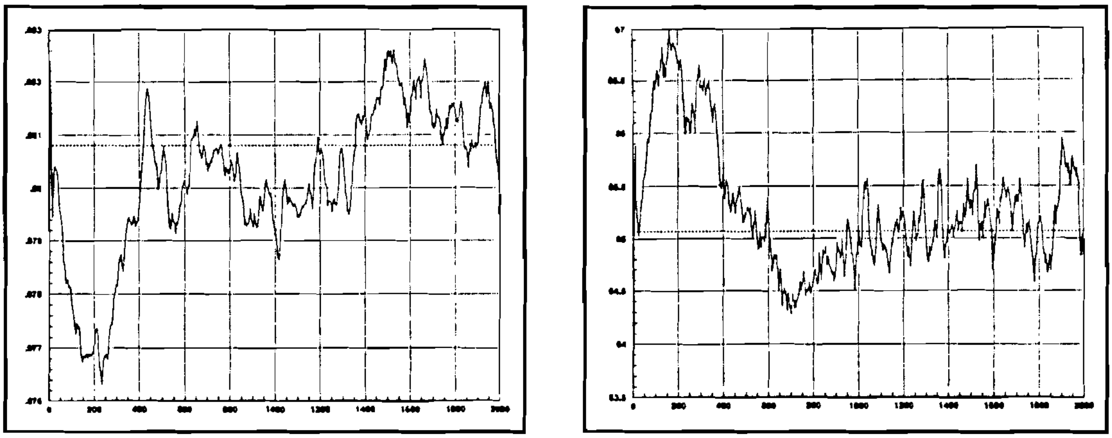


figure 7.5: *Output disturbances with 'IPCOS' controller (standard model).*

From figure 7.3 and 7.4 we may conclude that the step responses are not significantly faster than the openloop process step responses of appendix B. Decoupling however has been established. The disturbance rejection with controller of figure 7.5 can be compared with the openloop process disturbances with the same input signals; these outputs are given in figure 7.6. All information with respect to the disturbance variances and extremes are given in table 7.1.

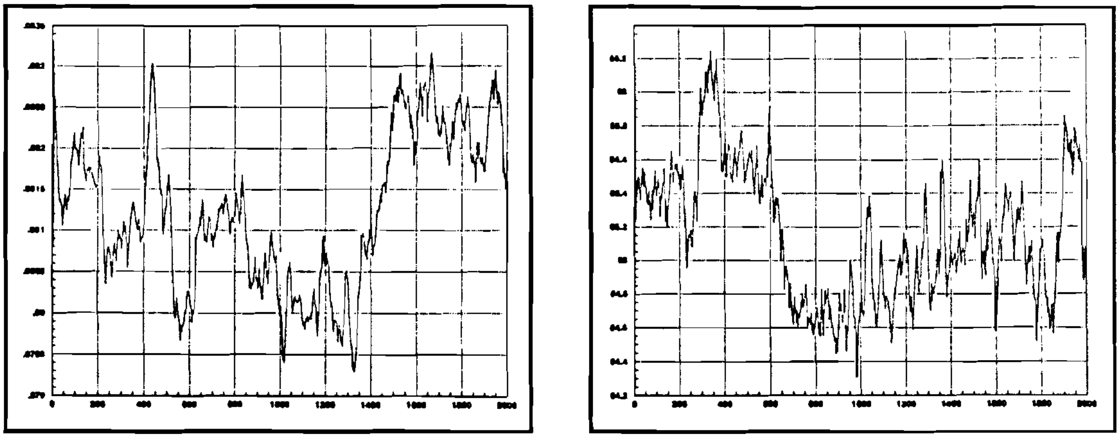


figure 7.6: *Openloop process output disturbances.*

Of course the same three experiments of section 7.2 were performed for the ARX based state space model on which a 'IPCOS' controller is designed. The controller was designed with the same parameters as for the standard model controller design. Results for these experiments are given in figure 7.7 to 7.9 and in table 7.1.

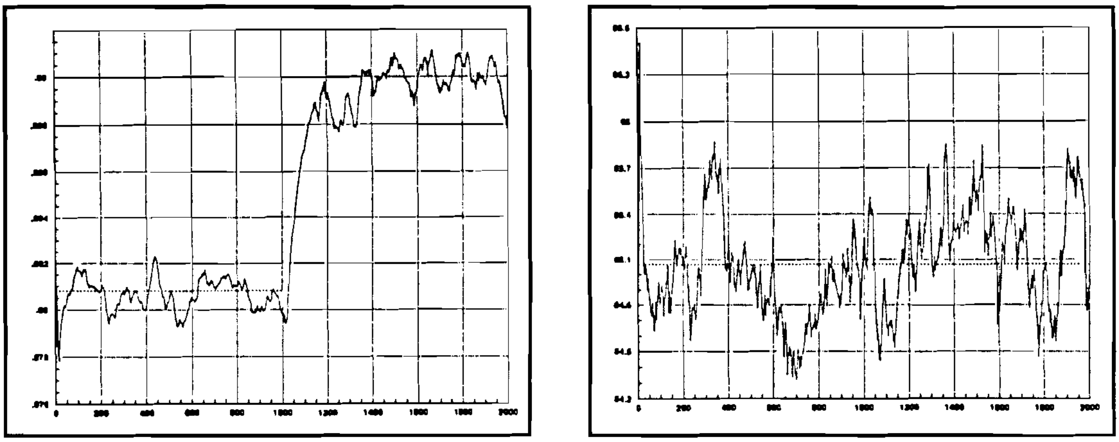


figure 7.7: *Step on output y1 and reaction on both outputs (ARX based model).*

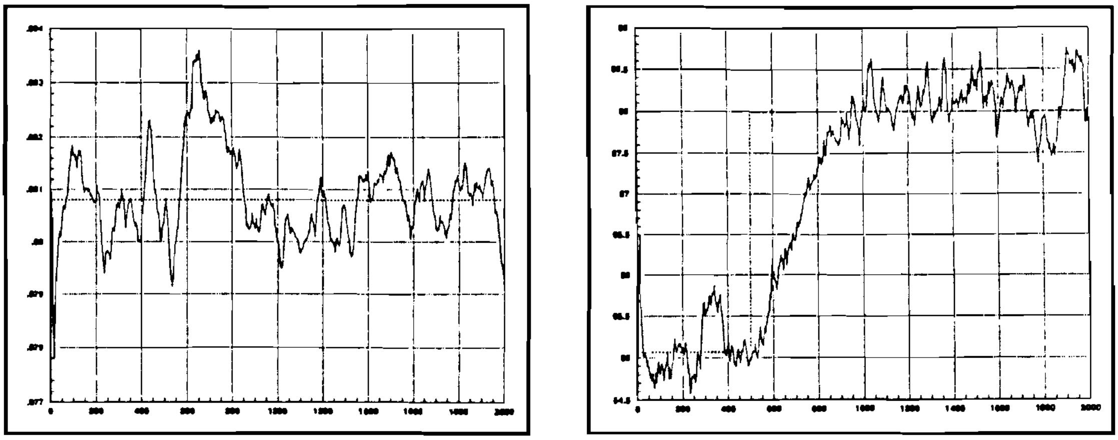


figure 7.8: *Step on output y2 and reaction on both outputs (ARX based model).*

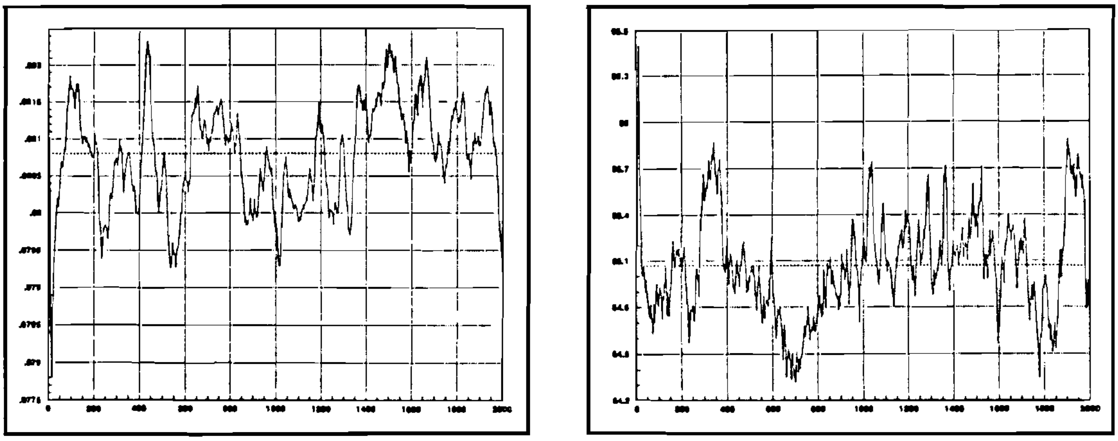


figure 7.9: *Output disturbances with 'IPCOS' controller (ARX based model).*

	var y1	var y2	max y1	max y2	min y1	min y2
openloop process	1.0E-6	0.1081	0.0832	65.887	0.0793	64.307
standard model	8.2E-7	0.1080	0.0826	65.948	0.0780	64.283
ARX model	4.5E-7	0.1018	0.0823	65.890	0.0792	64.319

table 7.1: *Noise characteristics of the openloop and controlled system (IMC control).*

When comparing the controlled step responses for both models one can see that the ARX based model with controller is much faster without giving up decoupling. This supports the theory that a better process model leads to a better controlled system. Also follows from table 7.1 that the disturbance rejection is much better for the ARX based model.

7.4 H-infinity control.

For both models a controller was designed with the H-infinity procedure. The H-infinity controller can be calculated as C in the schedule of IMC control (figure 7.1). For details on the controller design I refer to [12].
Only results of the experiments of section 7.2 are given in section 7.5.

7.5 Experiment results of the H-infinity controlled system.

First the H-infinity controlled system with the standard model is considered. Three experiments according to section 7.2 are performed. Results are given in figures 7.10, 7.11 and 7.12.

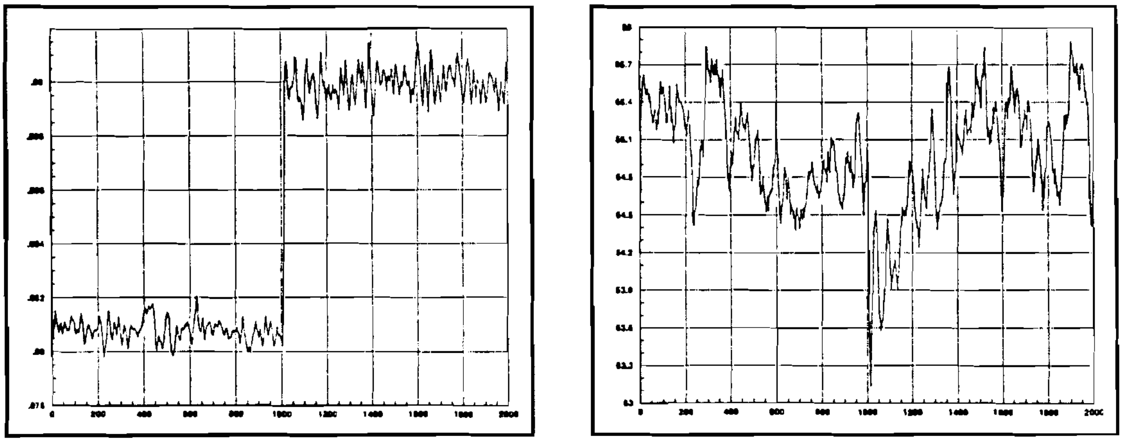


figure 7.10: *Step on output y1 and reaction on both outputs (standard model).*

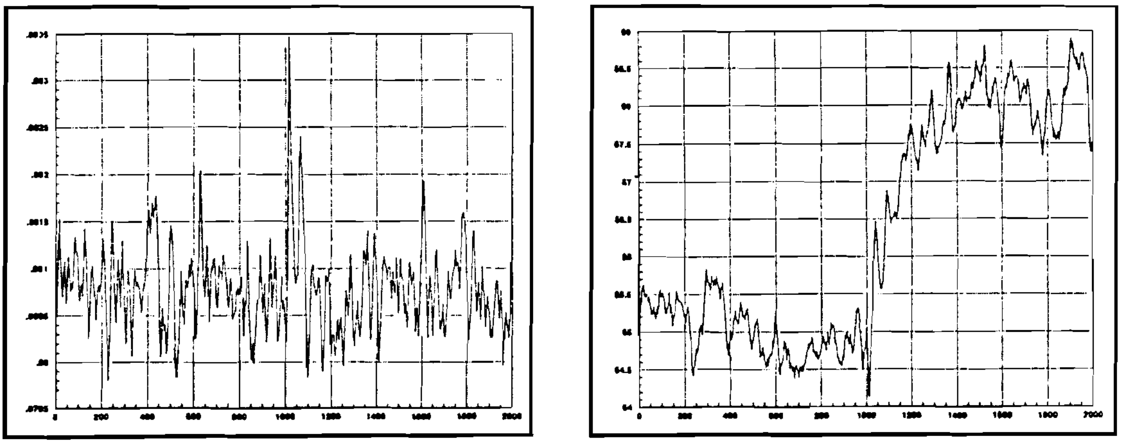


figure 7.11: *Step on output y2 and reaction on both outputs (standard model).*

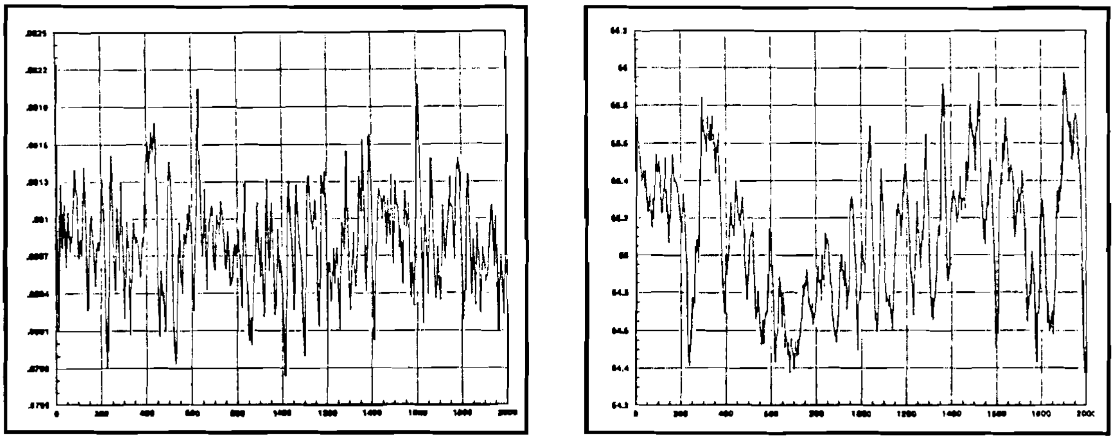


figure 7.12: *Output disturbances with H-infinity controller (standard model).*

From figure 7.10 and 7.11 we may conclude that the step responses are significantly faster than the openloop process step responses of appendix B and also of the IPCOS controller on both models. Decoupling has been established. The disturbance rejection with controller of figure 7.12 can be compared with the openloop process disturbances with the same input signals; these outputs are given in figure 7.6. All information with respect to the disturbances is given in table 7.2.

Of course the same three experiments of section 7.2 were performed for the ARX based state space model on which a H-infinity controller is designed. The controller was designed with the same parameters as for the standard model controller design. Results for these experiments are given in figure 7.13 to 7.15 and in table 7.2.

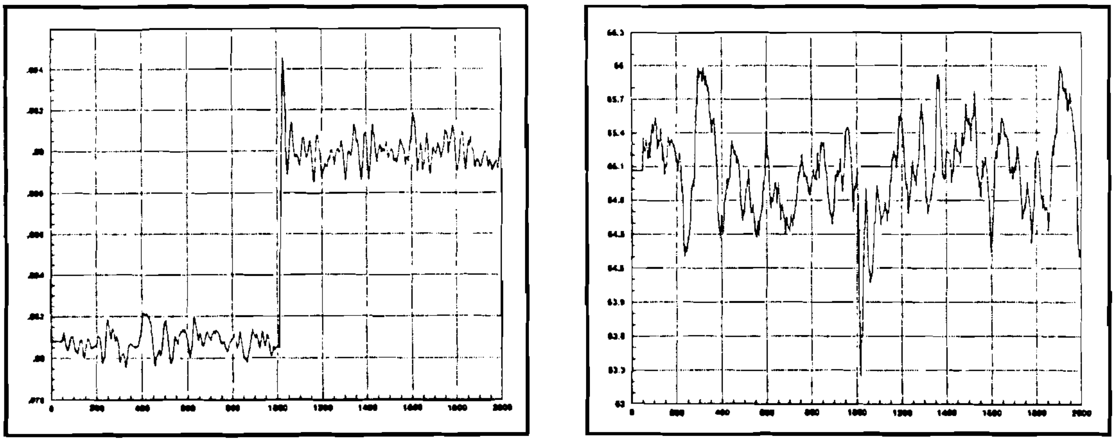


figure 7.13: *Step on output y1 and reaction on both outputs (ARX based model).*

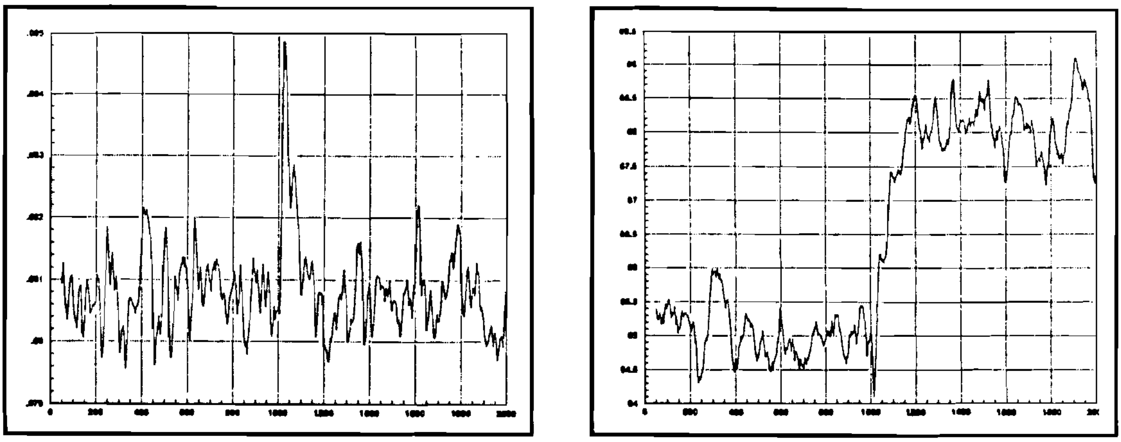


figure 7.14: *Step on output y2 and reaction on both outputs (ARX based model).*

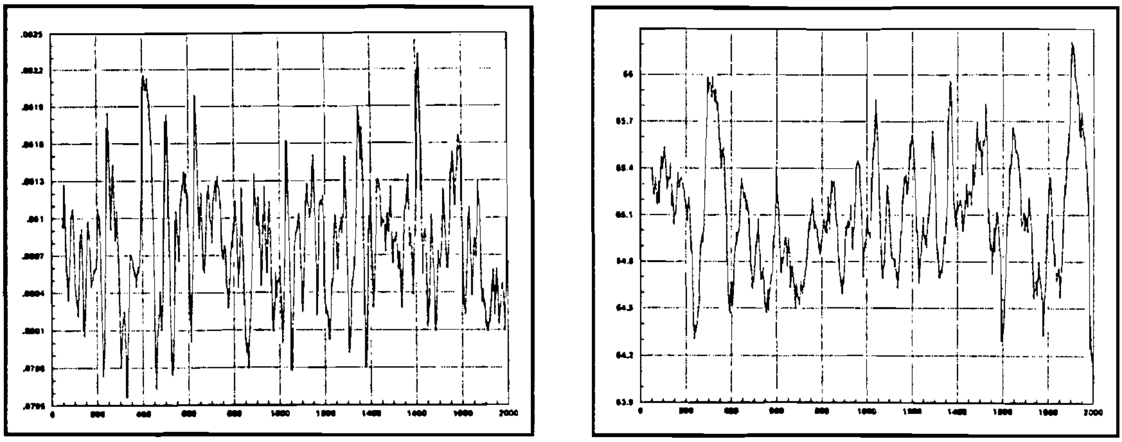


figure 7.15: *Output disturbances with H-inf. controller (ARX based model).*

	var y1	var y2	max y1	max y2	min y1	min y2
openloop process	1.0E-6	0.1081	0.0832	65.887	0.0793	64.307
standard model	1.4E-7	0.1361	0.0821	65.976	0.0797	64.373
ARX model	2.1E-7	0.1336	0.0823	66.220	0.0797	64.103

table 7.2: *Noise characteristics of the openloop and controlled system (H-inf. control).*

	var y1	var y2	max y1	max y2	min y1	min y2
openloop process	1.0E-6	0.1081	0.0832	65.887	0.0793	64.307
standard model IPCOS	8.2E-7	0.1080	0.0826	65.948	0.0780	64.283
ARX model IPCOS	4.5E-7	0.1018	0.0823	65.890	0.0792	64.319
standard model H-inf	1.4E-7	0.1361	0.0821	65.976	0.0797	64.373
ARX model H-inf	2.1E-7	0.1336	0.0823	66.220	0.0797	64.103

table 7.3: *Noise characteristics of the IPCOS and H-inf. controlled system.*

When comparing the controlled step responses for both models one can see that

there is not much difference in step response speed between both the controlled models, allthough the ARX based model is closer to the actual process. This implies that the used H-infinity technique is robust against modelling errors.
Another result is the great improvement in change-over speed with respect to the 'IPCOS' controllers on both models. Also the noise variance has been reduced by the H-infinity technique.
In practice the high change-over speed can be a drawback if the actuators cannot follow the fast steering signals of the controller.

7.6 Conclusions towards the controlled system.

From the controlled system experiments of the spray dryer process on both models we may derive the following conclusions:

The controlled system is improved with respect to change-over time (step response time), disturbance rejection and decoupling in comparison with the openloop process.

A better process model leads to a better controlled system with respect to change-over time and disturbance rejection when applying the 'IPCOS' IMC pole placement controllers with the same design parameters. For the main output y1 the noise variance is reduced by about 40% (ARX based model controlled system versus standard model controlled system). Change-over speed reduced from 2500 to 1000 seconds.

H-infinity control leads to faster change-over than the 'IPCOS' controllers without giving up stability and decoupling on the simulation process. In practice this might give problems with the actuator speed. The noise variance was reduced by about 80% in comparison with the openloop process.

H-infinity control can give robust control even when the model used for designing the controller has some modelling error.

8. Conclusions

Conclusions based on the MDSPE and ARX experiment results:

-1- standard identification of a milk spray dryer process does not provide a very good model for all frequencies and step responses of the model are poor (even with DC-correction).

-2- quite an improvement in estimating this process model can be made by specifying better input signals with aid of the model error upperbounds.

-3- a model for this process which fits for all frequencies and gives very good step responses can be made with all data sets which are measured in a standard identification procedure with aid of the ARX based state space model estimation which is developed in this thesis.

-4- DC correction is not necessary when using step responses as a part of the estimation data set.

Recommendations towards the program IPCOS shell:

-1- options can be included in the shell for using other than the final PRBNS data set for estimation of the process model. The appropriate scaling of the different data sets towards their (possible different) variance must in that case be considered. Hamming windowing is not necessary.

-2- automatic use of the fast PRBNS data set must be considered, because this inevitably reduces the model error at high frequencies.

-3- I recommend the use of ARX based state space model estimation with all data sets coupled for use on processes with strongly varying time constants. Processes with large and small time constants can be better described by an ARX model than by a FIR model because an ARX model has an infinite impulse response. A FIR model should give the same result when applying a large amount of parameters (and also a large number of data samples) but in practice this will not work.

Recommendations towards model estimation (by IPCOS group):

-1- the use of more than one data set (final PRBNS, fast PRBNS and stepresponses) can considerably improve model estimation at various frequencies.

-2- the use of coloured noise data sets (with input signals shaped by the process transfer function, the three sigma bound modelling error after the final PRBNS experiment or the frequency content of the controller steering signal) in combination with the final PRBNS data set afterwards can give an impressive model improvement at the frequency area where the modelling error is high. At other (high) frequency areas the modelling error remains about the same when using the final PRBNS data set.

-3- the ARX based state space model estimation works well on processes with strongly varying time constants. Results are strongly improved when adding step responses to the estimation data set. DC-correction is not necessary; only some model reduction technique. Besides this the algorithm is very fast (5 times faster compared to the IPCOS-shell).

Conclusions based on the controlled system experiment results:

-1- a better process model gives a better controlled system when applying the 'IPCOS' IMC controller design with the same design parameters for both models.

-2- H-infinity control gives a better controlled system than the controlled system controlled by the 'IPCOS' controller. Improvements are made on change-over speed and noise reduction.

Further research subjects:

-1- does the ARX based state space model estimation provide better models for all processes with strongly varying time constants? And on other processes with 'normal' time constants?

-2- can the ARX model be used as an initial model just like the FIR model in IPCOS shell?

-3- what is the optimal data set composed of the three data sets used? Can the optimum length be specified for each data set for use within the final estimation data set?

References

[1] Identification of industrial MIMO processes for fixed controllers.
T. Backx and A. Damen
Journal A, vol. 30, no. 1, 1989

[2] System identification.
T. Soderstrom and P. Stoica
Prentice Hall 1989

[3] Estimation of transfer functions: asymptotic theory and a bound of model uncertainty.
Y.C. Zhu
Int. J. Control, vol. 49, no. 6, 1989

[4] Black-box identification of MIMO transfer functions: asymptotic properties of prediction error models.
Y.C. Zhu
Int. J. of adaptive control and signal processing, vol. 3, 1989

[5] On the bounds of the modelling errors of black-box MIMO transfer function estimates.
Y.C. Zhu
EUT report 87-E-183, Eindhoven university of Technology, the Netherlands, november 1987

[6] System identification.
L. Ljung
Prentice Hall 1987

[7] Demonstration of system identification and advanced multivariable control design on a physical model of a spray dryer.
M. van Wijck and T. Backx
Datex IPCOS group (to appear)

[8] Identification of an industrial process: a Markov parameter approach.
T. Backx
Ph.D. Thesis, Dept. E, Eindhoven University of Technology, the Netherlands, 1988

[9] Identification and control of MIMO industrial processes: an integration approach.
Y.C. Zhu
Ph.D. Thesis, Dept. E, Eindhoven University of Technology, the Netherlands, 1990

[10] System identification.
P. Eykhoff
Wiley & sons 1974

[11] Multivariable process identification for robust control
Y.C. Zhu, A.C.P.M. Backx, P. Eykhoff
Datex IPCOS group
Dept. EE, Eindhoven University of Technology
(to appear)

[12] H-infinity controller design for a spray dryer
Pieter van Gelder
Master Science Thesis
Dept. Mathematics and Information Sciences
Eindhoven University of Technology
1991 (to appear)

standard model without noise

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
-0.2244	1.5921	-4.9257	8.4266	-8.3198	4.4511	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	-0.2244

Columns 8 through 12

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1.0000	0	0	0	0
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000
1.5921	-4.9257	8.4266	-8.3198	4.4511

b =

0.0042	-0.0115
0.0009	0.0304
0.0021	-0.0224
0.0024	-0.0278
0.0015	0.0357
0.0005	0.1347
-0.0029	0.0118
-0.0089	-0.0411
-0.0047	-0.1926
-0.0018	-0.3172
-0.0029	-0.3736
-0.0061	-0.3715

c =

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

d =

0.0032	-0.0010
-0.0047	-0.0011

standard model with noise

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
-0.2689	1.8948	-5.7490	9.5494	-9.0858	4.6594	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	-0.2689

Columns 8 through 12

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1.0000	0	0	0	0
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000
1.8948	-5.7490	9.5494	-9.0858	4.6594

b =

0.0005	-0.0116
0.0014	0.0275
0.0020	-0.0196
0.0030	-0.0214
0.0041	0.0407
0.0051	0.1365
-0.0113	0.0189
-0.0201	-0.0525
-0.0180	-0.1896
-0.0167	-0.3032
-0.0190	-0.3608
-0.0231	-0.3665

c =

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

d =

-0.0020	-0.0031
-0.0123	-0.0043

MDSPE_1 model

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
0	0	0	0	0	0	1.0000
0	0	0	0	0	0	0
-0.0148	0.1796	-0.8806	2.4482	-4.6679	6.6374	-6.6269
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Columns 8 through 14

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1.0000	0	0	0	0	0	0
3.9247	0	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
0	0	0	0	0	0	1.0000
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	-0.0148	0.1796	-0.8806	2.4482	-4.6679	6.6374

Columns 15 through 16

0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1.0000	0
0	1.0000
-6.6269	3.9247

b=

0.0013	-0.0082
0.0001	-0.0074
0.0050	-0.0039
0.0055	-0.0064
0.0097	0.0073
0.0082	0.1140
0.0037	0.2314
0.0027	0.2861
-0.0256	0.0062
-0.0265	-0.0085
-0.0251	-0.2220
-0.0334	-0.2951
-0.0289	-0.3251
-0.0313	-0.3304
-0.0403	-0.3032
-0.0430	-0.2645

c =

Columns 1 through 12

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0

Columns 13 through 16

0	0	0	0
0	0	0	0

d =

-0.0010	-0.0080
-0.0294	0.0041

coloured noise model

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
-0.5512	3.5415	-9.6028	14.0590	11.7189	5.2724	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	-0.5512

Columns 8 through 12

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1.0000	0	0	0	0
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000
3.5415	-9.6028	14.0590	-11.7189	5.2724

b =

-0.0144	0.0007
-0.0043	-0.0191
0.0019	-0.0175
0.0046	-0.0011
0.0044	0.0241
0.0023	0.0524
-0.0158	-0.0024
0.0020	-0.0497
0.0115	-0.0881
0.0148	-0.1155
0.0138	-0.1317
0.0104	-0.1376

c =

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

d =

-0.0048	-0.0075
0.0003	-0.0071

MDSPE_2 model

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
-0.4074	2.7275	-7.7502	11.9497	-10.5216	5.0020	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	-0.4074

Columns 8 through 12

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1.0000	0	0	0	0
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000
2.7275	-7.7502	11.9497	-10.5216	5.0020

b=

-0.0101	-0.0043
0.0032	-0.0087
0.0031	-0.0156
-0.0021	-0.0050
-0.0071	0.0254
-0.0093	0.0682
-0.0004	0.0052
0.0032	-0.0524
0.0047	-0.1065
0.0044	-0.1475
0.0033	-0.1711
0.0021	-0.1776

c =

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

d =

-0.0041	-0.0092
0.0003	-0.0136

MDSPE_2 model Hamming windowed

a =

Columns 1 through 7

0	1.0000	0	0	0	0	0
0	0	1.0000	0	0	0	0
0	0	0	1.0000	0	0	0
0	0	0	0	1.0000	0	0
0	0	0	0	0	1.0000	0
-0.5568	3.6221	-9.8908	14.5020	-12.0357	5.3592	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	-0.5568

Columns 8 through 12

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1.0000	0	0	0	0
0	1.0000	0	0	0
0	0	1.0000	0	0
0	0	0	1.0000	0
0	0	0	0	1.0000
3.6221	-9.8908	14.5020	-12.0357	5.3592

b=

0.0040	0.0118
-0.0003	-0.0261
-0.0041	-0.0237
-0.0071	0.0031
-0.0089	0.0418
-0.0095	0.0829
-0.0006	0.0048
0.0014	-0.0622
0.0012	-0.1109
0.0002	-0.1422
-0.0010	-0.1584
-0.0017	-0.1626

c =

1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0

d =

0.0065	-0.0192
-0.0010	-0.0112

ARX model based state space model of order 12.

a =

Columns 1 through 7

0.9766	0.0330	-0.0649	0.0000	-0.0099	0.0113	-0.0063
-0.0051	0.9802	0.1047	0.0009	0.0093	-0.0143	0.0066
0.0707	-0.0925	0.8448	0.1973	-0.0394	0.0402	-0.0241
0.0330	-0.0177	-0.2674	0.6302	-0.0540	-0.3979	-0.0758
0.0063	0.0103	-0.0013	0.0131	0.9484	-0.0127	-0.0914
-0.0144	0.0165	0.0595	0.4187	-0.0472	0.4819	-0.0804
-0.0040	0.0039	0.0271	0.0609	0.0046	-0.2663	0.8442
0.0138	0.0058	-0.0491	-0.1407	-0.1032	0.3058	0.1122
-0.0039	0.0057	0.0302	0.0186	-0.0646	-0.3342	-0.3936
0.0004	0.0031	-0.0028	0.0252	-0.0267	0.0661	-0.0008
0.0005	0.0044	0.0103	0.0120	-0.0055	-0.0928	0.0522
0.0007	0.0036	-0.0015	0.0026	-0.0463	0.0101	-0.0781

Columns 8 through 12

0.0050	-0.0152	-0.0049	-0.0003	0.0009
-0.0058	0.0166	0.0050	0.0004	-0.0011
0.0077	-0.0756	-0.0135	-0.0100	0.0076
0.0607	0.0483	-0.0415	0.0723	-0.0053
0.1208	-0.0914	-0.0424	0.0328	0.0138
0.3081	0.2485	-0.1065	0.1617	-0.0505
0.3247	-0.1859	-0.0514	-0.0104	-0.0067
0.5480	0.1339	0.2061	0.0132	0.1198
0.4744	-0.1138	0.3951	-0.3523	0.0199
-0.1257	-0.5131	0.5472	0.5735	-0.0761
-0.1736	0.3485	0.3144	0.1846	0.6907
0.0268	-0.3473	-0.4385	-0.0084	0.6383

b =

0.1114	-0.4077
0.2131	0.1597
0.0463	0.5226
0.0439	0.3743
-0.1253	0.0244
-0.0702	-0.1638
-0.0207	-0.0500
-0.1100	0.1151
-0.0349	-0.0521
-0.0258	-0.0010
-0.0451	-0.0103
-0.0312	0.0004

c =

Columns 1 through 7

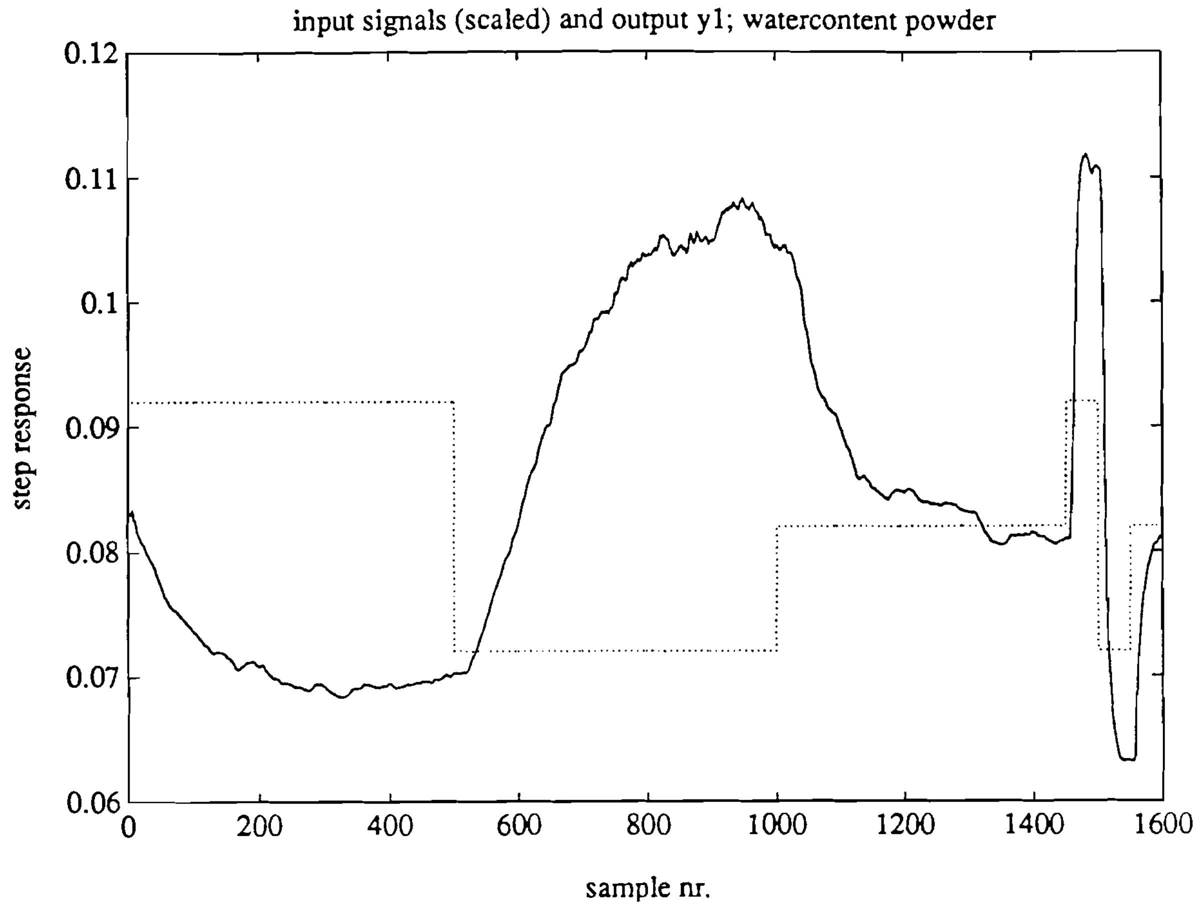
-0.1469	0.0978	-0.2418	0.1718	-0.0095	0.0893	0.0046
0.1526	-0.0881	0.0799	0.1260	0.0591	0.0179	0.0502

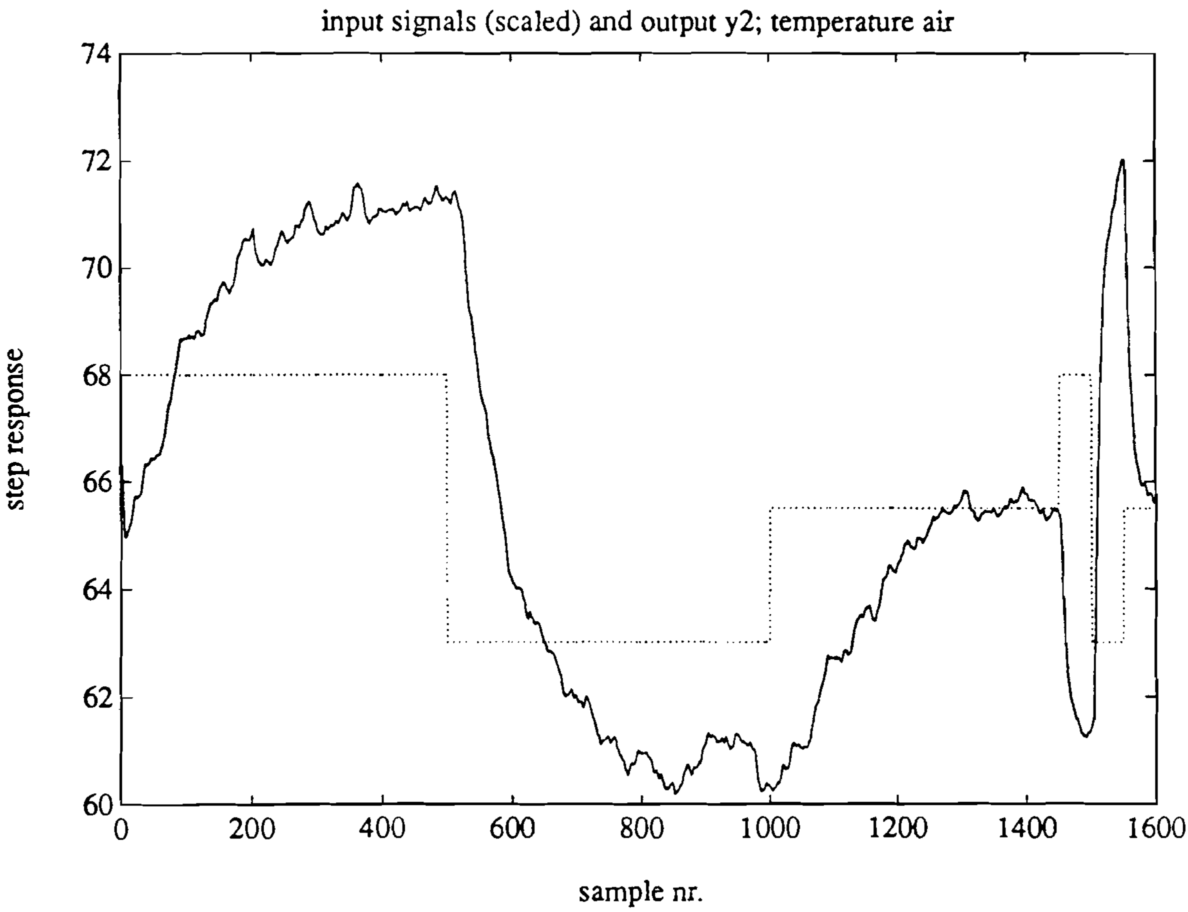
Columns 8 through 12

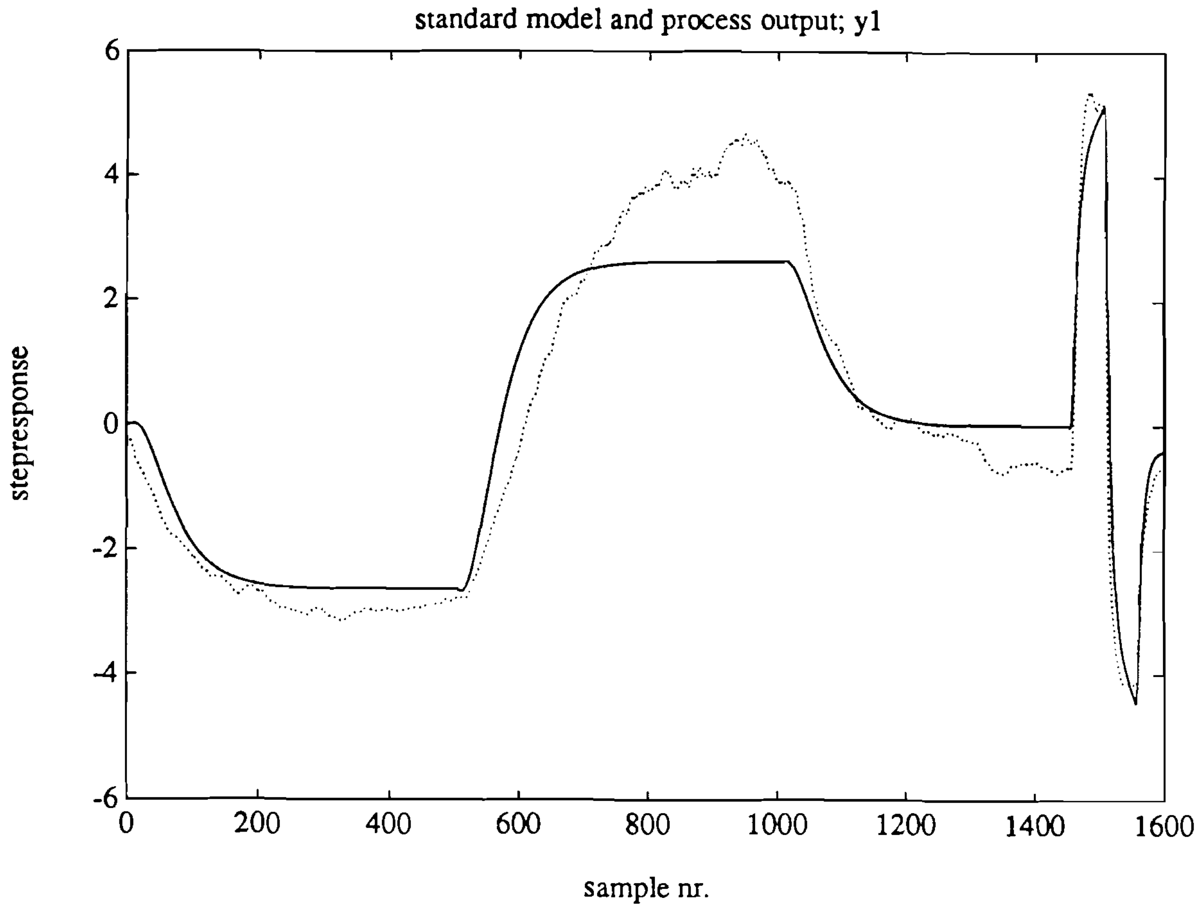
-0.0157	-0.0544	0.0054	-0.0252	0.0069
-0.0493	0.0555	0.0388	-0.0225	-0.0007

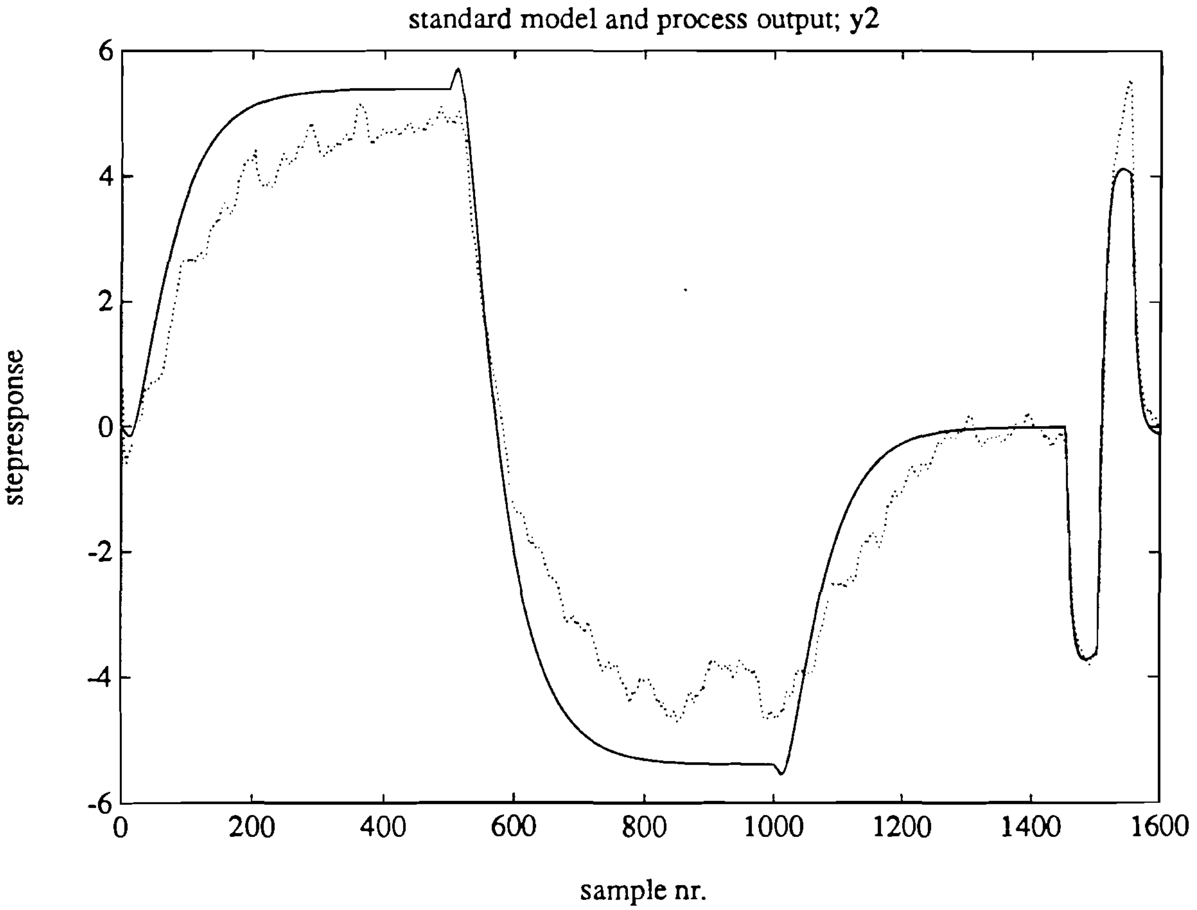
d =

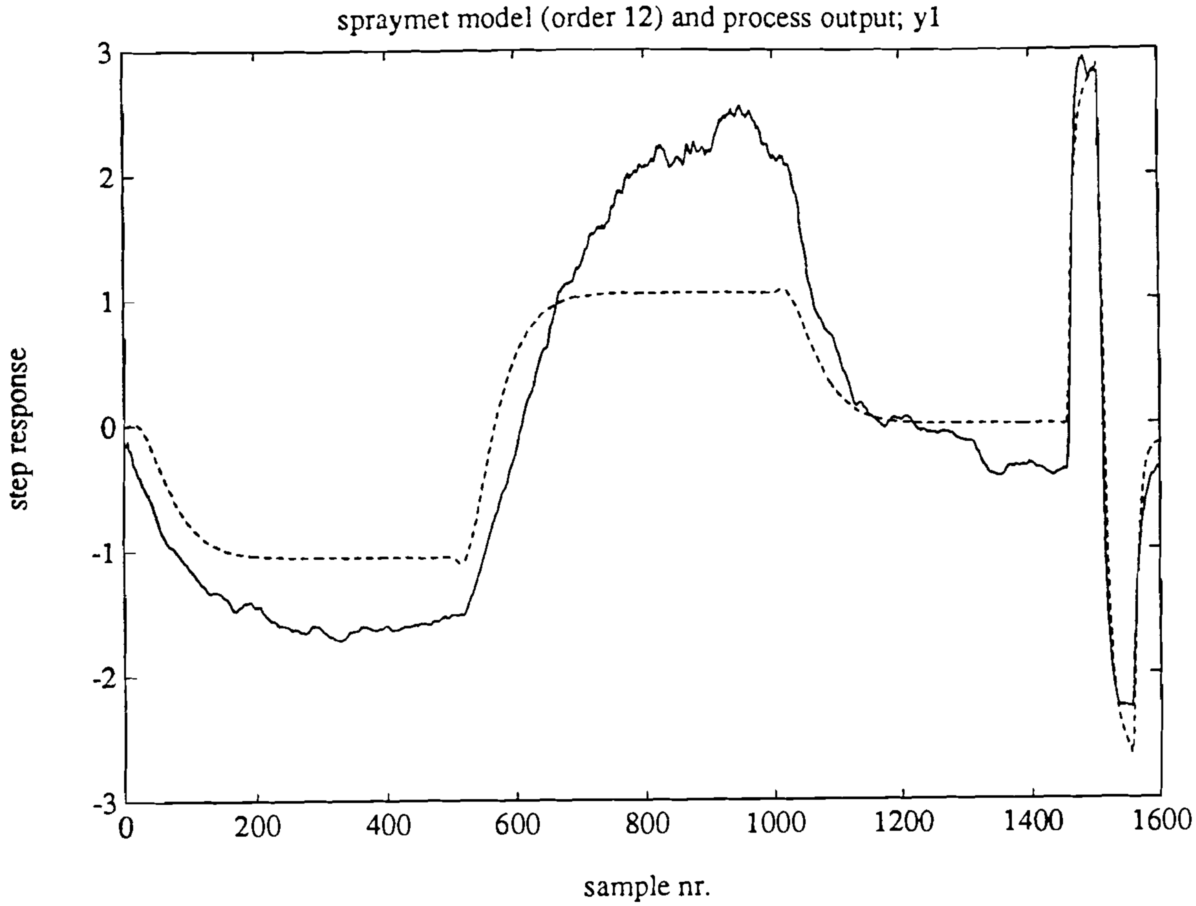
0.0007	0.0008
-0.0012	0.0000


input signals (scaled) and output y1; watercontent powder
step response
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0
200
400
600
800
1000
1200
1400
1600
sample nr.

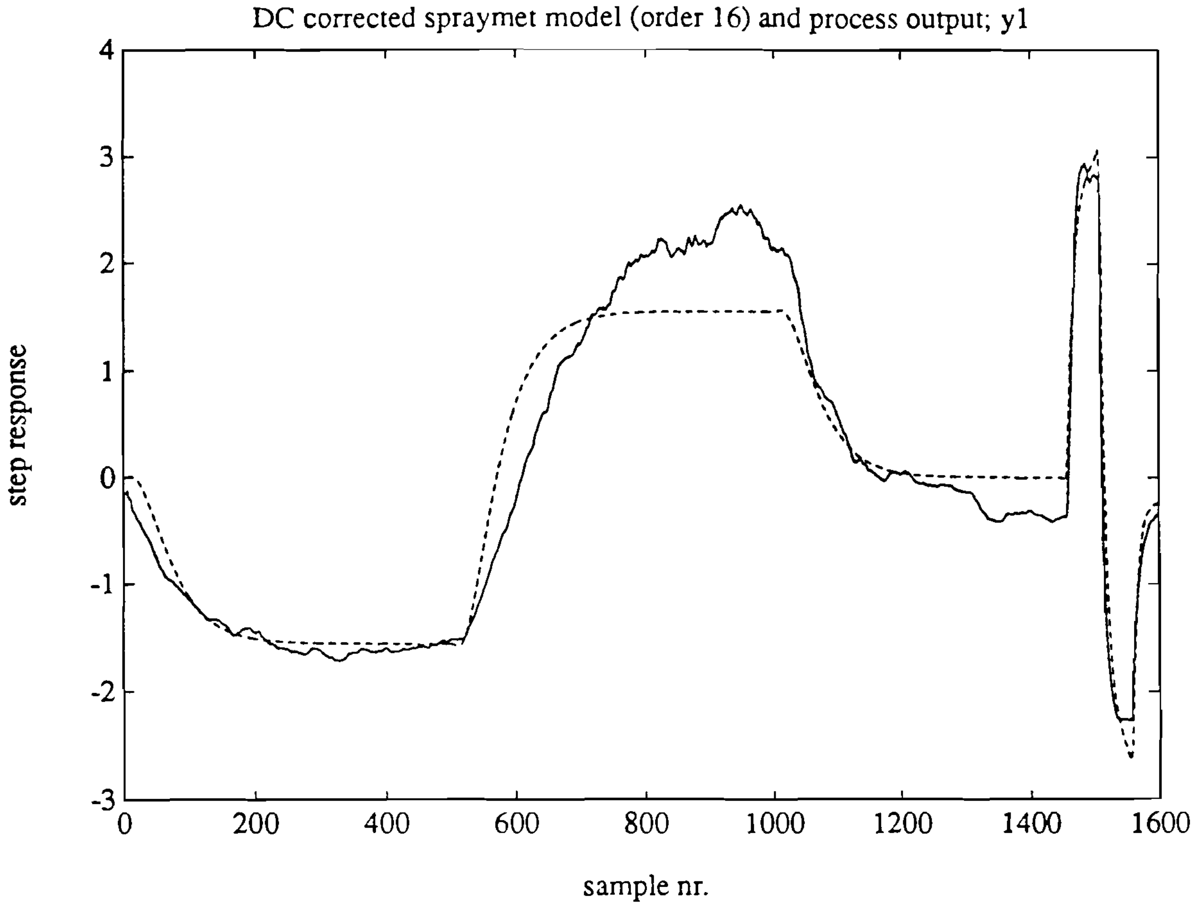

input signals (scaled) and output y2; temperature air
step response
74
72
70
68
66
64
62
60
0
200
400
600
800
1000
1200
1400
1600
sample nr.

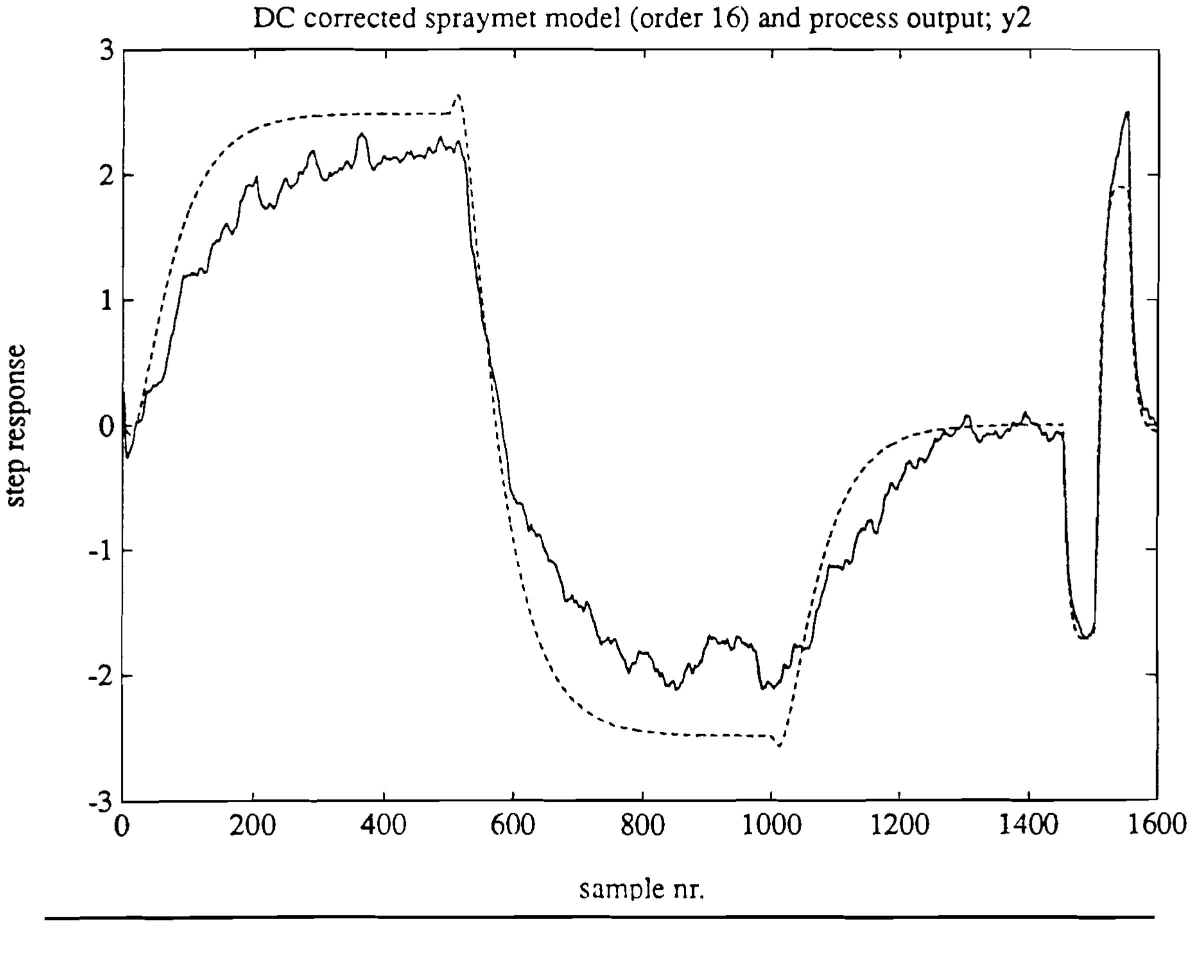

standard model and process output; y1
stepresponse
sample nr.
6
4
2
0
-2
-4
-6
0
200
400
600
800
1000
1200
1400
1600

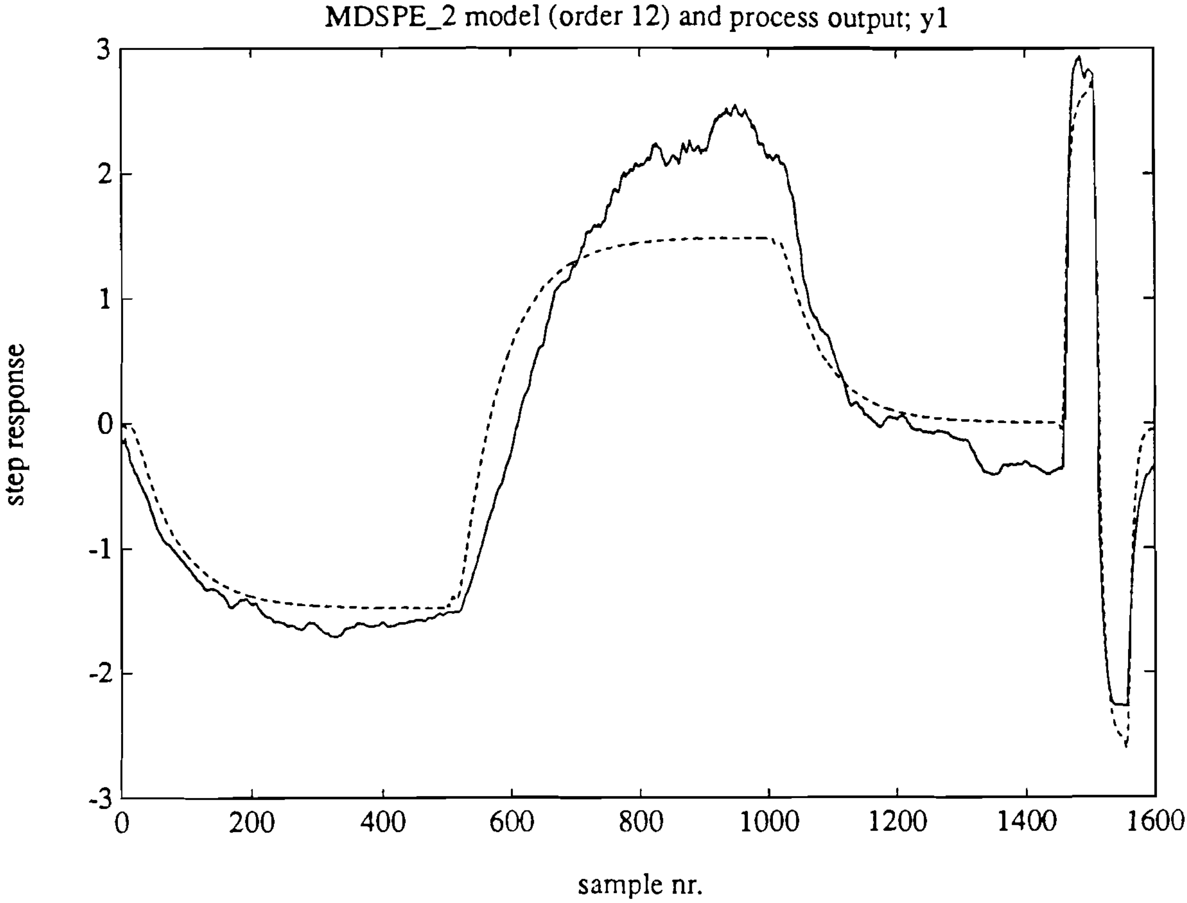

standard model and process output; y2
stepresponse
sample nr.
6
4
2
0
-2
-4
-6
0
200
400
600
800
1000
1200
1400
1600

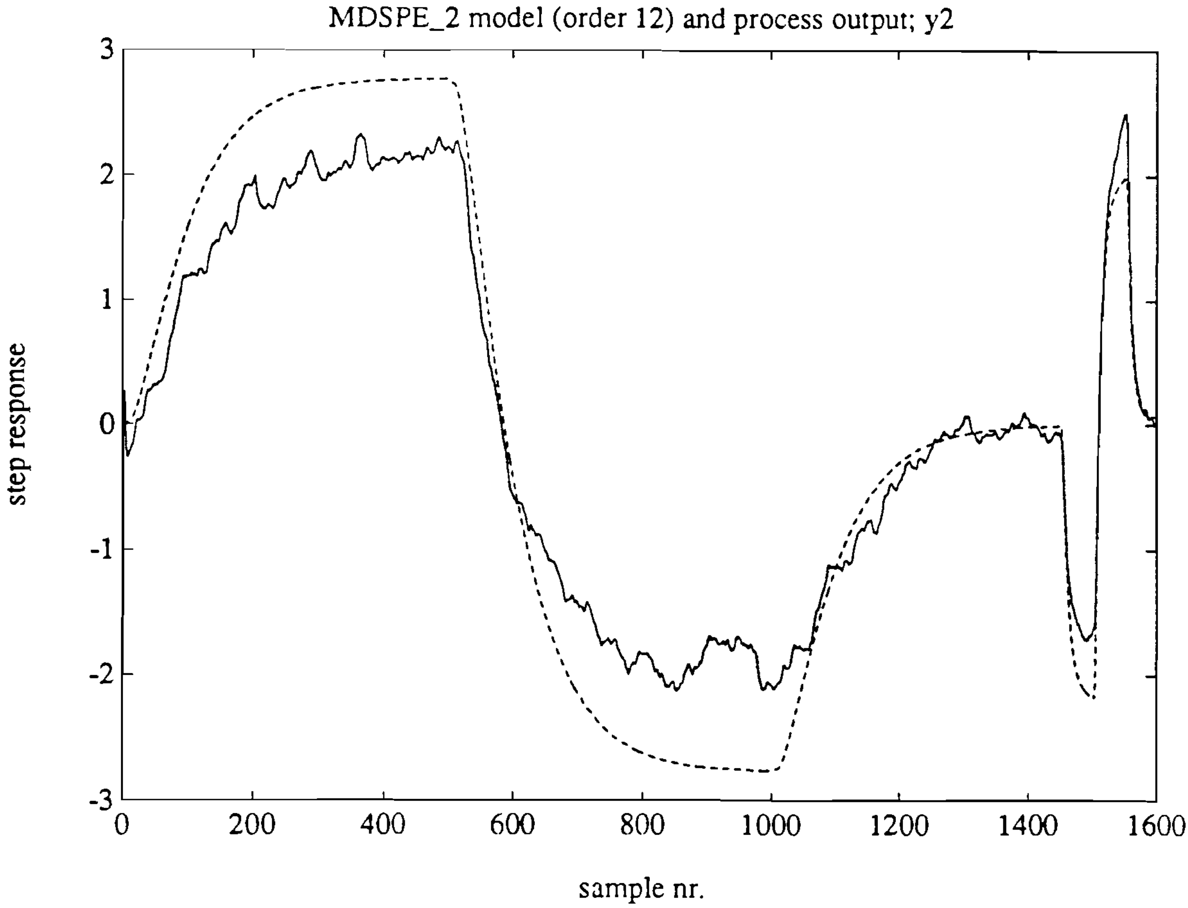

spraymet model (order 12) and process output; y1
step response
sample nr.
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600

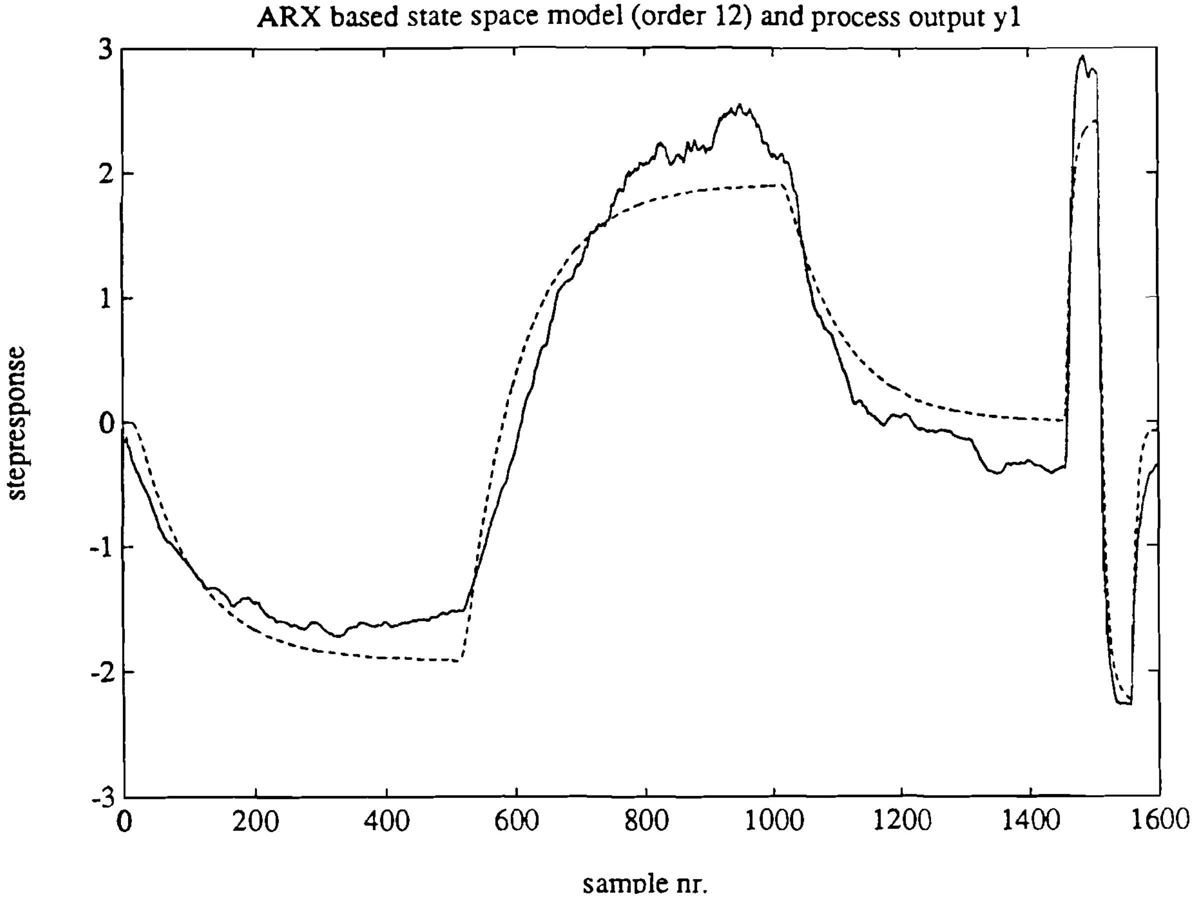

spraymet model (order 12) and process output; y2
step response
sample nr.
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
200
400
600
800
1000
1200
1400
1600

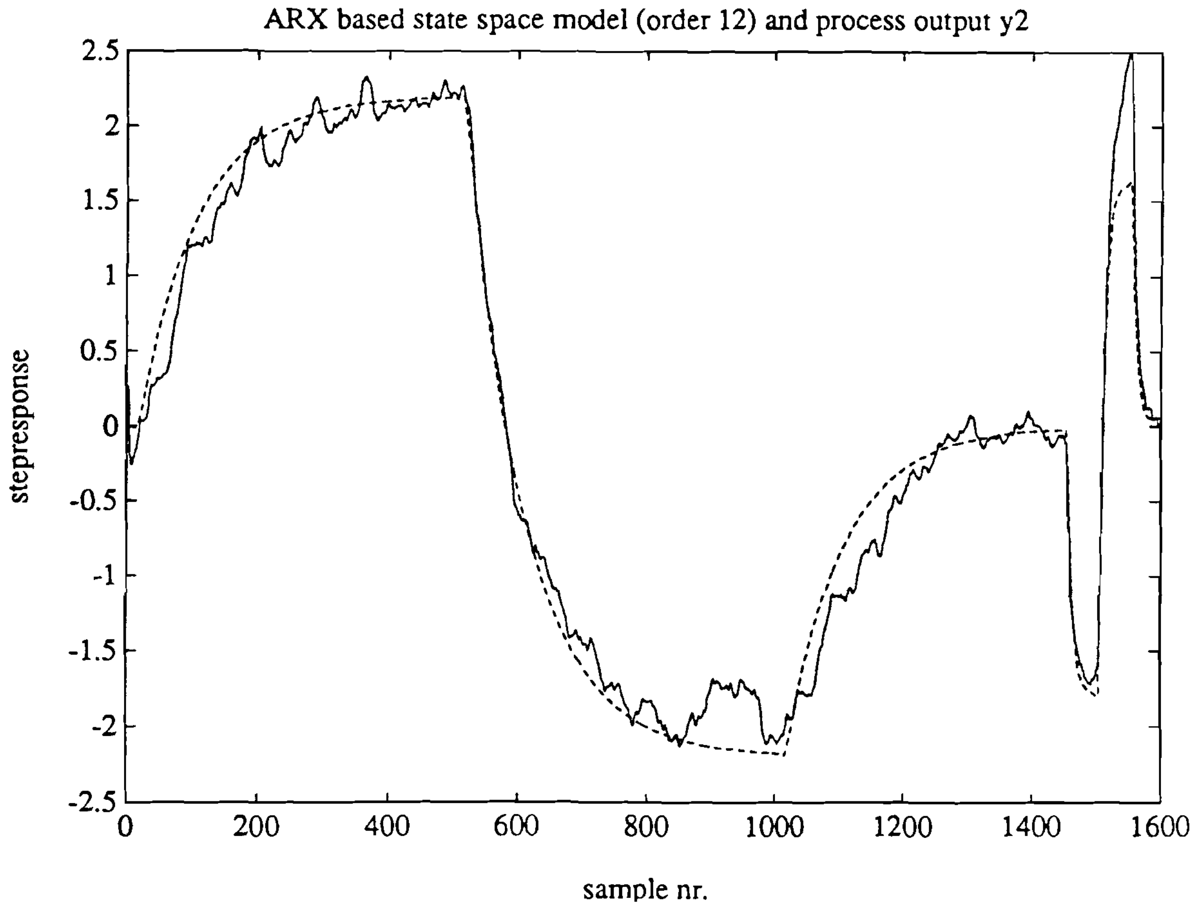

DC corrected spraymet model (order 16) and process output; y1
step response
sample nr.
4
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600


DC corrected spraymet model (order 16) and process output; y2
step response
sample nr.
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600


MDSPE_2 model (order 12) and process output; y1
step response
sample nr.
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600


MDSPE_2 model (order 12) and process output; y2
step response
sample nr.
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600


ARX based state space model (order 12) and process output y1
stepresponse
3
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
1400
1600
sample nr.


ARX based state space model (order 12) and process output y2
stepresponse
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
200
400
600
800
1000
1200
1400
1600
sample nr.

C-1: ARX MISO model estimation tool

```
function [TH]=arxest(est);
% [TH]=arxest(est)
%
% This function estimates a MISO ARX model for each output
% in 'theta' format
%
% input "est" is the IPCOS scaled estimation data set = [u y]
% this function is designed for a 2x2 MIMO system
% without time delays

outnr = input('for which output you want to estimate a MISO ARX model? ');
order = input('what must be the order for the estimated MISO ARX model? ');

input = est(:,1:2);
output = est(:,2+outnr);

th = arx([output input],[order order+1 order+1 0 0]);
```

C-2: Conversion of one MISO ARX model to one MISO state space model

```
function [A,B,C,D]=th2ss(TH)
% [A,B,C,D]=th2ss(TH)
%
% This function converts a MISO model in "theta" ARX format into
% observer canonical form state space realization.

[a,b]=polyform(TH);
m=min(size(b));
n=max(size(a))-1;
A=zeros(n,n);
A(:,1)=-a(2:n+1)';
A(1:n-1,2:n)=eye(n-1);
B=b(:,2:n+1)';
C=[1 zeros(1,n-1)];
D=b(:,1)';
```

C-3: Merging MISO state space models to one MIMO model

```
function [A,B,C,D]=merge(A1,B1,C1,D1,A2,B2,C2,D2)
%
% This function merges two MISO models in state space format
% into one high order MIMO state space model
%
[m,n] = size(A1);
dummy = zeros(m,n);

A = [A1 dummy;dummy A2];

B = [B1;B2];

l = length(C1);
dummy = zeros(l);

C = [C1 dummy;dummy C2];

D = [D1;D2];
```

C-4: State space MIMO model reduction with Schur method

function ddcmdrd.m
choose: type = 3 ; dccor = 0 ; bal = 0

```
function [am,bm,cm,dm,hsv] =ddcmdrd(a,b,c,d,type,k,dccor,bal)
% [am,bm,cm,dm,hsv] =ddcmdrd(a,b,c,d,type,k,dccor,bal)
%
% This routine performs a Schur method model reduction on the
% DISCRETE STABLE MODEL G(z):=(a,b,c,d)
%
% Based on the "TYPE" selected, you have the following options:
% 1). TYPE = 1  --- no: size "k" of the reduced order model.
% 2). TYPE = 2  --- find k-th order model such that the total error
%                          is less than "no".
% 3). TYPE = 3  --- disply all the Hankel SV prompt for "k" (in this
%                          case, no need to specify "no").
%
% If desired the routine corrects the static gain of the model:
%        dccor = 0 --- no DC correction
%              = 1 ---    DC correction
%
%        bal   = 0 --- no numerical balancing of a
%        bal   = 1 ---    numerical balancing of a
%        (am,bm,cm,dm): the reduced order model.
%        hsv          : the n hankel singular values.

%
%       REMARK:
%      1) If no DC correction is wanted the routine is identical to the
         routine
%        schbal of the robust control box.
%      2) it is assumed that a(k+1:n,k+1:n) is non singular if we
         perform DC
%        correction!!!!!!!!!
%
%
%   ROUTINES USED:
%     dcmdrd : performs model reduction in the continuous domain
%              routine of the IPCOS TOOLBOX
%     bilin  : performs a bilinear transform
%              routine of the ROBUST CONTROL TOOLBOX
%
% ---------------------------------------------------------------- %
%     author : Jobert Ludlage                                       %
%     version: 1.0                                                  %
%     date   : 18-8-1989                                            %
%     update : 3  28-12-1990                                        %
% ---------------------------------------------------------------- %
%
%  transform to the pseudo continuous domain
%
   if bal==1,
     [t,a]=balance(a);
     b=inv(t)*b;
     c=c*t;
   end;
%
   [ac,bc,cc,dc] = bilin(a,b,c,d,-1,'G_Bili',[1,1,1,-1]);
%
%  Perform continiuos time model reduction
%
   if (dccor == 1),
     [ah,bh,ch,dh,hsv] = dcmdrd(ac,bc,cc,dc,type,k,dccor);
```

```
   else
     [ah,bh,ch,dh,hsv] =  balmr(ac,bc,cc,dc,type,k);
   end
%
%  transform back to the discrete domain
%
   [am,bm,cm,dm] = bilin(ah,bh,ch,dh,1,'G_Bili',[1,1,1,-1]);
%
   stab=sort(abs(eig(am)));
   [ma,na]=size(a);
   if type==1,
     if stab(k)>1,
       elim=[k+1:ma];
       if k>ma, k=ma; elim=[];end;
       disp('<Now using dmodred>'),
       [am,bm,cm,dm]=dmodred(a,b,c,d,elim);
     end;
   end;
%
%  ------- End of ddcmdrd.m --------
```

function dcmdrd.m

```
function [am,bm,cm,dm,hsv] =dcmdrd(a,b,c,d,Type,no,dccor)
% This routine performs a Schur method model reduction on the
% CONTINUOUS MODEL G(s):=(a,b,c,d).
%
% (ahed,bhed,ched,dhed) = (slbig'*a*srbig,slbig'*b,c*srbig,d)
%
% Based on the "TYPE" selected, you have the following options:
% 1). TYPE = 1  --- no: size "k" of the reduced order model.
% 2). TYPE = 2  --- find k-th order model such that the total error
%                          is less than "no".
% 3). TYPE = 3  --- display all the Hankel SV prompt for "k" (in this
%                          case, no need to specify "no").
% If desired the DC value of the reduced model is corrected s.t. it
%     equals that of the full model:
%         1). dccor = 0 --- no DC correction
%         2). dccor = 1 ---    DC correction
%
%      REMARKS:
%     1) If no DC correction is wanted the routine is identical to the
%         routine schbal of the robust control box.
%      2) it is assumed that a(k+1:n,k+1:n) is nonsingular if we
%         perform DC
%         correction!!!!!!!!!!
%
%      (am,bm,cm,dm): the reduced order model
%      hsv          : the n hankel singular values.


%      ROUTINE USED:   from the ROBUST CONTROL BOX.
%         HKSV : Calculates the Hankel singular values
%         HQR10:
%
%  -----------------------------------------------------------
%
%     author : Jobert Ludlage IPCOS
%
%     version: 0.2 1990
%
%  -----------------------------------------------------------
%
```

```
%
%  perform Schur balancing
%
[ma,na] = size(a);
[md,nd] = size(d);
[p,q,hsv] = hksv(a,b,c);
Flag = exist('xxxxx0');
%
% ------ Model reduction based on your choice of TYPE:
%
if Type == 1
   kk = no;
end
%
if Type == 2
   tails = 0;
   kk = 1;
   for i = ma:-1:1
       tails = tails + hsv(i);
       if 2*tails > no
          kk = i;
          break
       end
   end
end
%
if Type == 3
   format short e
   format compact
   [mhsv,nhsv] = size(hsv);
   if mhsv < 60
      disp('    Hankel Singular Values:')
      hsv'
      kk = input('Please assign the k-th index for k-th order model
reduction:');
   else
      disp('    Hankel Singular Values:')
      hsv(1:60,:)'
      disp('                                    (strike a key for more ...)')
      pause
      hsv(61:mhsv,:)'
      kk = input('Please assign the k-th index for k-th order model
reduction:');
   end
   format loose
end
kk=kk+1;
%
% ------ Save all the states:
%
if kk > na
   am = a; bm = b; cm = c; dm = d;
   aug = [0 0];
   return
end
%
% ------ Disgard all the states:
%
if kk == 1
   am = zeros(ma,na); bm = zeros(ma,nd);
   cm = zeros(md,na); dm = c*inv(a)*b+d;
   bnd = 2*sum(hsv);
   aug = [na bnd];
   return,
end
%
```

```
% ------ k-th order Schur balanced model reduction:
%
bnd = 2*sum(hsv(kk:na));
strm = na-kk+1;
aug = [strm bnd];
%
% ------ Find the left-eigenspace basis :
%
ro = (hsv(kk-1)^2+hsv(kk)^2)/2.;
gammaa = p*q-ro*eye(na);
if Flag == 2
   [va,ta] = rschur(gammaa,1);          % FORTRAN code
else
   [va,ta,msa,swpa] = hqr10(gammaa);
end
vlbig = va(:,(na-kk+2):na);
%
% ------ Find the right-eigenspace basis :
%
gammad = -gammaa;
if Flag == 2
   [vd,td] = rschur(gammad,1);          % FORTRAN code
else
   [vd,td,msd,swpd] = hqr10(gammad);
end
vrbig = vd(:,1:kk-1);
%
% ------ Find the similarity transformation :
%
ee = vlbig'*vrbig;
[ue,se,ve] = svd(ee);
%
seih = diag(ones(kk-1,1)./sqrt(diag(se)));
slbig = vlbig*ue*seih;
srbig = vrbig*ve*seih;
%
am = slbig'*a*srbig;
bm = slbig'*b;
cm = c*srbig;
dm = d;
%
%  DC correction if wanted
%
if dccor == 1,

     vrsmall = va(:,1:(na+1-kk));
     vlsmall = vd(:,kk:na);

     ac22 = vlsmall'*a*vrsmall;
     ac21 = vlsmall'*a*srbig;
     ac12 = slbig'*a*vrsmall;
     bc2  = vlsmall'*b;
     cc2  = c*vrsmall;
     ai22 = inv(ac22);
     am = am - ac12*ai22*ac21;
     bm = bm - ac12*ai22*bc2;
     cm = cm - cc2*ai22*ac21;
     dm = dm - cc2*ai22*bc2;
end;
%   --------- end of dcmdrd.m ---------
```