TU/e technische
universiteit
eindhoven

Modelling and control of a Glass Melting Furnace

by

P.M. Nijsse

ABSTRACT

Modelling and control of a Glass Melting Furnace

Tools for the course Applied System Analysis

By Pieter Maarten Nijsse

Chair Person of Supervisory Committee Prof. dr. ir. A.C.P.M. Backx

Department of Control Systems

A glass melting furnace is difficult to control because of their complexity, high non-linearity's and slow dynamics.

Difficult processes within the industry are until now often controlled through the experience of process operators and process engineers. By using advanced control methods like Model Predictive Control (MPC) these processes can often be improved.

In this project a process of a glass melting furnace is developed from energy, mass and momentum equations. Software within Matlab/Simulink is developed for the pre-testing and pre-processing of the glass melting furnace and all its signals. Next a trajectory to derive a model from the process and implement it within the INCA software of IPCOS technology alone side of the process is discussed. The software environment is configured to apply the model predictive controller. Finally, the complete trajectory from the pre-testing and pre-processing till the design and tuning of the model predictive controller is used for the course Applied System Analysis (5P280) for the student exam assignment underlined with several manuals that discuss the trajectory.

Preface

This thesis is the result of my graduation project, the final part of my study of Electrical Engineer at the Technical University of Eindhoven.

I would like to thank prof. dr.ir. A.C.P.M. Backx and dr. ir. A.J.W. van den Boom for supervising this project and their assistance. Further I would like to thank ir. L.Huisman for helping me to understand the theory behind glass melting furnaces and his support during my graduation. Also thanks to ir. T.C. Dek for his contribution to help me work with the IPCOS technology products and his collegiality during our study of Electrical Engineer at the Technical University of Eindhoven.

Finally, I thank my family and Valesca Danker for their support during my study.

Pieter Maarten Nijsse

Table of Contents

1 Introduction

1.1 Introduction

Efficiency and flexibility are nowadays big issues in the industry. To achieve the high economic objectives set by the industry all efforts are taken. A part of those efforts is the advanced control techniques for process control. At the Technical University of Eindhoven the students can follow a course, applied system analysis (TSA) 5P280, which give them an introduction in the advanced control techniques. The aim of the course is to make the students familiar with state-of-the-art model-based, multivariable control techniques, which are at the moment used in the industry, conceive a test procedure for developing and validating of the required models and the practical design of a Model Predictive Controller. The course aims at providing insight in techniques, those are used for empirical modelling dynamic input output behaviour of multivariable systems with the use of linear, time invariance, discrete time models and design of those models based on Model Predictive Control systems.

The students learn to design a test procedure, with which the required models can be developed and validated in a structured manner. Using the obtained models a Model Predictive Controller will be designed, within the specifications with respect to the performance, robustness, and imposed constrains. The simplified industrial case based on a glass melting furnace is already in existence but in this report the furnace will be extended with several features. This report discusses the whole model of the glass melting furnace including the extra features; per-testing of the data and the automatisation of it in Matlab, with m-files and simulink; furthermore designing and validating of the model in Inca where the pre-tested data from Matlab is used and the implementation of it within a data server will be discussed.

2 Model of glass melting furnace derived from energy, mass and momentum equations.

2.1 Introduction

In a glass melting furnace solid raw materials are melted at high temperatures and the result is a glass melt which then is fined and homogenized. The molten glass contains a high amount of very small bubbles (containing mostly CO_2 and N_2) and also some solid materials (mainly SiO_2) that have to be dissolved in the melt. Furthermore, the glass melt may not be homogeneous. For the melting of the remaining solid particles it's important that they have a sufficient long residence time in the furnace at high temperatures along their trajectories. The small bubbles are released from the melt by letting them grow and so rise to the surface in the so called hot-spot. This process is called the fining process.

For melting, fining and homogenizing processes not only the temperature is important but also the velocities in the melt. High velocity gradients give rise to faster dissolution of the sand grains but on the other hand it may prevent bubbles from reaching the surface due to high drag forces. In glass melting furnaces several measurements are available, these are: bottom temperature measurements, crown temperature measurements, pressure measurements and level measurements. Other variables such as the fining zone where the release of bubbles take place, the so called hot-spot may be important. The problem is that the temperature within these zones can not be measured directly. To determine the temperature in these zones we use an estimator which is designed based on mathematical models that are available for the glass melting furnaces. Here we discuss the design of such an estimator for glass melting furnaces based on energy, mass and momentum equations models. Figure 2.1 shows a side view of a glass melting furnace with its measurements.

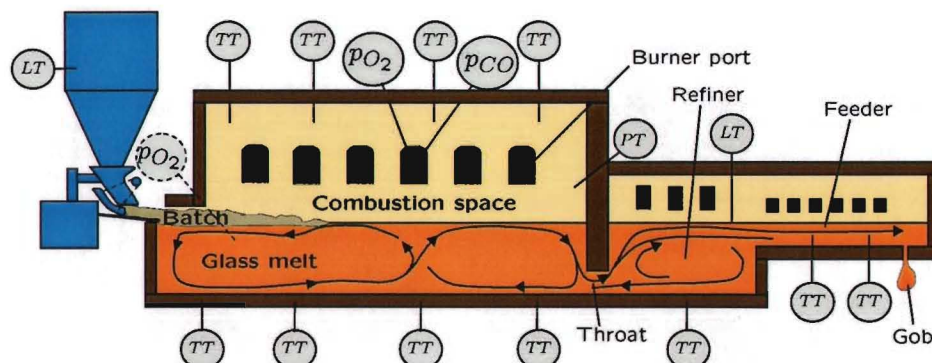## 2.2 Models of the glass melting furnace

Figure 2.1. A side view of a glass melting furnace with its measurements. TT is temperature transmitter, PT is pressure transmitter, LT is level transmitter, PO_2 is partial oxygen pressure and P_{CO} is partial carbon monoxide pressure. The sensor with dotted lines is a sensor for oxygen partial vapour pressure in the melt and it's in a test phase.

Glass melting furnaces are often modelled with CFD-codes (computational fluid dynamics) for design purposes or process operation improvement. Typical and very important properties of these models are the high level of details and the high computational effort needed to calculate a solution. These properties are in some cases important, like control of the glass melting furnace. If a model is used in a controller, it must allow fast enough calculations and the prediction of the dynamic behaviour in the important regions must be good enough. The calculations are good enough if it is in the region of a hundred times real time and the prediction is good enough if the controller make the process perform within the put performance criteria. Knowing the performance criteria of the model, one can image that a CFD-code with a very high number of grid points in general will not live up to these performance criteria of the model. TNO is putting a lot of effort into developing reduced models derived from CFD-code for especially glass melting furnaces. Until these models become available testing of controllers can be done on simple test models derived from first principles or identified by using input-output from a real furnace. In this chapter a simple model is proposed that may be useful for:

- Gaining insight in the dynamics of the chemical and physical processes involved in the melting of glass and their effects on the overall time dependent behaviour of glass melting furnaces;
- Flexible use which can also be easily extended or simplified and used for:
  - The design an testing of controllers;
  - Very fast calculation;
- Providing information about variables in glass melting furnaces that are important for the glass quality.

The model derived is based on furnace 15 of REXAM in Dongen, which is a container glass melting furnace. This furnace has a single recirculation loop in the glass melt and some smaller loops. In figure 2.2 the physical topology of a part of a glass melting furnace with a recirculation loop is given. The part of the glass melting furnace were the physical topology of is given is from the inlet of the furnace to the throat of the furnace, the combustion space with the glass melt beneath it.
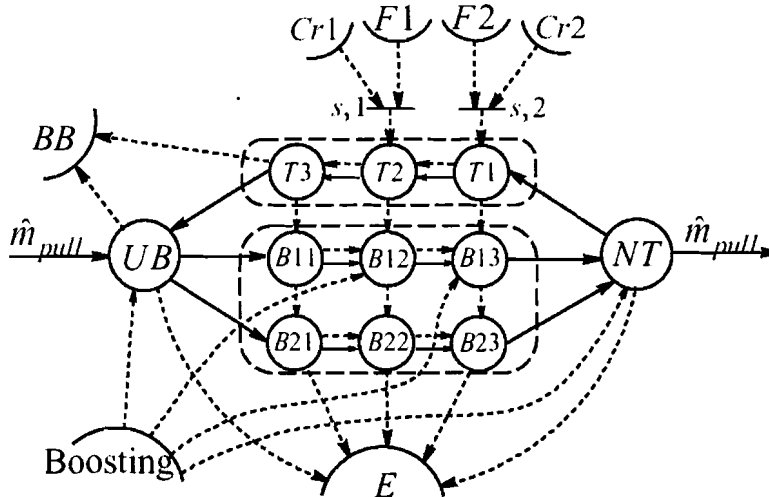
Figure 2.2: Physical topology of a glass melting furnace with a single recirculation loop.

As figure 2.2 displays that the glass melt has been divided into 11 regions. Further the boosting and the environment are indicated within figure 2.2. The physical topology is based on the method of Prof. H. A. Preisig described in "Modelling of dynamic systems" 1999. Indices used in the physical topology of a glass melting furnace description as given in table 2.1.

UB	Under Batch	BB	Batch Blanket
B	Bottom	FS	Fuel Supply
NT	Near Throat	CS	Combustion Space
T	Top	E	Environment

Table 2.1: Indices used in the physical topology of the glass melting furnace.

2.3 Main equations

In this approach single monitor points are chosen as outputs, and it is important that they are placed at relevant positions in the furnace. Using mass, energy and momentum balances it is possible to derive mathematical descriptions for the different parts in the system. Here we use the principle and the basic equations given in a preprint of the 15th IFAC world congress 2002 Barcelona, title "Estimation of process variable in a glass melting furnace" by Ir. L. Huisman. For the regions of these models there will be a set of constants, for the phase boundaries there will be a set of algebraic equations, for lumped, also called ideally stirred, regions there will be sets of differential equations and for the distributed systems it will be partial differential equations. Simple energy equations can be derived by making the following assumptions:

- The glass melt is a singular component with average properties;
- The pressure in the glass melting furnace is approximately constant;
- Mass flows in the top en bottom regions are restricted to the main flow direction;
- Heat flows in these regions are restricted to the main flow direction a vertical flow direction.

With these assumptions one can find for the lumped regions the next equation:

$$\rho C_p V \frac{dT}{dt} = \sum_{\forall m} \alpha_m \hat{m}_m C_p (T_m - T) + \sum_{\forall h} \alpha_h \hat{m}_h + \sum_{\forall l} \alpha_l \hat{m}_l \qquad (2.1)$$

The symbols are explained at the end of the report. The energy equations for the top and bottom regions are as follows:

$$\frac{\partial (C_p T)}{\partial t} = \frac{1}{\rho} \frac{\partial}{\rho \partial \xi_1} \left(k_{eff} \frac{\partial T}{\partial \xi_1} \right) + \frac{1}{\rho} \frac{\partial}{\rho \partial \xi_2} \left(k_{eff} \frac{\partial T}{\partial \xi_2} \right) - v_1 \frac{\partial (C_p T)}{\partial \xi_1} + \frac{S_e}{\rho} + \sum_{\forall k} \frac{\hat{q}_{wall,k}}{\rho} \qquad (2.2)$$

With ξ_1 as convective transport and ξ_2 as conductive transport they are independent spatial coordinates. In the source term S_e or in the work terms $\alpha_l \hat{w}_l$ the boosting power (input) can be included. Heat losses through the side walls are considered as sink terms.

Discretization of the spatial domain the energy equations, equation 2.2, are transformed into a set of ordinary differential equations (ODE).

The temperature description will be completed if the mass flows between the system regions are known. If the assumption is made that the pull rate is approximately equal to the furnace load (due to the presence of a level controller) then only one mass flow is still unknown. This unknown mass flow can be expressed as the ratio, $R = \dfrac{\hat{V}_T}{\hat{V}_{pull}}$, of the glass flowing back to the batch blanket (upper regions flow) and the pull rate. Glass flows in the regions can be described with the Navier-Stokes Equations, a microscopic momentum balance. By taking the dot product of the velocity vector v with the Navier-Stokes equations and integrate the resulting equation over the glass melt volume yields:

$$\dot{K}_{tot} = \sum_{\forall m} \alpha_m \hat{m}_m \left(\frac{1}{2}\frac{\langle v^3\rangle}{\langle v\rangle} + \Phi_m + \frac{P_m}{\rho} \right) + \hat{w}_{pl} - \sum_{\forall i} E_{f,i} + \sum_{\forall b} \rho\beta\langle v\rangle_2 g_2 \left(\langle T\rangle_b - \langle T\rangle \right) V_b \quad (2.3)$$

with $\dot{K}_{tot} = \dfrac{d}{dt} \int_V \left(\dfrac{1}{2}\rho v^2 \right) d\Omega$ the total kinetic energy in the glass melt and

$E_{f,i} = -\int_{V_i} (\tau : \nabla v) d\Omega$ the friction losses from the laminar flow patterns that are assumed

in domain part $\underline{i}$, where also the assumption is made that it only occurs in the top en bottom regions. If the assumption is made that the kinetic energy varies due to changes in the buoyancy forces in the melt only, then:

$$\dot{K}_{rc} = -\sum_{\forall i} E_{f,i} + \sum_{\forall b} \rho\beta\langle v\rangle_2 g_2 \left(\langle T\rangle_b - \langle T\rangle \right) V_b \quad (2.4)$$

Due to temperature differences in the melt this equation approximates the acceleration of the glass in a recirculation loop.

2.4 Equations parts glass melting furnace

A. Heat transfer in laminar flow regions.

For the laminar flow regions in the tank we can use a two dimensional model, therefore we can use the equation (2.2) for the calculations. But first we have to discretize the energy equation into the next equation which we can implement:

$$\frac{dT_{ij}}{dt} = \frac{1}{\rho C_p \Delta \xi_1}\left(\frac{K_{eff}}{\Delta \xi_1}(T_{i+1,j} + T_{i,j}) - \frac{K_{eff}}{\Delta \xi_1}(T_{i,j} + T_{i-1,j})\right)$$

$$+ \frac{1}{\rho C_p \Delta \xi_2}\left(\frac{K_{eff}}{\Delta \xi_2}(T_{i,j+1} + T_{i,j}) - \frac{K_{eff}}{\Delta \xi_2}(T_{i,j} + T_{i,j-1})\right) \qquad (2.5)$$

$$- \frac{\langle \overline{v}_1 \rangle}{\Delta \xi_1}(T_{i,j} + T_{i-1,j}) + \frac{S_{e,ij}}{\rho C_p} + \sum_{\forall k} \frac{\hat{q}_{wall,k}}{\rho C_p v_{ij}}$$

Here, a central difference interpolation scheme for the diffusive terms and an upwind scheme for the convective terms are applied. For the top flow three regions are taking and indexed along the flow direction, for the first region we get (upper right):

$$\frac{dT_{T1}}{dt} = \frac{1}{\rho C_p \Delta \xi_1}\left(\frac{K_{eff}}{\Delta \xi_1}(T_{T2} + T_{T1}) - \frac{K_{eff}}{\Delta \xi_1}(T_{T1} + T_{NT})\right)$$

$$+ \frac{1}{\rho C_p \Delta \xi_2}\left(\frac{K_{eff}}{\Delta \xi_2}(T_{B13} + T_{T1}) - \frac{K_{eff}}{\Delta \xi_2}(T_{T1} + T_{S,T1})\right) \qquad (2.6)$$

$$- \frac{\langle \overline{v}_1 \rangle}{\Delta \xi_1}(T_{T1} + T_{NT}) + \frac{\hat{w}_{bst,T1}}{\rho C_p} + \frac{\hat{q}_{T1|E}}{\rho C_p v_{T1}}$$

For the first of the middle regions we similarly get:

$$\frac{dT_{B11}}{dt} = \frac{1}{\rho C_p \Delta \xi_1}\left(\frac{K_{eff}}{\Delta \xi_1}(T_{B22} + T_{B11}) - \frac{K_{eff}}{\Delta \xi_1}(T_{B11} + T_{UB})\right)$$

$$+ \frac{1}{\rho C_p \Delta \xi_2}\left(\frac{K_{eff}}{\Delta \xi_2}(T_{B21} + T_{B11}) - \frac{K_{eff}}{\Delta \xi_2}(T_{B11} + T_{T3})\right) \qquad (2.7)$$

$$- \frac{\langle \overline{v}_1 \rangle}{\Delta \xi_1}(T_{B11} + T_{UB}) + \frac{\hat{w}_{bst,B11}}{\rho C_p} + \frac{\hat{q}_{B11|E}}{\rho C_p v_{B11}}$$

We can find similar relations for the other bottom regions.

B. Ideally stirred regions.

The region under the batch blanket is an ideally stirred region and an energy equation for a constant pressure zone can therefore be used:

$$\rho C_p V_{UB} \frac{dT_{UB}}{dt} = \rho C_p \hat{V}_{T3|UB} \left(T_{T3} - T_{UB} \right)$$
$$-\hat{V}_{pull} \Phi_{BB}^{BB1} \rho \Delta_R H$$
$$-\alpha_{eff} A_{UB|BB1} \left(T_{UB} - T_{BB1} \right)$$
$$-\frac{\lambda_{wall} A_{UB|E}}{\delta_{wall}} \left(T_{UB} - T_E \right)$$

(2.8)

with $\alpha_{eff} \approx \dfrac{k_{eff} \left(T_{UB} \right)_0}{\gamma_{UB} H_{UB}}$

The degree of mixing in the UB-region, which has an effect on the heat transfer coefficient α_{eff} , is indicated with γ_{UB}. Further more is assumed the chemical reactions take place outside the UB-region and the glass is treated as a single component having average properties that depend on the composition of the glass.
The region near the throat can be modelled similarly with:

$$\rho C_p V_{NT} \frac{dT_{NT}}{dt} = \rho C_p \hat{V}_{B13|NT} \left(T_{B13} - T_{NT} \right)$$
$$+\rho C_p \hat{V}_{B23|NT} \left(T_{B23} - T_{NT} \right)$$
$$-\alpha_{eff,NT} A_{CS|NT} \left(T_{NT} - T_E \right)$$
$$-\frac{\lambda_{wall} A_{NT|E}}{\delta_{wall}} \left(T_{NT} - T_E \right)$$

(2.9)

C. Model equations for the recirculation flow.

To calculate the recirculation flow within the glass melt furnace we make use of the Navier-Stokes equations with the assumption that the kinetic energy of the recirculation flow is only influenced by the size of the buoyancy forces in the melt what results in equation (2.4). It is assumed that the friction losses only occur in the top en bottom regions, indicated by T en B in the physical topology in figure 2.2, and that the flow can be modelled as one dimensional laminar flow between horizontal plates. Two cases can be considered:

• One dimensional flow (in one-direction) over one plate (Top-regions):

$$v_1 = \frac{\Delta P}{\mu L} \left(\frac{1}{2} \xi_2^2 - H \xi_2 \right)$$

(2.10)

In this case the volumetric flow will be:

7

$$\hat{V}_i = \int\limits_{A_i} \frac{\Delta P}{\mu L}\left(\frac{1}{2}\xi_2^2 - H\xi_2\right)d\Omega = \frac{-W\Delta P}{3\mu L}H_i^3 \tag{2.11}$$

and so

$$v_1 = \frac{-2\hat{V}_i}{WH_i^3}\left(\frac{1}{2}\xi_2^2 - H\xi_2\right) = \frac{-\gamma_i\hat{V}_i}{WH_i^3}\left(\frac{1}{6}\gamma_i\xi_2^2 - H\xi_2\right) \text{ with } \gamma_i = 3$$

- One dimensional flow between two plates (Bottom-regions):

$$v_1 = \frac{\Delta P}{2\mu L}\left(\frac{1}{2}\xi_2^2 - H\xi_2\right) = \frac{-\gamma_i \hat{V}_i}{WH_i^3}\left(\frac{1}{6}\gamma_i \xi_2^2 - H\xi_2\right) \text{ with } \gamma_i = 6 \tag{2.12}$$

The accumulation term from the velocity profile can be written as:

$$\dot{K}_{rc} = \left[\frac{\left(\frac{1}{60}\gamma_{T2}^2 + \frac{1}{15}\gamma_{T1}^2\right)\frac{1}{2}\rho V_T \hat{V}_{pull}^2}{W_T^2 H_T^2}R + \frac{\gamma_B^2 \rho V_B \hat{V}_{pull}^2}{60 W_B^2 H_B^2}(R+1)\right]\frac{dR}{dt} \tag{2.13}$$

with $R = \dfrac{\hat{V}_T}{\hat{V}_{pull}}$ a dimensionless backflow ratio and γ a constant depending on the

situation (one or two plates). For the friction we can derive:

$$E_{f,T1} = \frac{3\hat{V}_T^2 \mu L_{T1}}{W_T H_T^3} \quad , \quad E_{f,T2} = \frac{12\hat{V}_T^2 \mu L_{T2}}{W_T H_T^3} \text{ and } E_{f,B} = \frac{12\hat{V}_B^2 \mu L_B}{W_B H_B^3} \tag{2.14}$$

and the buoyancy terms can be written:

$$\sum_{\forall b}\rho\beta\langle\hat{v}_z\rangle g_z\left(\overline{T}_b - \overline{T}\right)V_b = \left(\frac{1}{2}+R\right)\frac{\rho\beta g V\hat{V}_{pull}}{WL}\left(T_{NT} - T_{UB}\right) \tag{2.15}$$

Using the equations (2.4), (2.13) and (2.15) we rewrite the kinetic energy equation and so obtain a nonlinear equation for R:

$$\frac{dR}{dt} = \frac{\dfrac{\mu L_B}{W_B H_B^3}\hat{V}_{pull}\left[\left(\dfrac{1}{2}+R\right)\left(T_{NT}-T_{UB}\right)\right]}{\left[\dfrac{\left(\dfrac{1}{60}\gamma_{T2}^2 + \dfrac{1}{15}\gamma_{T1}^2\right)\dfrac{1}{2}\rho V_T}{W_T^2 H_T^2}\hat{V}_{pull}^2 R + \dfrac{\gamma_B^2 \rho V_B}{60 W_B^2 H_B^2}\hat{V}_{pull}^2(R+1)\right]}$$

$$- \frac{15\dfrac{\mu L_T}{2W_T H_T^3}\hat{V}_{pull}^2 R^2 - 12\dfrac{\mu L_B}{W_B H_B^3}\hat{V}_{pull}^2(R+1)^2}{\left[\dfrac{\left(\dfrac{1}{60}\gamma_{T2}^2 + \dfrac{1}{15}\gamma_{T1}^2\right)\dfrac{1}{2}\rho V_T}{W_T^2 H_T^2}\hat{V}_{pull}^2 R + \dfrac{\gamma_B^2 \rho V_B}{60 W_B^2 H_B^2}\hat{V}_{pull}^2(R+1)\right]} \tag{2.16}$$

9

This equation describes the acceleration of the recirculation glass, due to change in the temperature differences in the melt. The acceleration is counteracted by the friction forces.

2.5 Realisation of model

The glass melting furnace can be described as shown in the paragraph 2.4 Equations parts glass melting furnace. By rewriting the differential equations of the heat transfer in laminar flow regions and ideally stirred regions for the temperatures and also the model equations for the recirculation flow a usable state space model can be derived. This state space model has 12 states which represent the 12 states of the physical topology of figure 2.2.The state space model equations are shown in appendix 1. Because the model of the glass melting furnace will be considered by the students who follow the course Applied System Analysis as the actual process from which they have to derive a model from, the glass melting furnace has to be as realistic as possible. Disturbances at the inputs and white noise with spikes at the outputs can make the model more realistic, as almost all industrial processes are corrupted with these types of "noise", in figure 2.3 a schematic representation is given.
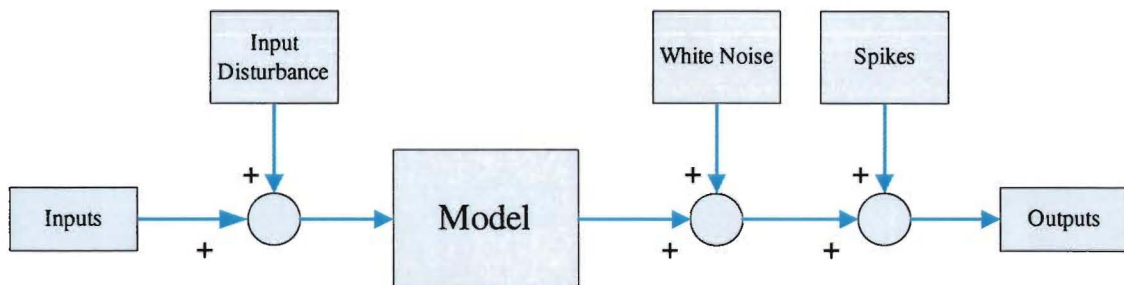
Figure 2.3: Schematic of model of glass melting furnace with disturbance, white noise and spikes.

The inputs of the model are corresponding to the physical topology as follows:
- Electrical boosting under the batch, region UB;
- Electrical boosting in the centre and near throat, B12, B13 and NT;
- The connected surface temperatures of T1, T2 and NT.
The outputs of the model are the temperatures which can be measured; these are:
- Temperature in region B21;
- Temperature in region B22;
- Temperature in region NT.

Now we have defined the glass melting furnace model by state space, the inputs, the outputs and several elements to make the model more realistic a Matlab/Simulink model can be derived with several underlying m-files. The Matlab/Simulink model and the underlying m-files can found on the cd-rom in the directory "course TSA

software\Matlab Simulink" subfolders "Simulink files" and "TSA m Files lower level" which is included in this report.

3 Signal preparation for identification.

3.1 Introduction

Process models obtained from input output signals have their quality heavily depending on the characteristics of the signals offered to the identification algorithms which derive the process models. The quality of the obtained model will be governed by the disturbances present in the signals. Disturbances which have a bad influence on the quality of the models are for example: spikes, drifts, offsets, significant different in powers of the various input and output signals.

Because of their negative influence on the quality of the models and because almost all industrial process data will be corrupted by these types of "noise" it's advisable to try to reduce these disturbances as much as possible. For this purpose dedicated signal processing techniques have been developed, applied within the course "applied system analysis", Technical University code 5P280, of Prof. dr. ir. A.C.P.M. Backx and dr. ir. A.J.W. van den Boom.

Another problem encountered in the industry is the present of, often relative long, time delays in the measured process transfers. If the signals applied to the identification algorithms are not corrected for these time delays, the delays have to be estimated by the algorithms. In most cases this implies many extra parameters, corresponding to these time delays, have to be used for the modelling. This too has a deteriorating influence on the quality of the models ultimately obtained.

3.2 Trend determination and correction

The (sub) process one wants to model for control purposes is mostly determined by a set of inputs and outputs that satisfy the following conditions:

- On-line measurement of inputs and outputs must be feasible;
- The selected inputs should have a direct (short time delay) and if possible, large influence on the process outputs;
- The inputs have to allow full control of the outputs;
- The permitted changes of the inputs have to enable measurable process responses with amplitudes and dynamic ranges that exceed those of the disturbances in the outputs.

In general, not all actual inputs which have a direct influence on the outputs are selected. The inputs that not have been selected for the modelling may, however, contribute to the changes found at the outputs of the (sub) process. During the modelling these changes are considered as coloured output noise. Often the characteristics of these contributions to the output noise are known and in many cases the effects are slow variations of the outputs: trends. These trends have a bad influence on the quality of the model obtained through identification due to the limited length of the data set used for the identification.

To show the influence of the limited data set, the output signal is split into two parts:

$Y = Y_p + Y_{tr}$

with: Y_p the process outputs without trend

Y_{tr} the trend signal added to the process outputs

An approach to separate the trend from a signal is based on the determination of the trend by filtering the signal with a low pass filter and subtraction of the obtained trend from the signal. However, filtering of a signal with a low pass filter introduces a phase shift between de original signal and the obtained trend signal. Subtraction of the two leaves unwanted signal components in the corrected signal. To overcome this phase shifting problem, the signal can be filtered twice: once with a causal low pass filter en once with a corresponding anti-causal filter. The average of the sum of the two filtered signals obtained will not be shifted in phase any more compared to the original signal. This can be mathematical expressed:

$$Y_{tr_k} = \sum_{i=0}^{\infty} h_i^c Y_{k-i} + \sum_{i=-\infty}^{-1} h_i^{ac} Y_{k-i} = \sum_{i=-\infty}^{\infty} h_i Y_{k-i} \tag{3.1}$$

with: h^c the causal filter impulse response

h^{ac} the anti-causal filter impulse response

and $h_i^c = h_{-i}^{ac}$ for $1 \leq i < \infty$

The design of the filter will be a compromise between the demands, determination of the trends as accurate as possible and that no relevant process output signal information will be lost.

Beside trends, signals can have other unwanted characters, offsets. Offsets have almost the same influence on the estimated parameters as a trend has. The offsets in the signals are assumed to be signal values that are constant in time. The changes due to the process dynamics are added to these offset levels.

3.3 Peak shaving

In the industrial practice is the disturbance of measured signals with spikes often a problem. This is due to the amplitude of those spikes which are mostly very large compared with the signal changes obtained from the process. Most of the time the spikes last from one sample to sometimes tens of samples. Considering that spikes are an important part of the noise energy they can have a great influence on the ultimate model although they have no relation with the process themselves. Therefore it is wise to remove the spikes as much as possible. This can be done in the following four steps:

- Clip the signal amplitudes to values never reached by the real process signals:

$$\tilde{s}_k = f_{cl} \cdot s_k \tag{3.2}$$

with: $\tilde{s}_k$ the clipped signal

s_k the measured signal corrupted with spikes

$$f_{cl} = \begin{cases} s_{max}\left(\dfrac{1}{s_k}\right) & for : s_k \geq s_{max} \\ 1 & for : s_{min} \leq s_k \leq s_{max} \\ s_{min}\left(\dfrac{1}{s_k}\right) & for : s_k \leq s_{min} \end{cases}$$

- Compute a trend signal from the clipped signal:

$$\overline{s}_k = \sum_{i=k-l}^{k} h_i \tilde{s}_{k-i} \tag{3.3}$$

- Compute the standard deviation of the trend-corrected, clipped signal:

$$s = \frac{1}{l}\sqrt{\sum_{i=0}^{l}(\tilde{s}_i - \overline{s}_i)^2} \tag{3.4}$$

- Interpolate all samples of the original signal that are outside the band which can be defined as α· standard deviation. The permitted signal band is given by:

$$x_k = \begin{cases} \overline{s}_k + \alpha \cdot s & \text{upper limiet} \\ \overline{s}_k - \alpha \cdot s & \text{lower limit} \end{cases}$$

All consecutive sample values that are outside the permitted band are replaced by values obtained from a linear interpolation starting from the least accepted sample value and reaching to the first next sample value within the permitted band after the spike.

The multiplication factor α has to be chosen such that no process signal values are outside the permitted signal band. With this method the noise energy caused by the spikes is reduced significantly and because of the interpolation the signal is kept at an average process signal level where at those sample moments no real process information is available. Using this signal treatment the influence of the spikes is reduced to a negligible level.

3.4 Estimation of time delays

Measured process responses can contain time delays. If those time delays are not corrected, they become a part of the estimated process model. Although they are a part of the model, the time delays do not contribute to the further dynamic behaviour of the process. Because the time delays involve extra model parameters (one extra Markov parameter for each delay sample) the number of parameters to be estimated increases when the modelling of the time delays has to be done by identification algorithms. When the time delays are not corrected, the parameters increases and so the computational effort required for the modelling of the process which therefore decreases the expected accuracy of the model.

A good approach to overcome this time delay problem is estimation of the time delays separately from and prior to model parameter estimation. The information obtained can then be used as a-priori knowledge to correct the measured process signals for these time delays as much as possible. When the estimated model is obtained, the time delays can then easily be included into the model.

Estimation of time delays in process input/output transfer can be done with correlation techniques. One method is based on the analysis of the cross-correlation between input and output signals. The assumption is made that the process is ergodic, the input signal applied to the process is stationary, white, inter-channel independent, zero mean noise sequence with covariance matrix R:

$$E\{u \cdot u^t\} = R = \sigma^2 I_p \tag{3.5}$$

Then the autocorrelation function of the input signal becomes:

$$\psi_{uu}(\tau) = \lim_{l \to \infty} \frac{1}{l+1} \cdot \sum_{i=0}^{l} u_{i+\tau} \cdot u_i^t = \begin{cases} \sigma^2 I_p & \tau = 0 \\ 0 & \tau \neq 0 \end{cases} \tag{3.6}$$

Furthermore, if the input signal is assumed not to be correlated with the output noise and using the properties of the input signal u, the following expression is obtained for the cross-correlation between outputs and inputs:

$$\psi_{yu}(\tau) = \sum_{j=0}^{\infty} \lim_{l \to \infty} \frac{1}{l+1} \cdot \sum_{i=0}^{l} M_j u_{k+i+\tau-j} \cdot u_{k+i}^t + \psi_{nu}(\tau) = \sum_{j=0}^{\infty} M_j \psi_{uu}(j-\tau) \tag{3.7}$$

In equation (3.7) the parameters M_j are the elements of the process impulse responses including the time delays. Substitution of equation (3.6) into (3.7) gives:

$$\psi_{yu}(\tau) = \sigma^2 M_\tau \tag{3.8}$$

Consequently the cross-correlation function obtained will have the shape of the impulse responses in case the input signal is a stationary, white, inter-channel independent, zero mean noise sequence. Otherwise the cross-correlation function is convolution of the impulse responses with auto-correlation function of the input.

The actual time delays can be found from the cross-correlation function by looking for the beginning of a sequence of values that differ significantly from zero. The first element of that sequence is the time delay.

3.5 Basic assumptions

Basic assumptions have to be made to derive a model from input output data, these are;

-       Process is linear;
-       Process is time invariant;
-       Process behaviour can be approximated by a finite complexity;
-       Process is causal;
-       Manipulated variables enable within their available operating range.

Without these basic assumptions it's not possible to derive a suitable model which allows fast enough calculations and the good enough prediction of the dynamic behaviour in the important regions. These basic assumptions can be derived and tested on the process this

is a part of the Course 5N060 ``Stochastic System Theory" taught by Dr. S. Weiland at the Technical University Eindhoven.

Test for linearity

Linear models are used in the methods for modelling the dynamic behaviour of processes. No industrial process, however, is linear. For a process to have a linear transfer the following condition has to fulfilled (superposition theorem):

$$u_1 \rightarrow y_1 \text{ and } u_2 \rightarrow y_2 \Rightarrow \alpha \cdot u_1 + \beta \cdot u_2 = \alpha \cdot y_1 + \beta \cdot y_2$$

with u_1 -input signal with an arbitrary amplitude ($u_1 \neq 0$);
      y_1 -the process response to input u_1;
      u_2 -second input signal ($u_2 \neq 0$);
      y_2 -the process response to input u_2;
      α, β -arbitrary constants.

First a test has to be done to test for steady state linearity of the process in the surroundings of the work point. When a process has non-linear steady-state responses then in most cases linearization is possible with linearization polynomial functions. For testing steady-state linearity a staircase test signal can be applied to each input of the process separately:

$$u_k = \alpha \bullet \sum_{i=0}^{s} \left[u(k - i \cdot \tau) - u(k - (s+1+i) \cdot \tau) - u(k - (2s+2+i) \cdot \tau) + u(k - (3s+3+i) \cdot \tau) \right] \quad (3.9)$$

with: u(j) a unit step at sample moment j.

The time interval τ between the steps most be chosen accordance to the response time of the process. By each step applied the process most reach a steady-state, then a steady-state analysis can be done properly. The maximum amplitude of the test signal has to exceed the amplitude range that has to be covered by the model. Each step of the stair case test signal has to have an equal amplitude range; therefore the responses to the different steps need to have the same amplitude if the process is steady-state linear.

Not only the steady-state non-linearity's but also the dynamic non-linearity's have to be investigated. To investigate the dynamic non-linearity's a similar test based on the test for the steady-state non-linearity's is used, but now as an amplitude modulated sinusoidal signal. Here the test signals consists of different frequencies of the wave carrier, thus it gives an insight in the non-linearity's at different frequencies.

Another test that can be used for the detection of dynamic non-linearity's is based on the analysis of process responses on Pseudo Random Binary Noise Sequences (PRBNS) of different amplitudes. The cross-correlation function of the applied PRBNS test signals on the inputs and the different amplitude responses of the outputs triggered by the input signals give an impression of the behaviour of the process in response to different amplitudes. Significant changes in the cross-correlation function are indications of non-linearity's.

Test for time invariancy

Time invariancy is an important specification from modelling the dynamic behaviour of processes. A process can not be correctly modelled if the behaviour of the process changes as a function of the time. For a process to have a time invariant transfer the following condition has to fulfilled:

The state space of the model (appendix 1) can be rewritten towards:

$$R_y\left(\frac{d}{dt}\right)[y] = R_x\left(\frac{d}{dt}\right)[x] \Rightarrow 0 = R_y\left(\frac{d}{dt}\right)[y] + R_x\left(\frac{d}{dt}\right)[x] \Rightarrow 0 = R\left(\frac{d}{dt}\right)\begin{bmatrix} x \\ y \end{bmatrix} \qquad (3.10)$$

If the process is stable then $R\left(\frac{d}{dt}\right)\begin{bmatrix} x \\ y \end{bmatrix} = 0$, therefore the model must be stable (disregarding numerical errors).

By selecting $\beta = \ker R\left(\frac{d}{dt}\right) \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix}(t) \in \beta$ then $\begin{bmatrix} x \\ y \end{bmatrix}(t + \tau) \in \beta$,

we can conclude that: $R\left(\frac{d}{dt}\right)\begin{bmatrix} x \\ y \end{bmatrix}(t) = 0$ for $\forall t$, so $R\left(\frac{d}{dt}\right)\begin{bmatrix} x \\ y \end{bmatrix}(t + \tau) = 0$ and therefore the process and model, are time invariant.

Test for causality

A signal is causal if f(t)=0 for t<0. So if we rewrite the state space of our model to the form: $\dot{\underline{x}} = A\underline{x} + \underline{b}u$, $y = c^T\underline{x} + du$ and we consider a causal input signal and the model is linear then the response of the model for t<0 on the causal input signal, which for t<0 will be a zero-input-signal, can only be a zero-output-signal. So a causal input signal will produce a causal output signal and therefore the model will be causal. This can simply be tested by using a causal input signal, for example a step.

3.6 Realisation of signal preparation

The realisation of the signal preparation will be done by several special written m-files in Matlab combined with a simulation in Simulink. With these m-files a total of five several simulations can be chosen for the signal preparation, these are:
-       Free run test;
-       Delay time test;
-       Step response test;

-      Stair case test;
-      Pseudo Random Binary Noise Sequence (PRBNS) test.

To run the m-files, two low pass filters have to be designed with the tool "fdatool" of Matlab. Within the m-files the output data can be filtered with those filters at a causal and anti-causal manner. This prohibits an extra phase shifting problem. A proper design of the two filters will eliminate the spikes for an important part therefore the remaining noise energy in the spikes will have a limited influence on the to devise ultimate model of the students. At this point the model as described in "Model of glass melting furnace derived from energy, mass and momentum equations" will be considered as the process.

With the free run test, m-file "FreeRun.m" the drift per time unit of all the process outputs can be measured.

The m-file "DelayTimeCalculation.m" all the time delays of the process in relation to the selected input and al the outputs can be estimated through cross correlation of the input signal and output signals.

The step response of the process in relation of all the inputs and outputs can be estimated with a first order response fitting, $y(t) = Gain \times \left(1 - e^{\frac{-(t - t_{delay})}{\tau}} \right)$ through the m-file "StepResponseEstimation.m".

The gain, tau and the time delay, together with the found time delay, "DelayTimeCalculation.m" can be estimated and/or checked.

The m-file "StairCase.m" has the same features as the m-file "StepResponseEstimation.m" except now a staircase signal is used, where all aspects such as time per stair, number of stairs of incurring slope and amplitude per stair will be designed by the user. A logical consequence is that the selected output response will be estimated per stair response by a first order response fitting describes as above. This will give the user an extra insight in the linearity of the process within its designed work area.

The Pseudo Random Binary Noise Sequence (PRBNS) test done with m-file "PRBNS.m" there the user is allowed to extract, with a designed PRBNS signal for all the inputs, the responses of the outputs of the glass melting furnace process. The user has the possibility to design the PRBNS input signals by selecting the time of the PBRNS signal per input, the maximum amplitude of that signal per input and the switching probability of the PBRNS signal per input.

All the m-files will store the data of both the inputs and the outputs with the actual real time per sample in such a way that it can be used as process data for obtaining a model of the glass melting furnace process with the Modeller of Inca.

Flow charts of the m-files can be found in appendix 2 to 6. Of course all parameters to run the m-files and the process, the glass melting furnace model within Simulink, are all ready preset through the usage of the m-files.

The actual use of the m-files including the filter design is discussed in the manual "User's Manual customized software" which can be found on the cd-rom included with this report in the directory "course TSA software\Installation Information Files". Within this manual the filter design, the filter properties, all the m-files and properties of all the m-files are discussed. With this manual the students who follow the course TSA will have a guideline to derive and prepare the data from the process, the glass melting furnace

model within Simulink. Also through the usage of the m-files derived and prepared data will automatically be rewritten into the needed structure for use in the application "Modeler" which will be discussed later on in this report.

4 Finite Impulse Response model

4.1 Introduction

A linear time invariant dynamic process is uniquely characterized by its impulse response. For stable processes, the impulse response will tend to zero for increasing time and may then be reduced. This results in so-called FIR (Finite Impulse Response) models or Markov parameters. For SISO (Single Input Single Output) process such a model is given by:

$$y(t) = g_1 u(t-1) + g_2 u(t-2) + \ldots + g_n u(t-n) = \sum_{i=1}^{N} g_i u(t-i) = \varphi(t)\theta$$

In matrix form :

$$y = \phi\theta \tag{4.1}$$

$$\text{where} : \theta = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} \quad \text{and } \phi = \begin{bmatrix} u(t-1) & u(t-2) & \ldots & u(t-n) \end{bmatrix}$$

By introducing the residual as: $y = \phi\theta + \varepsilon$ in matrix form,
where:

$$y = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(N) \end{bmatrix}, \qquad \varepsilon = \begin{bmatrix} \varepsilon(n+1) \\ \varepsilon(n+2) \\ \vdots \\ \varepsilon(N) \end{bmatrix} \text{and } \phi = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(1) \\ u(n+1) & \ddots & & \vdots \\ \vdots & & & \vdots \\ u(N) & \ldots & \ldots & u(N-n) \end{bmatrix}$$

Figure 4.1 shows the block diagram of a FIR model estimation of a SISO process.
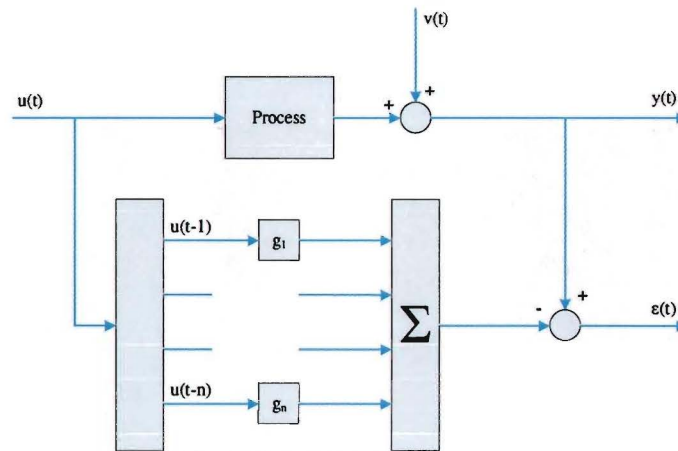
Figure 4.1: error generation in FIR model estimation.

Figure 4.1 gives us the output error of the misfit of the model compared to the process. With the least square method we can estimate the parameters of the model. First we calculation the sum of the squares of the simulation error or output error:

$$V_{OE} = \frac{1}{N} \sum_{t=1}^{N} \hat{\varepsilon}_{OE}(t)^2 \qquad (4.2)$$

So we can find the minimum of V_{OE} through:

$$\frac{\partial V_{OE}}{\partial g_i} = 0 \quad \text{with } i = 1...n \qquad (4.3)$$

The minimum of equation 4.3 give us the best fit of the model on to the process so with computer assistance the parameters of the FIR-model can be calculated.

The order of the FIR model has a direct influence on the error generation, Furthermore; noise on the signals can cause randomness in identified models. One can calculate the order of the FIR model through several mathematical theories or one can use the practical fist-rule: "The order of FIR-parameters is approximately equal to the time to steady state divided with the sample time".

FIR models have several advantages and disadvantages, these are as follows:
Advantages:
- Less a priori knowledge of the process is required to estimate a FIR model. Only the order of the process is required;
- FIR model estimation is statistically unbiased, the expectation of the estimation equals the true values, and consistent, the estimation tends to the true value when a number of data points to infinity.

Disadvantages:
- The FIR model often needs to have a larges number of parameters; the estimated process is not accurate for a finite number of data. The inaccuracy has to do with the variance of the estimation which is proportional to the number of parameters of the FIR model;
- A FIR model is often not suitable for linear control design methods due to the reduction of the very large FIR model.

The extension of the FIR model estimation to multivariable processes is straightforward in the same way as with a single variable process, described above. The generalization of the difference equation model leads to a so called matrix friction description (MFD). In general, parameterisation of MIMO MDF models for identification is a complicated problem this will not be discussed in this report.

4.2 Order selection, a simulation approach

In practice, the order of a process is seldom exactly known. Order selection should not be confused with model validation. Model order determination is an imported part in the process identification. Here we present a simple but effective technique to estimate the order of the process. The order of the FIR model is related to the time to steady state of the process or the settling time.

Order selection is nothing more then finding a model order that best explains the given process data. Here the task of model validation is necessary to check whether the model is good enough for the intended use. A model derived from data can give the best possible fit to that data, but that model doesn't have to be the best model for using purposes.

A method for validating a model is to simulate the estimated model using some real input data of the process and comparing the output of the model to the real output data of the process. In figure 4.2 (next page) a schematic representation of this model validation is given.
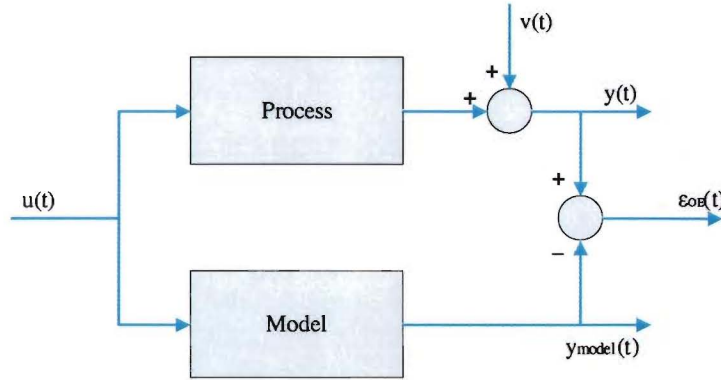
Figure 4.2: Order selection by simulation.

The goodness of the fit is measured by the sum of the squares of the simulation error or output error:

$$V_{OE} = \frac{1}{N} \sum_{t=1}^{N} \hat{\varepsilon}_{OE}(t)^2 \qquad (4.4)$$

where

$$\hat{\varepsilon}_{OE}(t) = y(t) - \frac{\hat{B}(q^{-1})}{\hat{A}(q^{-1})} u(t) = y(t) - Y_{model}(t)$$

Note that we are discussing here the FIR model so A(q)=1 therefore the equation error is equal to the output error.

For order selection we can use the output error:

$$\hat{\varepsilon}_{OE}(t) = \left(G_{process}(q) - G_{model}(q)\right)u(t) + v(t) \qquad (4.5)$$

22

Thus the error consists of the model misfit and the output disturbance. By looking at the formula for V_{OE} we can conclude in general that the loss function decreases as the order of the model increases. For estimating de order of the model we only have to look at the reduction of V_{OE} when increasing the order of the model. As long as the reduction is significantly, it is worthwhile increasing the order of the model. Of course the model validation should not be done on the data set which has been used to estimate the model, because a good fit of the model on that data set will not prove sufficiently the model quality.

5 MPSSM model

5.1 Introduction

An MPSSM (Minimal Polynomial Start Sequence Markov) parameter model is estimated from a FIR model. There are two advantages of using a MPSSM model compared to a FIR or a (pseudo) conical model for the identification of a multivariable process, these are:

- The model doesn't need any information about the systems structure, just the degree (r) of the polynomial has to be determined;
- The number of parameters of a model of degree r with p inputs, q outputs and a D-matrix equation (5.1) is small compared to the number of parameters n_i needed for the description of the dynamics of the same systems of its Markov parameters only equation (4.1):

$$n_m = (r+1) \cdot p \cdot q + r$$

respectively:

$$n_i = (m+1) \cdot p \cdot q$$

with r: order of the minimal polynomial.

$$y_k = \sum_{i=0}^{\infty} F_i \left(a_j, m_j \big| j \varepsilon I\right) u_{k-1} \qquad I = 1,2,.....r \tag{5.1}$$

with F_i i^{th} Markov parameter $\dim(F_i)=q \times p$.

        a_j j^{th} coefficient of minimal polynomial.

Following the first advantage, difficulties with determination of the structure indices from the measured process signals which are required for the MIMO (pseudo) conical models are avoided.

A MPSSM model can be determined directly from the measured input and output data. For the computation of the model parameters, however, a high order function of the polynomial coefficients have to be minimized. For the mathematical explanation, we refer you to "Identification of an Industrial Process: A Markov Parameter Approach" of professor Backx dated 1987.

5.2 Mathematical description of the system

For the mathematical description of a MPSSM model we introduce a state space model:

24

$$x_{k+1} = F \cdot x_k + G \cdot u_k$$
$$y_k = H \cdot x_k + D \cdot u_k$$

(5.2)

with x_k state vector at time k   dim[x_k]=n;
F is state matrix   dim[F]=n × n;
G is input matrix   dim [G]= n × p;
H is output matrix   dim [H]= q × n;
D is direct feed through matrix   dim[D]= q× p;

From this state space model we can derive a description of the input output behaviour of the system:

$$y_k = H \cdot F^k \cdot x_0 + \sum_{i=1}^{k} H \cdot F^{i-1} \cdot G \cdot u_{k-i} = D \cdot u_k \qquad k \geq 0$$

(5.3)

with x_0 the initial vector of the state vector.

The model is assumed to be completely controllable and observable. In general, models obtained by process identification techniques will be completely observable. The reason is that the models are based on observed process response(s) on applied input signal(s) to the process.

Information present in the changes of the state of the system that can be observed at the outputs of the system is the only information needed for modelling the process.

A system can have non-controllable parts, such as if the initial state of the system is not equal to zero or if other inputs then the ones used for modelling (noise inputs) excite non controllable states. If we assume that the initial state of a system is equal to zero, the following expressions for the Markov parameter of the system can be obtained from equation (4.1) and (5.1).

$$g_i = \begin{cases} D & for\ i = 0 \\ H \cdot F^{i-1} & for\ i > 0 \end{cases}$$

The definition of the MPSSM model can be derived after an annihilating polynomial has been given. This annihilating polynomial is a scalar polynomial with the following satisfaction:

$$f(F) = 0$$

(5.4)

If we apply the Cayley-Hamilton theorem (ref. Gantmacher, 1959, implemented by professor Backx, 1987) in the annihilating polynomial, we get:

$$f_{cp}(F) = F^n + \alpha_1 \cdot F^{n-1} + \ldots + \alpha_{n-1} \cdot F + \alpha_n \cdot I = 0$$

There we use the power of the state matrix F and multiply this with matrixes H and G, there we get:

$$g_j = \sum_{i=1}^{n} -\alpha_i \cdot m_{j-i} \quad for\ all\ j > n$$

(5.5)

The state matrix F also satisfies its minimal polynomial, the recurrence relation for the Markov parameters, equation (5.5), can be reduced to:

$$m_{r+j} = \sum_{i=1}^{r} -a_i \cdot m_{r+j-i}$$

with   r the degree of the minimal polynomial;
        a_i the coefficients of the minimal polynomial;

$$m_{r+j} = m_{r+j}(a_i, m_j | i \varepsilon I) \qquad\qquad I = 1,2,3,\ldots,r$$

Now we can see that a complete description of the system only requests the first row r Markov parameters, the so called start sequence of Markov parameters, and the minimal polynomial coefficients that give the continuation of the sequences of the Markov parameters.

For further mathematical underlying of the Minimal Polynomial Sequence Markov parameters model we refer you to "Identification of an Industrial Process: A Markov Parameter Approach" of professor Backx dated 1987.

5.3 Determination the degree of the minimal polynomial

The determination of the degree of the minimal polynomial has to be done on the basis of the available input/output data and/or with already estimated Markov parameters. In de industrial practice, the process in general does not fit in the model set so that's a problem with the determination of the degree of the MPSSM model. The choice of the degree r for MPSSM models simply has to be based on considerations with respect to the ability of the models from that model set to simulate the measured process behaviour with certain accuracy. In industrial identification practice the model is not constructed on the basis of a theoretical knowledge of the process dynamics due to the complexity of the processes. Therefore, no direct relation can be found between the process to be modelled and the degree (r) of the model.

In practice two different approaches are used for determination of the degree (r), these are:

- Several models with different degrees of (r) are estimated and compared to the measured process behaviour in relations to output errors of the several models and the process;

- From the estimated sequence of Markov parameters consisting of the main part of the process impulse responses and order are determined by looking for the most dependences between the estimated parameters.

The output error of the first approach is constructed in the same way as with the FIR model en therefore the schematic representation of figure 4.2 (order selection by simulation) can be applied.

26

6 Model realisation

6.1 Introduction

At this point we have discussed the process, signal preparation for identification and two different types of models. Here we discuss the realisation of a model derived through the use of the treated, prepared signals and the use of "Modeler", an application of IPCOS technology. Through the use of the application "Modeler" there will be the option to derive a total of four different models, namely:

- Finite Impulse Response;
- Equation Error;
- Output Error;
- Subspace.

The students who follow the course TSA have been given the process as described in chapter 2_, "Model of a glass melting furnace derived from energy, mass and momentum equations and have treated, prepared the signals through the use of the m-files discussed in chapter 3, "Signal preparation for identification".

6.2 The variables

The variables used to derive a model from the process are already preset. This is because the assignment for the students who follow the course TSA is already in existence before the expansion and modification of the assignment and there for the "red line" of the assignment has been followed as it was. We can divide the variables into two mean categories, namely input variables and output variables also called manipulated and controlled variables within the application "Modeler". These variables can be further more divided:

- Input variables
    - o Electrical boosting under the batch, region UB; This variable has the dimension Watt.
    - o Electrical boosting in the centre and near throat, B12, B13 and NT; This variable has the dimension Watt.
    - o The connected surface temperatures of T1, T2 and NT. This variable has the dimension Degree.
- Outputs variables
    - o Temperature in region B21; This variable has the dimension Degree.
    - o Temperature in region B22; This variable has the dimension Degree.
    - o Temperature in region NT. This variable has the dimension Degree.

The different regions, addressed with the variables, are visualised in figure 2.2, chapter "Model of glass melting furnace derived from energy, mass and momentum equations".

The input variables are the changes on to the process set point and the output variables are the actual temperatures within the glass melting furnace.

6.3 Modeler; the application

Model is an application of IPCOS technology where models of a process can be obtained through the usage of data derived of logging the process variables. As described in the previous sub-chapter the variables are known and with the usage of the m-files all needed data of the signals is already present, prepared and in the right structure for Modeler.

For the students who follow the course TSA a manual has been written, "Coordinating User's Manual" where the outlining of Modeler and the usage of Modeler are described. The description of the usage of Modeler is only to point the students in the right direction and so student will have to develop the model also with the usage of the manual of Modeler which is on the installation cd-rom of the IPCOS products. The level of accuracy of the obtained model is depending on the students affords to derive the correct model from the data. This means students can not use only the information in of the manual "Coordinating User's Manual" to derive the correct model. The "Coordinating User's Manual" can be found on the cd-rom included with this report in the directory: "course TSA software\Installation Information Files".

7 Software architecture

7.1 Introduction

For the implementation of the glass melting furnace model in Simulink in a software environment with the Model Predictive Controller (MPC), this software environment with a number of different software applications must be developed. This environment enables easy transfer of the data between the software applications. Through the software environment with a preset of the applications, which must be a standard, the students who follow the course TSA can easily implement their model and develop their controller on the basis of MPC technology.

This chapter describes the application architecture, the software application used in the environment, the used configuration files for the software environment and the preset standard used for the environment with the standard for the applications.

7.2 Application architecture

The application architecture used to apply the MPC to the glass melting furnace model in Simulink is visualised in figure 7.1. In this architecture the MPC uses the model developed by the students. The students must develop the model with data derived through the signal preparation files, see chapter 3, within the INCA application "Model". The glass melting furnace model in Simulink will be actual the process for the students.
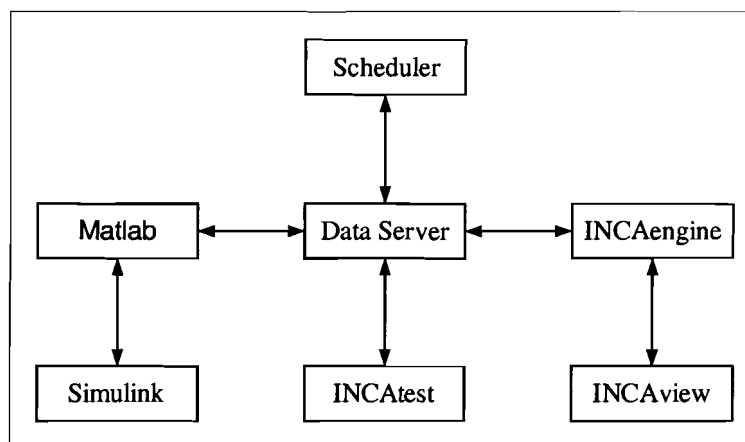
figure 7.1: Application architecture

The different applications within figure 7.1 will be briefly discussed:

Data server:
The INCAdataserver software form IPCOS Technology is a software program that serves as a gateway for process data. The data server connects the different applications to the glass melting furnace model, the actual process.

Scheduler:
The scheduler triggers the different applications to an active status in a defined sequence. The sequence as shown in figure 7.2 is the sequence used in the scheduler for the application architecture design for the course TSA.
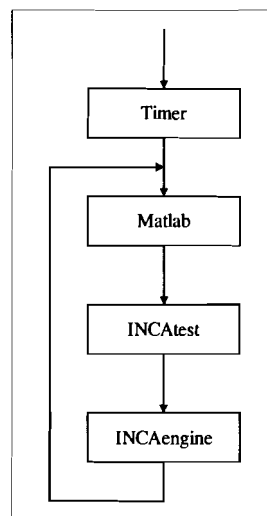
figure 7.2: Scheduler sequence.

Matlab/Simulink:
With Matlab a Simulink model of the glass melting furnace will be runned through an m-file every time Matlab is triggered. Simulink is an application of Matlab. The model of the glass melting furnace is here considered as the real process including the futures of a real process such as spikes.

INCAtest:
INCAtest is an application of the IPCOS technology used for logging and testing data. This application will enable visualisation of the signals present in the data server and logging of those signals. The logging of the data can be done in an ASCII format thereby simplifying the use of data within other applications for analyses and modelling.

INCAview:
The application INCAview of IPCOS technology visualises the variable present in INCAengine. The students can use this application for performance check of their developed controller, fine tuning of it.

INCAengine:
INCAengine of IPCOS technology is the software implementation of the MPC. This application contains all the optimization algorithms and calculates all the control signals. Advanced control systems based on prioritized and least squares control can be implemented with this application.

7.3 Configuration files

Through a set of configuration files we can set the INCA applications to a preset standard which students who follow the course TSA can easily use for their model implementation and controller design. The configuration files are comma separated and can be edited within Excel or any other text editor. Through the software architecture some INCA applications will use the same configuration files. The applications INCAview and INCAengine use the configuration file ControllerDefinition.csv, Data Server and INCAtest use VarDefDS.csv and the Scheduler uses RouteConfiguration.csv.

The contents of the used configuration files can be found on the cd-rom which is included with this thesis.

ControllerDefinition.csv

The controller has been configured through a set of tags within different sections of the ControllerDefinition.csv file.

The different sections can be defined into 3 sections, namely:
- Signal definition section;
- Controller section;
- Tuning section.

Within the signal definition section all in- and outputs are defined as well as the connections between the variables in the MPC and the data server. The controller section describes the settings for the controller and the tuning parameters for the variables in the MPC are defined in the tuning section.

VarDefDS.csv

The VarDefDS.csv file is like the ControllerDefinition.csv file configured through a set of tags within different sections. These sections are namely:
- Data signals section;
- INCA signals section;
- Simulator control section.

The data signals section tags are defined for the controlled, manipulated and disturbance variables and their properties. The INCA signals section tags define all start and stop configurations off the applications in relation to the data server. The last section, simulation control section, tags are defined to control Matlab/Simulink. Here actual process values are defined were the simulation within Simulink is depending on.

RouteConfiguration.csv

With this file the scheduler can define the sequence to active the applications. This is done by defining for each application the PrestartID and the AfterEndID (PrestartID for the first run of sequence and AfterEndID as a pointer to next application which must become active). No PreStartDelays and AfterEndDelays are used. The sequence used is visualized with figure 7.2.

7.4 Connect Matlab Simulink to the data server

The considered actual process is a simulation of a glass melting furnace within Simulink. The connecting made with the data server of INCA and Matlab is done with a specially design m-file "Data_Server_Link". By activating the m-file during the start up sequence of the scheduler the m-file will constantly wait for a sink from the data server and thereby running a part of its program. During this part actual stored data in the data server will be processed in the simulation of the glass melting furnace simulation and the data in the data server will be replace after the simulation with the new data of that simulation. Also visualisation of the simulation can be obtained through a scope which is build in the simulation thereby the students can follow the simulation as it was a real process.

7.5 Preset standard of the INCA applications

Through the configuration files, the designed m-files and Simulink files, the preset of the shortcut named "GlassmeltingFurnace" and the manuals (see next chapter for the shortcut and manuals) all applications can be preset to a standard were students who follow the course TSA can easily run all status and application as designed within the described software architecture.

8 User's software architecture implementation

8.1 Introduction

As described in the previous chapter software architecture has been designed to run the process and model parallel to each other and thereby designing the MPC. This all will be runned under a data server were also visualisation of signals is possible. To use this architectural software structure several manuals have been written to ensure that usage of this software will not be troubled by problems. These manuals are:
- Coordinating user's manual;
- Installation manual INCA and Matlab toolboxes;
- Installation manual customized software;
- User's manual customized software.

These manuals are supported by the standard manuals of INCA and Matlab. Here we briefly discus the layout of the designed manuals and the information within the manuals. The manuals can be found on the cd-rom which is included on the script in the directory : "course TSA software\Installation Information Files".

8.2 Coordinating user's manual

The outlining of the manual "Coordinating user's manual" is that it has as function to coordinate all other manuals as described as above and also extra information requiring the use of Modeler will be addressed. Information pointers toward the standard manuals of the INCA programs are also given.

8.3 Installation manual INCA and Matlab toolboxes

This manual will describe the installation of the IPCOS products, namely:
Installation IPCOS products:
- Requirements of target pc;
- Preparations;
- Actual installation;
- Installation log;
- Check on product versions;
- Product licensing;
- Check on licensed product (by dongle);
- Help files;
- Manual software requirements;
- Date/Time settings;

- Further system settings;
- Possible problems;

Uninstallation INCA:
- Uninstalling the IPCOS products;

Installation Matlab products:
- Requirements & installation;

8.4 Installation manual customized software

Within this manual two main directions can be defined, these are:
- Installation IPCOS software designed for course TSA;
- Installation Matlab software designed for course TSA.

The first direction will discuss the software description and installation and the properties of the shortcut for activating the Glass Melting Furnace designed software structure within INCA.

The second direction discusses the software description and installation where also the setting of an additional path within Matlab and the testing of the mxOPC.lic will be addressed.

8.5 User's manual customized software

The contents of the manual "User's manual customized software" provides a detailed view of the m-files and design requirements, the filter design and design properties, the m-files separately and the usage of the m-files separately. The m-files who are discussed including the usages of them are:
- GlobalSet.m;
- FreeRun.m;
- StepResponseEstimation.m;
- StairCase.m;
- TimeDelayCalculation.m;
- PRBNS.m

9 Conclusions and recommendations

9.1 Conclusions

The main conclusions that can be drawn from this project are:

- A model of a glass melting furnace derived from energy, mass and momentum equations was extended.
- The model will be approached by the students who follow the course TSA as the process to be modellated were the process has been projected as real as possible through the intentional corruption of types of "noise" such as input disturbances, white noise and spikes at the outputs.
- Through a series of designed m-files the students can design their own input signals for identification and thereby abstract the needed response signals. Combined with signal preparation options within the m-files without a lot of high programmer skills there for the m-files have a time redundancy factor.
- The m-files have a build in function to rewrite the structure of the signals store them so that they are immediately useable within the application Modeler, a product of IPCOS technology. Within the Modeler students can derive a suitable model of the process signals obtained from running the m-files.
- If the model, obtained with Modeler is stored at the proper directory of the used computer, simply by using the shortcut "GlassMeltingFurnace" a data server will be operational and hence the model and process are runned parallel of each other.
- A MPC can be designed. Modification of that MPC is part of INCAview, an application of IPCOS technology, including the visualisation of the signals, those of the model and those of the process. This process is again the model of the glass melting furnace derived from energy, mass and momentum equations including the realisation options.
- To install all the needed standard software and the designed software for the assignment of the course TSA manuals are rewritten supported by the standard manuals of the standard software. With these manuals the students are guided through the installation of the standard and the designed software up to the point of setting the shortcuts into the right properties. Also the manuals will guide the students through the different aspects of the assignment.

9.2 Recommendations

The result of this project are a process and model parallel runned from a data server combined with the design and tuning of a MPC. Further research in to the model abstractions methologies and the advantages and disadvantages of those results can be a recommendation.

Experimenting with other processes to model can give more insight into the complex theory behind the model predictive controller technology for the students who follow the course TSA at the University of Eindhoven. With this we recommend further development of new study cases where other processes are taken to model them and control them through the same principle described in this report.

References

[1]   Prof. dr. ir. Backx A.C.P.M.
        Identification of an industrial process: a Markov parameter approach.
        Eindhoven: Technical University Eindhoven
        1987

[2]   Prof. dr. ir. Backx A.C.P.M. & dr. ir. A.J.W. van den Boom
        Applied System Analysis
        Lecture Notes course
        October 200

[3]   Bitmead, Robert R.
        Gevers, Michel
        Wertz, Vincent
        Adaptive optimal control: the thinking man's GPC
        London: Prentice Hall, 1990
        ISBN 0-13-946823-4. -0-13-013277-2

[4]   Bosgra, O.H.
        de Jong, P.J.
        Verbruggen, H.B.
        Model Predictive Control: technieken, mogelijkheden en voordelen van MPC
        voor optimale procesbesturing: cursus Delft, 20-22 Maart 1995.
        Lecture Notes course in English
        's-Gravenhage: PATO, 1995

[5]   Ferziger, Joel H.
        Peric, Milovan
        Computational Methods for Fluid Dynamics.
        Springer
        ISBN 3-540-65373-2

[6]   Hof van den, Paul M.J.
        Van experimentele meetdata naar dynamisch model: methoden en interactieve
        programmatuur voor identificatie van dynamische systemen t.b.v. simulatie,
        voorspelling en regeling van industriele/fysische processen en mechanische
        servosystemen: PATO-cursus 18-20 September 2002.
        Den Haag: PATO Sectie Meet-, Regel- en Besturingstechniek, 2002

[7]   Kanjilal, Partha Pratim
        Adaptive prediction and predictive control
        London: Peter Peregrinus, 1995
        ISBN 0-86341-193-2

[8]   Huisman, Ir. L.
        Estimation of process variables in a glass melting furnace
        IFAC, 15th Triennail world congress 2002 Barcelona

[9]   Overschee van, Peter
        Moor de, Bart
        Subspace identification for linear systems. (Theory Implementation Applications).
        Dordrecht: Kluwer Academic Publishers, 1996

ISBN 0-7923-9717-7
[10]   Prof. Preisig, Heinz A.
       Modelling of dynamic systems
       1999
[11]   Rossiter, J,A,
       Model-based predictive control: a practical approach
       London: CRC Press, 2003
       ISBN 0-8493-1291-4
[12]   Zhu, Yucai
       Multivariable system identification for process control
       Amsterdam: Pergamon, 2001
       ISBN 0-08-043985-3

Symbols

Latin symbols

A_{ij}	transfer surface between i region and j region
C_p	heat capacity [J (kg.K)$^{-1}$]
$E_{f,i}$	friction losses in domain part i
g_2	gravitational acceleration (2 is the vertical direction) [ms^{-2}]
H_i	height of region i [m]
k_{eff}	effective heat conductivity
$\dot{K}_{tot}$	time derivative of total kinetic energy in the glass melt [J s^{-1}]
L_i	length of region i [m]
$\dot{m}$	mass flow rate [kg s^{-1}]
p	pressure
$\hat{q}_{wall,k}$	heat flow through the wall at region k [J s^{-1}]
R	backflow ratio
S_e	energy source term [J m^{-3} s^{-1}]
t	time [s]
T	temperature [K]
V	volume [m^3]
v	velocity [m s^{-1}]
v_{ij}	velocity in region ij [m s^{-1}]
v_1	velocity in main flow direction [m s^{-1}]
$\hat{w}_{pl}$	plunjer work [J s^{-1}]
W_i	width of region i [m]
B	bottom
BB	batch blanket
E	environment
FS	fuel supply
NT	near throat
T	top
UB	under batch
Y_p	process output without trend
Y_{tr}	trend signal
h^c	causal filter impulse response
h^{ac}	anti causal filter impulse response
$\tilde{s}_k$	clipped signal
s_k	measured signal corrupted with spikes

Greek symbols

α sign of a flow (-1 or 1 depending on the direction of the flow)

β thermal coefficient of volumetric expansion [K^{-1}]

ρ mass density [$kg\ m^{-3}$]

ξ_1 main flow direction coordinate [m]

ξ_2 vertical flow direction coordinate [m]

Φ potential energy [$kg\ m^2 s^{-2}$]

λ_{wall} thermal conductivity of wall

δ_{wall} thickness of wall [m]

γ_{wall} multiplication factor in friction term

μ viscosity [Pa.s]

Other symbols

Δ delta

Over lines

$\cdot$ indicates a time derivative

$\char`\^$ indicates a flow

Appendix 1; State space model

$$\dot{x}_1 = -\theta_{20}\theta_{21}\theta_{22}\theta_{23} - \theta_{23}\theta_{24}\theta_{26}(x_1 - \theta_{25})$$
$$+ \theta_{20}\theta_{26}x_{13}(x_{12} - x_1) - \theta_{26}\theta_{27}\theta_{28}(x_{12} - \theta_8)$$
(S1-1)

$$\dot{x}_2 = \theta_{67}[\theta_{49}(x_3 - x_2) - \theta_{50}(x_2 - x_1)] + \theta_1[\theta_2(x_5 - x_2) - \theta_3(x_2 - x_{11})]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_2 - x_1) - \theta_{27}\theta_{42}(x_2 - \theta_{30}) + \theta_{23}u_1$$
(S1-2)

$$\dot{x}_3 = \theta_{67}[\theta_{51}(x_4 - x_3) - \theta_{52}(x_3 - x_2)] + \theta_1[\theta_4(x_6 - x_3) - \theta_5(x_3 - x_{10})]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_3 - x_2) - \theta_{27}\theta_{42}(x_3 - \theta_{30}) + \theta_{23}u_2$$
(S1-3)

$$\dot{x}_4 = \theta_{67}[\theta_{53}(x_8 - x_4) - \theta_{54}(x_4 - x_3)] + \theta_1[\theta_6(x_7 - x_4) - \theta_7(x_4 - x_9)]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_4 - x_3) - \theta_{27}\theta_{42}(x_4 - \theta_{30}) + \theta_{23}u_3$$
(S1-4)

$$\dot{x}_5 = \theta_{67}[\theta_{55}(x_6 - x_5) - \theta_{56}(x_5 - x_1)] + \theta_1[\theta_8(\theta_{30} - x_5) - \theta_9(x_5 - x_2)]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_5 - x_1) - \theta_{27}\theta_{42}(x_5 - \theta_{30}) + \theta_{23}u_4$$
(S1-5)

$$\dot{x}_6 = \theta_{67}[\theta_{57}(x_7 - x_6) - \theta_{58}(x_6 - x_5)] + \theta_1[\theta_{10}(\theta_{30} - x_6) - \theta_{11}(x_6 - x_3)]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_6 - x_5) - \theta_{27}\theta_{42}(x_6 - \theta_{30}) + \theta_{23}u_5$$
(S1-6)

$$\dot{x}_7 = \theta_{67}[\theta_{59}(x_8 - x_7) - \theta_{60}(x_7 - x_6)] + \theta_1[\theta_{12}(\theta_{30} - x_7) - \theta_{13}(x_7 - x_4)]$$
$$-(1 + x_{12})\theta_{20}\theta_{29}(x_7 - x_6) - \theta_{27}\theta_{42}(x_7 - \theta_{30}) + \theta_{23}u_6$$
(S1-7)

$$\dot{x}_8 = \frac{1}{2}\theta_{20}\theta_{32}(1 + x_{12})(x_4 + x_8) + \frac{1}{2}\theta_{20}\theta_{32}(1 + x_{12})(x_7 + x_8)$$
$$-\theta_{27}\theta_{32}\theta_{33}(x_8 - \theta_{30})$$
(S1-8)

$$\dot{x}_9 = \theta_{67}[\theta_{61}(x_{10} - x_9) - \theta_{62}(x_9 - x_8)] + \theta_1[\theta_{14}(x_4 - x_9) - \theta_{15}(x_9 - \theta_{39})]$$
$$-x_{12}\theta_{20}\theta_{31}(x_9 - x_8) - \theta_{27}\theta_{42}(x_9 - \theta_{30})$$
(S1-9)

$$\dot{x}_{10} = \theta_{67}[\theta_{63}(x_{11} - x_{10}) - \theta_{64}(x_{10} - x_9)] + \theta_1[\theta_{16}(x_3 - x_{10}) - \theta_{17}(x_{10} - \theta_{40})]$$
$$-x_{12}\theta_{20}\theta_{31}(x_{10} - x_9) - \theta_{27}\theta_{42}(x_{10} - \theta_{30})$$
(S1-10)

$$\dot{x}_{11} = \theta_{67}[\theta_{65}(x_1 - x_{11}) - \theta_{66}(x_{11} - x_{10})] + \theta_1[\theta_{18}(x_2 - x_{11}) - \theta_{19}(x_{11} - \theta_{41})]$$
$$-x_{12}\theta_{20}\theta_{31}(x_{11} - x_{10}) - \theta_{27}\theta_{42}(x_{11} - \theta_{30})$$
(S1-11)

$$\dot{x}_{12} = \left[\theta_{20}^2\theta_{35}x_{12} + \theta_{20}^2\theta_{34}\left(1 + x_{12}\right)\right]^{-1}$$

$$\left(\theta_{20}\theta_{36}\left[\frac{1}{2}\left(x_8 - x_1\right) + x_{12}\left(x_8 - x_1\right)\right] - \theta_{20}^2\theta_{37}x_{12}^2 - \theta_{20}^2\theta_{38}\left(x_{12} + 1\right)^2\right)$$

(S1-12)

with:

$$\theta_1 = \frac{1}{\rho C_p \Delta\xi_2}$$

$$\theta_i = \frac{2k_{eff}\left(T_c\right)k_{eff}\left(T_n\right)}{\left[k_{eff}\left(T_c\right) + k_{eff}\left(T_n\right)\right]\Delta\xi_2}\bigg|_0$$

$i = 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19$

$$\theta_{20} = \hat{V}_{pull}$$

$$\theta_{21} = \phi_{BB}^{BB1}$$

$$\theta_{22} = \rho\Delta_R H$$

$$\theta_{23} = \frac{1}{\rho C_p}$$

$$\theta_{24} = \frac{k_{eff}\left(T_{UB}\right)_0 A_{UB|BB}}{\gamma_{UB}H_{UB}}$$

$$\theta_{25} = T_{BB1}$$

$$\theta_{26} = \frac{1}{V_{UB}}$$

$$\theta_{27} = \frac{\lambda_{wall}}{\rho C_p \delta_{wall}}$$

$$\theta_{28} = A_{UB|E}$$

$$\theta_{29} = \frac{1}{W_B H_B \Delta\xi_1}$$

$$\theta_{30} = T_E$$

$$\theta_{31} = \frac{1}{W_T H_T \Delta\xi_1}$$

$$\theta_{32} = \frac{1}{V_{NT}}$$

$$\theta_{33} = A_{NT|E}$$

$$\theta_{34} = \frac{\gamma_B^2 \rho V_B}{60 W_B^2 H_B^2}$$

$$\theta_{35} = \frac{\left(\frac{1}{4}\gamma_{T2}^2 + \gamma_{T1}^2\right)\rho V_T}{30 W_T^2 H_T^2}$$

$$\theta_{36} = \frac{\rho\beta g V}{WL}$$

$$\theta_{37} = \frac{\mu L_T}{2 W_T H_T^3}$$

$$\theta_{38} = \frac{\mu L_B}{W_B H_B^3}$$

$$\theta_{39} = T_{S,T1}$$

$$\theta_{40} = T_{S,T2}$$

$$\theta_{41} = T_{S,T3}$$

$$\theta_{42} = \frac{1}{W}$$

$$\theta_{43} = \frac{1}{W} + \frac{1}{\Delta\xi_1}$$

$$\theta_{44} = \Phi_{BB1}\Delta H$$

$$\theta_{45} = \frac{k_{eff}\left(NT\right)A_{NT|CS}}{\gamma_{NT}H_{NT}}$$

$$\theta_{46} = T_{NT,S}$$

$$\theta_i = \frac{2k_{eff}\left(T_c\right)k_{eff}\left(T_n\right)}{\left[k_{eff}\left(T_c\right) + k_{eff}\left(T_n\right)\right]\Delta\xi_1}\bigg|_0$$

$i = 49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66.$

$$\theta_{67} = \frac{1}{\rho C_p \Delta\xi_1}$$

Appendix 2; Flow chart m-file FreeRun

```
                              ┌─────────┐
                              │  Start  │
                              └────┬────┘
                                   │
                                   ▼
┌──────────┐   ┌───────────┐  Input<=zero  ◇ Enter ◇  Input=empty  ┌───────────┐   ┌──────────┐
│ Terminate│◄──│Error display│◄────────────│ simulation │─────────►│Error display│──►│ Terminate│
│Simulation│   └───────────┘              │   time    │           └───────────┘   │Simulation│
└──────────┘                               ◇─────◇                                 └──────────┘
                                             │ Input>zero
                                             ▼
                                    ┌─────────────────┐
                                    │    Set all      │
                                    │   simulation    │
                                    │  parameters of  │
                                    │    Simulink     │
                                    └────────┬────────┘
                                             ▼
                                    ┌─────────────────┐
                                    │    Simulate     │
                                    │ "run_nl_with_E  │
                                    │  1_Free_Run"    │
                                    └────────┬────────┘
                                             ▼
                                    ┌─────────────────┐
                                    │Calculate average Ts and│
                                    │ Standard Deviation of  │
                                    │     output data        │
                                    └────────┬────────┘
                                             ▼
                                    ╭─────────────────╮
                                    │    Display      │
                                    │   average Ts    │
                                    │  and Standard   │
                                    │   Deviation     │
                                    ╰────────┬────────╯
                                             ▼
                                        ◇ Enter ◇  Input=empty OR Input<=0 OR Input=undefined
                                        │ simulation │──────────────────────────────►
                                        │   time    │
                                        ◇─────◇                    ╭─────────────╮
                                          │ Input=Ts              │ Your choice is│
                                          │                       │   undefind    │
                                          │                       ╰──────┬────────╯
                                          │                              ▼
                                          │                       ◇ Enter ◇  Input=empty OR Input=<0 OR Input=undefined
                                          │                       │ simulation │──────────────────────────────►
                                          │                       │   time    │
                                          │                       ◇─────◇         ╭─────────────╮
                                          │                         │ Input=Ts   │ Your choice is│
                                          │◄────────────────────────┘            │ still undefined│
                                          │                                      ╰──────┬────────╯
                                          │                                             ▼
                                          │                                      ┌─────────────┐
                                          │                                      │ Original Ts will│
                                          │                                      │  be chosen.  │
                                          │                                      └──────┬──────┘
                                          │◄───────────────────────────────────────────┘
                                          ▼
                                    ┌─────────────┐
                                    │Resample Data│
                                    └──────┬──────┘
                                           ▼
     Input=No OR Input=Undefined   ◇ Do you want ◇  Input=Yes
   ◄─────────────────────────────── │ to filter? │ ────────────────►
                                     ◇──────◇
                                        │                    ┌─────────────────┐
                                        │                    ▼                 │
                                        │         Filter  ◇Filter or◇          │
                                        │        ◄────────│ Filter1 │ Input=Empty OR Input=undefined
                                        │                 ◇────◇───►
                                        │                   │ Filter 1
                                        ▼                   ▼
                             ┌───────────────┐    ┌───────────────┐
                             │ Filter output │    │ Filter output │
                             │ data with filter│  │data with filter1│
                             └───────┬───────┘    └───────┬───────┘
                                     │◄───────────────────┘
                             ◄───────┘
                             │
                             ▼
                          ⬡ (connector)
```

Calculate upper and low border output region 5

Plot signal region 5 and upper and low border in figure and pause

End pause?

No

Yes

Close figure and calculate upper and low border output region 6

Plot signal region 6 and upper and low border in figure and pause

End pause?

No

Yes

Close figure and calculate upper and low border output region 8

Plot signal region 8 and upper and low border in figure and pause

End pause?

No

yes

Close figure

```
Display all
simulation
parameters, user
choices, upper and
lower borders, drifts
of output signals
```

```
Calculate real time
all sample times
```

```
Rewrite all input and
output data into right
format for Modeller
```

```
Store all rewritten
data in c:\
Data_For_Modeler\
FreeRun
```

```
End
```

Appendix 3; Flow chart m-file PRBNS.m

Start

Enter time of PRBNS signal — Input=empty *OR* Input<=0

Input=not empty *OR* Input>0

Display:"Total simulation time will be 3 x time PBRNS signal"

Enter switching probability of PBRNS signal input 1 (between 0 & 1) — 0>Input *OR* Input>1

0=< Input=<1

Enter switching probability of PBRNS signal input 2 (between 0 & 1) — 0>Input OR Input>1

0=< Input=<1

Enter switching probability of PBRNS signal input 3 (between 0 & 1) — 0>Input *OR* Input>1

0=< Input=<1

Enter maximum amplitude PBRNS signal input 1 — Input=empty *OR* Input=0

Input=not empty

Enter maximum amplitude PBRNS signal input 2 — Input=empty *OR* Input=0

Input=not empty

Enter maximum amplitude PBRNS signal input 3 — Input=empty *OR* Input=0

Input=not empty

Compiling input
signals and set
all simulation
parameters of
Simulink

Simulate
"rxm_nl_with_E
1_PBRNS"

Calculate average Ts and
Standard Deviation of input
and output data

Display average
Ts and Standard
Deviation of input
and output data

Enter
simulation
time

Input=empty *OR* Input<=0 OR Input=undefined

Input=Ts

Your choice is
undefind

Enter
simulation
time

Input=empty *OR* Input<=0 *OR* Input=undefined

Input=Ts

Your choice is
still undefined

Smallest Ts will
be chosen

Resample Data

Input=No *OR* Input=Undefined

Do you want
to filter?

Input=Yes

Filter

Filter or
Filter1

Input=Empty *OR* Input=undefined

Filter 1

Filter output
data with filter

Filter output
data with filter 1

Plot all PBRNS
output signals

Display all
simulation
parameters, names of
input and output
signals

Calculate real time
all sample times

Rewrite all input and
output data into right
format for Modeller

Store all rewritten
data in c:\
Data_For_Modeler\
PBRNS.

End

Appendix 4; Flow chart m-file StairCase.m

No

Estimate
response?

Yes

Display:
Which output
do you want
to estimate?

Output
number? Input= not (1 OR 2 OR 3)

Input=(1 OR 2 OR 3)

Plot the complete
response of the
selected output and
pause

End
pause? No

Yes

Teller=0

Close figure

Teller=Teller+1

Plot the response of
the selected output
part (Teller) and
pause

End
pause? No

Yes

Close Figure

Display all simulation parameters, estimated parameters, names of input and output signals

Calculate real time of all sample times

Rewrite all input and output data into right format for Modeller

Store all rewritten data in c:\ Data_For_Modeler\ StairCase.

End

Appendix 5; Flow chart m-file StepResponseEstimation.m

Start

Select input — 1>Input>3

Input=1 OR 2 OR 3

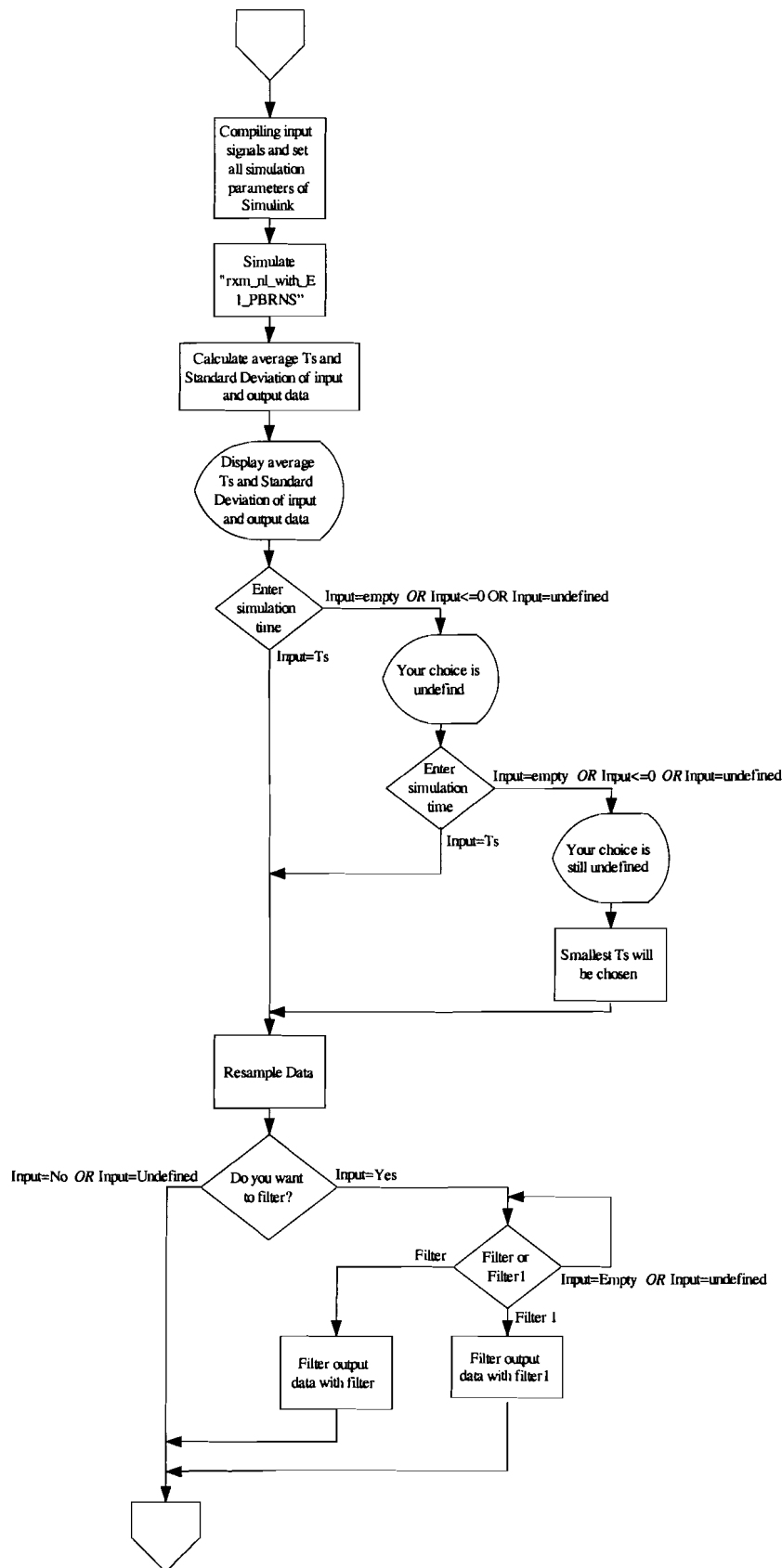Select Output — 1>Output>3

Output=1 OR 2 OR 3

Enter Start Time — Input=empty → Error display → Terminate Simulation

Input= not empty

Terminate Simulation ← Error display ← Input=<Start Time — Enter Stop Time — Input=empty → Error display → Terminate Simulation

Input= not empty

Terminate Simulation ← Error display ← Input=0 — Enter Amplitude step — Input=empty → Error display → Terminate Simulation

Input= not empty

Set all simulation parameters of Simulink

Simulate "rxm_nl_with_E 1_Step_Re-sponse_Fitting

Calculate average Ts and Standard Deviation of input and output data

Display average Ts and Standard Deviation of input and output data

55

```
Close figure
```

```
Display all
simulation
parameters, estimated
paramteres, names of
input and output
signals
```

```
Calculate real time
of all sample times
```

```
Rewrite all input and
output data into right
format for Modeller
```
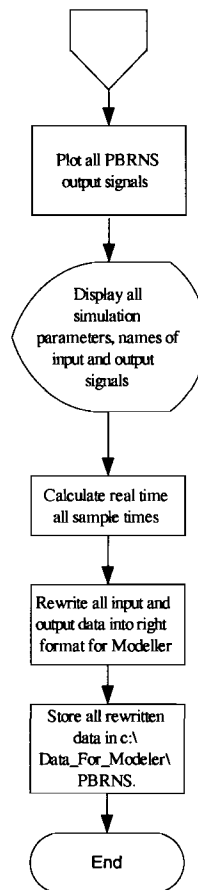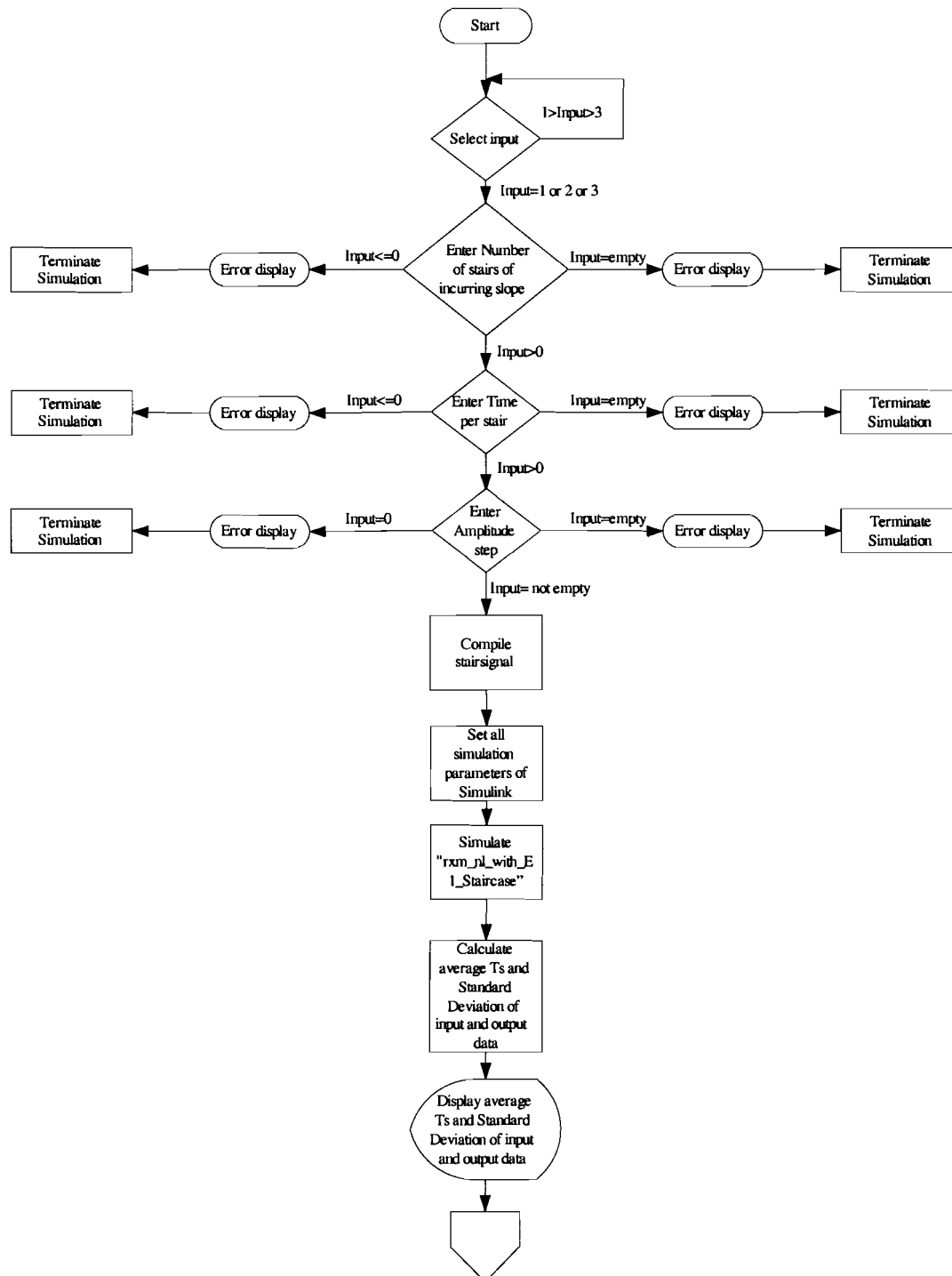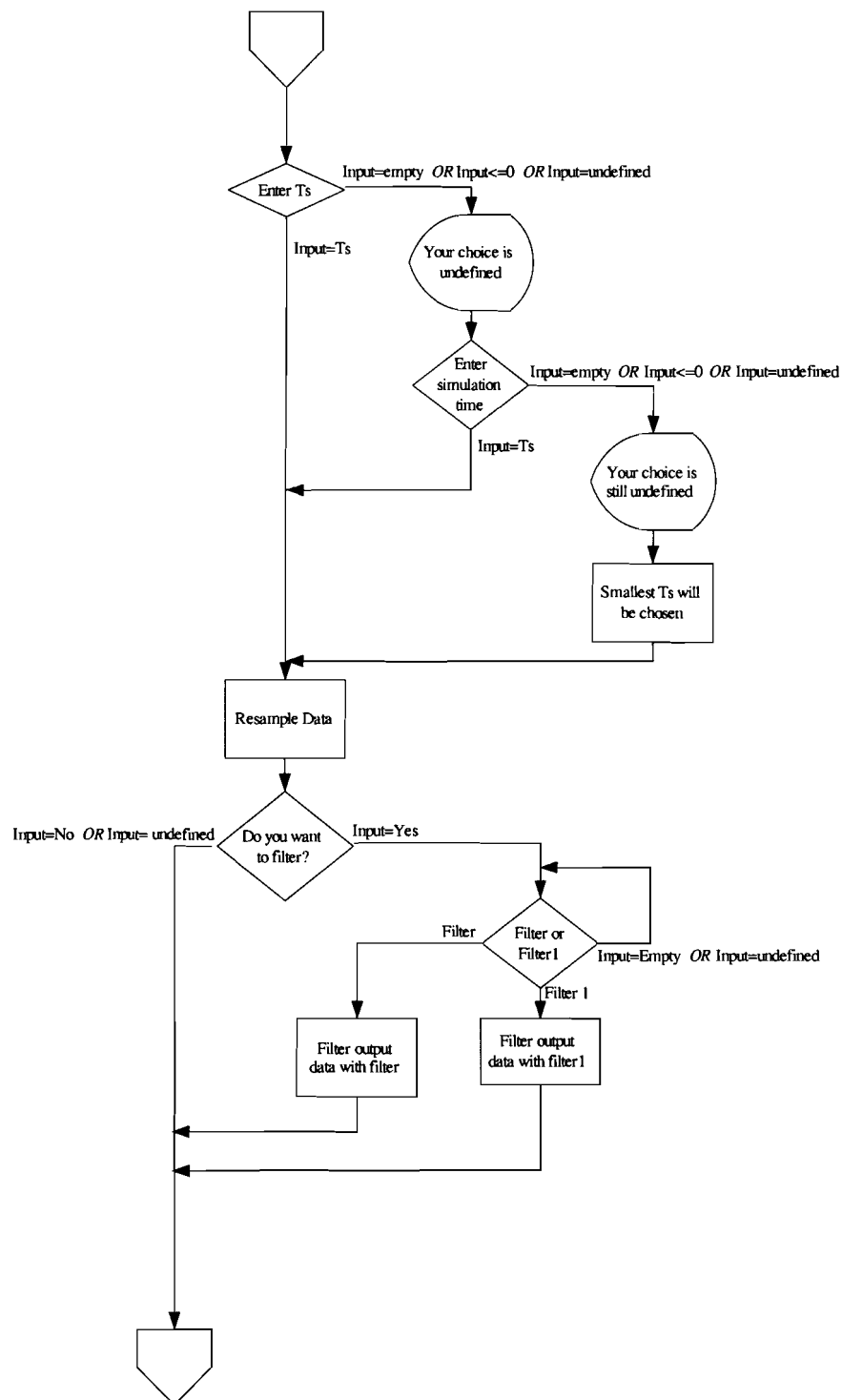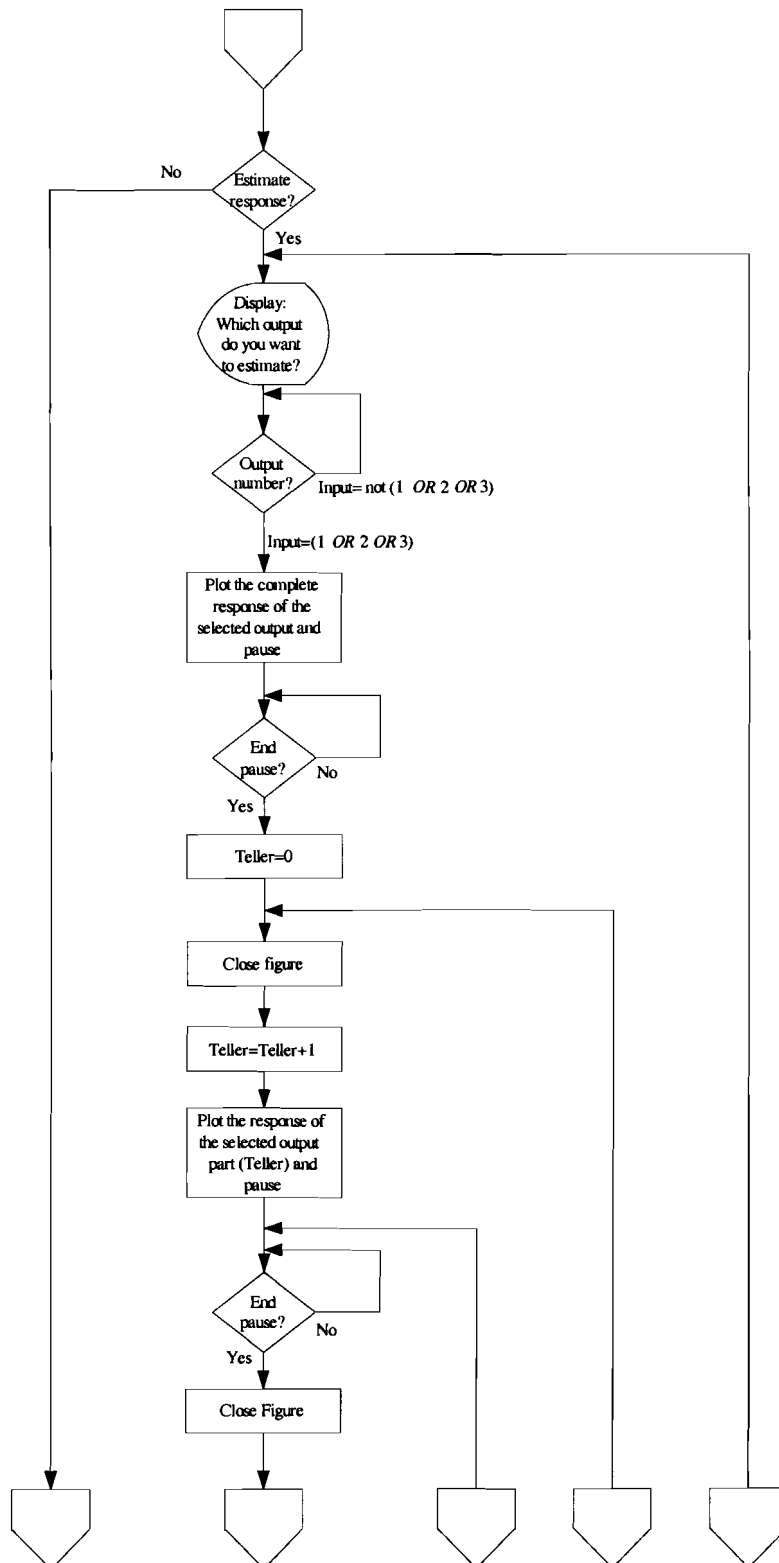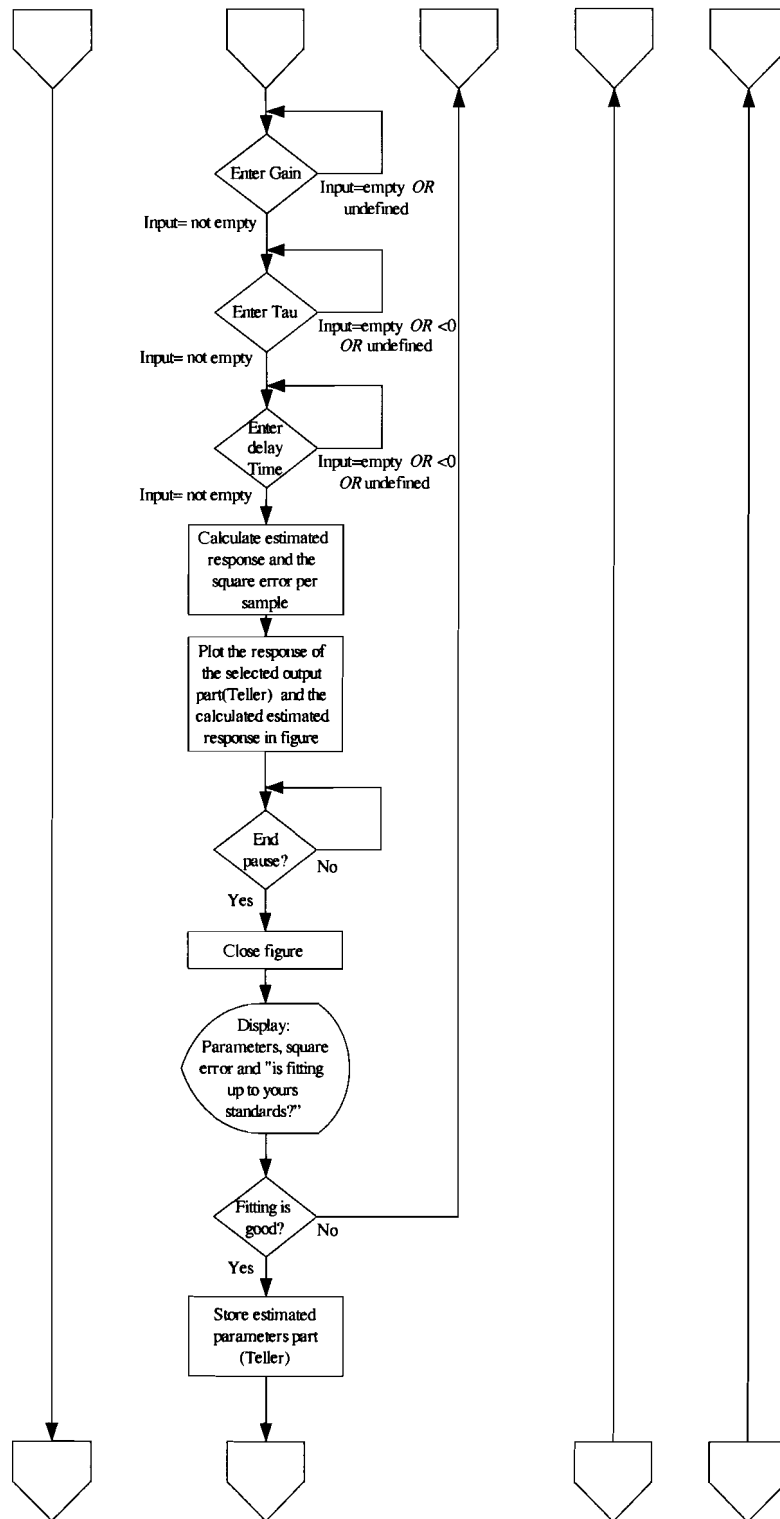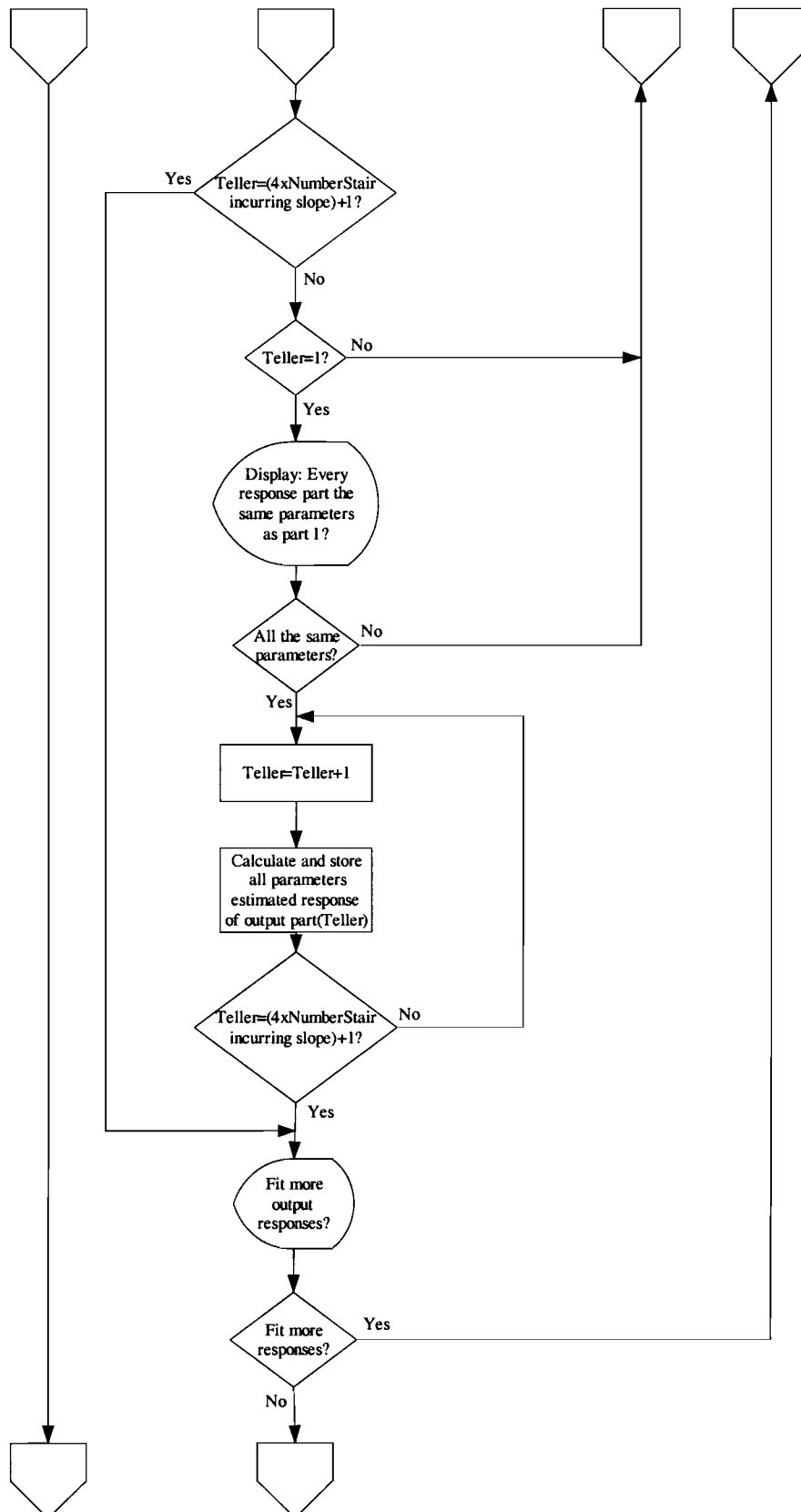
```
Store all rewritten
data in c:\
StepResponse.
```

```
End
```

Appendix 6; Flow chart m-file TimeDelayEstimation.m

Start

Select input — 1>Input>3 OR Input=undefined

Input=1 OR 2 OR 3

Set all simulation parameters of Simulink

Simulate "run_n1_with_E1_Delay_time_calculation"

Calculate average Ts and Standard Deviation of input and output data

Display average Ts and Standard Deviation of input and output data

Enter simulation time — Input=empty OR Input<=0 OR Input=undefined

Input=Ts

Your choice is undefind

Enter simulation time — Input=empty OR Input=undefined OR Input<=0

Input=Ts

Your choice is still undefined

Smallest Ts will be chosen

Resample Data

Input=No OR Input=Undefined — Do you want to filter? — Input=Yes

Filter — Filter or Filter1 — Input=Empty OR Input=undefined

Filter 1

Filter output data with filter

Filter output data with filter1

59

Calculate the cross
correlation of
selected input data
and data region 5.

Plot the cross
correlation of
selected input data
and data region 5.

End
pause?

No

Yes

Close figure and
calculate the cross
correlation of
selected input data
and data region 6.

Plot the cross
correlation of
selected input data
and data region 6.

End
pause?

No

Yes

Close figure and
calculate the cross
correlation of
selected input data
and data region 8.

Plot the cross
correlation of
selected input data
and data region 8.

End
pause?

No

yes

Close figure

Display all
simulation
parameters, names of
input and output
signals

Calculate real time
all sample times

Rewrite all input and
output data into right
format for Modeller

Store all rewritten
data in c:\
Data_For_Modeler\
Time_Delay_
Calculations.

End