The identification of the dynamics of a floating platform

Pronk, K.

Award date:
1988

5241

DEPARTMENT OF ELECTRICAL ENGINEERING
EINDHOVEN UNIVERSITY OF TECHNOLOGY
Group: Measurement and Control

THE IDENTIFICATION OF THE DYNAMICS OF A FLOATING PLATFORM

by K. Pronk

Report of a final project
performed between August 1987 and March 1988
in charge of prof.dr.ir. P. Eykhoff
under supervision of
dr.ir. A.A.H. Damen and dr.ir. A.J.W. van den Boom

Summary

This report contains the results of a final project performed by K. Pronk between August 1987 and March 1988 in the group Measurement and Control of the Department of Electrical Engineering of the Eindhoven University of Technology.

The task to perform was to find a simulation model of a laboratory model of a floating platform. The floating platform is a practical example of a multi-input-multi-output (MIMO) system. The simulation model will be used to design a controller to stabilize the platform.

To determine a model of the platform several software tools were used based upon output error methods. Final results were obtained from the programs DIRECTO [Oudbier,1986] which estimates a MPSSM-model and LS_SSM [Veltmeyer,1985] which determines a state space model. The simulations using the DIRECTO-output were improved after implementation of an explicit offset estimator in the program DIRECTO. Finally both simulations performed very well and very small output errors were obtained.

Looking more closely to the structure of the model it is possible to recognize the structure of the process in the model. The process is very symmetric: it consists of three almost identical subsystems that each include one float. Each input mainly excites one of those subsystems which is confirmed by the fact that the estimated input matrix of the model is block diagonal dominant. The symmetry of the system can be recognized in the structure of the system matrix as well. Each float causes a single pole pair but, because the subsystems are almost identical, it is very difficult to distinguish all the states if not all inputs are used. This causes a too low order estimation in the single-input-multi-output (SIMO) experiment.

Both the good simulation results and the structure of the model that reflects the structure of the actual process give good confidence in the accuracy of the model obtained.

Table of contents

Introduction

This report presents the results of a final project for obtainment of the masters degree at the Department of Electrical Engineering of the Eindhoven University of Technology.
The work was carried out at the group Measurement and Control. In this group a main research subject is System Identification and Parameter Estimation, especially of multi-input-multi-output (MIMO) systems. As a result of the research efforts in this field several model representations have been formulated. During the last years several software-packages were developed to estimate models and model-order of MIMO-systems.

In this report the identification of such a MIMO-system, a floating platform, is described. Several program packages were used for this purpose and, if necessary, improved. This report contains a complete description of the data collection, the data preprocessing (filtering) and the data processing i.e. the identification of the process. Based on the data collected in the experiments a mathematical model of the process is estimated. Finally the structure of the model is analysed and compared with physical knowledge of the real process.

This report and the estimation results could not have been accomplished without the help of the members of the group Measurement and Control. I want to thank everybody that has been helpfull to me during my final project. In particular I want to thank dr.ir. A.A.H. Damen, dr.ir. A.J.W. van den Boom and ing. J.W.J.J. Beckers.
Finally I thank Marina Roijakkers for assisting me in making the plots for this report.

1. Description of the process

1.1. General system description

The system of our concern is a laboratory model of a platform. The platform is floating on three floats. The position of the floats with respect to the platform can be controlled using three servomotors. A schematic drawing of the system is given in Fig.1.

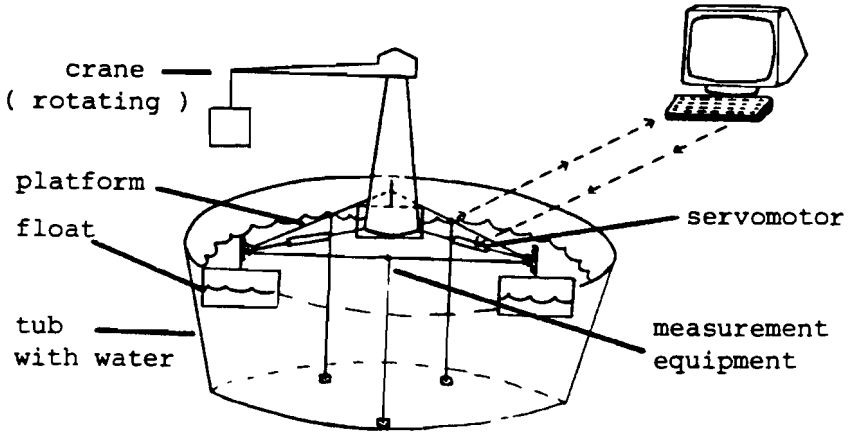


Fig.1 The floating platform.

By moving the floats vertically we try to eliminate any disturbances of the balance of the platform. To perform this task only two control inputs should be sufficient. The third control input is only strictly necessary if we want to choose a certain height.
In reality the main causes of disturbance could be mass-displacement, waves and wind. To eliminate the effects of these disturbances we use a personal computer to determine the control signals $\underline{u}_c$ for the servomotors.
As outputs we measure the heights of three points of the platform with respect to a fixed reference plane using roller-potentiometers as described by Dirks [Dirks,1986]. The output voltage of the roller-potentiometer is linearly related to the length of the string which is connected to the platform and to the bottem of the bassin . Neglecting for the moment the influence of translation and rotation of the platform in the horizantal plane the length of the string will be the height of the platform near the measure-point. In par.1.3. we will consider the effects of the movements in the horizontal plane as well as other drawbacks of the measurement equipment.

The output signals $\underline{y}$ are used as input signals for the controller when the system is operated in closed-loop configuration. A block-diagram of both the open-loop configuration and the closed-loop configuration is given in Fig.2a/b.

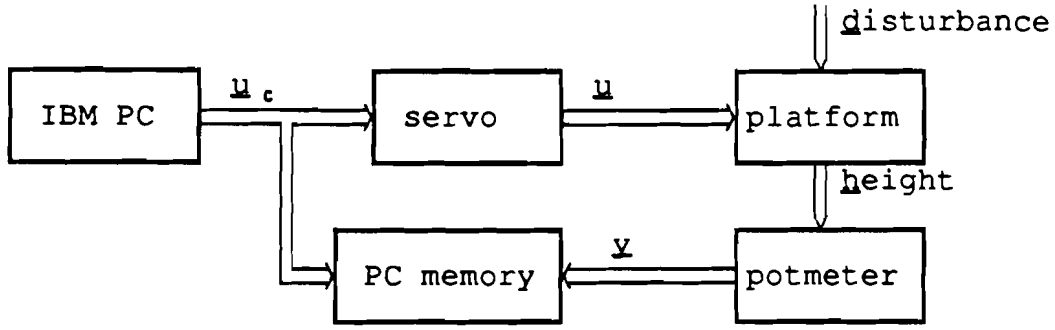


Fig.2a. Block-diagram of the open-loop configuration.

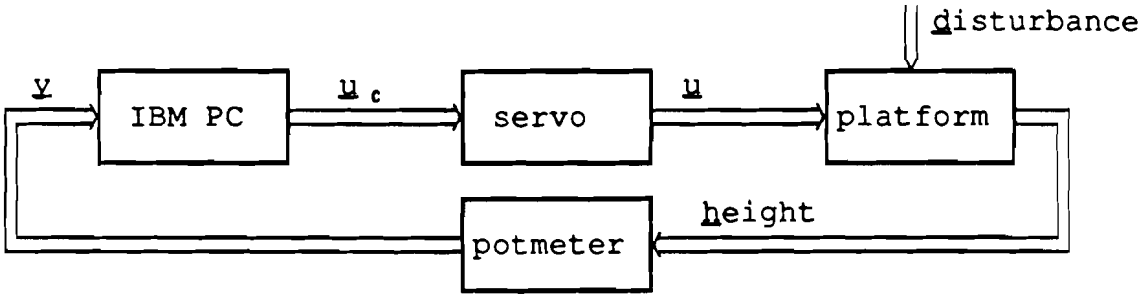


Fig.2b. Block-diagram of the closed-loop configuration.

Clearly the system is a MIMO-system with three inputs (the input signals of the servomotors) and three outputs (the three height measurements).
To design a proper controller it is necessary to obtain a good mathematical model of the process. This model is obtained by stochastic identification methods. For identification purposes we use the open-loop configuration.
Before starting the stochastic analysis however we want to gain some more insight in the expected behaviour of the system. For this reason we start by determining a rough estimation of the model based on physical knowledge. The structure of this theoretical model (e.g. the order of the system and the eigenvalues) will be used for model validation.
To be able to analyse the system theoretically we use the symmetry of the system. We distinguish three single-input-single-output (SISO) subsystems as described in the next paragraph and suppose that the behaviour of every subsystem will be approximately equal.

1.2. A SISO-subsystem

The SISO-subsystem is schematically drawn in Fig.3. In this subsystem there is only one control input and only one height measurement.

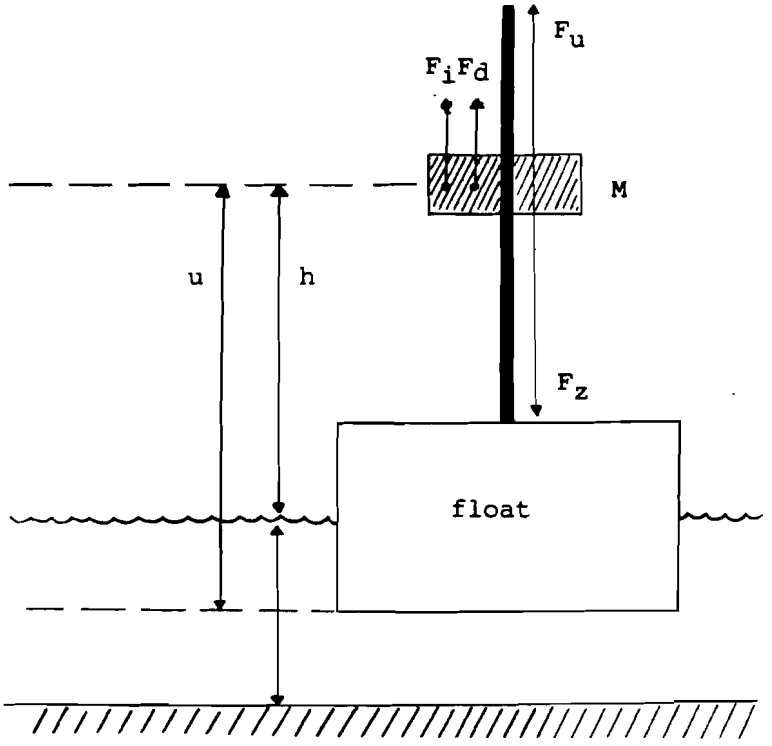


Fig.3. A SISO-subsystem.

The block-diagram of Fig.2. is still valid for this subsystem. An important insight into the behaviour of the total system can be obtained by studying the subsystem. In this subsystem there exist four important forces: the gravitation, the upward force of the water, a damping force and the inertia force. When no disturbances exist a balance between these forces will be reached determining the height of the platform with respect to the bottom of the bassin (h) and with respect to the bottom of the float (u).
The signal u is temporarily regarded as the input-signal and the signal h is the output signal.
To determine the transfer function between u and h we first consider the direction and the magnitude of the mentioned forces:

Fz: The gravitation.
This force is always directed downwards.
Its magnitude is Mass * Gravitation acceleration.

Fu: The upward force.
This force is always directed upwards.
Its magnitude is k(h-u) provided that the float is not completely immersed in the water and not on the bottem of the bassin, k being a parameter dependent on the size of the float.

Fd: The damping force.
This force only exists when the float is moving. In that case the direction of the force is always opposite to the direction of the movement.
Its magnitude is $D(\dot{h} - \dot{u})$, D being a parameter.

Fi: The inertia force.
This force only exists when the movement of the mass M is accelerating (positive or negative). In this case the direction of the force is opposite to the direction of the acceleration.
The magnitude of the force is Mass * $\ddot{h}$.

Next we determine the condition for balance between these forces. In this case the sum of the forces must be zero. Using the forces and their directions as described we obtain the following differential equation:

$$M\ddot{h} + D\dot{h} + kh = D\dot{u} + ku - Mg \qquad (1)$$

As mentioned in paragraph 1.1 a mass-displacement, and thus a change of effective mass of one subsystem, is considered to be a disturbance. To determine the transferfuction between input and output however the mass M is being supposed a constant. This minor simplification does not cause significant inadequacies [Daanen, 1985].
Performing Laplace-transformation we obtain the following equivalent relation in the s-domain:

$$h(s) = \frac{Ds + k}{Ms^2 + Ds + k} * u(s) \qquad (2)$$

Finally we consider the input signal. Until now the distance between the platform and the bottom of the float is used as input signal. The input signal of the complete subsystem is in fact the input voltage of the servomotor. To determine the transferfunction between the input voltage and the height we have to consider the transferfunction between the input voltage and the distance u. This transferfunction was determined using a step-function as input signal u_c and measuring u. The result of this measurement is plotted in Fig.4.

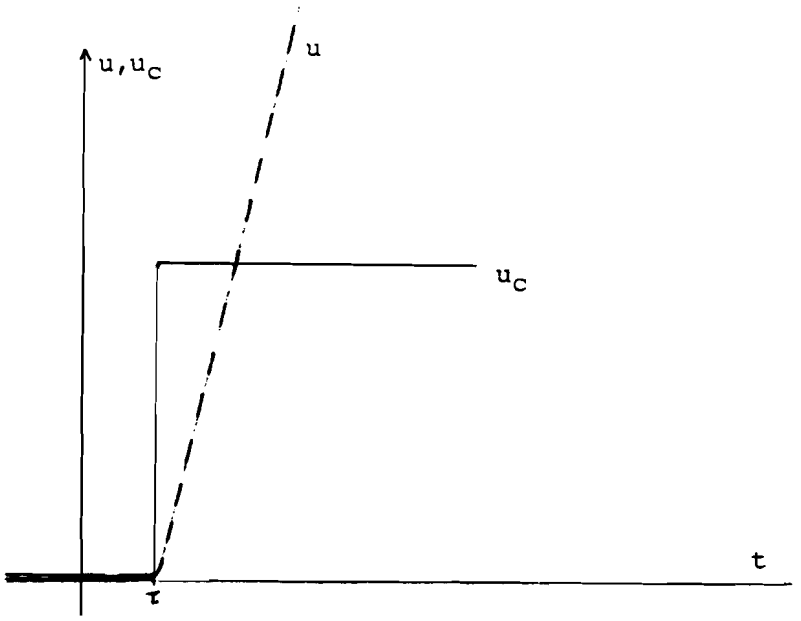


Fig.4. Experiment to determine the transferfunction of the servomotor.

It was concluded that the servomotor can be modelled best by a pure integrator. This integrator is put in cascade with the transferfunction of equation (2). The simplified block-diagram of the subsystem now becomes:

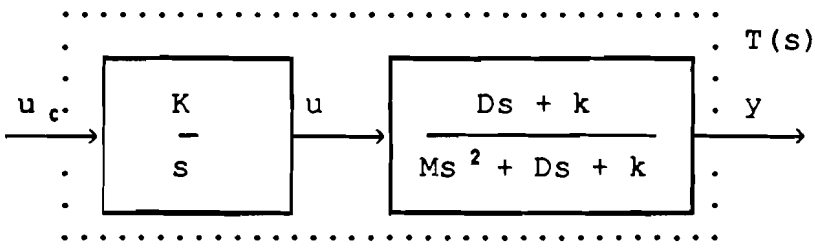


Fig.5. Block-diagram of a SISO-subsystem.

The total transferfunction between input and output is:

$$T(s) = \frac{K(Ds + k)}{s(Ms^2 + Ds + k)} \qquad (3)$$

In the next paragraph we will have a closer look at the output signals. Several drawbacks of the measurement equipment used until now will be discussed and an alternative will be briefly mentioned.

1.3. The height-measurements

As mentioned before a lot of time has been devoted by other students and their coaches to solve the measurement-problem in a cheap way. The setting of the problem looks quite simple: determine the absolute height of (three points of) the platform with respect to a fixed reference plane, in this case the bottom of the basin. Several techniques have been discussed. Until now best proposal of an algorithm to determine the height was introduced by Dirks [Dirks, 1986].
To determine the three output-signals h he used six distance-measurements. The distance of six points with respect to fixed reference points was measured using roller potentiometers. In this way he obtained six independent measurement signals. The floating platform has six degrees of freedom (3 translations and 3 rotations) so six measurement signals are sufficient. Dirks developed an algorithm to determine the three heights from the six measurement signals.
In practice however the implementation of this algorithm proves to be too slow to determine the heights in one sample moment. This will be a problem when the algorithm is used to determine the control signals. A second problem is that the algorithm proves to be not very reliable. Very often the algorithm does not converge. This, of course, also affects the identification because no proper height measurements can be calculated.
To cope with this problem we want to measure the height without interferingly measuring rotation and translation of the platform at the same time. Clearly using the measurements as sketched in Fig.1. this condition can not be met. A rotation or a translation of the platform will change the length of the string and will be regarded as a change in height, cf. Fig.6.

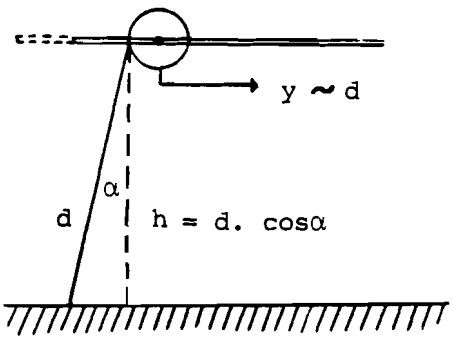


Fig.6. Misfit between the height and the length of the string.

From Fig.6. it can be concluded that rotation or translation of the platform will add a non-linear drift component to the output signals. Another serious drawback of the measurement equipment recently used is the fact that the elastic wires affect the behaviour of the system. They have a certain influence on the dampingsconstant D and they add a

springconstant.
At the moment new measurement equipment has been built. This equipment uses conduction measurements between two metal plates connected to the platform to determine the height of the platform. The new measurement equipment is not sensitive for translation or rotation of the platform. Because this new equipment is implemented only very recently and is not yet tested all the results in this report were obtained using the roller potentiometers as sketched in Fig.1. A drawback for open-loop identification using the new measurement equipment seems to be that this equipment is only linear in a limited working area which is not large enough to obtain samplestrings of sufficient length.
Because the new measurement equipment does no longer use the wires connected to the bottem is will be necessary to obtain a new model. This model will represent the system dynamics better (assuming the measurement equipment will prove to perform well during testing) because no longer the dynamics of the platform are influenced by the measurements or rotation and translation.

2. Model representations

2.1. Introduction

This chapter will briefly deal with the model representations used during the identification of the floating platform.
First two generally known model representations are described: The state space model in observability canonical form and the Markov parameters.
Next a system description will be given that is less well-known. The description is based on a start sequence of Markov parameters and the minimal polynomial coefficients. This representation is especially attractive because no structure indices are needed apart from the degree of the minimal polynomial.
At the end of this chapter the relations between the model representations are discussed and algorithms to convert the representations into each other are briefly mentioned.

2.2. State space model

The state space model is based on the definition of several states between input and output of the system. A state is a memory element (an integrator or a delay) whose contents are stored in the state-vector $\underline{x}$. From this definition it is clear that the minimum number of states (the minimum dimension of the state space) equals the order of the system. A state space model describes the relationships between inputs, outputs and states according to the following convention:

$$\begin{cases} \underline{x}(k+1) = A\,\underline{x}(k) + B\,\underline{u}(k) \\ \underline{y}(k) = C\,\underline{x}(k) + D\,\underline{u}(k) + OFF \end{cases} \qquad (4)$$

The model is characterized by the matrices [A,B,C,D,OFF] which is called a realisation. This is not a unique representation; for a particular combination of input- and output signals an infinite number of vectors $\underline{x}$ can be found leading to different realisations for the same process:
If [A,B,C,D,OFF] is a realisation of a particular process, then for each non-singular matrix T the realisation $[T^{-1}AT, T^{-1}B, CT, D, OFF]$ results in the same input/output behaviour. This means that for using algorithms for identification we have to choose one particular type of the set of state space models to guarantee unique results. Some well known (pseudo) unique forms are the Jordan canonical form, which will be discussed later, and the pseudo-canonical observability form [Damen, 1986].
Both realisations are attractive for their clear relations between inputs and/or outputs and the states. Moreover they contain a small number of parameters. For multi-output systems the observability form is unique but a number of structure indices have to be estimated. In the pseudo canonical or overlapping form most systems can be represented in a fixed structure, so that in this representation the structural indices are not estimated based on data but fixed beforehand. Only in rare cases the system will not fit in the modelset.

2.3. Markov parameters

The Markov-parameter model is based on the impulse response of the system. When the impulse-response of a causal SISO-system is called m(k) then the system response for a general input-signal u(k) can be written as:

$$y(k) = \sum_{j=0}^{\infty} m(j).u(k-j) \qquad (5)$$

This description is easily extended to the MIMO-case:

$$\underline{y}(k) = \sum_{j=0}^{\infty} M(j).\underline{u}(k-j) \qquad (6)$$

$\{M(j)\}_{j=0,\infty}$ is a sequence of matrices, called the Markov parameters, that provides the various impulse responses. The sequence of Markov-parameters consists of q.p sequences of impulse responses, where q is the number of outputs and p the number of inputs.
A Markov-parameter model is an input-output model: The output variables are written explicitly in terms of the input variables. This model is unique.
Another great advantage during identification is that the impulse response model is linear in the parameters. The number of parameters however is infinite.
This remark requires some discussion: In case of finite-dimensional systems the number of Markov parameters can be reduced at the cost of the linearity in the parameters. This will be dealt with in the next paragraph.

2.4. Minimal polynomial start sequence Markov parameters (MPSSM)

In this paragraph we have a closer look at the Markov sequence as described in the last paragraph. If we consider a first order SISO process we know that the impulse response obeys the following relation:

$$m(k) = a.m(k-1) \qquad k > 1$$

This relation can easily be extended for finite-dimensional higher order MIMO-processes:

$$M(k) = \sum_{j=1}^{r} a(j).M(k-j) \qquad k > r \qquad (7)$$

This means that a finite dimensional system is completely determined by $M(0), \{a(i), M(i)\}_{i=1,r}$. The coefficients a(i) are called the minimal polynomial coefficients and the set {M(i),i=0,r} is a start sequence of Markov-parameters. Together they form the MPSSM description of a process.
r is the degree of the minimal polynomial. If there are no poles with geometric multiplicity more than one this r equals the n indicating the dimension of the minimal state space realisation. In the sequel we will put r=n and mainly use n.
Clearly the big advantage of this description is the small amount of parameters without being restricted to a particular structure. A disadvantage is that the linearity in the parameters is lost in this representation.
More information about MPSSM-models can be found in [Backx,1987] and [Oudbier,1986]

2.5. Relationships between the model representations

The relation between the state space model and the impulse response model can be shown in a simple way by combining both equations of Equ. (4) :

$$\underline{y}(k) = \sum_{j=1}^{k} C.A^{j-1}B\,\underline{u}(k-j) + D\,\underline{u}(k) + C.A^{k}\underline{x}(0) + OFF \qquad (8)$$

from which it follows (neglecting initial conditions):

$$M(k) = \begin{cases} C.A^{k-1}B & k > 0 \\ D & k = 0 \end{cases} \qquad (9)$$

To determine a state space realisation of a given impulse response model is somewhat more complicated. Note that again we can choose one particular realisation. The algorithm to calculate such a realisation is called the Ho-Kalman algorithm:

1. Construct a Hankel matrix H:

$$H = \begin{bmatrix} M(1) & M(2) & . & M(n) \\ M(2) & & . & . \\ . & . & & . \\ M(n) & . & . & M(2n-1) \end{bmatrix} = \begin{bmatrix} CB & CAB & . & . & CA^{n-1}B \\ CAB & & & . & . \\ . & & . & & . \\ CA^{n-1}B & . & . & & CA^{2n-2}B \end{bmatrix}$$

$$= \Gamma . \Delta$$

$$\Gamma = \begin{bmatrix} C \\ C.A \\ . \\ . \\ C.A^{n-1} \end{bmatrix} \qquad \Delta = [\, B \quad B.A \quad . \quad . \quad B.A^{n-1} \,]$$

Γ is called the observability matrix.
Δ is called the controllability matrix.

2. Shift the Hankel matrix one column to the left:

$$\overleftarrow{H} = \begin{bmatrix} M(2) & . & . & M(n+1) \\ M(3) & & . & . \\ . & . & & . \\ M(n+1) & . & . & M(2n) \end{bmatrix}$$

Note that the matrix $\overleftarrow{H}$ is equal to the product of the observability matrix Γ, the system matrix A and the controllability matrix Δ:

$$\overleftarrow{H} = \Gamma.A.\Delta \quad \Rightarrow \quad A = \Gamma^{+}.\overleftarrow{H}.\Delta^{+}$$

where + indicates the peudo inverse.
supposing Γ and Δ have full rank and M is long enough.

3. Decompose H using singular value decomposition:

$$H = W.\Sigma.V^{T}$$

W and V being orthonormal matrices and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, σ_i being the singular values of H in descending order.

4. Choose $\Gamma = W.\Sigma^{1/2}$ and $\Delta = \Sigma^{1/2}.V^{T}$ (symmetric realisation)
The conditions for Σ are now satisfied.

5. Calculate A,B,C using $A = \Gamma^{+}.\overleftarrow{H}.\Delta^{+} = \Sigma^{-1/2}.W^{T}.\overleftarrow{H}.V.\Sigma^{-1/2}$
Note that $D = M(0)$.

Next we consider the rather straightforward relationships between the **Markov parameters** and the **MPSSM model**.
When given a MPSSM-model the Markov parameters can be calculated as follows:

$$M(k) = \begin{cases} M(k) & \text{if} \quad k \leq n \\ \sum_{i=1}^{n} a(i).M(k-i) & k > n \end{cases} \qquad (10)$$

When given a set of Markov parameters the MPSSM-model can be calculated as below:

$$\begin{cases} M(k) = M(k) & \text{if } k \leq n \\ \underline{a} = G^{+}.\underline{m} \end{cases} \qquad (11)$$

$$\underline{a} = - [\, a(1), \ldots, a(n) \,]$$

$$G = \begin{bmatrix} \mathrm{vec}(M_1) & \mathrm{vec}(M_2) & \ldots & \mathrm{vec}(M_r) \\ \mathrm{vec}(M_2) & \cdot & \ldots & \mathrm{vec}(M_{r+1}) \\ \cdot & & & \\ \cdot & & & \\ \mathrm{vec}(M_{m-r}) & \cdot & \ldots & \mathrm{vec}(M_{m-1}) \end{bmatrix}$$

$$\underline{m} = [\, \mathrm{vec}(M_{r+1})^T \; \mathrm{vec}(M_{r+2})^T \ldots \mathrm{vec}(M_m)^T \,]$$

$$\mathrm{vec}(M_i) = [\, M_{11}(i), M_{21}(i), \ldots, M_{q1}(i), M_{12}(i), \ldots, \ldots, M_{q2}(i), \ldots, M_{qp}(i) \,]^T$$

The relationships between the state space model and the MPSSM model can be derived using combinations of the techniques mentioned above. To construct a state space model from a MPSSM model first convert the MPSSM model into Markov parameters and then apply the Ho-Kalman algorithm.
Generically a MPSSM model converted to state space will result in a dimension of the state space n = r.min(p,q) as multiple poles are allowed. To reduce the order we can use either of the following methods [Backx,1987]:

We construct a matrix $\tilde{H}$ that is as close as possible to the Hankel matrix H in the Frobenius norm. The rank of H however is r:

$$H_{err} = \min_{\mathrm{rank}\tilde{H} = r} | H - \tilde{H} |_F$$

Using the matrix $\tilde{H}$ in stead of H we apply the Ho-Kalman algorithm as described above.
In general, however, it can not be guaranteed that the matrix $\tilde{H}$ has the required Hankel structure. This means that, to be able to apply the Ho-Kalman algorithm, we may be forced to choose another matrix which is not the best in the Frobenius norm.

Another possibility is to reduce the order by deleting the

least controllable and the least observable part of the state space model. To solve this problem the state space model has to be transformed into a balanced realisation after which the mentioned parts can be removed. An algorithm to determine the balanced realisation is given by Backx.

A state space model can be tranformed into a **MPSSM model**: the start sequence of Markov parameters can be determined as explained above, the minimal polynomial coefficients are the coefficients of the characteristic equation (if no multiple poles occur):

$$\det (zI - A) = 0 \qquad (12)$$

3. The identification

3.1 Introduction

After having dealt with several model representations we are now ready to start with the actual experiments to determine a mathematical model of the process. The mathematical model is calculated iteratively comparing during each iteration the calculated output $\hat{y}$ with the measured output y. The error between the two signals is calculated according to a predefined error criterion. In this case we use an output-error criterion because a complete simulation model is needed for the given control-method. The output error is calculated as the sum of the squared residuals:

$$V_0 = \sum_{k=0}^{N} e_0^2(k) = \sum_{k=0}^{N} (y(k) - \hat{y}(k))^2 \qquad (13)$$

where N is the number of data pairs $[\underline{u}(k), \underline{y}(k)]$
This error is minimized using the sequence of programs shown in Fig.7.

First the program ESMARK determines a finite Markov sequence based on the measured data. The length of the Markov sequence to be determined is entered by the user. In HANKEL a Hankel matrix with Markov parameters is constructed as described in the Ho-Kalman algorithm. The number of singular values of this matrix not equal to zero is equal to the order of the process (when no noise is present). Examining the singular values of the Hankel matrix the user enters the wanted dimension of the reduced state space realisation. A symmetric realisation is constructed using the Ho-Kalman algorithm for the truncated Hankel matrix.
Optionally the Markov-parameters are reconstructed for the given order using Equ.(9).
In the right hand branch these reconstructed Markov-parameters are used by DIRSTA to determine the initial minimal polynomial coefficiens needed in DIRECTO. Because we use the reconstructed Markov sequence (number of independent Markov-parameters = n) for this purpose Equ.(11) can be reduced to:

$$\underline{a}^T = H^{-1}.\underline{m}_n \qquad \underline{m}_n = [\, M(n+1), M(n+2), \ldots, M(2n) \,]^T$$

Finally DIRECTO will determine an MPSSM model using an output error criterion. This is the last step of the error-minimisation. To compare the results of DIRECTO with the results of LS_SSM the order of the MPSSM model has to be reduced to n using one of the methods given in par.2.5.
At the moment however none of the mentioned algorithms to obtain a reduced order state space realisation is implemented yet. Trying to obtain a reduced realisation using HANKEL again we find a non-optimal

realisation according to an output error criterion. The simulation results in this report are obtained generating a finite Markov string (order = r, l=100) from the MPSSM model which is used for simulation. In the left hand branch the state space realisation obtained by HANKEL is transformed by PCAN in a pseudo canonical observability form which is used as initial model in LS_SSM. LS_SSM will estimate a state space model in pseudo canonical observability form using an output error criterion.
For more detailed information about the programs the reader is referred to Appendix A and the manuals of LS_SSM [Veltmeyer, 1985] and DIRECTO [Oudbier, 1986]

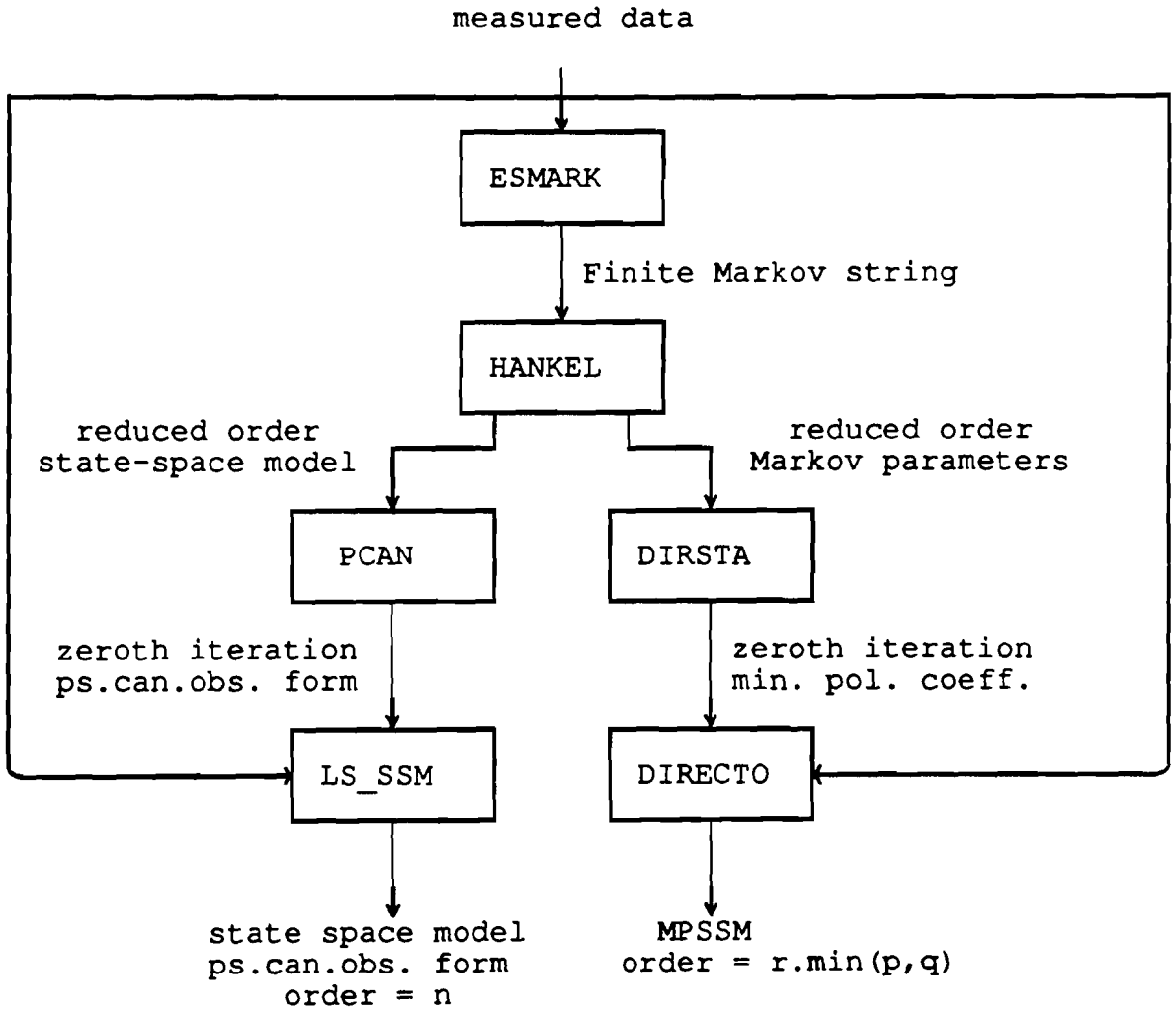


Fig.7. Programs used for identification.

3.2 Data collection and preprocessing

Before starting the experiments we have to make some choices to obtain data that are suitable for identification.
First of all the input signal should be sufficiently rich to excite all the states of the system. A proper choice for input signal is a uniform white noise sequence generated by the random-generator of Turbo Pascal used on IBM or Philips XT.
Next we have to choose the sample-frequency. To do so we determine the rise time of the process experimentally. When we choose the sample time at least 5-15 times as small as the rise time the signal contains enough information to estimate the dynamics of the process [Isermann,1980]. For identification purposes often one samples with a higher rate because then the preprocessing can be done with higher reliability. In our case the rise time is approximately 1 s. Because internal delays of about 0.02 s. are present it makes no sense to use extreme high sample rates. We chose the sampletime to be 0.1 s.
Now we are ready to start the experiments. Before using the obtained data for identification, however, we first have to perform prefiltering. The most common types of filtering are offset-filtering, dead-time filtering ,scaling and low-pass filtering.
To remove the dead-time of the system we first have to have a good estimation of the dead-time. This estimation is obtained using cross-correlation techniques. In this case the delays are small with respect to the sampletime, so we do not use dead-time filtering.
To perform offset-filtering it is necessary to have a good estimation of the offset. It is common use to remove the average value of the signal. However in general the average value of a signal is not equal to the offset of the signal. Especially in this case when the transfer function contains a pure integrator it is clear that it is impossible to remove the offset without removing a part of the information as well. For this reason we do not use this type of offset-filtering. Instead we use explicit offset identification. During the experiments however proper identification turned out to be very difficult when the output signals did not start from zero. For this reason the signal was shifted so that $\underline{y}(0)$ became $\underline{0}$ (i.e. performing offset-filtering using $\underline{y}(0)$ as offset-vector). During identification { OFF - $\underline{y}(0)$} is determined instead of the offset-vector.
Whether it is useful to perform low-pass filtering depends on the signal-noise ratio. When this ratio is very good there is no need for low-pass filtering and it could even remove a part of the information in the data. In our case, however, the quantisation noise sometimes became too large. This causes small negative real quantisation poles. Especially for DIRECTO these poles became problematic because DIRECTO does not allow poles in the left half z-plane. In general the existence of poles with a negative real part can be avoided by increasing the sample-frequency. This will however not affect the location of quantisation poles ($f_q = 0.5 * f_{samp}$). For this reason we used lowpass filtering. The low-pass filtering is performed by the following operation:

$$y'(k) = (y(k-1) + y(k) + y(k+1)) / 3$$

The cut-off frequency off this type of low-pass filtering is 1/3 * f_{samp} which is enough to remove the quantisation noise. Averaging only two samples would increase the bandwidth but at the same time introduce a phase shift that would cause an additional (delay) pole. The last type of preprocessing we want to consider is scaling the signals. Both LS_SSM and DIRECTO minimize the total absolute error between the signals and the simulations so a signal containing very little energy is discriminated compared to a signal containing more energy. This will result in larger relative errors in the low-power signals. By scaling we can avoid this discrimination at the cost of a higher absolute error. Because the absolute error is most important for control tasks we decided not to scale the signals.

Next we will use some special knowledge of the process to make the data more suited for identifiction. First of all we know that a pure integrator is present in the system. A pure integrator is very difficult to estimate because of convergence-problems for allmost unstable modes. To avoid problems during identifiaction we integrate the input signal before starting the identification. During the identification the integrated input signal is regarded as the input signal. After identification an additional integrator has to be put in cascade with the calculated model to obtain the complete model cf. Fig.8.

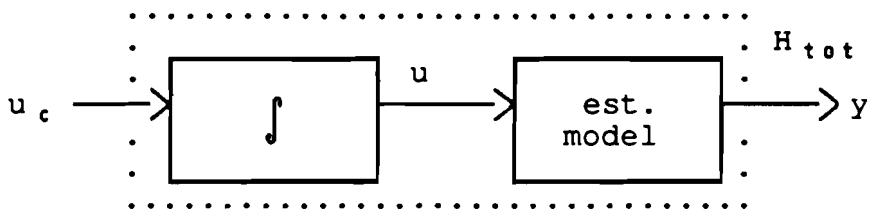


Fig.8. Dividing the model into two blocks before identification.

Finally we have to perform a last transformation of the output data to be able to compare the results with the theoretical results in Chapter 1. In the theoretical model the relation between the input signal u and the height of the float h is determined. The measured output signals however are not the heights of the floats but the heights inbetween two floats (cf. Fig.1). It can easily be seen that the heights of the floats can be calculated as below:

$\underline{h} = E \cdot \underline{y}$ $\underline{y}$ being the measured output

$$E = \begin{bmatrix} 1.0 & -1.0 & 1.0 \\ 1.0 & 1.0 & -1.0 \\ -1.0 & 1.0 & 1.0 \end{bmatrix}$$

This transformation should be performed before identification because it affects the calculation of the output error; the rotation causes a different weighting of the residues:

$$V_o = \sum_k (E\underline{y} - E\underline{\hat{y}})^T (E\underline{y} - E\underline{\hat{y}}) = \sum_k (\underline{y} - \underline{\hat{y}})^T E^T E (\underline{y} - \underline{\hat{y}})$$

3.3. The SISO-experiment

After having discussed the general structure of the process based on physical laws, having dealt briefly with several types of black box models and having considered the need of proper filtering before identification we now start the actual parameter estimation of the platform.
We start just considering a SISO-subsystem as shown in Fig.9.:

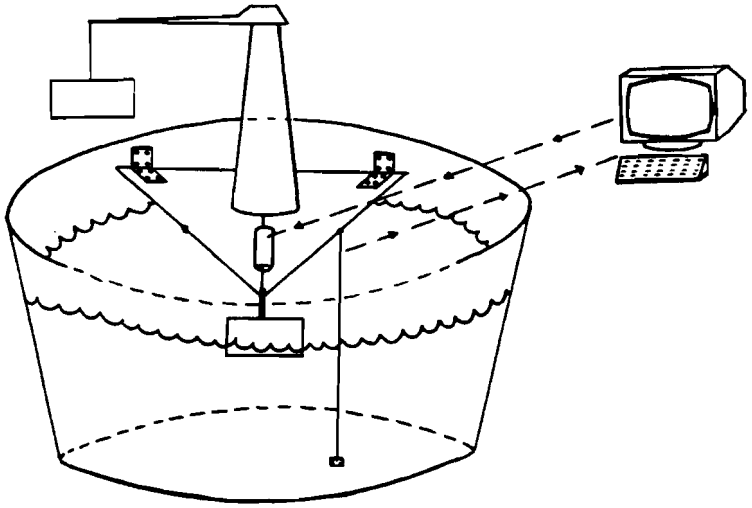


Fig.9. The SISO-experiment.

In this case only one servomotor is used and the height of only one float is measured. The other floats are taken out of the water temporarily and are fixed to the bassin. Note that in this configuration the system has only one degree of freedom and the height-measurement is no longer disturbed by translation or rotation of the platform.
The uniform zero-mean white noise input signal as described in par.3.2. is generated by the IBM-computer and applied to the servomotor.
The program that handles the input-generation and data collection and storage is written in Turbo Pascal. The program for the SISO-configuration is called MEETSISO.PAS. The output data collected by MEETSISO are pictured in Fig.10.
Horizontally the time in samplemoments is plotted. The sampletime is 0.1 s. Vertically the output samples are plotted in digital representation. This representation is obtained converting the output voltage of the potentiometers (-5V,5V) to an eight bit digital reprentation (0-256) and subtracting 128 to obtain a zero on output when the potentiometer generates 0V. In all the following plots showing output measurements or simulations the same unities are used.

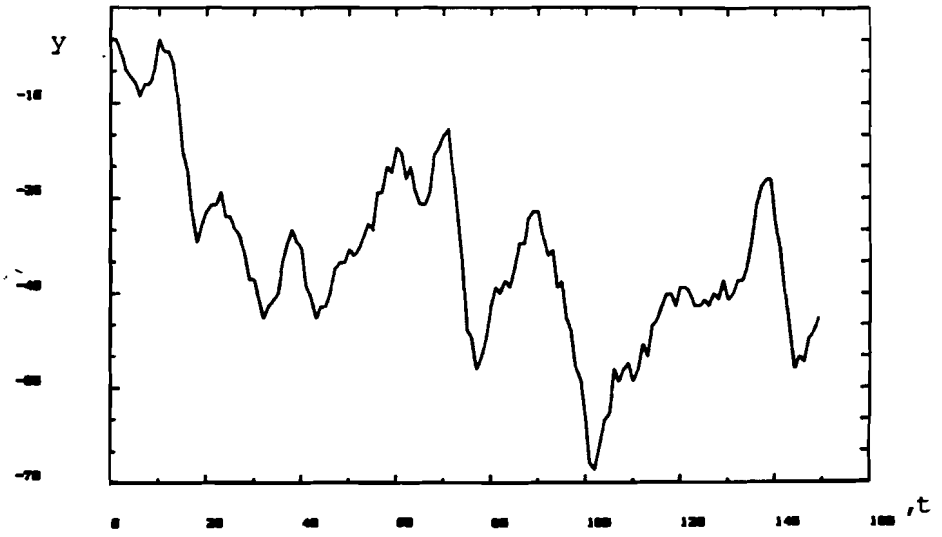


Fig.10. The output of the SISO-experiment.

The data pictured in Fig.10 will now be filtered and transformed as described in paragraph 3.2. After preprocessing we obtain the red line plotted in Fig.12. The filtered sequence is used to determine a mathematical model of the process. A number of programs was used according to the flow-diagram pictured in Fig.7. Intermediate and final results are discussed in this paragraph.
To determine an initial model we first estimate a sequence of Markov parameters using the program ESMARK. The wanted length of the Markov-sequence can be supplied by the user. We choose the length l=20. The amplitude of the 20th Markov parameter is more than 95% reduced compared to the first Markov parameters so the greatest part of the signal power is included in the first 20 Markov parameters. A simulation based on the estimated Markov-parameters is pictured in Fig.12 (blue line). Indeed the greatest part of the information is included in the model.
According to the Ho-Kalman algorithm we can construct a state space model using the finite Markov sequence. First we determine the order of the system considering the singular values of the Hankel-matrix plotted in Fig.11.

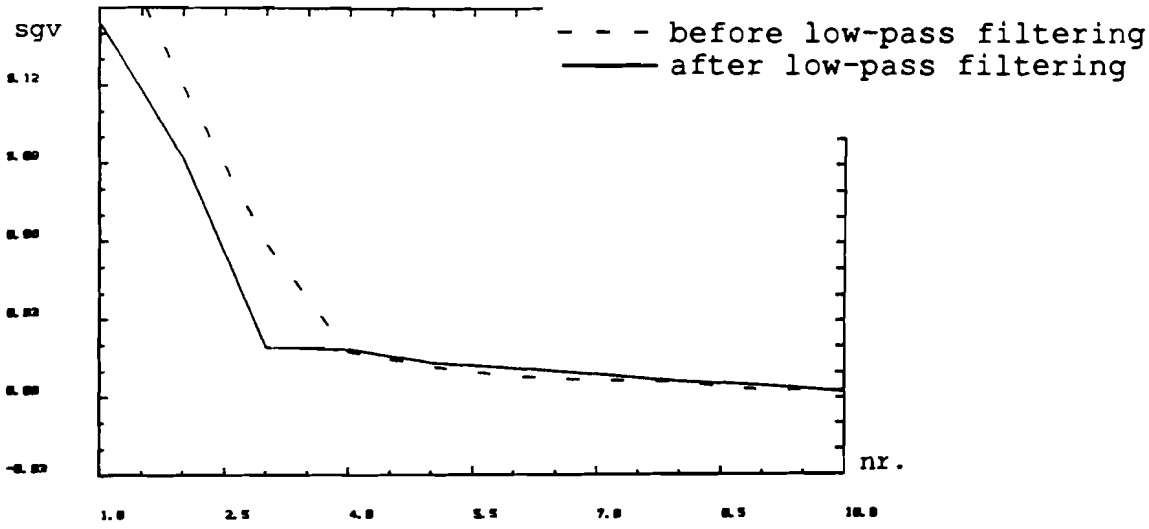


Fig.11. Singular values of the Hankel matrix.

A clear edge can be detected near the third singular value. This suggests that the magnitude of the third singular value has reached the noise level. When no noise is present this singular value would become zero if there are no more independent Markov parameters which means that the order of the system is reached. However when the signal is corrupted by noise the singular values will never become zero. Instead the noise level can be used to determine the order of the process. When a singular value enters the 90%-bound of the noise level we decide we can not distinguish more independent Markov-parameters. As an illustration in this plot also the singular values of the Hankel matrix before low pass filtering are plotted. We see that the quantisation noise increases the estimated model order. Moreover the magnitude of each singular value is increased with the square root of the quantisation noise power.
Rough calculations were performed to determine the noise level. Unfortunately the calculations did not match with the noise level indicated by the edge in Fig.11. Because the calculations are not very accurate we define from Fig.11 the order of the system (= the number of independent Markov-parameters) by two. The misfit of the noise level calculations and the results in Fig.11 may be caused by the fact that calculation of the noise level is based on assumptions (uncorrelated noise samples in Hankel matrix) which are not valid in this situation. To determine a second order state space realisation the program HANKEL was used. This program constructs a Hankel matrix according to the Ho-Kalman algorithm and calculates a symmetrical realisation. Using this state space realisation a new simulation was performed the results of which are plotted in Fig.13 (blue line).

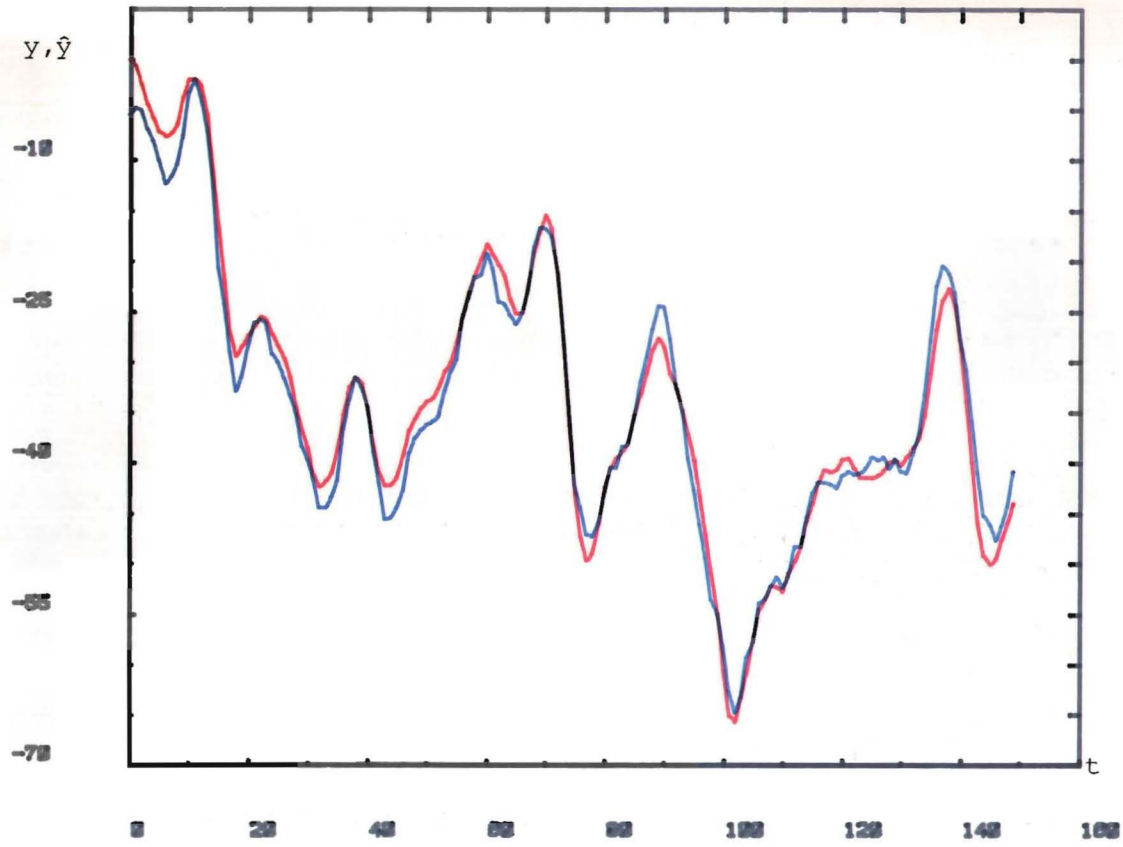


Fig.12. SISO-simulation using ESMARK output.

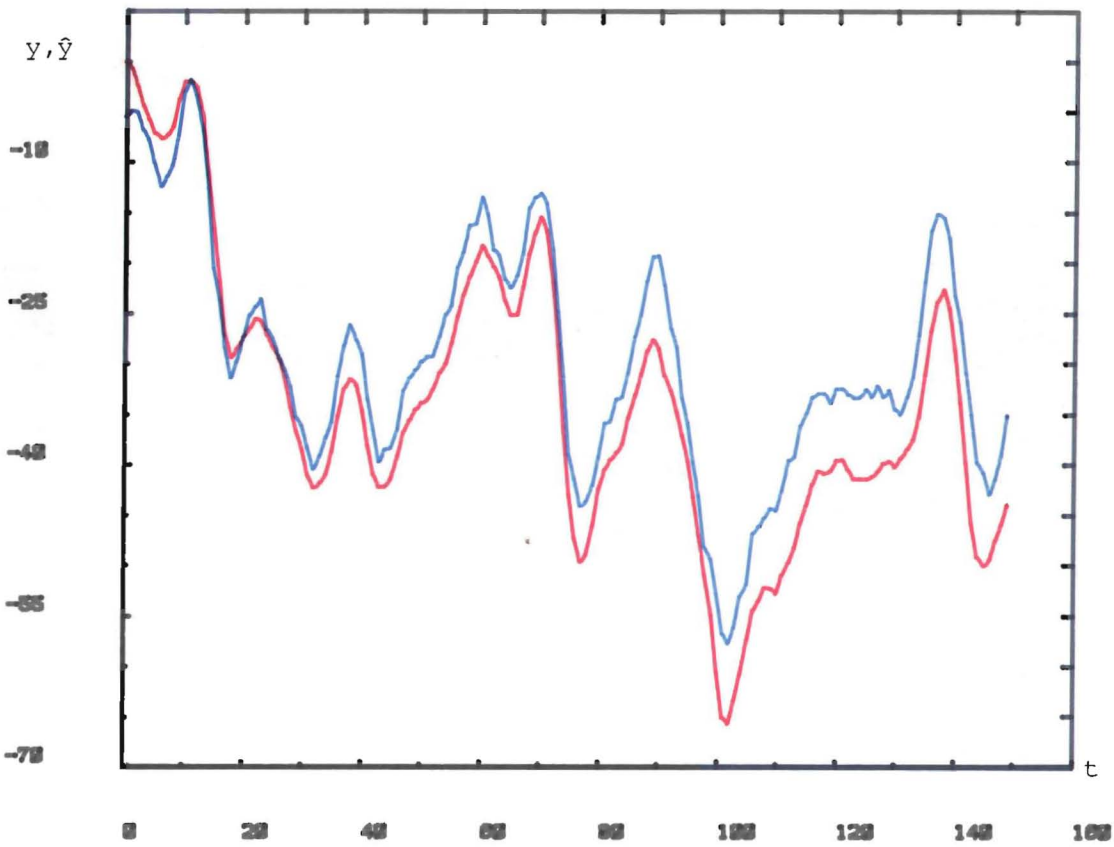


Fig.13. SISO-simulation using HANKEL output.

Comparing the simulation of Fig.13 with the simulation of Fig.11 we see that an important part of the information is lost during the order reduction. This is caused by the model reduction of the Markov string which, of course, does not give the best possible state space realisation in output error criterion.
The program HANKEL also reconstructs the estimated Markov-sequence for a second order realisation. The reconstructed Markov-sequence is stored too.

The results of HANKEL are used as initial estimations for the programs LS_SSM and DIRECTO . The state-space realisation produced by HANKEL is transformed in a pseudo-canonical observability form by PCAN and used for LS_SSM. A sequence of minimal polynomial coefficients is extracted from the reconstructed Markov-parameters and entered in DIRECTO.
The prgram LS_SSM determines a state space model using equation error, output error or innovation error criterions. In this case we need an output-error model because we want to design a controller based on the simulation model. The state space model obtained by LS_SSM is given in Appendix C-1.
The program DIRECTO determines a set of minimal polynomial coefficients and the start sequence of the Markov parameters. The parameters are estimated using an output error criterion. The obtained MPSSM-model is given in Appendix C-2.

Both programs should give better simulation results than the simulation based on the output of HANKEL because HANKEL estimates a model using the Hankel criterion which is not the same as the output error criterion. Evaluating the simulations with an output error criterion of course the results of DIRECTO and LS_SSM should be better. Moreover DIRECTO should be superiour to LS_SSM because the modelset of LS_SSM is a part of the modelset of DIRECTO (which is not restricted by given structure indices).
The simulaton results of LS_SSM and DIRECTO are shown in Fig.14. The red lines indicate the measured signals, the blue lines are the simulations. We immediately conclude that the both simualtions perform better than the simulations using HANKEL-output. To compare the results of LS_SSM and DIRECTO we need to have a closer look at the absolute errors between the simulations and the measured signals:

LS_SSM:	DIRECTO:
V_0 = 0.2830673E+01	V_0 = 0.2685179E+01

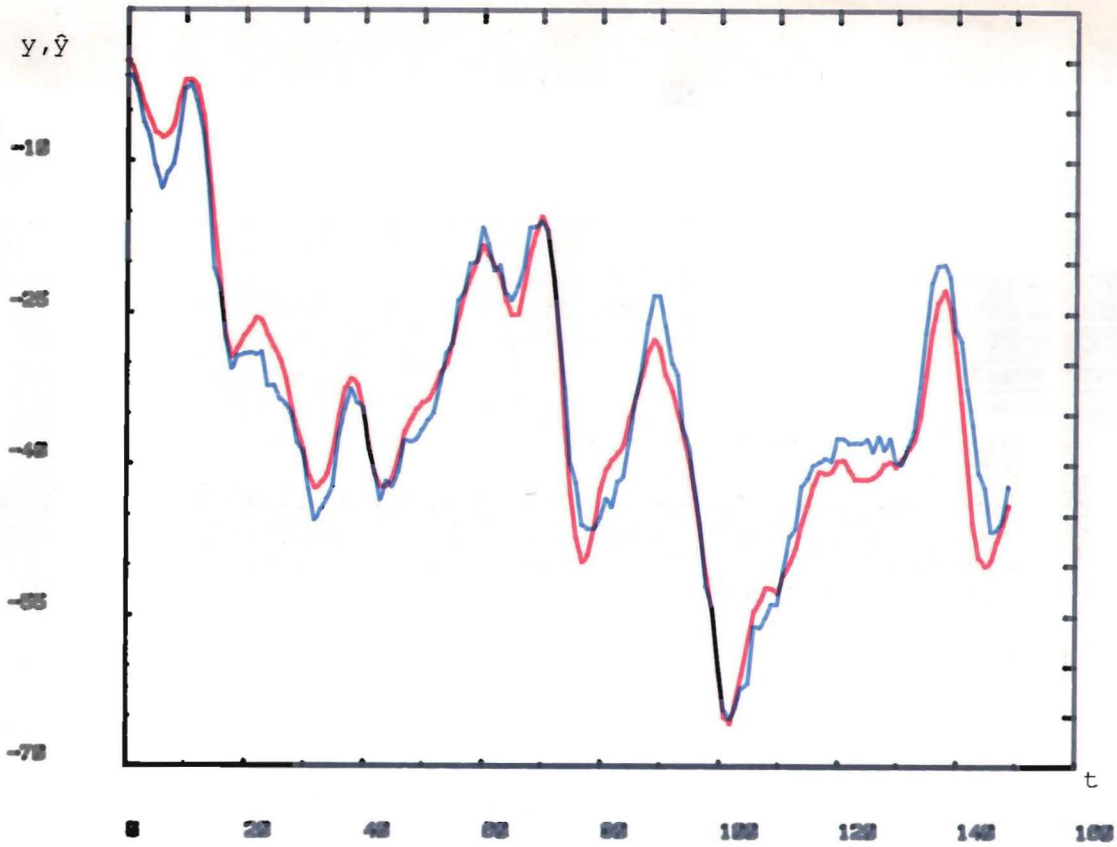


Fig.14a. SISO-simulation using LS_SSM output.

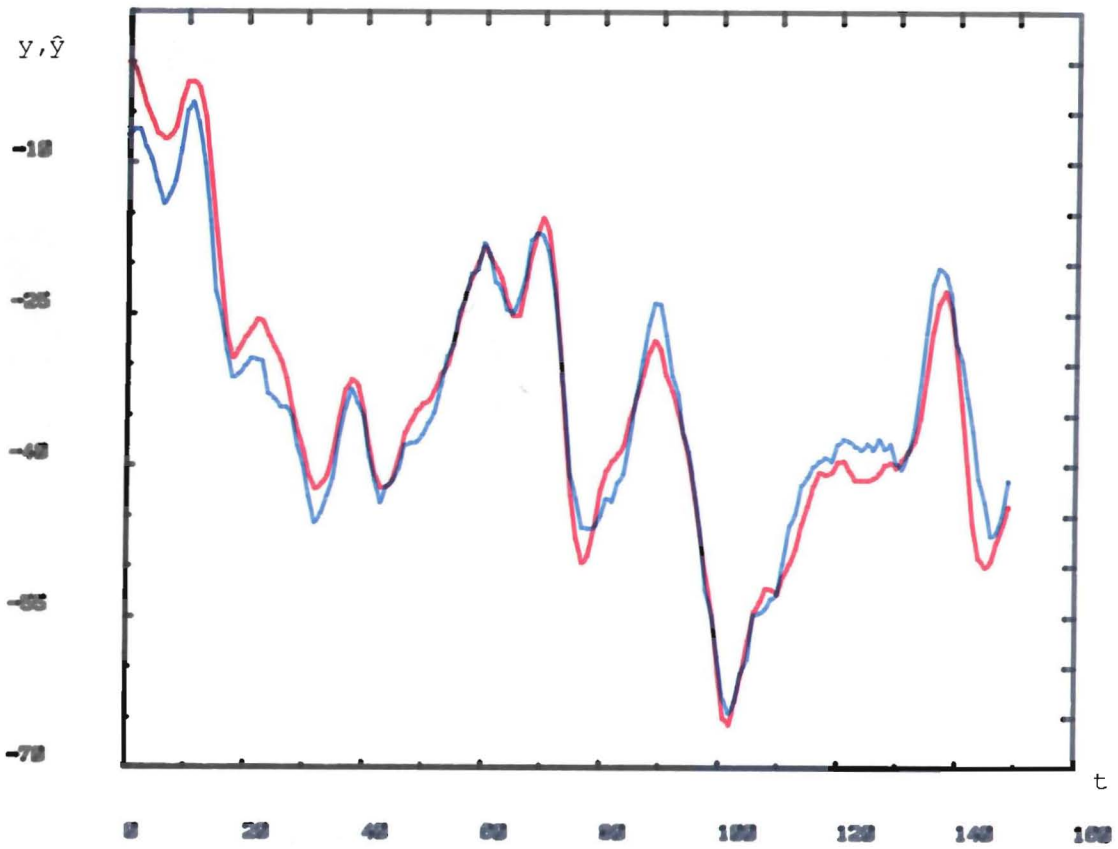


Fig.14b. SISO-simulation using DIRECTO output.

The output errors given above are calculated using the samples 15 up to 150. The first 15 samples are used by DIRECTO to determine the initial state. This initial state is not a part of the DIRECTO-output so it can not be used for simulation. For this reason the first part of the DIRECTO-simulation is clearly worse than the first part of the LS_SSM simulation. After this first part however the DIRECTO-simulation performs better as was expected.
To obtain proper results with DIRECTO an explicit offset-estimator was implemented. Prior to the implementation of the offset-estimation the errors in DIRECTO-simulations were larger than the errors in LS_SSM simulations.
The comparatively large errors between measured outputs and simulations are caused by the bad measurement conditions during the SISO experiment. These conditions will be improved before starting the SIMO and MIMO experiments.

3.4 The SIMO-experiment

Next we want to extend the identification for the single-input-multi-output (SIMO) case. In this experiment the platform is really floating in the water. Only one servomotor was excited and all three heights were measured. As mentioned the height measurements will be troubled by translation and rotation of the platform because sofar it is not feasible to measure absolute heights.
The excitation of the input-signal and the collection of the data were performed by the program MEETSIMO on IBM-computer. The collected data are plotted in Fig.15:

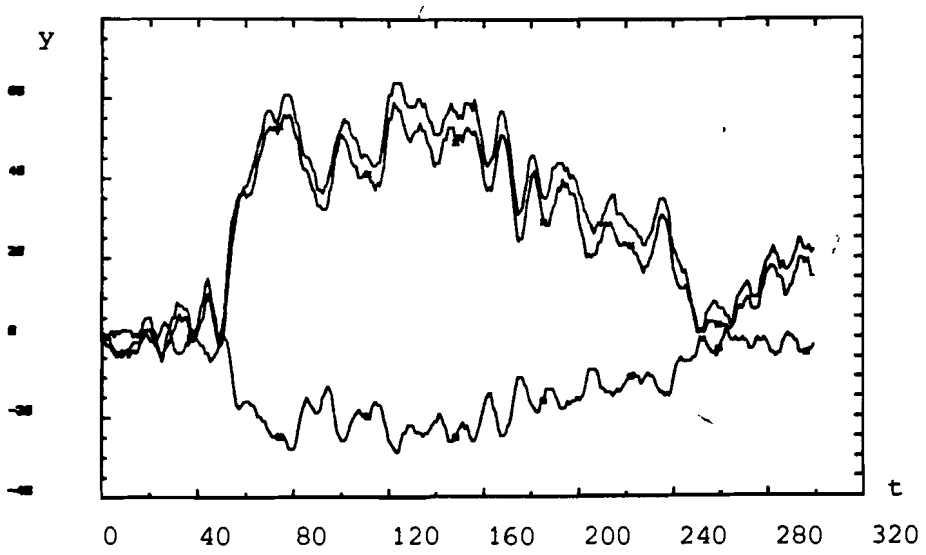


Fig.15. The outputs of the SIMO-experiment.

The behaviour of two outputs is almost similar and the behaviour of the third output is roughly speaking inverse. This can easily be explained when we consider the location of the measurement-points and remember that only one input was used cf. Fig.16.

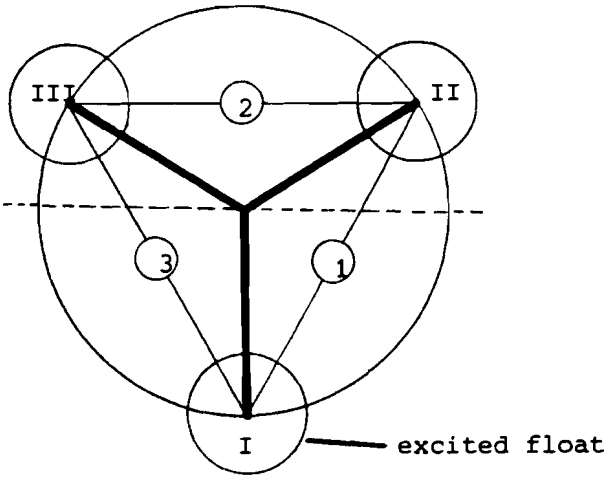


Fig.16. The location of the excitation and measurement points.

When float 3 is excited the platform will turn more or less over the x-axis shown in Fig.16 because of the inertia force. This causes the output-signals 1 and 3 to be almost equal and output 2 almost inverse. The symmetry, however, is not perfect mainly because of the crane on the platform.
Similarly to the SISO-experiment a Markov-sequence is estimated based on the filtered data. Again it was proved by simulation that truncation of the Markov string after 20 samples does not cause any significant loss of information. To determine the order of the system the singular values of the Hankel matrix are examined:

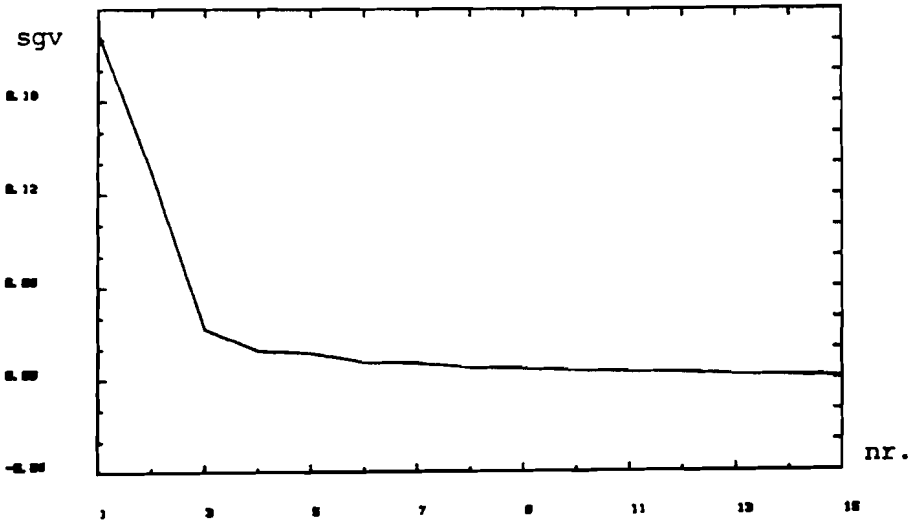


Fig.17. Singular values of the Hankel matrix.

In this case again the most significant edge occurs at the third

singular value (indicating a second order system). This is rather surprising because, when we assume that every float (every SISO-subsystem) contains two states, the order of the SIMO-system should be six unless several states are numerically identical. This situation can occur when several poles of the system are (almost) identical. In paragraph 4.2 we will discuss the pole-locations of the model and consider this phenomena more thoroughly. For the moment we just create a second order realisation as indicated by the singlar values of the Hankel matrix.
Now we want to use the second order realisation as an initial model for identification with LS_SSM and DIRECTO. To obtain an initial model for DIRECTO again the Markov parameters are reconstructed and minimal polynomial coefficients are calculated using DIRSTA. For LS_SSM, however, we need an initial state space model in the observability form. Because it is impossible to create a second order observability realisation of a system with three outputs we use a third order realisation as initial model for LS_SSM.
The model obtained by LS_SSM is given in Appendix C-3.
The model obtained using DIRECTO is given in Appendix C-4.

Both models were used for simulation. The simulated signals and the measured signals are plotted in Fig.18. The output errors are given below:

LS_SSM:	DIRECTO:
V_{01} = 0.2600305E+01	V_{01} = 0.3583565E+01
V_{02} = 0.2705085E+01	V_{02} = 0.2705748E+01
V_{03} = 0.2003381E+01	V_{03} = 0.2161680E+01
V_{0} = 0.7309200E+01	V_{0} = 0.8450993E+01

where V_{0i} is the absolute error in the i-th signal.
It strikes that the output error in the DIRECTO-simulation is larger than the error in the LS_SSM simulation. This is caused by the fact that the order n of the state space realisation is higher than the rank of the minimal polynomial determined by DIRECTO. Because of this we can not compare the results of LS_SSM and DIRECTO unless we reduce the order of the state space realisation obtained by LS_SSM. In general, however, the reduced state space realisation will not be an optimal realisation with respect to an output error criterion. This is an important disadvantage of the estimation of a state space model in the observability form as performed by LS_SSM.
Considering the simulations we see that during the first 200 samples the performance of the simulations is about equally good and the output-error is very small. During the last part of the simulation, however, a drift signal becomes apparent that could not be fitted by either of the identification programs. This drift signal may be caused by translation and/or rotation of the platform disturbing the height-measurements. For this reason we think this model can be improved by implementing better measurement methods.

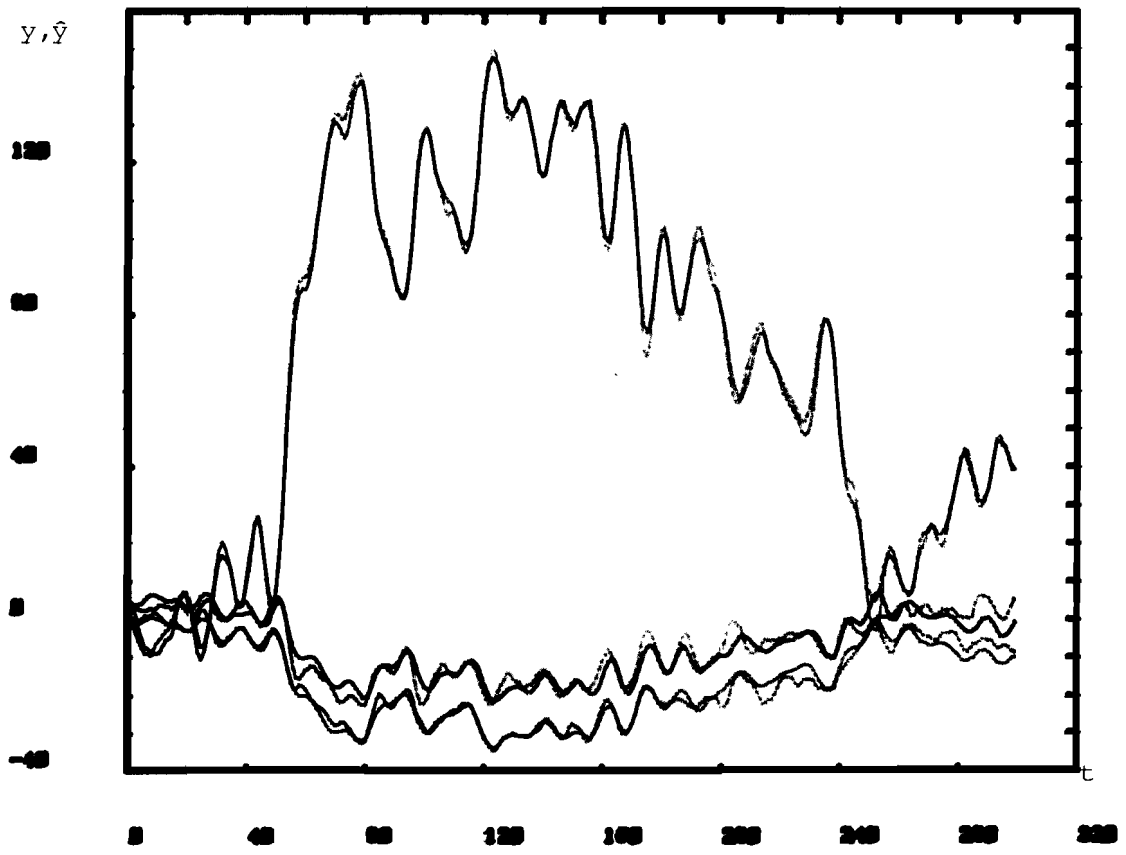


Fig.18a. SIMO-simulation using LS_SSM output.

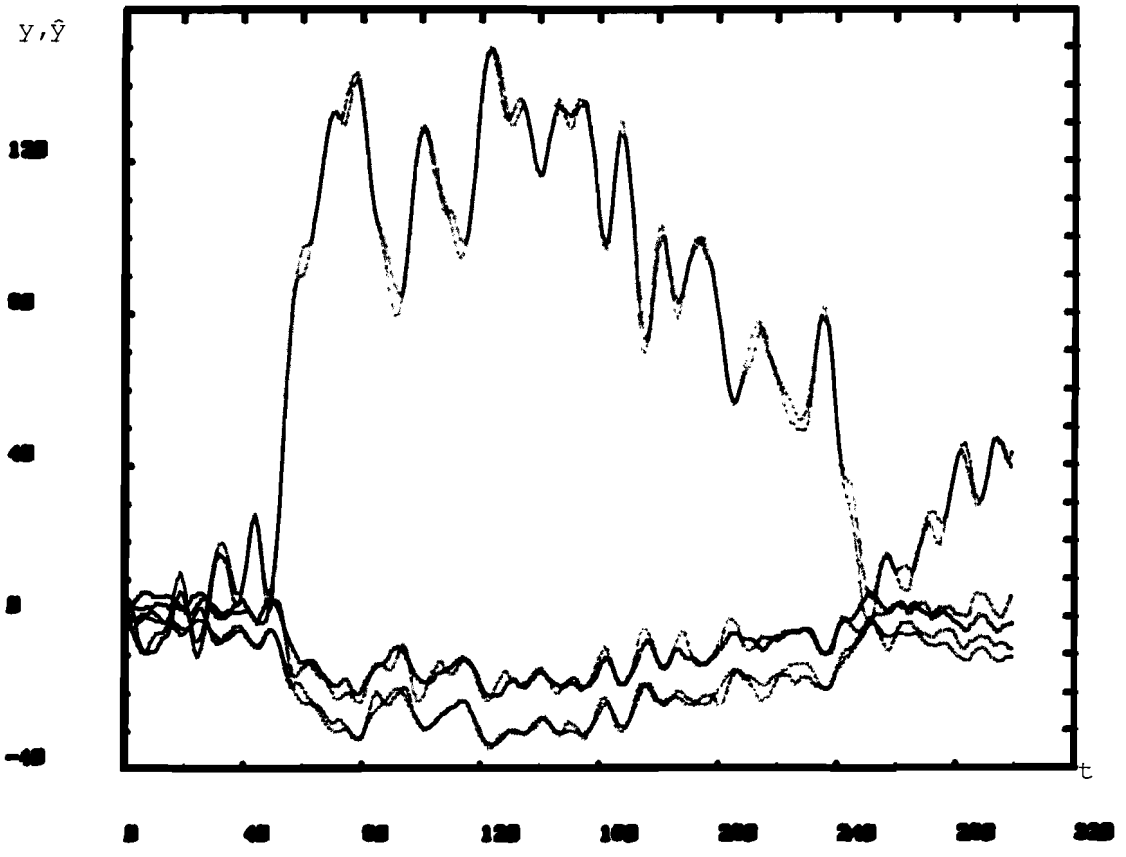


Fig.18b. SIMO-simulation using DIRECTO output.

3.5. The MIMO-experiment

Finally we will consider the complete MIMO-process. In this configuration three independent noise sequences are applied to the servomotors. All three input signals are stored together with the measured output signals. The generation of the inputsignals and the collection of the data are performed by the progran MEETMIMO on IBM computer. The measured outputs are plotted in Fig.19.

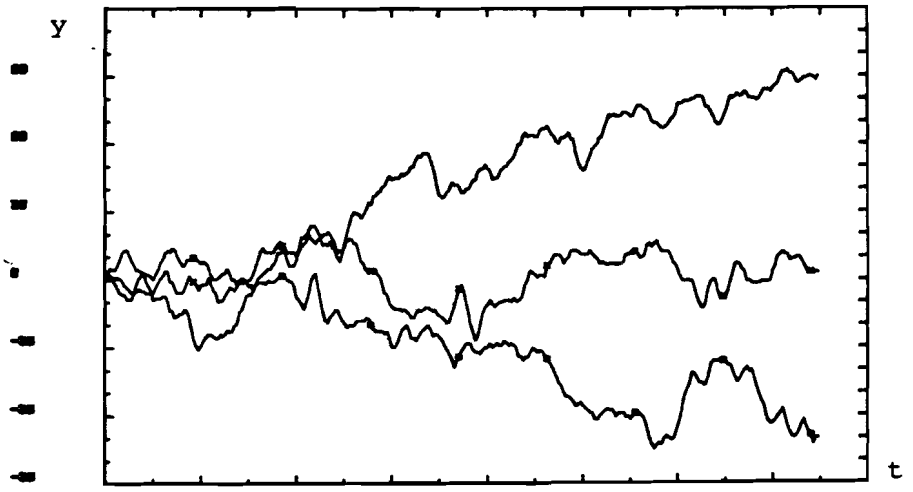


Fig.19. The outputs of the MIMO-experiment.

As before the input signals are filtered and transformed before using them for identification purposes.
First a finite Markov-string is estimated (l=20) using ESMARK. An estimation of the order of the system is obtained considering the singular values of the Hankel matrix produced by HANKEL:

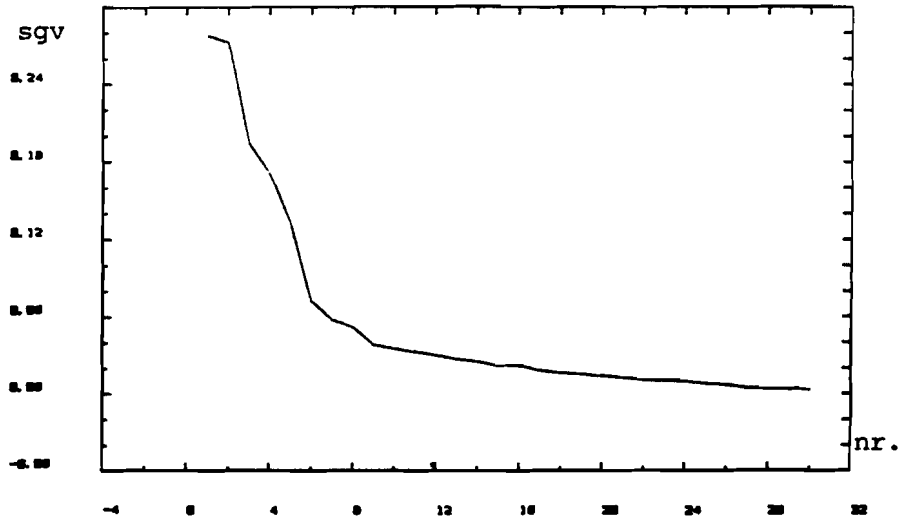


Fig.20. Singular values of the Hankel matrix.

In this case it is impossible to detect a clear edge. Also th calculations of the noise level seem to be not very reliable becaus they would result in an extreme high system order. Using our knowledg of the order of the system as described in par.3.4. and trying t approximate Fig.20 with two straight lines we decide to try a sixt order realisation.
Using HANKEL a sixth order Markov string and state space realisatio are constructed which are used to calculate initial models for LS_SS and DIRECTO.
The final results of DIRECTO and LS_SSM are given in Appendices C-5 an C-6

These results are used for simulation. The simulation results ar plotted in Fig.21a/b. The output errors are given below:

LS_SSM:	DIRECTO:
V_{01} = 0.1905310E+01	V_{01} = 0.1665260E+01
V_{02} = 0.2289953E+01	V_{02} = 0.2047058E+01
V_{03} = 0.3245606E+01	V_{03} = 0.2993687E+01
V_{0} = 0.7440869E+01	V_{0} = 0.6706005E+01

Both simulations perform very well. Clearly a sixth order model i sufficient to contain most of the information. As expected the outpu error in the DIRECTO-simulation is smaller than the output error i the LS_SSM simualtion.
At this point it may be useful to mention that all results presented i this report using DIRECTO were obtained after implementing a explicit offset estimator. Earlier results using DIRECTO output wer clearly worse than the results using LS_SSM. After the implementatio of the offset estimator the simulation results using DIRECTO outpu were at least as good as the simulations using LS_SSM output
In Fig.22 we show the pole-zero plots of the transfermatrix usin LS_SSM output. Considering the pole-zero cancellations we conclud that the order of the diagonal elements is effectively two and th order of the off-diagonal elements four. This can be explaine considering the process again. The diagonal elements represent th influence of an input on the nearest float. In this case only one floa and thus only two states are invloved. The off-diagonal element represent the cross-influences between two floats. Two floats ar invloved so the effective order will be four.
In the next chapter we will look more closely to the structure of th model. We will try to show that the expected structure of the process based on physical modelling) is reflected in the structure of th estimated model which would of course strengthen our confidence in th model.

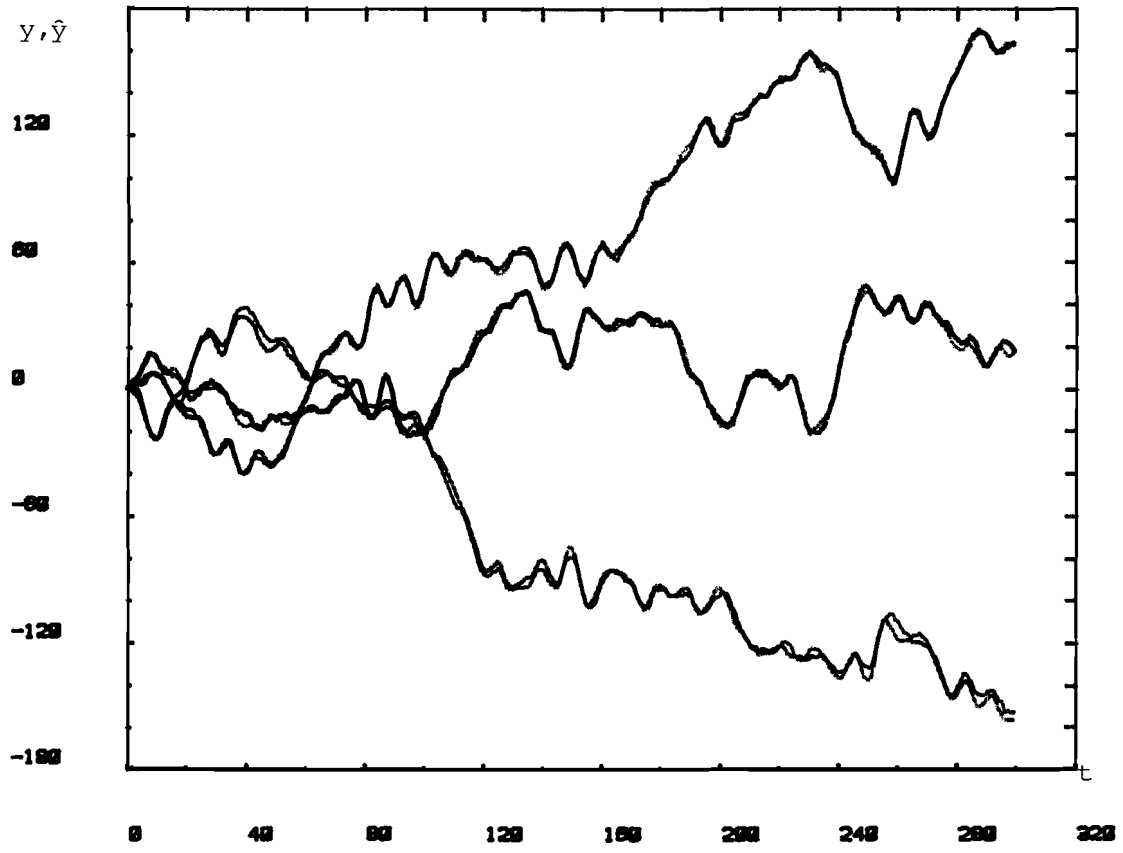


Fig.21a. MIMO-simulation using LS_SSM output.

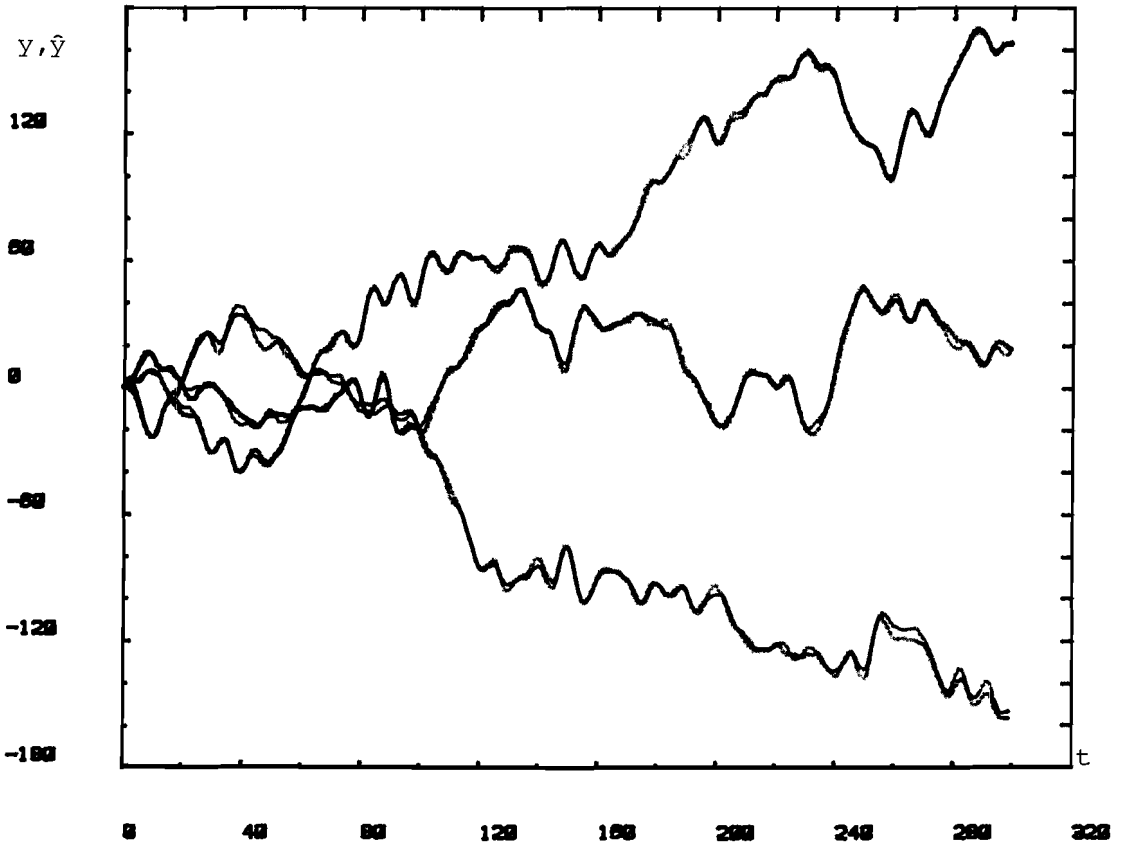


Fig.21b. MIMO-simulation using DIRECTO output.

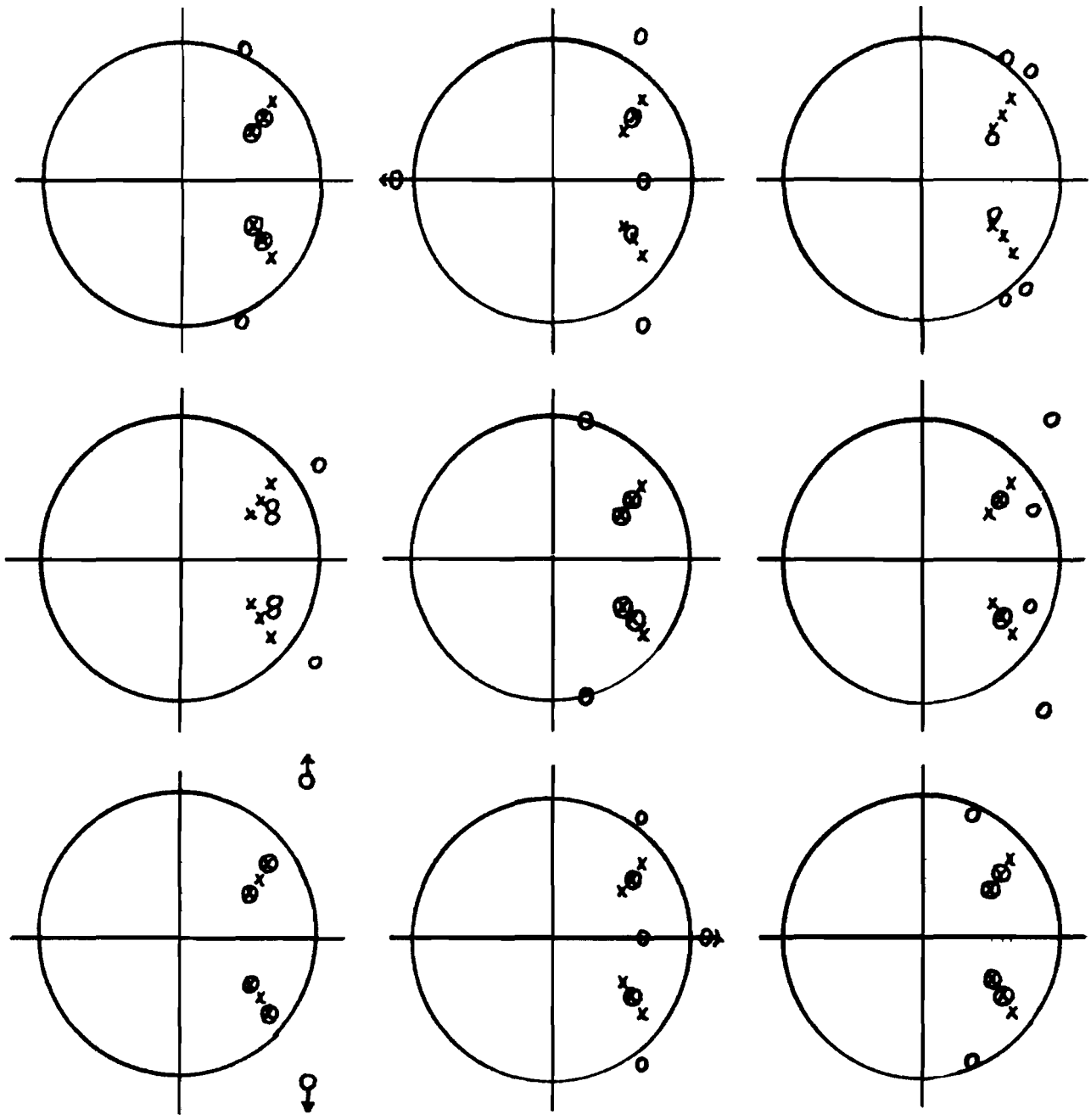


Fig.22 Pole-zero plots of the MIMO-transfermatrix.

4. MIMO model validation

4.1 Introduction

As mentioned before we will now consider more closely the estimated MIMO-model. We will try to assign a physical meaning to the estimated parameters. Especially we want to investigate whether it is possible to distinguish the influence of the three different subsystems, i.e. the three floats, in the model structure.
To investigate the structure we use two different special state space realisations : the Jordan canonical form and the observability canonical form with which we deal in paragraph 4.3 and 4.4 respectively. First of all however we examine the pole locations of the transfermatrix. In this paragraph we will try to explain the fact that when only one input is used (the SIMO situation) the order of the system decreases until about two, although theoretically the order of the system (supposing all states are coupled) should be the same as the order of the MIMO system.
After having considered the structure of the estimated model we will look more closely at the integrational part in the outputs. It has been noted that an important part of the total transferfunction between inputs and outputs is caused by the integrator. This dominancy may cause a worse estimation of the other states. To check this we will separate the integrational part and compare the rest of the simulation with the measured signals.

4.2 Pole locations

In this paragraph we have a closer look at the poles of the transfer matrix. To construct the transfer matrix we use for example the relation between the state space model and the Markov parameters as described in chapter 2. From this relation it is easy to understand the relation between the state space model and the transfer matrix model:

$$H(z) = C.(zI-A)^{-1}.B \qquad (14)$$

The elements of H(z) are all ratios of polynomials and can be regarded as the transfer functions between input i and output j. The poles of the transfer function are the zeros of the characteristic equation:
$\det(zI - A) = 0$
Considering the structure of the process which can be separated in three almost identical subsystems we expect three almost identical pole-pairs. This assumption is supported by the fact that the estimated order of the SIMO-process is only two (cf. par. 3.4). This phenomena can be explained by the following simplified example:

Consider the blockdiagram given in Fig.23 :

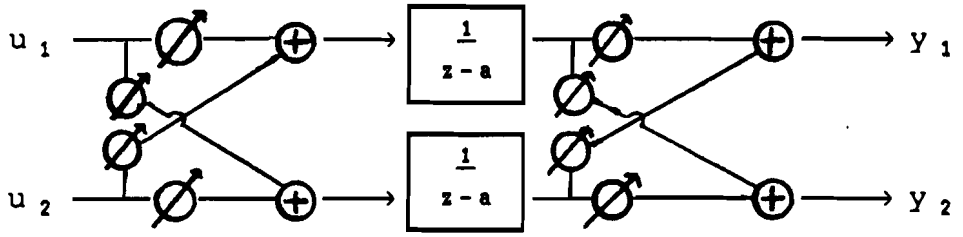


Fig.23 Example system with two identical poles.

In the example system shown in Fig.23 there exist always two independent observable states although the poles of the system are identical (assuming B non-singular and u_1, u_2 independent). However when only one input is used the system can be reduced to the following scheme:

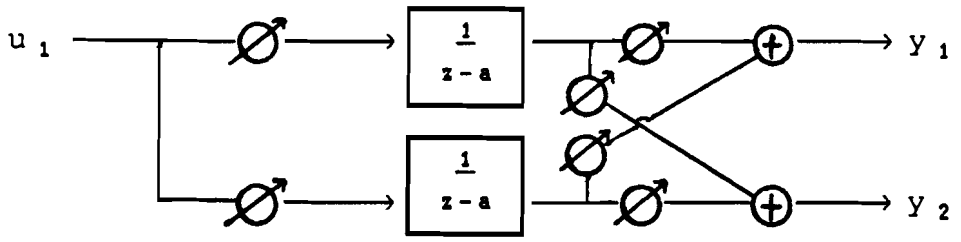


Fig.24 Example system using only one input.

When the initial states are neglected we can reduce the system of Fig.24 to the following scheme:

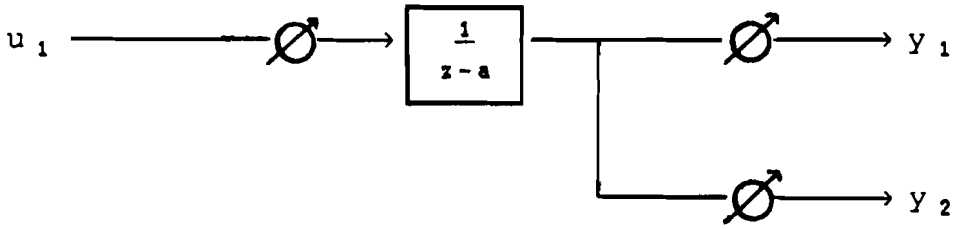


Fig.25 Reduced example system.

In this case no longer two independent observable states exist because we use only one input. In the same way the order of a system with three identical pole-pairs effectively reduces to two when only one input is used.

Returning to the platform we consider the poles of the initial model obtained by HANKEL in the MIMO-experiment:

$$\lambda_{1,2} = 0.7438 \pm j.\ 0.4887$$
$$\lambda_{3,4} = 0.6992 \pm j.\ 0.5019$$
$$\lambda_{5,6} = 0.6737 \pm j.\ 0.4129$$

Indeed we have three almost identical polepairs. However, as shown in the example, when three proper input signals are used (such as independent noise sequences) it is possible to recognize all six separate (!) states.
Using only one input however (filling the B-matrix with zeros for the other inputs) the number of independent states (= the number of independent Markov parameters) effectively reduces to two. We conclude that similar as in the example process the estimated low order of the SIMO-process is caused by the existence of multiple poles

and only one input.

Considering the final estimation results again the structure of the process is clearly reflected in the estimated model. As expected after physically modelling each float introduces a single polepair. The three polepairs are very near to each other because the three subsystems (i.e. the floats) are almost identical.
The poles of the state space realisation estimated with LS_SSM are the eigenvalues of the system-matrix:

$$\lambda_{1,2} = 0.7012 \pm j.\ 0.5615$$
$$\lambda_{3,4} = 0.6343 \pm j.\ 0.4510$$
$$\lambda_{5,6} = 0.5734 \pm j.\ 0.3347$$

The poles of the MPSSM model determined by DIRECTO are the roots of the minimal polynomial:

$$\lambda_{1,2} = 0.6581 \pm j.\ 0.6072$$
$$\lambda_{3,4} = 0.7574 \pm j.\ 0.3509$$
$$\lambda_{5,6} = 0.5948 \pm j.\ 0.4023$$

To find the symmetry as indicated by the poles it proved to be very important to use proper preprocessing (cf. par.3.2.). When this is left out the estimation routines will find other poles to cope with e.g. quantisation noise, delay or offset. Especially when the real system poles are very near to each other the estimators tend to use only one or two polepairs to model the dynamics of the system.

4.3. The Jordan form

A Jordan state space realisation is often used to determine the relations between states and inputs and outputs. In a pure Jordan form the A-matrix is diagonal matrix and the diagonal elements are the eigenvalues of the matrix (the poles of the system). In this case all states are decoupled.
This realisation however can only be found in case of cyclic systems having distinct eigenvalues. Cyclic systems are systems whose poles have all geo,etric multiplicity one. The geometric multiplicity of an eigenvalue is the dimension of the eigenspace of that eigenvalue. In case of non-cyclic systems and/or multiple eigenvalues some pseudo-Jordan realisations are known in which a maximum of states is decoupled.
In general the matrices of a Jordan form are complex because in general the eigenvalues of a process are complex. Calculating with complex matrices introduces a lot of numerical inaccuracy which can be avoided using pseudo-Jordan realisations containing only real matrices. In these realisation every complex polepair is combined in a so called Jordan block which is real.
The several pseudo-Jordan realisations are dealt with in Appendix B. At this moment it is not yet possible to gain more insight in the system using Jordan canonical realisations because of the lack of proper software. In future it might be useful to examine some pseudo-Jordan realisations of the model.

4.4 The observability pseudo canonical form.

The observability canonical form of a state space model is chracterized by the fact that the outputs are (some of) the states. The A-matrix has the following structure [Veltmeyer,1985]:

$$
A = \left[\begin{array}{c|c|c}
\begin{matrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_{1n} & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} \\
\hline
\begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_{2n} & \cdot & \cdot & \cdot \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} \\
\hline
\begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \end{matrix} & \begin{matrix} 0 & 1 & 0 \\ & & \ddots \\ & & 1 \\ a_{q1} & \cdot & \cdot \end{matrix}
\end{array}\right]
\begin{matrix} \\ \uparrow \\ \mu_1 \\ \downarrow \\ \\ \uparrow \\ \mu_2 \\ \downarrow \\ \\ \uparrow \\ \mu_q \\ \downarrow \end{matrix}
$$

$$
\begin{matrix} | & \leftarrow \mu_1 \rightarrow & | & \leftarrow \mu_2 \rightarrow & | & \leftarrow \mu_q \rightarrow & | \end{matrix}
$$

The structure indices μ_i are chosen based on physical knowledge and reamin fixed during estimation. Only in very rare cases the system will not be in the modelset due to an improper choice of the structure indices. This is caused by the overlapping properties of the observability form. The structure indices determine more or less the relative importance of each output. In the case of the floating platform we choose $\mu_1 = \mu_2 = \mu_3 = 2$ because we expect that each float has equal relevance. We expect a sixth order system so $x_1 = y_1$; $x_3 = y_2$; $x_5 = y_3$. We now can draw the following block-diagram:

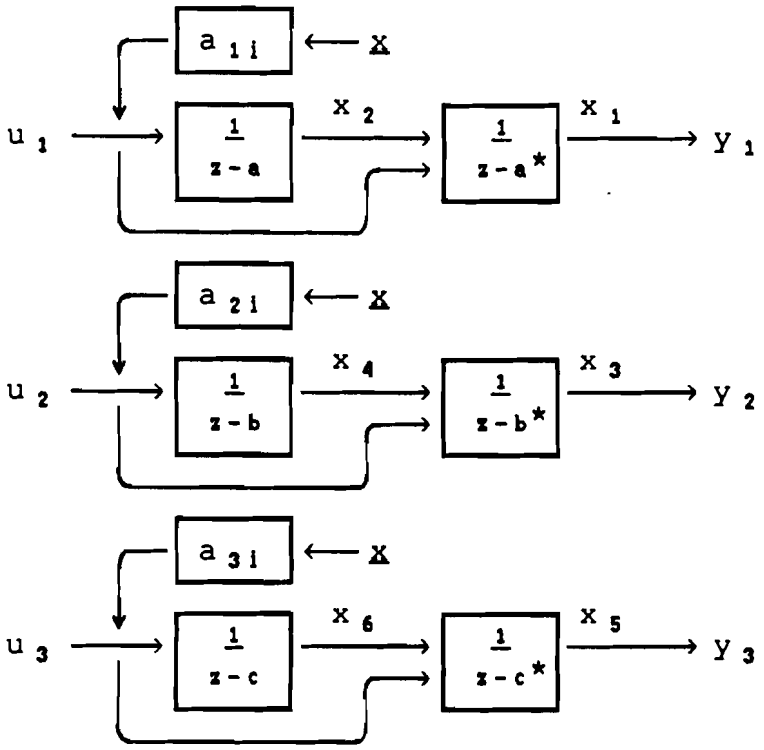


Fig.26. Block diagram of the observability pseudo canonical form if B is " block diagonal dominant".

In the figure above we supposed that the direct influence of an input at the states of another float could be neglected. This must become clear in the structure of the B-matrix in which the corresponding elements indicated in Fig.26 should be dominant. We will call such type of B-matrix "block diagonal dominant".
Moreover the block diagonal elements can be interpreted as the effects of the inputs on the height and the velocity of the nearest float. Because of the symmetry of the system we expect the blocks to be almost identical.
The program LS_SSM gives a observability canonical form on output. The output of DIRECTO is converted into a Markov string using DIRSIM. Using the Ho-Kalman algorithm implemented in the program HANKEL this string is reduced to a state space realisation of order n which is transformed into an observability form using PCAN. The state space realisation obtained using HANKEL is not a very good realisation with respect to an output error criterion. Better order reduction methods are described in par.2.5.
The results of LS_SSM are given in Appendix C-5 and the observability canonical realisation using DIRECTO-output is in Appendix C-7.
Because the state space realisation obtained using DIRECTO is not very

reliable we will only consider the LS_SSM output. For convenience the B-matrix is given below:

$$B = \begin{bmatrix} 0.1975442E-01 & -0.7823338E-02 & 0.5549456E-02 \\ 0.6562917E-01 & -0.7921277E-02 & -0.1080546E-01 \\ 0.3220015E-01 & 0.3733199E-01 & 0.1759723E-01 \\ 0.1302678E-02 & 0.7023374E-01 & -0.1391743E-01 \\ 0.2927007E-02 & -0.2044060E-01 & 0.3819956E-01 \\ -0.1695212E-01 & -0.3332312E-02 & 0.7593216E-01 \end{bmatrix}$$

This B-matrix is clearly block diagonal dominant. This means that the effect of the first input on the states of the first float are dominant compared to the effects of the other inputs.
Moreover the diagonal blocks are almost identical as expected from the symmetry of the system. The structure of the system as found from physical modelling and knowledge of the process is clearly reflected in the model:

- We have three almost identical subsystems.

- Each subsystem can be described by a second order model.

- Each input has a dominant influence on the nearest float.

As mentioned in par.3.2. the influence of a pure integrator is a very important part of the output signal. In the next paragraph we will consider dominance of the integrational term in the outputs more closely.

4.5 The integrational part of the outputs

In this paragraph we will deal with the dominance of the part in the output that is caused by the integrator. Because of the dominance of this part one could fear that the other system dynamics are estimated less accurately. This assumption is contradicted by the results discussed in earlier paragraphs of this chapter. Especially the location of the poles of the estimated model give good faith in the accuracy of the model.
The total transferfunction in the z-domain is given by:

$$H_{tot}(z) = \frac{z}{z-1} \cdot H(z)$$

where H(z) is the estimated transferfunction.
The residue of the transferfunction in z=1 is determined by substituting z=1 in $(z-1) * H_{tot}(z)$. Doing this for every element of the transfermatrix we obtain the following gain matrix K:

$$K = \begin{bmatrix} 0.2618915E+00 & -0.3871934E-01 & -0.5696259E-01 \\ -0.3510186E-01 & 0.2671739E+00 & -0.4850501E-01 \\ -0.4971524E-01 & -0.1751516E-01 & 0.2852709E+00 \end{bmatrix}$$

Denoting the integrational part in the outputs by $\underline{y}_{int}$ it holds that:

$$\underline{y}_{int} = K \cdot \underline{u}$$

(remember $\underline{u}$ is the integrated input vector)

Again we point out the symmetry in the K-matrix. The diagonal elements are clearly dominant which means that changing the distance between the platform and a float has a dominant influence on the height of the platform near that float. The signs of the elements of K can be explained considering Fig.16 again. Increasing the distance between the platform and the float will increase the height of the platform near that float which results in a positive gain on the diagonal. Because of the turning axis as indicated in Fig.16 however an increase in distance between one float and the platform will temporarily decrease the height of the platform near the other floats which causes negative off-diagonal gain factors. Because of the symmetry of the system all the off-diagonal elements and all the diagonal elements should be almost the same.

The signal $\underline{y}_{int}$ is now constructed and subtracted from the measure signals and the simulated signals. The non-integrational parts of th measured signals and the simulations are plotted in Fig.27. Again th red lines indicate the measured signals and the blue lines th simulations. Indeed the greatest part of the signal power is include in the integrational part!
The absolute and relative errors (= the absolute error devided by th output signal power) between the non-integrational part of th simulations and the non-integrational part of the measured signal are given below:

V_{o1} = 0.1991722E+01 V_{rel1} = 0.3146416E+00
V_{o2} = 0.2268321E+01 V_{rel2} = 0.3740882E+00
V_{o3} = 0.3149016E+01 V_{rel3} = 0.2548395E+00

Although the relative errors become rather large we conclude frc Fig.27 that the maximum error between measured output and simulatic is never larger than one or two bits. Because of this we decide tha although the integrational part is dominant the non-integrationa part is also well modelled.

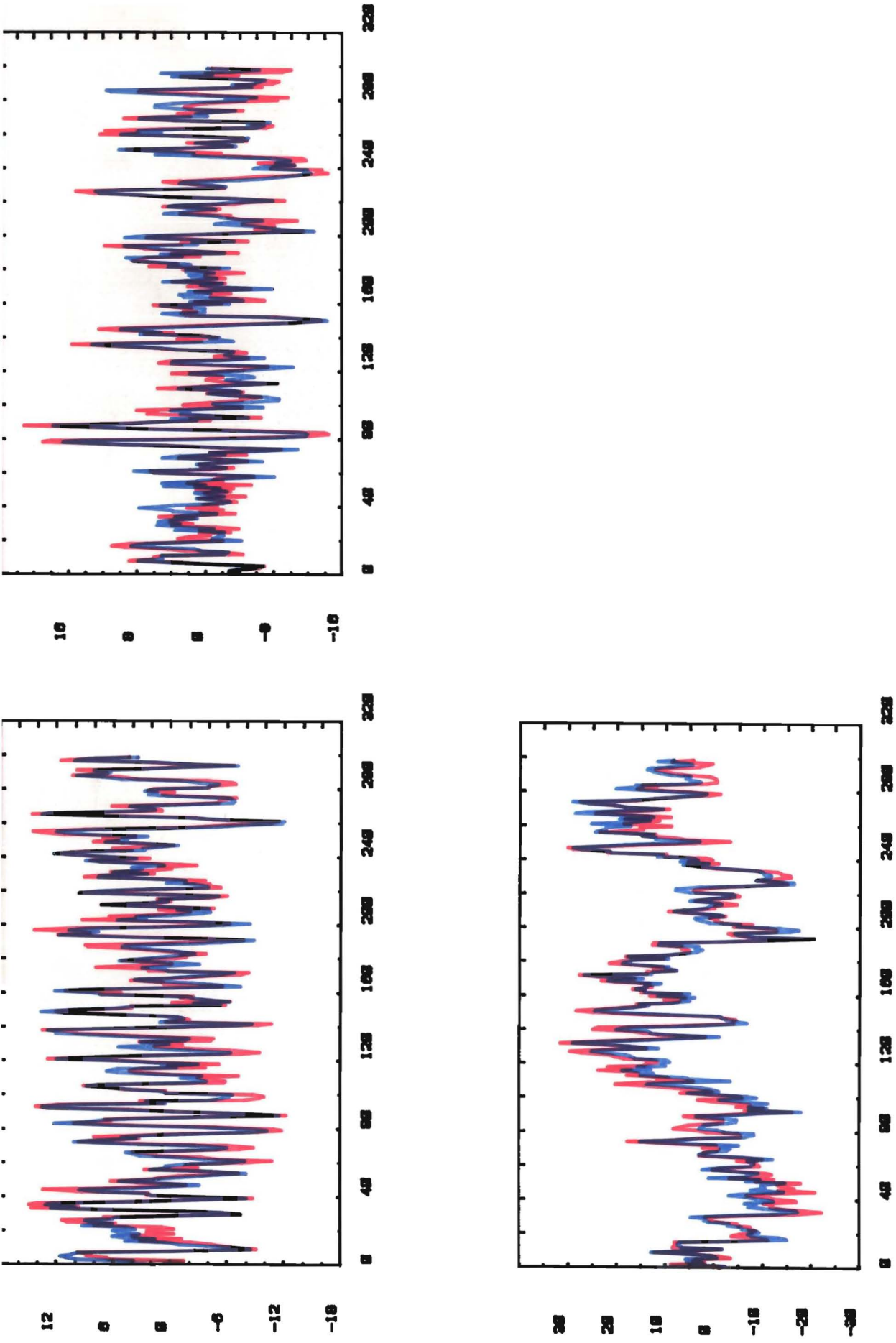


Fig.27. Non-integral parts of the respective outputs and simulations.

5. Conclusions

- Good simulation results were obtained with both DIRECTO and LS_SSM. After implementing an explicit offset-estimator DIRECTO simulation results were at least as good as LS_SSM results. DIRECTO however requires a very good initial model.

- The structure and the symmetry of the floating platform is clearly reflected in the model. We can distinguish three almost identical subsystems each containing two states.

- Proper preprocessing of the signals is essential to estimate the actual system dynamics. Before preprocessing already fairly good simulation results were obtained but the structure in the model is completely lost. Both general types of preprocessing (low-pass filtering) and more specific types of preprocessing only suited for this system were used.

- The identical subsystems cause multiple poles. Because of the existence of multiple poles it is impossible to recognize all separate states when not all inputs are used. This causes a lowly estimated model-order in the SIMO-experiment.

- A considerable part of the output is caused by the integrator. When the integrational part is subtracted from the signals only a low-power signal is left. Although this integrational part must play a dominant role during identification the non-integrational part is also estimated well.

- LS_SSM estimates a state space model in pseudo canonical observability form. Because of this it is impossible to estimate a model with a lower model-order than the number of outputs. In this case there exist dependencies in the outputs. To be able to estimate a model in this case too we first have to determine the dependencies in the outputs. Next we estimate a model using only the independent outputs. Finally we construct the output matrix with help of the determined dependencies.

- DIRECTO can not estimate systems with poles with a negative real part (in the z-domain). Normally this can be avoided using high sample-rates. When quantisation noise is present or when it is impossible to use high sample rates this may become a disadvantage.

Appendix A

```
C*************************************************************************
C
C                          PROGRAM ADSM01
C                                                        VERSION : 1.0
C*************************************************************************
C
C       VERSION          : 1.0
C       DATE             : 15-12-1987
C       AUTHOR           : KARSTEN PRONK
C       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                          FACULTY OF ELECTRICAL ENGINEERING
C       GROUP            : CONTROL AND MEASUREMENT
C       ADDRESS          : P.O. BOX 513
C                          5600 MB  EINDHOVEN
C       COUNTRY          : THE NETHERLANDS
C
C
C       PROGRAM DESCRIPTION:
C
C       THIS PROGRAM ADDS SEVERAL VARIABLES (ON REQUEST MULTIPLIED WITH
C       A CONSTANT SCALE-FACTOR) WHICH ARE STORED IN ER-STRUCTURED
C       SAMPLE FILES. THE RESULTS ARE STORED IN AN ER-STRUCTURED SAMPLE
C       FILE.
C
C*************************************************************************
C
C       Parameters:
C       MPSAMP=3000      Max. number of samples in a sample file;
C       MPSIGN=30        Max. number of signals in a sample file;
C
C*************************************************************************
C
C       Variables entered by user:
C
C       FILN1      [CHAR*40] : NAME OF THE ER-SAMPLE FILE WITH DATA
C       FILN2      [CHAR*40] : NAME OF THE OUTPUT ER-SAMPLE FILE
C
C*************************************************************************
```

```
C*****************************************************************************
C
                                PROGRAM ADSM02
C                                                                VERSION : 1.0
C*****************************************************************************
C
C       VERSION          : 1.0
C       DATE             : 10-2-1988
C       AUTHOR           : KARSTEN PRONK
C       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                          FACULTY OF ELECTRICAL ENGINEERING
C       GROUP            : CONTROL AND MEASUREMENT
C       ADDRESS          : P.O. BOX 513
C                          5600 MB  EINDHOVEN
C       COUNTRY          : THE NETHERLANDS
C
C
C       PROGRAM DESCRIPTION:
C
C       THIS PROGRAM READS A PICOS STRUCTURED DATA FILE AND MULTIPLIES
C       THE DATA WITH A MATRIX T SO THAT Y'=T*Y.
C       THE TRANSFORMED DATA ARE STORED IN A PICOS STRUCTURED FILE.
C
C*****************************************************************************
C
C       PARAMETERS:
C
C       MSIG   =  15   :  MAXIMUM NUMBER OF SIGNALS
C       MSAMP  = 500   :  MAXIMUM NUMBER OF SAMPLES
C
C*****************************************************************************
C
C       VARIABLES ENTERED BY USER:
C
C       FILN          [CHAR*40] : NAME OF THE PICOS-STRICTURED IMPUT FILE
C       FILOUT        [CHAR*40] : NAME OF THE PICOS-STRUCTURED OUTPUT FILE
C       NSIG          [INT]     : NUMBER OF SIGNALS TO BE READ
C       NSAMP         [INT]     : NUMBER OF SAMPLES
C       NRSIG         [INT]     : NUMBER OF SIGNALS TO BE CALCULATED
C       T(MSIG,MSIG)  [DBLE]    : TRANSFORMATION MATRIX
C
C*****************************************************************************
```

```
*****************************************************************************
*
                              PROGRAM DIRSTA
*                                                                  Version 1.0
*****************************************************************************

      VERSION          : 1.0
      DATE             : 28-10-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      PROGRAM DESCRIPTION:

      THIS PROGRAM IS USED FOR DETERMINING INITIAL VALUES FOR THE
      A-PARAMETERS USED IN DIRECTO (OUDBIER). THE MARKOV-PARAMETERS
      ARE READ FROM FILN AND THE A-PARAMETERS ARE STORED IN FILOUT.

         M(n+1) = a1*M(n) + a2*M(n-1) + ... + an*M(1)

      THE A-PARAMETERS ARE SOLVED USING:
      A(n x 1) = MI(n x n) * MV(n x 1)

*****************************************************************************

     PARAMETERS:

     MSIG   =    15 : MAXIMUM NUMBER OF SIGNALS TO BE READ
     MSAM   =   100 : MAXIMUM NUMBER OF SAMPLES TO BE READ
     MN     =    50 : MAXIMUM ORDER OF THE MIN. POLYNOMIAL

*****************************************************************************

     VARIABLES ENTERED BY THE USER:

     FILN      [CHAR*40]  : NAME OF THE ER-STRUCTURED INPUT SAMPLE FILE
                            CONTAINING THE MARKOV PARAMETERS.
     FILOUT    [CHAR*40]  : NAME OF THE MATLAB-STRUCTURED MATRIX FILE
                            CONTAINING INITIAL ESTIMATES FOR DIRECTO.
     N         [INT]      : NUMBER OF MINIMAL POLYNOMIAL COEFFICIENTS TO
                            BE CALCULATED.

*****************************************************************************
```

```
C*******************************************************************
C*
                          PROGRAM DISI01
C*                                                        VERSION 1.0
C*******************************************************************
C
C       VERSION          : 1.0
C       DATE             : 3-11-1987
C       AUTHOR           : KARSTEN PRONK
C       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                          FACULTY OF ELECTRICAL ENGINEERING
C       GROUP            : CONTROL AND MEASUREMENT
C       ADDRESS          : P.O. BOX 513
C                          5600 MB  EINDHOVEN
C       COUNTRY          : THE NETHERLANDS
C
C
C       PROGRAM DESCRIPTION:
C
C       THIS PROGRAM CAN BE USED TO EVALUATE THE RESULTS OF THE PROGRAM
C       DIRECTO. THE PROGRAM READS THE FIRST MARKOV-PARAMETERS AND THE
C       A-PARAMETERS AND DETERMINES A MARKOV-SEQUENCE OF GIVEN LENGTH
C       WHICH CAN BE USED FOR SIMULATION.
C
C*******************************************************************
C
C       PARAMETERS:
C
C       AMAX   =   6   : MAXIMUM NUMBER OF MIN.POLYNOMIAL COEFFICIENTS TO BE R
C       MMAX   =  35   : MAXIMUM LENGTH OF THE START MARKOV SEQUENCE TO BE REA
C       MARMAX= 100    : MAXIMUM LENGTH OF THE MARKOV SEQUENCE TO BE CALCULATE
C       MIP    =   5   : MAXIMUM NUMBER OF INPUTS
C       MIQ    =  10   : MAXIMUM NUMBER OF OUTPUTS
C       MSIG   =  15   : MAXIMUM NUMBER OF SIGNALS
C
C*******************************************************************
C
C       VARIABLES ENTERED BY THE USER:
C
C       FILN          [CHAR*40] : NAME OF THE MATLAB STRUCTURED INPUT FILE
C                                 CONTAINING DIRECTO OUTPUT.
C       FILOUT        [CHAR*40] : NAME OF THE ER-STRUCTURED SAMPLE OUTPUT FIL
C                                 CONTAINING THE MARKOV PARAMETERS.
C       L             [INT]     : LENGTH OF THE MARKOV SEQUENCE TO BE CALCULA
C       TITLE         [CHAR*8]  : TITLE OF THE CREATED DATASET
C
C*******************************************************************
```

```
*******************************************************************************

                              PROGRAM DISI02
                                                              VERSION 1.0
*******************************************************************************

      VERSION          : 1.0
      DATE             : 3-11-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      PROGRAM DESCRIPTION:

      THIS PROGRAM CAN BE USED TO EVALUATE THE RESULTS OF THE PROGRAM
      DIRECTO. THE PROGRAM READS THE DIRECTO OUTPUT AND CONSTRUCTS A
      ER-STRUCTURED DATAFILE (.MRK) THAT CAN BE USED FOR SIMULATION.

*******************************************************************************

     PARAMETERS:

     AMAX  =   9 : MAXIMUM NUMBER OF MIN.POLYNOMIAL COEFFICIENTS TO BE READ
     MMAX  =  35 : MAXIMUM LENGTH OF THE START MARKOV SEQUENCE TO BE READ
     MSIG  =  15 : MAXIMUM NUMBER OF SIGNALS (MSIG=MIP+MIQ)
     MIP   =   5 : MAXIMUM NUMBER OF INPUTS
     MIQ   =  10 : MAXIMUM NUMBER OF OUTPUTS
     MLOW  =   0 : LOWER DIMENSION OF MARKOV SEQUENCE TO BE CALCULATED
     MUP   = 100 : UPPER DIMENSION OF MARKOV SEQUENCE TO BE CALCULATED

*******************************************************************************

     VARIABLES ENETERED BY THE USER:

     FILN          [CHAR*40] : NAME OF THE MATLAB STRUCTURED INPUT FILE
                               CONTAINING DIRECTO OUTPUT.
     FILOUT        [CHAR*40] : NAME OF THE ER-STRUCTURED SAMPLE OUTPUT FILE
                               CONTAINING THE MARKOV PARAMETERS.
     LEN           [INT]     : LENGTH OF THE MARKOV SEQUENCE TO BE CALCULATED
     TITLE         [CHAR*8]  : TITLE OF THE CREATED DATASET

*******************************************************************************
```

```
C*****************************************************************************
C*
                              PROGRAM ENSF01
C*                                                                 Version 1.0
C*****************************************************************************
c
c  VERSION              : 1.0
C  AUTHOR               : KARSTEN PRONK
C  DATE                 : 2-9-1987
C  INSTITUTE            : EINDHOVEN UNIVERSITY OF TECHNOLOGIES
C                         FACULTY OF ELECTRICAL ENGINEERING
C  GROUP                : MEASUREMENT AND CONTROL
C  ADDRESS              : P.O. BOX 513
C                         5600 MB  EINDHOVEN
C  COUNTRY              : THE NETHERLANDS
C
C
C PROGRAM DESCRIPTION:
C
C This program writes a number of samples of I/O-signals into
c an standard structured signal file. The samples are in double
c precision and are given with use of the terminal.
C
c
c Parameters:
c       MSIG=15          Maximum number of signals stored;
c       MSAM=800         Maximum number of samples;
c
c Variables:
c FILN             (CH*40) :Name of file to be created;
c SYSIO(MSAM,MSIG)  (DBL)  :Two dimensional I/O-matrix;
c N                 (INT)  :Number of samples;
c NSIG              (INT)  :Number of signals;
c TITLE             (CH*8) :Titel of the data set;
c NAME(MSIG)        (CH*8) :Names of the signals;
c SBTYP1            (INT)  :Subtype of matrix;
C OFFSET(MSIG)      (DBL)  :Vector containing the offset-parameters
c SCALE(MSIG)       (DBL)  :Vector containing the scale-parameters
c
c*****************************************************************************
```

```
**********************************************************************
*                                                                    *
                          PROGRAM FILTER
*                                                                    *
**********************************************************************

      VERSION        : 1.0
      DATE           : 24-9-1987
      AUTHOR         : KARSTEN PRONK
      INSTITUTE      : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                       FACULTY OF ELECTRICAL ENGINEERING
      GROUP          : MEASUREMENT AND CONTROL
      ADDRESS        : P.O. BOX 513
                       5600 MB  EINDHOVEN
      COUNTRY        : THE NETHERLANDS

      PROGRAM DESCRIPTION:

      THIS PROGRAM READS A NONFORMATTED PICOS-STRUCTURED DATAFILE
      AND WRITES THE CONTENTS OF THIS FILE IN A STANDARD FORMATTED
      PICOS-STRUCTURED DATAFILE.
      IF NECESSARY THE DATA CAN BE SCALED AND/OR CORRECTED WITH
      THE INITIAL VALUE, THE BIAS OR AN ARBITRARY OFFSET.

**********************************************************************

      PARAMETERS:

      MN    =   10   : MAXIMUM NUMBER OF SIGNALS
      MSAMP =  500   : MAXIMUM NUMBER OF SAMPLES

**********************************************************************

      VARIABLES USED:

      FILN1             [CHAR] : NAME OF THE INPUT-FILE
      FILN2             [CHAR] : NAME OF THE PICOS STRUCTURED OUTPUT-FILE
      NSIG              [INT]  : NUMBER OF SIGNALS
      NSAMP             [INT]  : NUMBER OF SAMPLES
      NR                [INT]  : SAMPLENUMBER
      T                 [INT]  : REFERENCE TIME
      SYSIOS            [REAL] : MATRIX CONTAINING I/O-SAMPLES
      SYSIO             [DBL]  : COPY OF A ROW OF SYSIOS
      DELAY             [INT]  : VECTOR CONTAINING DELAY-TIMES
      MAXDEL            [INT]  : MAXIMUM DELAY

:*********************************************************************
```

```
C*****************************************************************
C                                                                 *
                        PROGRAM HANKEL
C                                                                 *
C*****************************************************************
C
c       This program creates an (approximate) decomposition of a
c       matrix sequence M(k), k=NST-1,NST+NNUM-1 into
c                      k-NST
c        M(k)=~ C A       B + D delta(k-NST+1).
c
c       with C = [NIQxNDIM]
c            A = [NDIMxNDIM]
c            B = [NDIMxNIP]
c            D = [NIQxNIP]
c
C       In a specific form this program creates an (approximate)
c       realization in state space form:
c               x(k+1) = A x(k) + B u(k)
c                 y(k) = C x(k) + D u(k)
c       based on a sequence of Markov parameters: M(k), k=0,NNUM
c       In terms of system identification this emans that a model is
c       constructed based upon impulse response data.
c
c       Exact modelling (minimal realization problem).
c       If a sequence of Markov parameters has been generated by a
c       state space model as described above with dimension NDIM,
c       then this state space model (or an equivalent form) will be
c       reproduced by this program, if the number of Markov parameters
c       to take into account is such that a singular Hankel matrix can be
c       constructed that has rank NDIM.
c
c       ************** Outline of the program ***********************
c
c       Given a sequence of NNUM matrices MARK(k), k=NST,..,NST+NNUM-1.
c       Based on these matrices a block Hankel matrix is constructed:
c
c               MARK(NST)     MARK(NST+1) . . . . . . . MARK(NST+NRC-1)
c               MARK(NST+1)   MARK(NST+2) . . . . . . . MARK(NST+NRC)
c                   .             .          . . . .          .
c                   .             .            . . .          .
c       HANK =      .             .          . . . .          .
c                   .             .          . . . .          .
c               MARK(NST+NRB-1)   .          . . . .     MARK(NST+NNUM-1)
c
c       Matrix HANK will have block dimensiosns [NRBxNRC] and real
c       dimensions [NROWxNCOL].
c       The choice of NRB, NRC that have to fulfil NRB+NRC=NNUM+1, can
c       be made by the user by choosing one of the options:
c          [1] : Automatic choice creating HANK as square as possible;
c          [2] : Automatic choice creating HANK as block square as
c                possible;
c          [3] : Manual input of NRB and NRC with NRB+NRC=NNUM+1.
c
c       A singular value decomposition is performed on this matrix:
c
c
c               HANK = MATU . SINGV . MATVT
c
c       with SINGV a diagonal matrix with ordered elements on the
c       diagonal, being greater than or equal to zero.
c
c       Moreover:          T                                  T
c                     (MATU)  . MATU = I and MATVT . (MATVT)  = I
```

The number of positive elements in SINGV equal the rank of the decomposed Hankel matrix.
The dimension of the model to be constructed (NDIM) now is chosen on the basis of these singular values.

There are three basic algorithms implemented for constructing the model {A,B,C,D}
1. Algorithm of van Zee/Bosgra,
2. Algorithm of Damen/Hajdasinski,
3. Algorithm of Kung;
All of these algorithms are based on the realization method of Ho/Kalman, in the modified version of Zeiger/McEwen.
An overview of the theory that is concerned with these algorithms, and an overview of corresponding references, can be found in:

* Van den Hof, P.M.J., "Approximate realization of noisy linear multivariable systems", Journal A, Vol. 25, 1984, No. 1, pp. 21-26.

* Van den Hof, P.M.J., "Approximate realization of noisy linear systems; the Hankel and Page matrix approach", M.Sc. Thesis, Eindhoven Univ. Techn., Dep. Electr. Eng., Group Measurement and Control, Dec. 1982.

* Damen, A.A.H. and A.K. Hajdasinski (1982) Practical tests with different approximate realizations based on the singular value decomposition of the Hankel matrix. In: Proc. 6th IFAC Symp. Identification and System Par. Estim., Washington D.C., June 1982, pp. 809-814.

****** Input information to be entered from the keyboard ****

FILN1 Name of an existing standard file containing a matrix sequence;
NNUM Number of matrices to take into account in the Hankel matrix;
NIP,NIQ Dimensions of the matrix sequence to take into account (possibly smaller than the available dimensions);
NST Index of first matrix element in the Hankel matrix;
NDIM A fixed choice for the dimension of the model to be created, or choice of the dimension based on the singular values of the decomposed Hankel matrix;
FILN2(5:10) A six character string for composing the name of the output file(s).
IRECM Integer for deciding if a seqeunce of reconstructed matrices has to be stored in a file;
IEIGV Integer for deciding if the eigenvalues of the resulting matrix A have to be calculated.
IOFF Integer for deciding if an offset vector has to be copied to the model, from an existing matrix file.
FILN5 Name of the matrix file containing this offset vector.

********** OUTPUT INFORMATION **********

Output files created:
HANK_____.SSM standard matrix file with state space model;
HANK_____.DOC (nonstandard) text file with documentation on the program run;
HANK_____.MAR on request: a sample file containing the reconstructed matrix seqeunce, ordered column wise;

```
cad     HANK_____.SGV  standard sample file containing the singular
cad                    values of the Hankel matrix.
c
c       On output a set of error values is calculated, relating the
c       original matrix sequence to the reconstructed matrix sequenece
c       in order to evaluate the performance of the approximate model.
c       For each entry in the matrix the absolute RMS error and the
c       relative RMS error are calculated over a given range of matrices.
c
c       abs. RMS(i,j) =
c
c                              NST+NNUM-1                                2
c             SQRT( 1/NNUM       SUM     (MARK(i,j,k)-MARK    (i,j,k))  )
c                               k=NST                        rec
c
c       rel. RMS(i,j) =
c
c                              NST+NNUM-1                                2
c                                SUM     (MARK(i,j,k)-MARK    (i,j,k))  )
c                               k=NST                        rec
c                    SQRT    (  ---------------------------------------  )
c
c                                      NST+NNUM-1             2
c                                        SUM     (MARK(i,j,k))
c                                       k=NST
c
c ------------------------------------------------------------------------
c
c
C       Maximum size of matrices M(k) = 5 x 5;              [MPIQxMPIP]
c       Maximum size of Hankel matrix = 100 x 100;          [MPROWxMPCOL]
c       Maximum dimension of the realization = 50;          [MPDIM]
c       Maximum range of matrices M(k): index -10 to index 200.
c                                                           [MPLOW,MPUP]
c       These maximum sizes can easily be altered, by changing the
c       appropriate parameters, indicated at the end of the respective
c       line.
c
c
C       Date: 04-05-1986
C       Author: Paul Van den Hof
c
c       Copyright:
c               P.M.J. Van den Hof
c               Delft University of Technology
c               Department of Mechanical Engineering
c               Laboratrory for Measurement and Control
c               Mekelweg 2, 2628 CD  Delft
c               The Netherlands.
c               Tel. 015 - 784509.
C
C       Version 1.1     Date: 21-5-1986
c               Fortran version adapted to standard F-77;
c               Dealing with situation of static model.
c       Version 1.2     Date: 13-6-1986
c               Fortran version completed with ERFILE/LIB
C       Version 1.3     Date: 19-6-1986
c               Scale/offset information in D*8 accuracy.
C       Version 1.4     Date: 23-6-1986
c               Corrections in Kung realization method implementation;
C       Version 1.5     Date: 2-10-1986
c               Corrections in printing the singular values: the singu-
c               lar values of the firstly created H-matrix are printed;
c               Extension of output file names to 4+6 characters.
```

```
      Version 1.6     Date: 3-2-1987
              Documentation added.
      Version 1.7     Date: 24-3-1987
              Minor corrections in checking on entered values.
      Version 1.8     Date:  2-4-1987
              Adding the possibility to copy an offset vector in the
              resulting state space model. Adding an extra check on
              the value of NNUM in order to guarantee correspondence
              with parameter statements. (NUMUP)
      Version 1.9     Date: 24-4-1987
              Minor corrections in constructing file names.
.d    Version 1.10    Date: 19-11-1987
.d            Adding construstion of standard sample file with
.d            singular values of Hankel matrix
.d            Modification made by: Karsten Pronk
```

```
C*************************************************************
C*                                                           *
      SUBROUTINE MATSVE(LUNIT,NMAX,MMAX,NSAMP,IP,IQ,X)
C*                                                           *
C*************************************************************
C
C
C     VERSION        : 1.0
C     DATE           : 28-9-1987
C     AUHOR          : KARSTEN PRONK
C     INSTITUTE      : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                      FACULTY OF ELECTRICAL ENGINEERING
C     GROUP          : MEASUREMENT AND CONTROL
C     ADDRESS        : P.O. BOX 513
C                      5600 MB  EINDHOVEN
C     COUNTRY        : THE NETHERLANDS
C
C
C     PARAMETER DESRIPTION:
C
C     LUNIT      [INT]  : LOGICAL UNIT NUMBER
C     NMAX       [INT]  : ROW-DIMENSION OF X
C     MMAX       [INT]  : COLUMN-DIMENSION OF X
C     NSAMP      [INT]  : NUMBER OF SAMPLES
C                         NSAMP<=NMAX
C     IP         [INT]  : NUMBER OF INPUTS
C     IQ         [INT]  : NUMBER OF OUTPUTS
C                         IP+IQ<=MMAX
C     X          [DBLE] : ARRAY CONTAINING THE I/O-SAMPLES
C
C
C     SUBROUTINE DESCRIPTION:
C
C     THIS SUBROUTINE WRITES THE CONTENTS OF THE ARRAY X
C     IN A MATLAB STRUCTURED SAMPLE FILE.
C
C     MATLAB STRUCTURE:
C
C     NSAMP,IP,IQ          FORMAT(I5,2I1)
C     IP inputs            FORMAT((4Z18))   1ST SAMPLE MOMENT
C     IQ outputs           FORMAT((4Z18))   1ST SAMPLE MOMENT
C       ..                    ..                ..
C     IQ outputs            ..               NSAMPst SAMPLE MOMENT
C
C*************************************************************
```

```
*****************************************************************************
*
                              PROGRAM PCAN01
*                                                                  VERSION 1.1
*****************************************************************************

      VERSION          : 1.0
      DATE             : 25-11-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      VERSION          : 1.1
      DATE             : 10-2-1988
      MODIFIED by      : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      PROGRAM DESCRIPTION:

      THIS PROGRAM DETERMINES A PSEUDO-CANONICAL OBSERVABILITY
      REALISATION OF A GIVEN STATE SPACE MODEL.


      REASON FOR MODIFICATION:

      IF IQ=1 THEN MU(1)=IN

*****************************************************************************

     PARAMETERS:

     MIN   = 10 : MAXIMUM DIMENSION OF THE MODEL
     MIP   =  6 : MAXIMUM NUMBER OF INPUTS
     MIQ   =  6 : MAXIMUM NUMBER OF OUTPUTS

*****************************************************************************

     VARIABLES ENTERED BY THE USER:

     FILN      [CHAR*40]  : NAME OF THE ER-STRUCTURED INPUT MODEL FILE
                            CONTAINING THE STATE SPACE MODEL.
     FILOUT    [CHAR*40]  : NAME OF THE ER-STRUCTURED OUTPUT MODEL FILE
                            CONTAING THE OBSERVABILITY FORM ON OUTPUT.
     MU(MIQ)   [INT]      : STRUCTURE INDICES

*****************************************************************************
```

```
C*****************************************************************
C*
                          PROGRAM PRMF01
C                                                      VERSION : 1.0
C*****************************************************************
C
C       VERSION          : 1.0
C       DATE             : 3-12-1987
C       AUTHOR           : KARSTEN PRONK
C       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                          FACULTY OF ELECTRICAL ENGINEERING
C       GROUP            : CONTROL AND MEASUREMENT
C       ADDRESS          : P.O. BOX 513
C                          5600 MB  EINDHOVEN
C       COUNTRY          : THE NETHERLANDS
C
C
C       PROGRAM DESCRIPTION:
C
C       THIS PROGRAM READS THE CONTENTS OF A STANDARD STRUCTURED MATRIX-
C       FILE AND PUTS THEM IN A DOCUMENT FILE.
C
C*****************************************************************
C
C       PARAMETERS:
C
C       MNUM   =  15 : MAXIMUM NUMBER OF MATRICSE TO BE PRINTED
C       MX,MY  =  50 : DECLARED FIRST AND SECOND DIMENSION OF THE MATRICES
C       MZL    =   1 : DECLARED LOWER THIRD DIMENSION
C       MZUP   =  30 : DECLARED UPPER THIRD DIMENSION
C
C*****************************************************************
C
C       VARIABLES ENTERED BY THE USER:
C
C       FILN     [CHAR*40]   : NAME OF THE ER-STRUCTURED INPUT MATRIX FILE
C       FILOUT   [CHAR*40]   : NAME OF THE OUTPUT DOCUMENT FILE
C
C*****************************************************************
```

```
C*************************************************************************

                              PROGRAM PRSF01
                                                        VERSION : 1.1
C*************************************************************************

      VERSION          : 1.0
      DATE             : 21-9-1987
      AUTHOR           : Eric van Beuningen
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      VERSION          : 1.1
      DATE             : 3-12-1987
      MODIFIED by      : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          :
      COUNTRY          :

      PROGRAM DESCRIPTION:
      This program writes the content of a standard structured sample
      file into a printable output file.The file is supposed to contain
      data in double precision accuracy.


      REASON FOR MODIFICATION:

      Output-format modifications.

C*************************************************************************

      LOCAL VARIABLES: NONE
      GLOBAL VARIABLES:
        FILN       (CH*40)   : Name of input file;
        FILN1      (CH*40)   : Name of output file;
        NSIG       (INT)     : Actual number of signals copied;
        NSAM       (INT)     : Actual number of samples copied;
        NSTTIM     (INT)     : First sample moment to be copied;
        SBTYP1     (INT)     : Subtype of sample file;
        SBTYP2     (INT)     : Sub-subtype of sample file;
        TITLE      (CH*8)    : Title of the data set;
        NAME(MSIG) (CH*8)    : Names of the signals;
        ISCOF      (INT)     : Scale and offset information available;
        LSCOF      (LOG)     : Logical for scale/offset correction of data;
        SCALE(MSIG)(DBLE)    : Scale parameters;
        OFFSET(MSIG)(DBLE)   : Offset parameters;
        IFAIL      (INT)     : Error code;
        SYSIO(MSAM,MSIG)
                   (DBLE)    : Signal matrix;
        NSTCOP     (INT)     : First sample moment to be printed;
        NSAMCP     (INT)     : Number of samples to be printed;
        NS         (INT)     : MIN(NSAMCP,NSTTIM+NSAM-NSTCOP)
        DAT        (CH*9)    : Data of creation;
        TIM        (CH*8)    : Time of creation;
        FORM,IFH   (INT)     : Auxiliary variables;
        I,J        (INT)     : Counters;
```

```
C*******************************************************************************
C*
                              PROGRAM RMSM02
C*                                                              VERSION : 1.0
C*******************************************************************************
C
C       VERSION          : 1.0
C       DATE             : 7-12-1987
C       AUTHOR           : KARSTEN PRONK
C       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                          FACULTY OF ELECTRICAL ENGINEERING
C       GROUP            : CONTROL AND MEASUREMENT
C       ADDRESS          : P.O. BOX 513
C                          5600 MB  EINDHOVEN
C       COUNTRY          : THE NETHERLANDS
C
C       PROGRAM DESCRIPTION:
C
C       THIS PROGRAM CALCULATES SQUARED ERRORS BETWEEN AND SQUARED VALUES
C       OF SIGNALS STORED IN ER-STRUCTURED DATA FILES. THE RESIDUES ARE
C       SAVED IN A ER-STRUCTURED DATA FILE.
C
C*******************************************************************************
C
C     PARAMETERS:
C
C     MSIG  =   30 : MAXIMUM NUMBER OF SIGNALS
C     MSAMP = 3000 : MAXIMUM NUMBER OF SAMPLES
C
C*******************************************************************************
C
C     VARIABLES ENTERED BY THE USER:
C
C     FILN1   [CHAR*40]  : NAME OF ER-STRUCTURED INPUT SAMPLE FILE
C     FILN2   [CHAR*40]  : NAME OF OUTPUT DOCUMENT FILE (OPTIONALLY)
C     FILOUT  [CHAR*40]  : NAME OF ER-STRUCTURED OUTPUT SAMPLE FILE CONTAIN
C                          THE RESIDUS ON OUTPUT.
C     ISTART  [INT]      : FIRST SAMPLE TO BE CONSIDERED
C     ISAMP   [INT]      : NUMBER OF SAMPLES TO BE CONSIDERED
C
C*******************************************************************************
```

```
*****************************************************************************
*
                                PROGRAM TRMATM
*                                                                  VERSION 1.1
*****************************************************************************

      VERSION          : 1.0
      DATE             : 11-11-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      VERSION          : 1.1
      DATE             : 21-12-1987
      MODIFIED by      : KARSTEN PRONK
      INSTITUTE        :     IDEM.
      GROUP            :
      ADDRESS          :
      COUNTRY          :

      PROGRAM DESCRIPTION:

      THIS PROGRAM CONVERTS A MATLAB STRUCTURED MATRIX FILE
      IN AN ER-STRUCTURED MATRIX FILE.


      REASON FOR MODIFICATION:

      DATE AND TIME SETTINGS INCLUDED.

      SEE ALSO DOCUMENTATION:


*****************************************************************************

     PARAMETERS :

     MNUM  =  15 : MAXIMUM NUMBER OF MATRICES
     MROW  =  50 : MAXIMUM ROW DIMENSION OF MATRICES
     MCOL  =  50 : MAXIMUM COLUMN DIMENSION OF MATRICES
     MZL   =   1 : DECLARED LOWER THIRD DIMENSION
     MZUP  =   3 : DECLARED UPPER THIRD DIMENSION

*****************************************************************************

     VARABLES ENTERED BY THE USER:

      FILN     [CHAR*40]   : NAME OF MATLAB-STRUCTURED INPUT MATRIX FILE
      FILOUT   [CHAR*40]   : NAME OF ER-STRUCTURED OUTPUT MATRIX FILE
      NUM      [INT]       : NUMBER OF MATRICES TO BE CONVERTED
      SBTYP1   [INT]       : SUBTYPE OF THE OUTPUT FILE
      SBTYP2   [INT]       : SUB-SUBTYPE OF THE OUTPUT FILE
      TITLE    [CHAR*8]    : TITLE OF THE CREATED DATASET

*****************************************************************************
```

```
C*****************************************************************
C*
                          PROGRAM TRMATS
C*                                                        VERSION 1.0
C*****************************************************************
c
c       VERSION          : 1.0
c       DATE             : 12-11-1987
c       AUTHOR           : KARSTEN PRONK
c       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
c                          FACULTY OF ELECTRICAL ENGINEERING
c       GROUP            : CONTROL AND MEASUREMENT
c       ADDRESS          : P.O. BOX 513
c                          5600 MB  EINDHOVEN
c       COUNTRY          : THE NETHERLANDS
c
c       PROGRAM DESCRIPTION:
c
c       THIS PROGRAM TRANSFERS A MATLAB STRUCTURED SAMPLE FILE
c       IN AN ER-STRUCTURED SAMPLE FILE.
c
C*****************************************************************
C
C       PARAMETERS:
C
C       MSIG =    10   : MAXIMUM NUMBER OF SIGNALS
C       MSAMP= 1000    : MAXIMUM NUMBER OF SAMPLES
C
C*****************************************************************
C
C       VARIABLES ENTERED BY THE USER:
C
C       FILN         [CHAR*40] : NAME OF THE MATLAB-STRUCTURED INPUT SAMPLE FI
C       FILOUT       [CHAR*40] : NAME OF THE ER-STRUCTURED OUTPUT SAMPLE FILE
C       NSCOP        [INT]     : NUMBER OF SAMPLES TO BE CONVERTED
C       NLOW         [INT]     : FIRST SAMPLE TO BE CONVERTED
C       SBTYP1       [INT]     : SUBTYPE OF THE OUTPUT FILE
C       NAME(NSIG)   [CHAR*8]  : NAMES OF THE SIGNALS
C       TITLE        [CHAR*8]  : TITLE OF THE CREATED DATASET
C
C*****************************************************************
```

```
*************************************************************************

                          PROGRAM TRMMAT
                                                          VERSION 1.0
*************************************************************************

      VERSION          : 1.0
      DATE             : 6-11-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      PROGRAM DESCRIPTION:

      THIS PROGRAM TRANSFORMS AN ER-STRUCTURED MATRIX FILE IN
      A MATLAB STRUCTURED MATRIX FILE.
      REMARK : A MATLAB STRUCTURED MATRIX FILE CAN ONLY CONTAIN
               TWO-DIMENSIONAL MATRICES !!!!!!!!!
               IF THE PROGRAM TRIES TO COPY A THREE-DIMENSIONAL
               MATRIX ONLY THE FIRST PAGE WILL BE COPIED.

*************************************************************************

    PARAMETERS:

    MNUM = 15 : MAXIMUM NUMBER OF MATRICES TO BE CONVERTED
    MX,MY= 50 : DECLARED FIRST AND SECOND DIMENSION OF MATRICES
    MZL  =  1 : DECLARED LOWER THIRD DIMENSION
    MZUP = 10 : DECLARED UPPER THIRD DIMENSION

*************************************************************************

    VARIABLES ENTERED BY THE USER:

    FILN      [CHAR*40] : NAME OF THE ER-STRUCTURED INPUT MATRIX FILE
    FILOUT    [CHAR*40] : NAME OF THE MATLAB-STRUCTURED OUTPUT MATRIX FILE

*************************************************************************
```

```
C*****************************************************************
C                                                                *
                        PROGRAM TRSM02
C                                                       Version 1.1    *
C*****************************************************************
C
C       This program transforms a general PICOS structured signal data
C       file into a standard ER structured sample file.
c       The extra information that is stored in the standard structured
c       sample file has to be entered by the user.
C       The accuracy of the PICOS file is single precision, of the
c       standard ER structured file is double precision.
c
c       On input a PICOS-formatted IO data file has to be available;
c       the name of the file has to be entered from the keyboard. Extra
c       information on the PICOS file also has to be entered:
c               - the number of samples to be copied,
c               - the number of signals to be copied,
c               - the names of the signals, and
c               - a title for the data set, and the subtype indication.
c
c       The PICOS format: sequential, formatted:
c               1X,I5,<NSIGN>(1X,E14.7) in single precision,
c       where NSIGN is the number of signals in the PICOS-file.
c
C
c       Routines: WRBI02        Writing base/info section of the file;
c                 WRSD01        Writing data section of the file;
c
c       Files:  FILN1 (Unit=LUN1):      Input PICOS file;
c               FILN2 (Unit=LUN2):      Output standard ER structured
c                                       sample file;
c
c       Paremeter statements:
c               MPSIG=30        Max. number of signals to be copied;
c
c
C       Author: Paul Van den Hof
c       Date:   27-6-1986
c
c       Copyright:
c       P.M.J. Van den Hof
c       Group Measurement and Control
c       Dept. Electrical Engineering
c       Eindhoven University of Technology
c       P.O. Box 513, 5600 MB  Eindhoven
c       The Netherlands
c
c       Version 1.0     Date: 27-6-1986
c               Fortran version is an adapted version of DTIOPT, and
c               satisfies the F-77 standard.
c
c       VERSION         : 1.1
C       DATE            : 17-9-1987
C       MODIFIED by     : Karsten Pronk
C       INSTITUTE       : EINDHOVEN UNIVERSITY OF TECHNOLOGY
C                         FACULTY OF ELECTRICAL ENGINEERING
C       GROUP           : CONTROL AND MEASUREMENT
C       ADDRESS         : P.O. BOX 513
C                         5600 MB EINDHOVEN
C       COUNTRY         : THE NETHERLANDS
C
C
C
```

```
      EXTENSIONS TO FORTRAN 77:

      FORMAT(<NSIG>E14.7) IS A REPEATING FORMAT:
         THE FORMAT E14.7 IS NSIG TIMES REPEATED

      REASON FOR MODIFICATION:

      IN VERSION 1.0 THE REPEATING FORMAT ((E14.7) IS USED
      THIS FORMAT STARTS READING ON A NEW LINE EACH REPETITION

      THE FORMAT IN LINE 9090 IS ADJUSTED ACCORDING TO PICOS-STANDARDS
      PICOS FORMAT:
      I5,1X,I5,<NSIG>(1X,E14.7)

*****************************************************************************

      Variables:

      NSIG    Number of signals to be copied;
      NSAMP   Number of samples to be copied;
      NST     Time moment of first sample to be copied;
      STREC   Start record nr. of data section in output file;
      MS      Declared row dimension of SYSIO
      NS      Actual row dimension of SYSIO
      SYSIO   Double precision row vector with signals;
      SYSIOS  Real row vector with signals;
      NLOW    First row of SYSIO to be copied;
      ISCOF   Indication for presence of scale/offset information;
*****************************************************************************
```

```
C*****************************************************************************
C*
                                  PROGRAM TRSMAT
C*                                                                    VERSION 1.
C*****************************************************************************
c
c       VERSION          : 1.0
c       DATE             : 6-11-1986
c       AUTHOR           : KARSTEN PRONK
c       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
c                          FACULTY OF ELECTRICAL ENGINEERING
c       GROUP            : CONTROL AND MEASUREMENT
c       ADDRESS          : P.O. BOX 513
c                          5600 MB  EINDHOVEN
c       COUNTRY          : THE NETHERLANDS
c
c
c       PROGRAM DESCRIPTION:
c
c       THIS PROGRAM CONVERTS AN ER-STRUCTURED SAMPLE-FILE IN
c       A MATLAB STRUCTURED SAMPLE FILE.
c
C*****************************************************************************
C
C       Parameters
C       MSIG = 10        (INT)    :       Maximum number of signals copied
C       MSAM = 1000      (INT)    :       Maximum number of signals copied
C
C       Variables
C       FILN     (CH*40) :       Filename
C       OUTFIL   (CH*40) :       Filename
C       IODAUN   (INT)   :       Logical Unit Number for communication
C       UNIN     (INT)   :       Input Logical Unit Number
C       UNOUT    (INT)   :       Output Logical Unit Number
C       NSIG     (INT)   :       Actual number of samples copied
C       NSAM     (INT)   :       Actual number of samples copied
C       NSTTIM   (INT)   :       First sample moment to be copied
C       SBTYP1   (INT)   :       Subtype of sample file
C       SBTYP2   (INT)   :       Subsubtype of sample file
C       TITLE    (CH*8)  :       Title of the data set
C       NAME(MSIG) (CH*8) :      Names of the signals
C       ISCOF    (INT)   :       Scale and offset invormation available
C                                ISCOF = 0 : NO
C                                ISCOF = 1 : YES
C       LSCOF    (LOG)   :       Logical for scale/offset correction of
C                                                                  data
C       SCALE(MSIG) (DBLE) :     Scale parameters
C       OFFSET(MSIG) (DBLE) :    Offset parameters
C       IFAIL    (INT)   :       ERROR code
C       SYSIO(MSAM,MSIG)
C                (DBLE)  :       Signal matrix
C       OUTF     (CHAR*(*)) :    Name of output file  Maximum length 10
C
C*****************************************************************************
```

```
*****************************************************************************

                              PROGRAM TRSPIC
                                                              VERSION 1.0
*****************************************************************************

      VERSION          : 1.0
      DATE             : 9-2-1988
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      PROGRAM DESCRIPTION:

      THIS PROGRAM CONVERTS AN ER-STRUCTURED SAMPLE-FILE IN
      A PICOS  STRUCTURED SAMPLE FILE.

*****************************************************************************

      Parameters
      MSAM = 10        (INT)    :       Maximum number of samples copied
      MSIG = 1000      (INT)    :       Maximum number of signals copied

      Variables
      FILN     (CH*40) :       Filename
      OUTFIL   (CH*40) :       Filename
      IODAUN   (INT)   :       Logical Unit Number for communication
      UNIN     (INT)   :       Input Logical Unit Number
      UNOUT    (INT)   :       Output Logical Unit Number
      NSIG     (INT)   :       Actual number of signals copied
      NSAM     (INT)   :       Actual number of samples copied
      NSTTIM   (INT)   :       First sample moment to be copied
      SBTYP1   (INT)   :       Subtype of sample file
      SBTYP2   (INT)   :       Subsubtype of sample file
      TITLE    (CH*8)  :       Title of the data set
      NAME(MSIG) (CH*8) :      Names of the signals
      ISCOF    (INT)   :       Scale and offset invormation available
                               ISCOF = 0 : NO
                               ISCOF = 1 : YES
      LSCOF    (LOG)   :       Logical for scale/offset correction of
                                                                  data
      SCALE(MSIG) (DBLE) :     Scale parameters
      OFFSET(MSIG) (DBLE) :    Offset parameters
      IFAIL    (INT)   :       ERROR code
      SYSIO(MSAM,MSIG)
               (DBLE)  :       Signal matrix
      OUTF     (CHAR*(*)) :    Name of output file  Maximum length 10

*****************************************************************************
```

```
C*******************************************************************************
C*
                              PROGRAM TRTHER01
C*                                                                    Version 1.0
C*******************************************************************************
c
c       VERSION          : 1.0
c       DATE             : 29-10-1987
c       AUTHOR           : KARSTEN PRONK
c       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
c                          FACULTY OF ELECTRICAL ENGINEERING
c       GROUP            : CONTROL AND MEASUREMENT
c       ADDRESS          : P.O. BOX 513
c                          5600 MB  EINDHOVEN
c       COUNTRY          : THE NETHERLANDS
c
c
c       PROGRAM DESCRIPTION:
c
c       THIS PROGRAM TRANSFORMS A THE-STRUCTURED MODEL-FILE (A,B,C,D,OFF,X0
c       IN AN ER-STRUCTURED MODEL-FILE.
c
c*******************************************************************************
C
C       PARAMETERS:
C
C       MNUM = 10 : MAXIMUM NUMBER OF MATRUCES TO BE READ
C       MIN  = 10 : MAXIMUM ORDER OF THE MODEL
C       MIP  =  5 : MAXIMUM NUMBER OF INPUTS
C       MIQ  = 15 : MAXIMUM NUMBER OF OUTPUTS
C       IDUM =  5 : DIMENSION DUMMY ARAY
C
C*******************************************************************************
C
C       VARIABLES ENTERED BY THE USER:
C
C       FILN    [CHAR*40] : NAME OF TH-STRUCTURED INPUT MODELFILE CONTAININ
C                           A STATE SPACE MODEL.
C       FILOUT  [CHAR*40] : NAME OF ER-STRUCTURED OUTPUT MODEL FILE
C
C*******************************************************************************
```

```
*****************************************************************************

                              SUBROUTINE WRMD06
                                                          VERSION  : 1.0
*****************************************************************************

      This subroutine writes 6 three-dimensional matrices into
      the data section of a standard structured matrix file.
      The output file is assumed to be opened at l.u.n. UN.
      Data file standard = 2.

     SUBROUTINE WRMD06 (UN,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6,MX,MY,MZL,
    1                   NZUP,NX,NY,NZL,BEGR,IERR)

*****************************************************************************


      VERSION          : 1.0
      DATE             : 2-12-1987
      AUTHOR           : KARSTEN PRONK
      INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
                         FACULTY OF ELECTRICAL ENGINEERING
      GROUP            : CONTROL AND MEASUREMENT
      ADDRESS          : P.O. BOX 513
                         5600 MB  EINDHOVEN
      COUNTRY          : THE NETHERLANDS


      SUBROUTINE DESCRIPTION:

      This subroutine writes 6 three-dimensional matrices into
      the data section of a standard structured matrix file.
      The output file is assumed to be opened at l.u.n. UN.
      Data file standard = 2.

*****************************************************************************

      Parameter list:
      UN      (INT)    (i): Logical unit number;
      MAT1(MX(1),MY(1),MZL(1):NZUP(1))
              (DBL)    (I): Matrix nr. 1;
      MAT2(MX(2),MY(2),MZL(2):NZUP(2))
              (DBL)    (I): Matrix nr. 2;
      MAT3(MX(3),MY(3),MZL(3):NZUP(3))
              (DBL)    (I): Matrix nr. 3;
      MAT4(MX(4),MY(4),MZL(4):NZUP(4))
              (DBL)    (I): Matrix nr. 4;
      MAT5(MX(5),MY(5),MZL(5):NZUP(5))
              (DBL)    (I): Matrix nr. 5;
      MAT6(MX(6),MY(6),MZL(6):NZUP(6))
              (DBL)    (I): Matrix nr. 6;
      MX(6)    (INT)   (i): Declared row dimension of the matrices;
      MY(6)    (INT)   (i): Declared column dimension of the matrices;
      MZL(6)   (INT)   (i): Declared lower bounds of third indices;
      NZUP(6)  (INT)   (i): Upper bound third index of the matrices;
      NX(6)    (INT)   (i): Actual row dimensions of the matrices;
      NY(6)    (INT)   (i): Actual column dimensions of the matrices;
      NZL(6)   (INT)   (i): Actual lower bounds of third indices;
      BEGR(6)  (INT)   (i): Start records of the data sections in file;
      IERR     (INT)   (o): Error code:
                            = 0 : successful completion;
                            = 1 : error in parameter list;
                            = 2 : error in WRITE-statement;
```

```
c*******************************************************************
c*
      SUBROUTINE WRSS03 (FILN,UN,A,B,C,D,OFF,X0,MIP,MIQ,MDIM,NIP,NIQ,
     1                   NDIM,SBTYP2,TITLE,IERR)
c*                                                        VERSION  : 1.0
c*******************************************************************
c
c       VERSION          : 1.0
c       DATE             : 2-12-1987
c       AUTHOR           : KARSTEN PRONK
c       INSTITUTE        : EINDHOVEN UNIVERSITY OF TECHNOLOGY
c                          FACULTY OF ELECTRICAL ENGINEERING
c       GROUP            : CONTROL AND MEASUREMENT
c       ADDRESS          : P.O. BOX 513
c                          5600 MB  EINDHOVEN
c       COUNTRY          : THE NETHERLANDS
c
c
c       SUBROUTINE DESCRIPTION:
c
c       THIS SUBROUTINE STORES A STATE SPACE MODEL {A,B,C,D,OFF,X0}
c       IN A STANDARD STRUCTURED MODEL FILE.
c
c*******************************************************************
c
C Parameters:
c FILN           (CH*40) : Name of the matrix file;                [I]
c UN             (INT)   : Logical unit number;                    [I]
c A(MDIM,NDIM)   (DBL)   : System matrix;                          [I]
c B(MDIM,NIP)    (DBL)   : Input matrix;                           [I]
c C(MIQ,NDIM)    (DBL)   : Output matrix;                          [I]
c D(MIQ,MIP)     (DBL)   : Input/output matrix;                    [I]
C OFF(NIQ)       (DBL)   : Offset vector on the output;            [I]
C X0(NDIM)       (DBL)   : Initial state vector                    [I]
c MIP            (INT)   : Maximum number of inputs                [I]
c MIQ            (INT)   : Maximum number of outputs               [I]
c MDIM           (INT)   : Maximum state space dimension;          [I]
c NIP            (INT)   : Actual number of inputs;                [I]
c NIQ            (INT)   : Actual number of outputs;               [I]
c NDIM           (INT)   : Actual state space dimension;           [I]
c SBTYP2         (INT)   : Sub-subtype of file;                    [I]
c TITLE          (CH*8)  : Title of data set;                      [I]
c IERR           (INT)   : Error code;                             [O]
c                          = 0 : Successful completion;
c                          = 1 : Error in parameter list;
c                          = 2 : Error during opening of input file;
c                          = 3 : Error in subroutine WRBI01;
c                          = 4 : Error in subroutine WRMD06;
c
c*******************************************************************
C
C       Local parameter:
C       NUM         (INT) : Number of matrices in S.S. model (PARAMETER);
c
c       Local variables:
c       IFAIL       (INT) : Error code
c       NAME(NUM)   (CH*8):Names of the matrices;
c       MX(NUM)     (INT) : Declared row dimensions of the matrices;
c       MY(NUM)     (INT) : Declared column dimensions of the matrices;
C       NX(NUM)     (INT) : Actual row dimensions of the matrices;
c       NY(NUM)     (INT) : Actual column dimensions of the matrices;
C       NZ(NUM)     (INT) : Actual length of third index (NZUP-NZL+1);
c       MZL(NUM)    (INT) : Declared lower bounds for third indices;
c       NZL(NUM)    (INT) : Actual lower bounds for third indices;
```

Appendix B

Jordan forms

As already mentioned in paragraph 4.3 the Jordan form of a state space model is a realisation that decouples all states. In this realisation the system-matrix is a diagonal matrix. The diagonal elements are the eigenvalues of the system matrix. For all systems with distinct eigenvalues this realisation can be found:

Corollary:
For every state space model with distict eigenvalues a transformation matrix T can be found such that $T^{-1}AT$ is a diagonal matrix and the input-output behaviour of the system $\{T^{-1}AT, T^{-1}B, CT, D\}$ is the same as the behaviour of the system $\{A,B,C,D\}$.

Proof:
cf. for example [Trentelman,1985]

In general the eigenvalues of the system matrix may be complex so the elements of the Jordan system matrix can be complex. This can reduce the stability of the algorithms using this form. Moreover for the moment it is impossible to store complex matrices in standard ER-structured model files.
Therefore we try to find a Jordan-like representation of a system with complex poles using only real matrices. To find this representation we use the fact that if a system has a complex pole z then also its conjugent z^* is a pole of the system.

Considering for the moment a simplified system with only one complex polepair $z_{1,2} = a \pm jb$ we obtain the following transfer function:

$$H(z) = \frac{-b}{z^2 - 2az + a^2 + b^2}$$

This second order system can be represented by the following state space realisation:

$$A = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The constructed A-matrix is called a Jordan-cage.
Now we extend our discussion to the case of multiple distinct polepairs. In this case we can construct a system matrix which has Jordan cages on its main diagonal. All the other elements are zero.

$$A = \begin{bmatrix} \begin{matrix} a_1 & b_1 \\ -b_1 & a_1 \end{matrix} & & 0 \\ & \ddots & \\ 0 & & \begin{matrix} a_n & b_n \\ -b_n & a_n \end{matrix} \end{bmatrix}$$

In this way we decoupled every single polepair instead of decoupling every pole of the system. This realisation is called a pseudo-Jordan realisation. There exist other ways to contruct a Jordan-like realisation with real matrices.

Next we want to consider systems with multiple poles. For the moment we concentrate on cyclic systems with (multiple) real poles.
Cyclic systems are characterized by the fact that the geometrical multiplicity of each eigenvalue is one. This means that the dimension of the eigenspace for each eigenvalue is one. The algebraic multiplicity on the other hand is the multiplicity of an eigenvalue in the charcteristic equation. This means that the algebraic multiplicity is always larger or equal to the geometric multiplicity. For cyclic systems both multiplicities are one for all poles. All SISO systems are cyclic.
If we consider a SISO systems with multiple poles we know that it is impossible to decouple all states. Considering for example the system described by the transfer function:

$$H(z) = \frac{1}{(z-1)^2}$$

then trying to write H(z) as A/(z-1) + B/(z-1) would lead to a contradiction.

For this situation we use another Jordan-like representation given below:

$$A = \begin{bmatrix} \lambda_1 & 1 & & & \\ & \lambda_1 & 1 & & \\ & & \lambda_1 & & 0 \\ & & & \ddots & \\ 0 & & & & \lambda_n \end{bmatrix}$$

In other words: Jordan-like blocks are constructed on the main diagonal. The elements of the main diagonal are the eigenvalues of the system matrix and the elements of the upper subdiagonal of each block (!!) are one. All other elements are zero.
Physically we can interprete this realisation as a decoupling of all distinct eigenvalues putting all multiple eigenvalues in cascade:

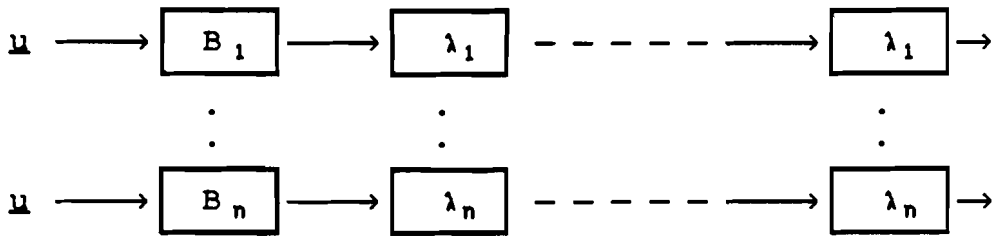


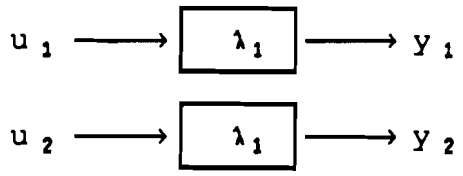
Fig.27 Jordan-like realisation of a system with multiple poles.

Clearly the above mentioned realisations can easily be combined for cyclic systems with multiple complex poles. In this situation we put all multiple polepairs in cascade resulting in the following structure of the system matrix:

$$A = \begin{bmatrix} a_1 & b_1 & & & & \\ -b_1 & a_1 & 1 & & 0 & \\ & & a_1 & b_1 & & \\ & & -b_1 & a_1 & & \\ & & & & \ddots & \\ & 0 & & & a_n & b_n \\ & & & & -b_n & a_n \end{bmatrix}$$

The main diagonal is filled with the Jordan cages we know from our discussion of a system with complex poles. The subdiagonal elements between each identical Jordan cage are filled with ones. All the other elements are zero. In this way we create a number of cascades of Jordan cages for every polepair.

Until now we restricted our discussion to cyclic systems. This restriction holds for most practical situations. Let us, however, now consider the following example:



In this situation we have a second order system with a multiple pole. Both the algebraic and the geometric multiplicity of the pole are two so this is not a cyclic system. Clearly for this situation we can construct a Jordan form:

$$A = \mathrm{diag}(\lambda_1, \lambda_1);\ B = \mathrm{diag}(1,1);\ C = \mathrm{diag}(1,1)$$

This type of realisation can be constructed for all systems with equal geometric and algebraic multiplicity. In general the geometric multiplicity is greater or equal to one and less or equal than the algebraic multiplicity [Trentelman,1985]
For this general case it can be proved [Guidorzi,1984] that a Jordan-like realisation can be constructed. In this situation again the main diagonal is filled with Jordan blocks. In this situation however the number of Jordan blocks for each distinct pole is equal to the geometrical multiplicity of that pole. An algorithm to construct this general Jordan form is given by Guidorzi in his paper. Clearly this realisation is not unique so this is a pseudo-canonical realisation. If we want to avoid using complex matrices we can again replace each polepair by a Jordan cage as shown above.

Name of the experiment : SISO

Estimation program : LS_SSM

Number of iterations : 150

Stop criterion : 0.1000000E-04

Name of the matrix data file: ESSS22IF2.SSM
Date and time of creation : 3-MAR-88 14:22:17
Title of the data set : ********
Model subtype 1 : 10
Model subtype 2 : 2
Data type of variables : D8
Number of matrices : 6

Name	rows	columns	depth
A	2	2	1
B	2	1	1
C	1	2	1
D	1	1	1
OFF	1	1	1
X0	2	1	1

A =

0.00000E+00 0.10000E+01
-0.57817E+00 0.12118E+01

B =

0.29888E-01
0.52231E-01

C =

0.10000E+01 0.00000E+00

D =

0.62640E-01

OFF =

-0.55485E+01

X0 =

Appendix C-2

Name of the experiment : SISO

Estimation program : DIRECTO

Number of iterations : 45

Stop criterion : 0.5268356E-07

Name of the matrix data file: MARK31433.STA
Date and time of creation : 9-MAR-88 09:41:12
Title of the data set : DIRECTO
Model subtype 1 : 30
Model subtype 2 : 1
Data type of variables : D8
Number of matrices : 4

Name	rows	columns	depth
A	1	2	1
MARK	1	3	1
OFF	1	1	1
V	1	1	1

A =

0.12459E+01-0.58752E+00

MARK =

0.55033E-01 0.43455E-01 0.52039E-01

OFF =

-0.77461E+01

V =

0.97336E+03

Appendix C-3

Name of the experiment : SIMO

Estimation program : LS_SSM

Number of iterations : 300

Stop criterion : 0.1000000E-04

Name of the matrix data file: ESSS30IF2.SSM
Date and time of creation : 4-MAR-88 09:14:48
Title of the data set : ********
Model subtype 1 : 10
Model subtype 2 : 3
Data type of variables : D8
Number of matrices : 6

Name	rows	columns	depth
A	3	3	1
B	3	1	1
C	3	3	1
D	3	1	1
OFF	3	1	1
X0	3	1	1

A =

-0.20298E+01 0.12560E+02-0.26840E+02
 0.12156E+01-0.42295E+01 0.11405E+02
 0.79294E+00-0.34706E+01 0.83753E+01

B =

 0.60360E-01
-0.48420E-02
-0.87520E-02

C =

 0.10000E+01 0.00000E+00 0.00000E+00
 0.00000E+00 0.10000E+01 0.00000E+00
 0.00000E+00 0.00000E+00 0.10000E+01

D =

 0.22175E-01
 0.56797E-02
-0.78001E-02

OFF =

Appendix C-4

Name of the experiment : SIMO

Estimation program : DIRECTO

Number of iterations : 54

Stop criterion : 0.5268356E-04

Name of the matrix data file: MARK40857.STA
Date and time of creation : 9-MAR-88 09:45:38
Title of the data set : DIRECTO
Model subtype 1 : 30
Model subtype 2 : 1
Data type of variables : D8
Number of matrices : 4

Name	rows	columns	depth
A	1	2	1
MARK	3	3	1
OFF	3	1	1
V	1	1	1

A =

0.14199E+01-0.67586E+00

MARK =

0.31932E-01 0.38452E-01 0.69405E-01
0.86496E-02-0.88056E-02-0.16283E-01
-0.49398E-02-0.25733E-02-0.12800E-01

OFF =

0.45192E+01
0.39178E+01
-0.28911E+01

V =

0.67056E+04

Name of the experiment : MIMO

Estimation program : LS_SSM

Number of iterations : 300

Stop criterion : 0.1000000E-04

Name of the matrix data file: ESSS33IF2.SSM
Date and time of creation : 3-MAR-88 21:07:34
Title of the data set : ********
Model subtype 1 : 10
Model subtype 2 : 3
Data type of variables : D8
Number of matrices : 6

Name	rows	columns	depth
A	6	6	1
B	6	3	1
C	3	6	1
D	3	3	1
OFF	3	1	1
X0	6	1	1

A =

```
 0.00000E+00 0.10000E+01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000
-0.61220E+00 0.12206E+01 0.16454E+00-0.24790E+00 0.74031E-02-0.65203
 0.00000E+00 0.00000E+00 0.00000E+00 0.10000E+01 0.00000E+00 0.00000
 0.32604E-01 0.66401E-01-0.62458E+00 0.13964E+01-0.13740E-01 0.13655
 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.10000
 0.16496E-01-0.56383E-01 0.98352E-01-0.21745E+00-0.56953E+00 0.12007
```

B =

```
 0.19754E-01-0.78233E-02 0.55495E-02
 0.65629E-01-0.79213E-02-0.10805E-01
 0.32200E-01 0.37332E-01 0.17597E-01
 0.13027E-02 0.70234E-01-0.13917E-01
 0.29270E-02-0.20441E-01 0.38200E-01
-0.16952E-01-0.33323E-02 0.75932E-01
```

C =

```
 0.10000E+01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000
 0.00000E+00 0.00000E+00 0.10000E+01 0.00000E+00 0.00000E+00 0.00000
 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.10000E+01 0.00000
```

D =

```
 0.77512E-01-0.30914E-02-0.17675E-01                    88
-0.33216E-01 0.48745E-01-0.24549E-01
-0.48354E-02 0.15661E-01 0.77801E-01


OFF      =


 0.25138E+01
 0.50400E+00
-0.15628E+01


X0       =


 0.43462E-02
-0.11050E-01
-0.33696E-02
 0.17618E-02
-0.11747E-02
 0.67984E-02
```

Name of the experiment : MIMO

Estimation program : DIRECTO

Number of iterations : 49

Stop criterion : 0.5268356E-07

Name of the matrix data file: MARK31653.STA
Date and time of creation : 7-MAR-88 15:41:23
Title of the data set : WIND.3
Model subtype 1 : 30
Model subtype 2 : 1
Data type of variables : D8
Number of matrices : 4

Name	rows	columns	depth
A	1	6	1
MARK	3	21	1
OFF	3	1	1
V	1	1	1

A =

0.40206E+01-0.73757E+01 0.77458E+01-0.48952E+01 0.17637E+01-0.2880(

MARK =

0.56915E-01-0.68639E-02-0.97834E-04 0.42867E-01-0.13606E-01 0.51018
0.62801E-01-0.75095E-02-0.13022E-01 0.62265E-01-0.10168E-01-0.19774
0.49844E-01-0.16047E-01-0.20707E-01 0.30022E-01-0.16961E-01-0.17446
0.72525E-02-0.90894E-02-0.10363E-01
-0.13483E-01 0.46299E-01-0.25227E-01 0.48988E-03 0.45303E-01-0.12937
-0.41869E-02 0.68600E-01-0.35495E-02-0.70780E-02 0.65186E-01-0.73030
-0.90533E-02 0.51396E-01-0.11834E-01-0.84115E-02 0.32681E-01-0.10216
-0.41928E-02 0.10955E-01-0.26379E-02
-0.10547E-01 0.20327E-01 0.32077E-01 0.41237E-02-0.48781E-02 0.53453
-0.78532E-02-0.14112E-01 0.76786E-01-0.16428E-01-0.16183E-01 0.75707
-0.19732E-01-0.14179E-01 0.57391E-01-0.15342E-01-0.96287E-02 0.27929
-0.44243E-02-0.33881E-02-0.32762E-02

OFF =

0.25302E+01
0.34563E+00
-0.14188E+01

V =

0.42167E+04

Appendix D

List of symbols used

A	system matrix
a(k)	minimal polynomial coefficients
a_i	real part of eigenvalue
B	input matrix
b_i	imaginary part of eigenvalue
C	output matrix
D	direct feed through
E	rotation matrix
e_o	residu
F_d	damping force
F_i	inertia force
F_u	upwards force
F_z	gravitation
H	Hankel matrix containg Markov parameters
h	height of the platform at a corner
M	mass
m	number of Markov parmeters
M(k)	Markov parameter (MIMO)
m(k)	Markov parameter (SISO)
N	number of datapairs
OFF	output offset vector
u	distance between platform and float, input for identification
u_c	input of servomotor
V_o	output error
x	state vector
X0	initial state
y	height of the platform at a measurepoint
Γ	observability matrix
Δ	controllability matrix
λ_i	eigenvalue, pole
μ_i	structure index
σ_i	singular value
Σ	diagonal matrix containing singular values

Appendix E

Literature

[Backx,1987] Backx,A.C.P.M.
Identification of an industrial process
- A Markov parameter approach.
Dr. dissertation TU Eindhoven

[Daanen,1985] Daanen,A.M.J.
Het ontwerpen van een besturing voor een drijvend platform.
Internal report TU Eindhoven,
Group Measurement and Control

[Damen,1986] Damen,A.A.H. and Boom,A.J.W. van den
Toegepaste systeemanalyse
TU Eindhoven, course 5P280

[Dirks,1986] Dirks, P.A.J.
To determine the position of a floating platform.
Internal report TU Eindhoven,
Group Measurement and Control

[Guidorzi,1984] Guidorzi,R.P.
State space decomposition into cyclic subspaces and transformation to the Jordan form.
American Control Conference New York, 1984

[Isermann,1980] Isermann,R.
Practical aspects of Process Identification
Automatica, Vol.16, pp.575-587

[Oudbier,1986] Oudbier,R.S.
A different approach to the minimal polynomial and start sequence of Markov parameters estimation problem (+ documentation DIRECTO)
PICOS-R-070 report

[Trentelman,1985] Trentelman,H.L. and Veltkamp,G.W.
Matrixtheorie 1 en 2
TU Eindhoven, courses 2F400/2F450

[Veltmeyer,1985] Veltmeyer,A.J.W.
Parameter estimation of multivariable processes represented by stochastic models in pseudo canonical form (+ manual LS_SSM)
Internal report TU Eindhoven
Group Measurement and Control