CODEN:LUTFD2/(TFRT-5259)/1-103/(1981)

RECURSIVE IDENTIFICATION USING A GENERAL MODEL STRUCTURE

HALLGRIMUR GUNNARSSON

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA

OKTOBER 1981

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 725
S 220 07 Lund 7 Sweden

Document name
MASTER THESIS

Date of issue
October 1981

Document number
CODEN:LUTFD2/(TFRT-5259)/1-103/(1981)

Author(s)
Hallgrimur Gunnarsson

Supervisor
Ivar Gustavsson

Sponsoring organization

Title and subtitle

Recursive identification using a general model structure

(Rekursiv identifiering för en generell modellstruktur)

Abstract

A program package, based on SIMNON (an interactive simulation program), for recursive identification has been developed. The program package uses a general model structure of the form:

$$(1+A) \cdot y(t) = \frac{B}{1+F} \cdot u(t) + \frac{1+C}{1+D} \cdot e(t) \qquad \text{(I)}$$

where A, F, B, D and C are polynomials in q^{-1} (the backward shift operator) such that A(0)=0 etc. and e(t) white noise. The order of the polynomials can be varied independently to arrive at a model structure that gives the "best" results. For this general structure many common identification methods such as the LS, ELS, ML and GLS can be classified as special cases in either of two approaches (LIP or CMA). Simulation experiments have been made using data generated by a relatively complex "process/system". The results indicate that the use of this general model structure can be recommended since it allows a greater flexibility and the same or a higher degree of accuracy can be achieved with fewer parameters compared to the common identification methods to obtain a good approximate description.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
103

Recipient's notes

Security classification

Distribution: The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

RECURSIVE IDENTIFICATION USING A GENERAL MODEL STRUCTURE

A Program Package
and
Simulations

Hallgrímur Gunnarsson

REFERAT

Ett programpaket, baserad på SIMNON (ett interaktivt simuleringsprogram), för rekursiv identifiering har tagits fram. Programpaketet använder en generell modell/system beskrivning som visas ovan (eq. I), där A, F, B, D och C är polynom i q^{-1} (baklänges shiftoperator). Polynomernas gradtal kan varieras vart för sig för att komma fram till den modellstruktur som ger "bäst" resultat. Detta beskrivningssätt innebär att några av de vanligaste identifieringsmetoderna t. ex. LS, ELS, ML och GLS kan visas vara specialfall i en av två metoder. Simuleringar har gjorts på data genererad av ett relativt komplext system/process. Resultaten indikerar att användandet av detta beskrivningssätt kan rekommenderas eftersom större flexibilitet och samma eller högre nogrannhet kan uppnås med färre antal parametrar jämfört med de vanliga identifieringsmetoderna för att uppnå en bra approximativ beskrivning.

PREFACE

The master thesis presented in this paper was carried out at the Department of Automatic Control, Lund Institute of Technology. I would like to express my gratitute to Tekn. dr. Ivar Gustavsson who initiated this work and Assistant Professor Tekn. dr. Björn Wittenmark who read the final manuscript for their criticism and help. Furthermore thanks to Kristján H. Bjartmarsson and Geir R. Jóhannesson for inspiring discussions and the loan of litterature.

Last but not least the warmest thanks to my wife Steinunn and my daugther Ingileif for their loving care and kindness.

Hallgrímur Gunnarsson

TABLE OF CONTENTS

1 INTRODUCTION

The last decade has seen quite extensive developements in the field of recursive identification. Most simulation and theoretical studies of identification methods have been concerned with their convergence and uniqueness to a systems "true" parameters. More recently some attention has been devoted to the concept of approximate identification i.e. to find, under given experimental conditions, an approximate description which contains the most relevant features of the system under observation rather than to determine its "true" parameters. This is in many cases the most realistic approach since the system is often more complex than a practical model would be. The sucess of adaptive regulators based on too "simple" models seems to justify this point of view. An application of a general model structure, due to L. Ljung, is in this context of a special interest. The model structure and the algorithms are described briefly in sec. 2. Note that many common recursive identification methods, such as the LS, GLS, ELS and ML are included as special cases in either of two approaches.

In order to gain some insight into the practical usefulness of these ideas and other related ones, a program package for simulation studies of recursive identification methods has been developed. The program is based on the interactive program package SIMNON [15] and thus includes all the facilities available there. It is assumed throughout that the user is familiar with SIMNON and its application.

The program package is, in short, best described by the following block diagram (fig 1.1) where selectors are set

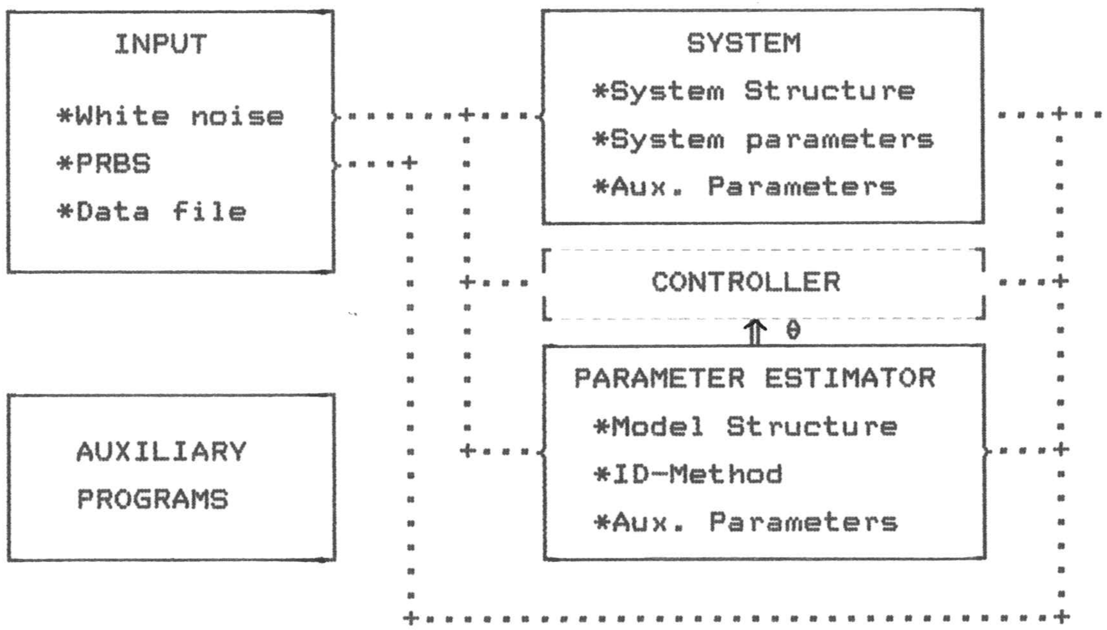


Fig 1.1

and connections made between the subsystems according to the intended use. Notice that SIMNON allows the user to implement his own subsystems e.g. to expand this program to include a regulator (see the dotted part of fig 1.1). Sec. 3 contains a description of the program structure followed by information on program start-up in sec. 4. The subsystems available are described in detail in sec. 5 along with some examples.

The program package has been used to do a series of identification experiments which are reported in sec. 6. There a number of different model structures are tested (using both of the above mentioned approaches/ methods) on input-output sequences generated by a relatively complex system("process"). The results are then compared to get the "best" model structure for this particular system.

For the benefit of those who would like to implement other identification methods into the program package or to use parts of it for their own purposes, a complete listing of all programs as well as some additional information is available at the Department. Contact the program librarian.

2 IDENTIFICATION METHODS AND ALGORITHMS

The application of identification techniques implies that a model is fitted to measured data u(1)......u(N), of the input signal and y(1).......y(N) of the output signal. It seems reasonable to assume that the greater the versatility of the model the greater the possibility of arriving at, in some sense, the best description. This has led to the use of a general model/system structure, due to L. Ljung [1] (see sec. 2.1) and the application of two common identification schemes to this model using two slightly different approaches (sec. 2.2-2.4). In sec. 2.5 a short description of the implemented Instrumental variables algorithm is given and in sec. 2.6 an Extended Kalman Filter Identification Algorithm. Finally sec. 2.7 contains notes on the loss functions used for accuracy valuation.

2.1 A General Model Structure

Consider a system/model described by:

$$(1+A(q^{-1}))\cdot y(t) = \frac{B(q^{-1})}{1+F(q^{-1})}\cdot u(t-Td) + \frac{1+C(q^{-1})}{1+D(q^{-1})}\cdot e(t) \qquad [2.1.1]$$

where A, F, B, D and C are polynomials in q^{-1} (the backward shift operator) of orders n_a, n_f, n_b, n_d and n_c respectively such that A(0)=0, F(0)=0 etc. that is:

$$A(q^{-1}) = a_1 q^{-1} + a_2 q^{-2} + ...a_n q^{-n} \qquad \text{etc.}$$

Td is the time delay and the sequence e(t) is supposed to be white noise.

In applications one or several of the polynomials may be zero. It is interesting to note that:

F = D = C = 0 gives the least squares model
F = D = 0 - the Aström model
F = C = 0 - the generalized least squares model
A = C = D = 0 - the model reference (output error) model

Remark 2.1.1 In the sequel the estimates will be denoted by a hat ^ above the letter. The subscript o is used to denote the true parameters or the system. The capital letters A, B, F, C and D will throughout refer to the above defined polynomials unless specified otherwise.

Introduce the quantities $z_o(t)$ and $v_o(t)$ as:

$$z_o(t) = \frac{B_o(q^{-1})}{1+F_o(q^{-1})} \cdot u(t-Td) \qquad [2.1.2]$$

$$v_o(t) = \frac{1+C_o(q^{-1})}{1+D_o(q^{-1})} \cdot e(t) \qquad [2.1.3]$$

a vector of old signal values φ_o and the parameter vector θ_o as:

$$\varphi_o(t) = \begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n\ ) \\ -z_o(t-1) \\ \vdots \\ -z_o(t-n\ ) \\ u(t-Td-1) \\ \vdots \\ u(t-Td-n\ ) \\ -v_o(t-1) \\ \vdots \\ -v_o(t-n\ ) \\ e(t-1) \\ \vdots \\ e(t-n\ ) \end{bmatrix} \begin{matrix} \left.\right\} \varphi_y \\ \\ \left.\right\} \varphi_u \\ \\ \left.\right\} \varphi_e \end{matrix} \qquad \theta_o(t) = \begin{bmatrix} a_1 \\ \vdots \\ a_{n_a} \\ f_1 \\ \vdots \\ f_{n_f} \\ b_1 \\ \vdots \\ b_{n_b} \\ d_1 \\ \vdots \\ d_{n_d} \\ c_1 \\ \vdots \\ c_{n_c} \end{bmatrix} \begin{matrix} \left.\right\} \theta_y \\ \\ \left.\right\} \theta_u \\ \\ \left.\right\} \theta_e \end{matrix} \qquad [2.1.5]/[2.1.4]$$

Thus the true system [2.1.1] can be written:

$$y(t) = \theta^T \cdot \varphi_o(t) + e(t) \qquad [2.1.6]$$

In a recursive identification situation where the θ_o vector is unknown the definition of $\varphi(t)$ poses certain problems because it depends on the knowledge of θ_o and the sequence $e(t)$. To solve the problem of defining $\varphi(t)$, θ_o can be replaced by the current estimate $\hat{\theta}$ and $e(t)$ by the residuals $\varepsilon(t)$, i.e. old estimates are assumed as given and then $\varphi(t+1)$ defined recursively from $\varphi(t)$ as ;

$$\varepsilon(t) = y(t) - \hat{\theta}^T \cdot \varphi(t) \qquad [2.1.7]$$

$$(1+\hat{F}(q^{-1},t)) \cdot z(t) = \hat{B}(q^{-1},t) \cdot u(t-Td) \qquad [2.1.8]$$

$$(1+\hat{D}(q^{-1},t)) \cdot v(t) = (1+\hat{C}(q^{-1},t)) \cdot \varepsilon(t) \qquad [2.1.9]$$

$$\varphi(t+1) = \begin{bmatrix} -y(t) \\ \vdots \\ -y(t-n_a+1) \\ -z(t) \\ \vdots \\ -z(t-n_f+1) \\ u(t-Td) \\ \vdots \\ u(t-Td-n_b+1) \\ -v(t) \\ \vdots \\ -v(t-n_d+1) \\ \varepsilon(t) \\ \vdots \\ \varepsilon(t-n_c+1) \end{bmatrix} \qquad [2.1.10]$$

Now the model can be written as:

$$y(t+1) = \hat{\theta}^T \cdot \varphi(t+1) + \varepsilon(t+1) \qquad [2.1.11]$$

2.2 Algorithms

Most recursive or on-line identification methods are derived by approximating some off-line method with a recursive scheme (see e.g.[9]) but it turns out that they can be described by essentially the same algorithm (cf. [7]) where some of the quantities involved are defined in a slightly different way depending on the method used. This algorithm can be derived in a similar manner using the general model structure described in the preceding section. The algorithm is the following:

Algorithm

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\cdot\varphi(t+1)\cdot\varepsilon(t+1) \qquad [2.2.1]$$

$$P(t+1) = \left[P(t) - \frac{P(t)\cdot\varphi(t+1)\cdot\varphi^T(t+1)\cdot P(t)}{\lambda(t+1)+\varphi^T(t+1)\cdot P(t)\cdot\varphi(t+1)}\right]/\lambda(t+1) \qquad [2.2.2]$$

$$\lambda(t+1) = \lambda_k\cdot\lambda(t) + (1-\lambda_k) \qquad [2.2.3]$$

$$\varepsilon(t+1) = y(t+1) - \hat{\theta}^T(t)\cdot\varphi(t+1) \qquad [2.2.4]$$

where $\theta(t)$ is the parameter vector, $P(t)$ the covariance matrix, $\lambda(t)$ the forgetting profile and $\varphi(t)$ a vector of old signal values. For exact definitions of θ and φ cf. sections 2.1 and 2.3-2.4.

In the notation of sec. 2.1:
Let $F = D = C = 0$, with θ, φ and ε defined accordingly, then the algorithm corresponds to a recursive minimization of the loss function $\Sigma\ \varepsilon^2$ viz. the LS-method.

The algorithm above can be simlified to yield a Stocastic Approximation version given below. In this algorithm the updating of the P-matrix is replaced by the updating of a scalar p i.e. the equations [2.2.1] and [2.2.2] are replaced by eq's. [2.2.5] and [2.2.6].

The SA-version

$$\hat{\theta}(t+1) = \hat{\theta}(t) + p(t)\cdot\varphi(t+1)\cdot\varepsilon(t+1) \qquad [2.2.5]$$

$$\frac{1}{p(t+1)} = \lambda(t+1)\cdot\frac{1}{p(t)} + \varphi(t+1)^T\cdot\varphi(t+1) \qquad [2.2.6]$$

$$\lambda(t+1) = \lambda_k\cdot\lambda(t) + (1-\lambda_k) \qquad [2.2.3]$$

$$\varepsilon(t+1) = y(t+1) - \hat{\theta}^T(t)\cdot\varphi(t+1) \qquad [2.2.4]$$

Remark 2.2.1 - The Forgetting Profile.
The choice of $\lambda_k = 1$ results in $\lambda(t) = \lambda(0)$. A choice of $\lambda(0) < 1$ is appropriate in applications with time varying parameters. By choosing $\lambda_k < 1$ and $\lambda(0) < 1$, $\lambda(t)$ will start in $\lambda(0)$ and tend to 1. This may improve the initial convergence of the algorithm. See parameters WTO: λ_k and WTI: $\lambda(0)$ in sec. 5.3.4c.

Remark 2.2.2 - A posteriori estimate of the residual.
The à posteriori estimate of the residual ε i.e. $\varepsilon(t+1)=y(t+1)-\hat{\theta}^T(t+1)\cdot\varphi(t+1)$ can be computed and used when updating the φ-vector. The à posteriori estimate usually gives a slightly better result. See parameter IPRER in sec.5.3.4d.

Remark 2.2.3 - Limitation of the residuals.
The residuals may be limited to a given value, i.e. If $|\varepsilon(t)| > \text{RLIM}$ then $\varepsilon(t)$ is given the value $\varepsilon(t)=\text{sign}(\varepsilon(t))\cdot\text{RLIM}$. See parameter RLIM in sec. 5.3.4d

Remark 2.2.4 - Modification in the updating of P(t)
The computation of the diagonal elements of the covariance matrix P(t) can be modified by adding the term:

$$[R_{ii} - \delta\cdot P_{ii}(t)]/\lambda(t+1) \qquad [2.2.7]$$

where R is the i-th element of a vector and δ is a constant, both chosen by the user. The modification with R can be derived from the relation between least squares estimation and a Kalman filter. It can be useful in situations with time varying parameters to make sure that the P-matrix does not tend to zero. The second term has been suggested by L. Ljung [4] to prevent from difficulties that may occur because of almost singular P-matrices. The corresponding modification for the SA-version is:

$$\frac{1}{p(t+1)} = \lambda(t+1)\cdot\frac{1}{p(t)} + \varphi^T(t+1)\cdot\varphi(t+1) + \delta \qquad [2.2.6a]$$

See parameters DELTA:δ and the parameter vector R1 in sec.5.3.4c

<u>Remark 2.2.5 - Initial values of the P-matrix</u>
The initial value for P is P(0)=100·I in the program but can be set to any desired value by the parameter-vector P0. See sec. 5.3.4e. The initial value for p(0) in algorithm 2 is p(0)=100, which can be changed by the parameter P01.

2.3 The LIP-Approach

The LIP-Approach to estimate θ is to use the algorithm [2.2.1]-[2.2.4] as it stands (or the SA-version) with θ defined as in eq. [2.1.5] and φ as in eq. [2.1.10]. The loss function in this case can be taken as:

$$V(\theta) = E\,|y(t)-\theta^T\cdot\varphi(t)|^2 = E\,\varepsilon^2(t) \qquad [2.3.1]$$

which can be minimized by solving:

$$\frac{d}{d\theta} V(\theta) = E\,\varphi(t)\cdot(y(t)-\hat{\theta}^T\cdot\varphi(t)) = E\,\varphi(t)\cdot\varepsilon(t) = 0 \qquad [2.3.2]$$

This corresponds to an application of the model [2.1.11] which appears to be Linear-In-the-Parameters. Hence, the name. This is, however, not true. In the definition of $\varphi(t)$ there are hidden dependencies on $\hat{\theta}$.

Notice that this approach is for:

F = D = 0 known as the ELS-method
A = C = D = 0 known as the parallel model reference method
F = C = D = 0 known as the LS-method

The convergence of this method can be analysed by the use of an associated differential equation as explained in [2,3,7]. Assume that $\theta(t)$ tends to some value θ and consider the sequence of vectors $\bar{\varphi}(t,\theta)$ defined by equations [2.1.7]-[2.1.10] as $t \to \infty$ with the constant model θ. Assume that it has reached stationarity. Introduce in the same manner the stationary process $\bar{\varepsilon}(t,\theta) = y(t) - \theta^T\bar{\varphi}(t,\theta)$ and let:

$$f(\theta) = E\,\bar{\varphi}(t,\theta)\cdot\bar{\varepsilon}(t,\theta) \qquad [2.3.3]$$

$$G(\theta) = E\,\bar{\varphi}(t,\theta)\cdot\bar{\varphi}^T(t,\theta) \qquad [2.3.4]$$

with the expectation over $(e(t))$. Then the differential equation becomes:

$$\dot{\theta} = R^{-1}\cdot f(\theta) \qquad [2.3.5a]$$

$$\dot{R} = G(\theta) - R \qquad [2.3.5b]$$

where e.g. (cf.[7])

$$R = \frac{\gamma(t)}{P(t)} \quad \text{and} \quad \gamma(t) = \frac{\gamma(t-1)}{\lambda(t)+\gamma(t-1)} \quad ; \quad \gamma(0) = 1$$

According to [2], then global stability of [2.3.5] implies convergence of the algorithm and also that possible convergence points of the algorithm correspond to stable stationary points of the differential equation.

Provided that the model tried for the identification is the same as that of the system then $\bar{\varepsilon}(t,\theta)$ can be expressed as:

$$\bar{\varepsilon}(t,\theta) = \begin{bmatrix} \frac{1+D_o}{1+C_o}\cdot\bar{\varphi}_y(t,\theta) \\ \frac{1+D_o}{(1+C_o)(1+F_o)}\cdot\bar{\varphi}_u(t,\theta) \\ \frac{1}{1+C_o}\cdot\bar{\varphi}_\varepsilon(t,\theta) \end{bmatrix}^T \cdot (\theta_o - \theta) + e(t) \qquad [2.3.6]$$

and by using the results of [3] the convergence will be tied to the positive realness of the transfer functions indicated in equation [2.3.6]. It is, however, not clear what can be said of the convergence properties when the system cannot be described by the model structure chosen for the identification.

2.4 The Criterion Minimization Approach

To account for the hidden dependencies on θ in the definition of $\varphi(t)$ the proper criterion to consider is:

$$V(\theta) = E\,|y(t)-\theta^T\cdot\varphi(t,\theta)|^2 = E\,\varepsilon(t,\theta)^2 \qquad [2.4.1]$$

The residual can be expressed as:

$$\varepsilon(t,\theta) = \frac{(1+D)\cdot(1+A)}{(1+C)}\cdot y(t) - \frac{B\cdot(1+D)}{(1+C)\cdot(1+F)}\cdot u(t-Td) \qquad [2.4.2]$$

and direct calculation verifies that:

$$-\frac{d\varepsilon(t,\theta)}{d\theta} = \begin{bmatrix} \frac{1+D}{1+C}\cdot\varphi_y(t,\theta) \\ \frac{1+D}{(1+C)\cdot(1+F)}\cdot\varphi_u(t,\theta) \\ \frac{1}{1+C}\cdot\varphi_\varepsilon(t,\theta) \end{bmatrix} = \begin{bmatrix} \psi_y(t,\theta) \\ \psi_u(t,\theta) \\ \psi_\varepsilon(t,\theta) \end{bmatrix} = \psi(t,\theta) \qquad [2.4.3]$$

Therefore the equation to be solved analogously to equation [2.3.1] is:

$$-\frac{d}{d\theta} V(\theta) = E\ \psi(t,\theta)\cdot\varepsilon(t,\theta) = 0 \qquad [2.4.4]$$

Similarly to the definition of $\varphi(t)$ (cf. equations [2.1.7]-[2.1.10]) the vector $\psi(t)$ is defined recursively in t as follows:

$$(1+\hat{C}(q^{-1},t))\cdot\psi_y(t) = (1+\hat{D}(q^{-1},t))\cdot\varphi_y(t) \qquad [2.4.5]$$

$$(1+\hat{C}(q^{-1},t))\cdot(1+\hat{F}(q^{-1},t))\cdot\psi_u(t) = (1+\hat{D}(q^{-1},t))\cdot\varphi_u(t) \qquad [2.4.6]$$

$$(1+\hat{C}(q^{-1},t))\cdot\psi_\varepsilon(t) = \varphi_\varepsilon(t) \qquad [2.4.7]$$

The Criterion Minimization Approach then is to use the algorithm [2.2.1]-[2.2.4] or the SA-version where $\varphi(t)$ is replaced by $\psi(t)$ in equations [2.2.1] and [2.2.2] or [2.2.5] and [2.2.6] but of course not in [2.2.4].

Notice that:

F = D = 0	gives the recursive ML-method
F = C = 0	gives the generalized least squares method
A = C = D = 0	gives the Extended Kalman Filter with no process noise

The differential equation associated with this approach is the same as in the preceding section except that now:

$$f(\theta) = E\ \bar{\psi}(t,\theta)\cdot\bar{\varepsilon}(t,\theta) \qquad [2.4.8]$$

$$G(\theta) = E\ \bar{\psi}(t,\theta)\cdot\bar{\psi}(t,\theta)^T \qquad [2.4.9]$$

i.e.

$$\dot{\theta} = R^{-1}\cdot E\ \bar{\psi}(t,\theta)\cdot\bar{\varepsilon}(t,w) = R^{-1}\cdot V(\theta) \qquad [2.4.10]$$

$$\dot{R} = E\ \bar{\psi}(t,\theta)\cdot\bar{\psi}^T(t,\theta) - R \qquad [2.4.11]$$

Since R is a positive definite matrix and $V(\theta)$ is positive it follows from [2] that now $\theta(t)$ converges to a local minimum of $V(\theta)$, even if the true system cannot be described within the chosen model structure. The uniqueness of the local minima of $V(\theta)$ will in general depend on the actual structure of the model.

Remark 2.4.1 - Stability check
It is necessary that the filtering according to equations [2.4.5]-[2.4.7] is exponentially stable for this approach to work. Therefore the stability of the F and C polynomials can be checked and if unstable projected in such a way that the polynomials have all roots inside the unit circle. See Parameter ISTAB in sec. 5.3.4d.

Remark 2.4.2 - A better initial convergence.
To obtain a better initial convergence in the start-up of this approach the filtering according to equations [2.4.5]-[2.4.7] can be omitted for IFIST number of steps. See parameter IFIST in sec 5.3.4d.

2.5 The Instrumental Variables Algorithm

The IV-algorithm implemented in the program package is basically the same as algorithm [2.2.1]-[2.2.4] in sec. 2.2. For a more extensive treatment cf. [7,12,13]. It corresponds to a model structure (cf. sec 2.1) where F = C = D = 0 that is:

$$(1+A(q^{-1})\cdot y(t) = B(q^{-1})\cdot u(t-Td) + e(t) \qquad [2.5.1]$$

Algorithm IV

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\cdot z(t+1)\cdot \varepsilon(t+1) \qquad [2.5.2]$$

$$P(t+1) = \left[P(t) - \frac{P(t)\cdot z(t+1)\cdot \varphi^T(t+1)\cdot P(t)}{\lambda(t+1)+\varphi^T(t+1)\cdot P(t)\cdot z(t+1)}\right]/\lambda(t+1) \qquad [2.5.3]$$

$$\lambda(t+1) = \lambda_k\cdot\lambda(t) + (1-\lambda_k) \qquad [2.5.4]$$

$$\varepsilon(t+1) = y(t+1) - \hat{\theta}^T(t)\cdot\varphi(t+1) \qquad [2.5.5]$$

where the vector z consists of the instrumental variables and other quantities defined as before (cf.sec. 2.1-2.2).

Notice also that Remarks 2.2.1 thru 2.2.5 apply here too.

Remark 2.5.1 - Different Instrumental Variables
The user can choose between using the old values of the input signal as in eq. [2.5.6] or defining them according to equations [2.5.7]-[2.5.8].

$$z(t) = (u(t-1),u(t-2)...........u(t-n_a-n_b))^T \qquad [2.5.6]$$

$$z(t) = (-x(t-1),....-x(t-n_a),u(t-1)....u(t-n_b))^T \qquad [2.5.7]$$

where

$$x(t) = z(t)^T \cdot \theta(t) \qquad [2.5.8]$$

See Parameter INVAU in sec. 5.3.4d.

Remark 2.5.2 - A better initial convergence
The algorithm can be started up by using the least squares estimate for IFIST number of steps, to obtain a better initial convergence. See Parameter IFIST in sec. 5.3.4d.

2.6 An Extended Kalman Filter Algorithm

The EKF is a well known approach to the estimation of parameters. In an extensive treatise by L. Ljung [4] the properties of the common EKF are analysed and it is shown there that with a modification of the algorithm, global convergence can be obtained for a general case.

Consider a single-input single-output model described by:

$$X(t+1) = A(\theta(t))\cdot X(t)+B(\theta(t))\cdot u(t-Td)+K(\theta(t))\cdot \varepsilon(t) \qquad [2.6.1]$$

$$y(t) = C(\theta(t))\cdot X(t) + \varepsilon(t) \qquad [2.6.2]$$

where X is the state vector and A, B, C and K are matrices and/or vectors cf. below. The modified algorithm implemented in the program package is based on the model above with the particular parametrization:

$$A(\theta(t)) = F + (V+K)\cdot\theta^{T}(t)\cdot H \qquad [2.6.3]$$

$$B(\theta(t)) = B \qquad [2.6.4]$$

$$K(\theta(t)) = K \qquad [2.6.5]$$

$$C(\theta(t)) = \theta^{T}(t)\cdot H \qquad [2.6.6]$$

where F, V, K, H and B are fixed and chosen as:

$$X(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-n_a) \\ u(t-1) \\ \vdots \\ u(t-n_b) \\ \varepsilon(t-1) \\ \vdots \\ \varepsilon(t-n_c) \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow n_a+1 \qquad K = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow n_a+n_b+1 \rightarrow \quad V = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$F = \begin{bmatrix} \begin{matrix} 00\ldots 0 \\ 10\ldots 0 \\ 010\ldots 0 \\ \vdots \quad \vdots \\ 0\ldots 010 \end{matrix} & 0 & 0 \\ 0 & \begin{matrix} 00\ldots 0 \\ 10\ldots 0 \\ 010\ldots 0 \\ \vdots \quad \vdots \\ 0\ldots 010 \end{matrix} & 0 \\ 0 & 0 & \begin{matrix} 00\ldots 0 \\ 10\ldots 0 \\ 010\ldots 0 \\ \vdots \quad \vdots \\ 0\ldots 010 \end{matrix} \end{bmatrix} \qquad \theta = \begin{bmatrix} -a_1 \\ \vdots \\ -a_{n_a} \\ b_1 \\ \vdots \\ b_{n_b} \\ c_1 \\ \vdots \\ c_{n_c} \end{bmatrix} \qquad H = I$$

This corresponds to a model structure (cf. sec. 2.1) of the form:

$$(1+A(q^{-1}))\cdot y(t) = B(q^{-1})\cdot u(t-Td) + (1+C(q^{-1}))\cdot\varepsilon(t)$$

The algorithm is the same as equations (8.18) in [4] and has the following form:

$$X(t+1) = (F+(V+K)\cdot\hat{\theta}^T(t))\cdot X(t) + B\cdot u(t-Td) + K\cdot\varepsilon(t) \quad [2.6.7]$$

$$\varepsilon(t) = y(t) - \hat{\theta}^T(t)\cdot X(t) \quad [2.6.8]$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + L(t)\cdot\varepsilon(t) \quad [2.6.9]$$

$$L(t) = \left[P_2(t)^T\cdot\hat{\theta}(t) + P_3(t)\cdot X(t) \right]/S_t \quad [2.6.10]$$

$$S_t = X(t)^T\cdot P_3(t)\cdot X(t) + 2\cdot\left[X(t)^T\cdot P_2(t)^T\cdot\hat{\theta}(t)\right] + 1 \quad [2.6.11]$$

$$P_2(t+1) = F\cdot P_2(t) + V\cdot\left[\hat{\theta}(t)^T\cdot P_2(t) + X(t)^T\cdot P_3(T)\right] \quad [2.6.12]$$

$$P_3(t+1) = P_3(t) - L(t)\cdot S_t\cdot L(t)^T - \delta\cdot P_3(t)\cdot P_3(t) \quad [2.6.13]$$

where δ is some small number, an adjustment to improve the stability and the convergence rate of the algorithm. See Parameter DELTA in sec. 5.3.4c.

Remark 2.6.1 - Stability check
Theorem 6.1 in [4] requires that θ is kept in a compact subset of the set:

$$D_s = \left[\theta \mid \text{the matrix } A(\theta) - K(\theta)\cdot C(\theta) \text{ is exp. stable}\right]$$

which, with the parametrization used here equals the requirement that the C parameters in θ have their roots inside the unit circle. It is therefore recommended that the user sets the parameter ISTAB to 1. See sec. 5.3.4d.

2.7 Accuracy

As already stated the aim is to use the general model structure (cf.sec. 2.1) to arrive at in some sense the "best" approximate description of a fairly complex system. In order to compare the results of identification experiments, the following measures of accuracy can be computed by the program. These are:

$$VE = V_1 = \frac{1}{n}\cdot\Sigma\ |\varepsilon(t,\theta(t))|^2 \qquad t = 1,..n \qquad [2.7.1]$$

$$V_2 = \frac{1}{n}\cdot\Sigma \left(\frac{\varepsilon(t,\theta(t))}{\lambda(t) + \varphi(t+1)^T\cdot P(t)\cdot\varphi(t+1)} \right)^2 \qquad [2.7.2]$$

The loss function V_1 expresses the variance of the one step prediction errors that will be obtained using the actual models $\theta(t)$. The same applies for the loss function V_2 except that it is based on the weighted one step prediction errors.

NB! Note that $V = n\cdot V_1$ and $WV = n\cdot V_2$ are declared as variables in the SIMNON sense and can therefore be viewed by using the command PLOT (or SHOW).

For the computation of the loss functions given below it is assumed that the first 100 pulse response coefficients (deterministic (h_d) resp. noise (h_n) parts) of the true system are known. The input-output sequence used may in that case have been generated as:

$$y(t) = \Sigma\ h_d(s)\cdot u(t-s) + \Sigma\ h_n(s)\cdot e(t-s)$$

or

$$y(t) = \frac{B}{(1+F)(1+A)}\cdot u(t) + \frac{1+C}{(1+D)(1+A)}\cdot e(t)$$

The loss functions:

$$VRS = V_3 = \frac{V_4}{\Sigma\ |h_d|^2} \qquad V_4 = \Sigma\ |h_d - \hat{h}_d|^2$$

refer to the deterministic part of the system or the A, B and F polynomials and the accuracy is measured as the difference in pulse responses between the true (h_d) and the identified ($\hat{h}_d$) system. Note that VRS equals V_4 normalized by $\Sigma\ h_d^2$

The same applies for the loss functions:

$$VRN = V_5 = \frac{V_6}{\Sigma \; |h_n|^2} \qquad\qquad V_6 = \Sigma \; |h_n - \hat{h}_n|^2$$

except that they refer to the noise part (the C,D and A polynomials).

<u>Remark 2.7.1 - Printout of loss functions</u> The loss functions are computed and written on the line printer at the end of every simulation if the parameter IHAT > 0, see sec. 5.3.4g.

<u>Remark 2.7.2 - Plot pulse responses</u>
The pulse responses $\hat{h}_d$ and $\hat{h}_n$ of the identified system can be placed on a file together with the true pulse responses h_d and h_n and plotted by the SHOW command. See parameter IFIGE in sec. 5.3.4g and the global variable name FNIM sec. 5.3.1.

<u>Remark 2.7.3 - When making several runs with the same structure</u>
When several runs are made (with the same model structure and method) in order to identify a given system, then the results can be placed on a file (FNST sec. 5.3.1), if the parameter IFIGE equals 2 or 3 see sec. 5.3.4g. The auxiliary programs STAT and VSTAT (subsystem AUXIL) can then be used to print and compute means, standard deviation , etc. cf. sec. 5.4.3 and 5.4.5.

<u>Remark 2.7.4 - Series of identification experiments.</u>
When a series of identification experiments have been made, it is possible to obtain a print-out of the loss functions VE, VRS and VRN in a tabular form to compare different structures or methods (prog. TAFLA sec. 5.4.6).

3 BASIC PROGRAM STRUCTURE

The program package consists of four subsystems as the reader already has seen in fig. 1. This section contains a short presentation of these and some supplementary notes.

The first subsystem, INPUT, can perform one of three alternatives at a time, namely:

1 Generate a sequence of independent random variables
2 Generate a pseudo random binary sequence
3 Read data from an IDPAC compatible data file cf. [16,17]

The last alternative can for instance be of use when performing recursive identification on data from a real process.

The second, SYSHG, simulates a single-input single-output linear discrete time system on a difference equation form (cf.sec. 2.1 and 5.2). This should suffice for most types of simulation but the user can of course implement his own, using the SIMNON simulation language, in case SYSHG is not adequate. By setting a parameter the subsystem can compute and create files with the pulse responses of the deterministic and the noise parts of the system. The pulse responses can then be used when evaluating the loss functions given in sec.2.7.

The third, IDHG, performs the identification and includes all the algorithms described in section 2 as well as the evaluation of the loss functions given in sec. 2.7. When IDHG is used to do a series of identification experiments on a system with known pulse responses but with different input/noise characteristics each time, the results can be placed on a file. By using the subsystem AUXIL (below) statistics of the results can then be computed.

The subsystem AUXIL is not a proper simulation subsytem but has been implemented to facilitate the inclusion of some programs which may come in handy from time to time in connection with identification experiments. The function of the programs is to:

1 Generate the pulse response of a transfer function, given poles and zeroes.
2 Generate an output sequence of a system, given the input sequence, the deterministic and the noise pulse responses and the signal to noise ratio.
3 Compute and print statistics on identification runs.
4 Convert a SIMNON data-file to an IDPAC compatible file.

When using the program package the systems must be connected together with a special CONNECTING SYSTEM as explained in the SIMNON manual. A list of useful CONNECTING SYSTEMS as well as a quick reference program guide is given in Appendix A.

The simulation subsystems are all implemented as discrete time systems written in FORTRAN. The user can easily implement his own subsystems either in the simulation language (e.g. continuous and/or nonlinear) as described in the SIMNON manual, or as a FORTRAN subsystem. In the latter case the programs are implemented into the program package by use of the utility programs CHAIN and ABS for the computer (PDP-15). This way the user creates his own version of the program package.

4 PROGRAM START-UP

The program package is available on DEC-tape 37.9 for the PDP-15 computer. The program is started up by the following sequence of commands:

```
$PIP

>T RK_DT1 SYSTEM XXX (D)
>T RK_DT1 SYSTEM XCT (B)
>T RK_RK <EXT> SIMNON XXX (D)
>T RK_RK <EXT> SIMNON XCT (B)


$A RK 3,5,7/RK <EXT> 4/NON 15,16
$BUFFS 6
$E SIMNON
```

<u>Remark 4.1.1 - Integration of continuous systems.</u>
Because of the program size the integration routine HAMPC, normally used in SIMNON, is not implemented. This means that whenever a continuous system is included in a SYST-command the command

```
>ALGOR RK
```

must be given before the simulation is started. It may also be necessary to decrease the error bound for the integration routine or choose the maximal time increment in the SIMU command appropriately cf. [15].

<u>Remark 4.1.2 - Macros</u>
The program package uses a special version of SIMNON, designed to simplify the inclusion of FORTRAN subsystems and provide more core memory for user programs. However, as a drawback, Macros that include the commands SIMU and SYST cannot be included.

5 DESCRIPTION OF SUBSYSTEMS

In this section a detailed description is given of the subsystems available, i.e. INPUT, SYSHG, IDHG and AUXIL, which together with the simulation examples in the next section should give a fairly good demonstration of how the program package can be used. Note that efficient use of the program package requires that the user is familiar with the description of the model structure and the algorithms, given in section 2.

There are eight global variables which are common to all systems, more or less, but as a rule need not be given any values since they are given default values by the program. The first five are concerned with print-out and the last three with .dat slot usage.

Global variable	Default value
IYEAR	0
MONTH	0
IDAY	0
ISIMN	0
IDOC	0
LUN1	5
LUN2	7
LUN3	3

If IDOC is set to one a documentation page will be written at the end of every simulation with the simulation number ISIMN, the given date and the values of all SIMNON declared parameters. IDOC=2 will inhibit printout of loss functions at the end of simulations. This is appropriate when doing several identification runs where the results are written onto files. Each time the command SIMU is given (without -CONT) one is added to ISIMN. If the ASSIGN command

$A RK 3,5,7/RK <EXT> 4/NON 15,16

given at the start-up is changed it is necessary to give the appropriate values to LUN1, LUN2 and LUN3 or a terminal error will result. In the examples given the computer printout on the terminal will be underlined.

5.1 SUBSYSTEM INPUT

The subsystem INPUT can perform one of three alternatives at a time depending on the value of the parameter NTYP.

NTYP: 1 Generate a sequence of independent random variables
: 2 Generate a Pseudo Random Binary Sequence
: 3 Read one or two values from a data-file at each sampling event.

5.1.1 GLOBAL VARIABLE

FILN This global variable defines the name of the data file from which data is to be read and must be set before the command SIMU is given if the subsystem is to be used to read data files.

NB! When the system INPUT is used to read data files the command SIMU -CONT cannot be used.

Example 5.1.1:
The command LET FILN.=FDATA (Max 5 characters) should be given when the name of the data file to be read is FDATA.BIN.

5.1.2 OUTPUTS

E Par NTYP = 1 gives E as a sequence of independent random N(O,σ) variables

PRBS Par NTYP=2 gives PRBS as a Pseudo Random Binary Sequence

U1
U2 Par NTYP=3 gives U1 as the value read from column no. NC1 in the data file FILN and if NC2>0, U2 as the value read from column no. NC2

NB! The Outputs are zero except for the right value of NTYP e.g. The output PRBS is zero for NTYP=3

5.1.3 PARAMETERS

a) To decide the type of active output
NTYP See above Default value :0

b) Parameters that define the output E
IODD Starting value for the random number generator. An odd number must be chosen. Default value: 95

SD:σ The standard deviation of the random variables. Default value: 1.

c) *Parameters defining the output PRBS*

IBP Basic period Default value: 1

NBIT Number of bits in the shift register. Range (3,17) Default value : 7

AMP The Amplitude of the PRBS output-signal. Default value: 1.

KNEP : 1 FOA-Knep
: 2 No Knep (Default)

ISTAR Specifies a starting point in the sequence 1, 2, 3, 4 Default value: 1

d) *Parameters defining the outputs U1 and U2*

NC1 The column number in the data file from where the output U1 is to be read. Default value: 1

NC2 The column number in the data file from where the output U2 is to be read. Default value: 0

NB! The output U2 is zero for NC2=0

e) *The sampling period*

DT Default value: 1.

Example 5.1.2:
To generate a sequence of random variables N(0,2), of length 1000 the following parameters are set:

```
>PAR NTYP:1
>,SD:2
>SIMU 0 1000
```

Notice that all other parameters except NODD are redundant. To get another sequence than the one given by the default value for NODD, then NODD is set to some odd value e.g.

```
>PAR NODD:29
>SIMU
```

Example 5.1.3:
To generate a Pseudo Random Binary sequence, length 2110, which is to be stored on file for later use in an identification experiment the following parameters are set:

```
>PAR NTYP:2
>PAR IBP:4
>PAR NBIT:10
>STORE PRBS
>SIMU 0 2110/PRBSU
```

In this case IBP is set to 4, which means that the shift register with feedback used to generate the sequence is clocked each 4:th sampling interval to obtain a more slowly changing signal than the signal obtained by the default value for IBP. NBIT which specifies the length of the shift register is set to 10. That means that the output sequence repeats itself after $2^{10}-1$ clock instants. Here after $(2^{10}-1)\cdot 4$ sampling events. To get another starting point in the periodic sequence the parameter ISTART can be set to 2,3 or 4. When KNEP:1 then the output is complemented by a flip-flop if the original sequence was negative. This gives a sequence with a asymptotically zero mean value, which the standard PRBS-signal has not. The global variable IDOC was set to 1 which gave the documentation below:

```
DOCUMENTATION                                          SIMU. NO:            2
*************                                          DATE     : 1980- 6-11


SYSTEM INPUT:
-------------
NTYP =         2
                         NBIT  =        10             NC1   =         1
IODD =        95         KNEP  =         0             NC2   =         0
SD   =   1.00000         ISTAR=          1             FILN  =
                         IBP   =         4
DT   =   1.00000         AMP   =   1.00000
```

Example 5.1.4
To read columns 1 and 4 of a data file named FDATA.BIN the following commands are given:

```
>LET FILN.=FDATA
>PAR NTYP:3
>,NC2:4
>SIMU
```

Notice that the default value for NC1 is used. U1 now get the value read from column no. 1 and U2 the value from column no. 4.

5.2 SUBSYSTEM SYSHG

The subsystem SYSHG simulates a single-input, single-output difference equation of a general structure (cf. sec. 2.1 as well)

$$(1+A(q^{-1}))\cdot y1(t) = \frac{B(q^{-1})}{1+F(q^{-1})}\cdot u(t-Td) + \frac{1+C(q^{-1})}{1+D(q^{-1})}\cdot e(t) \qquad [5.2.1]$$

$$y(t) = y1(t) + yc \qquad [5.2.2]$$

where A, F, B, D and C are polynomials in q^{-1} (the backward shift operator) of orders n_a, n_f, n_b, n_d and n_c respectively. The polynomials are such that A(0)=0, F(0)=0 etc. and e(t) is a sequence of independent random N(0,λ) variables. yc is a constant added to the output. As shown in sec. 2.1 the system can also be written as:

$$y(t) = \theta^T(t)\cdot\varphi(t) + e(t) + yc \qquad [5.2.3]$$

5.2.1 GLOBAL VARIABLES

FNRS
FNRN — These global variables define the names of the data files onto which the pulse responses of the deterministic (FNRS) and the noise part (FNRN) of the system are written. These variables must be set before the command SIMU is given if the parameter NIMR = 1. (cf. sec. 2.7)

5.2.2 INPUT

U:u(t-Td) — This is the input signal in the difference equation [5.2.1]

5.2.3 OUTPUT

Y:y(t) — This is the output signal from the difference equation [5.2.2] / [5.2.3].

5.2.4 PARAMETERS

a) To define the system structure

NAS: n_a	These are the orders of the A, F, B, D and the C polynomials respectively.
NFS: n_f	
NBS: n_b	
NDS: n_d	
NCS: n_c	Default values: 0
KDS: Td	The number of time delays Default value: 0

NB! The sum NPAR = NAS + NFS + NBS + NDS + NCS must not exceed 30, i.e. NPAR $\leq$ 30.

b) The Parameters of the system

YLEV: yc A constant added to the output of the system. Default value: 0.

THS1
THS2
.
.
.
THS30

The parameter vector θ (cf. eqs. [2.1.5]/[5.2.3])

Default values: 0

NB! When using the SIMNON commands SAVE and GET observe that SIMNON stores PAR and INIT values on source files in a manner that only the four most significant digits are stored. This may cause a stable system to become unstable if the THS parameters were given more accurate values.

c) Other parameters

NODD Starting value for the number generator, an odd number must be chosen. Default value: 19

LAMB: λ The standard deviation of the random variables Default value: 1

NIMR To decide whether pulse responses are to be computed or not.

NIMR=1	Computed
NIMR$\neq$1	Not computed (Default)

DT The sampling period. Default value: 1.

<u>Example 5.2.1:</u>
To simulate a system described by the following difference equation:

$$(1-q^{-1}+0.5q^{-1})\cdot y(t) = \frac{q^{-1}-0.9q^{-1}}{1-1.6q^{-1}+0.7q^{-1}}\cdot u(t-1) + (1-q^{-1}+0.2q^{-1})\cdot e(t)$$

requires the following parameters to be set:

```
>PAR NAS:2       "order of A-polynomial
>,NFS:2          "  -     - F      -
>,NBS:2          "  -     - B      -
>,NCS:2          "  -     - C      -
>,KD:1           "the time delay
>,THS1:-1.0      " a1
>,THS2:0.5       " a2
>,THS3:-1.6      " f1
>,THS4:0.7       " f2
>,THS5:1.0       " b1
>,THS6:-0.9      " b2
>,THS7:-1.0      " c1
>,THS8:0.2       " c2
>SIMU
```

5.3 SUBSYSTEM IDHG

The subsystem IDHG performs recursive identification using linear discrete time stocastic models. The user should refer to sec. 2 for information on the identification methods and algorithms. Implemented in the program are all the methods described there and an algorithm (KJADMH) implemented by I.Gustavsson not reviewed in this paper.

5.3.1 GLOBAL VARIABLES

FNRS
FNRN
: These global variables define the names of the data files containing the reference system (FNRS) pulse response and the reference noise (FNRN) pulse response to which the program must have access to in order to perform the accuracy computations described in sec. 2.7.

NB!
: These variables must be set before the command SIMU is given if the parameters IHAT or IFIGE are greater then zero. Cf. IHAT and IFIGE.

FNIM — This global variable defines the name of a file onto which the pulse responses of the reference and the identified system/noise are written. The pulse responses can be viewed by giving the command SHOW RS IS RN IN/<file name>, where RS,IS stand for reference resp. identified system or deterministic part and RN, IN for reference resp. identified noise part.

NB! This variable must be set before the command SIMU is given if the parameter IFIGE equals 1 or 3.

FNST — This global variable defines the name of a file onto which the parameter estimates and the accuracy results are written. If the named file is not present on disc when the command SIMU is given then a file with this name is created.
All necessary information concerning the identification run is written at the top of the file i.e. SIMNON parameters, the simulation time, the names of FNRS and FNRN. Then data for this particular run i.e. the simulation number, the given date followed by the parameter estimates and the values of the loss functions at the sampling events decided by the parameters IHAT and IHATN.
If the named file is present then the actual SIMNON parameters etc. are compared against the file head. When discrepancy occurs an error message is given if not then the results from the present run are added to the file.

NB! This variable must be set before the command SIMU is given if the parameter IFIGE equals 2 or 3.

5.3.2 INPUTS

Y — The output of the system to be identified.
U — The input to the system to be identified.

5.3.3 OUTPUTS

TH1 — The vector of parameter estimates.
TH2
.
.
.
TH10

5.3.4 PARAMETERS

a) METOD — <u>Identification Method</u>

- :1 LIP-Approach SA-Version
- :2 CMA-Approach SA-Version
- :3 LIP-Approach
- :4 CMA-Approach
- :5 The Instrumental Variables Method
- :6 The KJADMH-Algorithm
- :7 The Extended Kalman Filter Algorithm
- :0 Default

b) <u>Model Structure</u>

NA:n_a	The order of	the	A-Polynomial	cf. eq.	[2.1.1]		
NF:n_f	-	-	-	-	F	-	-
NB:n_b	-	-	-	-	B	-	-
ND:n_d	-	-	-	-	D	-	-
NC:n_c	-	-	-	-	C	-	-

Default values: 0

KD:Td The time delay Default value: 0

c) <u>Algorithm parameters</u>

WTI:$\lambda(0)$ The initial value for the forgetting profile Default value: 1

WTO:λ_k The forgetting profile parameter. Default value: 1

DELTA:δ An adjustment to improve the stability of algorithms. Default value: 0

R1:R1i A vector containing the diagonal elements of the parameter noise. Default values: 0

d) <u>Special feature parameters</u>

IPRER Parameter to decide whether an a priori or an a posteriori estimate of the residual is to be used.

- : 0 The a priori estimate (default)
- : 1 The a posteriori estimate

RLIM parameter to decide whether the residuals are to be limited or not

- < 0 No limitation (default)
- : R_1 If $|\varepsilon|>R_1$ then $\varepsilon = \text{sign}(\varepsilon)\cdot R_1$

IFIST a) if METOD: 2 or 4 Default value: 0

The number of steps in the start-up of the CMA-Approach for which the filtering according to equations [2.4.5]-[2.4.7] is omitted.

b) if METOD: 5
If IFIST > 0 then the algorithm is started up by Using the least squares estimates for IFIST number of steps.

ISTAB — Parameter to decide whether a stability check is made of the F and C polynomials or not.cf.[2.1.1]

:0 No stability check (default)

:1 The stability of the F and C polynomials is checked. If unstable, the parameter vector is adjusted halfway towards the last value and stability checked again. This is repeated until the polynomial has all zeros inside the unit circle.

:2 The stability of the F and C polynomials is checked. If unstable, then the polynomial is factored to give a polynomial with zeros inside the unit circle by using a spectral factorization subroutine.

INVAU — Parameter to decide whether to use the old input signals as the instrumental variables or to define them as in equations [2.5.7]-[2.5.8]

≤0 Defined as in [2.5.7]-[2.5.8] (Default)

>0 Old input signals.

e) <u>Initial Values</u>

THO:$\theta(0)$ — Vector of initial parameter estimates
Default values: 0.0

PO:$P_{ii}(0)$ — Vector containing the initial values of the diagonal elements of the covariance matrix P
Default values: 100.0

f) <u>Print-out control parameters during simulation</u>

IWRT Default value: -5
IWRT1 Default value: 0
IWRTN Default value: 100

IWRT≤0 No print-out

=1 A print out of the sampling time and the parameter estimate (θ-vector) is obtained on the line printer for the IWRT1 first sampling events and then every IWRTN:th sampling event thereafter

=2 Same as for IWRT=1 plus the insignals U and Y, the residual ε and the weighted residual $w\varepsilon$. Cf. sec. 2.7

=3 Same as for IWRT=2 plus the covariance matrix P

=4 Same as for IWRT=3 plus the φ-vector, the ψ-vector (CMA), the z-vector (IV).

=5 Same as for IWRT=4 plus the S-matrix which contains old signal values as follows:

$$S = \begin{bmatrix} -y(t-1), & -y(t-2)\ldots\ldots\ldots \\ -z(t-1), & -z(t-2)\ldots\ldots\ldots \\ u(t-1), & u(t-2)\ldots\ldots\ldots \\ -v(t-1), & -v(t-2)\ldots\ldots\ldots \\ \varepsilon(t-1), & \varepsilon(t-2)\ldots\ldots\ldots \end{bmatrix}$$

IWRT=6 Same as for IWRT=5 plus the TEE-matrix which contains old filtered signal values as follows:

$$TEE = \begin{bmatrix} \psi_y(t-1), & \psi_y(t-2)\ldots\ldots\ldots \\ \psi_z(t-1), & \psi_z(t-2)\ldots\ldots\ldots \\ \psi_u(t-1), & \psi_u(t-2)\ldots\ldots\ldots \\ \psi_v(t-1), & \psi_v(t-2)\ldots\ldots\ldots \\ \psi_\varepsilon(t-1), & \psi_\varepsilon(t-2)\ldots\ldots\ldots \end{bmatrix}$$

g) <u>Accuracy</u>

IHAT, IHATN: Parameters that control whether the loss functions in section 2.7 are computed or not.

IHAT<0 Not computed (Default)

IHAT≥0 The loss functions are computed for ISAMP=IHAT and every IHATN:th sampling event. The results are printed on the line printer at the end of the simulation, except when IHAT=0 or when the global variable IDOC=2

IFIGE Controls which files are to be generated by IDHG.

=0 No files generated (default)

=1 A file with the name assigned to the global variable FNIM is generated containing the reference and the identified pulse responses.

Column 1: The reference system pulse response
- 2: - identified - - -
- 3: - reference noise - -
- 4: - identified - - -

The command SHOW RS IS RN IN/< file name > plots the pulse responses.

=2 A file with the name assigned to the global variable FNST is generated (or written onto if present on disc) containing the sampling times when accuracy computations were made (cf.sec.2.7) the results of these as well as the actual parameter estimates. The programs in the

Subsystem AUXIL can then be used to compute means and standard deviations for any number of simulations.

=3 Same as for IFIGE=1 and IFIGE=2

NB! The global variables FNRS and FNRN must be set before the command SIMU is given if IHAT≥0 or or IFIGE>0, the global varible FNIM if IFIGE eq. 1 or 3 and the global variable FNST if IFIGE eq. 2 or 3.

h) Sampling period/time

DT	The sampling period	(default value: 1.0)
TS	The sampling time	

5.3.5 VARIABLES

RES: ε	The residual from the estimation algorithms.
WRES: $w\varepsilon$	The weighted residual $w\varepsilon=\varepsilon/(\lambda+\varphi^T \cdot P \cdot \varphi)$
V	The loss function $V = \Sigma\ \varepsilon^2$
WV	The loss function $WV = \Sigma\ w\varepsilon^2$

Example 5.3.1:

The file SDATA.BIN contains an input sequence to a system-process in column 1 and 10 different output sequences in columns 2 - 11 (cf. sec. 5.4.1-5.4.2 and 6.1 on how SDATA was generated). This systems pulse responses are available on files SIMPU.BIN (deterministic) and NIMPU.BIN (noise). The model structure to be used for the identification is:

$$y(t) = \frac{B}{1+F} \cdot u(t) + \frac{1+C}{1+D} \cdot \varepsilon(t)$$

where n =n =n =n =2 and the identification method the CMA-approach (cf. sec. 2.4) where the filtering according to equations [2.4.5]-[2.4.7] is to be omitted for the first 75 steps. The loss functions are to be evaluated for ISAMP = 100, 500, 1000, 1500, 2000. The identified pulse responses are to be plotted after the simulation.

This requires that the subsystem INPUT is used to read the data file SDATA.BIN and connected to IDHG by using the CONNECTING SYSTEM TENG2 (Appendix A). The sequence of commands given below illustrate how this simulation was run.

```
≥SYST INPUT IDHG TENG2  "
≥LET IDOC.=1            " Documentation page
≥,FILN.=SDATA           "
≥,FNRS.=SIMPU           " Ref. determ. pulse response
≥,FNRN.=NIMPU           "  -     noise    -       -
≥,FNIM.=TIFIS           " Identified pulse responses
≥PAR NTYP:3             "
≥,NC2:7                 " Column 7 in SDATA
≥,METOD:4               " The CMA-approach
≥,NF:2                  " Model
≥,NB:2                  "   -
≥,ND:2                  "   -
≥,NC:2                  "   -
≥,IPRER:1               " The à posteriori estimate of ε
≥,IFIST:75              " Filtering starts at ISAMP=75
≥,ISTAB:1               " Check stability of F and C pol.
≥,IHAT:100              " Default IHATN is used.
≥,IFIGE:1               " Generate the file TIFIS (FNIM)
≥SIMU 1 2000
```

At the end of the simulation the program prints a documentation page and the values of the loss functions which is shown below. The pulse responses can then be viewed by the commands:

```
≥SHOW RS IS/TIFIS
≥SHOW RN IN/TIFIS
```

DOCUMENTATION		SIMU. NO:	1
*************		DATE :	0- 0- 0

SYSTEM INPUT:

NTYP =	3				
		NBIT =	7	NC1 =	1
IODD =	95	KNEP =	0	NC2 =	7
SD =	1.00000	ISTAR=	1	FILN =	SDATA
		IBP =	1		
DT =	1.00000	AMP =	1.00000		

SYSTEM IDHG:

IDENTIFICATION METHOD: 4 CMA

NA =	0	IPRER=	1	IWRT =	-4
NF =	2	IFIST=	75	IWRT1=	0
NB =	2	ISTAB=	1	IWRTN=	100
ND =	2	INVAU=	0	IHAT =	100
NC =	2			IHATN=	500
KD =	0	NFST =	0	IFIGE=	1
WTI =	1.00000	PFST =	100.000	FNRS =	SIMPU
WTO =	1.00000	RKJ =	0.000000	FNRN =	NIMPU
RLIM =	-1.00000			FNIM =	TIFIS
DELTA=	0.000000	DT =	1.00000	FNST =	

TH(1) =	-1.59478	THO(1) =	0.000000	PO(1) =	100.000
TH(2) =	0.673481	THO(2) =	0.000000	PO(2) =	100.000
TH(3) =	0.762903E-01	THO(3) =	0.000000	PO(3) =	100.000
TH(4) =	0.372319E-01	THO(4) =	0.000000	PO(4) =	100.000
TH(5) =	-1.40728	THO(5) =	0.000000	PO(5) =	100.000
TH(6) =	0.459908	THO(6) =	0.000000	PO(6) =	100.000
TH(7) =	-0.809378	THO(7) =	0.000000	PO(7) =	100.000
TH(8) =	0.221173	THO(8) =	0.000000	PO(8) =	100.000

R1(1) =	0.000000
R1(2) =	0.000000
R1(3) =	0.000000
R1(4) =	0.000000
R1(5) =	0.000000
R1(6) =	0.000000
R1(7) =	0.000000
R1(8) =	0.000000

ACCURACY
****

0- 0- 0

SIMULATION NO: 1

$V = SUM[HR(I)-HI(I)]^2$ $VN = V/SUM[HR(I)]^2$ I=1,100

ISAMP	$VE = V_1$ $E[RES^2]$	V_2 $E[WRES^2]$	V_4 V[SYS]	$VRS = V_3$ VN[SYS]	V_6 V[NOISE]	$VRN = V_5$ VN[NOISE]
100	0.420249E-01	0.369927E-01	0.357327E-01	0.147233	4.39415	1.58595
500	0.336632E-01	0.315461E-01	0.543627E-02	0.223995E-01	0.376417	0.135857
1000	0.330492E-01	0.317992E-01	0.480495E-02	0.197982E-01	0.768192E-01	0.277258E-01
1500	0.320889E-01	0.311778E-01	0.506125E-02	0.208543E-01	0.177486	0.640588E-01
2000	0.323713E-01	0.316550E-01	0.543533E-02	0.223956E-01	0.884296E-01	0.319162E-01

5.4 SUBSYSTEM AUXIL

The subsystem AUXIL is not a proper simulation subsystem in the SIMNON sense but it has been implemented to facilitate the inclusion of some programs which may come in handy from time to time in connection with identification experiments. It has but one paremeter, to decide which program is to be called. Common to all the programs is that they operate in question and answer form via the Terminal.

```
ISUB        Parameter                  (default value: 0)
      :1    Call GENHG
      :2     -   SIMHG
      :3     -   STAT
      :4     -   FILCON
      :5     -   VSTAT
      :6     -   TAFLA
```

5.4.1 Program GENHG

By choosing poles and zeroes for a transfer function of the form:

$$y(t) = \frac{1 + b_1 q^{-1} + b_2 q^{-2} + \ldots\ldots b_n q^{-n}}{1 + f_1 q^{-1} + f_2 q^{-2} + \ldots\ldots f_m q^{-m}} \cdot u(t-1+ITYP)$$

the program computes the first 100 impulse response coefficients and places them on a file. If ITYP=0 (TYP=ONE BELOW) then the numerator is divided by the steady state gain to get a transfer function with a steady state gain equal to 1. The use of the program is best described by the following example.

Example 5.4.1:

```
>SYST AUXIL
>PAR ISUB:1
>SIMU 0 1

PROGRAM GENHG: GENERATION OF IMPULSE RESPONSES
TYP=STOP   : STOP
TYP=ONE    : ONE TIME DELAY IN THE NUMERATOR
TYP=NO     : NO TIME DELAY

TYP=NO
READ POLES
```

```
T=STOP : NO MORE POLES
T=R    : A REAL POLE
T=C    : COMPLEX POLES
T=R
REAL POLE      =0.92
T=R
REAL POLE      =0.75
T=C
REAL PART      =0.8
IMAGINARY PART =0.2
T=STOP

READ ZEROS
T=STOP : NO MORE ZEROS
T=R    : A REAL ZERO
T=C    : COMPLEX ZEROS
T=R
REAL ZERO      =0.5
T=R
REAL ZERO      =0.7
T=C
REAL PART      =0.75
IMAGINARY PART =0.1
T=STOP

FILE NAME FOR COEFFICIENTS OF IMPULSE RESPONSE:
IF FILE NAME=STOP THEN NO FILE IS GENERATED

FILE NAME=IMPUN
```

The program then gives a documentation page on the line printer. The print-out is given in sec. 6.1. The pulse response can be plotted by giving the command SHOW ONE/<file name> (NO/<file name> if NO was given as an answer at the start).

5.4.2 Program SIMHG

SIMHG is a program for simulation using impulse response model i.e. given the system impulse response coefficients h_d, the noise impulse coefficients h_n, an insignal u(t) (on file), an odd number for the random number generator and SNR the signal to noise ratio, the program computes the output of the equation given below and places it on a file.

$$y(t) = \Sigma\, h_d(s)\cdot u(t-s) + k\cdot\Sigma\, h_n(s)\cdot e(t-s)$$

$$= YS(t) + k\cdot YN(t)$$

where $k = \sqrt{\dfrac{Var(YS)}{Var(YN)\cdot SNR}}$

The insignal file must contain at least 2101 data points. The file generated by SIMHG is an IDPAC compatible data file and contains 2001 data points with the insignal in column 1 and the output signal in column 2. If more then one odd numbers is given the successive output signals are placed in columns 2, 3,...etc. Max 10 output signals are possible.

Example_5.4.2:

```
>SYST AUXIL
>PAR ISUB:2
>SIMU 0 1

PROGRAM_SIMHG

PROGRAM_FOR_SIMULATION_USING_IMPULSE_RESPONSE_MODEL

FILE_NAME_FOR_COEFFICIENTS_OF_SYSTEM_IMPULSE_RESPONSE
FILE_NAME_=_SIMPU

FILE_NAME_FOR_COEFFICIENTS_OF_NOISE_IMPULSE_RESPONSE
FILE_NAME_=_NIMPU

FILE_NAME_WITH_INPUT_DATA_=_UPRBS

COLUMN_NR._OF_INPUT_DATA_IN_UPRBS
NCOL_=_1

ODD_NUMBER_FOR_THE_NOISE_GENERATOR
ONE_NUMBER_FOR_EACH_SIMULATION,_AND_A_NEGATIVE_NUMBER
TO_STOP

KODD_=_7
       .    The last 4 lines are repeated for
       .    KODD = 17, 27, 37, 47, 57, 67, 77, 87, 97

DESIRED_SIGNAL_TO_NOISE_RATIO
SNR_=_10.0

FILE_NAME_FOR_THE_GENERATED_DATA_=_SDATA
```

This set of data was used in simulations reported in sec. 6 where the documentation printed by the program is given.

5.4.3 Program STAT

STAT computes the means and the standard deviations of the parameter estimates and the loss functions which the subsystem IDHG has placed on file FNST.

Example 5.4.3

```
>SYST AUXIL
>PAR ISUB:3
>SIMU 0 1

PROGRAM STAT

IF FILE NAME = STOP    :    STOP
PRESS RETURN

FILE NAME = M5311
```

The results are printed on the line printer and an example is given in Appendix B.

5.4.4 Program FILCON

In order to make use of the powerful IDPAC program package a program to convert SIMNON data files to IDPAC format is included. The use is best illustrated by an example:

Example 5.4.4:
In example 5.1.2 a PRBS sequence was generated to be used as an input signal for simulation by SIMHG. Here the file is converted to IDPAC format:

```
>SYST AUXIL
>PAR ISUB:4
>SIMU

PROGRAM FILCON
INPUT FILE NAME = STOP    :    STOP
PRESS RETURN
```

```
INPUT FILE NAME = PRBSU

OUTPUT FILE NAME = UPRBS

SAMPLE INTERVAL = 1.0

INPUT FILE HEAD:
100000     2     1     0     0     0     0     0     0     1
VARIABLES IN INPUT FILE
PRBS
NUMBER OF DATA RECORDS IN INPUT FILE : 4221

DELETE TIME COLUMN : YES

DO YOU WANT THE INPUT FILE DELETED : YES

INPUT FILE NAME = STOP

>
```

5.4.5 Program VSTAT

This program prints the results from identification experiments, i.e. the means and standard deviations of the loss functions, again best described by an example.

Example 5.4.5

```
>SYST AUXIL
>PAR ISUB:5
>SIMU 0 1


PROGRAM VSTAT

IF FILE NAME = STOP    :    STOP
PRESS RETURN

FILE NAME = M5311

PRINT-OUT FOR ISAMP = IHAT, IHATN, N*KH*IHATN
N = 1,2,......
KH = 4

FILE NAME = STOP

>
```

The results computed by VSTAT and a listing of the results from the simulation runs contained in the file M5311 is given in Appendix B.

5.4.6 Program TAFLA

In order to compare different structures the program TAFLA can be used to print the mean, standard deviation, minimum and maximum of the loss functions VE, VRS and VRN in a tabular form.

```
Example 5.4.6:
>SYST AUXIL
>PAR ISUB:6
>SIMU 0 1

PROGRAM TAFLA:
IF FILE NAME = STOP    :    STOP
IF FILE NAME = ENDFI   :    PRINT TAFLA
IF FILE NAME = NEWTS   :    NEW TSAMP
PRESS RETURN

FILE NAME = M1310

ISAMP = 100

FILE NAME = M2310
             .
             .
             .
FILE NAME = MA3X1
FILE NAME = ENDFI
```

The result is given at the beginning of appendix C.

6 SIMULATIONS

This section contains a report on a series of identification experiments designed so as to test the assumption made in section 2 that the general model structure would lead to a "better" description than commonly used model structures. This is of special interest when the system under observation is of a complex nature and the goal is to describe its most relevant features. The system used to generate the input-output sequences is described in sec. 6.1. A list of the model structures tried for the identification is given in sec. 6.2, where the number of parameters estimated is in all cases 8. This means that the amount of computing work is roughly the same for all structures. The results of the simulations are presented in sec. 6.3.

6.1 The "process/system"

The process/system that generated the data used for the identification was of the form:

$$y(t) = G(q^{-1})\cdot u(t) + k\cdot H(q^{-1})\cdot e(t)$$

$$= \Sigma\ h_d(s)\cdot u(t-s) + k\cdot\Sigma\ h_n(s)\cdot e(t-s)$$

$$= YS(t) + k\cdot YN(t)$$

where u(t) was a Pseudo Random Binary Sequence generated as in example 5.1.2 (cf. also 5.4.4). The noise e(t) was generated by a random number generator giving a normally distributed sequence with a mean of zero and variance of one.

The functions G and H were chosen to be of a relatively higher order than the models tried for the identification. By choosing poles and zeroes for G and H the program GENHG (sec. 5.4.1) was used to compute the transfer functions and the pulse responses which in turn were used by the program SIMHG (sec.5.4.2) to simulate the data. Ten different sequences were generated where the input signal was the same in all cases but with a different starting number for the noise generator for each output sequence. The signal to noise ratio was in all cases equal to 10, here defined as:

$$SNR = \frac{Var(YS)}{Var(k \cdot YN)}$$

Below follows the documentation printed by the programs SIMHG and GENHG.

```
PROGRAM SIMHG

PROGRAM FOR SIMULATION USING IMPULSE RESPONSE MODEL

FILE NAME SYSTEM RESPONSE  : SIMPU
FILE NAME NOISE RESPONSE   : NIMPU
FILE NAME INPUT DATA       : UPRBS
COLUMN NR. INPUT FILE      :  1

FILE NAME GENERATED DATA   : SDATA
                    COL  1: INPUT SIGNAL
                    COL  2: OUTPUT SIGNAL      KODD =     7
                    COL  3: OUTPUT SIGNAL      KODD =    17
                    COL  4: OUTPUT SIGNAL      KODD =    27
                    COL  5: OUTPUT SIGNAL      KODD =    37
                    COL  6: OUTPUT SIGNAL      KODD =    47
                    COL  7: OUTPUT SIGNAL      KODD =    57
                    COL  8: OUTPUT SIGNAL      KODD =    67
                    COL  9: OUTPUT SIGNAL      KODD =    77
                    COL 10: OUTPUT SIGNAL      KODD =    87
                    COL 11: OUTPUT SIGNAL      KODD =    97
SIGNAL TO NOISE RATIO      :  10.0000
```

```
PROGRAM GENHG:

ONE TIME DELAY
REAL POLE      = 0.40000
REAL POLE      = 0.80000
REAL POLE      = 0.90000
COMPLEX POLES = 0.80000 +/- 0.15000
COMPLEX POLES = 0.70000 +/- 0.25000
REAL ZERO     =-0.50000
REAL ZERO     = 0.95000
COMPLEX ZEROS = 0.60000 +/- 0.20000
COMPLEX ZEROS = 0.70000 +/- 0.10000

NUMERATOR POLYNOMIAL:
  0.762501D-01  -0.232563D+00   0.249719D+00  -0.828076D-01  -0.383919D-01
  0.351513D-01  -0.724376D-02

DENOMINATOR POLYNOMIAL:
  0.100000D+01  -0.510000D+01   0.111550D+02  -0.135550D+02   0.987118D+01
 -0.429981D+01   0.103416D+01  -0.105417D+00
```

IMPULSE RESPONSE COEFFICIENTS:

0	0.000000	25	-0.186160E-01	50	-0.118738E-02	75	-0.854672E-04
1	0.762501E-01	26	-0.165102E-01	51	-0.107248E-02	76	-0.769398E-04
2	0.156313	27	-0.145944E-01	52	-0.967970E-03	77	-0.692650E-04
3	0.196344	28	-0.128755E-01	53	-0.873012E-03	78	-0.623566E-04
4	0.208449	29	-0.113497E-01	54	-0.786849E-03	79	-0.561375E-04
5	0.200619	30	-0.100066E-01	55	-0.708774E-03	80	-0.505384E-04
6	0.179376	31	-0.883161E-02	56	-0.638123E-03	81	-0.454972E-04
7	0.150303	32	-0.780817E-02	57	-0.574273E-03	82	-0.409583E-04
8	0.117990	33	-0.691921E-02	58	-0.516640E-03	83	-0.368716E-04
9	0.859282E-01	34	-0.614799E-02	59	-0.464675E-03	84	-0.331922E-04
10	0.565161E-01	35	-0.547869E-02	60	-0.417863E-03	85	-0.298794E-04
11	0.311701E-01	36	-0.489684E-02	61	-0.375729E-03	86	-0.268968E-04
12	0.105080E-01	37	-0.438951E-02	62	-0.337829E-03	87	-0.242117E-04
13	-0.544246E-02	38	-0.394533E-02	63	-0.303754E-03	88	-0.217943E-04
14	-0.170443E-01	39	-0.355455E-02	64	-0.273130E-03	89	-0.196181E-04
15	-0.248818E-01	40	-0.320889E-02	65	-0.245615E-03	90	-0.176590E-04
16	-0.296298E-01	41	-0.290145E-02	66	-0.220894E-03	91	-0.158953E-04
17	-0.319619E-01	42	-0.262652E-02	67	-0.198686E-03	92	-0.143077E-04
18	-0.324934E-01	43	-0.237944E-02	68	-0.178733E-03	93	-0.128785E-04
19	-0.317517E-01	44	-0.215643E-02	69	-0.160804E-03	94	-0.115919E-04
20	-0.301661E-01	45	-0.195445E-02	70	-0.144692E-03	95	-0.104338E-04
21	-0.280711E-01	46	-0.177101E-02	71	-0.130209E-03	96	-0.939126E-05
22	-0.257166E-01	47	-0.160411E-02	72	-0.117187E-03	97	-0.845280E-05
23	-0.232817E-01	48	-0.145209E-02	73	-0.105477E-03	98	-0.760802E-05
24	-0.208886E-01	49	-0.131358E-02	74	-0.949437E-04	99	-0.684760E-05

FILE NAME: IMPUS → SIMPU

↑

Filcon:

PROGRAM GENHG:

NO TIME DELAY
REAL POLE = 0.92000
REAL POLE = 0.75000
COMPLEX POLES = 0.80000 +/- 0.20000
REAL ZERO = 0.50000
REAL ZERO = 0.70000
COMPLEX ZEROS = 0.75000 +/- 0.10000

NUMERATOR POLYNOMIAL:
0.100000D+01 -0.270000D+01 0.272250D+01 -0.121200D+01 0.200375D+00

DENOMINATOR POLYNOMIAL:
0.100000D+01 -0.327000D+01 0.404200D+01 -0.223960D+01 0.469200D+00

IMPULSE RESPONSE COEFFICIENTS:

0	1.00000	25	0.535218E-01	50	0.653868E-02	75	0.812604E-03
1	0.570000	26	0.496431E-01	51	0.601943E-02	76	0.747631E-03
2	0.544400	27	0.458915E-01	52	0.554024E-02	77	0.687843E-03
3	0.503848	28	0.422980E-01	53	0.509818E-02	78	0.632829E-03
4	0.454865	29	0.388893E-01	54	0.469059E-02	79	0.582209E-03
5	0.402650	30	0.356855E-01	55	0.431496E-02	80	0.535632E-03
6	0.351085	31	0.326993E-01	56	0.396899E-02	81	0.492778E-03
7	0.302849	32	0.299364E-01	57	0.365048E-02	82	0.453350E-03
8	0.259581	33	0.273956E-01	58	0.335739E-02	83	0.417075E-03
9	0.222081	34	0.250704E-01	59	0.308778E-02	84	0.383702E-03
10	0.190512	35	0.229505E-01	60	0.283985E-02	85	0.352999E-03
11	0.164581	36	0.210223E-01	61	0.261191E-02	86	0.324753E-03
12	0.143710	37	0.192709E-01	62	0.240235E-02	87	0.298768E-03
13	0.127163	38	0.176804E-01	63	0.220972E-02	88	0.274863E-03
14	0.114157	39	0.162353E-01	64	0.203264E-02	89	0.252872E-03
15	0.103931	40	0.149205E-01	65	0.186985E-02	90	0.232640E-03
16	0.957967E-01	41	0.137223E-01	66	0.172018E-02	91	0.214028E-03
17	0.891683E-01	42	0.126279E-01	67	0.158255E-02	92	0.196905E-03
18	0.835709E-01	43	0.116264E-01	68	0.145598E-02	93	0.181153E-03
19	0.786406E-01	44	0.107078E-01	69	0.133957E-02	94	0.166661E-03
20	0.741147E-01	45	0.986370E-02	70	0.123249E-02	95	0.153329E-03
21	0.698174E-01	46	0.908683E-02	71	0.113398E-02	96	0.141063E-03
22	0.656432E-01	47	0.837092E-02	72	0.104334E-02	97	0.129779E-03
23	0.615406E-01	48	0.771060E-02	73	0.959949E-03	98	0.119397E-03
24	0.574962E-01	49	0.710121E-02	74	0.883214E-03	99	0.109846E-03

FILE NAME:IMPUN → NIMPU

↑
FILCON

6.2 Model Structures and Simulations

The model structures tried for the identification are given in Table 1 below. Note that structure no. 2 is the LS-model, no. 4 the ELS-ML-model and no. 7 the GLS-model so that a comparison between these methods and the more general LIP and CMA approaches for other structures can be obtained.

Ten simulation runs were made for each structure (the ten sequences generated by SIMHG (6.1)) using:

a) The LIP-approach (sec. 2.3)

b) The CMA-approach (sec. 2.4) IFIST=0

c) do. IFIST=50

The loss functions VE, VRS and VRN given in sec. 2.7 were evaluated for these runs and their mean and standard deviations computed. The results are given and commented upon in the next section. The results as given by the program TAFLA (cf. sec. 5.4.6) are listed in appendix C. A more detailed listing of the results e.g. as given by the programs STAT and VSTAT (cf. sec. 5.4.3 and 5.4.5) as well as a complete listing of all results "logged" during the simulations is available on request from the author.

The following SIMNON parameters (cf.sec 5 for their function) had the same value for all runs as can be noted on the documentation pages given in the appendices for each FNST file name.

```
IPRER = 1        WTI   =  1.0     R1(i)  =   0.0
ISTAB = 1        WTO   =  1.0     THO(i) =   0.0
IHAT  = 100      RLIM  = -1.0     PO(i)  = 100.0     i=1,..10
IHATN = 500      DELTA =  0.0
```

(The parameter ISTAB is redundant for structures 1,2 and 7. ISTAB=0 for these runs.)

Table 1.
Model Structures

1 $y = B\cdot u + \varepsilon$ $n_b=8$	
2 $(1+A)\cdot y = B\cdot u + \varepsilon$ $n_a=4$ $n_b=4$	7 $(1+A)\cdot y = B\cdot u + \frac{1}{1+D}\cdot\varepsilon$ $n_a=3$ $n_b=3$ $n_d=2$
3 $y = \frac{B}{1+F}\cdot u + \varepsilon$ $n_b=4$ $n_f=4$	8 $y = \frac{B}{1+F}\cdot u + \frac{1}{1+D}\cdot\varepsilon$ $n_b=3$ $n_f=3$ $n_d=2$
4 $(1+A)\cdot y = B\cdot u + (1+C)\cdot\varepsilon$ $n_a=3$ $n_b=3$ $n_c=2$	9 $(1+A)\cdot y = \frac{B}{1+F}\cdot u + \frac{1+C}{1+D}\cdot\varepsilon$ $n_a=2$ $n_b=2$ $n_c=1$ $n_f=2$ $n_d=1$
5 $y = \frac{B}{1+F}\cdot u + \frac{1+C}{1+D}\cdot\varepsilon$ $n_b=2$ $n_c=2$ $n_f=2$ $n_d=2$	10 $(1+A)\cdot y = \frac{B}{1+F}\cdot u + (1+C)\cdot\varepsilon$ $n_a=2$ $n_b=2$ $n_c=2$ $n_f=2$
5A $n_b=3$ $n_c=1$ $n_f=3$ $n_d=1$	10A $n_a=3$ $n_b=2$ $n_c=1$ $n_f=2$
6 $(1+A)\cdot y = B\cdot u + \frac{1+C}{1+D}\cdot\varepsilon$ $n_a=2$ $n_b=2$ $n_c=2$ $n_d=2$	11 $(1+A)\cdot y = \frac{B}{1+F}\cdot u + \frac{1}{1+D}\cdot\varepsilon$ $n_a=2$ $n_b=2$ $n_f=2$ $n_d=2$
6A $n_a=3$ $n_b=3$ $n_c=1$ $n_d=1$	11A $n_a=2$ $n_b=2$ $n_f=3$ $n_d=1$

6.3 Results

6.3.1 The LIP-Approach

Fig 6.3.1 gives the value of the loss function VE at the sampling times 100, 500, 2000. The length of the lines in the diagram is twice the standard deviation. For all structures it can be noted that VE decreases as a function of the simulation time, which is to be expected. It is difficult to pinpoint the "best" structure by use of this criterion, but slightly better results are obtained for the structures 5A, 8, 7 and 5 in the said order. The structures 1 and 3 give the "worst" results which can be explained by, that no attempt is made there to model the noise.

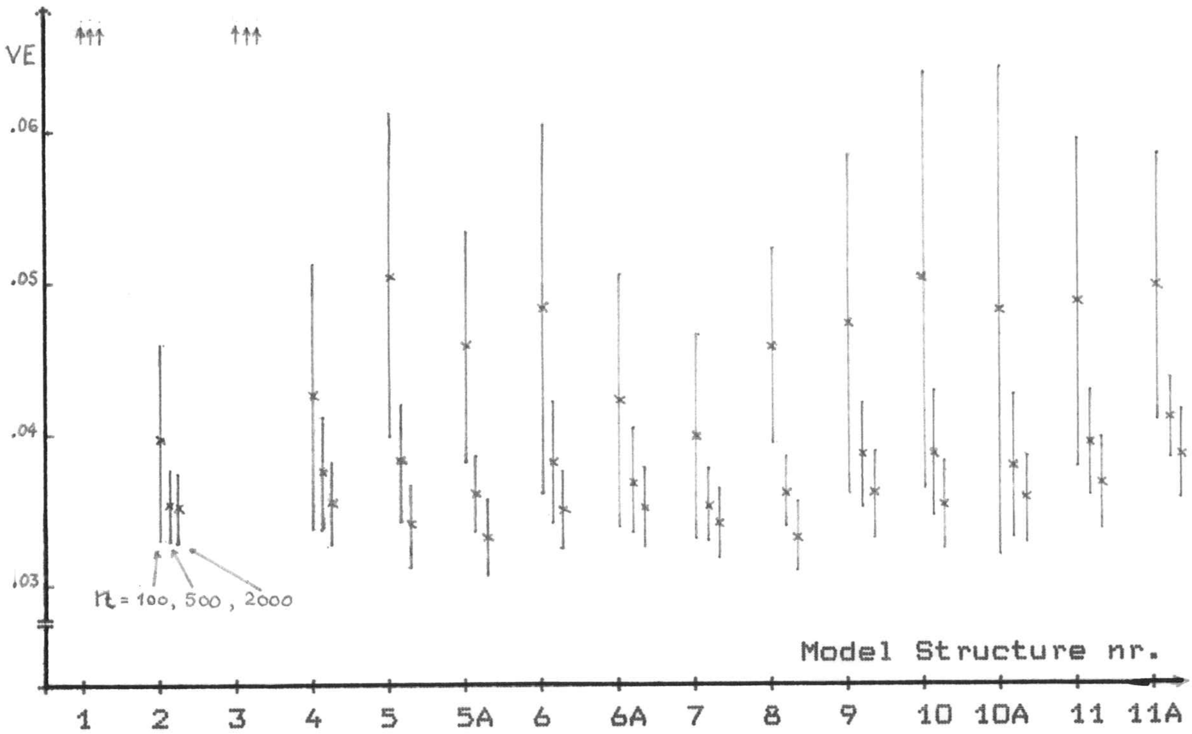


Fig. 6.3.1 The loss function $VE = 1/n \, \Sigma \varepsilon(t)^2$ for n=100, 500 and 2000 respectively for all model structures.

Fig. 6.3.2 shows the loss functions VRS and VRN at the sampling time ISAMP = 2000. Observe that VRS gives lower values for all structures than VRN which is reportedly not an uncommon phenomenon in identification experiments, i.e. the modelling of the noise is more difficult and those parameters converge more slowly.

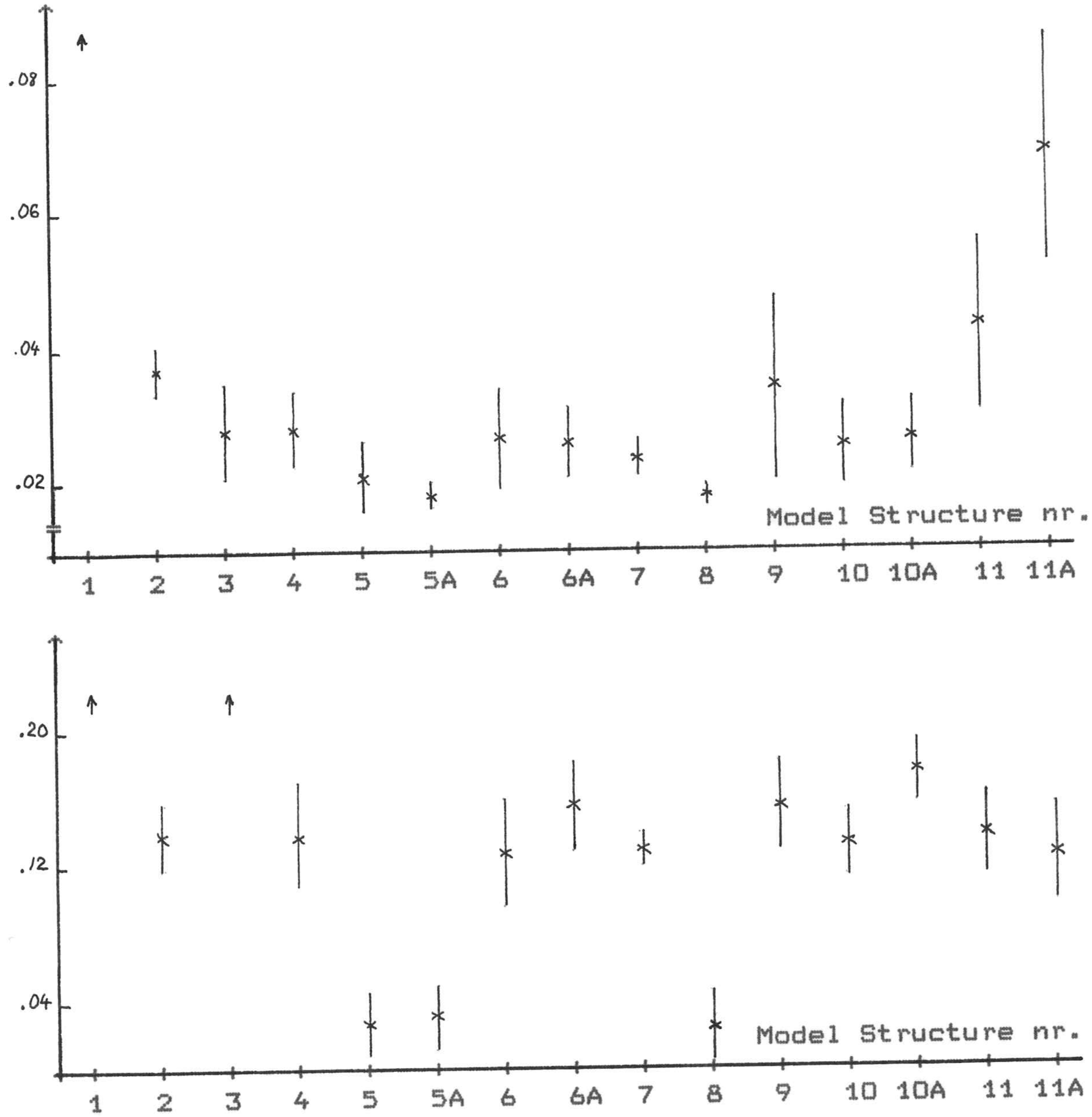


Fig 6.3.2 The loss functions VRS and VRN for all model structures.

The "best" results are obtained for the structures no. 5A, 5 and 8, i.e. they give relatively low values for both VRS and VRN which indicates that a good "fit" is obtained for the system and the noise. It is, however, not clear how to rank the structures. As for VE the "worst" results are obtained for structures 1 and 3. Considering the manner in which data was generated (sec. 6.1) it is interesting to note that the structures that most "resemble" the "process" give the "best" results.

Table 2 below gives the mean values and the standard deviations of the parameter estimates at the end of the simulations (ISAMP = 2000). Now these estimates do not of course represent any "true" values and therefore the values themselves are of limited interest except as a source for comparison between similar structures on one hand and between the LIP and the CMA-approaches on the other. Furthermore a large deviation could indicate that the algorithm had tended to different local minima for some of the runs or that convergence was slow.

Any really conclusive information is difficult to obtain but some interesting observations can be made. For instance that structures 5A and 8 where the deterministic part is modelled in the same manner do give very similar results. The same can be said to a degree of the structures 6A and 4 but now the modelling of the noise influences the estimates of the A-polynomial which becomes more apparent when compared with 7.

The rate of convergence is of course of interest in this context. As can be noted in sec. 6.2 the SIMNON parameters IHAT and IHATN had the values 100 and 500 respectively which means that the parameter estimates were logged at the sampling times 100, 500, 1000, 1500 and 2000. From this data (not reviewed here) and notes made during some of the simulations it was observed that the parameter estimates for:

a) Structure 2 - the LS method - converged fast. The estimates were close to their final values after ca. 100 samples and gave similar results for all runs.
b) Structure 4 - the ELS method - converged fast to begin with but after ca 100-200 the rate of convergence became slower. The estimates either increased or decreased almost linearly from 100 onwards. The resulting estimates varied much more than for the LS method.
c) Structure 5 were in general fairly close after 100 samples. The "noise" parameters d and c converged more slowly and varied more than the "system" parameters f and b.
d) Structure 5A were fairly close after 100-200 samples, this time the f-parameters showed a slightly slower rate of convergence.
e) Structure 8 were generally close to their final values after ca. 100 samples. It did occur though that some estimates converged more slowly.

Model nr.:

	2	3	4	5	5A	6	6A
a_1	−0.70 ± .02		−1.41 ± .14			−1.58 ± .03	−1.43 ± .08
a_2	−0.29 ± .02		0.39 ± .23			0.66 ± .03	0.41 ± .11
a_3	−0.03 ± .03		0.11 ± .10				0.11 ± .05
a_4	0.20 ± .02						
f_1		−.856 ±.117		−1.59 ± .04	−0.96 ± .08		
f_2		−.187 ±.153		0.68 ± .03	−0.37 ± .13		
f_3		−.096 ±.053			0.46 ± .06		
f_4		.305 ±.056					
b_1	.072 ±.007	.066 ±.013	.073 ±.007	.071 ±.005	.074 ±.006	.072 ±.006	.073 ±.007
b_2	.103 ±.010	.129 ±.017	.053 ±.020	.041 ±.012	.087 ±.012	.042 ±.012	.053 ±.018
b_3	.066 ±.010	−.003 ±.031	.005 ±.019		.017 ±.017		.002 ±.018
b_4	.035 ±.011	.022 ±.024					
d_1				−1.06 ± .23	−0.92 ± .01	0.22 ± .17	0.16 ± .07
d_2				0.14 ± .21		.088 ±.064	
c_1			−0.75 ± .16	−0.44 ± .22	−0.32 ± .02	−0.71 ± .15	−0.61 ± .05
c_2			0.11 ± .14	0.06 ± .07		0.14 ± .13	

	7	8	9	10	10A	11	11A
a_1	−0.97 ± .04		−1.56 ± .06	−1.59 ± .03	−1.33 ± .02	−1.47 ± .07	−1.07 ± .41
a_2	−0.31 ± .07		0.64 ± .05	0.66 ± .03	0.24 ± .04	0.56 ± .06	0.25 ± .32
a_3	0.41 ± .04				0.18 ± .02		
f_1		−0.97 ± .09	0.24 ± .17	0.33 ± .13	0.18 ± .17	0.24 ± .14	−0.03 ± .36
f_2		−0.35 ± .15	−0.10 ± .13	−0.09 ± .11	−.075 ±.099	−0.25 ± .11	−0.38 ± .19
f_3		0.45 ± .07					0.02 ± .15
b_1	.073 ±.007	.074 ±.006	.072 ±.007	.072 ±.006	.072 ±.006	.070 ±.007	.069 ±.007
b_2	.085 ±.013	.087 ±.010	.063 ±.019	.069 ±.015	.081 ±.016	.072 ±.022	.084 ±.013
b_3	.023 ±.012	.016 ±.015					
d_1	0.32 ± .06	−0.61 ± .02	0.27 ± .05			0.75 ± .08	0.30 ± .41
d_2	0.38 ± .04	−0.26 ± .01				0.25 ± .09	
c_1			−0.61 ± .07	−0.93 ± .04	−0.66 ± .03		
c_2				0.25 ± .04			

TABLE 2: LIP N = 2000 ISTAB = 1 IPRER = 1

Simulations were also made with the SIMNON parameter ISTAB=0 i.e. no stability check of the F and C polynomials. On average these runs gave results similar to the ones reported above but some parameter estimates occasionally tended to quite different values. However structures 3 and 11 gave overall quite different results and the same can be said to a degree of structures 5 i.e. the "noise" parameters d and c.

6.3.2 The CMA-approach

Fig. 6.3.3 gives the loss function VE at the sampling times 100, 500 and 2000 for the CMA-approach when IFIST=0. As before, for the LIP-case, VE decreases as a function of the simulation time, but in this case VE gives higher values and the deviation is consierably greater. This makes it very difficult to pick out any "best" model structures.

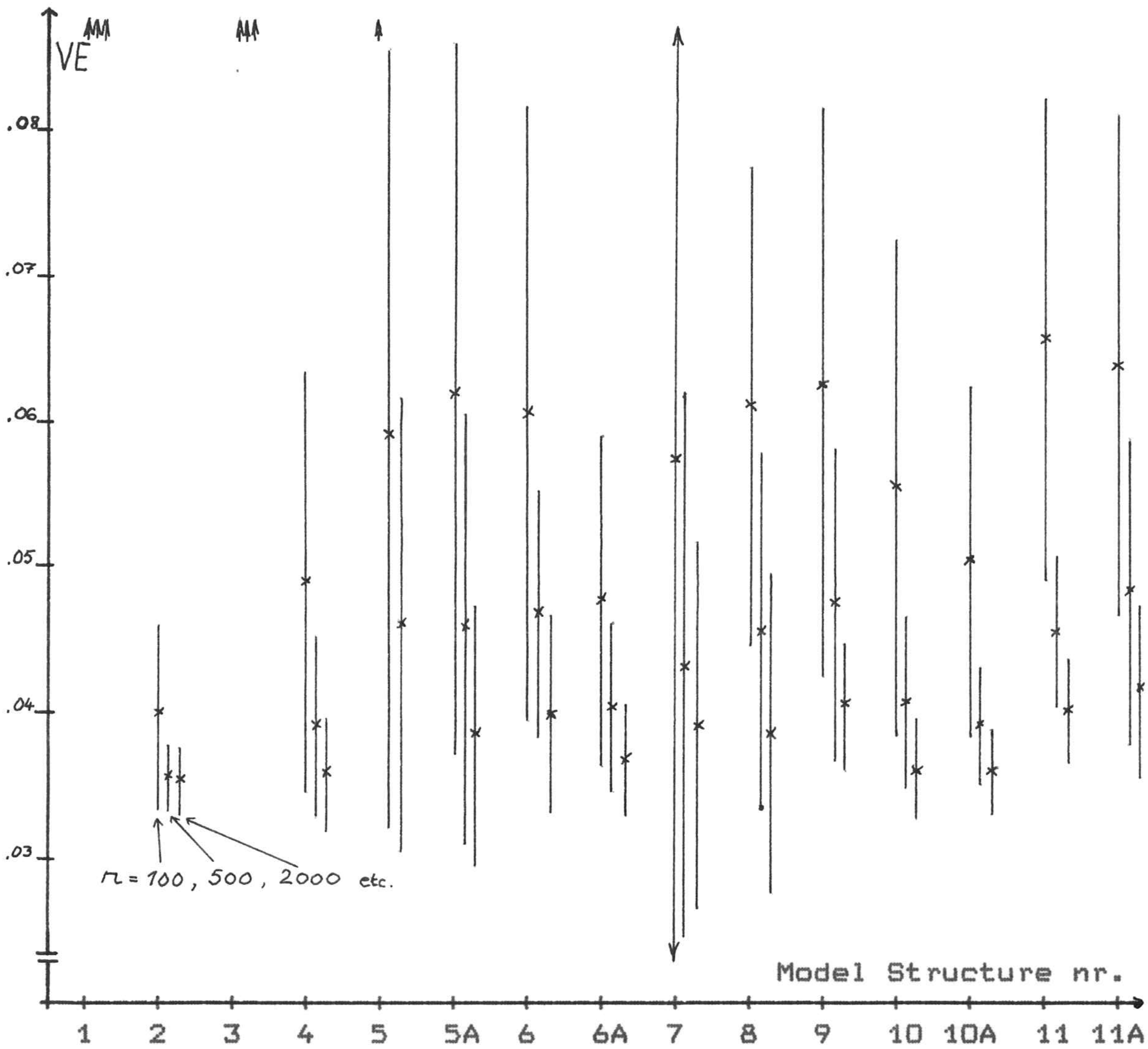


Fig. 6.3.3 The loss function VE for n=100, 500 and 2000
CMA : IFIST=0

Fig 6.3.4 gives VE as in fig 6.3.3 but now for the simulation runs when IFIST=50, and here a significant improvement is observed. The results obtained are very much the same as for the LIP-approach i.e. structures 5, 8, 5A and 7 seem to give slightly better results and again structure 3 gives the "worst" results.

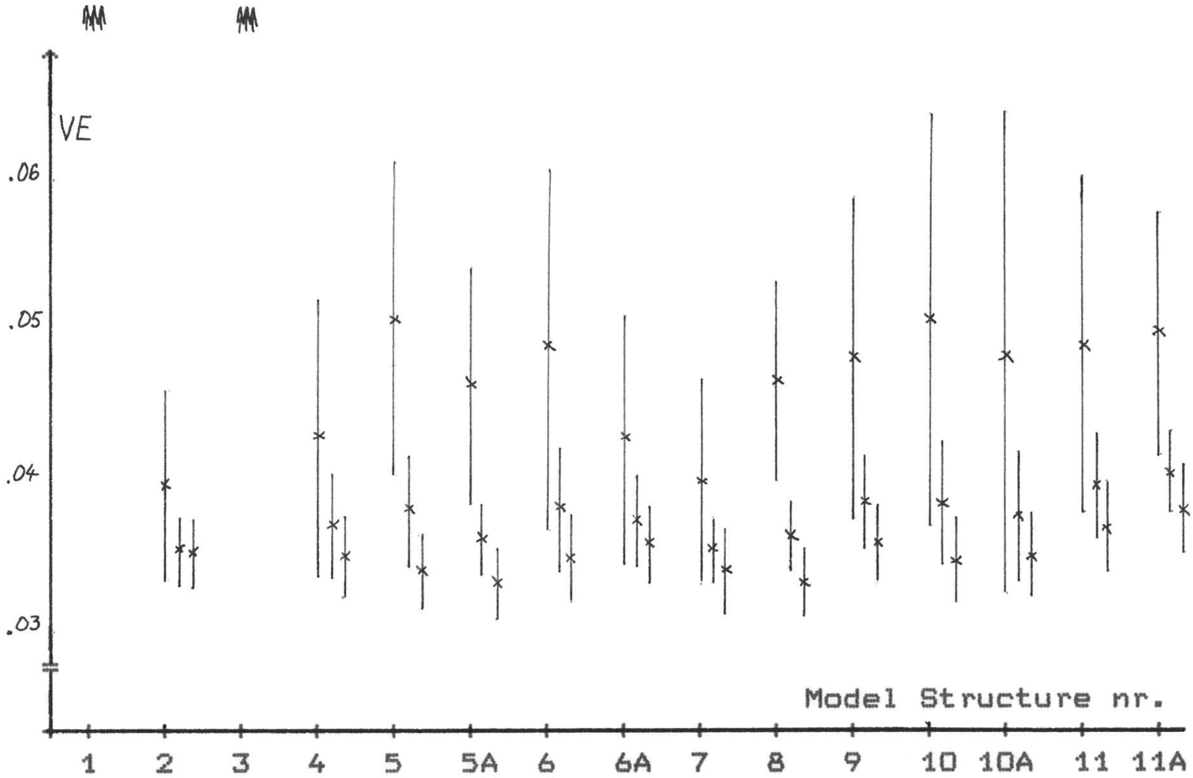


Fig. 6.3.4 The loss function VE for n=100, 500 and 2000
CMA : IFIST=50

Fig 6.3.5 shows the loss funtions VRS and VRN for IFIST=0. The relatively high deviation indicates that several local minima were reached or that convergence was slow. It is almost impossible to pick out any structure as the "best" one. Fig 6.3.6 shows VRS and VRN for IFIST=50. Now again a significant improvement is observed i.e. the loss functions give lower values and the deviation decreases and as in the LIP-case structures 5, 8 and 5A give the "best" results.

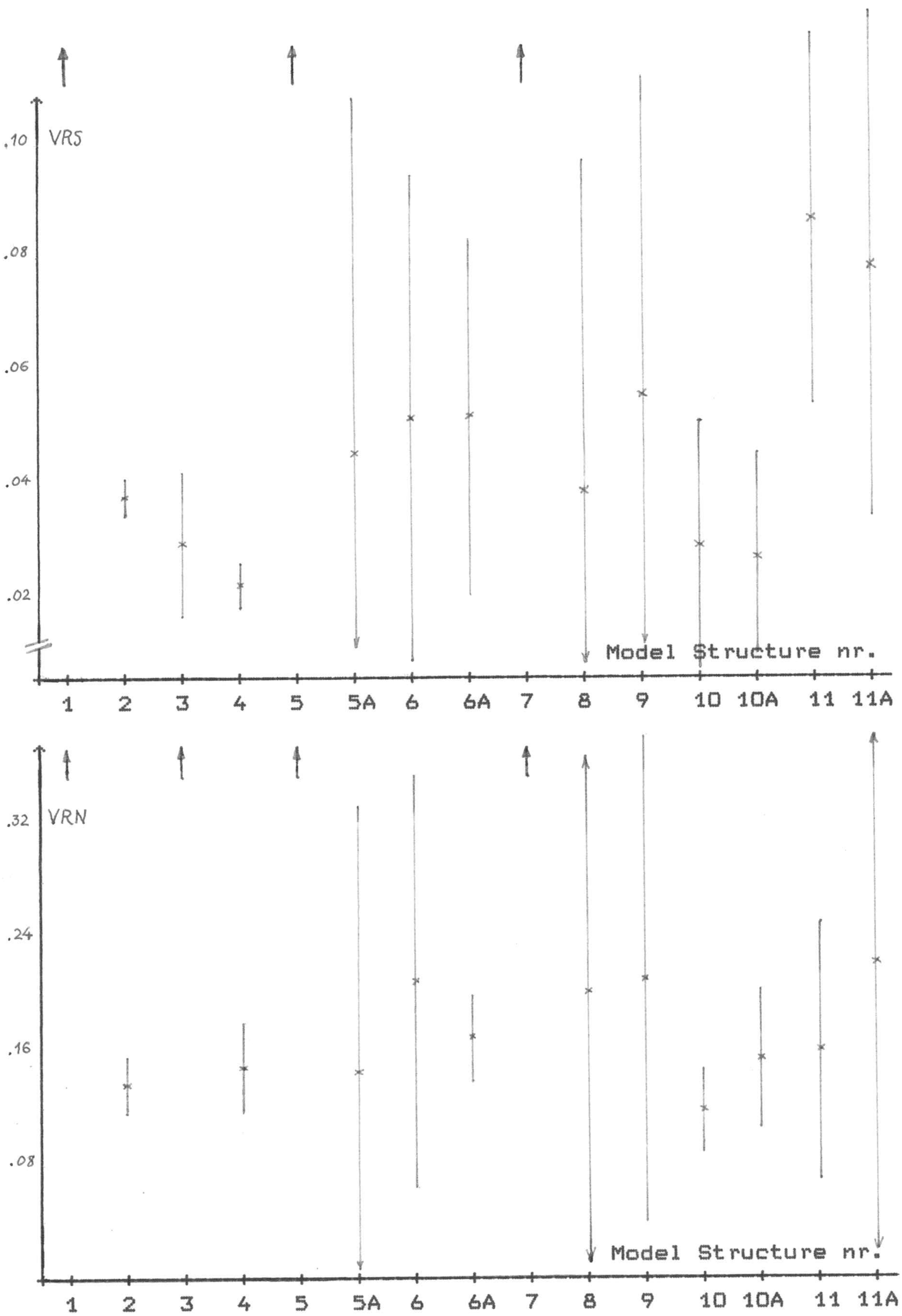


Fig. 6.3.5 The loss functions VRS and VRN for CMA : IFIST=0

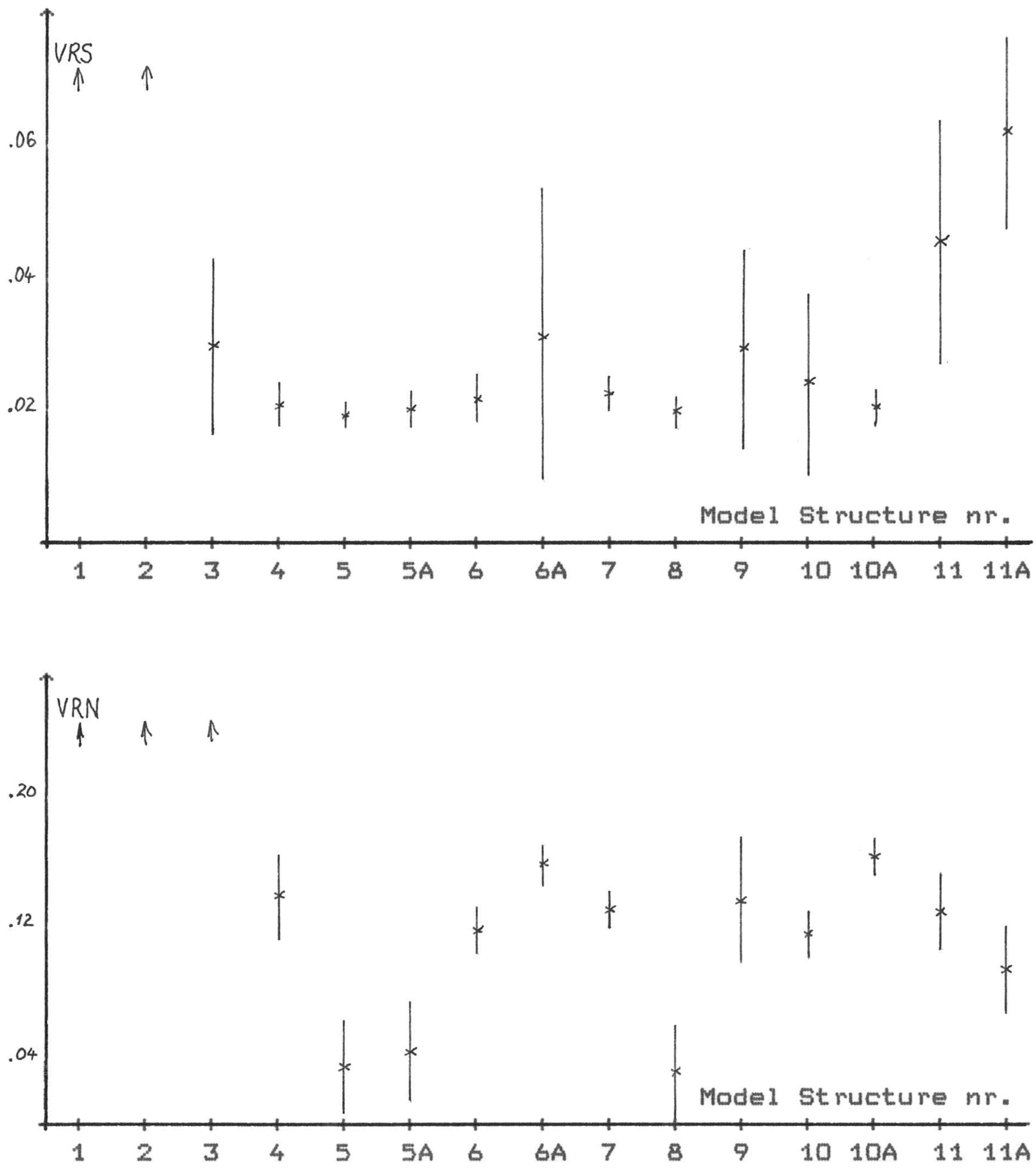


Fig. 6.3.6 The loss functions VRS and VRN for CMA : IFIST=50

Table 3 gives the mean values and deviations of the parameter estimates at ISAMP=2000 for the simulation runs with IFIST=50. It is interesting to note that for most of the structures the results are quite similar to those obtained for the LIP-approach with a slightly higher deviation though. Still, the differences for structure 3, 6A, 9 and 11A should be noted.

Model nr.:

	2	3	4	5	5A	6	6A
a_1	−0.70 ± .02		−1.46 ± .24			−1.60 ± .02	−1.26 ± .27
a_2	−0.29 ± .02		0.50 ± .34			0.67 ± .02	0.18 ± .31
a_3	−0.03 ± .03		0.10 ± .15				0.19 ± .07
a_4	0.20 ± .02						
f_1		−1.32 ± .56		−1.61 ± .01	−0.95 ± .24		
f_2		0.40 ± .86		0.69 ± .01	−0.37 ± .38		
f_3		−0.18 ± .37			0.45 ± .16		
f_4		0.20 ± .15					
b_1	.072 ±.007	.045 ±.020	.072 ±.008	.077 ±.003	.075 ±.006	.076 ±.005	.071 ±.007
b_2	.103 ±.010	.130 ±.023	.054 ±.025	.032 ±.006	.088 ±.021	.034 ±.009	.070 ±.020
b_3	.066 ±.010	−.020 ±.116	.004 ±.018		.017 ±.015		.012 ±.038
b_4	.035 ±.011	.020 ±.074					
d_1				−1.16 ± .25	−0.92 ± .01	0.09 ± .11	0.29 ± .24
d_2				0.23 ± .23		0.08 ± .02	
c_1			−0.80 ± .28	−0.55 ± .25	−0.32 ± .01	−0.87 ± .11	−0.28 ± .56
c_2			0.18 ± .22	0.11 ± .09		0.28 ± .08	

	7	8	9	10	10A	11	11A
a_1	−0.98 ± .03		−1.43 ± .46	−1.58 ± .09	−1.27 ± .02	−1.45 ± .10	−0.93 ± .52
a_2	−0.30 ± .05		0.57 ± .25	0.67 ± .08	0.12 ± .03	0.55 ± .09	0.17 ± .24
a_3	0.41 ± .02				0.25 ± .01		
f_1		−0.95 ± .12	−0.12 ± .59	0.23 ± .20	0.03 ± .26	0.16 ± .16	−0.28 ± .40
f_2		−0.38 ± .20	0.06 ± .24	−0.03 ± .16	0.05 ± .11	−0.23 ± .08	−0.42 ± .19
f_3		0.46 ± .08					0.20 ± .18
b_1	.073 ±.006	.075 ±.006	.074 ±.007	.072 ±.006	.071 ±.007	.070 ±.008	.072 ±.007
b_2	.085 ±.012	.088 ±.013	.042 ±.030	.061 ±.018	.077 ±.024	.068 ±.025	.068 ±.020
b_3	.020 ±.009	.016 ±.013					
d_1	0.33 ± .04	−0.61 ± .02	0.22 ± .40			0.73 ± .11	0.18 ± .51
d_2	0.37 ± .02	−0.26 ± .02				0.25 ± .10	
c_1			−0.52 ± .13	−0.95 ± .10	−0.60 ± .01		
c_2				0.33 ± .02			

TABLE 3: CMA N = 2000 ISTAB = 1 IPRER = 1 IFIST = 50

In order to get an idea of how the rate of convergence was for the CMA-approach it was observed that the parameter estimates for:

a) structure 4 - the ML method - behaved very much alike the LIP approach (the ELS=method). The rate of convergence became slower after ca. 100 samples and the estimates either increased or decreased almost linearly from 100 onwards.
b) structure 5 again behaved similarly to the LIP approach. Fairly close after ca. 100 samples with a slightly slower rate of convergence for the noise parameters.
c) structure 5A converged somewhat slower than for the LIP approach, but otherwise similar behavior. One run gave parameter estimates that differed considerably.
d) structure 7 - the GLS method - were in general fairly close after ca. 100 samples and gave similar results for all runs.
e) structure 8 were generally close to their final values after ca. 100 and as in the LIP case it did occur that some estimates converged more slowly.

7 CONCLUSIONS

To summarize the results of the simulations the following conclusions can be drawn.

The model structures that give the "best" approximate description of the "process" are structures no. 5, 5A and 8.

The LIP-approach gives better results than the CMA=approach (IFIST=0), but when CMA is run with IFIST=50 the results are roughly speaking the same. Ergo, the CMA is more susceptible to the initial values used at the start-up and it can be recommended that IFIST is set to some value greater than zero, say 50-100.

A comparison of the common identification methods i.e the LS, ELS, ML and GLS, with model structures 5, 5A and 8 gives that the use of the general model leads to "better" results than the methods mentioned above. The choice of a model structure has a far greater effect on the accuracy than the choice of approach i.e. LIP or CMA.

Now an important question in this context is how far can these results be generalized for an arbitrary "process". The results indicate that the use of the general model structure is to be recommended since it allows a greater flexibility and that the same or a higher degree of accuracy can be achieved with fewer parameters compared to the common identification methods. It is, however, at this point difficult to give any guidelines whether to use the LIP or the CMA-approach. Theoretically it may be said that the CMA-approach is to be preferred since it can be shown to be globally convergent even though the system cannot be described within the chosen model structure, but with the drawback that care must be taken at the start-up.

In order to confirm these results and possibly gain some additional information more simulations on data generated by different "processes" are desirable. These should then include experiments to test the influence of a lower signal to noise ratio, different starting values for the parameter vector θ and the covariance matrix P, the use of the forgetting profile as well as experiments to test the SA-versions. In addition a test where the models were used in connection with adaptive control could be made. Then the models could be judged by the performance of the regulator.

The EKF algorithm given in section 2.6 corresponds to a model structure where the polynomials F=0 and D=0. The algorithm is equivalent to the LS-method when C=0 which was confirmed during test runs but when C was set to some value greater than zero the parameters diverged and the algorithm gave useless results. The reason for this was not discovered and the problem is left to the interested reader to crack.

The aim has been to write the programs so as to make maintainance easy i.e. to smooth the way for any changes and/or improvements that may be needed. The program structure was partly restricted by the core memory space available. As mentioned in sec. 4 a special version of SIMNON was used in order to provide the necessary core memory space. The main shortcomings of this version is that the use of MACROs is limited, since they may not contain the commands SIMU and SYST. On a "larger" computer this will of course not cause any problems and therefore simulation experiments as the ones reported here become less tedious.

On hindsight some improvements on the program structure and facilities can be suggested.

The programs that evaluate and print the loss functions should preferably be split from the subsystem IDHG into a separate subsystem. This would make the programs more applicable as building bricks for a prospective user and implementation of amendments easier.

A facility to write the identification results onto a "system file" (in a similar manner as in IDPAC) and the same facility for the subsystem SYSHG to read such files. This might be useful when validating identification results.

In addition further improvements are possible concerning the use of data structures and conventions as e.g. outlined in [17]. This would make it easier to combine the use of the present programs and IDPAC or other program packages if the need arises.

8 REFERENCES

[1] L. Ljung:
Basic ideas in Recursive Identification.
Report LiTH-ISY-I-0179 Dept. of Electrical Engineering, Linköping University

[2] L. Ljung:
Analysis of Recursive Stochastic Algorithms.
IEEE Trans. AC-22 p. 551-575

[3] L. Ljung:
On Positive Real Transfer Functions and the Convergence of some Recursive Schemes.
IEEE Trans. AC-22 p. 539-551

[4] L. Ljung:
The Extended Kalman Filter as a Parameter Estimator for Linear Systems.
Report LiTH-ISY-I-0154 Dept. of Electrical Engineering, Linköping University.

[5] L. Ljung:
Convergence of an Adaptive Filter Algorithm.
Report LiTH-ISY-I-0120 Dept. of Elecrical Engineering, Linköping University.

[6] L. Ljung:
Prediction Error Identification Methods.
Report LiTH-ISY-I-139 Dept. of Electrical Engineering, Linköping University.

[7] T. Söderström, L. Ljung, I. Gustavsson:
A Comparative Study of Recursive Identification Methods.
Report TFRT-3085, Dept. of Automatic Control, Lund Institute of Technology, Sweden.

[8] T. Söderström, L. Ljung, I. Gustavsson:
On The Accuracy of Identification and the Design of Identification Experiments.
Report TFRT-3086, Dept. of Automatic Control, Lund Institute of Technology, Sweden.

[9] T. Söderström:
An On-line Algorithm for Approximate Maximum Likelihood Identification of Linear Dynamic Systems.
Report TFRT-3052, Dept. of Automatic Control, Lund Institute of Technology, Sweden.

[10] K. J. Åström, P. Eykhoff:
System Identification - A Survey.
Automatica 7 1971 p. 123-162.

[11] K. J. Åström:
Lectures on the Identification Problem
The Least Squares Method.
Report TFRT-3004, Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[12] K. Y. Wong, E. Polak:
Identification of Linear Discrete Time Systems
Using an Instrumental Variable Method.
IEEE Trans. AC-12 1967 p. 707-719.

[13] P. C. Young:
An Instrumental Variable Method for Real-time
Identification of a Noisy Process.
Automatica 6 1970 p. 271-287.

[14] H. Johannesson, J. O. Wesström:
Jämförelser af några Rekursiva Identifierings-
metoder.
Report TFRT-**** Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[15] H. Elmquist:
SIMNON - An Interactive Simulation Program for
Nonlinear Systems. User's Manual.
Report TFRT-3091, Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[16] J. Wieslander:
IDPAC - Commands, User's Guide.
Report TFRT-3157, Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[17] J. Wieslander:
Interactive Programs - General Guide.
Report TFRT-3156, Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[18] I. Gustavsson:
User's Guide for a Program Package for
Simulation of Self-Tuning Regulators.
Report TFRT-7149, Dept. of Automatic Control,
Lund Institute of Technology, Sweden.

[19] K. J. Åström:
Introduction to stochasic control theory
Academic Press 1970

[20] K. J. Åström:
Reglerteori
Almquist & Wiksell Uppsala 1968

[21] K. J. Åström:
Reglerteknik - Samplade system
TLTH/VBV 1971

[22] K. J. Åström:
Reglerteknik - Stokastiska system
TLTH/VBV 1971

Appendix A

Quick Reference Program Guide and Connecting Systems

```
Common Global Variables:
IYEAR        ( 0)      :
MONTH        ( 0)      :
IDAY         ( 0)      :
ISIMN        ( 0)      : Simulation nr.
IDOC         ( 0)      : =1 docum.page =2 no print loss func.
LUN1         ( 5)      : Logical unit nr. for RK
LUN2         ( 7)      :              "
LUN3         ( 3)      :              "


Subsystem INPUT:
Global Variable:
FILN         (  )      : Data file name

Outputs:
E            ( 0)      : White Noise Output  (NTYP=1)
PRBS         ( 0)      : Pseudo Random Binary Sequence (NTYP=2)
U1           ( 0)      : Output from column NC1 in FILN
U2           ( 0)      :   "       "      "   NC2     "

Parameters:
NTYP         ( 0)      : Type of active output
IODD         (95)      : Start nr. for random number gen.
SD           (1.0)     : Standard deviation of E
NBIT         ( 7)      : Nr. of bits in shift register
AMP          (1.0)     : Amplitude of PRBS
KNEP         ( 0)      : (=0 No knep ; =1 FOA-knep)
ISTAR        ( 1)      : Starting point in the seq. (1,2,3,4)
IBP          ( 1)      : Basic Period
NC1          ( 1)      : Column nr. for U1
NC2          ( 0)      :    "        "  U2
DT           (1.0)     : Sampling period


Subsystem SYSHG:
Global variables:
FNRS         (  )      : (Det.) File name pulse responses
FNRN         (  )      : (Noise)  -    -    -       -

Input:
U            (  )      :

Output:
Y            (  )      :

Parameters:
NAS          ( 0)      : Order of A-Polynomial
NFS          ( 0)      :   -     - F      -
NBS          ( 0)      :   -     - B      -
NDS          ( 0)      :   -     - D      -
NCS          ( 0)      :   -     - C      -
KDS          ( 0)      : Time delay
```

```
YLEV        ( 0)      : Constant level added to output

THS1        ( 0)      : The parameter vector of SYSHG
THS2          -       :            -
 :            -       :            -
 :            -       :            -
THS30         -       :            -

NODD        (19)      : Starting value for random number gen.
LAMB        (1.0)     : Standard deviation
NIMB        ( 0)      : =1 Generate pulse resp. on files
DT          ( 0)      : Sampling period
```

<u>Subsystem IDHG:</u>
<u>Global variables:</u>

```
FNRS        (   )     : File name ref. syst pulse resp.
FNRN        (   )     :   -    -   -   noise   -    -
FNIM        (   )     :   -    -  gen. pulse responses
FNST        (   )     :   -    -  statistics file
```

<u>Outputs:</u>

```
TH1         (   )     : Vector of parameter estimates
TH2         (   )     :    -    -    -       -
 :                    :
 :                    :
TH10        (   )     :    -    -    -       -
```

<u>Inputs:</u>

```
Y           (   )     : Output signal from syst to be ident.
U           (   )     : Input signal to syst to be identified
```

<u>Parameters:</u>

```
METOD       ( 0)      : Id. Method =1 SA-LIP, =2 SA-CMA,
                      : =3 LIP, =4 CMA, =5 IV, =7 Ex-Kalman
NA          ( 0)      : Order of A-Polynomial
NF          ( 0)      :   -    - F     -
NB          ( 0)      :   -    - B     -
ND          ( 0)      :   -    - D     -
NC          ( 0)      :   -    - C     -
KD          ( 0)      : Time delay

WTI         (1.0)     : Initial value Forgetting profile
WTO         (1.0)     : Forgetting profile parameter
DELTA       (0.0)     : Adjustment to improve stability
R11         (0.0)     :             -
R12         (0.0)     :             -
 :            :       :             :
 :            :       :             :
R110        (0.0)     :             -

IPRER       ( 0)      : =0 a priori/=1 a posteriori est.res.
```

```
RLIM       (-1.0)     : >0 if |ε|>RLIM => ε=sign(ε)*RLIM
                      : <0 no limitation.
IFIST      ( 0)       : Starting time for IV or filtering CMA
ISTAB      ( 0)       : Stability check =0 no check
                      :                 =1 INPOL
                      :                 =2 SPFZN
INVAU      ( 0)       : IV ≤0 see sec.2.5, >0 old input sig.

THO1       (0.0)      : Initial values - parameter estimates.
THO2       (0.0)      :                -
 :           :        :                :
 :           :        :                :
THO10      (0.0)      :                -

PO1        (100.0)    : Initial values for diagonal elements
PO2        (100.0)    : in the covariance matrix.
 :            :       :                :
 :            :       :                :
PO10       (100.0)    :                -
```

<u>Print-out control parameters:</u>

```
IWRT       (-4)       : ≤0 No print out
  IWRT1    ( 0)       : =1 ISAMP, TH for ISAMP≤IWRT1 and
  IWRTN    (100)      :    every IWRTN:th sampling event
                      : =2 (1) + U,Y,RES,WRES
                      : =3 (2) + P
                      : =4 (3) + φ,ψ,Z
                      : =5 (4) + S
                      : =6 (5) + TEE
IHAT       (-19)      : <0 No loss function computed
                      : =N Loss functions computed for
                      :    Isamp=N and every IHATN:th sampl-
IHATN      (500)      :    ing event thereafter.
IFIGE      ( 0)       : =0 No file generated
                      : =1 FNIM, =2 FNST, =3 FNIM+FNST
DT         (1.0)      : Sampling period
```

<u>Variables:</u>

```
RES                   : The residual see sec.2 and 5.3.5
WRES                  : Weighted
V                     : V = Σ ε
WV                    : WV = Σ θε
WT                    : The forgetting profile
```

<u>Subsystem AUXIL:</u>

```
ISUB       ( 0)       : =1 GENHG, =2 SIMHG, =3 STAT
                      : =4 FILCON,=5 VSTAT, =6 TAFLA
```

Connecting Systems:

```
CONNECTING SYSTEM TENG1
TIME T
U[SYSHG]=PRBS[INPUT]
U[IDHG]=PRBS[INPUT]
Y[IDHG]=Y[SYSHG]
END
```

```
CONNECTING SYSTEM TENG2
TIME T
U[IDHG]=U1[INPUT]
Y[IDHG]=U2[INPUT]
END
```

```
CONNECTING SYSTEM TENG3
TIME T
U[SYSHG]=E[INPUT]
Y[IDHG]=Y[SYSHG]
U[IDHG]=E[INPUT]
END
```

```
CONNECTING SYSTEM TENG4
TIME T
U[SYSHG]=PRBS[INPUT]
END
```

```
CONNECTING SYSTEM TENG5
TIME T
U[SYSHG]=PRBS[INPUT]
END
```

```
CONNECTING SYSTEM TENG6
TIME T
U[SYSHG]=U1[INPUT]
END
```

Appendix B

Examples of results written by the programs STAT and VSTAT.

Appendix B

SUBROUTINE STAT:

$V=SUM[HR(I)-HI(I)]^2$ $VR=V/SUM[HR(I)]^2$ $(I=1,100)$ $VE=E[RES^2]$

$VWE=E[WRES^2]$ VS=V[SYS] VRS=VR[SYS] VN=V[NOISE] VRN=VR[NOISE]

STATISTICS OF THE FOLLOWING REALIZATIONS:

FILE NAME : M5311

N	SIMULATION NO.	DATE
1	31	1980- 6-24
2	32	1980- 6-24
3	33	1980- 6-24
4	34	1980- 6-24
5	35	1980- 6-24
6	36	1980- 6-24
7	37	1980- 6-24
8	38	1980- 6-24
9	39	1980- 6-24
10	40	1980- 6-24

TIME= 100

PAR/LOSSF.	MEAN	DEVIATION
F 1	-1.30733	0.197750
F 2	0.415983	0.187965
B 1	0.509322E-01	0.304267E-01
B 2	0.113764	0.574948E-01
D 1	-1.20073	0.212940
D 2	0.283191	0.185388
C 1	-0.358316	0.220122
C 2	0.331164E-01	0.135719
VE	0.505060E-01	0.107565E-01
VWE	0.454062E-01	0.853462E-02
VS	0.255020E-01	0.143052E-01
VRS	0.105078	0.589429E-01
VN	0.823834	0.842092
VRN	0.297340	0.303930

TIME= 500

PAR/LOSSF.	MEAN	DEVIATION
F 1	-1.53628	0.863886E-01
F 2	0.623347	0.813888E-01
B 1	0.720632E-01	0.176585E-01
B 2	0.487745E-01	0.300331E-01
D 1	-1.17588	0.180391
D 2	0.249185	0.164146
C 1	-0.461108	0.165532
C 2	0.570931E-01	0.941346E-01
VE	0.382979E-01	0.397019E-02
VWE	0.372779E-01	0.350534E-02
VS	0.945862E-02	0.526059E-02
VRS	0.389731E-01	0.216757E-01
VN	0.255131	0.346679
VRN	0.920826E-01	0.125124

Appendix B

STAT CONT. FILE NAME : M5311

TIME= 1000

PAR/LOSSF.	MEAN	DEVIATION
F 1	-1.57780	0.704079E-01
F 2	0.661594	0.644192E-01
B 1	0.709215E-01	0.719614E-02
B 2	0.422132E-01	0.202452E-01
D 1	-1.11980	0.233810
D 2	0.190718	0.218149
C 1	-0.460455	0.218381
C 2	0.571035E-01	0.729977E-01
VE	0.356013E-01	0.283891E-02
VWE	0.350913E-01	0.264082E-02
VS	0.643135E-02	0.289343E-02
VRS	0.264996E-01	0.119220E-01
VN	0.139210	0.917792E-01
VRN	0.502441E-01	0.331252E-01

TIME= 1500

PAR/LOSSF.	MEAN	DEVIATION
F 1	-1.59065	0.469183E-01
F 2	0.672433	0.428574E-01
B 1	0.705198E-01	0.530450E-02
B 2	0.427313E-01	0.138014E-01
D 1	-1.08798	0.243221
D 2	0.164420	0.222304
C 1	-0.455173	0.216840
C 2	0.584994E-01	0.712548E-01
VE	0.345913E-01	0.275184E-02
VWE	0.342513E-01	0.262396E-02
VS	0.559344E-02	0.194018E-02
VRS	0.230471E-01	0.799430E-02
VN	0.923021E-01	0.744701E-01
VRN	0.333139E-01	0.268780E-01

TIME= 2000

PAR/LOSSF.	MEAN	DEVIATION
F 1	-1.59485	0.347953E-01
F 2	0.677206	0.316523E-01
B 1	0.713414E-01	0.533213E-02
B 2	0.401507E-01	0.122907E-01
D 1	-1.05635	0.231711
D 2	0.135608	0.211587
C 1	-0.438691	0.215219
C 2	0.586794E-01	0.710751E-01
VE	0.341690E-01	0.284288E-02
VWE	0.339140E-01	0.275433E-02
VS	0.511946E-02	0.122242E-02
VRS	0.210941E-01	0.503682E-02
VN	0.710221E-01	0.513904E-01
VRN	0.256335E-01	0.185480E-01

Appendix B

SUBROUTINE VSTAT:

$$V = SUM[HR(I)-HI(I)]^2 / SUM[HR(I)]^2 \qquad I=1,100$$

$$VE=E[RES^2] \qquad VRS=V[SYS] \qquad VRN=V[NOISE]$$

RESULTS OF THE FOLLOWING REALIZATIONS:

FILE NAME : M5311

N	SIMULATION NO.	DATE
1	31	1980- 6-24
2	32	1980- 6-24
3	33	1980- 6-24
4	34	1980- 6-24
5	35	1980- 6-24
6	36	1980- 6-24
7	37	1980- 6-24
8	38	1980- 6-24
9	39	1980- 6-24
10	40	1980- 6-24

IDENTIFICATION METHOD: 3 LIP

NA	=	0	IPRER=		1	IHAT =		100
NF	=	2	IFIST=		0	IHATN=		500
NB	=	2	ISTAB=		1	FNRS =		SIMPU
ND	=	2	INVAU=		0	FNRN =		NIMPU
NC	=	2	NFST =		0			
KD	=	0				DT	=	1.00000

WTI	=	1.00000	PFST	=	100.000	RLIM	=	-1.00000
WTO	=	1.00000	RKJ	=	0.000000	DELTA	=	0.000000

R1(1)	=	0.000000	THO(1)	=	0.000000	PO(1)	=	100.000
R1(2)	=	0.000000	THO(2)	=	0.000000	PO(2)	=	100.000
R1(3)	=	0.000000	THO(3)	=	0.000000	PO(3)	=	100.000
R1(4)	=	0.000000	THO(4)	=	0.000000	PO(4)	=	100.000
R1(5)	=	0.000000	THO(5)	=	0.000000	PO(5)	=	100.000
R1(6)	=	0.000000	THO(6)	=	0.000000	PO(6)	=	100.000
R1(7)	=	0.000000	THO(7)	=	0.000000	PO(7)	=	100.000
R1(8)	=	0.000000	THO(8)	=	0.000000	PO(8)	=	100.000

Appendix B

VSTAT CONT. FILE NAME : M5311

ISAMP	VE	VRS	VRN
100	0.379149E-01	0.885315E-01	0.227717
	0.398252E-01	0.773847E-01	1.05037
	0.577382E-01	0.136031	0.362052E-01
	0.492936E-01	0.848578E-01	0.349230
	0.406987E-01	0.378401E-01	0.777729E-01
	0.418780E-01	0.353598E-01	0.535841
	0.540512E-01	0.188850	0.238472
	0.723748E-01	0.192105	0.219977
	0.562466E-01	0.154345	0.190863
	0.550382E-01	0.554751E-01	0.469529E-01
MEAN:	0.505060E-01	0.105078	0.297340
DEV.:	0.107565E-01	0.589429E-01	0.303930
MIN :	0.379149E-01	0.353598E-01	0.362052E-01
MAX :	0.723748E-01	0.192105	1.05037
500	0.345671E-01	0.467546E-01	0.520680E-01
	0.336682E-01	0.190453E-01	0.394994E-01
	0.398239E-01	0.782396E-01	0.366754E-01
	0.386905E-01	0.284312E-01	0.183273E-01
	0.356160E-01	0.182066E-01	0.137009
	0.328852E-01	0.328523E-01	0.528769E-01
	0.408277E-01	0.756825E-01	0.112774
	0.443278E-01	0.258524E-01	0.429570
	0.431458E-01	0.277929E-01	0.257980E-01
	0.394267E-01	0.368738E-01	0.162278E-01
MEAN:	0.382979E-01	0.389731E-01	0.920826E-01
DEV.:	0.397019E-02	0.216757E-01	0.125124
MIN :	0.328852E-01	0.182066E-01	0.162278E-01
MAX :	0.443278E-01	0.782396E-01	0.429570
2000	0.301492E-01	0.234013E-01	0.672433E-01
	0.324563E-01	0.188764E-01	0.220830E-01
	0.378375E-01	0.329022E-01	0.351846E-01
	0.345862E-01	0.201473E-01	0.138086E-01
	0.309058E-01	0.185628E-01	0.218020E-01
	0.324246E-01	0.188795E-01	0.175224E-01
	0.333129E-01	0.199933E-01	0.100177E-01
	0.354600E-01	0.159778E-01	0.446657E-01
	0.387924E-01	0.166813E-01	0.638594E-02
	0.357647E-01	0.255195E-01	0.176214E-01
MEAN:	0.341690E-01	0.210941E-01	0.256335E-01
DEV.:	0.284288E-02	0.503682E-02	0.185480E-01
MIN :	0.301492E-01	0.159778E-01	0.638594E-02
MAX :	0.387924E-01	0.329022E-01	0.672433E-01

Appendix C

Results

Model nr.	a) LIP	b) CMA IFIST=0	c) CMA IFIST=50
1	M1310		
2	M2310		
3	M3311	M3411	M34Z1
4	M4311	M4411	M44Z1
5	M5311	M5411	M54Z1
5A	M53X1	M54X1	X54Z1
6	M6311	M6411	M64Z1
6A	M63X1	M64X1	X64Z1
7	M7310	M7410	M74Z0
8	M8311	M8411	M84Z1
9	M9311	M9411	M94Z1
10	M0311	M0411	M04Z1
10A	M03X1	M04X1	X04Z1
11	MA311	MA411	MA4Z1
11A	MA3X1	MA4X1	XA4Z1

Appendix C
LIP

N = 100

SUBROUTINE TAFLA:

MODEL INF.	VE			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.111305	0,027233	0,077879	0,162200
M2310 1 0 0 4 0 4 0 0 0 3	0.039716	0,006416	0,028940	0,047733
M3311 1 1 0 0 4 4 0 0 0 3	0.211457	0,060118	0,143876	0,306597
M4311 1 1 0 3 0 3 0 2 0 3	0.042500	0,009090	0,030036	0,061082
M5311 1 1 0 0 2 2 2 2 0 3	0.050506	0,010757	0,037915	0,072375
M53X1 1 1 0 0 3 3 1 1 0 3	0.045819	0,007510	0,031211	0,055627
M6311 1 1 0 2 0 2 2 2 0 3	0.048322	0,012429	0,030775	0,077184
M63X1 1 1 0 3 0 3 1 1 0 3	0.042231	0,008430	0,029690	0,055056
M7310 1 0 0 3 0 3 2 0 0 3	0.039935	0,006676	0,029631	0,050454
M8311 1 1 0 0 3 3 2 0 0 3	0.045977	0,006389	0,033372	0,054364
M9311 1 1 0 2 2 2 1 1 0 3	0.047388	0,011394	0,034045	0,071693
M0311 1 1 0 2 2 2 0 2 0 3	0.050544	0,013766	0,030015	0,080776
M03X1 1 1 0 3 2 2 0 1 0 3	0.048257	0,016161	0,029900	0,084382
MA311 1 1 0 2 2 2 2 0 0 3	0.048771	0,010958	0,031570	0,065162
MA3X1 1 1 0 2 3 2 1 0 0 3	0.049969	0,008692	0,032409	0,060937

Appendix C
LIP

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRS MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.329782	0.057149	0.240806	0.430341
M2310 1 0 0 4 0 4 0 0 0 3	0.064643	0.026279	0.027798	0.121761
M3311 1 1 0 0 4 4 0 0 0 3	0.492981	0.200333	0.267540	0.816606
M4311 1 1 0 3 0 3 0 2 0 3	0.069574	0.022688	0.038160	0.101025
M5311 1 1 0 0 2 2 2 2 0 3	0.105078	0.058943	0.035360	0.192105
M53X1 1 1 0 0 3 3 1 1 0 3	0.087523	0.032505	0.032180	0.134521
M6311 1 1 0 2 0 2 2 2 0 3	0.089918	0.056917	0.023096	0.186319
M63X1 1 1 0 3 0 3 1 1 0 3	0.087406	0.033608	0.036683	0.146976
M7310 1 0 0 3 0 3 2 0 0 3	0.059665	0.018606	0.031913	0.097469
M8311 1 1 0 0 3 3 2 0 0 3	0.071887	0.020231	0.030790	0.108218
M9311 1 1 0 2 2 2 1 1 0 3	0.109148	0.059279	0.036376	0.209836
M0311 1 1 0 2 2 2 0 2 0 3	0.093600	0.050917	0.025361	0.183487
M03X1 1 1 0 3 2 2 0 1 0 3	0.059582	0.029057	0.022835	0.113301
MA311 1 1 0 2 2 2 2 0 0 3	0.121462	0.047408	0.062564	0.176397
MA3X1 1 1 0 2 3 2 1 0 0 3	0.127497	0.067423	0.039538	0.243551

Appendix C
LIP

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRN MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M2310 1 0 0 4 0 4 0 0 0 3	0.220147	0.199793	0.068930	0.755903
M3311 1 1 0 0 4 4 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M4311 1 1 0 3 0 3 0 2 0 3	0.252217	0.129275	0.082479	0.458521
M5311 1 1 0 0 2 2 2 2 0 3	0.297340	0.303930	0.036205	1.050372
M53X1 1 1 0 0 3 3 1 1 0 3	0.434263	1.083011	0.009889	3.510337
M6311 1 1 0 2 0 2 2 2 0 3	0.254156	0.153653	0.051275	0.491150
M63X1 1 1 0 3 0 3 1 1 0 3	0.338513	0.273357	0.064083	0.798403
M7310 1 0 0 3 0 3 2 0 0 3	0.182043	0.116152	0.084762	0.485481
M8311 1 1 0 0 3 3 2 0 0 3	0.238945	0.271585	0.004671	0.918683
M9311 1 1 0 2 2 2 1 1 0 3	0.351384	0.390467	0.104313	1.169338
M0311 1 1 0 2 2 2 0 2 0 3	0.290162	0.190503	0.125191	0.696684
M03X1 1 1 0 3 2 2 0 1 0 3	0.255216	0.141096	0.092219	0.594660
MA311 1 1 0 2 2 2 2 0 0 3	0.274539	0.283671	0.054667	0.946019
MA3X1 1 1 0 2 3 2 1 0 0 3	0.199604	0.093663	0.097581	0.377943

Appendix C
LIP

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VE MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.134504	0.017570	0.104002	0.162346
M2310 1 0 0 / 4 0 4 0 0 0 3	0.035549	0.002285	0.032349	0.039092
M3311 1 1 0 / 0 4 4 0 0 0 3	0.142643	0.022805	0.122349	0.192547
M4311 1 1 0 / 3 0 3 0 2 0 3	0.037306	0.003779	0.032562	0.043002
M5311 1 1 0 / 0 2 2 2 2 0 3	0.038298	0.003970	0.032885	0.044328
M53X1 1 1 0 / 0 3 3 1 1 0 3	0.036051	0.002547	0.032301	0.038948
M6311 1 1 0 / 2 0 2 2 2 0 3	0.038006	0.004115	0.032918	0.047236
M63X1 1 1 0 / 3 0 3 1 1 0 3	0.036840	0.003400	0.032420	0.042242
M7310 1 0 0 / 3 0 3 2 0 0 3	0.035398	0.002189	0.032470	0.038111
M8311 1 1 0 / 0 3 3 2 0 0 3	0.036130	0.002234	0.032601	0.038847
M9311 1 1 0 / 2 2 2 1 1 0 3	0.038888	0.003434	0.033750	0.045387
M0311 1 1 0 / 2 2 2 0 2 0 3	0.038912	0.004210	0.033049	0.047114
M03X1 1 1 0 / 3 2 2 0 1 0 3	0.038088	0.004686	0.032746	0.046882
MA311 1 1 0 / 2 2 2 2 0 0 3	0.039445	0.003480	0.033606	0.044632
MA3X1 1 1 0 / 2 3 2 1 0 0 3	0.041350	0.002757	0.034925	0.043779

Appendix C
LIP

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRS MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.229529	0.034624	0.193208	0.302544
M2310 1 0 0 / 4 0 4 0 0 0 3	0.041817	0.008812	0.029681	0.056316
M3311 1 1 0 / 0 4 4 0 0 0 3	0.096937	0.032098	0.047208	0.157270
M4311 1 1 0 / 3 0 3 0 2 0 3	0.042022	0.010021	0.028448	0.061283
M5311 1 1 0 / 0 2 2 2 2 0 3	0.038973	0.021676	0.018207	0.078240
M53X1 1 1 0 / 0 3 3 1 1 0 3	0.029563	0.006254	0.021240	0.039455
M6311 1 1 0 / 2 0 2 2 2 0 3	0.038910	0.019769	0.018215	0.080768
M63X1 1 1 0 / 3 0 3 1 1 0 3	0.036897	0.009759	0.027538	0.059562
M7310 1 0 0 / 3 0 3 2 0 0 3	0.030018	0.006898	0.019895	0.040518
M8311 1 1 0 / 0 3 3 2 0 0 3	0.028840	0.006206	0.020092	0.038461
M9311 1 1 0 / 2 2 2 1 1 0 3	0.048928	0.017253	0.023816	0.090970
M0311 1 1 0 / 2 2 2 0 2 0 3	0.041226	0.013310	0.020710	0.067229
M03X1 1 1 0 / 3 2 2 0 1 0 3	0.031503	0.009725	0.019828	0.045669
MA311 1 1 0 / 2 2 2 2 0 0 3	0.057923	0.018825	0.037783	0.093834
MA3X1 1 1 0 / 2 3 2 1 0 0 3	0.086652	0.045278	0.046869	0.189476

Appendix C
LIP

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFII A F B D C K M	VRN MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M2310 1 0 0 / 4 0 4 0 0 0 3	0.153121	0.044914	0.087114	0.228170
M3311 1 1 0 / 0 4 4 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M4311 1 1 0 / 3 0 3 0 2 0 3	0.139375	0.027264	0.091736	0.185004
M5311 1 1 0 / 0 2 2 2 2 0 3	0.092083	0.125124	0.016228	0.429570
M53X1 1 1 0 / 0 3 3 1 1 0 3	0.091717	0.110079	0.008324	0.296460
M6311 1 1 0 / 2 0 2 2 2 0 3	0.134095	0.042422	0.082921	0.237919
M63X1 1 1 0 / 3 0 3 1 1 0 3	0.145275	0.033460	0.089118	0.196695
M7310 1 0 0 / 3 0 3 2 0 0 3	0.149641	0.035565	0.097572	0.203278
M8311 1 1 0 / 0 3 3 2 0 0 3	0.072873	0.078343	0.009928	0.264316
M9311 1 1 0 / 2 2 2 1 1 0 3	0.133664	0.034609	0.078830	0.206440
M0311 1 1 0 / 2 2 2 0 2 0 3	0.126512	0.040605	0.070653	0.210337
M03X1 1 1 0 / 3 2 2 0 1 0 3	0.167922	0.032996	0.125682	0.208400
MA311 1 1 0 / 2 2 2 2 0 0 3	0.166425	0.053571	0.107309	0.242627
MA3X1 1 1 0 / 2 3 2 1 0 0 3	0.150186	0.043440	0.096929	0.239142

Appendix C
LIP

N = 2000

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VE MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.136381	0.005550	0.124904	0.142690
M2310 1 0 0 / 4 0 4 0 0 0 3	0.035390	0.002470	0.031554	0.039954
M3311 1 1 0 / 0 4 4 0 0 0 3	0.111815	0.007317	0.103311	0.125980
M4311 1 1 0 / 3 0 3 0 2 0 3	0.035433	0.002764	0.031034	0.040064
M5311 1 1 0 / 0 2 2 2 2 0 3	0.034169	0.002843	0.030149	0.038792
M53X1 1 1 0 / 0 3 3 1 1 0 3	0.033300	0.002464	0.029364	0.037920
M6311 1 1 0 / 2 0 2 2 2 0 3	0.035069	0.002742	0.031176	0.039355
M63X1 1 1 0 / 3 0 3 1 1 0 3	0.035197	0.002679	0.031099	0.039715
M7310 1 0 0 / 3 0 3 2 0 0 3	0.034161	0.002496	0.030680	0.038735
M8311 1 1 0 / 0 3 3 2 0 0 3	0.033348	0.002446	0.029444	0.037751
M9311 1 1 0 / 2 2 2 1 1 0 3	0.036300	0.002979	0.031658	0.040235
M0311 1 1 0 / 2 2 2 0 2 0 3	0.035428	0.002878	0.030634	0.039468
M03X1 1 1 0 / 3 2 2 0 1 0 3	0.035831	0.002919	0.031337	0.040726
MA311 1 1 0 / 2 2 2 2 0 0 3	0.036834	0.003066	0.031952	0.041688
MA3X1 1 1 0 / 2 3 2 1 0 0 3	0.038945	0.002994	0.033701	0.043386

Appendix C
LIP

N = 2000

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI	A F B D C K M	VRS MEAN	DEV	MIN	MAX
M1310 1 0 0	0 0 8 0 0 0 3	0.216658	0.014496	0.201383	0.240266
M2310 1 0 0	4 0 4 0 0 0 3	0.037914	0.003391	0.034509	0.045074
M3311 1 1 0	0 4 4 0 0 0 3	0.028089	0.006443	0.018858	0.037416
M4311 1 1 0	3 0 3 0 2 0 3	0.028354	0.005560	0.020777	0.036636
M5311 1 1 0	0 2 2 2 2 0 3	0.021094	0.005037	0.015978	0.032902
M53X1 1 1 0	0 3 3 1 1 0 3	0.018378	0.001609	0.015530	0.020414
M6311 1 1 0	2 0 2 2 2 0 3	0.026572	0.006880	0.018816	0.042028
M63X1 1 1 0	3 0 3 1 1 0 3	0.026638	0.005205	0.017691	0.034360
M7310 1 0 0	3 0 3 2 0 0 3	0.024059	0.002435	0.019752	0.028096
M8311 1 1 0	0 3 3 2 0 0 3	0.018169	0.001400	0.016085	0.020010
M9311 1 1 0	2 2 2 1 1 0 3	0.034048	0.013438	0.020741	0.064208
M0311 1 1 0	2 2 2 0 2 0 3	0.026153	0.005972	0.018137	0.036623
M03X1 1 1 0	3 2 2 0 1 0 3	0.027217	0.005253	0.019305	0.034918
MA311 1 1 0	2 2 2 2 0 0 3	0.043263	0.012773	0.032083	0.076615
MA3X1 1 1 0	2 3 2 1 0 0 3	0.068723	0.015842	0.051259	0.097686

Appendix C
LIP

N = 2000

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRN MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M2310 1 0 0 4 0 4 0 0 0 3	0.136972	0.019069	0.107930	0.164765
M3311 1 1 0 0 4 4 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M4311 1 1 0 3 0 3 0 2 0 3	0.138963	0.029385	0.107283	0.198614
M5311 1 1 0 0 2 2 2 2 0 3	0.025633	0.018548	0.006386	0.067243
M53X1 1 1 0 0 3 3 1 1 0 3	0.032484	0.017746	0.009872	0.065512
M6311 1 1 0 2 0 2 2 2 0 3	0.125524	0.032075	0.081595	0.189713
M63X1 1 1 0 3 0 3 1 1 0 3	0.153713	0.024167	0.117149	0.193510
M7310 1 0 0 3 0 3 2 0 0 3	0.127990	0.010612	0.107762	0.141059
M8311 1 1 0 0 3 3 2 0 0 3	0.024259	0.020191	0.006361	0.060787
M9311 1 1 0 2 2 2 1 1 0 3	0.153047	0.027990	0.085413	0.176051
M0311 1 1 0 2 2 2 0 2 0 3	0.129985	0.021767	0.086316	0.160080
M03X1 1 1 0 3 2 2 0 1 0 3	0.174157	0.018122	0.139944	0.194517
MA311 1 1 0 2 2 2 2 0 0 3	0.137797	0.024043	0.087222	0.172917
MA3X1 1 1 0 2 3 2 1 0 0 3	0.124443	0.028332	0.084159	0.173619

Appendix C
CMA # IFIST = 0

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VE MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.111305	0,027233	0,077879	0,162200
M2310 1 0 0 / 4 0 4 0 0 0 3	0.039716	0,006416	0,028940	0,047733
M3411 1 1 0 / 0 4 4 0 0 0 4	0.408921	0,304185	0,153259	1.092426
M4411 1 1 0 / 3 0 3 0 2 0 4	0.048938	0,014356	0,032133	0,078923
M5411 1 1 0 / 0 2 2 2 2 0 4	0.093508	0,084637	0,036811	0,329827
M54X1 1 1 0 / 0 3 3 1 1 0 4	0,061702	0,024534	0,032315	0,119092
M6411 1 1 0 / 2 0 2 2 2 0 4	0.060869	0,021109	0,033381	0,100290
M64X1 1 1 0 / 3 0 3 1 1 0 4	0.047638	0,011292	0,032232	0,064822
M7410 1 0 0 / 3 0 3 2 0 0 4	0.057502	0,043138	0,030522	0.174068
M8411 1 1 0 / 0 3 3 2 0 0 4	0,061263	0,016415	0,035827	0,095976
M9411 1 1 0 / 2 2 2 1 1 0 4	0,062127	0,019533	0,032631	0,101710
M0411 1 1 0 / 2 2 2 0 2 0 4	0.055359	0,017061	0,033386	0,095830
M04X1 1 1 0 / 3 2 2 0 1 0 4	0.050448	0,011843	0,031096	0,065271
MA411 1 1 0 / 2 2 2 2 0 0 4	0,065579	0,016532	0,036038	0,094287
MA4X1 1 1 0 / 2 3 2 1 0 0 4	0,063987	0,017376	0,043392	0,104065

Appendix C
CMA # IFIST = 0

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VRS MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.329782	0.057149	0.240806	0.430341
M2310 1 0 0 / 4 0 4 0 0 0 3	0.064643	0.026279	0.027798	0.121761
M3411 1 1 0 / 0 4 4 0 0 0 4	1.813042	3.529560	0.156047	11.763588
M4411 1 1 0 / 3 0 3 0 2 0 4	0.104607	0.073740	0.041646	0.291142
M5411 1 1 0 / 0 2 2 2 2 0 4	0.516593	0.360129	0.110924	1.050433
M54X1 1 1 0 / 0 3 3 1 1 0 4	0.386226	0.370210	0.081733	1.077089
M6411 1 1 0 / 2 0 2 2 2 0 4	0.380804	0.260138	0.112716	0.954265
M64X1 1 1 0 / 3 0 3 1 1 0 4	0.162934	0.096868	0.044161	0.362862
M7410 1 0 0 / 3 0 3 2 0 0 4	0.233993	0.394997	0.046735	1.339492
M8411 1 1 0 / 0 3 3 2 0 0 4	0.344679	0.265529	0.116303	0.989423
M9411 1 1 0 / 2 2 2 1 1 0 4	0.416150	0.354481	0.100072	1.060922
M0411 1 1 0 / 2 2 2 0 2 0 4	0.138961	0.085375	0.052932	0.355032
M04X1 1 1 0 / 3 2 2 0 1 0 4	0.089298	0.076736	0.038203	0.297891
MA411 1 1 0 / 2 2 2 2 0 0 4	0.265400	0.164697	0.086144	0.517488
MA4X1 1 1 0 / 2 3 2 1 0 0 4	0.330823	0.351683	0.066882	1.048697

Appendix C
CMA # IFIST = 0

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VRN MEAN	DEV	MIN	MAX
M1310 1 0 0 / 0 0 8 0 0 0 3	0.639077	0,000000	0,639077	0,639077
M2310 1 0 0 / 4 0 4 0 0 0 3	0.220147	0.199793	0.068930	0.755903
M3411 1 1 0 / 0 4 4 0 0 0 4	0.639077	0,000000	0.639077	0,639077
M4411 1 1 0 / 3 0 3 0 2 0 4	0.237319	0.123914	0.038734	0.484526
M5411 1 1 0 / 0 2 2 2 2 0 4	0.619228	0,613527	0.160091	2,298954
M54X1 1 1 0 / 0 3 3 1 1 0 4	1.251730	2,374412	0.018345	7,634084
M6411 1 1 0 / 2 0 2 2 2 0 4	1.222505	1,413747	0.076457	4,686330
M64X1 1 1 0 / 3 0 3 1 1 0 4	0.619293	0,998572	0.063483	3,311567
M7410 1 0 0 / 3 0 3 2 0 0 4	0.443888	0,491036	0.081296	1,356936
M8411 1 1 0 / 0 3 3 2 0 0 4	2.444547	5,036653	0.088637	16,407843
M9411 1 1 0 / 2 2 2 1 1 0 4	1.464022	2,098873	0.020641	5,321782
M0411 1 1 0 / 2 2 2 0 2 0 4	0.298629	0,189285	0.082988	0,712399
M04X1 1 1 0 / 3 2 2 0 1 0 4	0.426052	0,364131	0.097477	1,345556
MA411 1 1 0 / 2 2 2 2 0 0 4	4.780703	10,364870	0.120744	33,219567
MA4X1 1 1 0 / 2 3 2 1 0 0 4	6.563300	18,278546	0,075062	58,481945

Appendix C
CMA # IFIST = 0

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VE MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.134504	0.017570	0.104002	0.162346
M2310 1 0 0 4 0 4 0 0 0 3	0.035549	0.002285	0.032349	0.039092
M3411 1 1 0 0 4 4 0 0 0 4	0.227731	0.118988	0.123989	0.474693
M4411 1 1 0 3 0 3 0 2 0 4	0.039154	0.006079	0.033199	0.049613
M5411 1 1 0 0 2 2 2 2 0 4	0.058732	0.026996	0.034247	0.131570
M54X1 1 1 0 0 3 3 1 1 0 4	0.045934	0.014623	0.032573	0.081637
M6411 1 1 0 2 0 2 2 2 0 4	0.046528	0.008756	0.033558	0.062280
M64X1 1 1 0 3 0 3 1 1 0 4	0.040315	0.005588	0.032853	0.047900
M7410 1 0 0 3 0 3 2 0 0 4	0.043042	0.018509	0.032770	0.093270
M8411 1 1 0 0 3 3 2 0 0 4	0.045644	0.012260	0.034016	0.077529
M9411 1 1 0 2 2 2 1 1 0 4	0.047479	0.010779	0.034610	0.070407
M0411 1 1 0 2 2 2 0 2 0 4	0.040649	0.005854	0.033655	0.052727
M04X1 1 1 0 3 2 2 0 1 0 4	0.039047	0.004124	0.033010	0.044214
MA411 1 1 0 2 2 2 2 0 0 4	0.045459	0.005148	0.035874	0.052418
MA4X1 1 1 0 2 3 2 1 0 0 4	0.048349	0.010700	0.040339	0.075216

Appendix C
CMA # IFIST = 0

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRS MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.229529	0.034624	0.193208	0.302544
M2310 1 0 0 4 0 4 0 0 0 3	0.041817	0.008812	0.029681	0.056316
M3411 1 1 0 0 4 4 0 0 0 4	0.163075	0.207953	0.019571	0.680841
M4411 1 1 0 3 0 3 0 2 0 4	0.033954	0.011584	0.019718	0.062975
M5411 1 1 0 0 2 2 2 2 0 4	0.319106	0.352230	0.019373	1.001346
M54X1 1 1 0 0 3 3 1 1 0 4	0.166265	0.292146	0.025643	0.954460
M6411 1 1 0 2 0 2 2 2 0 4	0.138524	0.108739	0.036552	0.310070
M64X1 1 1 0 3 0 3 1 1 0 4	0.086480	0.052775	0.019186	0.154685
M7410 1 0 0 3 0 3 2 0 0 4	0.159670	0.335967	0.026041	1.096281
M8411 1 1 0 0 3 3 2 0 0 4	0.138209	0.287494	0.021854	0.954892
M9411 1 1 0 2 2 2 1 1 0 4	0.153386	0.073954	0.025431	0.270938
M0411 1 1 0 2 2 2 0 2 0 4	0.039940	0.025320	0.017761	0.103136
M04X1 1 1 0 3 2 2 0 1 0 4	0.029798	0.017604	0.017026	0.071446
MA411 1 1 0 2 2 2 2 0 0 4	0.120036	0.054979	0.028348	0.216146
MA4X1 1 1 0 2 3 2 1 0 0 4	0.199056	0.240233	0.062169	0.859039

Appendix C
CMA # IFIST = 0

N = 500

SUBROUTINE TAFLA:

MODEL INF.	VRN			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M2310 1 0 0 4 0 4 0 0 0 3	0.153121	0.044914	0.087114	0.228170
M3411 1 1 0 0 4 4 0 0 0 4	0.639077	0.000000	0.639077	0.639077
M4411 1 1 0 3 0 3 0 2 0 4	0.169720	0.025983	0.125577	0.199573
M5411 1 1 0 0 2 2 2 2 0 4	0.696735	0.793455	0.032226	2.584596
M54X1 1 1 0 0 3 3 1 1 0 4	0.348535	0.409058	0.010199	1.285505
M6411 1 1 0 2 0 2 2 2 0 4	0.899463	1.594644	0.090072	5.033643
M64X1 1 1 0 3 0 3 1 1 0 4	0.259013	0.232944	0.077212	0.840200
M7410 1 0 0 3 0 3 2 0 0 4	0.513677	0.855287	0.108424	2.784766
M8411 1 1 0 0 3 3 2 0 0 4	0.300780	0.352510	0.016853	1.211449
M9411 1 1 0 2 2 2 1 1 0 4	1.180834	1.756288	0.066687	5.308486
M0411 1 1 0 2 2 2 0 2 0 4	0.160445	0.076171	0.079742	0.325797
M04X1 1 1 0 3 2 2 0 1 0 4	0.189039	0.080230	0.072934	0.365490
MA411 1 1 0 2 2 2 2 0 0 4	0.511855	0.572982	0.063528	1.936498
MA4X1 1 1 0 2 3 2 1 0 0 4	0.722027	0.984287	0.066245	2.601852

Appendix C
CMA # IFIST = 0

N = 2000

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VE MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.136381	0,005550	0,124904	0,142690
M2310 1 0 0 4 0 4 0 0 0 3	0.035390	0,002470	0,031554	0,039954
M3411 1 1 0 0 4 4 0 0 0 4	0.137820	0,039680	0,106568	0,225857
M4411 1 1 0 3 0 3 0 2 0 4	0.035969	0,003981	0,030896	0,044941
M5411 1 1 0 0 2 2 2 2 0 4	0.045908	0,015674	0,030096	0,086168
M54X1 1 1 0 0 3 3 1 1 0 4	0.038267	0,008851	0,029627	0,057080
M6411 1 1 0 2 0 2 2 2 0 4	0.040037	0,006551	0,030758	0,050262
M64X1 1 1 0 3 0 3 1 1 0 4	0.037629	0,003670	0,033629	0,044434
M7410 1 0 0 3 0 3 2 0 0 4	0.039283	0,012589	0,030778	0,072881
M8411 1 1 0 0 3 3 2 0 0 4	0.038567	0,010958	0,029916	0,068426
M9411 1 1 0 2 2 2 1 1 0 4	0.040374	0,004310	0,031660	0,048503
M0411 1 1 0 2 2 2 0 2 0 4	0.036007	0,003598	0,031515	0,041160
M04X1 1 1 0 3 2 2 0 1 0 4	0.035998	0,002818	0,031157	0,040223
MA411 1 1 0 2 2 2 2 0 0 4	0,040120	0,003595	0,033807	0,045801
MA4X1 1 1 0 2 3 2 1 0 0 4	0,041519	0,005971	0,035641	0,053957

Appendix C
CMA # IFIST = 0

N = 2000

SUBROUTINE TAFLA:

MODEL INF.	VRS			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.216658	0,014496	0.201383	0,240266
M2310 1 0 0 4 0 4 0 0 0 3	0,037914	0,003391	0.034509	0,045074
M3411 1 1 0 0 4 4 0 0 0 4	0.029623	0,012747	0.014453	0,057828
M4411 1 1 0 3 0 3 0 2 0 4	0,022724	0,003766	0,018744	0,028999
M5411 1 1 0 0 2 2 2 2 0 4	0.144571	0,267107	0.018688	0,864284
M54X1 1 1 0 0 3 3 1 1 0 4	0,045803	0,062863	0,018447	0,220174
M6411 1 1 0 2 0 2 2 2 0 4	0.051504	0,042902	0.020144	0.126416
M64X1 1 1 0 3 0 3 1 1 0 4	0.052326	0,031314	0,019025	0,088661
M7410 1 0 0 3 0 3 2 0 0 4	0.141541	0.315100	0.021705	1,023547
M8411 1 1 0 0 3 3 2 0 0 4	0.039399	0,058290	0,018321	0,205137
M9411 1 1 0 2 2 2 1 1 0 4	0.111850	0,055526	0,022020	0.210069
M0411 1 1 0 2 2 2 0 2 0 4	0,029207	0,022023	0,016989	0,090751
M04X1 1 1 0 3 2 2 0 1 0 4	0,027774	0,017532	0,016004	0,063125
MA411 1 1 0 2 2 2 2 0 0 4	0,086789	0,032692	0,033433	0,142156
MA4X1 1 1 0 2 3 2 1 0 0 4	0.078764	0,044511	0,025545	0,194619

Appendix C
CMA # IFIST = 0

N = 2000

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI A F B D C K M	VRN MEAN	DEV	MIN	MAX
M1310 1 0 0 0 0 8 0 0 0 3	0.639077	0.000000	0.639077	0.639077
M2310 1 0 0 4 0 4 0 0 0 3	0.136972	0.019069	0.107930	0.164765
M3411 1 1 0 0 4 4 0 0 0 4	0.639077	0.000000	0.639077	0.639077
M4411 1 1 0 3 0 3 0 2 0 4	0.148089	0.029485	0.118807	0.213277
M5411 1 1 0 0 2 2 2 2 0 4	0.580185	0.971431	0.021603	2.924808
M54X1 1 1 0 0 3 3 1 1 0 4	0.145081	0.188433	0.011220	0.614478
M6411 1 1 0 2 0 2 2 2 0 4	0.209498	0.146837	0.111681	0.548258
M64X1 1 1 0 3 0 3 1 1 0 4	0.160951	0.032825	0.126182	0.235066
M7410 1 0 0 3 0 3 2 0 0 4	0.449379	0.854657	0.117734	2.850376
M8411 1 1 0 0 3 3 2 0 0 4	0.203035	0.477979	0.007800	1.560137
M9411 1 1 0 2 2 2 1 1 0 4	0.290913	0.173616	0.065695	0.588783
M0411 1 1 0 2 2 2 0 2 0 4	0.118836	0.029603	0.059421	0.169149
M04X1 1 1 0 3 2 2 0 1 0 4	0.154887	0.050057	0.062746	0.221368
MA411 1 1 0 2 2 2 2 0 0 4	0.160244	0.091507	0.077017	0.375647
MA4X1 1 1 0 2 3 2 1 0 0 4	0.223494	0.244078	0.054598	0.719730

Appendix C
CMA # IFIST = 50

N = 100

SUBROUTINE TAFLA:

MODEL INF.	VE			
FN IPR IST IFII A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.345687	0.252056	0.156741	1.029811
M44Z1 1 1 50 3 0 3 0 2 0 4	0.042825	0.009140	0.030178	0.061591
M54Z1 1 1 50 0 2 2 2 2 0 4	0.050968	0.010463	0.038158	0.072094
X54Z1 1 1 50 0 3 3 1 1 0 4	0.046476	0.007784	0.031460	0.055471
M64Z1 1 1 50 2 0 2 2 2 0 4	0.048808	0.012177	0.031080	0.076179
X64Z1 1 1 50 3 0 3 1 1 0 4	0.042726	0.008041	0.030986	0.054557
M74Z0 1 1 50 3 0 3 2 0 0 4	0.040126	0.006725	0.029637	0.050658
M84Z1 1 1 50 0 3 3 2 0 0 4	0.046644	0.006593	0.033541	0.055380
M94Z1 1 1 50 2 2 2 1 1 0 4	0.048264	0.010797	0.034777	0.071172
M04Z1 1 1 50 2 2 2 0 2 0 4	0.050935	0.013779	0.030123	0.081138
X04Z1 1 1 50 3 2 2 0 1 0 4	0.048589	0.016214	0.029893	0.084963
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.049131	0.011082	0.031830	0.065455
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.049633	0.007930	0.033739	0.061507

Appendix C
CMA # IFIST = 50

N = 100

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VRS MEAN	DEV	MIN	MAX
M34Z1 1 1 50 / 0 4 4 0 0 0 4	4.516824	11.879302	0.266741	38.312447
M44Z1 1 1 50 / 3 0 3 0 2 0 4	0.068922	0.025278	0.033162	0.110355
M54Z1 1 1 50 / 0 2 2 2 2 0 4	0.105234	0.056418	0.022407	0.183155
X54Z1 1 1 50 / 0 3 3 1 1 0 4	0.085804	0.034904	0.046782	0.143760
M64Z1 1 1 50 / 2 0 2 2 2 0 4	0.092722	0.051389	0.036502	0.212120
X64Z1 1 1 50 / 3 0 3 1 1 0 4	0.090644	0.038141	0.039008	0.152616
M74Z0 1 1 50 / 3 0 3 2 0 0 4	0.060720	0.020307	0.028575	0.097069
M84Z1 1 1 50 / 0 3 3 2 0 0 4	0.072502	0.027654	0.024288	0.112870
M94Z1 1 1 50 / 2 2 2 1 1 0 4	0.152027	0.122180	0.038593	0.463090
M04Z1 1 1 50 / 2 2 2 0 2 0 4	0.098280	0.057192	0.028207	0.218662
X04Z1 1 1 50 / 3 2 2 0 1 0 4	0.062319	0.028522	0.024734	0.114927
MA4Z1 1 1 50 / 2 2 2 2 0 0 4	0.125158	0.040709	0.072302	0.170121
XA4Z1 1 1 50 / 2 3 2 1 0 0 4	0.113112	0.039853	0.053420	0.173590

Appendix C
CMA # IFIST = 50

N = 100

SUBROUTINE TAFLA:

MODEL INF.	VRN			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.639077	0.000000	0.639077	0.639077
M44Z1 1 1 50 3 0 3 0 2 0 4	0.245016	0.121941	0.102216	0.454445
M54Z1 1 1 50 0 2 2 2 2 0 4	0.224607	0.217364	0.048722	0.734498
X54Z1 1 1 50 0 3 3 1 1 0 4	0.278931	0.492977	0.018196	1.607566
M64Z1 1 1 50 2 0 2 2 2 0 4	0.208884	0.129933	0.057399	0.457301
X64Z1 1 1 50 3 0 3 1 1 0 4	0.256878	0.162208	0.063164	0.650561
M74Z0 1 1 50 3 0 3 2 0 0 4	0.184289	0.122297	0.083766	0.503028
M84Z1 1 1 50 0 3 3 2 0 0 4	0.509842	1.121060	0.004539	3.680140
M94Z1 1 1 50 2 2 2 1 1 0 4	0.692453	1.632071	0.088331	5.326800
M04Z1 1 1 50 2 2 2 0 2 0 4	0.249853	0.170328	0.078025	0.617256
X04Z1 1 1 50 3 2 2 0 1 0 4	0.265869	0.180019	0.109861	0.738064
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.249002	0.256989	0.062763	0.812273
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.738690	1.759630	0.072418	5.735079

Appendix C
CMA # IFIST = 50

N = 500

SUBROUTINE TAFLA:

MODEL INF.	VE			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.395696	0.706264	0.127958	2.404092
M44Z1 1 1 50 3 0 3 0 2 0 4	0.037167	0.003526	0.032891	0.041973
M54Z1 1 1 50 0 2 2 2 2 0 4	0.038012	0.003689	0.033260	0.043592
X54Z1 1 1 50 0 3 3 1 1 0 4	0.036219	0.002462	0.032281	0.039107
M64Z1 1 1 50 2 0 2 2 2 0 4	0.038171	0.004202	0.032905	0.045827
X64Z1 1 1 50 3 0 3 1 1 0 4	0.037421	0.002957	0.033480	0.041818
M74Z0 1 1 50 3 0 3 2 0 0 4	0.035573	0.002316	0.032611	0.038493
M84Z1 1 1 50 0 3 3 2 0 0 4	0.036339	0.002337	0.032763	0.039610
M94Z1 1 1 50 2 2 2 1 1 0 4	0.038609	0.002962	0.033811	0.043760
M04Z1 1 1 50 2 2 2 0 2 0 4	0.038589	0.003929	0.032991	0.046663
X04Z1 1 1 50 3 2 2 0 1 0 4	0.037796	0.004466	0.032857	0.046358
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.039717	0.003571	0.033714	0.044762
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.040520	0.002676	0.035031	0.043116

Appendix C
CMA # IFIST = 50

N = 500

SUBROUTINE TAFLA:

MODEL INF.	VRS			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.200082	0.306100	0.028330	1.048513
M44Z1 1 1 50 3 0 3 0 2 0 4	0.029651	0.007113	0.018443	0.042103
M54Z1 1 1 50 0 2 2 2 2 0 4	0.027673	0.011764	0.016939	0.053149
X54Z1 1 1 50 0 3 3 1 1 0 4	0.028745	0.009241	0.016621	0.045531
M64Z1 1 1 50 2 0 2 2 2 0 4	0.032818	0.020828	0.019026	0.089934
X64Z1 1 1 50 3 0 3 1 1 0 4	0.043129	0.019690	0.018336	0.078457
M74Z0 1 1 50 3 0 3 2 0 0 4	0.028298	0.005983	0.020357	0.038604
M84Z1 1 1 50 0 3 3 2 0 0 4	0.027538	0.009669	0.016443	0.047901
M94Z1 1 1 50 2 2 2 1 1 0 4	0.046003	0.029274	0.021160	0.112251
M04Z1 1 1 50 2 2 2 0 2 0 4	0.034521	0.019629	0.020542	0.087769
X04Z1 1 1 50 3 2 2 0 1 0 4	0.025769	0.011221	0.016964	0.055213
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.060768	0.019289	0.042743	0.108400
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.077755	0.026391	0.027701	0.129580

Appendix C
CMA # IFIST = 50

N = 500

SUBROUTINE TAFLA:

MODEL INF. FN IPR IST IFI / A F B D C K M	VRN MEAN	DEV	MIN	MAX
M34Z1 1 1 50 / 0 4 4 0 0 0 4	0.639077	0.000000	0.639077	0.639077
M44Z1 1 1 50 / 3 0 3 0 2 0 4	0.174880	0.030486	0.129388	0.204763
M54Z1 1 1 50 / 0 2 2 2 2 0 4	0.093397	0.089349	0.013322	0.271416
X54Z1 1 1 50 / 0 3 3 1 1 0 4	0.114951	0.139562	0.009060	0.340736
M64Z1 1 1 50 / 2 0 2 2 2 0 4	0.148065	0.041687	0.087634	0.212299
X64Z1 1 1 50 / 3 0 3 1 1 0 4	0.171512	0.055388	0.087033	0.245941
M74Z0 1 1 50 / 3 0 3 2 0 0 4	0.150056	0.031931	0.104040	0.191205
M84Z1 1 1 50 / 0 3 3 2 0 0 4	0.082665	0.082497	0.010149	0.245596
M94Z1 1 1 50 / 2 2 2 1 1 0 4	0.151857	0.046720	0.071427	0.225008
M04Z1 1 1 50 / 2 2 2 0 2 0 4	0.143264	0.038736	0.092389	0.208349
X04Z1 1 1 50 / 3 2 2 0 1 0 4	0.182264	0.040293	0.109921	0.238426
MA4Z1 1 1 50 / 2 2 2 2 0 0 4	0.153576	0.048111	0.088776	0.223753
XA4Z1 1 1 50 / 2 3 2 1 0 0 4	0.134189	0.053945	0.079032	0.220350

Appendix C
CMA # IFIST = 50

N = 2000

SUBROUTINE TAFLA:

MODEL INF.	VE			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.183211	0.195619	0.105191	0.738877
M44Z1 1 1 50 3 0 3 0 2 0 4	0.035070	0.002812	0.030710	0.040149
M54Z1 1 1 50 0 2 2 2 2 0 4	0.033844	0.002667	0.030088	0.038707
X54Z1 1 1 50 0 3 3 1 1 0 4	0.033314	0.002406	0.029494	0.037839
M64Z1 1 1 50 2 0 2 2 2 0 4	0.034899	0.002925	0.030889	0.040076
X64Z1 1 1 50 3 0 3 1 1 0 4	0.035833	0.002485	0.031371	0.040061
M74Z0 1 1 50 3 0 3 2 0 0 4	0.034273	0.002523	0.030749	0.038967
M84Z1 1 1 50 0 3 3 2 0 0 4	0.033359	0.002365	0.029624	0.037721
M94Z1 1 1 50 2 2 2 1 1 0 4	0.035811	0.002536	0.032816	0.039800
M04Z1 1 1 50 2 2 2 0 2 0 4	0.034992	0.002831	0.030640	0.038885
X04Z1 1 1 50 3 2 2 0 1 0 4	0.035381	0.002858	0.031113	0.040604
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.036922	0.003107	0.032049	0.041852
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.038020	0.002894	0.033695	0.042996

Appendix C
CMA # IFIST = 50

N = 2000

SUBROUTINE TAFLA:

MODEL INF.	VRS			
FN IPR IST IFI A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.029855	0.013428	0.016954	0.056691
M44Z1 1 1 50 3 0 3 0 2 0 4	0.021119	0.003034	0.017484	0.028715
M54Z1 1 1 50 0 2 2 2 2 0 4	0.019771	0.001712	0.017699	0.022254
X54Z1 1 1 50 0 3 3 1 1 0 4	0.020473	0.002716	0.016566	0.026515
M64Z1 1 1 50 2 0 2 2 2 0 4	0.021952	0.004128	0.018813	0.032851
X64Z1 1 1 50 3 0 3 1 1 0 4	0.031864	0.022247	0.018840	0.075172
M74Z0 1 1 50 3 0 3 2 0 0 4	0.023163	0.001874	0.020631	0.026838
M84Z1 1 1 50 0 3 3 2 0 0 4	0.019808	0.002252	0.016746	0.024462
M94Z1 1 1 50 2 2 2 1 1 0 4	0.029573	0.015327	0.019385	0.065491
M04Z1 1 1 50 2 2 2 0 2 0 4	0.024194	0.014156	0.018278	0.064312
X04Z1 1 1 50 3 2 2 0 1 0 4	0.020608	0.002762	0.016527	0.025827
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.045989	0.018530	0.035175	0.097624
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.062686	0.014483	0.022898	0.070963

Appendix C
CMA # IFIST = 50

N = 2000

SUBROUTINE TAFLA:

MODEL INF.	VRN			
FN IPR IST IFII A F B D C K M	MEAN	DEV	MIN	MAX
M34Z1 1 1 50 0 4 4 0 0 0 4	0.639077	0.000000	0.639077	0.639077
M44Z1 1 1 50 3 0 3 0 2 0 4	0.140407	0.025216	0.111819	0.173215
M54Z1 1 1 50 0 2 2 2 2 0 4	0.035887	0.028681	0.013051	0.093679
X54Z1 1 1 50 0 3 3 1 1 0 4	0.044351	0.033672	0.014319	0.109518
M64Z1 1 1 50 2 0 2 2 2 0 4	0.119367	0.014687	0.096449	0.143325
X64Z1 1 1 50 3 0 3 1 1 0 4	0.158821	0.014243	0.138203	0.187658
M74Z0 1 1 50 3 0 3 2 0 0 4	0.132286	0.010818	0.117392	0.148318
M84Z1 1 1 50 0 3 3 2 0 0 4	0.031971	0.033262	0.006254	0.097672
M94Z1 1 1 50 2 2 2 1 1 0 4	0.137582	0.039493	0.044946	0.167240
M04Z1 1 1 50 2 2 2 0 2 0 4	0.117595	0.015727	0.079464	0.134709
X04Z1 1 1 50 3 2 2 0 1 0 4	0.165906	0.013365	0.145344	0.188609
MA4Z1 1 1 50 2 2 2 2 0 0 4	0.130906	0.024297	0.071661	0.150932
XA4Z1 1 1 50 2 3 2 1 0 0 4	0.095182	0.027344	0.069003	0.154856

Program Listings

Ref.: H. Gunnarsson:
Recursive Identification using
a General Model Structure.
A Program Package and Simulations.

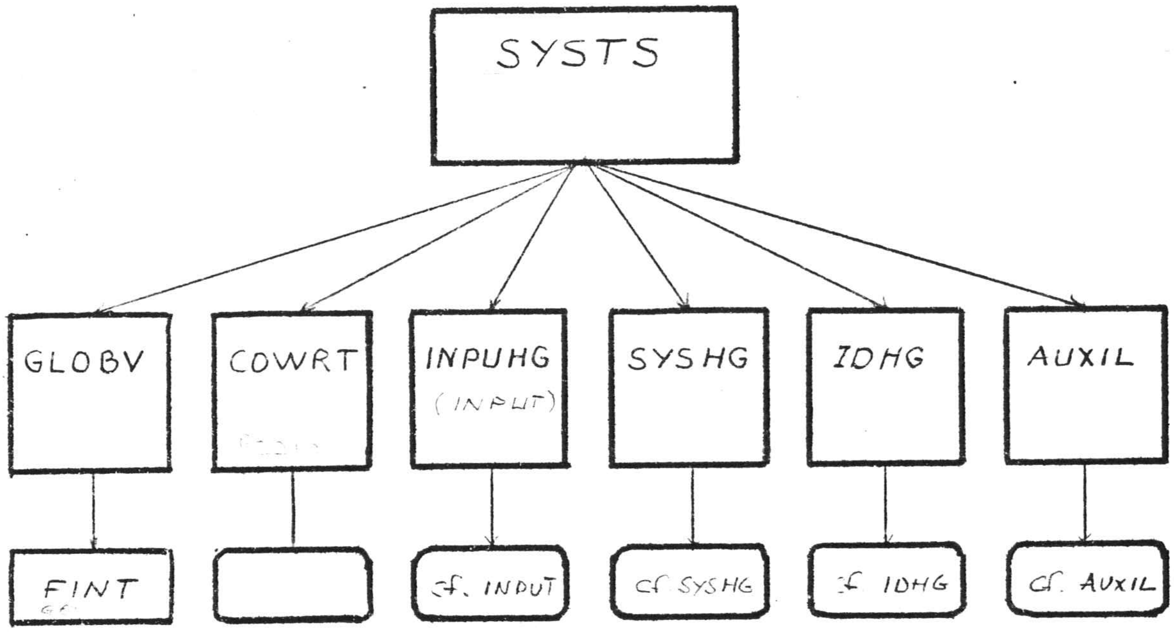
All programs are written in STANDARD FORTRAN with the exceptions given below. The following remarks may be of use for a prospective user.

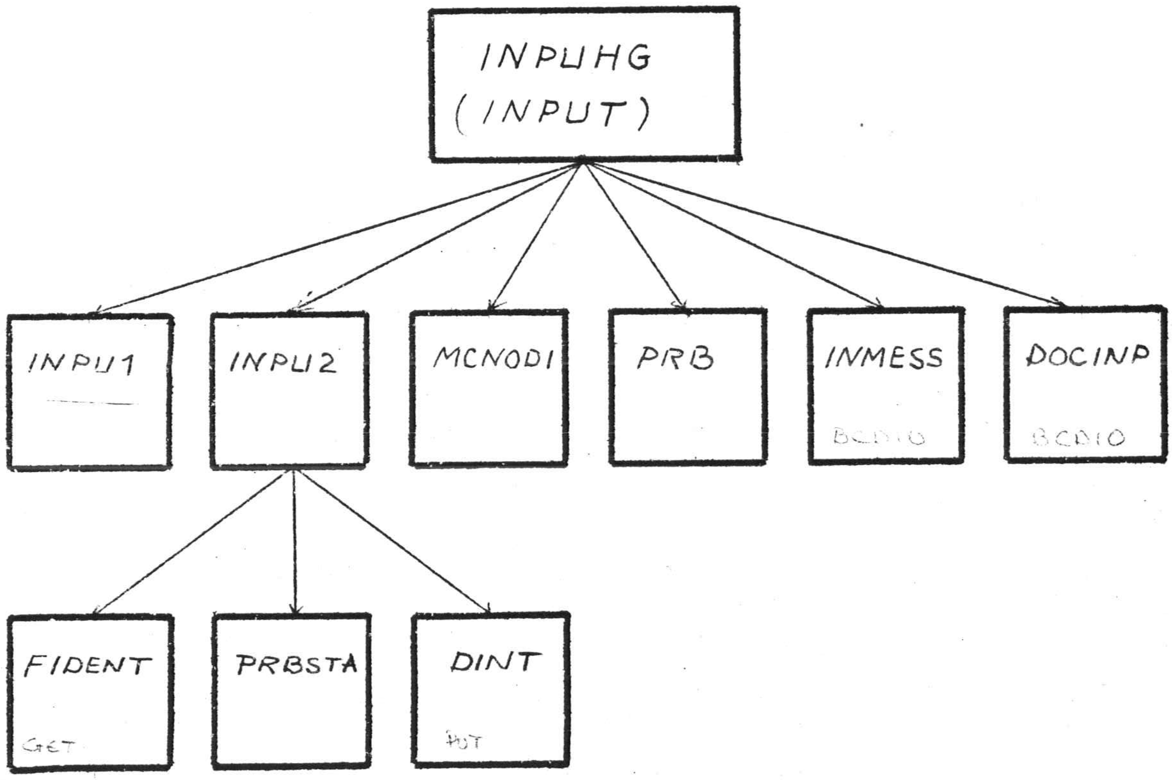
The library subroutine FILES (PDP-15) turned out to exist in more than one version so that the system routines FSTAT, SEEK, ENTER, CLOSE (and DLETE, RNAM) were used instead (cf. the PDP-manuals). These may have to be changed for use on other computers.

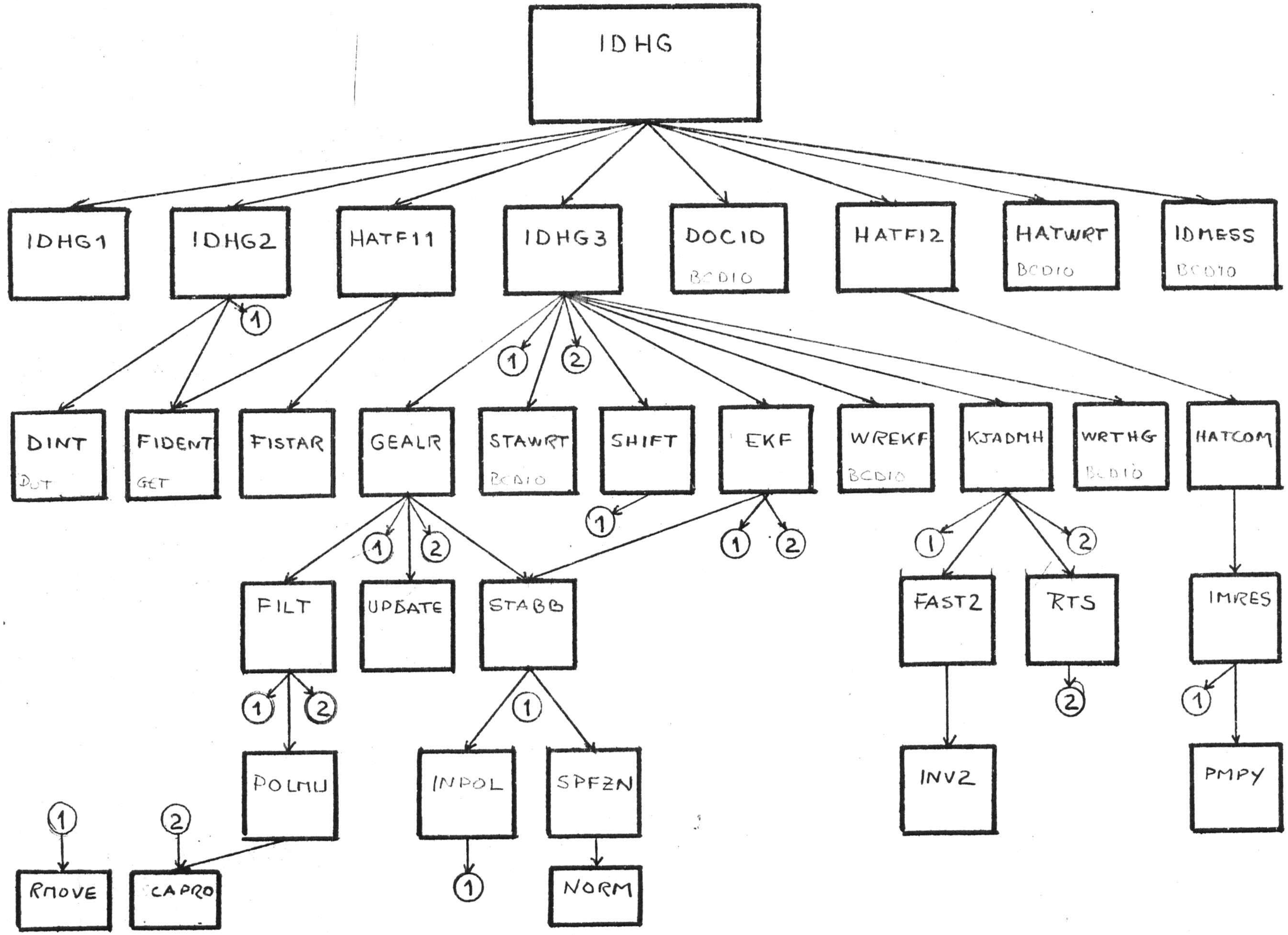
In the subsystem AUXIL the programs ask for input values. To get the cursor to stop at the end of the type-out on the terminal a special FORTRAN construction was used. May not work on other computers.

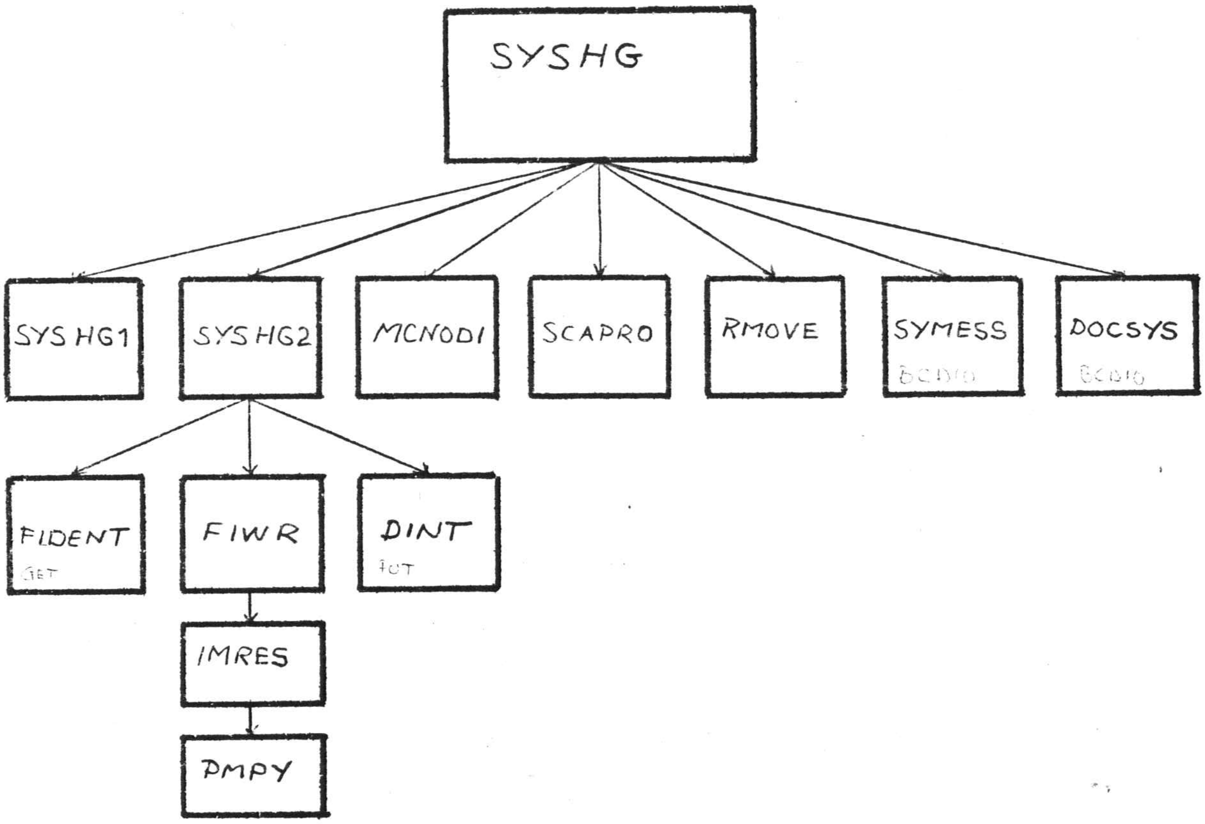
The library subroutines RMOVE, SCAPRO, POLMU, MCNODI, PRB and PRBSTA must be available or the programs must be changed accordingly. The SIMNON/INTRAC routines IDENT, INPUT, OUTPUT, PAR, PARV, TSAMP, FINT, DINT, FIDENT are assumed to function in the same manner as on the PDP-15.

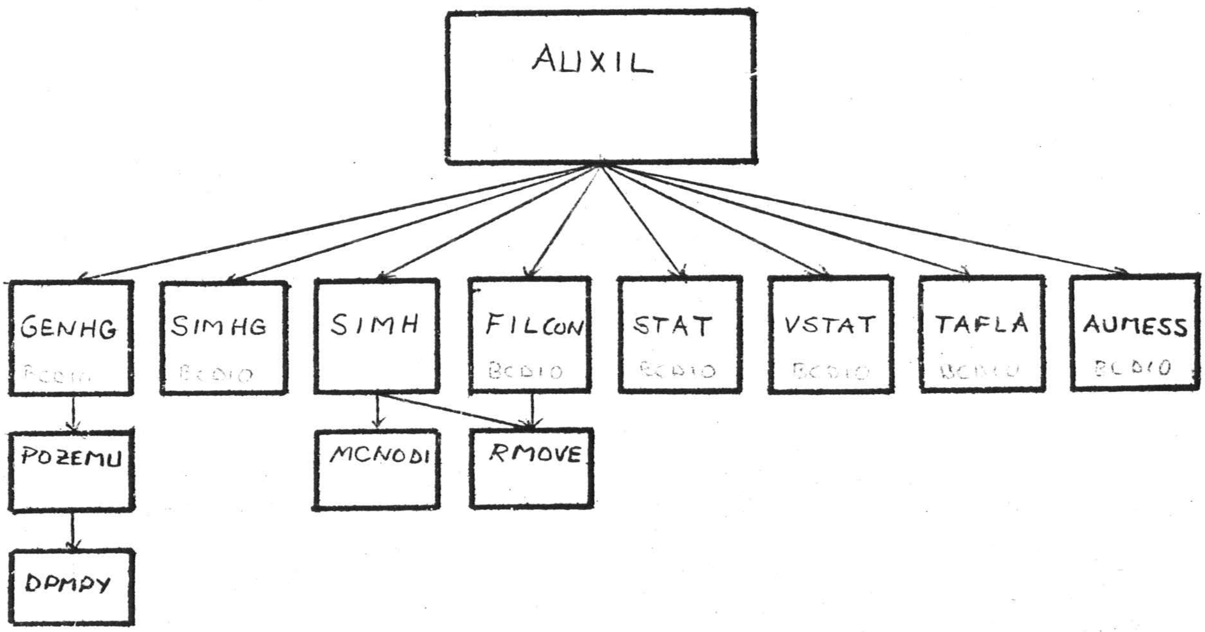
The figures below show the subroutine trees i.e. the program structure. May be helpful when designing an overlay core memory structure. Note that each subsystem is self contained i.e. with its own error routine and does not have to be run together with the other subsystems. No changes or only minor ones are needed to run the subsystems with other SIMNON subsystems written in FORTRAN viz. changes to avoid conflicts with the other programs in the use of blank COMMON.


SYSTS
GLOBV
COWRT
INPUHG
(INPUT)
SYSHG
IDHG
AUXIL
FINT
cf. INPUT
cf. SYSHG
cf. IDHG
cf. AUXIL


INPUHG
(INPUT)
INPU1
INPU2
MCNODI
PRB
INMESS
BCDIO
DOCINP
BCDIO
FIDENT
GET
PRBSTA
DINT
PUT


IDHG
IDHG1
IDHG2
HATF11
IDHG3
DOCID
BCDIO
HATF12
HATWRT
BCDIO
IDMESS
BCDIO
DINT
DUT
FIDENT
GET
FISTAR
GEALR
STAWRT
BCDIO
SHIFT
EKF
WREKF
BCDIO
KJADMH
WRTHG
BCDIO
HATCOM
FILT
UPDATE
STABB
FAST2
RTS
IMRES
POLMU
INPOL
SPFZN
INV2
PMPY
RMOVE
CAPRO
NORM


SYSHG
SYSHG1
SYSHG2
MCNODI
SCAPRO
RMOVE
SYMESS
DOCSYS
FIDENT
FIWR
DINT
IMRES
PMPY


AUXIL
GENHG
SIMHG
SIMH
FILCON
STAT
VSTAT
TAFLA
AUMESS
POZEMU
MCNODI
RMOVE
DPMPY

```
C        NAME: HGCOM                                NUMBER:
C        ----                                       ------
C
C        SUBTITLE: BLANK COMMON IN HGPAC
C        --------
C
C        LANGUAGE: STANDARD FORTRAN
C        --------
C
C        AUTHOR AND
C        IMPLEMENTOR: HALLGRIMUR GUNNARSSON         DATE: 1978-06-22
C        -----------                                ----
C
C        INSTITUTE:
C        ---------
C        DEPARTMENT OF AUTOMATIC CONTROL
C        LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C        ACCEPTED:                                  VERSION:
C        --------                                   -------
C
C-----------------------------------------------------------------------
C
         LOGICAL LSTOP,LDARK,MSTOP
         LOGICAL SA,IV,LFILT,PRERE,STABC,LSPFZ
C
         COMMON/DESTIN/IDUM,IPART
         COMMON/TIME/T
         COMMON/USER/LSTOP,LDARK
C
         COMMON LDK1,LDK2,KMESS,MSTOP,INHG,ISYSH,IDH,LINES,IDOC,IYEAR,
     *   MONTH,IDAY,ISIMN,LDK3,IDM2,IDM3,IDM4
C
         COMMON FNRS,FNRN,FNIM,FNST,FNWX,FNWY,IDM5,IDM6
C
         COMMON TYP,RIODD,RNBIT,RKNEP,RISTA,RNC1,RNC2,BPD,IBP,
     *   IXBP,K2,TSI,DTI,EI,SD,PRBS,U1,U2,W1(11),NC1,NC2,NTYP,IODD,
     *   NBIT,LA,IX,AMP,YI,FILN,ISTAR,KNEP,IDM13,IDM14
C
         COMMON RNFS,RNAS,RNBS,RNDS,RNCS,RNODD,RKDS,RIMR,IDM15,IDM16,
     *   IDM17,NIMR,NPMAX,KDMAX,IDM18,IDM19,IDM20,IDM21,US,YS,TSS,DTS,
     *   RLAMB,YLEV,THS(30),SPHI(30),OLDU(7),NFS,NAS,NBS,NDS,NCS,NST1,
     *   NST2,NST3,NST4,NST5,NPS1,NPS2,NPAR,NP1,KDS,KDX,NODD,V1,Z1,ES,Y1,
     *   IDM22,IDM23,IDM24,IDM25,IDM26
C
         COMMON IDM27,IDM28,IDM29,IDM30,RMETO,RKD,RSTAB,RPRER,RFIST,
     *   FIGE,UINVA,RNFST,RWRT,RWRT1,RWRTN,RHAT,RHATN,WTI,RNF,RNA,RNB,
     *   RND,RNC,PFST,THO(10),PO(10),TH10(10),P10(10),ISTAB,IPRER,IDM31,
     *   IDM32,IDM33,IDM34,METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     *   ICK,ISAMP,IFIST,SA,IV,LFILT,PRERE,STABC,LSPFZ,IDIM1,
     *   IDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,IST1,IST2,IST3,IST4,IST5,
     *   IPS1,IPS2,NPD1,INVAU,IDM35,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     *   DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *   PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *   P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
         COMMON RSUB
C
         STOP
         END
```

Main and Common Programs

```
C       NAME: SYSTS                                       NUMBER:
C       ----                                              ------
C
C       SUBTITLE: PROGRAM TO LINK HGPAC TO SIMNON
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON                DATE: 1978-06-22
C       -----------                                       ----
C
C       INSTITUTE:
C       ------ --
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                         VERSION:
C       --------                                          -------
C
C
C
C-----------------------------------------------------------------------
C
C
C
C       SUBROUTINES REQUIRED:
C       ---------------------
C
C       GLOBV   SYSHG   IDHG2   IMRES   STABB   STAT
C       COWRT   SYSHG1  IDHG3   WRTHG   EKF     FILCON
C       NAMES   SYSHG2  SHIFT   DOCID   KJADMH  AUMESS
C       INPUHG  SYMESS  HATFI1  IDMESS  AUXIL
C       INPU1   DOCSYS  HATFI2  STAWRT  GENHG
C       INPU2   FIWR    FISTAR  GEALR   POZEMU
C       INMESS  IDHG    HATCOM  FILT    SIMHG
C       DOCINP  IDHG1   HATWR   UPDATE   SIMH
C
C
C       SPFZN   INPOL   RTS     FAST2   INV2    PMPY    DPMPY
C
C
C
C
C       LIBRARY SUBROUTINES:
C       --------------------
C
C       FIDENT  FINT    DINT    RMOVE   SCAPRO
C       NORM    MCNODI  PRB     PRBSTA  GET     PUT
C
```

```
C
C
C       REF.
C       H.GUNNARSSON:   RECURSIVE IDENTIFICATION USING A
C                       GENERAL MODEL STRUCTURE. A PROGRAM
C                       PACKAGE AND SIMULATIONS.
C
C
C
C-----------------------------------------------------------------------
        SUBROUTINE SYSTS
C
        COMMON/DESTIN/ISYST,IPART
        COMMON/NSYSTS/NSYST
        COMMON/NBLCOM/IBLC
C
        COMMON IDM1(9),IYEAR
C
        NSYST=5
        IBLC=2505
C
C
        IF(IPART.EQ.1.OR.IPART.EQ.4) CALL GLOBV
        IF(IYEAR.EQ.-901) CALL COWRT
C
        GO TO(1,2,3,4,5),ISYST
C
    1   CALL INPUHG
        RETURN
C
    2   CALL SYSHG
        RETURN
C
    3   CALL IDHG
        RETURN
C
    4   CALL AUXIL
        RETURN
C
    5   CONTINUE
        RETURN
C
        END
```

```
C       NAME: GLOBV                                    NUMBER:
C       ----                                           ------
C
C       SUBTITLE: TO GET VALUES OF GLOBAL VARIABLES
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON             DATE: 1978-06-22
C       -----------                                    ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                      VERSION:
C       --------                                       -------
C
C------------------------------------------------------------------------
C
C       SUBROUTINES REQUIRED:         FINT         GET
C
C------------------------------------------------------------------------
C
        SUBROUTINE GLOBV
C
        LOGICAL LSTOP,MSTOP
C
        COMMON/USER/LSTOP
        COMMON LUN1,LUN2
        COMMON KMESS,MSTOP,KDUM(4),IDOC,IYEAR,MONTH,IDAY,ISIMAN,LUN3
C
        DATA RLU1,RLU2,RLU3/'LUN1','LUN2','LUN3'/
C
        CALL FINT(5HISIMN,5H     ,ISIMAN,IND1)
        IF(IND1.LE.0) GO TO 10
        ISIMAN=0
C
   10   CALL FINT(5HIYEAR,5H     ,IYEAR,IND1)
        IF(IND1.LE.0) GO TO 20
        IYEAR=0
C
   20   CALL FINT(5HMONTH,5H     ,MONTH,IND1)
        IF(IND1.LE.0) GO TO 30
        MONTH=0
C
   30   CALL FINT(4HIDAY,4H    ,IDAY,IND1)
```

```
      IF(IND1.LE.0) GO TO 40
      IDAY=0
C
   40 CALL FINT(4HLUN1,4H    ,LUN1,IND1)
      IF(IND1.LE.0) GO TO 45
      LUN1=5
      GO TO 50
   45 IF(LUN1.GE.1.AND.LUN1.LE.16) GO TO 50
      WRITE(9,1000) RLU1
      GO TO 99
C
   50 CALL FINT(4HLUN2,4H    ,LUN2,IND1)
      IF(IND1.LE.0) GO TO 55
      LUN2=7
   55 IF(LUN2.GE.1.AND.LUN2.LE.16.AND.LUN1.NE.LUN2) GO TO 60
      WRITE(9,1000) RLU2
      GO TO 99
C
   60 CALL FINT(4HLUN3,4H    ,LUN3,IND1)
      IF(IND1.LE.0) GO TO 65
      LUN3=3
   65 IF(LUN3.GE.1.AND.LUN3.LE.16.AND.LUN3.NE.LUN1.AND.LUN3.NE.LUN2)
     * GO TO 70
      WRITE(9,1000) RLU3
      GO TO 99
C
   70 CALL FINT(4HIDOC,4H    ,IDOC,IND1)
      IF(IND1.LE.0) GO TO 80
      IDOC=0
C
 1000 FORMAT(1X,A5,' HAS BAD VALUE')
C
C
   80 RETURN
C
   99 LSTOP=.TRUE.
      KMESS=100
      RETURN
      END
```

```
C       NAME: IMRES                                        NUMBER:
C       ----                                               -------
C
C       SUBTITLE:
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                          VERSION:
C       --------                                           -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE:
C       -------
C       GIVEN A PARAMETER VECTOR TH THE SUBROUTINE COMPUTES THE FIRST
C       100 IMPULSE RESPONSE COEFFICIENTS (VECTOR H) OF EITHER THE
C       DETERMINISTIC PART (ITYP=0) OR THE NOISE PART (ITYP=1)
C
C       PARAMETERS:
C       ----------
C       IMRES(ITYP,N,NSTART,TH,H)
C       TH        - VECTOR OF PARAMETER ESTIMATES
C                   (F,A,B,D,C)  CF.MODEL BELOW
C       N(1)=NF           NSTART(1)=1
C       N(2)=NA           NSTART(2)=1+NF
C       N(3)=NB           NSTART(3)=1+NF+NA
C       N(4)=ND           NSTART(4)=1+NF+NA+NB
C       N(5)=NC           NSTART(5)=1+NF+NA+NB+ND
C
C                         B          1+C
C       (1+F)*Y(T)=          *U(T) +     *BC(T)
C                        1+A         1+D
C
C                       ITYP=0      ITYP=1
C
```

```
C       SUBROUTINES REQUIRED:   PMPY    RMOVE
C
C-----------------------------------------------------------------------
C
        SUBROUTINE IMRES(ITYP,N,NSTART,TH,H)
C
        DIMENSION BC(30),FAD(30),U(130),Y(130),W1(30),X(30)
     *  ,N(1),NSTART(1),TH(1),H(1)
C
        K=2*ITYP
        NP=N(K+2)
        NN=N(K+3)
        NST1=NSTART(K+2)
        NST2=NSTART(K+3)
        FAD(1)=1.
        BC(1)=1.
        CALL RMOVE(TH(NST1),1,FAD(2),1,NP)
        CALL RMOVE(TH(NST2),1,BC(2),1,NN)
C
        IFAD=NP+1
        IBC=NN+1
C
        IF(N(1).EQ.0) GO TO 60
        W1(1)=1.
        NF=N(1)
        CALL RMOVE(TH(1),1,W1(2),1,NF)
        CALL PMPY(X,IX,FAD,IFAD,W1,NF+1)
        IFAD=IX
        NP=IFAD-1
        DO 10 I=1,IX
   10   FAD(I)=X(I)
C
C
C       INITIALIZATION
   60   DO 70 I=1,130
        U(I)=0.
   70   Y(I)=0.
        U(31)=1.
C
C       COMPUTATION OF THE FIRST 100 IMPULSE RESPONSE COEFFICIENTS
        DO 110 I=31,130
        IF(NP.LE.0) GO TO 90
        DO 80 J=1,NP
   80   Y(I)=Y(I)-FAD(J+1)*Y(I-J)
   90   DO 100 J=1,IBC
  100   Y(I)=Y(I)+BC(J+1-ITYP)*U(I-J+ITYP)
  110   H(I-30)=Y(I)
C
        RETURN
        END
```

```
C       NAME: PMPY                                     NUMBER:
C       ----                                           ------
C
C       SUBTITLE: MULTIPLIES TWO REAL POLYNOMIALS
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                      VERSION:
C       --------                                       -------
C
C
C
C
C-----------------------------------------------------------------------
C
C
        SUBROUTINE PMPY(Z,IDIMZ,X,IDIMX,Y,IDIMY)
C
C
        DIMENSION Z(1),X(1),Y(1)
C
        IF(IDIMX*IDIMY) 10,10,20
   10   IDIMZ=0
        GO TO 50
C
   20   IDIMZ=IDIMX+IDIMY-1
        DO 30 I=1,IDIMZ
   30   Z(I)=0.
C
        DO 40 I=1,IDIMX
        DO 40 J=1,IDIMY
        K=I+J-1
   40   Z(K)=X(I)*Y(J)+Z(K)
   50   RETURN
        END
```

Subsystem INPUT

```
C       NAME: INPUHG                                NUMBER:
C       ----                                        ------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE:
C       ------------
C       SIMNON STRUCTURED SUBROUTINE
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C-----------------------------------------------------------------------
C
C       PURPOSE:
C       =======
C       TO GENERATE, OR READ FROM A FILE, INPUT SIGNALS TO
C       OTHER SIMNON PROGRAMS.
C
C
C       THE PROGRAM GIVES DIFFERENT OUTPUTS, DEPENDING ON THE
C       PARAMETER NTYP
C
C       NTYP=1  - GIVES OUTPUT E AS WHITE NOISE WITH STANDARD
C                 DEVIATION SD.
C       NTYP=2  - GIVES OUTPUT PRBS AS PSEUDO RANDOM BINARY
C                 SEQUENCE
C       NTYP=3  - GIVES OUTPUT U1 (AND U2 OPT.) BY READING FROM
C                 A DATA FILE AT EVERY SAMPLING EVENT.
C
C
C       GLOBAL VARIABLES:   (DEFAULT VALUES [ ] )
C       ----------------
C       LDK1     - [5] LOGICAL UNIT NO.  (.DAT SLOT)
C       LDK2     - [7]            DO.
C       IYEAR    - [0]
C       MONTH    - [0]
C       IDAY     - [0]
C       IDOC     - [0] IF IDOC=1 A DOCUMENTATION PAGE IS PRINTED OUT
C                      AT THE END OF EVERY SIMULATION
C       NB.      THESE VARIBLES ARE SET OUTSIDE INPUHG BY A CALL
C                TO THE SUBROUTINE GLOBV FROM SYSTS
C       FILN     - THE NAME OF THE FILE TO BE READ
```

Subsystem INPUT

```
C
C       OUTPUTS:
C       --------
C       E         - WHITE NOISE OUTPUT
C       PRBS      - PSEUDO RANDOM BINARY SEQUENCE OUTPUT
C       U1        - OUTPUT READ FROM A DATA FILE  COLNR.=NC1
C       U2        - OUTPUT READ FROM A DATA FILE  COLNR.=NC2
C                   VALID ONLY IF NC2>0
C
C
C       SIMNON* PARAMETERS/PROGRAM VARIABLES   DEFAULT VALUE: [  ]
C       ------------------------------------
C
C       NTYP*     - [0] TYPE OF OUTPUT  (SEE ABOVE)
C
C       IODD*     - [95] ODD NUMBER,STARTING VALUE FOR THE NOISE GENERATOR
C       SD*       - [1.] STANDARD DEVIATION
C
C       IBP*      - [1] BASIC PERIOD
C       NBIT*     - [7] NUMBER OF BITS IN THE SHIFT REGISTER  RANGE (3,17)
C       AMP*      - [1.] AMPLITUDE OF THE OUTPUT SIGNAL (PRBS)
C       KNEP*     - [0]  1=FOA-KNEP  0=NO KNEP
C       ISTAR*    - [1] SPECIFIES A STARTING POINT IN THE SEQUENCE 1,2,3,4
C       LA        - THE FEEDBACK POLYNOMIAL
C       IX        - THE SHIFT REGISTER
C       Y         - OUTPUT FROM PRB
C
C       NC1*      - [1] COLUMN NUMBER IN DATA FILE FOR OUTPUT U1
C       NC2*      - [0] COLUMN NUMBER IN DATA FILE FOR OUTPUT U2
C       FILN      - THE NAME OF THE FILE TO BE READ
C       K2        - NUMBER OF COLUMNS IN FILE
C
C       DT*       - [1.]  THE SAMPLING INTERVAL
C       TS*       - TSAMP
C       W1        - WORK VECTOR                SIZE(11)
C
C
C-----------------------------------------------------------------------
C
C
C       CHARACTERISTICS:
C       ================
C
C       SUBROUTINES REQUIRED:           (LIBRARY)
C       ---------------------
C       INPU1                            FIDENT
C       INPU2                            DINT
C       INMESS                           MCNODI
C       DOCINP                           PRB
C                                        PRBSTA
C
C-----------------------------------------------------------------------
C
```

Subsystem INPUT

```
      SUBROUTINE INPUHG
C     =================
C
      LOGICAL LSTOP,MSTOP
C
      COMMON/DESTIN/IDUM,IPART
      COMMON/TIME/T
      COMMON/USER/LSTOP
C
      COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(4),IDOC,IDM2(38),IBP,IXBP,K2,
     * TS,DT,E,SD,PRBS,U1,U2,W1(11),NC1,NC2,NTYP,IODD,NBIT,LA,IX,AMP,Y
C
      GO TO(1,1,1,4,5,6,7,8),IPART
C-----------------------------------------------------------------------
C
C     IDENTIFICATION-DECLARATIONS-CONSTANT ASSIGNMENTS
C     ================================================
    1 CALL INPU1
      RETURN
C
C-----------------------------------------------------------------------
C
C     INITIAL
C     =======
    4 CALL INPU2
      RETURN
C
C-----------------------------------------------------------------------
C
C     OUTPUT
C     ======
    5 GO TO (51,52,53),NTYP
C
   51 CALL MCNODI(IODD,E)
      E=SD*E
      RETURN
C
   52 IXBP=IXBP-1
      IF(IXBP.GT.0) RETURN
      CALL PRB(NBIT,LA,IX,AMP,Y)
      PRBS=Y
      IXBP=IBP
      RETURN
C
   53 READ(LDK2,END=90) (W1(I),I=1,K2)
      U1=W1(NC1)
      IF(NC2.GT.0) U2=W1(NC2)
      RETURN
C
C-----------------------------------------------------------------------
C
C     DYNAMICS
C     ========
```

```
    6   TS=T+DT
        RETURN
C
C-----------------------------------------------------------------------------------
    7   RETURN
C-----------------------------------------------------------------------------------
C
C       FINAL COMPUTATIONS
C       ==================
    8   IF(NTYP.EQ.3) CALL CLOSE(LDK2)
        IF(MSTOP)CALL INMESS
        IF(KMESS.EQ.1.AND.IDOC.EQ.1) CALL DOCINP
        RETURN
C
C-----------------------------------------------------------------------------------
C
C
   90   KMESS=11
   91   MSTOP=.TRUE.
        LSTOP=.TRUE.
        RETURN
        END
```

Subsystem INPUT

```
C       NAME: INPU1                                 NUMBER:
C       ----                                        ------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE:
C       ------------
C       SIMNON STRUCTURED SUBROUTINE
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C
C-----------------------------------------------------------------------
C
C
C       PURPOSE:
C       =======
C       TO PERFORM PARTS 1,2 AND 3 OF INPUHG
C       CF. COMMENTS TO INPUHG
C
C
C-----------------------------------------------------------------------
C
C
        SUBROUTINE INPU1
C       ================
C
        LOGICAL LSTOP,MSTOP
C
        COMMON/DESTIN/IDUM,IPART
        COMMON/USER/LSTOP
C
        COMMON IDM1(2),KMESS,MSTOP,INHG,ISYSH,IDH,IDM2(24),TYP,RIODD,
     *  RNBIT,RKNEP,RISTA,RNC1,RNC2,BPD,IBP,IXBP,K2,TS,DT,E,SD,PRBS,
     *  U1,U2,IDM(27),LA,IX,AMP,Y
C
C
C
C
        GO TO(1,2,3,99,99,99,99,99),IPART
C-----------------------------------------------------------------------
```

Subsystem INPUT

```
C
C       IDENTIFICATION
C       ==============
    1   CALL IDENT(5HDISCR,5HINPUT)
        INHG=0
        ISYSH=0
        IDH=0
        RETURN
C
C-----------------------------------------------------------------------------
C       DECLARATIONS
C       ============
C
    2   CALL OUTPUT(E,4HE    )
        CALL OUTPUT(PRBS,4HPRBS)
        CALL OUTPUT(U1,4HU1  )
        CALL OUTPUT(U2,4HU2  )
C
        CALL PAR(TYP,4HNTYP)
C
        CALL PAR(RIODD,4HIODD)
        CALL PAR(SD,4HSD   )
C
        CALL PAR(BPD,4HIBP )
        CALL PAR(RNBIT,4HNBIT)
        CALL PAR(AMP,4HAMP )
        CALL PAR(RKNEP,4HKNEP)
        CALL PAR(RISTA,5HISTAR)
C
        CALL PAR(RNC1,4HNC1 )
        CALL PAR(RNC2,4HNC2 )
C
        CALL PAR(DT,4HDT   )
        CALL TSAMP(TS,4HTS   )
C
        RETURN
C
C-----------------------------------------------------------------------------
C
C       CONSTANT ASSIGNMENTS
C       ====================
C
    3   MSTOP=.FALSE.
        KMESS=1
C
        RIODD=95.
        SD=1.
        E=0.
C
        BPD=1.0
        PRBS=0.
        AMP=1.
        RKNEP=0.
```

Subsystem INPUT

```
      LA=0
      IX=0
      Y=0.
      RISTA=1.
      RNBIT=7.
C
      RNC1=1.
      RNC2=0.
      U1=0.
      U2=0.
C
      DT=1.
      INHG=1
C
      RETURN
C------------------------------------------------------------------------
   99 LSTOP=.TRUE.
      MSTOP=.TRUE.
      RETURN
      END
```

Subsystem INPUT

```
C       NAME: INPU2                                  NUMBER:
C       ----                                         ------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON           DATE: 1978-06-22
C       -----------                                  ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                    VERSION:
C       --------                                     -------
C
C
C
C-------------------------------------------------------------------------
C
C
C       PURPOSE:
C       =======
C       TO PERFORM PART 4 OF INPUHG
C       CF. COMMENTS TO INPUHG
C
C
C-------------------------------------------------------------------------
C
C
        SUBROUTINE INPU2
C       ================
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION FNAM(2)
C
        COMMON/USER/LSTOP
C
C
        COMMON LDK1,LDK2,KMESS,MSTOP,INHG,ISYSH,IDH,IDM1(5),ISIMN,
     *  IDM2(18),TYP,RIODD,RNBIT,RKNEP,RISTA,RNC1,RNC2,BPD,IBP,IXBP,
     *  K2,IDM4(26),KOD(10),NC1,NC2,NTYP,IODD,NBIT,LA,IX,AMP,Y,FILN,
     *  ISTAR,KNEP
C
C
        DATA FNAM(2)/' BIN'/
C
```

Subsystem INPUT

```
C-----------------------------------------------------------------------
C
C         INITIAL
C        =======
C
    4   MSTOP=.FALSE.
        KMESS=1
        IXBP=1
C
        NTYP=TYP+0.1
        IODD=RIODD+0.1
C
        IBP=BPD+0.1
        NBIT=RNBIT+0.1
        KNEP=RKNEP+0.1
        ISTAR=RISTA+0.1
C
        NC1=RNC1+0.1
        NC2=RNC2+0.1
C
C
        IF(NTYP.LT.1.OR.NTYP.GT.3) KMESS=2
        IF(MOD(IODD,2).EQ.0.OR.IODD.LT.1) KMESS=3
        IF(NBIT.LT.3.OR.NBIT.GT.17) KMESS=4
        IF(KNEP.LT.0.OR.KNEP.GT.1) KMESS=5
        IF(ISTAR.LT.1.OR.ISTAR.GT.4) KMESS=6
        IF(NC1.LT.1) KMESS=7
        IF(KMESS.GT.1) GO TO 99
C
C
        IF(NTYP.NE.3) GO TO 20
        CALL FIDENT('FILN',4H    ,FILN,IND1)
        IF(IND1.EQ.0) GO TO 5
        KMESS=9
        GO TO 99
C
    5   FNAM(1)=FILN
        CALL FSTAT(LDK2,FNAM,J)
        IF(J.EQ.0) GO TO 91
        CALL SEEK(LDK2,FNAM)
        READ(LDK2,END=90) (KOD(I),I=1,10)
        K2=KOD(2)
        IF(NC1.GT.11.OR.NC2.GT.11.OR.NC1.GT.K2.OR.NC2.GT.K2) KMESS=10
        IF(KMESS.GT.1) GO TO 99
C
C
   20   IF(NTYP.EQ.2) CALL PRBSTA(NBIT,LA,IX,AMP,Y,ISTAR,KNEP)
        IF(IDH.EQ.1.OR.ISYSH.EQ.1) RETURN
        ISIMN=ISIMN+1
        CALL DINT('ISIMN',4H    ,ISIMN,IND1)
        IF(IND1.GT.0) GO TO 92
        RETURN
C
```

Subsystem INPUT

```
C-----------------------------------------------------------------------
C
   90   KMESS=11
        GO TO 99
   91   KMESS=12
        GO TO 99
   92   KMESS=8
   99   MSTOP=.TRUE.
  100   LSTOP=.TRUE.
        RETURN
        END
```

Subsystem INPUT

```
C       NAME: INMESS                                NUMBER:
C       ----                                        ------
C
C       SUBTITLE: ERROR MESSAGES
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C
C-----------------------------------------------------------------------
C
C
C       GIVES ERROR MESSAGES FOR THE SYSTEM INPUT (INPUHG) WHEN
C       SIMULATIONS HAVE TO BE STOPPED
C
C
C-----------------------------------------------------------------------
C
C
        SUBROUTINE INMESS
C
        LOGICAL MSTOP
C
        COMMON IDM1(2),KMESS,MSTOP,IDM2(93),FILN
C
        IF(KMESS.GE.13) RETURN
C
C
        LU=9
        MSTOP=.FALSE.
C
        WRITE(LU,1000)
 1000   FORMAT(' SYSTEM INPUT')
C
        GO TO(1,2,3,4,5,6,7,8,9,10,11,12),KMESS
C
```

Subsystem INPUT

```
    1   WRITE(LU,101)
        GO TO 100
    2   WRITE(LU,102)
        GO TO 100
    3   WRITE(LU,103)
        GO TO 100
    4   WRITE(LU,104)
        GO TO 100
    5   WRITE(LU,105)
        GO TO 100
    6   WRITE(LU,106)
        GO TO 100
    7   WRITE(LU,107)
        GO TO 100
    8   WRITE(LU,108)
        GO TO 100
    9   WRITE(LU,109)
        GO TO 100
   10   WRITE(LU,110) FILN
        GO TO 100
   11   WRITE(LU,111) FILN
        GO TO 100
   12   WRITE(LU,112) FILN
        GO TO 100
  100   RETURN
C
C
  101   FORMAT(' HVERNIG TOKST DETTA***  FEL I PROGRAM***')
C
C
  102   FORMAT(' PAR NTYP MUST BE 1,2 OR 3')
  103   FORMAT(' PAR IODD MUST BE AN ODD NUMBER IN THE RANGE 0-131072')
  104   FORMAT(' PAR NBIT MUST BE IN THE RANGE (3,17)')
  105   FORMAT(' PAR KNEP MUST BE EQUAL TO ZERO OR 1')
  106   FORMAT(' PAR ISTAR MUST BE 1,2,3 OR 4')
  107   FORMAT(' PAR NC1 MUST BE GREATER OR EQUAL TO ONE')
  108   FORMAT(' THE DEPOSIT OF A NEW VALUE FOR THE GLOBAL
     * VARIABLE ISIMN FAILED')
  109   FORMAT(' FILN IS UNDEFINED')
  110   FORMAT(' PAR NC1(OR NC2)>11 OR NC1(OR NC2)> NUMBER OF COLUMNS IN
     * THE INPUT FILE ',A5)
  111   FORMAT(' EOF    END OF FILE ',A5)
  112   FORMAT(' FILE ',A5,' NOT FOUND')
C
C
        END
```

Subsystem INPUT

```
C       NAME: DOCINP                                  NUMBER:
C       ----                                          ------
C
C       SUBTITLE: DOCUMENTATION FOR THE SYSTEM INPUT (INPUHG)
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON             DATE: 1978-06-22
C       -----------                                    ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                      VERSION:
C       --------                                       -------
C
C
C------------------------------------------------------------------------
C
        SUBROUTINE DOCINP
C
        COMMON IDM1(7),LINES,IDOC,IYEAR,MONTH,IDAY,ISIMN,IDM2(20),RIODD,
     *  IDM3(12),IBP,IDM5(4),DTI,EI,SD,IDM4(28),NC1,NC2,NTYP,IODD,NBIT,
     *  LA,IX,AMP,YI,FILN,ISTAR,KNEP
C
        LODD=RIODD+0.1
C
C
        WRITE(6,10) ISIMN
   10   FORMAT('1DOCUMENTATION',35X,'SIMU. NO:',I11)
        WRITE(6,20) IYEAR,MONTH,IDAY
   20   FORMAT(' *************',35X,'DATE    :',I5,'-',I2,'-',I2)
        LINES=2
C
        WRITE(6,30)
   30   FORMAT(//' SYSTEM INPUT:')
        WRITE(6,40)
   40   FORMAT(' ------------')
        WRITE(6,50) NTYP
   50   FORMAT(' NTYP =',I9)
        WRITE(6,60) NBIT,NC1
   60   FORMAT(25X,'NBIT =',I9,9X,'NC1  =',I9)
        WRITE(6,70) LODD,KNEP,NC2
   70   FORMAT(' IODD =',I9,9X,'KNEP =',I9,9X,'NC2  =',I9)
        WRITE(6,80) SD,ISTAR,FILN
```

Subsystem INPUT

```
   80   FORMAT(' SD    =',G13.6,5X,'ISTAR=',I9,9X,'FILN =',4X,A5)
        WRITE(6,85) IBP
   85   FORMAT(25X,'IBP   =',I9)
        WRITE(6,90) DTI,AMP
   90   FORMAT(' DT    =',G13.6,5X,'AMP   =',G13.6)
        LINES=LINES+10
C
        RETURN
        END
```

Subsystem SYSHG

```
C       NAME: SYSHG                                   NUMBER:
C       ----                                          ------
C
C       SUBTITLE:
C       ---------
C       SIMULATION OF A SINGLE INPUT, SINGLE OUTPUT
C       SYSTEM OF A GENERAL STRUCTURE.
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SIMNON STRUCTURED SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       ---------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON      DATE: 1979-02-18
C       -----------                             ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                               VERSION:
C       ---------                               --------
C
C
C-----------------------------------------------------------------------------
C
C
C       PURPOSE:
C       =======
C       TO COMPUTE THE OUTPUT FROM A SINGLE INPUT, SINGLE OUTPUT
C       SYSTEM OF A GENERAL STRUCTURE (GIVEN BELOW).
C
C       THE SYSTEM STRUCTURE:
C
C                           B               1+C
C       (1+F)*Y1(T) =  --- *U(T-KD)  +  --- *E(T)                     (1)
C                          1+A              1+D
C
C       Y(T) = Y1(T) + YLEV                                           (2)
C                                                  -1
C       WHERE  F,A,B,D AND C ARE POLYNOMIALS IN Q    (THE BACKWARDS SHIFT
C       OPERATOR) OF ORDERS NFS,NAS,NBS,NDS AND NCS RESPECTIVELY SUCH
C       THAT F(0)=0, A(0)=0 ETC.
C       Y(T) IS THE OUTPUT, U(T) IS THE INPUT AND E(T) IS A SEQUENCE OF
C       INDEPENDENT RANDOM N(0,LAMB) VARIABLES. YLEV IS A CONSTANT AND
C       KD THE TIME DELAY.
C
```

Subsystem SYSHG

```
C
C       INTRODUCE:
C
C                 B
C       Z(T) = --- *U(T-KD)    OR    Z(T) = -A*Z(T) + B*U(T-KD)          (3)
C               1+A
C
C               1+C
C       V(T) = --- *E(T)       OR    V(T) = -D*V(T) + C*E(T)              (4)
C               1+D
C
C
C       THE OUTPUT Y IS COMPUTED AS:
C
C
C       Y1(T) = PHI(T) *THETA + E(T)      (# THE TRANSPOSED VECTOR)   (5)
C
C       Y(T) = Y1(T) + YLEV                                              (6)
C
C
C       WHERE:
C
C                  I -Y1(T-1)      I                   I F(1)    I
C                  I      :        I                   I   :     I
C                  I      :        I                   I   :     I
C                  I -Y1(T-NFS)    I                   I F(NFS)  I
C                  I -Z(T-1)       I                   I A(1)    I
C                  I      :        I                   I   :     I
C                  I      :        I                   I   :     I
C                  I -Z(T-NAS)     I                   I A(NAS)  I
C                  I  U(T-KD-1)    I                   I B(1)    I
C                  I      :        I                   I   :     I
C       PHI(T) =   I      :        I    AND THETA =    I   :     I      (7),(8)
C                  I      :        I                   I   :     I
C                  I  U(T-KD-NBS)  I                   I B(NBS)  I
C                  I -V(T-1)       I                   I D(1)    I
C                  I      :        I                   I   :     I
C                  I      :        I                   I   :     I
C                  I -V(T-NDS)     I                   I D(NDS)  I
C                  I  E(T-1)       I                   I C(1)    I
C                  I      :        I                   I   :     I
C                  I      :        I                   I   :     I
C                  I  E(T-NCS)     I                   I C(NCS)  I
C
C
C
C       GLOBAL VARIABLES:    (DEFAULT VALUES [ ] )
C       -----------------
C       LDK1     - [5] LOGICAL UNIT NO.   (.DAT SLOT)
C       LDK2     - [7]             DO.
C       IYEAR    - [0]
C       MONTH    - [0]
C       IDAY     - [0]
```

Subsystem SYSHG

```
C       IDOC     - [O] IF IDOC=1 A DOCUMENTATION PAGE IS PRINTED
C                     AT THE END OF EVERY SIMULATION
C       NB:      THESE VARIBLES ARE SET OUTSIDE SYSHG BY A CALL
C                TO THE SUBROUTINE GLOBV FROM SYSTS
C       FNRS     - FILE NAMES FOR THE SYSTEM RESP. NOISE IMPULSE
C       FNRN       RESPONSES.   SEE PAR NIMR.
C
C
C       OUTPUT:
C       ------
C       Y        - THE OUTPUT OF THE SYSTEM  CF.EQ.4
C
C
C       INPUT:
C       -----
C       U        - THE INPUT TO THE SYSTEM CF.EQ.4
C
C
C       PARAMETERS:
C       ----------
C       NFS      - THE ORDER OF THE F-POLYNOMIAL
C       NAS      - THE ORDER OF THE A-POLYNOMIAL
C       NBS      - THE ORDER OF THE B-POLYNOMIAL
C       NDS      - THE ORDER OF THE D-POLYNOMIAL
C       NCS      - THE ORDER OF THE C-POLYNOMIAL
C       KD       - THE TIME DELAY   NOTE:  NPAR=NFS+NAS+NBS+NDS+NCS<31
C       THS      - VECTOR OF SYSTEM PARAMETERS    (THETA)  SEE EQ.8
C                  (FPAR,APAR,BPAR,DPAR,CPAR)
C       YLEV     - CONSTANT ADDED TO THE OUTPUT OF THE SYSTEM
C       LAMB     - STANDARD DEVIATION OF THE NOISE
C       NODD     - STARTING VALUE FOR THE RANDOM NUMBER GENERATOR
C       DT       - SAMPLING PERIOD
C       NIMR     - PARAMETER TO DECIDE WHETHER THE IMPULSE RESPONSES
C                  OF THE DETERMINISTIC AND THE NOISE PARTS OF THE SYST-
C                  EM ARE TO BE COMPUTED AND PLACED ON FILES FNRS,FNRN
C                  NIMR=0       NOT COMPUTED
C                  NIMR=1       COMPUTED
C
C                  STARTING POINTS OF THE SYSTEM PARAMETERS IN THETA
C                  NST1=1
C                  NST2=1+NFS
C                  NST3=1+NFS+NAS
C                  NST4=1+NFS+NAS+NBS
C                  NST5=1+NFS+NAS+NBS+NDS
C                  NPS1=NAS+NBS
C                  NPS2=NDS+NCS
C                  NP1=NPAR-1
C                  KDX=KD+1
C
C       OBSERVE THAT A CHANGE HAS BEEN MADE IN NOTATIONS,
C       CF. DECLARATIONS SEC.2 SYSHG1 AND THE SUBROUTINE DOCSYS.
C
C       THE MODEL AS THE USER SEES IT, IS:
```

Subsystem SYSHG

```
C
C                              B                1+C
C       (1+A)*Y1(T) =  ---  *U(T-KD)  +  ---  *E(T)
C                             1+F               1+D
C
C
C-----------------------------------------------------------------------
C
C
C       CHARACTERISTICS:
C       ===============
C
C       SUBROUTINES REQUIRED:                      (LIBRARY)
C       ---------------------
C       SYSHG1                                       SCAPRO
C       SYSHG2                                       RMOVE
C       SYMESS                                       MCNODI
C       DOCSYS                                       DINT
C       FIWR                                         FIDENT
C       IMRES
C       PMPY
C
C-----------------------------------------------------------------------
C
        SUBROUTINE SYSHG
C
        LOGICAL MSTOP
C
        COMMON/DESTIN/IDUM,IPART
        COMMON/TIME/T
C
        COMMON IDM1(2),KMESS,MSTOP,IDM2(4),IDOC,IDM3(120),U,Y,TS,DT,
     *  RLAMB,YLEV,THS(30),PHI(30),OLDU(7),NFS,NAS,NBS,NDS,NCS,NST1,
     *  NST2,NST3,NST4,NST5,NPS1,NPS2,NPAR,NP1,KD,KDX,NODD,V1,Z1,E,Y1
C
C
        GO TO(1,1,1,4,5,6,7,8),IPART
C
C-----------------------------------------------------------------------
C
C       IDENTIFICATION-DECLARATIONS-CONSTANT ASSIGNEMENTS
C       ================================================
C
    1   CALL SYSHG1
        RETURN
C
C-----------------------------------------------------------------------
C
C       INITIAL
C       =======
C
    4   CALL SYSHG2
        RETURN
```

Subsystem SYSHG

```
C
C-----------------------------------------------------------------------
C
C       OUTPUTS
C       =======
C
    5   CALL MCNODI(NODD,E)
        E=RLAMB*E
        Y1=SCAPRO(PHI(1),1,THS(1),1,NPAR)+E
        Y=Y1+YLEV
        RETURN
C
C-----------------------------------------------------------------------
C
C       DYNAMICS
C       ========
C
    6   IF(NAS.GT.0) Z1=SCAPRO(PHI(NST2),1,THS(NST2),1,NPS1)
        IF(NDS.GT.0) V1=SCAPRO(PHI(NST4),1,THS(NST4),1,NPS2)+E
C
        IF(NP1.GT.0) CALL RMOVE(PHI(NP1),-1,PHI(NPAR),-1,NP1)
        IF(KD.GT.0) CALL RMOVE(OLDU(KD),-1,OLDU(KDX),-1,KD)
C
        OLDU(1)=U
        IF(NFS.GT.0) PHI(NST1)=-Y1
        IF(NAS.GT.0) PHI(NST2)=-Z1
        IF(NBS.GT.0) PHI(NST3)=OLDU(1+KD)
        IF(NDS.GT.0) PHI(NST4)=-V1
        IF(NCS.GT.0) PHI(NST5)=E
C
        TS=T+DT
C
        RETURN
C
C-----------------------------------------------------------------------
    7   RETURN
C-----------------------------------------------------------------------
C
C       FINAL COMPUTATIONS
C       ==================
C
    8   IF(MSTOP) CALL SYMESS
        IF(KMESS.EQ.1.AND.IDOC.EQ.1) CALL DOCSYS
        RETURN
        END
```

Subsystem SYSHG

```
C       NAME: SYSHG1                                 NUMBER:
C       ----                                         ------
C
C       SUBTITLE:
C       --------
C       SIMULATION OF A SINGLE INPUT, SINGLE OUTPUT
C       SYSTEM OF A GENERAL STRUCTURE.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SIMNON STRUCTURED SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1979-02-18
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO PERFORM PARTS 1,2 AND 3 OF SYSHG
C       =======
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
        SUBROUTINE SYSHG1
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION RN(5)
C
        COMMON/DESTIN/IDUM,IPART
        COMMON/USER/LSTOP
C
        COMMON IDM1(3),MSTOP,INHG,ISYSH,IDH,IDM2(96),RNFS,RNAS,RNBS,
     *  RNDS,RNCS,RNODD,RKD,RIMR,IDM3(4),NPMAX,KDMAX,IDM4(4),U,Y,TS,
     *  DT,RLAMB,YLEV,THS(30)
C
        EQUIVALENCE (RN(1),RNFS)
C
```

Subsystem SYSHG

```
C
C
C
C
      KDMAX=6
      NPMAX=30
C
      GO TO(1,2,3,99,99,99,99,99),IPART
C
C-----------------------------------------------------------------------
C
C     IDENTIFICATION
C     ==============
C
    1 CALL IDENT(5HDISCR,5HSYSHG)
      INHG=0
      ISYSH=0
      IDH=0
      RETURN
C
C-----------------------------------------------------------------------
C
C     DECLARATIONS
C     ============
C
    2 CALL INPUT(U,4HU    )
      CALL OUTPUT(Y,4HY    )
C
      CALL PAR(RNFS,4HNAS )
      CALL PAR(RNAS,4HNFS )
      CALL PAR(RNBS,4HNBS )
      CALL PAR(RNDS,4HNDS )
      CALL PAR(RNCS,4HNCS )
      CALL PAR(RKD,4HKDS )
C
      CALL PARV(THS,NPMAX,4HTHS )
      CALL PAR(YLEV,4HYLEV)
C
      CALL PAR(RNODD,4HNODD)
      CALL PAR(RLAMB,4HLAMB)
C
      CALL PAR(RIMR,4HNIMR)
C
      CALL PAR(DT,4HDT   )
      CALL TSAMP(TS,4HTS   )
C
      RETURN
C
C-----------------------------------------------------------------------
C
C     CONSTANT ASSIGNEMENTS
C     =====================
C
```

Subsystem SYSHG

```
    3   DO 31 I=1,5
   31   RN(I)=0.
C
        DO 32 I=1,NPMAX
   32   THS(I)=0.
C
        RKD=0.
        YLEV=0.
        DT=1.
        RNODD=19.
        RLAMB=1.
        RIMR=0.
        ISYSH=1
C
        RETURN
C
C-----------------------------------------------------------------------
C
  99    LSTOP=.TRUE.
        MSTOP=.TRUE.
        RETURN
        END
```

Subsystem SYSHG

```
C       NAME: SYSHG2                                   NUMBER:
C       ----                                           ------
C
C       SUBTITLE:
C       --------
C       SIMULATION OF A SINGLE INPUT, SINGLE OUTPUT
C       SYSTEM OF A GENERAL STRUCTURE.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1979-02-18
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                    VERSION:
C       --------                                     -------
C
C
C
C-----------------------------------------------------------------------
C
C
C       PURPOSE: TO PERFORM PART 4 OF SYSHG
C       =======
C
C
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C
C
        SUBROUTINE SYSHG2
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION RN(5),N(5),NST(5),FNAM(2)
C
        COMMON/USER/LSTOP
C
        COMMON LDK1,LDK2,KMESS,MSTOP,INHG,ISYSH,IDH,IDM1(5),ISIMN,
     *  IDM2(4),FNRS,FNRN,IDM3(82),RNFS,RNAS,RNBS,RNDS,RNCS,RNODD,RKD,
     *  RIMR,IDM4(3),NIMR,NPMAX,KDMAX,IDM5(16),THS(30),PHI(30),OLDU(7),
```

Subsystem SYSHG

```
     *  NFS,NAS,NBS,NDS,NCS,NST1,NST2,NST3,NST4,NST5,NPS1,NPS2,NPAR,
     *  NP1,KD,KDX,NODD
C
        EQUIVALENCE (RN(1),RNFS),(N(1),NFS),(NST(1),NST1)
C
        DATA FNAM(2)/' BIN'/
C
C
C
C-----------------------------------------------------------------------
C
C       INITIAL
C       =======
C
    4   MSTOP=.FALSE.
        KMESS=1
        EPS=0.1
C
        DO 41 I=1,30
   41   PHI(I)=0.
C
        DO 42 I=1,7
   42   OLDU(I)=0.
C
        NPAR=0
        NST1=1
        DO 43 I=1,5
        N(I)=RN(I)+EPS
        IF(N(I).LT.0) GO TO 90
        IF(I.GT.1) NST(I)=NST(I-1)+N(I-1)
   43   NPAR=NPAR+N(I)
C
        NP1=NPAR-1
        KD=RKD+EPS
C
        NODD=RNODD+EPS
        NIMR=RIMR+EPS
C
        KDX=KD+1
        NPS1=NAS+NBS
        NPS2=NDS+NCS
C
        IF(NPAR.GT.NPMAX.OR.NPAR.LT.1) KMESS=16
        IF(KD.LT.0.OR.KD.GT.KDMAX) KMESS=17
        IF(MOD(NODD,2).EQ.0.OR.NODD.LT.1) KMESS=18
        IF(NIMR.LT.0.OR.NIMR.GT.1) KMESS=19
        IF(KMESS.GT.1) GO TO 99
C
        IF(NIMR.NE.1) GO TO 49
        CALL FIDENT('FNRS',4H    ,FNRS,IND1)
        IF(IND1.EQ.0) GO TO 46
        KMESS=23
        GO TO 99
```

Subsystem SYSHG

```
   46    FNAM(1)=FNRS
         CALL FSTAT(LDK1,FNAM,J)
         CALL CLOSE(LDK1)
         IF(J.EQ.-1) GO TO 91
C
         CALL FIDENT('FNRN',4H     ,FNRN,IND1)
         IF(IND1.EQ.0) GO TO 48
         KMESS=23
         GO TO 99
   48    FNAM(1)=FNRN
         CALL FSTAT(LDK1,FNAM,J)
         CALL CLOSE(LDK1)
         IF(J.EQ.-1) GO TO 92
         ITYP=0
         CALL FIWR(ITYP,N,NST,THS,FNRS)
         ITYP=1
         CALL FIWR(ITYP,N,NST,THS,FNRN)
C
   49    IF(IDH.EQ.1) RETURN
         ISIMN=ISIMN+1
         CALL DINT('ISIMN',4H     ,ISIMN,IND1)
         IF(IND1.GT.0) GO TO 93
         RETURN
C
C-----------------------------------------------------------------------
C
   90    KMESS=20
         GO TO 99
   91    KMESS=21
         GO TO 99
   92    KMESS=22
         GO TO 99
   93    KMESS=15
   99    MSTOP=.TRUE.
  100    LSTOP=.TRUE.
         RETURN
         END
```

```
C       NAME: FIWR                                        NUMBER:
C       ----                                              ------
C
C       SUBTITLE:
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON        DATE: 1978-06-22
C       -----------                               ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                 VERSION:
C       --------                                  -------
C
C
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO GENERATE A FILE , FN, WITH THE IMPULSE RESPONSE
C                OF THE SYSTEM DEFINED BY ITYP,N,NST,THS)
C                CF. COMMENTS TO IMRES
C
C-----------------------------------------------------------------------
C
        SUBROUTINE FIWR(ITYP,N,NST,THS,FN)
C
        DIMENSION H(100),N(1),NST(1),THS(1),FI(2),KOD(10)
C
        COMMON LDK1,LDK2
C
        DATA FI(2)/' BIN'/
C
        FI(1)=FN
        DO 10 I=1,10
   10   KOD(I)=0
        KOD(1)=100
        KOD(2)=1
        KOD(4)=50
        CALL ENTER(LDK1,FI)
        WRITE(LDK1) (KOD(I),I=1,10)
```

Subsystem SYSHG

```
      CALL IMRES(ITYP,N,NST,THS,H)
C
      DO 20 I=1,100
   20 WRITE(LDK1) H(I)
C
      CALL CLOSE(LDK1)
      RETURN
      END
```

Subsystem SYSHG

```
C       NAME: SYMESS                                     NUMBER:
C       ----                                             ------
C
C       SUBTITLE: ERROR MESSAGES
C
```


Subsystem SYSHG

```
        ISMESS=KMESS-14
C
        GO TO(15,16,17,18,19,20,21,22,23),ISMESS
C
   15   WRITE(LU,115)
        GO TO 100
   16   WRITE(LU,116)
        GO TO 100
   17   WRITE(LU,117)
        GO TO 100
   18   WRITE(LU,118)
        GO TO 100
   19   WRITE(LU,119)
        GO TO 100
   20   WRITE(LU,120)
        GO TO 100
   21   WRITE(LU,121) FNRS
        GO TO 100
   22   WRITE(LU,122) FNRN
        GO TO 100
   23   WRITE(LU,123)
        GO TO 100
  100   RETURN
C
C
  115   FORMAT(' THE DEPOSIT OF A NEW VALUE FOR THE GLOBAL
     * VARIABLE ISIMN FAILED')
C
  116   FORMAT(' NPAR=NAS+NFS+NBS+NDS+NCS MUST BE.GT.ZERO.AND.LE.30')
  117   FORMAT(' PAR KD MUST BE IN THE RANGE  (0,6)')
  118   FORMAT(' PAR NODD MUST BE AN ODD NUMBER IN THE RANGE 0-131072')
  119   FORMAT(' PAR NIMR MUST BE EITHER 0 OR 1')
  120   FORMAT(' NFS,NAS,NBS,NDS,NCS MUST ALL BE .GT.OR.EQ. TO ZERO')
  121   FORMAT(' A FILE WITH THE NAME ',A5,' IS PRESENT ON DISK, '
     * ,'ASSIGN A NEW NAME TO FNRS')
  122   FORMAT(' A FILE WITH THE NAME ',A5,' IS PRESENT ON DISK, ',
     * 'ASSIGN A NEW NAME TO FNRN')
  123   FORMAT(' FNRS OR FNRN  UNDEFINED')
C
        END
```

Subsystem SYSHG

```
C       NAME: DOCSYS                                      NUMBER:
C       ----                                              -------
C
C       SUBTITLE: DOCUMENTATION FOR THE SYSTEM SYSHG
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       ---------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON                DATE: 1978-06-22
C       -----------                                       ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                         VERSION:
C       ---------                                         -------
C
C
C-----------------------------------------------------------------------------
C
        SUBROUTINE DOCSYS
C
        LOGICAL MSTOP
C
C
        COMMON IDM1(4),INHG,IDM2(2),LINES,IDOC,IYEAR,MONTH,IDAY,ISIMN,
     *  IDM3(4),FNRS,FNRN,IDM4(92),RNODD,IDM5(7),NIMR,IDM6(12),DTS,
     *  RLAMB,YLEV,THS(30),IDM7(74),NFS,NAS,NBS,NDS,NCS,IDM8(7),NPAR
C
C
        LNODD=RNODD+0.1
C
C
        IF(INHG.EQ.1) GO TO 25
        WRITE(6,10) ISIMN
   10   FORMAT('1DOCUMENTATION',35X,'SIMU. NO:',I11)
        WRITE(6,20) IYEAR,MONTH,IDAY
   20   FORMAT(' *************',35X,'DATE     :',I5,'-',I2,'-',I2)
        LINES=2
C
   25   WRITE(6,110)
  110   FORMAT(//' SYSTEM SYSHG:')
        WRITE(6,40)
   40   FORMAT(' -------------')
```

Subsystem SYSHG

```
      WRITE(6,120) NFS,KDS,RLAMB
  120 FORMAT(' NAS  =',I9,9X,'KDS  =',I9,9X,'LAMB =',G13.6)
      WRITE(6,130) NAS,LNODD,YLEV
  130 FORMAT(' NFS  =',I9,9X,'NODD =',I9,9X,'YLEV =',G13.6)
      WRITE(6,140) NBS,NIMR,DTS
  140 FORMAT(' NBS  =',I9,9X,'NIMR =',I9,9X,'DT   =',G13.6)
      WRITE(6,150) NDS,FNRS
  150 FORMAT(' NDS  =',I9,33X,'FNRS =',4X,A5)
      WRITE(6,160) NCS,FNRN
  160 FORMAT(' NCS  =',I9,33X,'FNRN =',4X,A5/)
      K=MOD(NPAR,3)
      KK=NPAR-K
      WRITE(6,161) (I,THS(I),I=1,KK)
  161 FORMAT(3(' THS(',I2,')=',G13.6,2X))
      KK=KK+1
      IF(K.EQ.1) WRITE(6,162) KK,THS(KK)
  162 FORMAT(' THS(',I2,')=',G13.6)
      IF(K.EQ.2) WRITE(6,163) (I,THS(I),I=KK,NPAR)
  163 FORMAT(2(' THS(',I2,')=',G13.6,2X))
      LINES=LINES+10+KK/3
      IF(K.GT.0) LINES=LINES+1
C
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: IDHG                                      NUMBER:
C       ----                                            -------
C
C       SUBTITLE: RECURSIVE IDENTIFICATION
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SIMNON STRUCTURED SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON         DATE: 1978-06-22
C       -----------                                ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C
C       ACCEPTED:                                       VERSION:
C       ---------                                       -------
C
C-----------------------------------------------------------------------
C
C       SUBROUTINES REQUIRED:                           LIBRARY
C       --------------------
C       IDHG1    HATFI2   WRTHG    FILT     SPFZN    FIDENT   NORM
C       IDHG2    FISTAR   DOCID    UPDATE   INPOL    DINT     MCNODI
C       IDHG3    HATCOM   IDMESS   STABB    FAST2    INV2     PRB
C       SHIFT    HATWRT   STAWRT   EKF      RTS      RMOVE    PRBSTA
C       HATFI1   IMRES    GEALR    KJADMH   PMPY     SCAPRO
C
C       REF.: H. GUNNARSSON      RECURSIVE IDENTIFICATION USING A GENERAL
C                                MODEL STRUCTURE. A PROGRAM PACKAGE AND
C                                SIMULATIONS.
C
C-----------------------------------------------------------------------
C
        SUBROUTINE IDHG
C
        LOGICAL MSTOP
        COMMON/DESTIN/IDUM,IPART
        COMMON IDM1(2),KMESS,MSTOP,IDM2(4),IDOC
C
        GO TO(1,1,1,4,5,5,5,8),IPART
C-----------------------------------------------------------------------
C
    1   CALL IDHG1
        RETURN
C
    4   CALL IDHG2
        CALL HATFI1
        RETURN
C
```

Subsystem IDHG

```
    5   CALL IDHG3
        RETURN
C
    8   IF(KMESS.GT.1) GO TO 90
        IF(IDOC.EQ.1) CALL DOCID
        CALL HATFI2
        CALL HATWRT
   90   IF(MSTOP) CALL IDMESS
        RETURN
        END
```

Subsystem IDHG

```
C       NAME: IDHG1                                   NUMBER:
C       ----                                          -------
C
C       SUBTITLE: RECURSIVE IDENTIFICATION
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SIMNON STRUCTURED SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON        DATE: 1978-06-22
C       -----------                               ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                 VERSION:
C       --------                                  -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO PERFORM PARTS 1,2 AND 3 OF IDHG
C       =======
C
C-----------------------------------------------------------------------
C
C
        SUBROUTINE IDHG1
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION RN(5)
C
        COMMON/DESTIN/IDUM,IPART
        COMMON/USER/LSTOP
C
C
        COMMON IDM1(3),MSTOP,INHG,ISYSH,IDH,IDM2(10),FDUM(6),IDM3(280),
     *  RMETO,RKD,RSTAB,RPRER,RFIRST,FIGE,UINVA,RNFST,RWRT,RWRT1,RWRTN,
     *  RHAT,RHATN,WTI,RNF,RNA,RNB,RND,RNC,PFST,THO(10),PO(10),TH10(10),
     *  P10(10),IDM4(26),IDIM1,IDIM2,IDM5(7),IPDIM,IDM6(12),WV,V,WTO,
     *  TS,DT,WRES,RLIM,DELTA,Y,WT,RES,U,RKJ,IDM7(12),R1(10),IDM8(200),
     *  TH(10),IDM9(280),TH1(10)
C
C
        EQUIVALENCE (RN(1),RNF)
C
```

Subsystem IDHG

```
      DATA FX/'    '/
C
      IDIM1=5
      IDIM2=20
      IPDIM=10
C
C
      GO TO(1,2,3,99,99,99,99,99),IPART
C
C-----------------------------------------------------------------------
C
C     IDENTIFICATION
C     ==============
C
    1 CALL IDENT(5HDISCR,4HIDHG)
      INHG=0
      ISYSH=0
      IDH=0
      RETURN
C-----------------------------------------------------------------------
C
C     DECLARATIONS
C     ============
C
    2 CALL INPUT(U,4HU   )
      CALL INPUT(Y,4HY   )
C
      CALL OUTPUV(TH,IPDIM,4HTH  )
      CALL OUTPUV(TH1,IPDIM,4HTF  )
C
      CALL PAR(RMETO,5HMETOD)
C
      CALL PAR(RNF,4HNA  )
      CALL PAR(RNA,4HNF  )
      CALL PAR(RNB,4HNB  )
      CALL PAR(RND,4HND  )
      CALL PAR(RNC,4HNC  )
      CALL PAR(RNFST,4HNFST)
      CALL PAR(RKD,4HKD  )
C
      CALL PAR(WTI,4HWTI )
      CALL PAR(WTO,4HWTO )
      CALL PAR(DELTA,5HDELTA)
      CALL PARV(R1,IPDIM,4HR1  )
C
      CALL PAR(RPRER,5HIPRER)
      CALL PAR(RLIM,4HRLIM)
      CALL PAR(RSTAB,5HISTAB)
      CALL PAR(RFIRST,5HIFIST)
      CALL PAR(UINVA,5HINVAU)
      CALL PAR(RKJ,4HKJ  )
C
      CALL PARV(THO,IPDIM,4HTHO )
```

Subsystem IDHG

```
      CALL PARV(TH10,IPDIM,4HTFO )
      CALL PARV(PO,IPDIM,4HPO   )
      CALL PARV(P10,IPDIM,4HP10 )
      CALL PAR(PFST,4HPFST)
C
      CALL PAR(RWRT,4HIWRT)
      CALL PAR(RWRT1,5HIWRT1)
      CALL PAR(RWRTN,5HIWRTN)
      CALL PAR(RHAT,5HIHAT )
      CALL PAR(RHATN,5HIHATN)
      CALL PAR(FIGE,5HIFIGE)
C
      CALL VAR(RES,4HRES )
      CALL VAR(WRES,4HWRES)
      CALL VAR(V,4HV    )
      CALL VAR(WV,4HWV   )
      CALL VAR(WT,4HWT   )
C
      CALL PAR(DT,4HDT   )
      CALL TSAMP(TS,4HTS   )
      RETURN
C-----------------------------------------------------------------------------------------------------------
C
C     CONSTANT ASSIGNMENTS
C     ====================
C
C
    3 RMETO=0.0
C
      DO 310 I=1,IDIM1
  310 RN(I)=0.
      RNFST=0.
      RKD=0.0
C
      WTI=1.
      WTO=1.
      DELTA=0.
C
      RPRER=0.
      RLIM=-1.
      RSTAB=0.
      RFIRST=0.
      UINVA=0.
      RKJ=0.
C
      DO 320 I=1,IPDIM
      THO(I)=0.
      TH10(I)=0.
      R1(I)=0.
      PO(I)=100.
  320 P10(I)=100.
      PFST=100.
C
```

Subsystem IDHG

```
      RWRT1=0.0
      RWRT=-5.
      RWRTN=100.
C
      RHAT=-20.
      RHATN=500.
      FIGE=0.
C
      DO 330 I=1,6
  330 FDUM(I)=FX
      DT=1.
C
      IDH=1
C
      RETURN
C
C-----------------------------------------------------------------------
C
   99 LSTOP=.TRUE.
      MSTOP=.TRUE.
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: IDHG2                                      NUMBER:
C       ----                                             -------
C
C       SUBTITLE: RECURSIVE IDENTIFICATION
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       ---------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       ------------                                ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       ---------                                   --------
C
C
C
C-------------------------------------------------------------------------
C
C       PURPOSE: TO PERFORM PART 4 OF IDHG
C
C-------------------------------------------------------------------------
C
C
C
      SUBROUTINE IDHG2
C
      LOGICAL LSTOP,MSTOP,SA,IV,LFILT,STABCK,PRERES,LSPFZN
C
      DIMENSION RN(5),N(5),RMOVO(984),NSTART(5)
C
      COMMON/USER/LSTOP
C
      COMMON IDM1(2),KMESS,MSTOP,IDM2(8),ISIMN,IDM3(296),
     * RMETO,RKD,RSTAB,RPRER,RFIRST,FIGE,UINVA,RNFST,
     * RWRT,RWRT1,RWRTN,RHAT,RHATN,WTI,RNF,RNA,RNB,RND,RNC,PFST,
     * THO(10),PO(10),TH10(10),P10(10),ISTAB,IPRER,
     * IDM4(4),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     * ICK,ISAMP,IFIRST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,IDIM1,
     * IDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     * NPS1,NPS2,NP1,INVAU,IDM5,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
```

Subsystem IDHG

```
     *   DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *   PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *   P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
C
C
         EQUIVALENCE (RN(1),RNF)
         EQUIVALENCE (N(1),NF),(NSTART(1),NST1),(P(1,1),RMOVO(1))
C
C-----------------------------------------------------------------------------------------
C
C        INITIAL
C        =======
C
C****** RESET  ******
    4    EPS=0.1
         KMESS=1
         MSTOP=.FALSE.
         SA=.FALSE.
         IV=.FALSE.
         LFILT=.FALSE.
         PRERES=.FALSE.
         STABCK=.FALSE.
         LSPFZN=.FALSE.
         ISAMP=0
         NPDIM=0
         NST1=1
C
         DO 410 I=1,984
  410    RMOVO(I)=0.0
C
C
C****** METHOD ******
         METOD=RMETO+EPS
         IF(METOD.LT.1.OR.METOD.GT.7) KMESS=24
         IF(METOD.EQ.1.OR.METOD.EQ.2) SA=.TRUE.
         IF(METOD.EQ.2.OR.METOD.EQ.4) LFILT=.TRUE.
         IF(METOD.EQ.5) IV=.TRUE.
C
C
C****** MODEL STRUCTURE ******
         KD=RKD+EPS
         IF(KD.LT.0)KMESS=34
         NFST=RNFST+EPS
C
         DO 420 I=1,IDIM1
         N(I)=RN(I)+EPS
         IF(N(I).LT.0) KMESS=25
         NPDIM=NPDIM+N(I)
         IF(NPDIM.GT.IPDIM)KMESS=27
         IF(I.GT.1) NSTART(I)=NSTART(I-1)+N(I-1)
         IF(KMESS.GT.1)GO TO 99
  420    CONTINUE
```

Subsystem IDHG

```
C
      DO 421 I=1,IDIM1
  421 NSH(I)=N(I)
C
      IF(.NOT.LFILT) GO TO 423
      DO 422 I=1,3
  422 NSH(I)=MAX0(1+ND,N(I))
C
  423 NSH(3)=NSH(3)+KD
C
      DO 425 I=1,IDIM1
      IF(NSH(I).GT.IDIM2) KMESS=26
      IF(KMESS.GT.1) GO TO 99
  425 CONTINUE
C
      NPS1=NA+NB
      NPS2=ND+NC
      NP1=NPDIM-1
C
      IF(NA.EQ.0.AND.ND.EQ.0.AND.NC.EQ.0) LFILT=.FALSE.
      IF(.NOT.LFILT.AND.METOD.EQ.2) METOD=1
      IF(.NOT.LFILT.AND.METOD.EQ.4) METOD=3
      IF(IV.AND.(NA.GT.0.OR.ND.GT.0.OR.NC.GT.0))KMESS=28
C     IF(METOD.EQ.6.AND.NA,NB....     ) KMESS=29
      IF(METOD.EQ.7.AND.(NA.GT.0.OR.ND.GT.0)) KMESS=30
C
      IF(METOD.NE.6) GO TO 440
      DO 430 I=1,5
  430 NSH(I)=20
C
C
C****** SPEC PAR ******
  440 IPRER=RPRER+EPS
      IF(IPRER.LT.0.OR.IPRER.GT.1) KMESS=31
      IF(IPRER.EQ.0) PRERES=.TRUE.
      ISTAB=RSTAB+EPS
      IF(ISTAB.LT.0.OR.ISTAB.GT.2) KMESS=32
      IF(ISTAB.EQ.1.OR.ISTAB.EQ.2) STABCK=.TRUE.
      IF(ISTAB.EQ.2) LSPFZN=.TRUE.
      IF(NA.EQ.0.AND.NC.EQ.0) STABCK=.FALSE.
C
      IFIRST=RFIRST+EPS
      INVAU=UINVA+EPS
C
C
C****** INITIAL VALUES ******
      CALL RMOVE(TH0(1),1,TH(1),1,NPDIM)
      CALL RMOVE(TH10(1),1,TH1(1),1,NPDIM)
C
      DO 450 I=1,NPDIM
      P(I,I)=P0(I)
  450 P1(I,I)=P10(I)
C
```

Subsystem IDHG

```
      IF(METOD.LE.2) P(1,1)=1./P(1,1)
      WT=WTI
      V=0.0
      WV=0.0
      E(1,1)=1./PFST
      E(2,2)=1./PFST
C
C****** PRINT-OUT ******
      IWRT=RWRT+EPS
      IWRTN=RWRTN+EPS
      IWRT1=RWRT1+EPS
C
C****** ACCURACY ******
      IHAT=RHAT+EPS
      IHATN=RHATN+EPS
      IFIGE=FIGE+EPS
      IF(IFIGE.LT.0.OR.IFIGE.GT.3) KMESS=39
      IF(IFIGE.GT.0.AND.IFIGE.LT.4.AND.IHAT.LT.0) KMESS=33
C
      IF(KMESS.GT.1) GO TO 99
      ISIMN=ISIMN+1
      CALL DINT('ISIMN',4H    ,ISIMN,IND1)
      IF(IND1.GT.0) GO TO 98
      RETURN
C
C-----------------------------------------------------------------------
C
   98 KMESS=29
   99 MSTOP=.TRUE.
      LSTOP=.TRUE.
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: IDHG3                                    NUMBER:
C       ----                                           -------
C
C       SUBTITLE: RECURSIVE IDENTIFICATION
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SIMNON STRUCTURED SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO PERFORM PARTS 5,6 AND 7 OF IDHG
C       =======
C
C-----------------------------------------------------------------------
C
        SUBROUTINE IDHG3
C
        LOGICAL LSTOP,MSTOP
C
        COMMON/DESTIN/IDUM,IPART
        COMMON/TIME/T
        COMMON/USER/LSTOP
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(431),
     *  METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),ICK,
     *  ISAMP,IDM2(7),IDIM1,IDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,
     *  NST2,NST3,NST4,NST5,NPS1,NPS2,NP1,INVAU,IDM3(3),WV,V,
     *  WTO,TS,DT,WRES,RLIM,DELTA,Y,WT,RES,U,IDM5(234),
     *  TH(10),S(5,20),PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10)
C
        GO TO(99,99,99,99,5,6,7,99),IPART
C-----------------------------------------------------------------------
C
C       OUTPUT
C       ======
    5   GO TO(51,51,51,51,51,52,53),METOD
C
   51   CALL GEALR
```

Subsystem IDHG

```
      IF(ICK.EQ.10) CALL STAWRT(ISAMP,TH,OLDTH,NPDIM)
      RETURN
C
   52 INKA=1
      CALL KJADMH(INKA)
      RETURN
C
   53 CALL EKF
      RETURN
C-----------------------------------------------------------------------
C
C     DYNAMICS
C     ========
    6 WT=WTO*WT+1.-WTO
      ISAMP=ISAMP+1
      TS=T+DT
      V=V+RES*RES
      WV=WV+WRES*WRES
C
      IF(IWRT.LE.0) GO TO 600
      L=MOD(ISAMP,IWRTN)
      IF(ISAMP.GT.IWRT1.AND.L.NE.0) GO TO 600
      IF(METOD.EQ.7) CALL WREKF
      IF(METOD.LE.6) CALL WRTHG
C
  600 IF(IHAT.LT.0) GO TO 601
      L=MOD(ISAMP,IHATN)
      IF(ISAMP.NE.IHAT.AND.L.NE.0) GO TO 601
      WRITE(LDK1) ISAMP,V,WV,(TH(I),I=1,NPDIM),(TH1(J),J=1,NPDIM)
C
  601 CALL SHIFT(S,IDIM1,NSH)
      S(1,1)=-Y
      IF(METOD.EQ.6) GO TO 602
      S(3,1)=U
      S(5,1)=RES
C
C
  602 GO TO(61,61,61,61,62,63,64),METOD
C
   61 IF(NA.GT.0) S(2,1)=-SCAPRO(PHI(NST2),1,TH(NST2),1,NPS1)
      IF(ND.GT.0) S(4,1)=-SCAPRO(PHI(NST4),1,TH(NST4),1,NPS2)-RES
      RETURN
C
   62 IF(INVAU.LE.0) Z1=SCAPRO(Z(1),1,TH(1),1,NPDIM)
      IF(NP1.GT.0) CALL RMOVE(Z(NP1),-1,Z(NPDIM),-1,NP1)
      IF(INVAU.GT.0) GO TO 620
      Z(1)=-Z1
      Z(NST3)=S(3,KD+1)
      RETURN
C
  620 Z(1)=S(3,KD+1)
      RETURN
C
```

Subsystem IDHG

```
   63   INKA=2
        CALL KJADMH(INKA)
        RETURN
C
   64   RETURN
C
C-----------------------------------------------------------------------
    7   RETURN
C-----------------------------------------------------------------------
C
   99   LSTOP=.TRUE.
        MSTOP=.TRUE.
        RETURN
        END
```

Subsystem IDHG

```
C       NAME: SHIFT                                    NUMBER:
C       ----                                           -------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON             DATE: 1978-06-22
C       -----------                                    ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                      VERSION:
C       --------                                       -------
C
C
C
C
C------------------------------------------------------------------------
C
C       PURPOSE:  TO SHIFT A MATRIX S ALONG LINES
C                 S               - THE MATRIX TO BE SHIFTED
C                 IS              - DECLARED FIRST DIMENSION OF S
C                 NSH(IS)         - VECTOR CONTAINING THE NUMBER OF
C                                   ELEMENTS TO BE SHIFTED IN EACH LINE
C
C------------------------------------------------------------------------
C
        SUBROUTINE SHIFT(S,IS,NSH)
C
        DIMENSION S(IS,1),NSH(1)
C
C
        DO 10 I=1,IS
        NS=NSH(I)
        NS1=NS-1
        IF(NS.LE.1) GO TO 10
        CALL RMOVE(S(I,NS1),-IS,S(I,NS),-IS,NS1)
   10   CONTINUE
C
        RETURN
        END
```

```
C       NAME: GEALR                                      NUMBER:
C       ----                                             -------
C
C       SUBTITLE: ONE STEP OF A GENERAL RECURSIVE
C       --------  IDENTIFICATION ALGORITHM
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS: RECURSIVE IDENTIFICATION
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON           DATE: 1978-06-26
C       -----------                                  ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                    VERSION:
C       --------                                     -------
C
C
C
C
C-----------------------------------------------------------------------
C
C
C       PURPOSE:
C       =======
C       TO COMPUTE ONE STEP OF A GENERAL RECURSIVE IDENTIFICATION
C       ALGORITHM ACCORDING TO THE DESCRIPTION GIVEN BELOW.
C
C
C       THE MODEL USED FOR THE IDENTIFICATION IS:
C
C                      B                1+C
C       (1+F)Y(T)=  --- * U(T-KD) +  --- * E(T)
C                     1+A               1+D
C
C                   -1          -1          -1          -1          -1     -1
C       WHERE F=F(Q  ), A=A(Q  ), B=B(Q  ), D=D(Q  ), C=C(Q  ) (Q    IS
C       THE BACKWARD SHIFT OPERATOR)  ARE POLYNOMIALS OF ORDERS
C       NF,NA,NB,ND,NC  RESPECTIVELY SUCH THAT F(0)=0, A(0)=0 ETC.
C
C       INTRODUCE:
C
C                 B                                      1+C
C       Z(T)=  --- * U(T-KD)        AND        V(T)=  --- * E(T)
C                1+A                                     1+D
C
C
```

Subsystem IDHG

```
C
C
C       OLD SIGNAL VALUES ARE TO BE STORED IN THE S-MATRIX
C       AS FOLLOWS:
C
C           I -Y(T-1),-Y(T-2),-Y(T-3),.......................I
C           I -Z(T-1),-Z(T-2),-Z(T-3),.......................I
C       S = I U(T-1),U(T-2),U(T-3),..........................I
C           I -V(T-1),-V(T-2),-V(T-3),.......................I
C           I E(T-1),E(T-2),E(T-3),..........................I
C
C
C       THE ALGORITHM:
C       -------------
C                                    T
C       RES=E(T)=Y(T)-PHI(T) *THETA(T-1)                         (1)
C
C       THETA(T)=THETA(T-1)+K(T)*E(T)                            (2)
C
C                         P(T-1)*ZV(T)
C       K(T)= -------------------------                          (3)
C                WT+WPSI(T)   *P(T-1)*ZV(T)
C
C                                         T
C       P(T)=[P(T-1)-K(T)*WPSI(T)  *P(T-1)]/WT                   (4)
C
C
C       WHERE:
C
C                 I -Y(T-1)       I                  I F(1)   I
C                 I      :        I                  I   :    I
C                 I      :        I                  I   :    I
C                 I -Y(T-NF)      I                  I F(NF)  I
C                 I -Z(T-1)       I                  I A(1)   I
C                 I      :        I                  I   :    I
C                 I      :        I                  I   :    I
C                 I -Z(T-NA)      I                  I A(NA)  I
C                 I  U(T-KD-1)    I                  I B(1)   I
C                 I      :        I                  I   :    I
C       PHI(T) =  I      :        I     AND THETA =  I   :    I
C                 I      :        I                  I   :    I
C                 I  U(T-KD-NB)   I                  I B(NB)  I
C                 I -V(T-1)       I                  I D(1)   I
C                 I      :        I                  I   :    I
C                 I      :        I                  I   :    I
C                 I -V(T-ND)      I                  I D(ND)  I
C                 I  E(T-1)       I                  I C(1)   I
C                 I      :        I                  I   :    I
C                 I      :        I                  I   :    I
C                 I  E(T-NC)      I                  I C(NC)  I
C
C
C
```

Subsystem IDHG

```
C       REMARK 1.: Z(T) = WPSI(T) EXCEPT WHEN THE INSTRUMENTAL
C                  VARIABLES METHOD IS USED. IN THAT CASE THE
C                  ZV-VECTOR CONSISTS OF THE INSTRUMENTAL
C                  VARIABLES.
C
C       REMARK 2.: WPSI(T)=PHI(T) WHEN THE SUBROUTINE FILT IS
C                  NOT USED.  WPSI(T)= OLD FILTERED SIGNAL
C                  VALUES WHEN THE SUBROUTINE FILT IS USED.
C                  CF. COMMENTS TO FILT.
C
C       REMARK 3.: ADJUSTMENTS TO IMPROVE THE STABILITY OF
C                  THE ALGORITHM ARE IMPLEMENTED IN THE UPDATING
C                  SUBROUTINE UPDATE (R1,DELTA). FOR
C                  FURTHER REFERENCE SEE COMMENTS TO UPDATE.
C
C
C       STOCHASTIC APPROXIMATION VERSION
C       --------------------------------
C                          T
C       RES=Y(T)-PHI(T) *THETA(T-1)                            (1)
C                                T
C       P1(T)=WT*P1(T-1)+WPSI(T) *WPSI(T)    (P1=P[1,1])       (6)
C
C       THETA(T)=THETA(T-1)+RES*WPSI(T)/P1(T)                  (7)
C
C
C       ARGUMENTS:
C       ---------
C
C       GEALRI(S,ISDIM1,P,NPDIM,IPDIM,THETA,PHI,Z,TEE,KD,N,NSTART,R1,Y,
C    *  WT,RES,ISAMP,IFIRST,RLIM,SA,IV,LFILT,PRERES,STABCK,LSPFZN,OLDTH,
C    *  WPSI,W1,W2,W3,DELTA,WRES,ICK,PX,RX)
C
C       S         - MATRIX CONTAINING OLD SIGNAL VALUES CF. DESCRIPT-  (I)
C                   ION GIVEN ABOVE.   DIMENSIONED(ISDIM1,1)
C       ISDIM1    - DECLARED FIRST DIMENSION OF S AND TEE.             (I)
C       P         - THE COVARIANCE MATRIX                              (I/O)
C                   SIZE(NPDIM,NPDIM), DIMENSIONED(IPDIM,1)
C       NPDIM     - ACTUAL DIMENSION OF P, ACTUAL NUMBER OF PARAMETERS (I)
C       IPDIM     - DECLARED FIRST DIMENSION OF P                      (I)
C       THETA     - VECTOR OF PARAMETER ESTIMATES, SIZE(NPDIM)         (I/O)
C       PHI       - VECTOR CONTAINING OLD SIGNAL VALUES CORRESPONDING  (O)
C                   TO THE PARAMETERS.   SIZE(NPDIM)
C       Z         - VECTOR CONTAINING OLD SIGNAL VALUES  E.G. THE      (I)
C                   INSTRUMENTAL VARIABLES    SIZE(NPDIM)
C       TEE       - MATRIX CONTAINING OLD FILTERED SIGNAL VALUES       (I/O)
C                   SIZE(ISDIM1,1), DIMENSIONED(ISDIM1,1)
C       KD        - TIME DELAY                                         (I)
C       N         - VECTOR DEFINING THE MODEL STRUCTURE  CF. MODEL     (I)
C                   GIVEN ABOVE.   SIZE(ISDIM1)
C                   N(1)=NF
C                   N(2)=NA
C                   N(3)=NB
```

```
C                       N(4)=ND
C                       N(5)=NC
C       NSTART    - VECTOR CONTAINING THE STARTING POINTS OF THE       (I)
C                   PARAMETERS IN THETA.    SIZE(ISDIM1)
C         NST1      NSTART(1)=1
C         NST2      NSTART(2)=1+NF
C         NST3      NSTART(3)=1+NF+NA
C         NST4      NSTART(4)=1+NF+NA+NB
C         NST5      NSTART(5)=1+NF+NA+NB+ND
C       R1        - VECTOR CONTAINING THE DIAGONAL ELEMENTS FROM THE   (I)
C                   COVARIANCE MATRIX OF THE PARAMETER NOISE SIZE(NPDIM)
C       Y         - LAST OUTPUT VALUE                                  (I)
C       WT        - WEIGHTING PARAMETER (FORGETTING PROFILE)           (I)
C       RES       - RESIDUAL                                           (O)
C       ISAMP     - SAMPLING TIME                                      (I)
C       IFIRST    - STARTING TIME                                      (I)
C                   A)  LFILT=.TRUE.   FOR FILTERING OLD SIGNAL VALUES
C                   B)  IV=.TRUE.      FOR INSTRUMENTAL VAR. TO BE USED
C       RLIM      - LIMITATION ON THE ABSOLUTE VALUE OF THE RESIDUALS, (I)
C                   IF NEGATIVE  NO LIMITATION.
C       SA        - LOGICAL PARAMETER                                  (I)
C                   SA=.FALSE.  THE PARAMETER VECTOR AND THE COVARIANCE
C                               MATRIX ARE UPDATED BY USE OF EQ. (1)-(4)
C                   SA=.TRUE.   STOCHASTIC APPROXIMATION  EQ. (5)-(7)
C       IV        - LOGICAL PARAMETER                                  (I)
C                   IV=.FALSE.  ZV=WPSI IN EQ. (1)-(4)
C                   IV=.TRUE.   THE INSTRUMENTAL VARIABLES METHOD
C                               CAN BE STARTED-UP BY USING THE LS-METHOD
C                               FOR IFIRST NUMBER OF STEPS.
C                               ZV= THE INSTRUMENTAL VARIABLES
C       LFILT     - LOGICAL PARAMETER                                  (I)
C                   LFILT=.FALSE. LIP APPROACH,SUBROUTINE FILT NOT USED
C                   LFILT=.TRUE.  CMA APPROACH,SUBROUTINE FILT USED
C       PRERES    - LOGICAL PARAMETER DEFINING HOW THE RESIDUAL IS     (I)
C                   TO BE COMPUTED.
C                   PRERES=.FALSE.  THE RESIDUAL COMPUTATION IS BASED
C                                   ON THE UPDATED PARAMETER ESTIMATE.
C                   PRERES=.TRUE.   THE RESIDUAL COMPUTATION IS BASED
C                                   ON THE PREVIOUS PARAMETER ESTIMATE.
C       STABCK    - LOGICAL PARAMETER       (STABILITY CHECK)          (I)
C                   STABCK=.FALSE.  NO STABILITY CHECK
C                   STABCK=.TRUE.   THE STABILITY OF THE A AND C POLY-
C                                   NOMIALS (CF.MODEL) IS CHECKED. IF
C                   UNSTABLE THE PARAMETER VECTOR IS ADJUSTED HALFWAY
C                   TOWARDS THE LAST VALUE AND STABILITY CHECKED
C                   AGAIN. THIS IS REPEATED UNTIL THE POLYNOMIAL HAS
C                   ALL ZEROES INSIDE THE UNIT CIRCLE. IF THE LOGICAL
C                   PARAMETER LSPFZN=.TRUE. THEN THE POLYNOMIAL IS
C                   FACTORED TO GIVE A POLYNOMIAL WITH ZEROS INSIDE
C                   THE UNIT CIRCLE BY USING THE SUBROUTINE SPFZN.
C       LSPFZN    - LOGICAL PARAMETER   (SEE STABCK)                   (I)
C       OLDTH     - VECTOR OF OLD PARAMETER ESTIMATES  SIZE(NPDIM)   (I/O)
C       WPSI      - VECTOR CONTAINING OLD (FILTERED) SIGNAL VALUES
```

```
C                    CORRESPONDING TO THE PARAMETERS   SIZE(NPDIM)
C       W1       - WORK VECTOR    SIZE(NPDIM)
C       W2       - WORK VECTOR    SIZE(NPDIM)
C       W3       - WORK VECTOR    SIZE(NPDIM)
C       PX       - WORK MATRIX    SIZE(NPDIM,NPDIM)
C       RX       - WORK MATRIX    SIZE(NPDIM,NPDIM)
C       DELTA    - ADJUSTMENT TO IMPROVE THE STABILITY OF THE          (I)
C                  ALGORITHM.
C       WRES     - WEIGHTED RESIDUAL                                   (I)
C       ICK      - CONTROL PARAMETER   STABILITY CHECK                 (O)
C                  IF THE SUBROUTINE STABB HAS FAILED TO ADJUST THE
C                  THETA-VECTOR TO STABILITY  ICK AND THETA ARE
C                  RETURNED:    ICK=10
C                               THETA=OLDTH
C
C
C-----------------------------------------------------------------------
C
C
C       CHARACTERISTICS:
C       ================
C
C       SUBROUTINES REQUIRED:                   (LIBRARY)
C       ---------------------
C       FILT                                     SCAPRO
C       UPDATE                                   RMOVE
C       STABB                                    POLMU
C                                                INPOL
C                                                SPFZN
C
C
C       REF. H. GUNNARSSON:     RECURSIVE IDENTIFICATION USING A
C                               GENERAL MODEL STRUCTURE. A PROGRAM
C                               PACKAGE AND SIMULATIONS.
C
C-----------------------------------------------------------------------
C
C
C       SUBROUTINE GEALRI(S,ISDIM1,P,NPDIM,IPDIM,THETA,PHI,Z,TEE,KD,N,
C    *  NSTART,R1,Y,WT,RES,ISAMP,IFIRST,RLIM,SA,IV,LFILT,PRERES,STABCK,
C    *  LSPFZN,OLDTH,WPSI,W1,W2,W3,DELTA,WRES,ICK,PX,RX)
C
C       LOGICAL SA,IV,LFILT,PRERES,STABCK
C
C       DIMENSION S(ISDIM1,1),P(IPDIM,1),TEE(ISDIM1,1),THETA(1),R1(1),
C    *  N(1),NSTART(1),Z(1),PHI(1),OLDTH(1),WPSI(1),W1(1),W2(1),W3(1),
C    *  PX(IPDIM,1),RX(IPDIM,1)
C
C-----------------------------------------------------------------------
C
        SUBROUTINE GEALR
C
        LOGICAL SA,IV,LFILT,PRERES,STABCK,LSPFZN
```

Subsystem IDHG

```
C
          DIMENSION W1(10),W2(10),W3(10),TEE(5,10),N(5),NSTART(5),
     *    PX(10,10),RX(10,10)
C
          COMMON KDUM1(435),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     *    ICK,ISAMP,IFIRST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,ISDIM1,
     *    ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     *    NPS1,NPS2,NP1,INVAU,IDM35,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     *    DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *    PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *    P1(10,10),TK(100)
C
          EQUIVALENCE (TH1(1),W1(1)),(Z2(1),W2(1)),(Z2(11),W3(1))
          EQUIVALENCE (FI(1),TEE(1,1)),(PX(1,1),P1(1,1)),(RX(1,1),TK(1))
          EQUIVALENCE (N(1),NF),(NSTART(1),NST1)
C
C
C         FORM PHI  (FROM S)
C
          J1=0
          DO 20 I=1,ISDIM1
          K1=0
          N1=N(I)
          IF(N1.LE.0) GO TO 20
          IF(I.EQ.3) K1=KD
          DO 10 J=1,N1
          J1=J1+1
          K2=K1+J
          PHI(J1)=S(I,K2)
   10     WPSI(J1)=PHI(J1)
   20     CONTINUE
C
C
C         COMPUTE RESIDUAL
C
          RES=Y-SCAPRO(PHI(1),1,TH(1),1,NPDIM)
          IF(RLIM.GT.0.AND.ABS(RES).GT.RLIM) RES=SIGN(RLIM,RES)
C
          IF(LFILT.AND.ISAMP.GT.IFIRST) CALL FILT(S,TEE,TH,N,WPSI,KD,
     *    NSTART,ISDIM1,W1,W2,W3)
C
          IF(STABCK) CALL RMOVE(TH(1),1,OLDTH(1),1,NPDIM)
C
          CALL UPDATE(P,NPDIM,IPDIM,WPSI,Z,TH,RES,R1,DELTA,WT,SA,IV,W1,
     *    W2,W3,WRES,ISAMP,IFIRST)
C
          IF(STABCK) CALL STABB(TH,OLDTH,N,NSTART,NPDIM,LSPFZN,Z,W1,
     *    W2,W3,ICK,IPDIM,PX,RX)
C
          IF(PRERES) GO TO 30
          RES=Y-SCAPRO(PHI(1),1,TH(1),1,NPDIM)
          IF(RLIM.GT.0.AND.ABS(RES).GT.RLIM) RES=SIGN(RLIM,RES)
C
```

Subsystem IDHG

```
   30    RETURN
         END
```

```
C       NAME: FILT                                    NUMBER:
C       ----                                          -------
C
C       SUBTITLE: FILTERING OF OLD SIGNAL VALUES FOR
C       --------  RECURSIVE IDENTIFICATION ALGORITHMS
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS: RECURSIVE IDENTIFICATION
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON            DATE: 1978-06-22
C       -----------                                   ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                     VERSION:
C       --------                                      -------
C
C
C
C-----------------------------------------------------------------------
C
C
C       PURPOSE: TO FILTER OLD SIGNAL VALUES, CORRESPONDING TO THE
C       =======  PARAMETERS, TO GET A VECTOR PSI. PSI IS USED TO
C       UPDATE THE COVARIANCE MATRIX IN THE CRITERION
C       MINIMIZATION APPROACH TO RECURSIVE IDENTIFICATION ALGORITHMS.
C
C       THE MODEL USED FOR THE IDENTIFICATION IS:
C
C                      B           1+C
C       (1+F)Y(T)=  ---*U(T-KD)+----*E(T)
C                     1+A          1+D
C
C                    -1        -1        -1        -1        -1      -1
C       WHERE  F=F(Q   ), A=A(Q   ), B=B(Q   ), D=D(Q   ), C=C(Q   )  (Q    IS
C       THE BACKWARD SHIFT OPERATOR)  ARE POLYNOMIALS OF ORDERS NF,NA,
C       NB,ND,NC AND F(0)=0,A(0)=0 ETC.
C
C
C       THE OLD SIGNAL VALUES ARE TO BE STORED AS FOLLOWS:
C
C           [-Y(T-1),-Y(T-2)......................]
C           [-Z(T-1),-Z(T-2)......................]
C       S = [ U(T-1), U(T-1)......................]
C           [-V(T-1),-V(T-2)......................]
C           [ E(T-1), E(T-2)......................]
```

Subsystem IDHG

```
C
C
C       THE PSI-VECTOR IS DEFINED RECURSIVELY AS:
C
C       (1+C)*YPSI(T)=(1+D)*YPHI(T)
C       (1+C)*(1+A)*ZPSI(T)=(1+D)*ZPHI(T)
C       (1+C)*(1+A)*UPSI(T)=(1+D)*UPHI(T)
C       (1+C)*VPSI(T)=VPHI(T)
C       (1+C)*EPSI(T)=EPHI(T)
C
C
C       YPSI(T)=[YW(T-1)...YW(T-NF)]  AND  YPHI(T)=[-Y(T-1)...-Y(T-NF)]
C       ETC. YPSI,ZPSI ETC. FORM THE PSI VECTOR WHICH CORRESPONDS TO THE
C       PHI-VECTOR (WHICH IT REPLACES WHEN UPDATING THE COVARIANCE
C       MATRIX) AND THE PARAMETER VECTOR THETA I.E.
C
C              I YW(T-1)  I          I -Y(T-1)  I          I F(1)  I
C              I    :     I          I     :    I          I   :   I
C              I YW(T-NF) I          I -Y(T-NF) I          I F(NF) I
C              I ZW(T-1)  I          I -Z(T-1)  I          I A(1)  I
C              I    :     I          I     :    I          I   :   I
C              I ZW(T-NA) I          I -Z(T-NA) I          I A(NA) I
C              I UW(T-1)  I          I  U(T-1)  I          I B(1)  I
C       PSI(T)=I    :     I  PHI(T)=I     :    I  THETA=I   :   I
C              I UW(T-NB) I          I  U(T-NB) I          I B(NB) I
C              I VW(T-1)  I          I -V(T-1)  I          I D(1)  I
C              I    :     I          I     :    I          I   :   I
C              I VW(T-ND) I          I -V(T-ND) I          I D(ND) I
C              I EW(T-1)  I          I  E(T-1)  I          I C(1)  I
C              I    :     I          I     :    I          I   :   I
C              I EW(T-NC) I          I  E(T-NC) I          I C(NC) I
C
C
C       THE OLD FILTERED SIGNAL VALUES ARE STORED IN THE TEE-MATRIX
C       IN THE FOLLOWING MANNER:
C
C             I YW(T-1),YW(T-2)............... I
C             I ZW(T-1),ZW(T-2)............... I
C       TEE = I UW(T-1),UW(T-2)............... I
C             I VW(T-1),VW(T-2)............... I
C             I EW(T-1),EW(T-1)............... I
C
C
C
C
C       ARGUMENTS:
C       ---------
C
C       S       - MATRIX CONTAINING OLD SIGNAL VALUES CF. THE MODEL   (I)
C                 GIVEN ABOVE, SIZE (ISDIM,  )
C       TEE     - MATRIX CONTAINING OLD FILTRATED SIGNAL VALUES     (I/O)
C                 SIZE (ISDIM, )
C       THETA   - VECTOR OF PARAMETER ESTIMATES SIZE(NF+NA+NB+ND+NC) (I)
```

```
C       N        - VECTOR CONTAINING NUMBER OF PARAMETERS            (I)
C                  N(1)=NF
C                  N(2)=NA
C                  N(3)=NB
C                  N(4)=ND
C                  N(5)=NC
C       PSI      - VECTOR OF OLD FILTERED SIGNAL VALUES CORRESPONDING (O)
C                  TO THE PARAMETERS  SIZE (NF+NA+NB+ND+NC)
C       KD       - TIME DELAY                                        (I)
C       NSTART   - VECTOR CONTAINING THE STARTING POINTS OF THE      (I)
C                  PARAMETERS IN THETA
C                  NSTART(1)=1
C                  NSTART(2)=1+NF
C                  NSTART(3)=1+NF+NA
C                  NSTART(4)=1+NF+NA+NB
C                  NSTART(5)=1+NF+NA+NB+ND
C       ISDIM    - DECLARED FIRST DIMENSION OF S AND TEE             (I)
C       W1       - WORK VECTOR
C       W2       - WORK VECTOR
C       W3       - WORK VECTOR
C
C-----------------------------------------------------------------------
C
C       LIBRARY SUBROUTINES REQUIRED:   RMOVE   POLMU   SCAPRO
C       ----------------------------
C
C       REF.: H. GUNNARSSON, RECURSIVE IDENTIFICATION USING A GENERAL
C                            MODEL STRUCTURE. A PROGRAM PACKAGE AND
C                            SIMULATIONS.
C-----------------------------------------------------------------------
C
        SUBROUTINE FILT(S,TEE,THETA,N,PSI,KD,NSTART,ISDIM,W1,W2,W3)
C
        DIMENSION S(ISDIM,1),TEE(ISDIM,1),THETA(1),N(1),PSI(1)
        DIMENSION NSTART(1),W1(1),W2(1),W3(1)
C
C
        J1=0
C
        NA=N(2)
        ND=N(4)
        NC=N(5)
C
        NAST=NSTART(2)
        NDST=NSTART(4)
        NCST=NSTART(5)
C
        NU=NA+NC
        NU1=NU-1
C
        NK=MAX0(ND,NA+NC,N(1),N(3))
        NK1=NK-1
C
```

Subsystem IDHG

```
C
      IF(NA.LE.0.OR.NC.LE.0) GO TO 10
      NW1=NA-1
      NW2=NC-1
      CALL RMOVE(THETA(NAST),1,W1(1),1,NA)
      CALL RMOVE(THETA(NCST),1,W2(1),1,NC)
      CALL POLMU(W1,W2,NW1,NW2,W3)
C
C
   10 DO 60 I=1,ISDIM
      N1=N(I)
      K1=0
      IF(N1.LE.0) GO TO 60
      IF(I.EQ.3) K1=KD
      W=S(I,K1+1)
C
      GO TO(30,20,20,40,40),I
C
   20 IF(NA.GT.0) W=W-SCAPRO(TEE(I,1),ISDIM,THETA(NAST),1,NA)
      IF(NA.GT.0.AND.NC.GT.0)W=W-SCAPRO(TEE(I,2),ISDIM,W3(1),1,NU1)
   30 IF(ND.GT.0) W=W+SCAPRO(S(I,K1+2),ISDIM,THETA(NDST),1,ND)
   40 IF(NC.GT.0) W=W-SCAPRO(TEE(I,1),ISDIM,THETA(NCST),1,NC)
C
      CALL RMOVE(TEE(I,NK1),-ISDIM,TEE(I,NK),-ISDIM,NK1)
      TEE(I,1)=W
C
      DO 50 J=1,N1
      J1=J1+1
   50 PSI(J1)=TEE(I,J)
   60 CONTINUE
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: UPDATE                                  NUMBER:
C       ----                                          ------
C
C       SUBTITLE:
C       --------
C       COVARIANCE MATRIX AND  PARAMETER VECTOR UPDATING
C       FOR RECURSIVE IDENTIFICATION ALGORITHMS.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS: RECURSIVE IDENTIFICATION
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON     DATE: 1978-03-17
C       -----------                            ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                              VERSION:
C       --------                               -------
C
C
C
C-----------------------------------------------------------------------
C
C
C
C       PURPOSE: TO UPDATE THE COVARIANCE MATRIX P AND THE PARAMETER
C       -------  VECTOR THETA ACCORDING TO THE EQUATIONS GIVEN BELOW.
C
C                                                  T
C                 [         P(T)*Z(T+1)*PHI(T+1)  *P(T)]
C       P(T+1) = [P(T) - ---------------------------]/WT         (1)
C                 [          WT+PHI (T+1)*P(T)*Z(T+1)  ]
C
C       WHERE   MEANS THE TRANSPOSED MATRIX
C
C
C       THE DIAGONAL ELEMENTS OF P CAN BE ADJUSTED BY
C
C       P[T+1](I,I)=P[T+1](I,I)+[R(I)-DELTA*P[T](I,I)**2]/WT     (2)
C
C
C
C       THETA(T+1)=THETA(T)+RK(T+1)*RES                          (3)
C
C       WHERE:
```

Subsystem IDHG

```
C
C                      P(T)*Z(T+1)
C      RK(T+1) = -------------------------                      (4)
C                WT+PHI  (T+1)*P(T)*Z(T+1)
C
C
C      REMARK: Z = PHI EXCEPT WHEN IV = .TRUE.
C
C                                T
C      WRES = RES/[WT+PHI(T+1)  *P(T)*Z(T+1)]                   (5)
C
C
C      STOCHASTIC APPROXIMATION EQUATIONS:
C      ----------------------------------
C                                          T
C      P[1,1](T+1)=WT*P[1,1](T)+PHI(T+1)  *PHI(T+1)  + DELTA    (6)
C
C      THETA(T+1)=THETA(T)+RES*PHI(T+1)/P[1,1](T+1)             (7)
C                                T
C      WRES = RES/[WT+PHI(T+1)  *PHI(T+1)/P(1,1)]               (8)
C
C
C
C
C      ARGUMENTS:
C      ---------
C
C      UPDATE(P,NPDIM,IPDIM,PHI,Z,THETA,RES,R,DELTA,WT,SA,IV,RK,W2,
C      W3,WRES,ISAMP,IFIRST)
C
C      P        - THE COVARIANCE MATRIX, SIZE (NPDIM*NPDIM),        (I/O)
C                 DIMENSIONED (IPDIM,   )
C      NPDIM    - ACTUAL DIMENSION OF P, ACTUAL NUMBER OF PARAMETERS (I)
C      IPDIM    - DECLARED FIRST DIMENSION OF P                      (I)
C      PHI      - VECTOR CONTAINING OLD SIGNAL VALUES CORRESPONDING  (I)
C                 TO THE PARAMETERS  SIZE  (NPDIM)
C      Z        - VECTOR CONTAINING OLD SIGNAL VALUES FOR THE        (I)
C                 INSTRUMENTAL VARIABLES  SIZE  (NPDIM)
C      THETA    - VECTOR OF PARAMETER ESTIMATES  SIZE  (NPDIM)      (I/O)
C      RES      - RESIDUAL                                           (I)
C      R        - VECTOR CONTAINING THE DIAGONAL ELEMENTS FROM THE   (I)
C                 COVARIANCE MATRIX OF THE PARAMETER NOISE  SIZE (NPDIM)
C      DELTA    - ADJUSTMENT TO IMPROVE STABILITY OF THE ALGORITHM   (I)
C      WT       - WEIGHTING PARAMETER                                (I)
C      SA       - LOGICAL VARIABLE   STOCHASTIC APPROXIMATION        (I)
C                 SA=.FALSE.  THE PARAMETER VECTOR AND THE COVARIANCE
C                             MATRIX ARE UPDATED BY USE OF EQUATIONS
C                             (1)-(4)
C                 SA=.TRUE    THE PARAMETER VECTOR AND THE COVARIANCE
C                             MATRIX ARE UPDATED BY THE METHOD OF
C                             STOCHASTIC APPROXIMATION.
C       IV      - LOGICAL VARIABLE   INSTRUMENTAL VARIABLES METHOD   (I)
C                 IV=.FALSE. Z=PHI IN EQUATIONS (1)-(4)
```

```
C                    IV=.TRUE.  P AND THETA ARE UPDATED BY USE OF Z
C                               THE INSTRUMENTAL VARIABLES METHOD CAN BE
C                               STARTED-UP BY USING THE LS-METHOD FOR
C                               IFIRST NUMBER OF STEPS.
C       RK       - WORK VECTOR  SEE EQ.(3)  SIZE (NPDIM)
C       W2       - WORK VECTOR  SIZE (NPDIM)
C       W3       - WORK VECTOR  SIZE (NPDIM)
C       WRES     - WEIGHTED RESIDUAL                                    (O)
C       ISAMP    - SAMPLING TIME                                        (I)
C       IFIRST   - STARTING TIME FOR THE IV-METHOD                      (I)
C
C
C----------------------------------------------------------------------
C
C
C       CHARACTERISTICS:
C       ---------------
C
C       LIBRARY SUBROUTINES REQUIRED:
C       SCAPRO
C
C
C----------------------------------------------------------------------
C
        SUBROUTINE UPDATE(P,NPDIM,IPDIM,PHI,Z,THETA,RES,R1,DELTA,WT,SA,
     *  IV,RK,W2,W3,WRES,ISAMP,IFIRST)
C
        LOGICAL SA,IV
        DIMENSION P(IPDIM,1),PHI(1),Z(1),THETA(1),R1(1),RK(1),W2(1),
     *  W3(1)
C
        IF(SA) GO TO 120
C
C****** THE DELTA ADJUSTMENT COMPUTATION: ******
        IF(DELTA.LE.0.0) GO TO 20
        DO 10 I=1,NPDIM
   10   W3(I)=DELTA*P(I,I)**2/WT
C
C****** UPDATE P-MATRIX ******
   20   DO 30 I=1,NPDIM
   30   W2(I)=SCAPRO(PHI(1),1,P(1,I),1,NPDIM)
C
        IF(IV.AND.IFIRST.LT.ISAMP) GO TO 50
C
C****** THE SYMMETRIC CASE ******
        GAMMA=WT+SCAPRO(W2(1),1,PHI(1),1,NPDIM)
        DO 40 I=1,NPDIM
        RK(I)=W2(I)/GAMMA
        DO 40 J=1,I
        P(I,J)=(P(I,J)-RK(I)*W2(J))/WT
   40   P(J,I)=P(I,J)
        GO TO 70
C
```

Subsystem IDHG

```
C****** THE UNSYMMETRIC CASE ******
   50   GAMMA=WT+SCAPRO(W2(1),1,Z(1),1,NPDIM)
        DO 60 I=1,NPDIM
        RK(I)=SCAPRO(P(I,1),IPDIM,Z(1),1,NPDIM)/GAMMA
        DO 60 J=1,NPDIM
   60   P(I,J)=(P(I,J)-RK(I)*W2(J))/WT
C
C
C****** THE R1-ADJUSTMENT TO P ******
   70   DO 80 I=1,NPDIM
   80   P(I,I)=P(I,I)+R1(I)/WT
C
C****** THE DELTA-ADJUSTMENT TO P ******
        IF(DELTA.LE.0.0) GO TO 100
        DO 90 I=1,NPDIM
   90   P(I,I)=P(I,I)-W3(I)
C
C
C****** UPDATE PARAMETER ESTIMATES ******
  100   DO 110 I=1,NPDIM
  110   THETA(I)=THETA(I)+RES*RK(I)
        WRES=RES/GAMMA
        RETURN
C
C
C****** STOCHASTIC APPROXIMATION VERSION ******
  120   VN=SCAPRO(PHI(1),1,PHI(1),1,NPDIM)
        P(1,1)=WT*P(1,1)+VN+DELTA
        DO 130 I=1,NPDIM
  130   THETA(I)=THETA(I)+RES*PHI(I)/P(1,1)
        WRES=RES/(WT+VN/P(1,1))
        RETURN
        END
```

Subsystem IDHG

```
C       NAME: STABB                                  NUMBER:
C       ----                                         ------
C
C       SUBTITLE: STABILITY CHECK
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                  ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                    VERSION:
C       ---------                                    -------
C
C
C-----------------------------------------------------------------------
C
C
C       PURPOSE: TO CHECK THE STABILITY OF THE A AND C POLYNOMIALS
C       =======  (CF. THE MODEL GIVEN BELOW) FOR USE IN RECURSIVE
C       IDENTIFICATION ALGORITHMS. IF EITHER POLYNOMIAL IS UNSTABLE
C       THE PROGRAM REACTS IN ONE OF TWO WAYS:
C
C       A)  LSPFZN=.FALSE.  THE PARAMETER VECTOR IS ADJUSTED HALFWAY
C                           TOWARDS THE LAST VALUE (OLDTH) AND STABILITY
C                           CHECKED AGAIN. THIS IS REPEATED UNTIL ALL
C                           ZEROES LIE INSIDE THE UNIT CIRCLE.
C
C       B)  LSPFZN=.TRUE.   THE POLYNOMIAL IS FACTORED TO GIVE A
C                           POLYNOMIAL WITH ZEROES INSIDE THE UNIT
C                           CIRCLE BY USING THE SUBROUTINE SPFZN
C
C
C       THE MODEL USED FOR THE IDENTIFICATION IS:
C
C
C                    B           1+C
C       (1+F)*Y(T)=---*U(T-KD)+---*E(T)
C                   1+A          1+D
C
C
C                     -1          -1          -1          -1          -1      -1
C       WHERE  F=F(Q  ), A=A(Q  ), B=B(Q  ), D=D(Q  ), C=C(Q  )  (Q   IS
C       THE BACKWARD SHIFT OPERATOR) ARE POLYNOMIALS OF ORDERS NF,NA,NB,
C       ND,NC AND F(0)=0, A(0)=0 ETC.
C
```

```
C       ARGUMENTS:
C       ---------
C       STABB(THETA,OLDTH,N,NSTART,NPDIM,LSPFZN,DIFF,W1,W2,W3,ICK,IPDIM,
C     * PX,RX)
C
C       THETA    - VECTOR OF PARAMETER ESTIMATES  SIZE(NPDIM)        (I/O)
C       OLDTH    - VECTOR OF OLD PARAMETER ESTIMATES  SIZE(NPDIM)      (I)
C       N        - VECTOR CONTAINING NUMBER OF PARAMETERS              (I)
C                  N(1)=NF
C                  N(2)=NA
C                  N(3)=NB
C                  N(4)=ND
C                  N(5)=NC
C       NSTART   - VECTOR CONTAINING THE STARTING POINTS OF THE        (I)
C                  PARAMETERS IN THETA
C                  NSTART(1)=1
C                  NSTART(2)=1+NF
C                  NSTART(3)=1+NF+NA
C                  NSTART(4)=1+NF+NA+NB
C                  NSTART(5)=1+NF+NA+NB+ND
C       IPDIM    - DECLARED FIRST DIMENSION OF PX AND RX               (I)
C       NPDIM    - ACTUAL NUMBER OF PARAMETERS  (=DIM. COVM)           (I)
C       LSPFZN   - LOGICAL VARIABLE                                    (I)
C       DIFF     - WORK VECTOR      SIZE (NPDIM)
C       W1       - WORK VECTOR      SIZE (NPDIM)
C       W2       - WORK VECTOR      SIZE (NPDIM)
C       W3       - WORK VECTOR      SIZE (NPDIM)
C       PX       - WORK MATRIX      SIZE (NPDIM*NPDIM)
C       RX       - WORK MATRIX      SIZE (NPDIM*NPDIM)
C       ICK      - CONTROL PARAMETER                                   (O)
C                  ICK IS RETURNED ICK=10 IF THE SUBROUTINE HAS
C                  FAILED TO ADJUST THE THETA VECTOR TO STABILITY.
C                  IN THAT CASE THETA IS RETURNED THETA=OLDTH.
C
C-----------------------------------------------------------------------
C
C       LIBRARY SUBROUTINES REQUIRED:
C       ----------------------------
C       RMOVE               INPOL
C       NORM                SPFZN
C
C-----------------------------------------------      ------------------
C
        SUBROUTINE STABB(THETA,OLDTH,N,NSTART,
     *  W2,W3,ICK,IPDIM,PX,RX)
C
        LOGICAL LSPFZN,INPOL
C
        DIMENSION THETA(1),OLDTH(1),N(1),NST
        DIMENSION W3(1),PX(IPDIM,1),RX(IPDI
C
        RMIN=RMACON(1)
        TEST=100.*RMIN
```

Subsystem IDHG

```
      ICK=0
      S1=1.
C
      DO 10 I=1,NPDIM
   10 DIFF(I)=THETA(I)-OLDTH(I)
C
C
C
      DO 60 J=2,5,3
      IF(N(J).LE.0) GO TO 60
      NPAR=N(J)
      NST=NSTART(J)
      K=NST+NPAR-1
C
      IF(LSPFZN) GO TO 50
C
C
C****** ALTERNATIVE A  SEE ABOVE ******
   20 W1(1)=1.
      CALL RMOVE(THETA(NST),1,W1(2),1,NPAR)
      IF(INPOL(W1,NPAR,1.,W2,W3)) GO TO 60
      S1=AMAX1(S1/2.,TEST)
      DO 30 I=1,NPDIM
   30 THETA(I)=THETA(I)-S1*DIFF(I)
C
      DMAX=0.0
      DO 40 I=NST,K
      XX=S1*DIFF(I)/THETA(I)
      X=ABS(XX)
      IF(X.GT.DMAX) DMAX=X
   40 CONTINUE
      IF(DMAX.LT.TEST) GO TO 70
      GO TO 20
C
C
C****** ALTERNATIVE B  SEE ABOVE ******
   50 W1(1)=1.
      CALL RMOVE(THETA(NST),1,W1(2),1,NPAR)
      IF(INPOL(W1,NPAR,1.,W2,W3)) GO TO 60
      CALL SPFZN(W1,W2,CO,NPAR,TEST,ICK,IPDIM,PX,RX)
      IF(ICK.EQ.1) GO TO 70
      CALL RMOVE(W2(1),1,THETA(NST),1,NPAR)
C
C
   60 CONTINUE
      RETURN
C
C
   70 CALL RMOVE(OLDTH(1),1,THETA(1),1,NPDIM)
      ICK=10
      RETURN
      END
```

Subsystem IDHG

```
      SUBROUTINE POLMU(P1,P2,NP1,NP2,P)
C
C     MULTIPLIES TWO REAL POLYNOMIALS
C
C     P1(1)*S**NP1 + P1(2)*S**(NP1-1) + ... + P1(NP1+1)
C     P2(1)*S**NP2 + P2(2)*S**(NP2-1) + ... + P2(NP2+1)
C
C     TO FORM THE PRODUCT
C
C     P(1)*S**NP + P(2)*S**(NP-1) + ... + P(NP+1) , WHERE NP=NP1+NP2
C
C     P1      - VECTOR CONTAINING COEFFICIENTS FOR POLYNOMIAL *1,
C               SIZE NP1+1, (I)
C     P2      - VECTOR CONTAINING COEFFICIENTS FOR POLYNOMIAL *2,
C               SIZE NP2+1, (I)
C     NP1     - DEGREE OF POLYNOMIAL *1, (I)
C     NP2     - DEGREE OF POLYNOMIAL *2, (I)
C     P       - VECTOR RETURNED CONTAINIG COEFFICIENTS FOR THE
C               RESULTING POLYNOMIAL, SIZE (NP1+NP2+1), (O)
C
C     SUBROUTINES REQUIRED:
C             SCAPRO
C
C     AUTHORS ULF BORISSON AND JAN HOLST 1971-03-09
C     REVISED CLAES KALLSTROM 1972-05-17
C     REVISED JAN STERNBY 1975-02-17
C
      DIMENSION P1(1),P2(1),P(1)
C
      NP11=NP1+1
      NP12=NP1+2
      NP21=NP2+1
      NPTVA=NP12+NP2
      NMIN=MIN0(NP1,NP2)
C
      IF(NMIN.EQ.0) GO TO 150
      DO 100 I=1,NMIN
      P(I)=SCAPRO(P1(1),1,P2(I),-1,I)
      P(NPTVA-I)=SCAPRO(P1(NP12-I),1,P2(NP21),-1,I)
100   CONTINUE
C
  150 IF(NMIN.EQ.NP1) GO TO 300
C
      DO 200 I=NP21,NP11
      P(I)=SCAPRO(P1(I-NP2),1,P2(NP21),-1,NP21)
200   CONTINUE
C
      RETURN
C
300   DO 400 I=NP11,NP21
      P(I)=SCAPRO(P1(1),1,P2(I),-1,NP11)
400   CONTINUE
C
```

Subsystem IDHG

```
      RETURN
      END
```

Subsystem IDHG

```
      LOGICAL FUNCTION INPOL(A,N,R, W1,W2)
C
C     TESTS IF ALL ZEROES OF THE REAL POLYNOMIAL
C     A(1)*S**N + A(2)*S**(N-1)+...+A(N+1)
C     LIE INSIDE A CIRCLE WITH RADIUS R
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C     REVISED TOMAS SCHONTHAL 1976-07-29
C
C     REFERENCE: K.J. ASTROM : INTRODUCTION TO STOCHASTIC
C                              CONTROL THEORY, CHAP.5
C
C     INPOL    - RETURNED .TRUE. IF ALL ZEROES LIE INSIDE THE
C                CIRCLE
C                RETURNED .FALSE. IF AT LEAST ONE ZERO LIES
C                OUTSIDE OR ON THE CIRCLE
C     A        - VECTOR CONTAINING THE COEFFICIENTS OF THE
C                POLYNOMIAL SIZE (N+1)
C     N        - DEGREE PF THE POLYNOMIAL
C     R        - RADIUS OF THE CIRCLE
C     W1       - WORK VECTOR, SIZE (N+1)
C     W2       - WORK VECTOR, SIZE (N+1)
C
C     NOTE: A ZERO ON THE CIRCLE MAY UNDER SOME CIRCUMSTANCES
C     CAUSE INPOL TO BE .TRUE. DUE TO THE BINARY REPRESENTATION
C
C     SUBROUTINES REQUIRED:
C              RMOVE
C
C
      DIMENSION A(1), W1(1),W2(1)
C
      NA=N
      NN=NA+1
      CALL RMOVE(A,1, W1,1, NN)
      RR=R
      IF(W1(1).GE. 0.0) GOTO 5
C
      RR=-RR
      W1(NN)=-W1(NN)
C
5     DO 10 I=1,NA
      J=NN-I
      W1(J)=W1(J)*RR
      RR=RR*R
10    CONTINUE
C
20    S=W1(NN)/W1(1)
      W2(1)=W1(1)-S*W1(NN)
      IF(W2(1).LE. 0.0) GOTO 70
C
      IF(NA.LE.1) GOTO 60
C
```

Subsystem IDHG

```
      DO 50 I=2,NA
      W2(I)=W1(I)-S*W1(NN-I+1)
50    CONTINUE
C
      CALL RMOVE(W2,1, W1,1, NA)
      NN=NA
      NA=NA-1
      GO TO 20
C
60    INPOL=.TRUE.
      RETURN
C
70    INPOL=.FALSE.
      RETURN
      END
```

Subsystem IDHG

```
      SUBROUTINE SPFZN(B,C,CO,N,EPS,IND,IW,P,R)
C     THIS SUBROUTINE DETERMINES A POLYNOMIAL
C        C(Z)= CO*(Z**N+C(1)*Z**(N-1)+...+C(N))
C     WITH ZEROES INSIDE THE UNIT CIRCLE SUCH THAT
C        C(Z)*C{Z)=B(Z)*B{Z)
C     WHERE
C        B(Z)=Z**N+B(1)*Z**(N-1)+...+B(N)
C
C     AND B DENOTES THE RECIPROCIAL POLYNOMIAL
C
C     REFERENCE K.J. ASTROM ∀ SPECTRAL FACTORIZATION ALGORITHM
C
C     AUTHOR K.J. ASTROM 1971-12-29
C
C        B   - VECTOR CONTAINING COEFFICIENTS OF POLYNOMIAL
C              TO BE FACTORED. IT IS ASSUMED THAT LEADING
C              COEFFICIENT OF B IS 1.
C        C   - VECTOR CONTAINING COEFFICIENTS OF THE FACTORED
C              POLYNOMIAL AS GIVEN ABOVE. NOTICE THAT
C              THE COEFFICIENTS ARE NORMALIZED BY CO
C        CO  - COEFFICIENT OF LEADING TERM OF POLYNOMIAL C
C        N   - DEGREE OF POLYNOMIALS B AND C (MAX 15)
C        EPS- TEST QUANTITY TO STOP ITERATION OF RICCATI EQUATION
C        IND- INDICATOR RETURNED AS 1 IF THE ITERATION
C              DOES NOT CONVERGE RETURNED AS -NLOOP OTHERWISE
C
C     SUBROUTINES REQUIRED
C        NORM
C
      DIMENSION B(1),C(1)
      DIMENSION P(IW,1),R(IW,1)
C
      NLOP=1000
      NP1=N+1
      DO 10 I=1,NP1
      DO 10 J=1,NP1
      R(I,J)=0.0
10    P(I,J)=0.0
      DO 11 I=1,N
11    P(I,I)=1.
      NLOOP=0
C
C     MAIN LOOP COMPUTE SOLUTION OF RICCATI EQUATION
C
20    RI=1./(1.+P(1,1))
      NLOOP=NLOOP+1
      DO 21 I=1,N
      DO 21 J=1,N
21    R(I,J)=P(I,J)
      DO 22 I=1,N
      DO 22 J=1,N
22    P(I,J)=B(I)*B(J)+R(I+1,J+1)-(R(I+1,1)+B(I))*(R(J+1,1)+B(J))*RI
      DO 25 I=1,N
```

Subsystem IDHG

```
      DO 25 J=1,N
25    R(I,J)=P(I,J)-R(I,J)
C
C     TEST FOR STEADY STATE
C
      CALL NORM(R,N,IW,RNORM)
      CALL NORM(P,N,IW,PNORM)
      IF(RNORM-EPS*PNORM) 24,24,23
24    IND=-NLOOP
      GO TO 29
23    IF(NLOOP-NLOP) 20,20,28
28    IND=1
C
C     COMPUTE C
C
29    T=1.+P(1,1)
      RI=1./T
      DO 30 I=1,N
30    C(I)=(P(I+1,1)+B(I))*RI
      CO=SQRT(T)
      RETURN
      END
```

Subsystem IDHG

```
      SUBROUTINE NORM(A,N,IA,S)
C
C     THE SUBROUTINE COMPUTES THE MINIMAXNORM OF A WHERE
C     A=N*N-MATRIX.
C     AUTHOR, K.MORTENSSON 1967-07-31.
C
C     A- MATRIX OF ORDER N*N.
C     N- ORDER OF THE MATRIX A (NO MAX, MIN 1).
C     IA- DIMENSION PARAMETER.
C     S- RESULTING NORM.
C
C     SUBROUTINE REQUIRED
C            NONE
C
      DIMENSION A(1,1)
C
      S=0.
      S1=0.
      DO 20 J=1,N
      R=0.
      DO 10 I=1,N
      R=R+ABS(A(I,J))
10    CONTINUE
      IF(R-S1) 20,20,15
15    S1=R
20    CONTINUE
C
C     S1 NOW CONTAINS MAX OVER THE COLUMNS.
C
      DO 40 I=1,N
      R=0.
      DO 30 J=1,N
      R=R+ABS(A(I,J))
30    CONTINUE
      IF(R-S) 40,40,35
35    S=R
40    CONTINUE
C
C     S NOW CONTAINS MAX OVER THE ROWS.
C
      IF(S-S1) 60,60,50
50    S=S1
60    RETURN
      END
```

Subsystem IDHG

```
C       NAME: EKF                                       NUMBER:
C       ----                                            ------
C
C       SUBTITLE: ONE STEP OF A RECURSIVE EKF
C       --------  IDENTIFICATION ALGORITHM
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS: RECURSIVE EKF
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE: 1978-10-12
C       -----------                                     ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                       VERSION:
C       --------                                        -------
C
C
C-----------------------------------------------------------------------
C
C
C
C
C       PURPOSE: TO PERFORM ONE STEP OF THE RECURSIVE EKF IDENTIFICATION
C       =======  ALGORITHM ACCORDING TO THE EQUATIONS GIVEN BELOW.
C
C       CONSIDER A SINGLE-INPUT, SINGLE-OUTPUT SYSTEM DESCRIBED BY:
C
C       X(T+1) = A(TH(T)) * X(T) + B(TH(T)) * U(T-KD) + K(TH(T)) * E(T)
C
C       Y(T) = C(TH(T)) * X(T) + E(T)
C
C       WITH THE PARTICULAR PARAMETRIZATION:
C
C       A(TH(T)) = FF + (V+K)*TH(T)∠H
C
C       B(TH(T)) = B
C
C       K(TH(T)) = K
C
C       C(TH(T)) = TH(T)∠ H
C
C       WHERE   DENOTES THE TRANSPOSED MATRIX AND FF, V, K, H AND B
C       ARE FIXED AND CHOSEN AS :
C
```

Subsystem IDHG

```
C                                                                                    I 0 I
C              I Y(T-1)   I      I 0 I               I 1 I                           I : I
C              I    :     I      I : I               I 0 I                           I : I
C              I    :     I      I : I               I : I                           I : I
C              I Y(T-NF)  I      I 0 I               I : I                           I : I
C              I U(T-1)   I      I 1 I  <NF+1        I : I                           I : I
C              I    :     I      I 0 I               I : I                           I : I
C      X(T) =  I    :     I    B=I : I             K=I : I                         V=I : I
C              I    :     I      I : I               I : I                           I : I
C              I U(T-NB)  I      I : I               I 0 I                           I 0 I
C              I E(T-1)   I      I : I               I 1 I   < NF+NB+1 >             I-1 I
C              I    :     I      I : I               I 0 I                           I 0 I
C              I    :     I      I : I               I : I                           I : I
C              I E(T-NC)  I      I 0 I               I 0 I                           I 0 I
C
C
C
C                                                I                  I -F(1)   I
C           I 0.....0                            I                  I   :     I
C           I 10....0                            I                  I   :     I
C           I 0     :       0           0        I  NF ROWS         I   :     I
C           I :     :                            I                  I -F(NF)  I
C           I 0...010                            I                  I  B(1)   I
C           I             0.....0                I                  I   :     I
C           I             10....0                I                  I   :     I
C       FF= I    0        0     :       0        I  NB ROWS   TH(T) = I   :     I
C           I             :     :                I                  I  B(NB)  I
C           I             0   010                I                  I  C(1)   I
C           I                       0.....0      I                  I   :     I
C           I                       10....0      I                  I   :     I
C           I    0          0       0     :      I  NC ROWS         I   :     I
C           I                       :     :      I                  I  C(NC)  I
C           I                       0...010      I
C
C       H = I
C
C       THIS RESULTS IN THE FOLLOWING ALGORITHM DERIVED FROM THE
C       MODEL (6.1) AND EQUATIONS (8.18) IN REF. 1.
C
C       X(T+1)=(FF+(V+K)*TH(T)}*X(T)+B*U(T-KD)+K*E(T)
C
C       E(T)=Y(T)-TH(T)∠X(T)
C
C       TH(T+1)=TH(T)+L(T)*E(T)
C
C       L(T)=[P2(T)∠TH(T)+P3(T)*X(T)]/ST
C
C       ST=X(T)∠P3(T)*X(T)+2*[X(T)∠P2(T)∠TH(T)]+1
C
C       P2(T+1)=FF*P2(T)+V*[TH(T)∠P2(T)+X(T)∠P3(T)]
C
C       P3(T+1)=P3(T)-L(T)*ST*L(T)±DELTA*P3(T)*P3(T)
C
C       CF:
C       Y(T)+F(1)*Y(T-1)+...+F(NF)*Y(T-NF)=B(1)*U(T-1)+...+B(NB)*U(T-NB)
```

```
                                      +E(T)+C(1)*E(T-1)+...+C(NC)*E(T-NC)
C
C
C      ARGUMENTS:
C      ----------
C      EKF(TH,P2,P3,IPDIM,NPDIM,X,Y,U,RES,DELTA,N,NSTART,RL,WP3X,WP2TH,
C    * WL,PX,S,ISDIM,KD,STABCK,LSPFZN,ICK,OLDTH,WD,W2,W3,W4,RX)
C      TH       - VECTOR OF PARAMETER ESTIMATES  SIZE(NPDIM)          (I/O)
C      P2       - COVARIANCE MATRIX, SIZE(NPDIM,NPDIM)                (I/O)
C                 DIMENSIONED (IPDIM,1)
C      P3       - COVARIANCE MATRIX, SIZE(NPDIM,NPDIM)                (I/O)
C                 DIMENSIONED (IPDIM,1)
C      IPDIM    - DECLARED FIRST DIMENSION OF P                       (I)
C      NPDIM    - ACTUAL DIMENSION OF P, ACTUAL NUMBER OF             (I)
C                 PARAMETERS
C      X        - STATE VECTOR SIZE(NPDIM)                            (I/O)
C      Y        - LAST OUTPUT   CF. MODEL                             (I)
C      U        - LAST INPUT    CF. MODEL                             (I)
C      RES      - RESIDUAL
C      DELTA    - ADJUSTMENT TO IMPROVE THE STABILITY AND THE         (I)
C                 CONVERGENCE RATE OF THE ALGORITHM
C      N        - VECTOR WITH MODEL/PARAMETER INFORMATION             (I)
C      NSTART   - VECTOR WITH PARAMETER INFORMATION                   (I)
C                 N(1)=NF        NSTART(1)=NST1=1
C                 N(2)                NSTART(2)  NOT USED
C                 N(3)=NB        NSTART(3)=NST3=1+NF
C                 N(4)                NSTART(4)  NOT USED
C                 N(5)=NC       NSTART(5)=NST5=1+NF+NB
C      S        - MATRIX CONTAINING OLD SIGNAL VALUES
C                 S(1,-)        OLD -Y
C                 S(2,-)        NOT USED
C                 S(3,-)        OLD U
C                 S(4,-)        NOT USED
C                 S(5,-0        OLD RES
C      ISDIM    - DECLARED FIRST DIMENSION OF S  (=5)                 (I)
C      KD       - THE TIME DELAY                                      (I)
C      STABCK   - LOGICAL PARAMETER    (STABILITY CHECK)              (I)
C                 .FALSE.        NO STABILITY CHECK
C                 .TRUE.        THE STABILITY OF THE C-POLYNOMIAL  IS
C                               CHECKED. THIS IS EQUIVALENT TO A
C                 STABILITY CHECK OF THE MATRIX A(TH)-K(TH)*C(TH).
C                 IF THE POLYNOMIAL IS UNSTABLE THEN THE PARAMETER
C                 VECTOR IS ADJUSTED HALFWAY TOWARDS THE LAST
C                 VALUE AND STABILITY CHECKED AGAIN.THIS IS REPEATED
C                 UNTIL THE POLYNOMIAL HAS ALL ZEROES INSIDE THE UNIT
C                 CIRCLE. IF THE LOGICAL PARAMETER LSPFZN IS .TRUE.
C                 THEN THE POLYNOMIAL IS FACTORED TO GIVE A POLYNOMIAL
C                 WITH ZEROES INSIDE THE UNIT CIRCLE BY USING THE
C                 SUBROUTINE SPFZN.
C      LSPFZN   - LOGICAL PARAMETER  (SEE STABCK)                     (I)
C      ICK      - CONTROL PARAMETER  STABILITY CHECK                  (O)
C                 IF THE SUBROUTINE STABB HAS FAILED TO ADJUST THE
C                 TH-VECTOR TO STABILITY THEN ICK AND TH ARE
C                 RETURNED :    ICK=10
```

Subsystem IDHG

```
C                                  TH=OLDTH
C       PX       - WORK MATRIX       SIZE(NPDIM,NPDIM)
C       RX       -        DO.                  DO.
C       RL       - WORK VECTOR       SIZE(NPDIM)
C       WL       -        DO.                  DO.
C       WP3X     -        DO.                  DO.
C       WP2TH    -        DO                   DO.
C       OLDTH    -        DO.                  DO.
C       WD       -        DO.                  DO.
C       W2       -        DO.                  DO.
C       W3       -        DO.                  DO.
C       W4       -        DO.                  DO.
C
C       SUBROUTINES REQUIRED:    STABB    SCAPRO   RMOVE    INPOL
C       --------------------              SPFZN    NORM
C
C       REFERENCES:
C       ----------
C       1.  L.LJUNG     THE EXTENDED KALMAN FILTER AS A PARAMETER ESTI-
C                       MATOR FOR LINEAR SYSTEMS.REPORT LITH-ISY-I-0154
C       2.  L.LJUNG     CONVERGENCE OF AN ADAPTIVE FILTER ALGORITHM
C                       REPORT  LITH-ISY-I-0120 }
C
C                       } DEPARTMENT OF ELECTRICAL ENGINEERING
C                          LINKOPING UNIVERSITY
C
C-----------------------------------------------------------------------
C
C       SUBROUTINE EKF(TH,P2,P3,IPDIM,NPDIM,X,Y,U,RES,DELTA,N,NSTART,RL,
C    *  WP3X,WP2TH,WL,PX,S,ISDIM,KD,STABCK,LSPFZN,ICK,OLDTH,WD,W2,W3,
     * W4,RX)
C
C       DIMENSION TH(1),P2(IPDIM,1),P3(IPDIM,1),X(1),NSTART(1),N(1)
C       DIMENSION RL(1),WP3X(1),WP2TH(1),WL(1),PX(IPDIM,1),S(ISDIM,1)
C       DIMENSION OLDTH(1),WD(1),W2(1),W3(1),W4(1),RX(IPDIM,1)
C
C       NP1=NPDIM-1
C-----------------------------------------------------------------------
C
        SUBROUTINE EKF
C
        LOGICAL SA,IV,LFILT,PRERES,STABCK,LSPFZN
C
        DIMENSION WP3X(10),WP2TH(10),WL(10),NSTART(5),N(5),RX(10,10)
        DIMENSION P2(10,10),P3(10,10),X(10),RL(10),PX(10,10),W2(10)
        DIMENSION W3(10),WD(10),W4(10)
C
        COMMON KDUM1(435),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     *  ICK,ISAMP,IFIST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,ISDIM1,
     *  ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     *  NPS1,NPS2,NP1,INVAU,IANT,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     *  DELTA,Y,WT,RES,U,RKJ,SIGMA(6),R1(10),P(10,10),TH(10),S(5,20),
     *  PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
```

Subsystem IDHG

```
     *  P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
        EQUIVALENCE (TH1(1),WP3X(1)),(Z2(1),WP2TH(1)),(Z2(11),WL(1))
        EQUIVALENCE (P2(1,1),TK(1)),(P3(1,1),P1(1,1)),(X(1),PHI(1))
        EQUIVALENCE (RL(1),WPSI(1)),(PX(1,1),FI(1)),(N(1),NF)
        EQUIVALENCE (RX(1,1),TA(1,1)),(W2(1),TB(1,1)),(W3(1),TB(1,2))
        EQUIVALENCE (WD(1),Z(1)),(W4(1),TB(11,1)),(NSTART(1),NST1)
C
C
C****** COMPUTE RESIDUAL ******
        RES=Y-SCAPRO(TH(1),1,X(1),1,NPDIM)
        IF(RLIM.GT.0.AND.ABS(RES).GT.RLIM) RES=SIGN(RLIM,RES)
C
C
C****** P3 SYMMETRIC THEREFORE  X∠P3 = X∠P3 = (P3*X) ******
C***    COMPUTE P3*X     ***
        DO 10 I=1,NPDIM
   10   WP3X(I)=SCAPRO(P3(I,1),IPDIM,X(1),1,NPDIM)
C
C
C****** COMPUTE P2∠TH ******
        DO 20 I=1,NPDIM
   20   WP2TH(I)=SCAPRO(P2(1,I),1,TH(1),1,NPDIM)
C
C
C****** COMPUTE ST AND L(T) ******
        ST=SCAPRO(WP3X(1),1,X(1),1,NPDIM)+2*SCAPRO(X(1),1,WP2TH(1),1,
     *NPDIM)+1
        DO 30 I=1,NPDIM
        WL(I)=WP2TH(I)+WP3X(I)
   30   RL(I)=WL(I)/ST
C
C
C****** COMPUTE TH ******
        IF(STABCK) CALL RMOVE(TH(1),1,OLDTH(1),1,NPDIM)
C
        DO 40 I=1,NPDIM
   40   TH(I)=TH(I)+RL(I)*RES
        IF(STABCK) CALL STABB(TH,OLDTH,N,NSTART,NPDIM,LSPFZN,WD,W4,W2
     *,W3,ICK,IPDIM,PX,RX)
C
C
C****** COMPUTE NEW STATE VECTOR X ******
        CALL RMOVE(X(NP1),-1,X(NPDIM),-1,NP1)
        X(1)=Y
        X(NST3)=U
        IF(KD.GT.0) X(NST3)=S(3,KD)
        X(NST5)=RES
C
C
C****** UPDATE P2 ******
C***    MULTIPLICATION BY F MEANS THAT ALL ROWS ARE MOVED     ***
C***    ONE STEP DOWNWARDS AND ROWS 1,NF+1,NF+NB+1 BECOME ZERO.    ***
```

Subsystem IDHG

```
      DO 45 I=1,NP1
      KDIM=NPDIM-I+1
      DO 45 J=1,NPDIM
   45 P2(KDIM,J)=P2(KDIM-1,J)
      DO 50 I=1,NPDIM
      P2(1,I)=0.0
      P2(NST3,I)=0.0
   50 P2(NST5,I)=0.0
C****** THE MULTIPLICATION  V*[TH∠P2 + X∠P3] = 0  EVERYWHERE ******
C***    EXCEPT ROW NF+NB+1 WHICH EQUALS -WL COMPUTED ABOVE    ***
      DO 60 I=1,NPDIM
   60 P2(NST5,I)=-WL(I)
C
C****** UPDATE P3 ******
C***    COMPUTE DELTA*P3*P3    ***
      DO 70 I=1,NPDIM
      DO 70 J=1,I
      PX(I,J)=DELTA*SCAPRO(P3(I,1),IPDIM,P3(1,J),1,NPDIM)
   70 PX(J,I)=PX(I,J)
C****** L(T) * ST * L(T) = WL * RL COMPUTED ABOVE ******
      DO 80 I=1,NPDIM
      DO 80 J=1,I
      P3(I,J)=P3(I,J)-WL(I)*RL(J)-PX(I,J)
   80 P3(J,I)=P3(I,J)
C
C****** COMPUTE A POSTERIORI RESIDUAL ******
      IF(PRERES) GO TO 90
      RES=Y-SCAPRO(TH(1),1,X(1),1,NPDIM)
      IF(RLIM.GT.0.AND.ABS(RES).GT.RLIM) RES=SIGN(RLIM,RES)
      X(NST5)=RES
C
   90 RETURN
      END
```

Subsystem IDHG

```
C       NAME: KJADMH                                       NUMBER:
C       ----                                               -------
C
C       SUBTITLE:
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       ---------
C
C       AUTHOR:       IVAR GUSTAFSSON
C       REVISED:      HALLGRIMUR GUNNARSSON
C       IMPLEMENTOR:  HALLGRIMUR GUNNARSSON                DATE: 1978-06-22
C       -----------                                        ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                          VERSION:
C       ---------                                          --------
C
C
C
C
C--------------------------------------------------------------------------------
C
C
        SUBROUTINE KJADMH(INKA)
C
C
        DIMENSION W1(10),W2(10),W3(10),N(5)
C
        COMMON/TIME/T
        COMMON KDUM1(455),ISDIM1,ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,
     *  IPDIM,KDUM2(11),NFST,KDUM3(14),
     *  DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *  PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *  P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
C
        EQUIVALENCE (OLDTH(1),W1(1)),(Z(1),W2(1)),(WPSI(1),W3(1))
        EQUIVALENCE (N(1),NF)
C
        GO TO(10,210), INKA
C
   10   TTS=T+1.
        CALL FAST2(TA,TB,FI,TK,E,NFST,U,Y,RES,TTS,WT)
```

Subsystem IDHG

```
C
      J1=0
      DO 30 I=1,3
      K1=0
      IF(I.EQ.2) K1=KD
      N1=N(I)
      IF(N1.LE.0) GO TO 30
      DO 20 J=1,N1
      J1=J1+1
      K2=K1+J
   20 PHI(J1)=S(I,K2)
   30 CONTINUE
C
      WOI(4)=Y-SCAPRO(PHI(1),1,TH(1),1,NPDIM)
      IF(RKJ.LT.0.5) WOI(4)=WOI(4)-RES
C
      CALL RTS(PHI,WOI(4),WT,R1,DELTA,TH,P,NPDIM,IPDIM,W1,W2,W3)
C
      NAB=N(1)+N(2)
      NCK=N(3)
      IF(NCK.LE.0) RETURN
C
      WOI(1)=Y+SCAPRO(TH(NAB+1),1,S(4,1),ISDIM1,NCK)
      WOI(2)=U-SCAPRO(TH(NAB+1),1,S(5,1),ISDIM1,NCK)
      WOI3=SCAPRO(TH(NAB+1),1,Z2(1),1,NCK)
      WOI(3)=RES-WOI3
C
      J1=0
      DO 60 I=4,5
      K1=0
      IF(I.EQ.5) K1=KD
      N1=N(I-3)
      IF(N1.LE.0) GO TO 60
      DO 50 J=1,N1
      J1=J1+1
      K2=K1+J
   50 PHI(J1)=S(I,K2)
   60 CONTINUE
C
      DO 70 I=1,NCK
      J1=J1+1
   70 PHI(J1)=Z2(I)
C
C
      RES3=WOI(1)-SCAPRO(PHI(1),1,TH1(1),1,NPDIM)+WOI3
      IF(RKJ.LT.0.5) RES3=RES3-RES
C
C
      CALL RTS(PHI,RES3,WT,R1,DELTA,TH1,P1,NPDIM,IPDIM,W1,W2,W3)
C
C
C
      RETURN
```

Subsystem IDHG

```
C
C
C
  210   S(2,1)=U
        S(3,1)=RES
C
        IF(N(3).LE.0) RETURN
        CALL RMOVE(Z2(19),-1,Z2(20),-1,19)
C
        S(4,1)=-WOI(1)
        S(5,1)=WOI(2)
        Z2(1)=WOI(3)
C
C
C
        RETURN
C
        END
```

Subsystem IDHG

```
      SUBROUTINE FAST2(TA,TB,FI,TK,E,N,UU,YY,RES1,TTK,WT)
C
C     PERFORMS EXTENDED LEAST SQUARES IDENTIFICATION OF THE
C     PARAMETERS OF THE MODEL
C     Y(T)+A(1)*Y(T-1)+...+A(N)*Y(T-N)=B(1)*U(T-1)+...+
C     +B(N)*U(T-N)+E(T)
C     USING A FAST VERSION.
C     REF.: MORF,LJUNG: FAST ALGORITHMS FOR RECURSIVE IDENTI-
C     FICATION
C
C     N IS THE ORDER OF THE MODEL /MAX=50/
C     UU IS THE LAST INPUT
C     YY IS THE LAST OUTPUT
C     RES1: RESIDUAL COMPUTED
C     TTK IS THE TIME NORMALIZING FACTOR
C
C     SUBROUTINES REQUIRED
C             INV2
C
      DIMENSION TA(100,2),TB(100,2),FI(100),TK(100),E(2,2)
      DIMENSION TC(100,2),TKBAR(102),RES(2),ETA(2)
      DIMENSION EI(2,2),TEMP(2,2),TEMPI(2,2)
C
10    N2=2*N
      N2M=N2-2
      N2P=N2+2
C
50    SL=YY
      DO 110 I=1,N2
110   SL=SL+TA(I,1)*FI(I)
      RES(1)=SL
      RES1=SL
      SL=-UU
      DO 120 I=1,N2
120   SL=SL+TA(I,2)*FI(I)
      RES(2)=SL
C
      DO 130 I=1,N2
      DO 130 J=1,2
130   TA(I,J)=TA(I,J)-TK(I)*RES(J)
C
      G=1.
      DO 135 I=1,N2
  135 G=G-FI(I)*TK(I)
C
      DO 140 I=1,2
      DO 140 J=1,2
  140 E(I,J)=E(I,J)+(G*RES(I)*RES(J)-E(I,J))/(TTK+1.)
C
      CALL INV2(E,EI)
C
      DO 160 I=1,2
      SL=0.
```

Subsystem IDHG

```
        DO 150 J=1,2
  150   SL=SL+G*EI(I,J)*RES(J)
160     TKBAR(I)=SL/(TTK+1.)
        DO 180 I=3,N2P
        SL=TK(I-2)
        DO 170 J=1,2
170     SL=SL+TA(I-2,J)*TKBAR(J)
180     TKBAR(I)=SL
C
        DO 185 I=1,2
185     ETA(I)=FI(N2M+I)
        IF(N2M) 285,285,275
275     DO 280 I=1,N2M
280     FI(N2+1-I)=FI(N2-1-I)
285     FI(1)=YY
        FI(2)=-UU
        DO 190 I=1,2
        DO 190 J=1,N2
190     ETA(I)=ETA(I)+TB(J,I)*FI(J)
C
        DO 200 I=1,2
        DO 200 J=1,2
200     TEMP(I,J)=-TKBAR(N2+I)*ETA(J)
        DO 210 I=1,2
210     TEMP(I,I)=1.+TEMP(I,I)
C
        CALL INV2(TEMP,TEMPI)
C
        DO 220 I=1,N2
        DO 220 J=1,2
220        TB(I,J)=TB(I,J)-TKBAR(I)*ETA(J)
        DO 240 I=1,N2
        DO 240 J=1,2
        SL=0.
        DO 230 K=1,2
230     SL=SL+TB(I,K)*TEMPI(K,J)
240     TC(I,J)=SL
        DO 250 I=1,N2
        DO 250 J=1,2
250     TB(I,J)=TC(I,J)
C
        DO 270 I=1,N2
        SL=TKBAR(I)
        DO 260 J=1,2
260     SL=SL-TB(I,J)*TKBAR(N2+J)
270     TK(I)=SL
C
C
        RETURN
        END
```

Subsystem IDHG

```
      SUBROUTINE INV2(E,EI)
C
C     INVERTS A SECOND ORDER MATRIX
C
      DIMENSION E(2,2),EI(2,2)
C
      DET=E(1,1)*E(2,2)-E(1,2)*E(2,1)
      EI(1,1)=E(2,2)/DET
      EI(2,2)=E(1,1)/DET
      EI(1,2)=-E(1,2)/DET
      EI(2,1)=-E(2,1)/DET
C
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: RTS                                          NUMBER:
C       ----                                               ------
C
C       SUBTITLE:
C       --------
C       UPDATING ONE STEP OF THE PARAMETER VECTOR AND THE COVARIANCE MA-
C       TRIX FOR RECURSIVE IDENTIFICATION ALGORITHMS
C
C       LANGUAGE: FORTRAN IV
C       --------
C
C       KEYWORDS:
C       --------
C       RECURSIVE IDENTIFICATION
C
C       AUTHOR: IVAR GUSTAVSSON
C       REVISED: H. GUNNARSSON
C       IMPLEMENTOR: H. GUNNARSSON                         DATE:1979-12-15
C       -----------                                        ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                          VERSION:
C       --------                                           -------
C
C-----------------------------------------------------------------------
C
C       PURPOSE:
C       =======
C       THE COVARIANCE MATRIX UPDATING FORMULA IS
C                                T             T                 2
C       P(N+1)=[P(N)-K(N+1)*FI  (N+1)*P(N)+R1 *I-DELTA*(P(N)*I)  ]/WT
C
C       WHERE
C                                        T
C       K(N+1)=P(N)*FI(N+1)/[WT+FI(N+1)  *P(N)*FI(N+1)]
C       K(N+1)=P(N)*FI(N+1)/[WT+FI(N+1)  *P(N)*FI(N+1)]
C
C       AND I IS A UNIT DIAGONAL MATRIX.
C
C       THE PARAMETER ESTIMATES ARE UPDATED AS
C
C       T(N+1)=T(N)+K(N+1)*RES
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       ARGUMENTS:
C       ---------
C       RTS(FI,RES,WT,R1,DELTA,T,P,NP,IP,W1,W2,W3)
```

Subsystem IDHG

```
C
C       FI      - VECTOR CONTAINING OLD SIGNAL VALUES CORRESPONDING TO
C                 THE PARAMETERS, SIZE (NP), (I)
C       RES     - RESIDUAL, (I)
C       R1      - VECTOR CONTAINING THE DIAGONAL ELEMENTS OF THE COVARI-
C                 ANCE MATRIX OF THE PARAMETER NOISE, SIZE (NP), (I)
C       DELTA   - ADJUSTMENT TO IMPROVE THE STABILITY OF THE ALGORITHM,
C                 (I)
C       T       - VECTOR OF PARAMETER ESTIMATES, SIZE (NP), (I/O)
C       P       - COVARIANCE MATRIX, SIZE (NP,NP), DIMENSIONED (IP, ),
C                 (I/O)
C       NP      - ACTUAL NUMBER OF PARAMETERS, (I)
C       IP      - DECLARED FIRST DIMENSION OF P, (I)
C       W1      - WORK VECTOR, SIZE (NP)
C       W2      - WORK VECTOR, SIZE (NP)
C       W3      - WORK VECTOR, SIZE (NP)
C
C       CHARACTERISTICS:
C       ===============
C
C       LIBRARY SUBROUTINES REQUIRED:
C       -----------------------------
C       SCAPRO
C
C-----------------------------------------------------------------------
C
        SUBROUTINE RTS(FI,RES,WT,R1,DELTA,T,P,NP,IP,W1,W2,W3)
C
        DIMENSION FI(1),R1(1),T(1),P(IP,1)
        DIMENSION W1(1),W2(1),W3(1)
C
C
C****** THE DELTA COMPUTATION ******
        IF(DELTA.LE.0) GO TO 20
        DO 10 I=1,NP
   10   W3(I)=DELTA*P(I,I)**2/WT
C
C
C****** UPDATING OF THE P-MATRIX ******
   20   DO 30 I=1,NP
        SL=SCAPRO(P(I,1),IP,FI(1),1,NP)
        W1(I)=SL
   30   W2(I)=SL
C
        SL=WT+SCAPRO(FI(1),1,W1(1),1,NP)
C
        DO 50 I=1,NP
        W1(I)=W1(I)/SL
        DO 40 J=1,I
        P(I,J)=(P(I,J)-W1(I)*W2(J))/WT
   40   P(J,I)=P(I,J)
   50   P(I,I)=P(I,I)+R1(I)/WT
C
```

Subsystem IDHG

```
C****** THE DELTA ADJUSTMENT TO P ******
        IF(DELTA.LE.0.) GO TO 70
        DO 60 I=1,NP
   60   P(I,I)=P(I,I)-W3(I)
C
C****** UPDATING OF THE PARAMETER VECTOR ******
   70   DO 80 I=1,NP
   80   T(I)=T(I)+RES*W1(I)
C
        RETURN
        END
```

Subsystem IDHG

```
C       NAME: HATFI1                                NUMBER:
C       ----                                        ------
C
C       SUBTITLE:
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C       -----------                                 ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO GET FILE NAMES
C                TEST THE NAMED FILES FOR PRESENCE ON DISK
C
C-----------------------------------------------------------------------
C
        SUBROUTINE HATFI1
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION FNAM(2)
C
        COMMON/USER/LSTOP
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(8),ISIMN,IDM2(4),FNRS,FNRN,
     *  FNIM,FNST,FNWX,FNWY,IDM3(410),IHAT,IDM4(35),IFIGE
C
        DATA FNAM(2)/' BIN'/
C
        IF(IHAT.LT.0.AND.IFIGE.NE.1) RETURN
C
C
C       FILE NAME REF. SYSTEM
        CALL FIDENT('FNRS',4H    ,FNRS,IND1)
```

Subsystem IDHG

```
      IF(IND1.EQ.0) GO TO 10
      KMESS=40
      GO TO 99
   10 FNAM(1)=FNRS
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.-1) GO TO 20
      KMESS=35
      GO TO 99
C
C
C     FILE NAME REF. NOISE
   20 CALL FIDENT('FNRN',4H    ,FNRN,IND1)
      IF(IND1.EQ.0) GO TO 30
      KMESS=41
      GO TO 99
   30 FNAM(1)=FNRN
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.-1) GO TO 40
      KMESS=36
      GO TO 99
C
C
   40 IF(IFIGE.NE.1.AND.IFIGE.NE.3) GO TO 60
C
C     FILE NAME REF/IDENTIFIED SYSTEM/NOISE
      CALL FIDENT('FNIM',4H    ,FNIM,IND1)
      IF(IND1.EQ.0) GO TO 50
      KMESS=42
      GO TO 99
   50 FNAM(1)=FNIM
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.0) GO TO 60
      KMESS=37
      GO TO 99
C
C
C
   60 IPO=3
      IF(IFIGE.LT.2.OR.IFIGE.GT.3) GO TO 80
C
C     FILE NAME STATISTCS FILE
      IPO=1
      CALL FIDENT('FNST',4H    ,FNST,IND1)
      IF(IND1.EQ.0) GO TO 70
      KMESS=43
      GO TO 99
   70 FNAM(1)=FNST
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.-1) IPO=2
C
   80 IF(IHAT.GE.0) CALL FISTAR(IPO)
      IF(KMESS.GT.1) GO TO 99
      RETURN
```

Subsystem IDHG

```
C
C
C
   99   MSTOP=.TRUE.
  100   LSTOP=.TRUE.
        ISIMN=ISIMN-1
        CALL DINT('ISIMN',4H     ,ISIMN,IND1)
        RETURN
        END
```

Subsystem IDHG

```
C        NAME: FISTAR                                        NUMBER:
C        ----                                                -------
C
C        LANGUAGE: STANDARD FORTRAN
C        ---------
C
C        PROGRAM TYPE: SUBROUTINE
C        -------------
C
C        AUTHOR AND
C        IMPLEMENTOR: HALLGRIMUR GUNNARSSON           DATE: 1978-06-22
C        -----------                                  ----
C
C        INSTITUTE:
C        ----------
C        DEPARTMENT OF AUTOMATIC CONTROL
C        LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C        ACCEPTED:                                    VERSION:
C        ---------                                    --------
C
C
C
C-----------------------------------------------------------------------
C
C        PURPOSE: TO INITIATE WORK FILE
C                 TO INITIATE STATISTICS FILE, IF NOT PRESENT, OR TO
C                 TEST ACTUAL PARAMETERS AGAINST PARAMETER LIST IN FILE
C                 HEAD.
C
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FISTAR(IPO)
C
      LOGICAL MSTOP
C
      DIMENSION FNAM(2),KOD(10),RPAR(59),SPAR(59),IPAR(17),KPAR(17)
C
      COMMON/SIMARG/T1,T2
C
C
      COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(8),ISIMN,IDM2(4),FNRS,FNRN,
     * FNIM,FNST,FNWX,FNWY,IDM3(306),WTI,IDM4(10),PFST,THO(10),PO(10),
     * TH10(10),P10(10),ISTAB,IPRER,IDM5(4),METOD,IDM6(3),IHAT,IHATN,
     * IDM7(7),IFIST,IDM8(8),NF,NA,NB,ND,NC,KD,NPDIM,IDM9(9),INVAU,
     * IDM10(2),NFST,IDM11(4),WTO,TS,DT,WRES,RLIM,DELTA,Y,WT,RES,U,
     * RKJ,WOI(6),R1(10)
C
      DATA FNAM(2)/' BIN'/,FNU1X/'XHGX'/,FNU2X/'YHGY'/
C
      FNWX=FNU1X
      FNWY=FNU2X
C
```

Subsystem IDHG

```
      IF(IPO.EQ.3) GO TO 100
C
      IPAR(1)=(T2-T1)+0.1
      IPAR(2)=NPDIM
      IPAR(3)=NPDIM+11
      IPAR(4)=METOD
      IPAR(5)=NF
      IPAR(6)=NA
      IPAR(7)=NB
      IPAR(8)=ND
      IPAR(9)=NC
      IPAR(10)=NFST
      IPAR(11)=KD
      IPAR(12)=IPRER
      IPAR(13)=ISTAB
      IPAR(14)=IFIST
      IPAR(15)=INVAU
      IPAR(16)=IHAT
      IPAR(17)=IHATN
C
      RPAR(1)=WTI
      RPAR(2)=WTO
      RPAR(3)=DELTA
      RPAR(4)=RLIM
      RPAR(5)=RKJ
      RPAR(6)=PFST
      RPAR(7)=DT
      RPAR(8)=FNRS
      RPAR(9)=FNRN
C
      DO 10 I=10,19
      RPAR(I)=R1(I-9)
      RPAR(I+10)=THO(I-9)
      RPAR(I+20)=TH10(I-9)
      RPAR(I+30)=PO(I-9)
   10 RPAR(I+40)=P10(I-9)
C
C
      FNAM(1)=FNST
      IF(IPO.EQ.2) GO TO 80
C
      CALL ENTER(LDK1,FNAM)
      WRITE(LDK1) (IPAR(I),I=1,17),(RPAR(J),J=1,59)
      CALL CLOSE(LDK1)
      GO TO 100
C
   80 CALL SEEK(LDK1,FNAM)
      READ(LDK1) (KPAR(I),I=1,17),(SPAR(J),J=1,59)
      CALL CLOSE(LDK1)
C
      DO 90 I=1,17
      IF(IPAR(I).NE.KPAR(I)) GO TO 998
   90 CONTINUE
```

Subsystem IDHG

```
C
C
      EPS=1.E-6
      DO 95 I=1,59
      IF(RPAR(I)-SPAR(I).GT.EPS) GO TO 998
   95 CONTINUE
C
C     INITIATE WORK FILE
  100 FNAM(1)=FNWX
      KOD(1)=-99999
      KOD(2)=3+2*NPDIM
      KOD(3)=NPDIM
      CALL DLETE(LDK1,FNAM,J)
      CALL ENTER(LDK1,FNAM)
      WRITE(LDK1) (KOD(I),I=1,10)
      RETURN
C
  998 KMESS=38
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: HATFI2                                     NUMBER:
C       ----                                             -------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON           DATE: 1978-06-22
C       -----------                                  ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                        VERSION:
C       --------                                         -------
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO PERFORM ACCURACY COMPUTATIONS FOR IDHG AND PLACE
C                THE RESULTS ON FILE  (YHGY)
C                GENERATE A FILE WITH THE IDENTIFIED SYSTEM'S IMPULSE
C                RESPONSE
C
C-----------------------------------------------------------------------
C
        SUBROUTINE HATFI2
C
        LOGICAL LSTOP,MSTOP
C
        DIMENSION F1(2),F2(2),KOD(10),THS(10),TH1S(10),RNA(4)
C
        COMMON/USER/LSTOP
        COMMON/IMPRES/HSR(100),HNR(100),HSI(100),HNI(100)
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(13),FNRS,FNRN,FNIM,FNST,FNWX,
     *  FNWY,IDM2(410),IHAT,IDM3(7),ISAMP,IDM4(9),N(5),KD,NPDIM,IPDIM,
     *  NSTART(5),IDM5(5),IFIGE,NFST,WV,V,IDM6(254),TH(10),IDM7(280),
     *  TH1(10)
C
        DATA F1(2)/' BIN'/,F2(2)/' BIN'/,BLANK/'    '/
        DATA RNA(1),RNA(2),RNA(3),RNA(4)/'RS','IS','RN','IN'/
C
        IF(IHAT.LT.0.AND.IFIGE.NE.1) RETURN
        CALL CLOSE(LDK1)
C
C
C****** READ REF. IMPULSE RESPONSES ******
        F1(1)=FNRS
        F2(1)=FNRN
```

Subsystem IDHG

```
      CALL SEEK(LDK1,F1)
      READ(LDK1) (KOD(I),I=1,10)
      CALL SEEK(LDK2,F2)
      READ(LDK2) (KOD(I),I=1,10)
      DO 10 I=1,100
      READ(LDK1,END=90) HSR(I)
      READ(LDK2,END=90) HNR(I)
   10 CONTINUE
C
      CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
C
      IF(IHAT.LT.0) GO TO 45
C
C****** COMPUTE LOSS FUNCTIONS AND PLACE ON FILE FNWY=YHGY ******
      F1(1)=FNWX
      F2(1)=FNWY
      CALL SEEK(LDK1,F1)
      READ(LDK1) (KOD(I),I=1,10)
      CALL DLETE(LDK2,F2,J)
      CALL ENTER(LDK2,F2)
      KOD(2)=NPDIM+7
      KOD(3)=NPDIM
      WRITE(LDK2) (KOD(I),I=1,10)
C
C
   20 READ(LDK1,END=30) NSAMP,VS,WVS,(THS(I),I=1,NPDIM),(TH1S(J),J=1,
     * NPDIM)
      CALL HATCOM(N,NSTART,THS,HSR,HNR,HSI,HNI,V1,V2,V3,V4)
      VS=VS/FLOAT(NSAMP)
      WVS=WVS/FLOAT(NSAMP)
      WRITE(LDK2) NSAMP,VS,WVS,(THS(I),I=1,NPDIM),V1,V2,V3,V4
      GO TO 20
C
   30 IF(NSAMP.GE.ISAMP) GO TO 40
      CALL HATCOM(N,NSTART,TH,HSR,HNR,HSI,HNI,V1,V2,V3,V4)
      VS=V/FLOAT(ISAMP)
      WVS=WV/FLOAT(ISAMP)
      WRITE(LDK2) ISAMP,VS,WVS,(TH(I),I=1,NPDIM),V1,V2,V3,V4
   40 CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
      CALL DLETE(LDK1,F1,J)
C
C
C
   45 IF(.NOT.(IFIGE.EQ.1.OR.IFIGE.EQ.3)) GO TO 60
C
C****** GENERATE A FILE CONTAINING REF. AND IDENTIFIED PULSE RESPONSES *
      F1(1)=FNIM
      CALL ENTER(LDK1,F1)
      DO 46 I=1,10
   46 KOD(I)=0
      KOD(1)=100000
```

Subsystem IDHG

```
      KOD(2)=5
      KOD(3)=1
      KOD(10)=4
      WRITE(LDK1) (KOD(I),I=1,10)
      DO 47 I=1,4
   47 WRITE(LDK1) RNA(I),BLANK
      CALL HATCOM(N,NSTART,TH,HSR,HNR,HSI,HNI,V1,V2,V3,V4)
      RT=0.0
C
      DO 50 I=1,100
      RT=RT+1.0
   50 WRITE(LDK1) RT,HSR(I),HSI(I),HNR(I),HNI(I)
      CALL CLOSE(LDK1)
C
   60 RETURN
C
C
C-----------------------------------------------------------------------
C
   90 KMESS=40
   99 MSTOP=.TRUE.
      LSTOP=.TRUE.
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: HATCOM                                      NUMBER:
C       ----                                              ------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON             DATE: 1978-06-22
C       -----------                                    ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                       VERSION:
C       --------                                        -------
C
C
C-----------------------------------------------------------------------
C
C
        SUBROUTINE HATCOM(N,NSTART,TH,HU,HE,HS,HN,V1,V2,V3,V4)
C
C       N,NSTART,TH      SEE COMMENTS TO IMRES
C       HU,HE            REFERENCE SYSTEM,NOISE IMPULSE RESPONSES
C       HS,HN            ACTUAL(TH) SYSTEM,NOISE IMPULSE RESPONSES
C       V1,V2,V3,V4      LOSS FUNCTIONS  (SEE USER'S GUIDE HGPAC)
C
C
C
        DIMENSION HS(1),HN(1),HU(1),HE(1),N(1),NSTART(1),TH(1)
C
C
C
   10   ITYP=0
        CALL IMRES(ITYP,N,NSTART,TH,HS)
        ITYP=1
        CALL IMRES(ITYP,N,NSTART,TH,HN)
        SIGMU=0.
        EHU2=0.
        SIGME=0.
        EHE2=0.
C
        DO 15 I=1,100
        SIGU=HS(I)-HU(I)
        SIGMU=SIGMU+SIGU**2
        EHU2=EHU2+HU(I)**2
        SIGE=HN(I)-HE(I)
        SIGME=SIGME+SIGE**2
```

Subsystem IDHG

```
   15   EHE2=EHE2+HE(I)**2
C
        V1=SIGMU
        V2=SIGMU/EHU2
        V3=SIGME
        V4=SIGME/EHE2
C
        RETURN
        END
```

Subsystem IDHG

```
C       NAME: HATWRT                                     NUMBER:
C       ----                                             ------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE: 1978-06-22
C       -----------                                      ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                        VERSION:
C       --------                                         -------
C
C
C
C-----------------------------------------------------------------------
C
C       PURPOSE: TO PRINT THE RESULTS OF THE ACCURACY COMPUTATIONS
C                AND IF IFIGE EQ.2 OR 3 PLACE THE RESULTS ON
C                FILE FNST
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE HATWRT
C
        LOGICAL MSTOP
C
        DIMENSION F1(2),F2(2),KOD(10),RDUM(16),THS(10),IPAR(17),RPAR(59)
        DIMENSION IX(5)
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(4),IDOC,IYEAR,MONTH,IDAY,
     *  ISIMN,IDM2(4),FNRS,FNRN,FNIM,FNST,FNWX,FNWY,IDM3(410),IHAT,
     *  IDM4(23),NPDIM,IDM5(11),IFIGE
C
        DATA F1(2)/' BIN'/,F2(2)/' BIN'/
C
C
        K99=-99999
        NPD6=NPDIM+6
        IF(IHAT.LT.0) RETURN
        IF(IHAT.EQ.0) GO TO 125
        IF(IDOC.EQ.2) GO TO 125
C
        WRITE(6,10) IYEAR,MONTH,IDAY,ISIMN
```

Subsystem IDHG

```
   10   FORMAT('1ACCURACY',34X,I4,'-',I2,'-',I2,34X,'SIMULATION NO:',
     *  I5,5X,'I')
        WRITE(6,20)
   20   FORMAT(1X,'********')
        WRITE(6,30)
   30   FORMAT(//21X,'2',21X,'2')
        WRITE(6,40)
   40   FORMAT(1X,'V = SUM[HR(I)-HI(I)]',5X,'VN = V/SUM[HR(I)]',6X,
     *  'I=1,100')
        WRITE(6,50)
   50   FORMAT(//17X,'2',15X,'2')
        WRITE(6,60)
   60   FORMAT(1X,'ISAMP',6X,'E[RES ]',8X,'E[WRES ]',9X,'V[SYS]',10X,'VN
     *[SYS]',8X,'V[NOISE]',8X,'VN[NOISE]')
        WRITE(6,70)
   70   FORMAT(1X,'-------+----------------+----------------+---------------
     *-+----------------+----------------+----------------+')
C
C
        F1(1)=FNWY
        CALL SEEK(LDK1,F1)
        READ(LDK1) (KOD(I),I=1,10)
C
  100   READ(LDK1,END=120) NSAMP,VS,WVS,(THS(I),I=1,NPDIM),V1,V2,V3,V4
        WRITE(6,110) NSAMP,VS,WVS,V1,V2,V3,V4
  110   FORMAT(1X,I5,6(' I ',G13.6),' I')
        GO TO 100
  120   CALL CLOSE(LDK1)
C
  125   IF(IFIGE.LT.2.OR.IFIGE.GT.3) GO TO 200
C
C
        F2(1)=FNWX
        F1(1)=FNST
        CALL SEEK(LDK2,F1)
        READ(LDK2) (IPAR(I),I=1,17),(RPAR(J),J=1,59)
        CALL ENTER(LDK1,F2)
        WRITE(LDK1) (IPAR(I),I=1,17),(RPAR(J),J=1,59)
C
  130   READ(LDK2,END=150) (IX(I),I=1,5),(RDUM(J),J=1,NPD6)
        WRITE(LDK1) (IX(I),I=1,5),(RDUM(J),J=1,NPD6)
        GO TO 130
C
  150   CALL CLOSE(LDK2)
        CALL DLETE(LDK2,F1,J)
        F1(1)=FNWY
        CALL SEEK(LDK2,F1)
        READ(LDK2) (KOD(I),I=1,10)
        WRITE(LDK1) K99,ISIMN,IYEAR,MONTH,IDAY,(RDUM(I),I=1,NPD6)
C
  160   READ(LDK2,END=180) NSAMP,V,WV,(THS(I),I=1,NPDIM),V1,V2,V3,V4
        WRITE(LDK1) NSAMP,(IX(J),J=1,4),(THS(I),I=1,NPDIM),V,WV,V1,V2,
     *V3,V4
```

Subsystem IDHG

```
        GO TO 160
  180   CALL CLOSE(LDK1)
        CALL CLOSE(LDK2)
C
C
        F1(1)=FNST
        CALL RENAM(LDK1,F2,F1,J)
        IF(J.EQ.-1) GO TO 200
  190   KMESS=40
        MSTOP=.TRUE.
        RETURN
C
  200   F2(1)=FNWY
        CALL DLETE(LDK1,F2,J)
        IF(J.EQ.0) GO TO 190
        RETURN
C
        END
```

Subsystem IDHG

```
C       NAME: STAWRT                                  NUMBER:
C       ----                                          -------
C
C       SUBTITLE:
C       ---------
C
C       LANGUAGE: STANDARD FORTRAN
C       ---------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON            DATE:
C       -----------                                   ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                     VERSION:
C       ---------                                     -------
C
C
C
C
C
C
C-----------------------------------------------------------------------
C
      SUBROUTINE STAWRT(ISAMP,TH,OLDTH,NPDIM)
      DIMENSION TH(1),OLDTH(1)
      WRITE(6,10) ISAMP
   10 FORMAT(' STAWRT ISAMP=',I6,' F OR C POL NOT STABLE,OLDTH USED')
      RETURN
      END
```

Subsystem IDHG

```
C        NAME: WRTHG                                  NUMBER:
C        ----                                         -------
C
C        SUBTITLE: PRINT-OUT DURING SIMULATION
C        ---------
C
C        LANGUAGE: STANDARD FORTRAN
C        ---------
C
C        PROGRAM TYPE: SUBROUTINE
C        -------------
C
C        IMPLEMENTOR: HALLGRIMUR GUNNARSSON         DATE: 1978-06-22
C        ------------                               ----
C
C        INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C        ----------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C        ACCEPTED:                                    VERSION:
C        ---------                                    --------
C
C
C------------------------------------------------------------------------------
C
      SUBROUTINE WRTHG
C
      LOGICAL SA,IV,LFILT,PRERES,STABCK,LSPFZN
      DIMENSION TEE(5,10),N(5)
C
      COMMON KDUM1(435),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     * ICK,ISAMP,IFIST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,ISDIM1,
     * ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     * NPS1,NPS2,NP1,INVAU,IDM35,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     * DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     * PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     * P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
      EQUIVALENCE (FI(1),TEE(1,1)),(NF,N(1))
C
C
C****** WRITE ISAMP - TH - (TH1) ******
      WRITE(6,10) ISAMP
   10 FORMAT(//'0ISAMP:',I5)
      WRITE(6,20) (TH(I),I=1,NPDIM)
   20 FORMAT(' TH:'/(1X,8G14.6))
      IF(METOD.EQ.6) WRITE(6,25) (TH1(I),I=1,NPDIM)
   25 FORMAT(' TF:'/(1X,8G14.6))
      IF(IWRT.EQ.1) RETURN
C
C
C****** WRITE Y - U - RES - WRES - WT  ******
      WRITE(6,30) Y,U,RES,WRES,WT
   30 FORMAT(3H Y:,G14.6,5X,2HU:,G14.6,5X,4HRES:,G14.6,5X,5HWRES:,G14.6
```

Subsystem IDHG

```
     *5X,3HWT:,G14.6)
        IF(IWRT.EQ.2) RETURN
C
C
C****** WRITE P - ( P1 - TA - E ) ******
        WRITE(6,40)
   40   FORMAT(10H P-MATRIX:)
        DO 70 I=1,NPDIM
        WRITE(6,50) (P(I,K),K=1,NPDIM)
   50   FORMAT(1X,8G14.6)
   70   CONTINUE
        IF(METOD.NE.6) GO TO 75
        WRITE(6,71)
   71   FORMAT(11H P1-MATRIX:)
C
        DO 72 I=1,NPDIM
        WRITE(6,50) (P1(I,K),K=1,NPDIM)
   72   CONTINUE
C
        WRITE(6,73)
   73   FORMAT(' TA:')
        N2=2*NFST
        WRITE(6,50) (TA(2*I-1,1),I=1,NFST)
        WRITE(6,50) (TA(2*I,1),I=1,NFST)
C
        WRITE(6,74) E(1,1),E(1,2),E(2,1),E(2,2)
   74   FORMAT(' E:'/1X,2G14.6/1X,2G14.6)
   75   IF(IWRT.EQ.3) RETURN
C
C
C****** WRITE PHI - WPSI - Z - (FI) ******
        WRITE(6,80) (PHI(I),I=1,NPDIM)
   80   FORMAT(' PHI:'/(1X,8G14.6))
C
        WRITE(6,90) (WPSI(I),I=1,NPDIM)
   90   FORMAT(' PSI:'/(1X,8G14.6))
C
        WRITE(6,100) (Z(I),I=1,NPDIM)
  100   FORMAT(' Z:'/(1X,8G14.6))
C
        IF(METOD.NE.6) GO TO 105
        WRITE(6,102) (FI(I),I=1,N2)
  102   FORMAT(' FI:'/(1X,8G14.6))
  105   IF(IWRT.EQ.4) RETURN
C
C
C****** WRITE S - (Z2) ******
        WRITE(6,110)
  110   FORMAT(' S-MATRIX:')
        DO 130 I=1,ISDIM1
        WRITE(6,50) (S(I,J),J=1,20)
  130   CONTINUE
C
```

Subsystem IDHG

```
      IF(METOD.NE.6) GO TO 135
      WRITE(6,132) (Z2(I),I=1,20)
  132 FORMAT(' Z2:'/(1X,8G14.6))
  135 IF(IWRT.EQ.5) RETURN
C
C
C****** WRITE TEE - (TK) ******
      WRITE(6,140)
  140 FORMAT(' TEE-MATRIX:')
      DO 160 I=1,ISDIM1
      WRITE(6,50) (TEE(I,K),K=1,10)
  160 CONTINUE
C
      IF(METOD.NE.6) RETURN
      WRITE(6,170) (TK(I),I=1,N2)
  170 FORMAT(' TK:'/(1X,8G14.6))
      RETURN
      END
```

Subsystem IDHG

```
C       NAME: WREKF                                     NUMBER:
C       ----                                            ------
C
C       SUBTITLE: PRINT-OUT DURING SIMULATION
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON               DATE: 1978-06-22
C       -----------                                     ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                       VERSION:
C       --------                                        -------
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE WREKF
C
        LOGICAL SA,IV,LFILT,PRERES,STABCK,LSPFZN
        DIMENSION N(5),NSTART(5),WP3X(10),WP2TH(10),WL(10)
        DIMENSION P2(10,10),P3(10,10),X(10),RL(10),PX(10,10)
C
        COMMON KDUM1(435),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     *  ICK,ISAMP,IFIST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,ISDIM1,
     *  ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     *  NPS1,NPS2,NP1,INVAU,IDM35,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     *  DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *  PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *  P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
        EQUIVALENCE (TH1(1),WP3X(1)),(Z2(1),WP2TH(1)),(Z2(11),WL(1))
        EQUIVALENCE (P2(1,1),TK(1)),(P3(1,1),P1(1,1)),(X(1),PHI(1))
        EQUIVALENCE (RL(1),WPSI(1)),(PX(1,1),FI(1))
        EQUIVALENCE (N(1),NF),(NSTART(1),NST1)
C
C
C
        DATA A1,A2,A3,A4/'ISAMP','TH:','P3-MA','P2-MA'/
        DATA A5,A6,A7,A8/'PX-MA','S -MA','TRIX:','X:'/
        DATA A9,A10,A11,A12/'WP3X:','WP2TH','WL:','RL:'/
        DATA A13,A14,A15/'Y:','U:','RES:'/
C
C
   10   FORMAT(//1X,A5,I5)
   20   FORMAT(1X,A5/(1X,8G14.6))
```

Subsystem IDHG

```
   30    FORMAT(1X,2A5)
   40    FORMAT(1X,8G14.6)
   50    FORMAT(1X,3(A5,4X,G14.6))
C
C
C
C****** WRITE ISAMP - TH ******
         WRITE(6,10) A1,ISAMP
         WRITE(6,20) A2,(TH(I),I=1,NPDIM)
         IF(IWRT.EQ.1) RETURN
C
C
C****** WRITE Y - U - RES ******
         WRITE(6,50) A13,Y,A14,U,A15,RES
         IF(IWRT.EQ.2) RETURN
C
C
C****** WRITE P3 - P2 ******
         WRITE(6,30) A3,A7
         DO 55 I=1,NPDIM
   55    WRITE(6,40) (P3(I,K),K=1,NPDIM)
         WRITE(6,30) A4,A7
         DO 60 I=1,NPDIM
   60    WRITE(6,40) (P2(I,K),K=1,NPDIM)
         IF(IWRT.EQ.3) RETURN
C
C
C****** WRITE X - RL ******
         WRITE(6,20) A8,(X(I),I=1,NPDIM)
         WRITE(6,20) A12,(RL(I),I=1,NPDIM)
         IF(IWRT.EQ.4) RETURN
C
C
C****** WRITE S ******
         WRITE(6,30) A6,A7
         DO 70 I=1,ISDIM1
   70    WRITE(6,40) (S(I,J),J=1,20)
         IF(IWRT.EQ.5) RETURN
C
C
C****** WRITE WP3X - WP2TH - WL - PX ******
         WRITE(6,20) A9,(WP3X(I),I=1,NPDIM)
         WRITE(6,20) A10,(WP2TH(I),I=1,NPDIM)
         WRITE(6,20) A11,(WL(I),I=1,NPDIM)
         WRITE(6,30) A5,A7
         DO 80 I=1,NPDIM
   80    WRITE(6,40) (PX(I,K),K=1,NPDIM)
         RETURN
         END
```

Subsystem IDHG

```
C       NAME: DOCID                                     NUMBER:
C       ----                                            ------
C
C       SUBTITLE: DOCUMENTATION OF PARAMETERS
C       --------  FOR THE SUBSYSTEM IDHG
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE: 1978-06-22
C       -----------                                     ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                       VERSION:
C       --------                                        -------
C
C-----------------------------------------------------------------------
C
        SUBROUTINE DOCID
C
        LOGICAL MSTOP,SA,IV,LFILT,PRERES,STABCK,LSPFZN
C
        DIMENSION A(7)
C
        COMMON IDM1(2),KMESS,MSTOP,INHG,ISYSH,IDH,LINES,IDOC,IYEAR,
     *  MONTH,IDAY,ISIMN,IDM2(4),FNRS,FNRN,FNIM,FNST,IDM3(310),WTI,
     *  IDM4(10),PFST,THO(10),PO(10),TH10(10),P10(10),ISTAB,IPRER,
     *  IDM5(4),METOD,IWRT,IWRT1,IWRTN,IHAT,IHATN,NSH(5),
     *  ICK,ISAMP,IFIST,SA,IV,LFILT,PRERES,STABCK,LSPFZN,ISDIM1,
     *  ISDIM2,NF,NA,NB,ND,NC,KD,NPDIM,IPDIM,NST1,NST2,NST3,NST4,NST5,
     *  NPS1,NPS2,NP1,INVAU,IDM35,IFIGE,NFST,WV,V,WTO,TS,DT,WRES,RLIM,
     *  DELTA,Y,WT,RES,U,RKJ,WOI(6),R1(10),P(10,10),TH(10),S(5,20),
     *  PHI(10),OLDTH(10),Z(10),WPSI(10),TH1(10),Z2(20),FI(100),
     *  P1(10,10),TK(100),TA(100,2),TB(100,2),E(2,2)
C
        DATA A(1)/'LIP'/,A(2)/'CMA'/,A(3)/'LIP'/,A(4)/'CMA'/,A(5)/'IV'/
        DATA A(6)/'KJADM'/,A(7)/'EKF'/,B1/'SA'/,B2/'  '/
C
        B=B2
        IF(METOD.LE.2) B=B1
        IF(INHG.EQ.1.OR.ISYSH.EQ.1) GO TO 115
        WRITE(6,10) ISIMN
   10   FORMAT('1DOCUMENTATION',35X,'SIMU. NO:',I11)
        WRITE(6,20) IYEAR,MONTH,IDAY
   20   FORMAT(' *************',35X,'DATE    :',I5,'=',I2,'=',I2)
```

Subsystem IDHG

```
        LINES=2
C
  115   WRITE(6,180)
  180   FORMAT(//' SYSTEM IDHG:')
        WRITE(6,40)
   40   FORMAT(' ------------')
        WRITE(6,190) METOD,A(METOD),B
  190   FORMAT(' IDENTIFICATION METHOD:',I2,2X,A5,2X,A5)
        WRITE(6,200) NF,IPRER,IWRT
  200   FORMAT(/' NA   =',I9,9X,'IPRER=',I9,9X,'IWRT =',I9)
        WRITE(6,210) NA,IFIST,IWRT1
  210   FORMAT(' NF   =',I9,9X,'IFIST=',I9,9X,'IWRT1=',I9)
        WRITE(6,220) NB,ISTAB,IWRTN
  220   FORMAT(' NB   =',I9,9X,'ISTAB=',I9,9X,'IWRTN=',I9)
        WRITE(6,230) ND,INVAU,IHAT
  230   FORMAT(' ND   =',I9,9X,'INVAU=',I9,9X,'IHAT =',I9)
        WRITE(6,240) NC,IHATN
  240   FORMAT(' NC   =',I9,33X,'IHATN=',I9)
        WRITE(6,250) KD,NFST,IFIGE
  250   FORMAT(' KD   =',I9,9X,'NFST =',I9,9X,'IFIGE=',I9)
C
        WRITE(6,260) WTI,PFST,FNRS
  260   FORMAT(/' WTI  =',G13.6,5X,'PFST =',G13.6,5X,'FNRS =',4X,A5)
        WRITE(6,270) WTO,RKJ,FNRN
  270   FORMAT(' WTO  =',G13.6,5X,'RKJ  =',G13.6,5X,'FNRN =',4X,A5)
        WRITE(6,280) RLIM,FNIM
  280   FORMAT(' RLIM =',G13.6,29X,'FNIM =',4X,A5)
        WRITE(6,290) DELTA,DT,FNST
  290   FORMAT(' DELTA=',G13.6,5X,'DT   =',G13.6,5X,'FNST =',4X,A5/)
C
        DO 320 I=1,NPDIM
        IF(I.LT.10) WRITE(6,300) I,TH(I),I,THO(I),I,PO(I)
  300   FORMAT(' TH(',I1,')  =',G13.6,3X,'THO(',I1,') =',G13.6,3X,'PO(',
     *  I1,')  =',G13.6)
        IF(I.GE.10) WRITE(6,310) I,TH(I),I,THO(I),I,PO(I)
  310   FORMAT(' TH(',I2,') =',G13.6,3X,'THO(',I2,')=',G13.6,3X,'PO(',
     *  I2,') =',G13.6)
  320   CONTINUE
        WRITE(6,325)
  325   FORMAT(1X)
        LINES=LINES+18+NPDIM+1
C
        IF(METOD.NE.6) GO TO 360
        IF(64-LINES.GT.NPDIM) GO TO 326
        WRITE(6,10) ISIMN
        WRITE(6,370)
        LINES=1
  326   DO 350 I=1,NPDIM
        IF(I.LT.10) WRITE(6,330) I,TH1(I),I,TH10(I),I,P10(I)
  330   FORMAT(' TH1(',I1,') =',G13.6,3X,'TFO(',I1,') =',G13.6,3X,'P10('
     *  ,I1,') =',G13.6)
        IF(I.GE.10) WRITE(6,340) I,TH1(I),I,TH10(I),I,P10(I)
  340   FORMAT(' TH1(',I2,')=',G13.6,3X,'TFO(',I2,')=',G13.6,3X,'P10(',
```

Subsystem IDHG

```
     *   I2,')=',G13.6)
  350    CONTINUE
         WRITE(6,325)
         LINES=LINES+NPDIM+1
C
  360    IF(64-LINES.GT.NPDIM) GO TO 380
         WRITE(6,10) ISIMN
         WRITE(6,370)
  370    FORMAT('+',15X,'(CONT)'/)
C
  380    DO 410 I=1,NPDIM
         IF(I.LT.10) WRITE(6,390) I,R1(I)
  390    FORMAT(' R1(',I1,')  =',G13.6)
         IF(I.GE.10) WRITE(6,400) I,R1(I)
  400    FORMAT(' R1(',I2,') =',G13.6)
  410    CONTINUE
C
         RETURN
         END
```

Subsystem IDHG

```
C        NAME: IDMESS                                  NUMBER:
C        ----                                          ------
C
C        SUBTITLE: ERROR MESSAGES WHEN SIMULATIONS HAVE TO BE STOPPED
C        --------
C
C        LANGUAGE: STANDARD FORTRAN
C        --------
C
C        PROGRAM TYPE: SUBROUTINE
C        ------------
C
C        AUTHOR AND
C        IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE: 1978-06-22
C        -----------                                 ----
C
C        INSTITUTE:
C        ---------
C        DEPARTMENT OF AUTOMATIC CONTROL
C        LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C        ACCEPTED:                                     VERSION:
C        --------                                      -------
C
C
C
C
C-----------------------------------------------------------------------
C
C
         SUBROUTINE IDMESS
C
         LOGICAL MSTOP
C
         COMMON IDM1(2),KMESS,MSTOP,IDM2(13),FNRS,FNRN,FNIM,FNST
C
         DATA FN1,FN2,FN3,FN4/'FNRS','FNRN','FNIM','FNST'/
C
C
         IF(KMESS.LE.23.OR.KMESS.GT.43) RETURN
C
C
         LU=9
         MSTOP=.FALSE.
C
         WRITE(LU,3000)
 3000    FORMAT(' SYSTEM IDHG')
C
         IDMES=KMESS-22
C
C
         GO TO (1,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,
     *   42,43),IDMES
C
```

Subsystem IDHG

```
C
    1    WRITE(LU,101)
         GO TO 100
   24    WRITE(LU,124)
         GO TO 100
   25    WRITE(LU,125)
         GO TO 100
   26    WRITE(LU,126)
         GO TO 100
   27    WRITE(LU,127)
         GO TO 100
   28    WRITE(LU,128)
         GO TO 100
   29    WRITE(LU,129)
         GO TO 100
   30    WRITE(LU,130)
         GO TO 100
   31    WRITE(LU,131)
         GO TO 100
   32    WRITE(LU,132)
         GO TO 100
   33    WRITE(LU,133)
         GO TO 100
   34    WRITE(LU,134)
         GO TO 100
   35    WRITE(LU,135) FNRS
         GO TO 100
   36    WRITE(LU,135) FNRN
         GO TO 100
   37    WRITE(LU,137) FNIM
         GO TO 100
   38    WRITE(LU,138) FNST
         GO TO 100
   39    WRITE(LU,139)
         GO TO 100
   40    WRITE(LU,140) FN1
         GO TO 100
   41    WRITE(LU,140) FN2
         GO TO 100
   42    WRITE(LU,140) FN3
         GO TO 100
   43    WRITE(LU,140) FN4
         GO TO 100
  100    RETURN
C
C
  101    FORMAT(' HVERNIG TOKST DETTA*-*  FEL I PROGRAM*-*')
C
C
  124    FORMAT(' PAR METOD MUST BE IN THE RANGE (1,7)')
  125    FORMAT(' NA,NF,NB,ND,NC MUST ALL BE GREATER OR EQUAL TO ZERO')
  126    FORMAT(' PARAMETERS NB+KD MUST BE LESS THAN 20')
  127    FORMAT(' NPDIM=NA+NF+NB+ND+NC TOO LARGE,MUST BE LESS THAN 10')
```

```
  128   FORMAT(' IV IDENTIFIES ONLY THE A AND B POLYNOMIALS, THEREFORE
     *  MUST NF=ND=NC=0')
  129   FORMAT(' THE DEPOSIT OF THE GLOBAL VARIABLE ISIMN FAILED')
  130   FORMAT(' EKF IDENTIFIES ONLY THE A,B AND C POLYNOMIALS'/
     *  ' NF AND ND MUST BE ZERO')
  131   FORMAT(' IPRER MUST BE EQUAL TO ZERO OR 1')
  132   FORMAT(' PAR ISTAB MUST BE EQUAL TO ZERO ,1 OR 2')
  133   FORMAT(' IF PAR IFIGE .GT. 1 THEN PAR IHAT MUST BE .GE.0')
  134   FORMAT(' PAR KD NEG,MUST BE GREATER OR EQUAL TO ZERO')
  135   FORMAT(' FILE ',A5,' NOT FOUND')
  137   FORMAT(' THE FILE ',A5,' IS PRESENT ON DISK'/' DELETE, CHANGE'
     *  ,' THE NAME OR ASSIGN A NEW NAME TO FNIM')
  138   FORMAT(' THE FILE ',A5,' REFERS TO SIMULATIONS WITH OTHER
     * PARAMETERS THAN THOSE PRESENT'/' ASSIGN A NEW NAME TO ‡NST OR
     * CHANGE PARAMETERS')
  139   FORMAT(' PAR IFIGE MUST BE 0,1,2 OR 3')
  140   FORMAT(1X,A5,' IS UNDEFINED')
C
        END
```

Subsystem AUXIL

```
C       NAME: AUXIL                                   NUMBER:
C       ----                                          -------
C
C       SUBTITLE: AUXILIARY SUBSYSTEM TO INCLUDE THE SUBROUTINES
C       --------  GENHG,SIMHG,FILCON AND STAT IN HGPAC
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON            DATE:
C       -----------                                   ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                     VERSION:
C       --------                                      -------
C
C
C
C
C-----------------------------------------------------------------------
C
C       SUBROUTINES REQUIRED:   GENHG   POZEMU  DPMPY   SIMHG   WRRESU
C                               SIMH    FILCON  STAT    AUMESS
C                   LIBRARY:    MCNODI
C
C       REF. H. GUNNARSSON: RECURSIVE IDENTIFICATION USING A GENERAL
C                           MODEL STRUCTURE. A PROGRAM PACKAGE AND
C                           SIMULATIONS.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE AUXIL
C
      LOGICAL LSTOP,MSTOP
      COMMON/DESTIN/IDUM,IPART
      COMMON/USER/LSTOP
C
      COMMON IDM1(2),KMESS,MSTOP,INHG,ISYSH,IDH,IDM2(2485),
     * KODD(11),RSUB
C
      GO TO (1,2,3,4,5,6,7,8),IPART
C
C****** IDENTIFICATION ******
    1 CALL IDENT(5HDISCR,5HAUXIL)
```

Subsystem AUXIL

```
          INHG=0
          ISYSH=0
          IDH=0
          RETURN
C
C
C****** DECLARATIONS ******
    2     CALL PAR(RSUB,4HISUB)
          CALL PAR(DT,4HDT   )
          CALL TSAMP(TS,4HTS   )
          RETURN
C****** CONSTANT ASSIGNMENT ******
    3     RSUB=0.
          DT=1.0
          RETURN
C
C
C****** INITIAL ******
    4     KMESS=1
          MSTOP=.FALSE.
          IF(INHG.EQ.1.OR.ISYSH.EQ.1.OR.IDH.EQ.1) GO TO 99
          ISUB=RSUB+0.1
          IF(ISUB.LT.1.OR.ISUB.GT.5) GO TO 98
C
C
          GO TO(41,42,43,44,45),ISUB
C
   41     CALL GENHG
          GO TO 49
C
   42     CALL SIMHG(KODD,SNR,FNIN,FNUT,NCOL)
          IF(NCOL.LT.0) GO TO 49
          CALL SIMH(KODD,SNR,FNIN,FNUT,NCOL)
          IF(KMESS.GT.1) GO TO 100
          GO TO 49
C
   43     CALL STAT
          GO TO 49
C
   44     CALL FILCON
          GO TO 49
C
   45     CALL WRRESU
          GO TO 49
C
   49     LSTOP=.TRUE.
          RETURN
C
C
    5     LSTOP=.TRUE.
          RETURN
C
    6     TS=T+DT
```

Subsystem AUXIL

```
          RETURN
C
     7    RETURN
C
C
     8    IF(MSTOP) CALL AUMESS
          RETURN
C
    98    KMESS=54
          GO TO 100
    99    KMESS=53
   100    MSTOP=.TRUE.
          LSTOP=.TRUE.
          RETURN
          END
```

Subsystem AUXIL

```
C       NAME: GENHG                                      NUMBER:
C       ----                                             ------
C
C       SUBTITLE: PROGRAM FOR GENERATION OF IMPULSE
C       --------  RESPONSES TO BE USED FOR SIMULATION.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE:
C       -----------                                     ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                        VERSION:
C       --------                                         -------
C
C
C
C
C
C
C------------------------------------------------------------------------
C
C       SUBROUTINES REQUIRED:   POZEMU  DPMPY
C
C------------------------------------------------------------------------
C
C
C
C
        SUBROUTINE GENHG
C
C
        LOGICAL MSTOP
        DOUBLE PRECISION A,B,C,E,G,S1,S2
        DIMENSION FNAM(2)
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(531),A(3),B(30),C(30),E(30),
     *  U(130),Y(130),IFIH(10),G,S1,S2
C
C****** THE PROGRAM USES A PART OF THE BLANK COMMON AREA IN HGPAC ******
C**     THE VECTOR A STARTS IN THE FIRST CELL USED BY THE COVARIANCE   **
C**     MATRIX P IN SUBSYSTEM IDHG. SEE HGCOM.                         **
C
C
```

Subsystem AUXIL

```
C
      DATA FNAM(2)/' BIN '/,TYPO/'ONE  '/,TYP1/'NO   '/,ST/'STOP '/
      DATA TEX1/'POLE '/,TEX2/'ZERO '/,IALTM/ 764000/
      DATA YES/'Y    '/,RNO/'N    '/,BLANK/'     '/
C
C****** READ TYP=ONE OR NO ; TYP=ONE MEANS ONE TIME DELAY IN THE ******
C**     NUMERATOR, TYP=NO MEANS NO DELAY; IF TYP=STOP ; STOP          **
C
      IMAX=30
C
   10 WRITE(9,20) IALTM
   20 FORMAT(///' PROGRAM GENHG: GENERATION OF IMPULSE RESPONSES'//
     * ' TYP=STOP  : STOP'/
     * ' TYP=ONE   : ONE TIME DELAY IN THE NUMERATOR'/
     * ' TYP=NO    : NO TIME DELAY'//
     * ' TYP= ',A1)
C
      ITYP=-99
      READ(8,30) TYP
   30 FORMAT(A4)
      IF(TYP.EQ.TYPO) ITYP=0
      IF(TYP.EQ.TYP1) ITYP=1
      IF(TYP.EQ.ST) GO TO 310
      IF(ITYP.EQ.-99) GO TO 10
C
      WRITE(6,40) TYP
   40 FORMAT('1PROGRAM GENHG:',64X,'I'//1X,A4,'TIME DELAY')
C
      CALL POZEMU(B,A,C,IB,IMAX,TEX1)
      CALL POZEMU(E,A,C,IE,IMAX,TEX2)
C
C****** DENOMINATOR AND NUMERATOR POLYNOMIALS ARE NOW COMPUTED. ******
C**     IF TYP=ONE COMPUTE STEADY STATE GAIN G AND DIVIDE THE        **
C**     NUMERATOR COEFFICIENTS BY G TO GET A TRANSFER FUNCTION WITH **
C**     STEADY STATE GAIN EQUAL TO 1.                                **
C
   50 IF(ITYP.EQ.1) GO TO 90
      S1=0.
      DO 60 I=1,IB
   60 S1=S1+B(I)
      S2=0.
      DO 70 I=1,IE
   70 S2=S2+E(I)
      G=S2/S1
      DO 80 I=1,IE
   80 E(I)=E(I)/G
C
C
C****** WRITE DENOMINATOR AND NUMERATOR POLYNOMIALS ******
   90 WRITE(6,100)
  100 FORMAT(/' NUMERATOR POLYNOMIAL:')
      WRITE(6,105) (E(I),I=1,IE)
  105 FORMAT(5D15.6)
```

Subsystem AUXIL

```
      WRITE(6,110)
  110 FORMAT(/' DENOMINATOR POLYNOMIAL:')
      WRITE(6,105) (B(I),I=1,IB)
C
C****** INITIALIZATION ******
      DO 120 I=1,130
      U(I)=0.
  120 Y(I)=0.
      U(31)=1.
C
C****** COMPUTATION OF THE FIRST 100 IMPULSE RESPONSE COEFFICIENTS. ****
C
      IBM1=IB-1
      DO 150 I=31,130
      Y(I)=0.
      DO 130 J=1,IBM1
  130 Y(I)=Y(I)-B(J+1)*Y(I-J)
      DO 140 J=1,IE
  140 Y(I)=Y(I)+E(J)*U(I-J+ITYP)
  150 CONTINUE
C
C
C****** WRITE THE COEFFICIENTS ******
      WRITE(6,160)
  160 FORMAT(/' IMPULSE RESPONSE COEFFICIENTS:')
      DO 180 I=0,24
      K=I+75
      WRITE(6,170) (J,Y(J+31),J=I,K,25)
  170 FORMAT(4(1X,I2,1X,G14.6,1X))
  180 CONTINUE
C
C
C****** GET FILE NAME AND WRITE THE COEFF. ONTO THE FILE ******
  190 WRITE(9,200) IALTM
  200 FORMAT(//' FILE NAME FOR COEFFICIENTS OF IMPULSE RESPONSE:'/
     *' IF FILE NAME=STOP THEN NO FILE IS GENERATED'//' FILE NAME=',A1)
  210 READ(8,220)  FN
  220 FORMAT(A5)
      IF(FN.EQ.ST) GO TO 10
      DO 215 I=1,10
  215 IFIH(I)=0
      IFIH(1)=100000
      IFIH(2)=2
      IFIH(3)=1
      IFIH(10)=1
      FNAM(1)=FN
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.-1) GO TO 260
  230 WRITE(6,240) FN
  240 FORMAT(/' FILE NAME:',A5/)
      CALL ENTER(LDK1,FNAM)
      WRITE(LDK1) (IFIH(I),I=1,10)
      WRITE(LDK1) TYP,BLANK
```

Subsystem AUXIL

```
          RT=0.0
          DO 250 I=31,130
          RT=RT+1.0
          WRITE(LDK1) RT,Y(I)
  250     CONTINUE
C
          CALL CLOSE(LDK1)
C
          GO TO 10
C
  260     WRITE(9,270) FN,IALTM
  270     FORMAT(' A FILE NAMED ',A5,' IS PRESENT ON DISK '/
     *    ' DO YOU WANT IT DELETED *-*0  TYPE Y OR N  (YES OR NO)  ',A1)
          READ(8,280) YN
  280     FORMAT(A1)
          IF(YN.EQ.YES) GO TO 230
          IF(YN.EQ.RNO) GO TO 290
          GO TO 260
  290     WRITE(9,300) IALTM
  300     FORMAT(' CHOOSE ANOTHER NAME =',A1)
          GO TO 210
C
  310     RETURN
          END
```

Subsystem AUXIL

```
C       NAME: POZEMU                                    NUMBER:
C       ----                                            -------
C
C       SUBTITLE: READS AND MULTIPLIES ROOTS (POLES OR
C       --------  ZEROES) TO GIVE THE POLYNOMIAL.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON             DATE:
C       -----------                                    ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                      VERSION:
C       --------                                       -------
C
C
C
C
C
C
C------------------------------------------------------------------------
C
C       SUBROUTINES REQUIRED:   DPMPY
C
C------------------------------------------------------------------------
C
C
C
        SUBROUTINE POZEMU(X,Y,Z,IX,IMAX,TEX)
C
        DOUBLE PRECISION X,Y,Z
C
        DIMENSION X(1),Y(1),Z(1)
C
        DATA ST/'STOP '/,RR/'R    '/,CC/'C    '/,IALTM/ 764000/
C
C
        IX=1
        Y(1)=1.
        X(1)=1.
C
C
C
C****** READ TYPE OF POLE OR ZERO ******
```

```
   10    WRITE(9,20) TEX,TEX,TEX,TEX
   20    FORMAT(//' READ ',A4,'S:'/' T=STOP : NO MORE ',A4,'S'/
     *   ' T=R    : A REAL ',A4/' T=C    : COMPLEX ',A4,'S'/)
C
   30    WRITE(9,40) IALTM
   40    FORMAT(/' T= ',A1)
         READ(8,50) T
   50    FORMAT(A4)
         K=0
         IF(T.EQ.ST) RETURN
         IF(T.EQ.RR) GO TO 60
         IF(T.EQ.CC) GO TO 100
         GO TO 10
C
C
C****** REAL POLE OR ZERO ******
   60    WRITE(9,70) TEX,IALTM
   70    FORMAT(' REAL ',A4,'        = ',A1)
         READ(8,80) Y2
   80    FORMAT(F10.6)
         Y(2)=-Y2
         WRITE(6,90) TEX,Y2
   90    FORMAT(' REAL ',A4,5X,'=',F8.5)
         K=1
         GO TO 140
C
C
C****** COMPLEX POLES OR ZEROES ******
  100    WRITE(9,110) IALTM
  110    FORMAT(' REAL PART      = ',A1)
         READ(8,80) Y2
         WRITE(9,120) IALTM
  120    FORMAT(' IMAGINARY PART = ',A1)
         READ(8,80) Y3
         Y(2)=-2.*Y2
         Y(3)=Y2*Y2+Y3*Y3
         WRITE(6,130) TEX,Y2,Y3
  130    FORMAT(' COMPLEX ',A4,'S =',F8.5,' +/-',F8.5)
         K=2
C
C
C****** MULTIPLICATION ******
  140    CALL DPMPY(Z,IZ,Y,K+1,X,IX)
         IX=IZ
         DO 150 I=1,IZ
  150    X(I)=Z(I)
         IF(IX.EQ.IMAX) GO TO 160
         GO TO 30
C
C
  160    WRITE(9,170) TEX
  170    FORMAT(' ARRAY LIMITS REACHED, NO MORE ',A4,'S POSSIBLE')
         RETURN
```

Subsystem AUXIL

END

Subsystem AUXIL

```
C       NAME: DPMPY                                        NUMBER:
C       ----                                               ------
C
C       SUBTITLE: MULTIPLIES TWO REAL POLYNOMIALS
C       --------  IN DOUBLE PRECISION.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                          VERSION:
C       --------                                           -------
C
C
C
C
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE DPMPY(Z,IDIMZ,X,IDIMX,Y,IDIMY)
C
        DOUBLE PRECISION Z,X,Y
C
C
        DIMENSION Z(1),X(1),Y(1)
C
        IF(IDIMX*IDIMY) 10,10,20
   10   IDIMZ=0
        GO TO 50
C
   20   IDIMZ=IDIMX+IDIMY-1
        DO 30 I=1,IDIMZ
   30   Z(I)=0.
C
        DO 40 I=1,IDIMX
        DO 40 J=1,IDIMY
        K=I+J-1
   40   Z(K)=X(I)*Y(J)+Z(K)
   50   RETURN
        END
```

Subsystem AUXIL

```
C        NAME: SIMHG                                        NUMBER:
C        ----                                               ------
C
C        SUBTITLE: SIMULATION USING IMPULSE RESPONSE MODEL
C        --------
C
C        LANGUAGE: STANDARD FORTRAN
C        --------
C
C        PROGRAM TYPE: SUBROUTINE
C        ------------
C
C        KEYWORDS:
C        --------
C
C        IMPLEMENTOR: HALLGRIMUR GUNNARSSON         DATE: 1978-06-22
C        -----------                                ----
C
C        INSTITUTE:
C        ---------
C        DEPARTMENT OF AUTOMATIC CONTROL
C        LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C        ACCEPTED:                                   VERSION:
C        --------                                    -------
C
C
C
C--------------------------------------------------------------------------
C
C
         SUBROUTINE SIMHG(KODD,SNR,FNIN,FNUT,NCOL)
C
         LOGICAL MSTOP
C
         DIMENSION FNAM(2),KODD(11)
C
         COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(531),ZU(100),ZE(100),KOD(10),
     *   Z(4)
C
         DATA FNAM(2)/' BIN'/,IALTM/ 764000/
C
C
         WRITE(9,10)
   10    FORMAT(' PROGRAM SIMHG'//' PROGRAM FOR SIMULATION',
     *   ' USING IMPULSE RESPONSE MODEL'/)
C
C
C****** READ FILE NAME FOR COEFFICIENTS OF SYSTEM IMPULSE RESPONSE.*****
C****** READ COEFFICIENTS ******
   20    WRITE(9,30) IALTM
   30    FORMAT(' FILE NAME FOR COEFFICIENTS OF SYSTEM IMPULSE RESPONSE'/
     *   ' FILE NAME = ',A1)
         READ(8,40) F1
```

Subsystem AUXIL

```
   40   FORMAT(A5)
        FNAM(1)=F1
        CALL FSTAT(LDK1,FNAM,J)
        IF(J.EQ.0) GO TO 290
        CALL SEEK(LDK1,FNAM)
        READ(LDK1,END=310) (KOD(I),I=1,10)
        DO 50 I=1,100
        READ(LDK1,END=310) Z(1)
   50   ZU(I)=Z(1)
        CALL CLOSE(LDK1)
C****** READ FILE NAME FOR COEFFICIENTS OF NOISE IMPULSE RESPONSE.******
C****** READ COEFFICIENTS ******
        WRITE(9,60) IALTM
   60   FORMAT(/' FILE NAME FOR COEFFICIENTS OF NOISE IMPULSE RESPONSE'/
     *  ' FILE NAME = ',A1)
        READ(8,40) F2
        FNAM(1)=F2
        CALL FSTAT(LDK1,FNAM,J)
        IF(J.EQ.0) GO TO 290
        CALL SEEK(LDK1,FNAM)
        READ(LDK1,END=310) (KOD(I),I=1,10)
        DO 70 I=1,100
        READ(LDK1,END=310) Z(1)
   70   ZE(I)=Z(1)
        CALL CLOSE(LDK1)
C
C
C****** READ FILE NAME WITH THE INPUT DATA ******
        WRITE(9,80) IALTM
   80   FORMAT(/' FILE NAME WITH INPUT DATA = ',A1)
        READ(8,40) FNIN
        FNAM(1)=FNIN
        CALL FSTAT(LDK1,FNAM,J)
        IF(J.EQ.0) GO TO 290
        CALL SEEK(LDK1,FNAM)
        READ(LDK1) (KOD(I),I=1,10)
        IF(KOD(1).LT.2101) GO TO 310
        CALL CLOSE(LDK1)
C
C
C****** READ NCOL INPUT ******
   90   WRITE(9,100) FNIN,IALTM
  100   FORMAT(/' COLUMN NR. OF INPUT DATA IN',A6/' NCOL= ',A1)
        READ(8,110) NCOL
  110   FORMAT(I2)
        IF(NCOL.LE.KOD(2).AND.NCOL.GT.0) GO TO 125
        WRITE(9,120)
  120   FORMAT(' BAD VALUE FOR NCOL')
        GO TO 90
C
C****** READ STARTING VALUE(S) FOR THE NOISE GENERATOR ******
  125   IK=1
  130   WRITE(9,140) IALTM
```

```
  140   FORMAT(/' ODD NUMBER FOR THE NOISE GENERATOR:'/
     *   ' ONE NUMBER FOR EACH SIMULATION, AND A NEGATIVE NUMBER ',
     *   ' TO STOP'//' KODD = ',A1)
        READ(8,150) KODDI
  150   FORMAT(I5)
        KODD(IK)=KODDI
        IF(KODDI.LE.0) GO TO 180
        IF(MOD(KODDI,2).EQ.1) GO TO 170
        WRITE(9,160)
  160   FORMAT(' KODD MUST BE AN ODD NUMBER')
        GO TO 130
  170   IK=IK+1
        IF(IK.LT.11) GO TO 130
        KODD(11)=-1
C
C
C****** READ SIGNAL TO NOISE RATIO ******
  180   WRITE(9,190) IALTM
  190   FORMAT(/' DESIRED SIGNAL TO NOISE RATIO'/' SNR= ',A1)
        READ(8,200) SNR
  200   FORMAT(F10.6)
        IF(SNR.LE.0) GO TO 180
C****** READ FILE NAME FOR THE GENERATED DATA ******
  210   WRITE(9,220) IALTM
  220   FORMAT(/' FILE NAME FOR THE GENERATED DATA = ',A1)
        READ(8,40) FNUT
C
C
C****** PRINT-OUT ******
        WRITE(6,10)
        WRITE(6,230) F1,F2
  230   FORMAT(' FILE NAME SYSTEM RESPONSE : ',A5/
     *   ' FILE NAME NOISE RESPONSE  : ',A5)
C
        WRITE(6,240) FNIN,NCOL
  240   FORMAT(' FILE NAME INPUT DATA        : ',A5/
     *   ' COLUMN NR. INPUT FILE     : ',I2)
C
        WRITE(6,250) FNUT
  250   FORMAT(/' FILE NAME GENERATED DATA   : ',A5/
     *   21X,'COL  1: INPUT SIGNAL')
C
        IK=2
  260   WRITE(6,270) IK,KODD(IK-1)
  270   FORMAT(21X,'COL ',I2,': OUTPUT SIGNAL',5X,'KODD =',I6)
        IK=IK+1
        IF(KODD(IK-1).LE.0) GO TO 275
        GO TO 260
C
  275   WRITE(6,280) SNR
  280   FORMAT(' SIGNAL TO NOISE RATIO       : ',F8.4)
        RETURN
C
```

```
C
C****** ERROR MESSAGES ******
  290   WRITE(9,300) FNAM(1)
  300   FORMAT(' FILE ',A5,' NOT FOUND')
        NCOL=-9
        RETURN
C
  310   CALL CLOSE(LDK1)
        WRITE(9,320) FNAM(1)
  320   FORMAT(' FILE ',A5,' WRONG SIZE')
        NCOL=-9
        RETURN
C
C
        END
```

Subsystem AUXIL

```
C       NAME: SIMH                                      NUMBER:
C       ----                                            ------
C
C       SUBTITLE: SIMULATION USING IMPULSE RESPONSE MODEL
C       --------  CF. PROGRAM SIMHG.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE: 1978-06-22
C       -----------                                     ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                       VERSION:
C       --------                                        -------
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE SIMH(KODD,SNR,FNIN,FNUT,NCOL)
C
        LOGICAL MSTOP
        DIMENSION FIN(2),FUT(2),FW1(2),FW2(2),FW3(2),KODD(11)
C
        COMMON LDK1,LDK2,KMESS,MSTOP,IDM1(9),LDK3,IDM2(521),ZU(100),
     *  ZE(100),KOD(10),Z(4),EA(110),UA(110),Z1(4),W(30)
C
C
        DATA FIN(2),FUT(2),FW1(2),FW2(2),FW3(2)/5*' BIN'/
        DATA FW1(1)/'XHGX'/,FW2(1)/'YHGY'/,FW3(1)/'ZHGZ'/
C
C
        FIN(1)=FNIN
        FUT(1)=FNUT
        IK=1
   10   IKODD=KODD(IK)
        IF(IKODD.LE.0) RETURN
        CALL SEEK(LDK1,FIN)
        READ(LDK1,END=900) (KOD(I),I=1,10)
        KOD(1)=2001
        KOD(2)=3
        KOD(4)=50
        CALL ENTER(LDK2,FW1)
        WRITE(LDK2) (KOD(I),I=1,10)
C
C****** SIMULATION LOOP ******
```

Subsystem AUXIL

```
      SUMY=0.
      SUMY2=0.
      SUME=0.
      SUME2=0.
C
      DO 40 I=1,2101
      CALL MCNODI(IKODD,E)
      READ(LDK1,END=900) (Z(J1),J1=1,NCOL)
      EA(1)=E
      UA(1)=Z(NCOL)
      IF(I.LE.100) GO TO 30
      YU=SCAPRO(ZU(1),1,UA(1),1,100)
      YE=SCAPRO(ZE(1),1,EA(1),1,100)
C
   30 CALL RMOVE(UA(99),-1,UA(100),-1,99)
      CALL RMOVE(EA(99),-1,EA(100),-1,99)
C
C
      IF(I.LE.100) GO TO 40
      Z(1)=UA(1)
      Z(2)=YU
      Z(3)=YE
      SUMY=SUMY+YU
      SUMY2=SUMY2+YU**2
      SUME=SUME+YE
      SUME2=SUME2+YE**2
      WRITE(LDK2) (Z(J2),J2=1,3)
   40 CONTINUE
      CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
C
C
C****** COMPUTE SKAL ******
      VARY=SUMY2/2000-SUMY**2/2001/2000
      VARE=SUME2/2000-SUME**2/2001/2000
      SKAL=SQRT(VARY/VARE/SNR)
C
C
C****** COMPUTE OUTPUT SIGNAL, PLACE ON FILE FW2 ******
      CALL SEEK(LDK1,FW1)
      READ(LDK1,END=900) (KOD(I),I=1,10)
      KOD(1)=2001
      KOD(2)=2
      KOD(4)=50
      CALL ENTER(LDK2,FW2)
      WRITE(LDK2) (KOD(I),I=1,10)
C
   45 READ(LDK1,END=50) (Z(J4),J4=1,3)
      Z1(1)=Z(1)
      Z1(2)=Z(2)+SKAL*Z(3)
      WRITE(LDK2) (Z1(J3),J3=1,2)
      GO TO 45
   50 CONTINUE
```

Subsystem AUXIL

```
      CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
      CALL DLETE(LDK1,FW1,J)
C
C
C****** TRANSFER OUTPUT SIGNAL TO FUT ******
      IF(IK.GT.1) GO TO 60
      CALL RENAM(LDK1,FW2,FUT,J)
      GO TO 80
C
   60 CALL SEEK(LDK1,FUT)
      READ(LDK1) (KOD(I),I=1,10)
      K=KOD(2)
      K1=K+1
      KOD(2)=K1
      CALL SEEK(LDK2,FW2)
      READ(LDK2)
      CALL ENTER(LDK3,FW3)
      WRITE(LDK3) (KOD(I),I=1,10)
   65 READ(LDK1,END=70) (W(J),J=1,K)
      READ(LDK2) RIN,RUT
      W(K1)=RUT
      WRITE(LDK3) (W(J),J=1,K1)
      GO TO 65
C
   70 CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
      CALL CLOSE(LDK3)
C
      CALL DLETE(LDK1,FW2,J)
      CALL DLETE(LDK1,FUT,J)
      CALL RENAM(LDK1,FW3,FUT,J)
C
C
   80 IK=IK+1
      GO TO 10
C
C
  900 KMESS=51
      MSTOP=.TRUE.
      CALL CLOSE(LDK1)
      CALL CLOSE(LDK2)
      CALL CLOSE(LDK3)
      RETURN
      END
```

```
C       NAME: FILCON                                 NUMBER:
C       ----                                         ------
C
C       SUBTITLE: CONVERSION OF SIMNON-GENERATED DATA
C       --------  FILES TO IDPAC COMPATIBLE FILES.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON           DATE:
C       -----------                                  ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                    VERSION:
C       --------                                     -------
C
C
C
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE FILCON
C
        DIMENSION FNIN(2),FNUT(2),FNW(2)
C
        COMMON LDK1,LDK2,IDM(1293),IT(10),VNAM(2),RTDAT(50),WTDAT(50)
C
        DATA STANZ,FNIN(2),FNUT(2),FNW(2),IALTM/'STOP',3*' BIN', 764000/
        DATA RJA/'YES'/,FNW(1)/'QHGQ'/
C
C
C
C****** START ******
        EPS=1.E-6
        WRITE(9,10)
   10   FORMAT(//' PROGRAM FILCON:'/
     *  ' INPUT FILE NAME=STOP  :   STOP'/' PRESS RETURN')
        READ(8,20)
   20   FORMAT(1X)
C
C****** INPUT FILE-NAME ******
   30   WRITE(9,40) IALTM
   40   FORMAT(/' INPUT FILE NAME=',A1)
        READ(8,50) FNIN(1)
```

```
   50    FORMAT(A5)
         IF(FNIN(1).EQ.STANZ) RETURN
         CALL FSTAT(LDK1,FNIN,J)
         IF(J.EQ.-1) GO TO 70
         WRITE(9,60) FNIN(1)
   60    FORMAT(' FILE ',A5,' NOT FOUND')
         GO TO 30
C
   70    CALL SEEK(LDK1,FNIN)
         READ(LDK1) (IT(J),J=1,10)
C****** OUTPUT FILE NAME ******
         WRITE(9,80) IALTM
   80    FORMAT(/' OUTPUT FILE NAME=',A1)
         READ(8,50) FNUT(1)
C
C****** SAMPLE INTERVAL ******
         WRITE(9,90) IALTM
   90    FORMAT(/' SAMPLE INTERVAL=',A1)
         READ(8,100) DT
  100    FORMAT(F10.6)
C
C****** WRITE FILE HEAD AND VARIABLES IN INPUT-FILE ******
         WRITE(9,120) IT
  120    FORMAT(/' INPUT FILE HEAD:'/1X,10I6)
         I=IT(10)
         NVAR=I+1
C
         WRITE(9,130)
  130    FORMAT(' VARIABLES IN INPUT FILE:')
         DO 150 J=1,I
         READ(LDK1) VNAM(1),VNAM(2)
         WRITE(9,140) VNAM(1),VNAM(2)
  140    FORMAT(1X,A5,2X,A5)
  150    CONTINUE
C
C****** COUNT DATA RECORDS, GET START TIME ******
         NV=0
         READ(LDK1,END=170) (RTDAT(J),J=1,NVAR)
         NV=1
         T1=RTDAT(1)
  160    READ(LDK1,END=170)
         NV=NV+1
         GO TO 160
C
  170    CALL CLOSE(LDK1)
         WRITE(9,180) NV
  180    FORMAT(' NUMBER OF DATA RECORDS IN INPUT FILE:',I6)
C
C****** DELETE TIME COLUMN OR NOT ******
         WRITE(9,185) IALTM
  185    FORMAT(/' DELETE TIME COLUMN *-*0 ',A1)
         KSV=1
         READ(8,50) SVAR
```

```
          IF(SVAR.EQ.RJA) KSV=2
          NUVAR=NVAR-KSV+1
C
C
C
C****** CONVERSION TO IDPAC FORMAT ******
          CALL SEEK(LDK1,FNIN)
          READ(LDK1)
          DO 190 J=1,I
  190     READ(LDK1)
C
          IT(1)=NV
          IT(10)=0
          IT(2)=NUVAR
          DX=DT/0.02
          IT(4)=IFIX(DX)
          CALL ENTER(LDK2,FNUT)
          WRITE(LDK2) (IT(KK),KK=1,10)
C
          NS=0
C
C****** FIND T (=RTDAT(1)) = T1 ******
  200     IF(ABS(T1-RTDAT(1)).LT.EPS) GO TO 210
          READ(LDK1,END=220) (RTDAT(J),J=1,NVAR)
          GO TO 200
C
C****** FIND LAST T = T1 AND THEN WRITE WTDAT ONTO OUTPUT FILE ******
  210     CALL RMOVE(RTDAT(1),1,WTDAT(1),1,NVAR)
          READ(LDK1,END=215) (RTDAT(J),J=1,NVAR)
          IF(ABS(T1-RTDAT(1)).LT.EPS) GO TO 210
          WRITE(LDK2) (WTDAT(J),J=KSV,NVAR)
          T1=T1+DT
          NS=NS+1
          GO TO 200
C
  215     WRITE(LDK2) (WTDAT(J),J=KSV,NVAR)
          NS=NS+1
C
  220     CALL CLOSE(LDK1)
          CALL CLOSE(LDK2)
C
C
          CALL SEEK(LDK1,FNUT)
          READ(LDK1) (IT(J),J=1,10)
          IT(1)=NS
          CALL ENTER(LDK2,FNW)
          WRITE(LDK2) (IT(J),J=1,10)
C
  230     READ(LDK1,END=240) (RTDAT(J),J=1,NUVAR)
          WRITE(LDK2) (RTDAT(J),J=1,NUVAR)
          GO TO 230
C
  240     CALL CLOSE(LDK1)
```

Subsystem AUXIL

```
      CALL CLOSE(LDK2)
      CALL DLETE(LDK1,FNUT,J)
      CALL RENAM(LDK1,FNW,FNUT,J)
C
      WRITE(9,250) IALTM
  250 FORMAT(/' DO YOU WANT THE INPUT-FILE DELETED *-* : ',A1)
      READ(8,50) SVAR
      IF(SVAR.EQ.RJA) CALL DLETE(LDK1,FNIN,J)
      GO TO 30
C
      END
```

Subsystem AUXIL

```
C       NAME: STAT                                  NUMBER:
C       ----                                        ------
C
C       SUBTITLE:
C       --------
C       COMPUTATION OF MEANS AND STANDARD DEVIATIONS
C       OF IDENTIFICATION RESULTS FROM IDHG.
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       KEYWORDS:
C       --------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON          DATE:
C       -----------                                 ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                   VERSION:
C       --------                                    -------
C
C
C
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE STAT
C
        DIMENSION LPOL(5),FNAM(2),N(5),KVF(6)
C
        COMMON LDK1,LDK2,IDM(1293),TH(10),SUM(16),SUMKV(16),ST(16),
     *  RMEAN(16),RDEV(16),IPAR(17),RPAR(59)
C
        EQUIVALENCE (IPAR(5),N(1))
C
        DATA FNAM(2)/' BIN'/,IALTM/ 764000/,STANZ/'STOP '/,LPOL(5)/'C'/
        DATA LPOL(1),LPOL(2),LPOL(3),LPOL(4)/'A','F','B','D'/
        DATA KVF(1),KVF(2),KVF(3),KVF(4)/'E ','WE','S ','RS'/
        DATA KVF(5),KVF(6)/'N ','RN'/
C
C
C
C****** START, GET FILE NAME ******
   10   WRITE(9,20)
   20   FORMAT(//' PROGRAM STAT:'//' IF FILE NAME = STOP  :  STOP '/
     *  ' PRESS RETURN')
```

Subsystem AUXIL

```
      READ(8,30)
   30 FORMAT(1X)
   40 WRITE(9,50) IALTM
   50 FORMAT(//' FILE NAME = ',A1)
      READ(8,60) FNAM(1)
   60 FORMAT(A5)
      IF(FNAM(1).EQ.STANZ) RETURN
      CALL FSTAT(LDK1,FNAM,J)
      IF(J.EQ.-1) GO TO 80
      WRITE(9,70) FNAM(1)
   70 FORMAT(' FILE ',A5,' NOT FOUND')
      GO TO 40
C
C****** WRITE OUTPUT PAGE HEAD ******
   80 WRITE(6,90)
   90 FORMAT('1SUBROUTINE STAT:',55X,'I'/' ---------------'/)
      WRITE(6,100)
  100 FORMAT(19X,'2',19X,'2',26X,'2')
      WRITE(6,110)
  110 FORMAT(' V=SUM[HR(I)-HI(I)]',5X,'VR=V/SUM[HR(I)]',5X,'(I=1
     *,100)',5X,'VE=E[RES ]')
      WRITE(6,120)
  120 FORMAT(/11X,'2')
      WRITE(6,130)
  130 FORMAT(' VWE=E[WRES ]  VS=V[SYS]  VRS=VR[SYS]
     *VN=V[NOISE]  VRN=VR[NOISE]')
      WRITE(6,140)
  140 FORMAT(/' ------------------------------------------------------------
     *-----------------'/)
C
C
C****** PRINT DATA ON REALIZATIONS ******
      WRITE(6,150) FNAM(1)
  150 FORMAT(' STATISTICS OF THE FOLLOWING REALIZATIONS:'//
     * '  N  SIMULATION NO.    DATE',23X,'FILE NAME : ',A5)
      CALL SEEK(LDK1,FNAM)
      READ(LDK1) (IPAR(I),I=1,17),(RPAR(J),J=1,59)
      NPDIM=IPAR(2)
      NPS=NPDIM+6
      KANT=0
      KPOINT=0
C
  160 READ(LDK1,END=190) K,NSIM,IYEAR,MONTH,IDAY
      IF(K.NE.-99999) GO TO 180
      KANT=KANT+1
      WRITE(6,170) KANT,NSIM,IYEAR,MONTH,IDAY
  170 FORMAT(1X,I2,5X,I4,8X,I4,'-',I2,'-',I2)
      GO TO 160
C
  180 KPOINT=KPOINT+1
      GO TO 160
C
  190 CALL CLOSE(LDK1)
```

```
C
C
C****** START STATISTICS COMPUTATIONS ******
        IF(MOD(KPOINT,KANT).NE.0) GO TO 390
        KPOINT=KPOINT/KANT
        RK=FLOAT(KANT)
        LINES=KANT+14
C
C
        DO 340 NX=1,KPOINT
C
        DO 200 I=1,16
        SUM(I)=0.
  200   SUMKV(I)=0.
C
        CALL SEEK(LDK1,FNAM)
        READ(LDK1)
        J=NPS+4
        LINES=LINES+J
        IF(64-LINES.GE.0) GO TO 220
        WRITE(6,210) FNAM(1)
  210   FORMAT('1STAT CONT.',39X,'FILE NAME : ',A5)
        LINES=1+J
  220   CONTINUE
C
        DO 230 I=1,NX
  230   READ(LDK1,END=350)
C
  240   READ(LDK1,END=370) ISAMP,K1,K2,K3,K4,(ST(I),I=1,NPS)
        DO 250 I=1,NPS
        SUM(I)=SUM(I)+ST(I)
  250   SUMKV(I)=SUMKV(I)+ST(I)**2
C
        DO 260 I=1,KPOINT
  260   READ(LDK1,END=270)
        GO TO 240
C
  270   CALL CLOSE(LDK1)
        DO 280 I=1,NPS
        RMEAN(I)=SUM(I)/RK
        RDEV(I)=(SUMKV(I)-SUM(I)**2/RK)/(RK-1.)
        IF(RDEV(I).LE.0) RDEV(I)=0.0
  280   RDEV(I)=SQRT(RDEV(I))
C
C****** PRINT RESULTS ******
        WRITE(6,290) ISAMP
  290   FORMAT(//' TIME=',I5)
        WRITE(6,300)
  300   FORMAT(5X,' PAR/LOSSF.',6X,'MEAN',12X,'DEVIATION')
        KPS=N(1)
        KPOL=1
        KPAR=1
        DO 320 I=1,NPDIM
```

Subsystem AUXIL

```
  302   IF(I.LE.KPS.AND.KPS.NE.0) GO TO 305
        KPOL=KPOL+1
        KPS=KPS+N(KPOL)
        KPAR=1
        GO TO 302
  305   WRITE(6,310) LPOL(KPOL),KPAR,RMEAN(I),RDEV(I)
  310   FORMAT(10X,A1,I2,5X,G14.6,4X,G14.6)
        KPAR=KPAR+1
  320   CONTINUE
C
        K=NPDIM+1
        DO 340 I=K,NPS
        LL=I-NPDIM
        WRITE(6,330) KVF(LL),RMEAN(I),RDEV(I)
  330   FORMAT(10X,'V',A2,5X,G14.6,4X,G14.6)
  340   CONTINUE
        GO TO 40
C
C
C****** ERROR MESSAGES ******
  350   WRITE(9,360)
  360   FORMAT(' FEL  OUT 230')
        RETURN
C
  370   WRITE(9,380)
  380   FORMAT(' FEL  OUT  240')
        RETURN
C
  390   WRITE(9,400)
  400   FORMAT(' FEL OUT 190+')
        RETURN
        END
```

Subsystem AUXIL

```
C       NAME: VSTAT                                         NUMBER:
C       ----                                                ------
C
C       SUBTITLE: PRINT RESULTS OF SIMULATIONS
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON                  DATE:
C       -----------                                         ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                           VERSION:
C       --------                                            -------
C
C
C
C-----------------------------------------------------------------------
C
        SUBROUTINE VSTAT
C
        DIMENSION FNAM(2),A(7)
C
        COMMON LDK1,LDK2,IDM(1293),THS(10),R1(10),PO(10),THO(10),
     *  X1(10),X2(10),SUM(3),QSUM(3),VMIN(3),VMAX(3),RMEAN(3),DEV(3),
     *  VV(3)
C
        DATA FNAM(2)/' BIN'/,IALTM/ 764000/,STANZ/'STOP '/
        DATA A(1)/'LIP'/,A(2)/'CMA'/,A(3)/'LIP'/,A(4)/'CMA'/,A(5)/'IV'/
        DATA A(6)/'KJADM'/,A(7)/'EKF'/,B1/'SA'/,B2/'    '/
        DATA RM/'MEAN:'/,DE/'DEV.:'/,VMI/'MIN :'/,VMA/'MAX :'/
C
C
C****** START, GET FILE NAME ******
   10   WRITE(9,20)
   20   FORMAT(//' PROGRAM VSTAT:'//' IF FILE NAME = STOP   :   STOP '/
     *  ' PRESS RETURN')
        READ(8,30)
   30   FORMAT(1X)
   40   WRITE(9,50) IALTM
   50   FORMAT(//' FILE NAME = ',A1)
        READ(8,60) FNAM(1)
   60   FORMAT(A5)
        IF(FNAM(1).EQ.STANZ) RETURN
        CALL FSTAT(LDK1,FNAM,J)
        IF(J.EQ.-1) GO TO 80
```

```
      WRITE(9,70) FNAM(1)
   70 FORMAT(' FILE ',A5,' NOT FOUND')
      GO TO 40
C
   80 WRITE(9,75) IALTM
   75 FORMAT(/' PRINT-OUT FOR ISAMP = IHAT, IHATN, N*KH*IHATN',
     * ' N=1,2..... '/' KH = ',A1)
      READ(8,76) KH
   76 FORMAT(I5)
C
C****** WRITE OUTPUT PAGE HEAD ******
      WRITE(6,90)
   90 FORMAT('1SUBROUTINE VSTAT:',53X,'I'/' ----------------'/)
      WRITE(6,100)
  100 FORMAT(21X,'2',12X,'2')
      WRITE(6,110)
  110 FORMAT(' V = SUM[HR(I)-HI(I)] /SUM[HR(I)]',6X,'I=1,100')
      WRITE(6,120)
  120 FORMAT(/9X,'2')
      WRITE(6,130)
  130 FORMAT(' VE=E[RES ]     VRS=V[SYS]     VRN=V[NOISE]')
C
C
C****** PRINT DATA ON REALIZATIONS ******
      WRITE(6,150) FNAM(1)
  150 FORMAT(///' RESULTS OF THE FOLLOWING REALIZATIONS:'//
     * '  N  SIMULATION NO.    DATE',23X,'FILE NAME : ',A5)
C
      CALL SEEK(LDK1,FNAM)
      READ(LDK1) IST,NPDIM,IX1,METOD,NF,NA,NB,ND,NC,NFST,KD,IPRER,
     * ISTAB,IFIST,INVAU,IHAT,IHATN,WTI,WTO,DELTA,RLIM,RKJ,PFST,DT,
     * FNRS,FNRN,(R1(I),I=1,10),(THO(J),J=1,10),(X1(II),II=1,10),
     * (PO(JJ),JJ=1,10),(X2(IJ),IJ=1,10)
C
C
      KANT=0
      KPOINT=0
C
  160 READ(LDK1,END=190) K,NSIM,IYEAR,MONTH,IDAY
      IF(K.NE.-99999) GO TO 180
      KANT=KANT+1
      WRITE(6,170) KANT,NSIM,IYEAR,MONTH,IDAY
  170 FORMAT(1X,I2,5X,I4,8X,I4,'-',I2,'-',I2)
      GO TO 160
C
  180 KPOINT=KPOINT+1
      GO TO 160
C
  190 CALL CLOSE(LDK1)
C
C
C
      B=B2
```

```
      IF(METOD.LE.2) B=B1
      WRITE(6,195) METOD,A(METOD),B
195   FORMAT(////' IDENTIFICATION METHOD:',I2,2X,A5,2X,A5)
      WRITE(6,200) NF,IPRER,IHAT
200   FORMAT(/' NA    =',I9,9X,'IPRER=',I9,9X,'IHAT =',I9)
      WRITE(6,210) NA,IFIST,IHATN
210   FORMAT(' NF    =',I9,9X,'IFIST=',I9,9X,'IHATN=',I9)
      WRITE(6,220) NB,ISTAB,FNRS
220   FORMAT(' NB    =',I9,9X,'ISTAB=',I9,9X,'FNRS =',4X,A5)
      WRITE(6,230) ND,INVAU,FNRN
230   FORMAT(' ND    =',I9,9X,'INVAU=',I9,9X,'FNRN =',4X,A5)
      WRITE(6,240) NC,NFST
240   FORMAT(' NC    =',I9,9X,'NFST =',I9)
      WRITE(6,250) KD,DT
250   FORMAT(' KD    =',I9,33X,'DT   =',G13.6)
C
      WRITE(6,260) WTI,PFST,RLIM
260   FORMAT(//' WTI  =',G13.6,5X,'PFST =',G13.6,5X,'RLIM =',G13.6)
      WRITE(6,270) WTO,RKJ,DELTA
270   FORMAT(' WTO  =',G13.6,5X,'RKJ  =',G13.6,5X,'DELTA=',G13.6//)
C
      DO 320 I=1,NPDIM
      IF(I.LT.10) WRITE(6,300) I,R1(I),I,THO(I),I,PO(I)
300   FORMAT(' R1(',I1,')  =',G13.6,3X,'THO(',I1,') =',G13.6,3X,'PO(',
     * I1,')  =',G13.6)
      IF(I.GE.10) WRITE(6,310) I,R1(I),I,THO(I),I,PO(I)
310   FORMAT(' R1(',I2,') =',G13.6,3X,'THO(',I2,')=',G13.6,3X,'PO(',
     * I2,') =',G13.6)
320   CONTINUE
      WRITE(6,350) FNAM(1)
C
C
C****** START PRINT-OUT ******
C
      IF(MOD(KPOINT,KANT).NE.0) GO TO 460
      KPOINT=KPOINT/KANT
C
      WRITE(6,335)
335   FORMAT(/' ISAMP  I',7X,'VE',7X,'I',6X,'VRS',7X,'I',6X,'VRN',7X,'
     *I'/' -------+---------------+---------------+---------------+')
      WRITE(6,340)
C
      LINES=4
C
      DO 410 NX=1,KPOINT
C
      ISWR=0
C
      DO 345 I=1,3
      SUM(I)=0.0
      QSUM(I)=0.0
      VMIN(I)=9.9999E+70
345   VMAX(I)=-9.9999E+70
```

Subsystem AUXIL

```
C
         CALL SEEK(LDK1,FNAM)
         READ(LDK1)
         LINES=LINES+KANT+7
         IF(64-LINES.GE.0) GO TO 360
         WRITE(6,350) FNAM(1)
  350    FORMAT('1VSTAT CONT.',38X,'FILE NAME : ',A5)
         WRITE(6,335)
         WRITE(6,340)
         LINES=KANT+11
  360    CONTINUE
C
  340    FORMAT(8X,'I',3(16X,'I'))
C
         DO 370 I=1,NX
  370    READ(LDK1,END=420)
C
  380    READ(LDK1,END=440) ISAMP,K1,K2,K3,K4,(THS(I),I=1,NPDIM),
     *   VV(1),WVE,V1,VV(2),V3,VV(3)
         IF(ISAMP.EQ.IHAT.OR.ISAMP.EQ.IHATN) GO TO 381
         KHH=KH*IHATN
         IF(MOD(ISAMP,KHH).EQ.0) GO TO 381
         CALL CLOSE(LDK1)
         LINES=LINES-KANT-7
         GO TO 410
C
  381    DO 382 I=1,3
         SUM(I)=SUM(I)+VV(I)
         QSUM(I)=QSUM(I)+VV(I)**2
         IF(VV(I).LT.VMIN(I)) VMIN(I)=VV(I)
         IF(VV(I).GT.VMAX(I)) VMAX(I)=VV(I)
  382    CONTINUE
C
         WRITE(6,385) (VV(I),I=1,3)
  385    FORMAT(7X,3(' I ',G14.6),' I')
         IF(ISWR.EQ.1) GO TO 388
         WRITE(6,387) ISAMP
  387    FORMAT('+',I5)
         ISWR=1
  388    DO 390 I=1,KPOINT
  390    READ(LDK1,END=400)
         GO TO 380
C
  400    CALL CLOSE(LDK1)
         WRITE(6,340)
         RK=FLOAT(KANT)
         DO 402 I=1,3
         RMEAN(I)=SUM(I)/RK
         DEV(I)=(QSUM(I)-SUM(I)**2/RK)/(RK-1.)
         IF(VMIN(I).EQ.VMAX(I)) DEV(I)=0.0
         IF(DEV(I).LT.0) DEV(I)=0.0
  402    DEV(I)=SQRT(DEV(I))
C
```

Subsystem AUXIL

```
      WRITE(6,403) RM,(RMEAN(I),I=1,3)
  403 FORMAT(5X,A5,G14.6,2(' I ',G14.6),' I')
      WRITE(6,403) DE,(DEV(I),I=1,3)
      WRITE(6,403) VMI,(VMIN(I),I=1,3)
      WRITE(6,403) VMA,(VMAX(I),I=1,3)
      WRITE(6,340)
      WRITE(6,340)
C
  410 CONTINUE
      GO TO 40
C
C
C****** ERROR MESSAGES ******
  420 WRITE(9,430)
  430 FORMAT(' FEL  OUT 400')
      RETURN
C
  440 WRITE(9,450)
  450 FORMAT(' FEL  OUT  380')
      RETURN
C
  460 WRITE(9,470)
  470 FORMAT(' FEL OUT 330+')
      RETURN
      END
```

Subsystem AUXIL

```
C       NAME: TAFLA                                         NUMBER:
C       ----                                                -------
C
C       SUBTITLE: PRINT RESULTS OF SIMULATIONS
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON       DATE:
C       -----------                              ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                VERSION:
C       --------                                 -------
C
C
C
C----------------------------------------------------------------------
C
        SUBROUTINE TAFLA
C
        DIMENSION FNAM(2),FNUT(2),VKJ(3)
C
        COMMON LDK1,LDK2,IDM(1293),THS(10),R1(10),PO(10),THO(10),
     *  X1(10),X2(10),SUM(3),QSUM(3),VMIN(3),VMAX(3),RMEAN(3),DEV(3),
     *  VV(3),RME(3),DE(3),XMI(3),XMA(3)
C
        DATA FNAM(2)/' BIN'/,IALTM/ 764000/,STANZ/'STOP '/
        DATA RNOF/'ENDFI'/,FNUT(1)/'XIBHX'/,FNUT(2)/' BIN'/,NTS/'NEWTS'/
        DATA VKJ(1)/'VE'/,VKJ(2)/'VRS'/,VKJ(3)/'VRN'/
C
    5   ISTAR=-999
        LST=-99
C
C
C****** START, GET FILE NAME ******
   10   WRITE(9,20)
   20   FORMAT(//' PROGRAM TAFLA:'//' IF FILE NAME = STOP   :  STOP '/
     *  ' IF FILE NAME = ENDFI   :  PRINT TAFLA'/
     *  ' IF FILE NAME = NEWTS   :  NEW ISAMP'/
     *  ' PRESS RETURN')
        READ(8,30)
   30   FORMAT(1X)
   40   WRITE(9,50) IALTM
   50   FORMAT(//' FILE NAME = ',A1)
        READ(8,60) FNAM(1)
```

Subsystem AUXIL

```
   60    FORMAT(A5)
         IF(FNAM(1).EQ.STANZ) RETURN
         IF(FNAM(1).EQ.RNOF) GO TO 500
         IF(FNAM(1).EQ.NTS) GO TO 80
         IF(LST.EQ.-99) GO TO 80
   63    CALL FSTAT(LDK1,FNAM,J)
         IF(J.EQ.-1) GO TO 90
   65    WRITE(9,70) FNAM(1)
   70    FORMAT(' FILE ',A5,' NOT FOUND OR WRONG TYPE')
         GO TO 40
C
   80    WRITE(9,75) IALTM
   75    FORMAT(/' ISAMP = ',A1)
         ISTAR=-999
         READ(8,76) KSAMP
   76    FORMAT(I5)
         IF(LST.EQ.-99) GO TO 63
         GO TO 40
C
C
   90    LST=0
C
         CALL SEEK(LDK1,FNAM)
         READ(LDK1) IST,NPDIM,IX1,METOD,NF,NA,NB,ND,NC,NFST,KD,IPRER,
      *  ISTAB,IFIST,INVAU,IHAT,IHATN,WTI,WTO,DELTA,RLIM,RKJ,PFST,DT,
      *  FNRS,FNRN,(R1(I),I=1,10),(THO(J),J=1,10),(X1(II),II=1,10),
      *  (PO(JJ),JJ=1,10),(X2(IJ),IJ=1,10)
C
C
         KANT=0
         KPOINT=0
C
  160    READ(LDK1,END=190) K,NSIM,IYEAR,MONTH,IDAY
         IF(K.NE.-99999) GO TO 180
         KANT=KANT+1
         GO TO 160
C
  180    KPOINT=KPOINT+1
         GO TO 160
C
  190    CALL CLOSE(LDK1)
C
C
         IF(ISTAR.NE.-999) GO TO 200
         CALL ENTER(LDK2,FNUT)
         ISTAR=0
C
C
  200    IF(MOD(KPOINT,KANT).NE.0) GO TO 65
         KPOINT=KPOINT/KANT
C
C
         DO 345 I=1,3
```

```
      SUM(I)=0.0
      QSUM(I)=0.0
      VMIN(I)=9.9999E+70
  345 VMAX(I)=-9.9999E+70
C
      CALL SEEK(LDK1,FNAM)
      READ(LDK1)
C
C
  380 READ(LDK1,END=65) ISAMP,K1,K2,K3,K4,(THS(I),I=1,NPDIM),
     * VV(1),WVE,V1,VV(2),V3,VV(3)
C
      IF(ISAMP.EQ.KSAMP) GO TO 381
      GO TO 380
  381 DO 382 I=1,3
      SUM(I)=SUM(I)+VV(I)
      QSUM(I)=QSUM(I)+VV(I)**2
      IF(VV(I).LT.VMIN(I)) VMIN(I)=VV(I)
      IF(VV(I).GT.VMAX(I)) VMAX(I)=VV(I)
  382 CONTINUE
C
      DO 390 I=1,KPOINT
  390 READ(LDK1,END=400)
      GO TO 380
C
  400 CALL CLOSE(LDK1)
      RK=FLOAT(KANT)
      DO 402 I=1,3
      RMEAN(I)=SUM(I)/RK
      DEV(I)=(QSUM(I)-SUM(I)**2/RK)/(RK-1.)
      IF(VMIN(I).EQ.VMAX(I)) DEV(I)=0.0
      IF(DEV(I).LT.0) DEV(I)=0.0
  402 DEV(I)=SQRT(DEV(I))
C
C
  410 CONTINUE
      WRITE(LDK2) FNAM(1),METOD,NF,NA,NB,ND,NC,KD,IPRER,ISTAB,
     * IFIST,(RMEAN(I),DEV(I),VMIN(I),VMAX(I),I=1,3)
      GO TO 40
C
C
  500 CALL CLOSE(LDK2)
C
C
      DO 650 KJ=1,3
      CALL SEEK(LDK2,FNUT)
C
C****** WRITE OUTPUT PAGE HEAD ******
      WRITE(6,520)
  520 FORMAT('1SUBROUTINE TAFLA:',53X,'I'/' ----------------'/)
      WRITE(6,540) VKJ(KJ)
  540 FORMAT(' MODEL INF.    I',22X,A3,22X,'I')
      WRITE(6,550)
```

```
  550   FORMAT(' --------------+-----------+-----------+----------
    *-+-----------+')
        WRITE(6,555)
  555   FORMAT(' FN IPR IST IFII',4(11X,'I'))
        WRITE(6,556)
  556   FORMAT(' A F B D C K M I',4(11X,'I'))
        WRITE(6,560)
  560   FORMAT('+',14X,'I    MEAN    I    DEV    I    MIN    I    MAX',
    *   '    I')
        WRITE(6,550)
C
C
  565   READ(LDK2,END=600) FN,MET,IF,IA,IB,ID,IC,KD,IPR,ISTA,IFI,
    *   (RME(I),DE(I),XMI(I),XMA(I),I=1,3)
C
        WRITE(6,570)
  570   FORMAT(15X,'I',4(11X,'I'))
        WRITE(6,580) FN,IPR,ISTA,IFI
  580   FORMAT(1X,A5,2I2,I4,' I',4(11X,'I'))
        WRITE(6,590) IF,IA,IB,ID,IC,KD,MET,RME(KJ),DE(KJ),XMI(KJ),
    *   XMA(KJ)
  590   FORMAT(7I2,' I ',4(F9.6,' I '))
        GO TO 565
C
  600   CALL CLOSE(LDK2)
C
  650   CONTINUE
C
        GO TO 5
C
        END
```

Subsystem AUXIL

```
C       NAME: AUMESS                                     NUMBER:
C       ----                                             ------
C
C       SUBTITLE: ERROR MESSAGES FOR THE SUBSYSTEM AUXIL
C       --------
C
C       LANGUAGE: STANDARD FORTRAN
C       --------
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       AUTHOR AND
C       IMPLEMENTOR: HALLGRIMUR GUNNARSSON              DATE:
C       -----------                                      ----
C
C       INSTITUTE: DEPARTMENT OF AUTOMATIC CONTROL
C       ---------  LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C       ACCEPTED:                                        VERSION:
C       --------                                         -------
C
C
C
C-------------------------------------------------------------------------
C
        SUBROUTINE AUMESS
C
        LOGICAL MSTOP
C
        COMMON IDM1(2),KMESS,MSTOP
C
        IF(KMESS.LT.51.OR.KMESS.GT.54) RETURN
C
        LU=9
        MSTOP=.FALSE.
        IAMESS=KMESS-50
C
        WRITE(LU,4000)
 4000   FORMAT(' SYSTEM AUXIL')
C
        GO TO(1,2,3,4),IAMESS
C
    1   WRITE(LU,151)
        GO TO 100
    2   WRITE(LU,152)
        GO TO 100
    3   WRITE(LU,153)
        GO TO 100
    4   WRITE(LU,154)
        GO TO 100
C
  100   RETURN
```

Subsystem AUXIL

```
C
C
  153   FORMAT(' SYSTEM AUXIL IS NOT TO BE USED TOGETHER WITH OTHER '
     *  'SYSTEMS')
  154   FORMAT(' PAR ISUB MUST BE .GE.1.AND.LE.5')
  151   FORMAT(' END OF FILE  (INPUT TO SIMHG)')
  152   FORMAT(' HVERNIG TOKST THETTA *-*  FEL I PROGRAM *-* ')
C
        RETURN
        END
```