ARdock, an Auto-Regressive model analyzer

Makio Ishiguro[1)3)] , Hiroko Kato[2)] and Hirotugu Akaike[1)3)]

1) The Institute of Statistical Mathematics
2) NTT Communication Science Laboratories
3) The Graduate University for Advanced Studies

Mathematical formulas for model fitting and interpretation of the linear multivariate time series and system are collected. A model, E-MARTS model is proposed for the purpose of system analysis. Model fitting is performed by the least squares method via Householder transformation and the numerical maximization of the log likelihood whose calculation is realized by applying the Kalman filtering technique. The complete ARdock package is composed of this monograph and a source file that contains all the necessary computational routines. This package is so designed that it can readily be used by the users interested in analyzing their own data, and also serves as a bank of subroutines for the general use by statistical model builders. The ARdock package is distributed from ISMLIB of the Institute of Statistical Mathematics(http://www.ism.ac.jp/software/ismlib/soft.e.html).

COMPUTER SCIENCE MONOGRAPHS

A publication

of

The Institute of Statistical Mathematics

Contents

0 Subject Index

N.B. Numbers refer to the related pages.

1   Introduction

Multivariate AR model(MAR model) is a compact package of information about the system to which the model is fitted. The information is stored in the form of the coefficient matrices and the variance covariance matrix of innovations. The physical meaning of these matrices are difficult to grasp except for the models of order 1 or 2. However, by virtue of the linearity of the AR model, which allows the use of the principle of superposition, we can obtain sufficient quantitative knowledge about the system by finite number of purposefully designed experiments or simulations.

We made ARdock package composed of (1) this monograph and (2) FORTRAN source code file. There are two aims:

1. to offer a powerful tool of system analysis and

2. to supply a package of statistical softwares for general use by model builders.

The construction of this monograph is as follows. Equations and notations of the MAR model and its relatives are collected in Section 2. Tools to study the physical nature of a MAR model are collected in Section 3. Equations concerning the fitting of models of Section 2 are given in Section 4. The implementation of the analytic computations explained in sections 2 through 4 as a FORTRAN program is detailed in Section 5. Section 6 is the user's manual of ARdock where examples of application are included.

2   AR model

2.1   AR model and Linear System

Let us consider a linear system of the K variables $(x_{1t}, x_{2t}, \ldots, x_{kt})$ driven by a colored noise $u_t = (u_{1t}, u_{2t}, \ldots, u_{kt})$:

$$(1) \qquad x_{kt} = \sum_{j \neq k} \sum_{m=1}^{M'} a_{kjm} x_{j(t-m)} + u_{kt}$$

$$(2) \qquad u_{kt} = \sum_{m=1}^{M'} b_{km} u_{k(t-m)} + \varepsilon_{kt},$$

where u_{kt} is the movement of the system originating in x_k (ε_{kt} is a white noise). Let $M = 2M'$. With a convention that not explicitly defined variables are treated as zeros, from equations (1) and (2) we have

$$x_{kt} - \sum_{m=1}^{M} b_{km} x_{k(t-m)} = \sum_{j \neq k} \left[\sum_{m=1}^{M} \left\{ a_{kjm} - \sum_{n=1}^{m-1} b_{kn} a_{kj(m-n)} \right\} x_{j(t-m)} \right] + \varepsilon_{kt}.$$

Defining

$$(3) \qquad \begin{cases} A_{kkm} & = \quad b_{km} \\ A_{kjm} & = \quad a_{kjm} - \displaystyle\sum_{n=1}^{m-1} b_{kn} a_{kj(m-n)} \end{cases}$$

we have an AR model:

$$(4) \qquad x_{kt} = \sum_{m=1}^{M} \sum_{j=1}^{K} A_{kjm} x_{j(t-m)} + \varepsilon_{kt},$$

or

$$(5) \qquad \boldsymbol{x}_t = \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m} + \boldsymbol{\varepsilon}_t,$$

in matrix notation, where $\boldsymbol{x}_t = (x_{1t}, x_{2t}, \ldots, x_{Kt})^T$. We define a general Gaussian AR model by (5) with the assumption that

(6)
$$\varepsilon_t \sim N(0, \Sigma).$$

This model has

(7)
$$MK^2 + (K+1)K/2.$$

parameters:

(8)
$$\{\Sigma, A_1, A_2, \ldots, A_M\}.$$

The reasoning could be the other way around. We can start from the AR model (4). It is equivalent to

(9)
$$x_{kt} - \sum_{m=1}^{M} A_{kkm} x_{k(t-m)} = \sum_{j \neq k} \left[\sum_{m=1}^{M} A_{kjm} x_{j(t-m)} \right] + \varepsilon_{kt}.$$

Define the noise series $\{u_{kt}\}$ generated from ε_{kt} by the feedback loop by

(10)
$$u_{kt} = \sum_{m=1}^{M} A_{kkm} u_{k(t-m)} + \varepsilon_{kt}$$

and, assuming the invertibility, express it in the moving average form

(11)
$$u_{kt} = \varepsilon_{kt} - \sum_{m=1}^{\infty} a_{kkm} \varepsilon_{k(t-m)}.$$

By formally applying the moving average operation to the both sides of (9), we have

$$
\begin{aligned}
x_{kt} &= \sum_{j \neq k} \sum_{m=1}^{M} A_{kjm} \left(x_{j(t-m)} - \sum_{n=1}^{\infty} a_{kkn} x_{j(t-n-m)} \right) + u_{kt} \\
&= \sum_{j \neq k} \left\{ \sum_{m=1}^{M} A_{kjm} x_{j(t-m)} - \sum_{m=1}^{M} \sum_{n=1}^{\infty} A_{kjm} a_{kkn} x_{j(t-n-m)} \right\} + u_{kt} \\
&= \sum_{j \neq k} \left\{ \sum_{m=1}^{M} A_{kjm} x_{j(t-m)} - \sum_{n=1}^{\infty} \sum_{s=2}^{\infty} A_{kj(s-n)} a_{kkn} x_{j(t-s)} \right\} + u_{kt} \\
&= \sum_{j \neq k} \left\{ A_{kj1} x_{j(t-1)} + \sum_{s=2}^{\infty} \left(A_{kjs} - \sum_{n=1}^{\infty} a_{kkn} A_{kj(s-n)} \right) x_{j(t-s)} \right\} + u_{kt} \\
&= \sum_{j \neq k} \sum_{s=1}^{\infty} a_{kjs} x_{j(t-s)} + u_{kt},
\end{aligned}
$$

where

(12)
$$
\begin{cases}
a_{kj1} = A_{kj1} \\
a_{kjs} = A_{kjs} - \sum_{n=1}^{\infty} a_{kkn} A_{kj(s-n)}
\end{cases}.
$$

Using the relations between a_{kjs} and A_{kjs} analogous to that between u_{kt} and ε_{kt} expressed by equations (11) and (10), we can rewrite the second equation of (12) in the moving average form to the auto-regressive form to get

(13)
$$
\begin{cases}
a_{kj1} = A_{kj1} \\
a_{kjm} = \sum_{n=1}^{m-1} A_{kkm} a_{kj(m-n)} + A_{kjm} \quad (m = 2, \ldots)
\end{cases}.
$$

Eqs. (3) and (13) imply the equivalence relation:

(14)
$$a_{kjm} = 0 \quad (m = 1, 2, \ldots) \iff A_{kjm} = 0 \quad (m = 1, 2, \ldots).$$

2.2 PAR model

There are cases where each variable represents a stationary variation around a deterministic trend. A model for this type of data is defined by

$$x_{it} - \sum_{p=0}^{P} \tilde{b}_{ip} t^p = \sum_{j=1}^{K} \sum_{m=1}^{M} A_{ijm}\left(x_{j(t-m)} - \sum_{p=0}^{P} \tilde{b}_{jp}(t-m)^p\right) + \varepsilon_{it} \quad (i = 1, \ldots, K).$$

or

$$\boldsymbol{x}_t - \sum_{p=0}^{P} \tilde{\boldsymbol{b}}_p t^p = \sum_{m=1}^{M} A_m\left(\boldsymbol{x}_{t-m} - \sum_{p=0}^{P} \tilde{\boldsymbol{b}}_p(t-m)^p\right) + \boldsymbol{\varepsilon}_t.$$

This is equivalent to

$$(15) \qquad \boldsymbol{x}_t = \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m} + \sum_{p=0}^{P} \boldsymbol{b}_p t^p + \boldsymbol{\varepsilon}_t,$$

if $\boldsymbol{b}_p = (b_{1p}, b_{2p}, \ldots, b_{Kp})^T$ $(p = 0, 1, \ldots, P)$ are suitably chosen. The number of free parameters in this model is

$$(16) \qquad (P+1)K + MK^2 + (K+1)K/2.$$

We will call this a Polynomial trend AR(PAR) model. If $P = 2$, the equivalence

$$
\begin{aligned}
\boldsymbol{x}_t &= \tilde{\boldsymbol{b}}_0 + \tilde{\boldsymbol{b}}_1 t + \sum_{m=1}^{M} A_m(\boldsymbol{x}_{t-m} - \tilde{\boldsymbol{b}}_0 - \tilde{\boldsymbol{b}}_1(t-m)) + \boldsymbol{\varepsilon}_t \\
&= B_0 \tilde{\boldsymbol{b}}_0 + B_0 \tilde{\boldsymbol{b}}_1 t - B_1 \tilde{\boldsymbol{b}}_1 + \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m} + \boldsymbol{\varepsilon}_t \\
&= \boldsymbol{b}_0 + \boldsymbol{b}_1 t + \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m} + \boldsymbol{\varepsilon}_t
\end{aligned}
$$

holds, where we assumed

$$B_p = - \sum_{m=0}^{M} m^p A_m.$$

If B_0 is not singular, the transformation from $\{\boldsymbol{b}_0, \boldsymbol{b}_1\}$ to $\{\tilde{\boldsymbol{b}}_0, \tilde{\boldsymbol{b}}_1\}$ is given by

$$\tilde{\boldsymbol{b}}_1 = B_0^{-1} \boldsymbol{b}_1$$

$$\tilde{\boldsymbol{b}}_0 = B_0^{-1}(\boldsymbol{b}_0 + B_1 \tilde{\boldsymbol{b}}_1).$$

[State space representation] A state space representation of the PAR model (15) is given by

$$(17) \qquad \begin{cases} \boldsymbol{z}_t = \Psi \boldsymbol{z}_{t-1} + \Gamma \boldsymbol{\varepsilon}_t \\ \boldsymbol{x}_t = \Phi \boldsymbol{z}_t, \end{cases}$$

$$\Psi = \begin{pmatrix} F_X & F_{XT} \\ 0 & F_T \end{pmatrix}$$

where F_X is a $KM \times KM$ matrix, F_{XT} is a $KM \times K(P+1)$ matrix and F_T is a $K(P+1) \times K(P+1)$ matrix. An ordinary MAR case is obtained by formally setting $P = -1$. If, for example, $P = 1$ and $K = 2$,

$$F_X = \begin{pmatrix} A_1, & A_2, & \cdots & A_M \\ I, & O, & \cdots & O \\ \vdots & & \cdots & \vdots \\ O, & \cdots & I, & O \end{pmatrix}, \quad F_{XT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad F_T = \begin{pmatrix} 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\Gamma = \begin{pmatrix} I \\ O \\ \vdots \\ O \end{pmatrix}, \quad \Phi = (I, 0, 0, \ldots, 0),$$

and

$$\boldsymbol{z}_t^T = (\boldsymbol{x}_t^T, \boldsymbol{x}_{t-1}^T, \ldots, \boldsymbol{x}_{t-M+1}^T, q_{1t}, q_{1(t-1)}, q_{2t}, q_{2(t-1)}).$$

2.3 MARTS model

Suppose that K-dimensional time series $\mathbf{y}_n = (y_{1n}, \ldots, y_{Kn})^T$ $(n = 1, \ldots, N)$ has the structure

(18)
$$\mathbf{y}_n = \boldsymbol{x}_n + \boldsymbol{t}_n + \boldsymbol{s}_n + \boldsymbol{r}_n,$$

where $\boldsymbol{x}_n, \boldsymbol{t}_n$, and $\boldsymbol{s}_n$ represent the AR, trend and seasonal component, respectively, and $\boldsymbol{r}_n$ represents the observation noise. Each element of the trend component $\boldsymbol{t}_n = (t_{1n}, t_{2n}, \ldots, t_{Kn})^T$ is assumed to satisfy the stochastic difference equation

(19)
$$\begin{aligned} \Delta^{d+1} t_{kn} &= \xi_{kn} \quad (k = 1, \ldots, K \ n = 1, \ldots, N), \\ \boldsymbol{\xi}_n &= (\xi_{1n}, \xi_{1n}, \ldots, \xi_{Kn})^T \sim N(\mathbf{0}, \Sigma_T), \\ \Sigma_T &= \begin{pmatrix} \sigma_{T1}^2 & & & \\ & \sigma_{T2}^2 & & \\ & & \ddots & \\ & & & \sigma_{Tk}^2 \end{pmatrix} \end{aligned}$$

where $\Delta t_n = t_n - t_{n-1}$. $\{\boldsymbol{\xi}_n\}$ is a white noise sequence. The length of the period seasonal component of the k-th channel $\{s_{k1}, s_{k2}, \ldots, s_{kn}\}$ is denoted by L_k and it is assumed that

(20)
$$\begin{aligned} \sum_{m=0}^{L_k-1} s_{k(n-m)} &= \eta_{kn} \quad (k = 1, \ldots, K \ n = 1, \ldots, N), \\ \boldsymbol{\eta}_n &= (\eta_{1n}, \eta_{2n}, \ldots, \eta_{Kn})^T \sim N(\mathbf{0}, \Sigma_S), \\ \Sigma_S &= \begin{pmatrix} \sigma_{S1}^2 & & & \\ & \sigma_{S2}^2 & & \\ & & \ddots & \\ & & & \sigma_{Sk}^2 \end{pmatrix}. \end{aligned}$$

For quarterly time series $L_k = 4$. For series without seasonal variation we assign $L_k = 1$ and $\sigma_{Sk}^2 = 0$. $\{\boldsymbol{\eta}_n\}$ is a white noise sequence. We also assume that observation noise $\{\boldsymbol{r}_n\}$ is a white noise sequence: $\boldsymbol{r}_n \sim N(\mathbf{0}, \Sigma_R)$.

$$\Sigma_R = \begin{pmatrix} \sigma_{R1}^2 & & \\ & \ddots & \\ & & \sigma_{Rk}^2 \end{pmatrix}$$

We call this model MARTS(Multivariate AR around Trend and Seasonal component) model. This model is a multivariate version of the DECOMP model of Kitagawa(1981).

Define

$$\begin{aligned} \boldsymbol{z}_n^T &\equiv (\boldsymbol{x}_n^T, \boldsymbol{x}_{n-1}^T, \ldots, \boldsymbol{x}_{n-M+1}^T, t_{1n}, t_{1(n-1)}, \ldots, \\ &\quad t_{1(n-d)}, t_{2n}, \ldots, t_{K(n-d)}, s_{1n}, \ldots, s_{1(n-L_1+2)}, s_{2n}, \ldots, s_{K(n-L_K+2)}) \end{aligned}$$

and

$$\boldsymbol{v}_n^T \equiv (\boldsymbol{\varepsilon}_n^T, \boldsymbol{\xi}_n^T, \boldsymbol{\eta}_n^T), \quad \Sigma_V = E\{\boldsymbol{v}_n \boldsymbol{v}_n^T\}$$

then suitably defining matrices Ψ, Γ and Φ, a state space representation:

$$
(21) \qquad \left\{
\begin{array}{rcl}
z_n &=& \Psi z_{n-1} + \Gamma v_n \\
y_n &=& \Phi z_n + r_n
\end{array}
\right. .
$$

is obtained. The parameter of this model is

$$
(22) \qquad \boldsymbol{\theta} = \text{Independent elements in } \{\Sigma_T, \Sigma_S, \Sigma_R, z_0, \Sigma, A_1, \ldots, A_M\}.
$$

Ψ, Φ, Γ and Σ_V are

$$
\Psi = \begin{pmatrix} F_X & & \mathbf{0} \\ & F_T & \\ \mathbf{0} & & F_S \end{pmatrix}, \quad
\Phi = (H_X, H_T, H_S), \quad
\Gamma = \begin{pmatrix} G_X \\ G_T \\ G_S \end{pmatrix} \quad \text{and} \quad
\Sigma_V = \begin{pmatrix} \Sigma & & \mathbf{0} \\ & \Sigma_T & \\ \mathbf{0} & & \Sigma_S \end{pmatrix}.
$$

If $K = 2, M = 2, d = 2$ and $L_1 = L_2 = 4$, for example,

$$
F_X = \begin{pmatrix} A_1 & A_2 \\ I & \mathbf{0} \end{pmatrix}, \quad
F_T = \begin{pmatrix} 2 & -1 & & \mathbf{0} \\ 1 & 0 & & \\ & & 2 & -1 \\ \mathbf{0} & & 1 & 0 \end{pmatrix},
$$

$$
F_S = \begin{pmatrix}
-1 & -1 & -1 & & & \\
1 & 0 & 0 & & \mathbf{0} & \\
0 & 1 & 0 & & & \\
& & & -1 & -1 & -1 \\
& \mathbf{0} & & 1 & 0 & 0 \\
& & & 0 & 1 & 0
\end{pmatrix},
$$

$$
G_X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad
G_T = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad
G_S = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},
$$

$$
H_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad
H_T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad
H_S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.
$$

2.4 SAR model

To investigate the causality relation between variables, we use a Subset AR(SAR) model. The SAR model specified by the *index matrix S* is defined by:

$$
(23) \qquad x_{it} = \sum_{\{\forall j | S_{ij}=1\}} \sum_{m=1}^{M} A_{ijm} x_{j(t-m)} + \varepsilon_{it} \quad (i = 1, \ldots, K),
$$

where S is a $K \times K$ matrix of indicators each element takes the value 0 or 1. For a given S the number of free parameter of the model is

$$
(24) \qquad M \sum_{i,j} S_{i,j} + (K+1)K/2.
$$

2.5 TAR model

To introduce another physically meaningful structure to AR model, let us start from an ARX (AR with exogenous variables) model

$$
(25) \qquad \boldsymbol{y}_t = \sum_{m=1}^{M} A_m^{yy} \boldsymbol{y}_{t-m} + \sum_{m=1}^{M} A_m^{yw} \boldsymbol{w}_{t-m} + \boldsymbol{\zeta}_t,
$$

where $\boldsymbol{y}_t$ is a K_y-vector of controlled variables and $\boldsymbol{w}_t$ is a K_w-vector of control variables. If the task of stabilizing the system is posed, and if some effort is to be payed to avoid too abrupt motion in control, then the problem can be formalized as one to minimize the cost:

$$(26) \qquad J = E\left\{ \sum_{t=1}^{N} \boldsymbol{y}_{t+1}^T U_t \boldsymbol{y}_{t+1} + \boldsymbol{w}_t^T R_t \boldsymbol{w}_t + (\boldsymbol{y}_{t+1} - \boldsymbol{y}_t)^T U_t^\Delta (\boldsymbol{y}_{t+1} - \boldsymbol{y}_t) \right.$$
$$\left. + (\boldsymbol{w}_t - \boldsymbol{w}_{t-1})^T R_t^\Delta (\boldsymbol{w}_t - \boldsymbol{w}_{t-1}) \right\},$$

where we assume that

$$U_t = \tau^t U = \tau^t \begin{pmatrix} u_1 & & 0 \\ & \cdot & \\ 0 & & u_{K_y} \end{pmatrix}$$

$$R_t = \tau^t R = \tau^t \begin{pmatrix} u_{(K_y+1)} & & 0 \\ & \cdot & \\ 0 & & u_K \end{pmatrix}$$

$$U_t^\Delta = \tau^t U^\Delta = \tau^t \begin{pmatrix} u_1^\Delta & & 0 \\ & \cdot & \\ 0 & & u_{K_y}^\Delta \end{pmatrix}$$

$$R_t^\Delta = \tau^t R^\Delta = \tau^t \begin{pmatrix} u_{(K_y+1)}^\Delta & & 0 \\ & \cdot & \\ 0 & & u_K^\Delta \end{pmatrix}.$$

$K = K_y + K_w$. The optimal control for this cost is easily obtained(see Appendix E). Under this control the dynamics of the whole system is described by

$$(27) \qquad \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{w}_t \end{pmatrix} = \sum_{m=1}^{M} \begin{pmatrix} A_m^{yy} & A_m^{yw} \\ A_m^{wy} & A_m^{ww} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_{t-m} \\ \boldsymbol{w}_{t-m} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\zeta}_t \\ G_1 \boldsymbol{\zeta}_t \end{pmatrix}.$$

Replace the noise $G_1 \boldsymbol{\zeta}_t$ with $\boldsymbol{\xi}_t$ independent of $\boldsymbol{\zeta}_t$, then we get the *Teleological* AR(TAR) model(Ishiguro (1998))

$$(28) \qquad \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{w}_t \end{pmatrix} = \sum_{m=1}^{M} \begin{pmatrix} A_m^{yy} & A_m^{yw} \\ A_m^{wy} & A_m^{ww} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_{t-m} \\ \boldsymbol{w}_{t-m} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\zeta}_t \\ \boldsymbol{\xi}_t \end{pmatrix}$$

derived from the ARX model(25) and cost(26).

The parameter set of TAR model is

$$(29) \qquad \{\tau, u_1, \ldots, u_K, u_1^\Delta, \ldots, u_K^\Delta\} \cup \{A_m^{yy}, A_m^{yw}; m = 1, \ldots, M\} \cup \{\Sigma_y, \Sigma_w\}.$$

Since we have to set a constraint, say $u_1 = 1$, and $\Sigma_y(= E\{\boldsymbol{\zeta}_t \boldsymbol{\zeta}_t^T\})$ and $\Sigma_w(= E\{\boldsymbol{\xi}_t \boldsymbol{\xi}_t^T\})$ are symmetric matrices, the number of free parameters of this model is

$$2 \times K + K_y \times K \times M + (K_y + 1)K_y/2 + (K_w + 1)K_w/2.$$

If TAR model is successfully fitted to given time series data, parameters τ, U, R, U^Δ and R^Δ will reveal the teleological nature or the subsystem which generates $\boldsymbol{w}$.

2.6   E-MARTS model

MARTS, SAR and TAR are not necessarily mutually exclusive. It is easy to think a TAR model whose ARX part has SAR structure. The AR component of the MARTS model could be a TAR model whose ARX part has an SAR structure. This consideration leads to the general model

$$(30) \qquad \boldsymbol{y}_n = \boldsymbol{x}_n + \boldsymbol{q}_n + \boldsymbol{r}_n,$$

where

- the dynamic component $\boldsymbol{x}_n$ is either of

  > MAR
  > MAR ∩ TAR
  > SAR
  > SAR ∩ TAR

- the mean component $\boldsymbol{q}_n$ is either of

  > $\boldsymbol{t}_n + \boldsymbol{s}_n$ where $\boldsymbol{t}_n$ is smoothly varying trend and $\boldsymbol{s}_n$ is the seasonal component
  > polynomial trend

- the observation error component $\boldsymbol{r}_n$ is either of

  > a white noise sequence
  > none

The specification of the mean component and the error component could be channel wise. We call (30) the E-MARTS model, 'E' stands for 'Extended'.

3 Experiments and Physical Interpretation

3.1 Simulation

Given the Cholesky decomposition, or the square root, L of the matrix Σ

$$(31) \qquad \Sigma = LL^T, \quad L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{K1} & l_{K2} & \cdots & l_{KK} \end{pmatrix},$$

$\{\varepsilon_t\}$ can be expressed in the form:

$$(32) \qquad \varepsilon_t = Lw_t \quad w_t \sim N(0, I).$$

Using the matrix L, AR model (5) can be expressed in the form:

$$(33) \qquad x_{it} = \sum_{j=1}^{K} \sum_{m=1}^{M} A_{ijm} x_{j(t-m)} + \sum_{j=1}^{i-1} l_{ij} w_{jt} + l_{ii} w_{it} \quad (i = 1, \ldots, K).$$

This expression is useful for the simulation and parameter estimation.

3.2 Power Building Profile

The stability of AR model can be judged by observing the locations of the roots of the characteristic equation:

$$(34) \qquad \det\left[I - \sum_{m=1}^{M} A_m z^m \right] = 0.$$

If all roots are out of the unit circle then the AR model is stationary. Another way of judging the stationarity is to compute power building profile

$$(35) \qquad \{P_{kk}^B(t); k = 1, 2, \ldots, K \quad t = 0, 1, 2 \ldots\}$$

by the recursion:

$$(36) \qquad P^B(t) = \Psi P^B(t-1)\Psi^T + \Gamma \Sigma \Gamma^T \qquad (t = 1, 2, \ldots),$$

where Ψ and Γ are those of eq. (17) with the polynomial order $P = -1$ and

(37)
$$P^B(0) = \begin{pmatrix} O & \cdots & O \\ \vdots & & \vdots \\ O & \cdots & O \end{pmatrix}.$$

The system is stationary iff

(38)
$$P_{kk}^B(\infty) = \lim_{t \to \infty} P_{kk}^B(t) < \infty$$

for all k.

3.3   Power Spectrum and Coherency

For a stationary AR model (5) the frequency response from $\{\varepsilon\}$ to $\{x\}$ is given by

(39)
$$F(f) = [I - \sum_{m=1}^{M} A_m e^{-i2\pi \Delta t f m}]^{-1} \quad (0 \leq f \leq 1/2\Delta t)$$

then, the power and cross spectrum density matrix of $\{x_t\}$ is given by

(40)
$$P(f) = \Delta t F(f) \Sigma F^*(f) \quad (0 \leq f \leq 1/2\Delta t),$$

where Δt is the sampling interval and '*' denotes the conjugate transpose. If $P_{kk}^B(\infty) < \infty$ for all $k \leq K$,

(41)
$$P_{kk}^B(\infty) = \int P_{kk}(f) df = \text{Var}\{x_{kt}\}$$

holds. A natural *scaling factor* for each variable is defined by

(42)
$$\sigma_k = \sqrt{P_{kk}^B(\infty)} \quad (k = 1, 2, \ldots, K)$$

The (simple) coherency between two variables at frequency f is defined by

(43)
$$\gamma^2(f) = \frac{|P_{jk}(f)|^2}{P_{jj}(f) P_{kk}(f)}$$

This quantity is the square of correlation coefficient between two components at frequency f.

3.4   Relative Power Contribution

Assume that Σ has the block diagonal structure

(44)
$$\begin{pmatrix} \Sigma_{11}^b & 0 & \cdots & 0 \\ 0 & \Sigma_{22}^b & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Sigma_{rr}^b \end{pmatrix},$$

specified by the parameter

(45)
$$\{b(1), b(2), \ldots, b(r)\},$$

and

(46)
$$\Sigma_{jk}^b = \begin{pmatrix} \sigma_{b(j-1)+1,b(k-1)+1}, & \cdots, & \sigma_{b(j-1)+1,b(k)} \\ \vdots & & \vdots \\ \sigma_{b(j),b(k-1)+1}, & \cdots, & \sigma_{b(j),b(k)} \end{pmatrix} \quad (j = 1, 2, \ldots, r).$$

Introduce the same partition to $P(f)$ and $F(f)$:

$$P(f) = \begin{pmatrix} P_{11}^b(f) & P_{12}^b(f) & \cdots & P_{1r}^b(f) \\ P_{21}^b(f) & P_{22}^b(f) & \ddots & \vdots \\ \vdots & \ddots & \ddots & P_{(r-1)r}^b(f) \\ P_{r1}^b(f) & \cdots & P_{r(r-1)}^b(f) & P_{rr}^b(f) \end{pmatrix}$$

$$F(f) = \begin{pmatrix} F_{11}^b(f) & F_{12}^b(f) & \cdots & & F_{1r}^b(f) \\ F_{21}^b(f) & F_{22}^b(f) & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & F_{(r-1)r}^b(f) \\ F_{r1}^b(f) & \cdots & & F_{r(r-1)}^b(f) & F_{rr}^b(f) \end{pmatrix}.$$

then we have the expression

$$(47) \qquad P_{pp}^b(f) = \Delta t \sum_{q=1}^{r} F_{pq}^b(f) \Sigma_{qq}^b F_{pq}^{b*}(f) .$$

This means that power spectrum of the k-th variable is a sum of positive terms:

$$(48) \qquad \begin{aligned} P_{kk}(f) \;=\; & \text{the part derived from } \Sigma_{11}^b \\ & + \text{ the part derived from } \Sigma_{22}^b \\ & + \cdots + \text{the part derived from } \Sigma_{rr}^b. \end{aligned}$$

The *relative power contribution* from the p-th block to $P_{kk}(f)$ is defined by

$$(49) \qquad \mathrm{RPC}_{kp}(f) \equiv \frac{\text{the part derived from } \Sigma_{pp}^b}{P_{kk}(f)}.$$

3.5 Impulse Response Function and Frequency Response Function

From (10),

$$(50) \qquad u_{kt} = \begin{cases} 0 & (t < 0) \\ 1 & (t = 0) \\ 0 & (t > 0) \end{cases} \iff \varepsilon_{kt} = \begin{cases} 0 & (t < 0) \\ -A_{kkt} & (0 \le t) \end{cases}$$

where, we assume
$$(51) \qquad A_{kk0} = -1.$$

Thus the response of the system to the impulse input to k-th variable is calculated from (4) by:

$$(52) \qquad R_{jkt} = \begin{cases} 0 & (t < 0,) \\ \displaystyle\sum_{m=1}^{M}\sum_{i=1}^{K} A_{jim} R_{ik(t-m)} - A_{kkt} & (0 \le t, j = k) \\ \displaystyle\sum_{m=1}^{M}\sum_{i=1}^{K} A_{jim} R_{ik(t-m)} & (0 \le t, j \ne k) \end{cases}.$$

For the comparison of impulse responses, the normalized response defined by

$$(53) \qquad \tilde{R}_{jkt} = \frac{\sigma_k}{\sigma_j} R_{jkt}$$

is useful. σ_k and σ_j are scaling factors defined by eq.(42). Note that $\{\tilde{R}_{jkt}\}$ is a dimensionless quantity.

The response of the system(1) to the input

$$(54) \qquad u_{kt} = e^{-i2\pi\Delta t f t}$$

is given by

$$(55) \qquad F_{jk}(f)e^{-i2\pi\Delta t f t},$$

where

$$(56) \qquad F_{jk}(f) = \sum_{t=0}^{\infty} R_{jk}(t)e^{-i2\pi\Delta t f t}. \quad (0 \le f \le 1/2\Delta t)$$

F_{jk} is called the frequency response from the k-th to the j-th variable. The normalized dimensionless response is defined by

$$(57) \qquad \tilde{F}_{jk}(f) = \frac{\sigma_q}{\sigma_j} F_{jk}(f).$$

3.6 Step Response Function

The step input are generated from (10) by

$$
(58) \qquad u_{kt} = \left\{ \begin{array}{ll} 0 & (t < 0) \\ 1 & (t \geq 0) \end{array} \right. \iff \varepsilon_{kt} = \left\{ \begin{array}{ll} 0 & (t < 0) \\ -\displaystyle\sum_{m=0}^{t} A_{kkm} & (0 \leq t) \end{array} \right. .
$$

The response of the system to the step input to k-th variable and its normalized value are given by

$$
(59) \qquad S_{jkt} = \left\{ \begin{array}{ll} 0 & (t < 0,) \\ \displaystyle\sum_{m=1}^{M}\sum_{i=1}^{K} A_{jim}S_{ik(t-m)} - \displaystyle\sum_{m=0}^{t} A_{kkm} & (0 \leq t, j = k) \\ \displaystyle\sum_{m=1}^{M}\sum_{i=1}^{K} A_{jim}S_{ik(t-m)} & (0 \leq t, j \neq k) \end{array} \right. ,
$$

and

$$
(60) \qquad \tilde{S}_{jkt} = \frac{\sigma_k}{\sigma_j} S_{jkt},
$$

respectively.

3.7 Interactive System Analysis

Above mentioned methods can be regarded as basically the same thing differ only in the choice of the form of the input. They are summarized in Table 1.

Table 1. Test inputs to a linear system

method	input u	to see ...
simulation	random noise(2)	if the model reproduces realistic time series
impulse response	impulse function(50)	time domain feature
step response	step function(58)	time domain feature
frequency response	trigonometric function(54)	frequency domain feature

The operation to mask the matrix $A(= [A_{ij}])$ with mask $C(= [c_{ij}])$ is defined by

$$
(61) \qquad C \cap A = [c_{ij}A_{ij}].
$$

When the coefficient matrices are masked by one and the same mask we call the model thus obtained a *masked AR model*. All the masks we use are defined with $c_{ij} = 0$ or 1. Application of the methods in Table 1 to masked AR models is sometimes quite useful for the understanding of the physical nature of a system.

[**Note**] Suitable form of graphical outputs are indispensable for correct understanding of the system. There, the scaling factor (42) should be fixed at that derived from the original not-masked AR model.

3.8 Causality Analysis

Eq.(14) implies that the existence/nonexistence of the causality relation between variables are equivalent to nonzero/zero of off diagonal elements of coefficient matrices. Therefore, we can check the existence/nonexistence of causality relations by evaluating the goodness of fit of SAR models. Once the causality relation, for example, from j-th to i-th variable is detected, we can find the nature of the relation by watching what happens when this relation is eliminated.

Feedback characteristic matrix with respect to variable k, C^k, is defined as the matrix whose (i, j)-th element is given by $C_{ij}^k(\infty)$, where

$$(62) \qquad C_{ij}^k(t) = \frac{P_{kk}^B(t; \text{AR model masked by} N^{ij})}{P_{kk}^B(t; \text{original AR model})},$$

where N^{ij} is defined as the mask such that the i, j-th component is 0 and other components are 1. If $C_{ij}^k(\infty) > (<)1$, the power of the k-th variable increases (decreases) when the path from j-th to i-th variable is cut.

Note that there can be cases where $P(f; \text{AR model masked by} N^{ij}) \neq P(f; \text{original model})$ and $C_{ij}^k(\infty) = 1$.

4 Model Fitting and Information Criterion

Assume that we have K channel time series data of length N

$$(63) \qquad \{x_{kt} | k = 1, 2, \ldots, K, t = 1, 2, \ldots, N\}.$$

4.1 Sequential Least Squares Procedure

Log likelihood of the k-th model of eq.(33), when $\{w_{jt} : j = 1, 2, \ldots, k-1\}$ are known and initial values $\{x_{kt} | k = 1, 2, \ldots, K, t = 1, 2, \ldots, N_0\}$ are fixed is given by

$$(64) \qquad -\frac{N_1}{2} \log 2\pi - \frac{N_1}{2} \log l_{kk}^2 - \frac{1}{2l_{kk}^2} \sum_{t=N_0+1}^{N} (x_{it} - \sum_{j=1}^{K} \sum_{m=1}^{M} A_{ijm} x_{j(t-m)} + \sum_{j=1}^{k-1} l_{ij} w_{jt})^2.$$

Let $I = N_0 + 1$ and define $\boldsymbol{a}_k$ by

$$(65) \qquad \boldsymbol{a}_k = (A_{k11}, A_{k21}, \cdots, A_{kK1}, A_{k12}, \cdots, A_{kK2}, \cdots, A_{k1M}, \cdots, A_{kKM})^T$$

then its maximum log likelihood estimate is obtained by the minimization of

$$(66) \qquad \sum_{t=I}^{N} \left(x_{it} - \sum_{j=1}^{K} \sum_{m=1}^{M} A_{ijm} x_{j(t-m)} + \sum_{j=1}^{k-1} l_{ij} w_{jt} \right)^2$$

$$\equiv \quad ||\boldsymbol{x}_k - (X, W_{k-1}) \begin{pmatrix} \boldsymbol{a}_k \\ l_{k1} \\ \vdots \\ l_{k,k-1} \end{pmatrix} ||^2.$$

where

$$(X, W_{k-1}, \boldsymbol{x}_k) \equiv \begin{pmatrix} x_{1,I-1}, & x_{1,I-2}, & \cdots, & x_{K,I-M}, & w_{1,I} & \cdots, & w_{(k-1),I} & x_{k,I} \\ x_{1,I}, & x_{1,I-1}, & & x_{K,I-M+1}, & w_{1,I+1} & & w_{(k-1),I+1} & x_{k,I+1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ x_{1,N-1}, & x_{1,N-2}, & & x_{K,N-M}, & w_{1,N} & \cdots, & w_{(k-1),N} & x_{k,N} \end{pmatrix}.$$

(67)

The estimate of l_{kk} is given by

$$(68) \qquad \hat{l}_{kk}^2 = \frac{1}{N_1} \{\text{Minimum value of eq.(66)}\},$$

where $N_1 = N - N_0$ and the AIC of the model is given by

$$(69) \qquad \text{AIC}_M^k = N_1 \log 2\pi + N_1 \log \hat{l}_{kk}^2 + N_1 + 2(MK + k).$$

W_k is prepared by calculating

$$(70) \qquad \boldsymbol{w}_k = \frac{1}{l_{kk}} \left[\boldsymbol{x}_k - (X, W_{k-1}) \begin{pmatrix} \boldsymbol{a}_k \\ l_{k1} \\ \vdots \\ l_{k,k-1} \end{pmatrix} \right],$$

and then we can repeat the minimization of (66) and computation of (70) for increased k until k equals to K.

When this iterative procedure is finished, AIC of the whole model is given by

$$(71) \qquad \text{AIC}_M = \sum_{k=1}^{K} \text{AIC}_M^k.$$

The relation between AR model parameter(8) and the estimate obtained by this sequential method is

$$(72) \qquad \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_K \\ l_{11} & l_{21} & \cdots & l_{K1} \\ & l_{22} & & l_{K2} \\ & & \ddots & \vdots \\ & & & l_{KK} \end{pmatrix} \equiv \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_M^T \\ L^T \end{pmatrix} = (A_1, A_2, \ldots, A_M, L)^T.$$

[SAR model]   It is easy to modify this procedure to fit SAR model. The only changes are to replace $\sum_{j=1}^{K}$ in eq. (64) by $\sum_{\{\forall j | S_{kj}=1\}}$ and MK in (69 by $\sum_{j=1}^{K} S_{kj}$.

[PAR model]   To fit PAR model, use

$$\| \boldsymbol{x}_i - (B, X, W_{k-1}) \begin{pmatrix} b_{k0} \\ b_{k1} \\ \vdots \\ b_{kP} \\ \boldsymbol{a}_k \\ l_{kl} \\ \vdots \\ l_{k,k-1} \end{pmatrix} \|^2$$

instead of (66) where B is a $N_1 \times (P+1)$ matrix defined by

$$B = \begin{pmatrix} 1, & I, & \ldots, & I^P \\ 1, & I+1, & \ldots, & (I+1)^P \\ \vdots, & & & \\ 1, & N, & \ldots, & N^P \end{pmatrix}.$$

Then the modified sequential method returns

$$(73) \qquad (\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_P, A_1, A_2, \ldots, A_M, L)^T$$

instead of (72).

[Missing Observation]   One not quite efficient but easy way of fitting AR model to a set of data where some observations are missing is to apply above procedure after removing all rows containing *missing values* from the matrix (67). At least this gives a good initial guess for the exact method explained in Appendix C.

4.2 Modeling of Variance-covariance Matrix

The log likelihood of the model (5,6) is

$$
(74) \quad \ell(\boldsymbol{\theta}) = \sum_{t=N_0+1}^{N} \log \phi_K(\boldsymbol{x}_t | \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m}, \Sigma)
$$

$$
= -\frac{N_1 K}{2} \log 2\pi - \frac{N_1 K}{2} \log \det \Sigma
$$

$$
-\frac{1}{2} \sum_{t=N_0+1}^{N} (\boldsymbol{x}_t - \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m})^T \Sigma^{-1} (\boldsymbol{x}_t - \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m})
$$

If $\{A_m\}$ is fixed, the prediction error, or the residual sequence, and the maximum log likelihood estimate of Σ are respectively given by

$$
(75) \quad \boldsymbol{r}_t = \boldsymbol{x}_t - \sum_{m=1}^{M} A_m \boldsymbol{x}_{t-m} \qquad (t = N_0+1, N_0+2, \ldots, N)
$$

and

$$
(76) \quad \hat{\Sigma} = \frac{1}{N_1} \sum_{t=N_0+1}^{N} \boldsymbol{r}_t \boldsymbol{r}_t^T.
$$

If the block diagonal structure(44) is assumed, the MLE of Σ_{jj}^b is given by

$$
(77) \quad \tilde{\Sigma}_{jj}^b = \frac{1}{N_1} \sum_{t=N_0+1}^{N} (r_{b(j-1)+1,t}, \ldots, r_{b(j),t})^T (r_{b(j-1)+1,t}, \ldots, r_{b(j),t})
$$

and consequently

$$
(78) \quad \text{AIC}\{b(1), b(2), \ldots, b(r)\}
$$

$$
= K N_1 \log 2\pi + N_1 \sum_{j=1}^{r} \log \det \tilde{\Sigma}_{jj}^b + N_1 K
$$

$$
+ 2MK^2 + \sum_{j=1}^{r} (b(j) - b(j-1) + 1)(b(j) - b(j-1)).
$$

When $r = 1$ and $b(1) = K$, this AIC is equal to AIC_M (71).

4.3 Use of the Kalman Filter

If a state space model is defined by (21) or (17) the distribution of $\boldsymbol{Z}_n = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$ and $\boldsymbol{Y}_n = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ given $\boldsymbol{Z}_n$ are

$$
\pi(\boldsymbol{Z}_n | \boldsymbol{\theta}) = \phi_p(\boldsymbol{z}_0 | \mu_0, \Sigma_0) \prod_{n=1}^{N} \phi_p(\boldsymbol{z}_n | \Psi \boldsymbol{z}_{n-1}, \Gamma \Sigma_V \Gamma^T)
$$

$$
f(\boldsymbol{Y}_n | \boldsymbol{Z}_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \phi_K(\boldsymbol{y}_n | \Phi \boldsymbol{z}_n, \Sigma_R).
$$

Here, $\phi_p(\cdot | \boldsymbol{\mu}, \Sigma)$ denotes the density function of the p-dimensional normal distribution $N(\boldsymbol{\mu}, \Sigma)$, and $\Sigma_V = E\{\boldsymbol{v}_n \boldsymbol{v}_n^T\}$. The dimension of the state vector $\boldsymbol{z}_n$ is denoted by p. From the marginal distribution of $\boldsymbol{Y}_n$

$$
f(\boldsymbol{Y}_n | \boldsymbol{\theta}) = \int f(\boldsymbol{Y}_n | \boldsymbol{Z}_n, \boldsymbol{\theta}) \pi(\boldsymbol{Z}_n | \boldsymbol{\theta}) d\boldsymbol{Z}_n,
$$

defining $f(\boldsymbol{y}_n | \boldsymbol{Y}_{n-1}, \boldsymbol{\theta}) = f(\boldsymbol{Y}_n | \boldsymbol{\theta}) / f(\boldsymbol{Y}_{n-1} | \boldsymbol{\theta})$, we get

$$
f(\boldsymbol{Y}_N | \boldsymbol{\theta}) = \prod_{n=1}^{N} f(\boldsymbol{y}_n | \boldsymbol{Y}_{n-1}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \phi_K(\boldsymbol{y}_n | \Psi \boldsymbol{z}_{n|n-1}, \Phi V_{n|n-1} \Phi^T + \Sigma_R),
$$

where $\boldsymbol{z}_{n|n-1}$ and $V_{n|n-1}$ are efficiently computed by Kalman filtering technique (cf. Appendix C). The log likelihood of the model is calculated by

$$\ell(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \phi_K(\boldsymbol{y}_n | \Psi \boldsymbol{z}_{n|n-1}, \Phi V_{n|n-1} \Phi^T + \Sigma_R).$$

If we assume initial data of length N_0 are fixed, the log likelihood is given by

$$(79) \qquad \ell(\boldsymbol{\theta}) = \sum_{n=N_0+1}^{N} \log \phi_K(\boldsymbol{y}_n | \Psi \boldsymbol{z}_{n|n-1}, \Phi V_{n|n-1} \Phi^T + \Sigma_R).$$

The maximum log likelihood estimate of parameter $\boldsymbol{\theta}$ is obtained numerically by maximizing $\ell(\boldsymbol{\theta})$.

5 Implementation

5.1 Basic Principles

5.1.1 Operating mode

ARdock is designed basically for interactive use, but its use in batch mode is also taken cared well. A smart way of using ARdock is to run ARdock in the batch mode for model fitting, and run it in the conversational mode for the system analysis.

5.1.2 Expected users

ARdock is designed anticipating three kind of users summarized in Table 2.

Table 2. Expected users of ARdock

category	description
end-user	those who are interested only in using ARdock in its distributed form
modifier	those who are interested in modifying ARdock or re-using subroutines of ARdock package for his/her own purpose
author	the present authors ourselves

We distribute the source code some part of which can be meaningful only for the present authors. This reduces readability of the code but it also reduces the liable discrepancy between the distributed version and the authors' private version.

5.1.3 Language

The code is written in FORTRAN77. There are parts coded so that parallel computation is enabled. Message Passing Interface(MPI) is employed for the message passing. C preprocessor directives are employed for user category dependent source code switching; file inclusion; and macro substitution.

5.1.4 Licencing policy

We make public the ARdock source code under the open market licence policy proposed by the Institute of Statistical Mathematics.

Open Market Licence for software(version:OML-SW-E-1996)

0. In the following, "Copyright_OML notice" means "the copyright notice with a statement saying that the said software is made public under this Open Market Licence for software".

1. On the condition that "Copyright_OML notice" and attached statements are copied as they are, all or any portion of the said software can be copied or redistributed.

2. Modification of the software is permitted, provided that the exact place of the modification, the date and the name of the modifier are shown conspicuously.

3. It is forbidden to apply for a patent on the software which utilizes the said software.

4. The said software is distributed without any warranty. The copyright holder takes no responsibility for any damage caused by the use of the said software.

5. Any interactively controlled application made utilizing the said software has to show "Copyright_OML notice" and attached statements conspicuously when it is started up.

For the healthy development of software science, suitable form of the licencing is important. We definitely would like to claim our copyright to our ARdock package. And we would like to use our copyright to let all concerned people use ARdock source code freely under two conditions that (1) our credit is properly referred and (2) statements attached to the copyright note saying that the original ARdock source code is distributed from the specified site without charge are kept as they are. We permit modification or redistribution of ARdock source code as far as these two conditions are met. The redistribution may be with or without charge.

It might seem we are abandoning our right. It is not so. We are using our right in our way. We do not mind if a person or organization who is informed that charge-free original ARdock source code is available decides to get possibly-slightly-modified-version of ARdock from other sources. In short, we guarantee that our charge-free original will be always in the open market. Anybody can sell anything there if he/she can find a buyer.

We find the policy advocated by the 'Open Source' school people is very close to ours.

[**Recommended form of reference/acknowledgment**]   When you are to publish software xxxxx, which utilizes some subroutines of ARdock-package, here is an example of making reference to ARdock.

xxxxx Copyright, 20yy uuuuu

xxxxx utilizes subroutines of ARdock package, with here and there modifications. The places of modifications are indicated in the code.

ARdock_package
Copyright_OML 1999 M. Ishiguro, H. Kato and H. Akaike
The source code of this package can be obtained from ISMLIB[1] of the Institute of Statistical Mathematics without any charge. On the conditions that terms of the OPEN MARKET LICENCE[2] for software are observed and that the copyright holder nor the Institute of Statistical Mathematics take no responsibility on any result from the use, programs in the package can be used or modified freely. On the condition that this note is attached as it is, this package can be redistributed.
(1) http://www.ism.ac.jp/software/ismlib/soft.e.html
(2) ftp://ftp.ism.ac.jp/pub/ISMLIB/OML/OML-SW-E-1996

5.2   Source Code

5.2.1   Distributed files

ARdock package summarized in Table 3 can be obtained from
   ftp://ftp.ism.ac.jp/pub/ISMLIB/ARDOCK/

The location is also reachable through
   http://www.ism.ac.jp/software/ismlib/soft.e.html

Table 3. ARdock package

file name	contents
README	Latest announcement
ardock.F	main program and subroutines
ardock.env	common setting
sample.dat	sample data
ardock.dvi	dvi file of this monograph
Makefile	A prototype 'makefile' for Unix environment
TUTORIAL.TXT	See Section 5.4.6

5.2.2 Design parameters

Table 4a. Parameters of ardock.F[1]

parameter	standard setting	meaning
MAXN	2000	maximum data length
MAXCH	10	maximum number of channels
MAXF	80	frequency domain grid count
MAXLAG	20	maximum AR model order
MAXPAR	200	maximum number of free parameters of a model

(1) First fifteen lines of ardock.F

```
#define ARDOCKLINK
c#define MPI
c#define MODIFIER "MODIFIER"

#define MAXN 2000
#define MAXCH 10
#define MAXF 80
#define MAXLAG 30
#define MAXPAR 200

#define IOUNIT 18
#define KEEPUNIT 99
#define TUTUNIT 98
#define BUGLOG 91
#define BUGMAP 92
```

Table 4b. Parameters in ardock.env[1]

parameter	standard setting	meaning
MPIPROCS	1[2]	number of parallel processors
NWORK	1000000	size of real type work area
IWORK	10000	size of integer type work area

(1) ardock.env

```
parameter (mpiprocs = 1)
dimension mPIL ( MPIPROCS ) , mPIB ( MPIPROCS )
common  /commpi/ iNFO , mYID , nPROCS , mPI , mPIL , mPIB
parameter (nwork=1000000,iwork=10000)
eal*8 wK ( NWORK )
integer iWK ( IWORK )
common  /comwork/  wK , iWK , limflag
```

Work areas necessary for the execution of subroutines are taken in the common work areas
wk and iwk.

(2) If user's computer is a parallel machine, set this value appropriately.

Table 4c. I/O unit number assignment

parameter	standard setting	related file(s)
IOUNIT	18	data file, AR file etc.
KEEPUNIT	99	@KEEP.KEY
TUTUNIT	98	TUTORIAL.TXT
BUGMAP	92	bug.map
BUGLOG	91	bug.log

Table 4d. user dependent parameter setting

parameter	end-user	modifier	author
ARDOCKLINK	define[1]	define[2]	define[2]
MPI	(3)	(3)	define[3]
MODIFIER	—	"MODIFIER"[4]	"MODIFIER"
AUTHOR	—	—	define[5]

(1) The first line of the distributed file is '#define ARDOCKLINK'. End-users should not erase this line.

(2) Those parts marked by #ifdef ARDOCKLINK — #endif are to customize subroutines for ARdock. To reuse those subroutines un-define the parameter ARDOCKLINK.

(3) The second line of the distributed file is 'c#define MPI'. If user's computer is a parallel machine, activate this line as a directive by removing first 'c'. Those parts marked by #ifdef MPI— #endif are for parallel computation.

(4) The third line of the distributed file is 'c#define MODIFIER "MODIFIER"'. Activate this line as a directive for C pre-processor by removing first 'c'. Those parts marked by #ifdef MODIDIER — #endif are parts for modifiers. End-users have nothing to do with those parts.

(5) Those parts marked by #ifdef AUTHOR — #endif are parts for the authors' private version. We are not obliged to give any explanation concerning the parts for 'author'.

5.2.3 Compilation

The object file is obtained by the following procedure:

1. If necessary set appropriate values to parameters summarized in Tables 4a, 4b and 4c.

2. Define appropriate parameters for C pre-processor summarized in Table 4d.

3. Apply C pre-processor.

4. Compile by a FORTRAN compiler

So far, the test of ARdock is successfully done in environments summarized in the following Table.

Table 5. Computation Environments

Hardware	Operating System	Compiler
HITACHI SR8000	HI-UX/MPP	FORTRAN90
HP9000V2250	HP-UX 11.0	HP Fortran 90
IBM R/6000 SP	AIX Version 4	AIX XL Fortran
SGI Origin 2000/4-DS-I	IRIX Release 6.4 IP27	FORTRAN90
SPARCstation 5	Sun OS 4.1.4-JLE 1.1.4	g77 version0.5.23
PC-9801 nx/C	MS-DOS 5.00A-H	Microsoft FORTRAN Version 4.01

5.3 Subroutines

5.3.1 Principal variables

Table 6a. Principal Variables

variable	mathematical expression	comment
Ndata	N	length of data
Kch	K	number of channels
Lag	M	AR model order
Ar	$\{A_1, A_2, \ldots, A_M\}$	AR coefficient matrix
Var	$\Sigma = LL^T$	variance covariance matrix
Parl	$(\boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_P, A_1, A_2, \ldots, A_M, L)^T$	eq.(73)
Power	$P(f)$ and RPC	power spectrum (40) and RPC(49)
Pbp	$P^B(t)$	Power building profile (36)
Mask	C	in eq.(61)
El	ℓ	log likelihood
Ib	b	block structure parameter (45)
Nblock	r	number of blocks
Scale	σ_k	scaling factor (42)
Tscale	Δt	sampling interval (see Table 3)
Sqrvar	L	eq.(31)
Hh	X	X in eq.(86)
dATA	x	data matrix (see Table 3)
lAGOPT	M	AR model order
lAGMAX	N_0	initial data length in (64)
iNDEXM	S	SAR model index matrix in eq.(23)
mEAN	P+1	0: mean component = trend + seasonal ?0: mean component = polynomial of order P
nSYS	K_y	dimension of controlled sub-system(28)
lISTPE	$(L_1, L_2, \ldots, L_K)$	list of periodicity in (20)

Table 6b. Important Control Variables

variable	comment
iNFO	info level. See Section 6.2.
mYID	processor-id of parallel computer.
nPROCS	number of parallel processors.
mPI	parallel processing flag. 0: no 1: yes

5.3.2 Subroutine coding

[Typical example]

```
      subroutine sadjust (
    i Abuse , Nch , Iauto , Lpe , Ndata , Lagmax ,            .......(0)(1)
    a Y , Trend , Season ,                                   .......(0)(1)
    o Trinit , Vtrend , Vseas )                              .......(0)(1)
      implicit real*8 ( a-h , o-z )
#include "ardock.env"                                        ..........(2)
      dimension Y ( Ndata ) , Trend ( Ndata ) , Season ( Ndata ) ,
    * Trinit (2)
      character*1  io , stio
      data inisub /0/
c     ...................................................
c     sadjust                                                ..........(3)
c     Copyright_OML 1999 M. Ishiguro & H. Kato
c      The source code of this subroutine, as a part of the ARdock package,
c      can be obtained from ISMLIB(ftp://ftp.ism.ac.jp/pub/ISMLIB/)
c      of the Institutec of Statistical Mathematics without any charge.
c      On the conditions that terms of the OPEN MARKET LICENCE for software
c      (ftp://ftp.ism.ac.jp/pub/ISMLIB/OML/OML-SW-E-1996) are observed and
c      that the copyright holder nor the Institue of Statistical Mathematics
```

```
c          take no responsibility on any result from the use, this subroutine
c           can be used or modified freely. On the condition that this note is
c           attached as it is, this subroutine can be redistributed.
           io = 'I'                                                    .........(4)
 9100   continue                                                       .........(5)
           if( io .eq. 'O') then
              nw0  =  nw ('}' , nw0)                                    .........(6)
              iw0  =  iw ('}' , iw0)
           end if
#ifdef MODIFIER                                                        .........(7)
           myself = 9                                                   .........(8)
           if( iNFO .ge. myself .and. inisub .eq. 0) then
              call selfcom ('sadjust', io , stio )                      ........(9)
              call showm3 ('I',
      i     '1.Adjusts one dimensional time series data for trend',
      i     '  and seasonality by BAYSEA procedure (Appendix B). ',
      i     '2.Reserves work area and calls sadjuw.            ')
              call showr ('I','Abuse' ,  Abuse ,   1 , 1 , 1)
              call showi ('I','Nch' ,    Nch ,     1 , 1 , 1)
              call showi ('I','Iauto' ,  Iauto ,   1 , 1 , 1)
              call showi ('I','Lpe' ,    Lpe ,     1 , 1 , 1)
              call showi ('I','Ndata' ,  Ndata ,   1 , 1 , 1)
              call showi ('I','Lagmax' , Lagmax ,  1 , 1 , 1)
              call showr ('I','Y' ,      Y ,       1 , 1 , Ndata )
              call showr ('O','Trinit' , Trinit ,  1 , 1 , 2)
              call showr ('O','Vtrend' , Vtrend ,  1 , 1 , 1)
              call showr ('O','Vseas' ,  Vseas   , 1 , 1 , 1)
              call selfend ( myself , io , inisub )
           end if
#endif
           if( io .ne. 'O') then
catch-        call nw
              nw0  =  nw ('{' , 0)                                      ...........(6)
              iw0  =  iw ('{' , 0)
              goto 9200                                                 ...........(5)
           end if
c           ------
           return                                                       ...........(4)
c           ======
 9200   continue                                                        ...........(5)
           io = 'O'                                                     ...........(4)
           ia  =  Ndata
           lenh   =  Ndata * (4 + 1)                                    ...........(10)
           lenh2  =  Ndata * 2 + 1
           mdc0   =  Ndata * 2 * (4 * 2 + 1)
           call sadjuw (                                                ...........(11)
      i Abuse , Nch , Iauto , Lpe , Ndata , Lagmax , mdc0 ,
      i lenh , lenh2 ,
      a Y , Trend , Season ,
      o Trinit , Vtrend , Vseas ,
      w wk ( nw ('ftrn' , ia )) , wk ( nw ('fsea' , ia )) ,            ...........(6)
      w wk ( nw ('psds' , ia )) ,
      w wk ( nw ('psdt' , ia )) ,
      w wk ( nw ('irreg' , ia )) ,
      w wk ( nw ('f' , ia )) ,
      w wk ( nw ('h2' , lenh2)) , wk ( nw ('dc' , mdc0)) ,
      w wk ( nw ('h' , lenh )))
           goto 9100                                                    ...........(5)
        end
```

(0) arguments: Usually, arguments are arranged in order of 'those whose values are not changed in the routine' then those whose values are set in the subroutine'. Variables which have both characters come between. Within each category character variables come first, then real variables and finally integer variable.

(1) continuation mark: In 'subroutine' and 'call' sentences, arguments used as input only are marked by 'i' in the 6-th column, those for output are by 'o'. In some routine these role are reversed depending on some control input. In that case marks 'I' and 'O' are used. Those variables used for input and output at the same time are marked with 'a'. Variables used as work area in the subroutine are marked with 'w'.

(2) See Section 5.2.2

(3) Copyright note

(4) the execution of this routine is controled by the variable 'io'.

(5) label:

- − 9100: exit
- − 9199: emergency stop or exit
- − 9200: main part. No 'return' nor 'stop' sentences will be used here. 'goto 9100' or go to '9199 is used.
- − 7xxx: No label headed by '7' will be used by the original author. They are for modifiers' use.

(6) Work space management: Work space for this job is kept, used and released.

(7) See Section 5.2.2, especially Table 4d.

(8) myself:

- − ≥ iNFO: Print outs 'self comment' once at their first calls. See Section 6.2.
- − = 0: Print outs 'self comment' every time.
- − = −9: Print outs 'self comment', executes this routine once and stops.

(9) *self comment function*: 'Self comment function' is provided. According to user's choice, every subroutine prints out their own name and values of given arguments etc. Equation or Section numbers referred to is of this monograph.

(10) variables:

- − global constant: Capitalized.
- − variables in common area: Lower case initial followed by capital letters.
- − subroutine argument: Initial letter is capitalized.
- − local variable: lower case only.

(11) Actual computation is done by the subroutine 'sadjuw'.

(*) spaces:

- − Variables are insulated from other elements including delimiters like '(', ',' etc. by spaces. This convention will help easy searches for variables.
- − In every 'subroutine sentence', the space after the word 'subroutine' will be single.
- − In every 'calling sentence', the space after the word 'call' will be single.

(*) comment: Comments in the source code are minimal. Efforts will be continued to improve the readability of the code.

(*) if-block: Indented. The use of too big if-clause is avoided so that the structure of routine is easy to see.

(*) do-loop: Indented.

(*) Parallel processing: Some subroutines are coded for parallel processing. To make these subroutines 'portable', a flag system is devised. If there exists two subroutines for parallel processing, and one calls the other, then the inside routine runs sequentially. Study use of mpion and mpioff and the flag mPI in the common 'commpi' in 'ardock.env' file.

[Another example]

```
      subroutine jdate (                          . . . . . . . . . . . . . .(1)
     i Ymd2date ,
     I Ymd , Time ,
     O Date , Atime ,
     o Dayoweek )
      implicit integer * 4 ( a - z )
#include "ardock.env"
      character*(*) Ymd2date
      real * 8 Atime , ttime
      dimension mon (13)
      character*1  io , stio , viv , vov
```

```fortran
          data mon  / 0,31,61,92,122,153,184,214,245,275,306,337,366 /
          data inisub /0/
c         ..................................................
c         jdate
c         Copyright_OML 1999 M. Ishiguro & H. Kato
c          The source code of this subroutine, as a part of the ARdock package,
c           can be obtained from ISMLIB(ftp://ftp.ism.ac.jp/pub/ISMLIB/)
c           of the Institute of Statistical Mathematics without any charge.
c           On the conditions that terms of the OPEN MARKET LICENCE for software
c           (ftp://ftp.ism.ac.jp/pub/ISMLIB/OML/OML-SW-E-1996) are observed and
c           that the copyright holder nor the Institute of Statistical Mathematics
c           take no responsibility on any result from the use, this subroutine
c           can be used or modified freely. On the condition that this note is
c           attached as it is, this subroutine can be redistributed.
          io = 'I'
 9100     continue
          if( io .eq. 'O') then
              Dayoweek  =  mod ( Date + 2 , 7)
              if( Dayoweek .lt. 0) Dayoweek = Dayoweek + 7
          end if
#ifdef MODIFIER
          myself = 9
          if( iNFO .ge. myself .and. inisub .eq. 0) then
              call selfcom ('jdate', io , stio )
              call showm2 ('I',
     i        ' Converts calendar data to the day count from',
     i        ' Jun. 1. 1601.AD., and vice versa.')
              call showc ('I','Ymd2date' , Ymd2date ,   1 , 1 , 1)
              if( Ymd2date .eq. '-->') then
                  viv = 'I'
                  vov = 'O'
              else
                  viv = 'O'
                  vov = 'I'
              end if
              call showi (viv,'Ymd' ,        Ymd ,        1 , 1 , 1)
              call showi (viv,'Time' ,       Time ,       1 , 1 , 1)
              call showi (vov,'Date' ,       Date ,       1 , 1 , 1)
              call showr (vov,'Atime' ,      Atime ,      1 , 1 , 1)
              call showi ('O','Dayoweek' ,   Dayoweek ,   1 , 1 , 1)
              call selfend ( myself , io , inisub )
          end if
#endif
          if( io .ne. 'O') then
              goto 9200
          end if
c         ------
          return
c         ======
 9200     continue
          io = 'O'
          if( Ymd2date .ne. '-->') goto 9201
c         --------------------------------------
          year   =  Ymd / 10000
          day   =  Ymd - year * 10000
          month  =  day / 100
          day   =  day - month * 100
          hour   =  Time / 10000
          second  =  Time - hour * 10000
          minute  =  second / 100
          second  =  second - minute * 100
          Atime  =  hour + minute / 60.d0 + second / 3600.d0
          Atime  =  Atime  / 24.d0
          nm  =  month - 2
          ny  =  year - 1600
          if( nm .le. 0) then
              nm  =  nm + 12
              ny  =  ny - 1
          end if
          nl  =  ny / 4
          nll  =  ny / 100
          nlll  =  ny / 400
          Date  =  ny * 365 + nl - nll + nlll + mon ( nm ) + day
          goto 9100
c         =========
 9201     continue
          jd = Date
          j400  =  ( jd - 1) / 146097
          jd  =  jd - j400 * 146097
          j100  =  ( jd - 1) / 36524
          if( j100 .eq. 4) j100 = 3
          jd  =  jd - j100 * 36524
          j4  =  ( jd - 1) / 1461
          jd  =  jd - j4 * 1461
          j1  =  ( jd - 1) / 365
```

```
              if( j1 .eq. 4) j1  =  3
              jd  =  jd - j1 * 365
              year  =  1600 + j1 + j4 * 4 + j100 * 100 + j400 * 400
              do 1 j = 1 , 12
                 if( jd .le. mon ( j + 1)) then
                    month = j + 2
                    day  =  jd - mon ( j )
                    if( month .gt. 12) then
                       month  =  month - 12
                       year  =  year + 1
                    end if
                    goto 9
c                   ======
                 end if
c                -------
      1          continue
      9       continue
              Ymd  =  year * 10000 + month * 100 + day
              ttime  =  Atime * 24.d0
              hour  =  ttime
              ttime  =  ttime - hour
              ttime  =  ttime * 60.d0
              minute  =  ttime
              ttime  =  ttime - minute
              second  =  ttime * 60.d0
              Time  =  hour * 10000 + minute * 100 + second
              goto 9100
           end
```

(1) See [Info level selection] in Section 6.2.

5.3.3   Annotated list of subroutines

Subroutines are classified into sub-packages shown in Table 7.

Table 7. sub packages

#	sub-package	type	contents
1	main-pack	A	easydall fitar getar getdata gui0 main
2	ardock-pack	A	ardock arsimu autodock click1 click2 cohgraph gui1 gui2 gui3 maskedar pbpgraph pwfile pwgraph subaic subwin1
3	datard-pack	R	clab cpar cset datard fparam gparam ipar iset jdate rpar rparam rset
4	arfit-pack	R	arfile arfilw arfit arfitw arorder build click3 indexset marfit sarfit sssolver subwin2
5	marts-pack	R	gmarts gmartw marts martspar martsw remtsbx subabh subqrs tarmodel
6	misc-pack	R	maskrs ready
7	pwspx-pack	R	pwspx subcoh subnosx
8	pwevol-pack	A	llset pwevol pwprof
9	lgraph-pack	R	gfile lgraph
10	linop-pack	R	cxhushld cxsolve fouger fouge1 hushld invmat posmat rn01 rnor solve
11	dall-pack	R	bill dall eval innerp lftint numdif
12	kalman-pack	R	both clr kalman kalmaw left right fmatrix
13	baysea-pack	R	decode exhsld exsolve init sadjust sadjuw setd setdc setx subsea
14	mpi-pack	R	mpiinit mpioff mpion mpisoff mpison mpisres mpistop
15	keysave-pack	R	jfrex keympi keysave markinp
16	areachk-pack	R	areachk iw nw
17	selfcom-pack	R	oml selfcom selfend showc showi showm showm2 showm3 showr showw them
18	bug-pack	R	bug bughtrap bugnet bugp bugvtrap debug sbuggle

N.B. A-type sub-packages are those specialized for ARdock. R-type sub-packages are re-usable ones.

  1. main-pack

        required sub-packages: ardock-pack areachk-pack arfit-pack baysea-pack bug-pack dall-
             pack datard-pack kalman-pack keysave-pack lgraph-pack linop-pack marts-pack misc-
             pack (mpi-pack) pwevol-pack pwspx-pack (selfcom-pack)

1.2 getar

- Reads AR model file to recover AR model parameters.
- See Section 5.4.2 for the proper form of the file.

1.3 getdata

- Reads time series data file. See Section 5.4.1 for the proper form of the file.

1.4 fitar

- Assembles and fits an E-MARTS model to get the AR component parameters.
- See Section 2.6.

1.5 easydall

- Reserves work area and calls subroutine dall for the log likelihood maximization.
- See Appendix D

1.6 gui0

- Prompts user for the choice between 'data' and 'model'.

2. ardock-pack

required sub-packages: areachk-pack arfit-pack baysea-pack bug-pack datard-pack kalman-pack keysave-pack lgraph-pack linop-pack misc-pack (mpi-pack) pwevol-pack pwspx-pack (selfcom-pack)

2.1 ardock

- Realizes interactive system analysis. See Section 6, especially Figure 1a.

2.2 arsimu

- Executes simulations by the basic model(1)

Cmd	input $\boldsymbol{u}_t$	output $\boldsymbol{x}_t$
i	impulse function(50)	impulse response(52)
s	step function(58)	step response(59)
f	trigonometric function (54)	frequency response(56)
n	simulated innovation(2,32)	simulated time series

2.3 maskedar

- Produces masked AR model(61)

2.4 autodock

- Produces *feedback characteristic matrix*(62) for given 'watching set' of variables.

2.5 subaic

- Evaluates the goodness of block diagonal structure of innovation matrix(44).
- d_AIC is the difference of AIC(78) and AIC(71) of the ordinary MAR model(4).

2.6 pbpgraph

- Displays power building profile (35) on the character display.

2.7 pwgraph

- Displays power spectrum (40) and relative power contribution (49) on the character display.

2.8 cohgraph

- Displays graph of coherency (43) on the character display.

2.9 pwfile

- Writes power spectrum and relative power contribution to a file (cf. Section 5.4.3).

2.10 gui1

- Interprets an interactive analysis command.

2.11 gui2

- Prompts the user to set block structure (45).

2.12 gui3

- Prompts the user to set frequency range of power spectrum(40), coherency (43) and frequency response (56).

2.13 click1

- Accepts an interactive analysis command.

2.14 click2

- Accepts a mask edit command.

2.15 subwin1

- Prompts the user for an interactive analysis command.

3. datard-pack

required sub-packages: areachk-pack keysave-pack misc-pack (mpi-pack) (selfcom-pack)

3.1 datard

- Reads time series data prepared in the proper form(see Section 5.4.1).

3.2 rparam

- Read/writes time series parameter set (cf. Section 5.4.1)

3.3 fparam

- Read/writes general parameter set.

3.4 gparam

- Read/writes an item of parameter set.

3.5 jdate

- Converts calendar data to the day count from Jun. 1. 1601.AD.

variable	type	meaning
Ymd	i	for example 19980714 for Jul. 14, 1998 AD
Time	i	133206 for 1 o'clock 32 minutes 6 second PM
Date	i	day count from Jan.1, 1601 AD
Atime	r	(hour + minute / 60 + second / 3600)/24
Dayoweek	i	Mon, Tue, ..., Sat, Sun/1, 2, ..., 6, 0

- The inverse transformation is possible.

3.6 cpar

- Recalls 'c' type time series parameter. (cf. See Section 5.4.1)

3.7 ipar

- Recalls 'i' type time series parameter. (cf. See Section 5.4.1)

3.8 rpar

- Recalls 'r' type time series parameter. (cf. See Section 5.4.1)

3.9 clab

- Recalls the label of a channel. (cf. See Section 5.4.1)

3.10 cset

- Sets 'c' type time series parameter. (cf. See Section 5.4.1)

3.11 iset

- Sets 'i' type time series parameter. (cf. See Section 5.4.1)

3.12 rset

- Sets 'r' type time series parameter. (cf. See Section 5.4.1)

4. arfit-pack

   required sub-packages: areachk-pack datard-pack keysave-pack linop-pack misc-pack (mpi-pack) (selfcom-pack)

   4.1 arfit

   - Reserves work area and calls arfitw

   4.2 arfitw

   - Fits MAR/SAR/PAR model (see Section 2.2, 2.4 and 4.1) to given data.

   4.3 marfit

   - Least squares method for MAR/PAR model(15) fitting.

   4.4 sarfit

   - SAR model fit and index matrix search(cf. eq.(23))

variable	meaning
aick	AIC_M^k (69)
aic	AIC_M (71)

   - The causality degree matrix is shown.

     [**Causality degree matrix**] Assume that the subset AR model specified by index matrix $\hat{S}$ is MAICE(Minimum AIC Estimate). Let $\hat{S}^{ij}$ be the index matrix identical to $\hat{S}$ with the only exception of $\hat{S}_{ij}^{ij} = 1 - \hat{S}_{ij}$, then we have

     $$\mathrm{AIC}(\hat{S}) < \mathrm{AIC}(\hat{S}^{ij}).$$

     The *causality degree matrix* D is defined as the matrix whose (i,j)-th element is defined by

     (80) $\qquad D_{ij} = (2 \times \hat{S}_{ij} - 1) \times (\mathrm{AIC}(\hat{S}^{ij}) - \mathrm{AIC}(\hat{S})).$

     $\hat{S}_{ij}$'s value is known from the sign of D_{ij}, and $|D_{ij}| = \mathrm{AIC}(\hat{S}^{ij}) - \mathrm{AIC}(\hat{S})$.

   4.5 build

   - Builds up linear equation matrix(67)

   4.6 indexset

   - Expands a row of the index matrix S for the construction of matrix (67) for SAR model fit.

   4.7 sssolver

   - Composes and solves subset regression equation(66) to get LSE of subset AR model (23) parameter. Only columns corresponding $S_{ij} = 1$ of eq.(67) are produced.

   4.8 click3

   - Edits the index matrix

   4.9 subwin2

   - Shows the index matrix

   4.10 arorder

   - MAICE of AR order.

   4.11 arfile

   - Calls arfilw

   4.12 arfilw

   - Read/writes AR model file

5. marts-pack

required sub-packages: areachk-pack arfit-pack datard-pack kalman-pack linop-pack (mpi-pack) (selfcom-pack)

5.1 marts

- Computes the log likelihood of E-MARTS model (30) for given parameter using state space representation (21)

variable	type	meaning
Par	r	parameter vector $\boldsymbol{\theta}$ (22)
Npar	i	dimension of $\boldsymbol{\theta}$
El	r	log likelihood ℓ of E-MARTS model
Lel	i	> 0 if ℓ is successfully computed

- Reserves work area and calls martsw. (cf. See Section 2.6)

5.2 martsw

variable	type	meaning
My	i	dimension of $\boldsymbol{y}_n$ in eq.(21)
Mv	i	dimension of $\boldsymbol{v}_n$ in eq.(21)
Mz	i	dimension of $\boldsymbol{z}_n$ in eq.(21)
Mn	i	length of data
Maxl	i	work area size for Ψ and Φ
Z	r	$\boldsymbol{z}_{t|t}$ in eq.(102)
V	r	$V_{t|t}$ in (103)
G	r	Γ in eq.(21)
Sigv	r	$\Sigma_V \equiv E\{\boldsymbol{v}_n \boldsymbol{v}_n^T\}$. See (21)
Sigr	r	$\Sigma_R \equiv E\{\boldsymbol{r}_n \boldsymbol{r}_n^T\}$. See (21)
Vf , Irf , Icf , Nf	–	$\Psi_{\text{Irf(j), Icf(j)}} = \text{Vf(j)} \quad (j = 1, 2, \cdots, \text{Nf})$
Vh , Irh , Ich , Nh	–	$\Phi_{\text{Irh(j), Ich(j)}} = \text{Vh(j)} \quad (j = 1, 2, \cdots, \text{Nh})$

- Σ_R is set equal to zero matrix in the present version.

5.3 martspar

variable	type	meaning
Cmd	c	'<---':code
		'--->':decode
Key	c	'tvar': Σ_T
		'svar': Σ_S
		'trinit': trend initial value
		'sight': τ in eq.(29)
		'v-weight': $u_2, u_3, \ldots, u_K$ in eq.(29)
		'd-weight': $u_1^\Delta, u_2^\Delta, \ldots, u_K^\Delta$ in eq. (29)
		'var': Σ
		'ar': $\{A_1, A_2, \ldots, A_M\}$
		'model' : model structure print out
Nch	i	channel selection
Par	r	$\boldsymbol{\theta}$ (22)
Ir , Ic	i	Val and Ival are Ir $\times$ Ic matrices
Val	r	input/output real matrix
Ival	i	input/output integer matrix

- Code/decode E-MARTS model parameter vector $\boldsymbol{\theta}$. (See Section 2.6).

5.4 gmarts

- Produces an initial guess of the parameter of the E-MARTS model(30).
- Calls gmartw.

5.5 gmartw

- (a) Apply BAYSEA procedure to each channel data, to obtain residual series, initial value of the trend and estimates of Σ_T and Σ_S.
- (b) fit MAR (or SAR) model to the multivariate residual series to obtain a initial guess of $\{A_m\}$ and Σ.
- (c) if necessary, user can force the TAR structure for the AR component. Then this routine requests initial guesses of TAR model parameters.

5.6 tarmodel

- Builds teleological model (28) by constructing (127).

5.7 remtsbx

- Constructs a diagonal matrix.

5.8 subabh

- Computes eqs.(117) and (118).

5.9 subqrs

- Computes eq.(125)

6. misc-pack miscellany routines

required subpackages: (mpi-pack) (selfcom-pack)

6.1 ready

- Searches for the data body in a file.

6.2 maskrs

- Resets mask C in eq.(61).

7. pwspx-pack

required subpackages: linop-pack (mpi-pack) (selfcom-pack)

7.1 pwspx

- Computes power spectrum(40).
  caution: Should be careful about the use of eq.(40). When AR model is fitted by the least squares method, for example, the model might not be stationary and the obtained value has no physical meaning.

7.2 subnosx

- Computes relative power contribution(49).

7.3 subcoh

- Computes choherency (43).

8. pwevol-pack

required subpackages: areachk-pack kalman-pack (mpi-pack) (selfcom-pack)

8.1 pwevol

- Constructs the recursion equation(36) and computes power building profile (35) and the scaling factor (42).

8.2 pwprof

- Computes power building profile(35)

8.3 llset

- Constructs the state transition matrix Ψ in eq.(36)

9. lgraph-pack

required subpackages: (mpi-pack) (selfcom-pack)

9.1 lgraph

- Displays a graph of a timeseries on the character display.

9.2 gfile

- Outputs graphics data summarized in Table 11. in Ssection 5.4.3.

10. linop-pack

required subpackages: (mpi-pack) (selfcom-pack)

10.1 hushld

- Executes Householder transformation(89) of a real matrix.

variable	meaning
Hh N, K	$N \times K$ matrix Hh is X of eq. (86)

10.2 solve

- Solves Householder transformed linear equation(90).

variable	meaning
Hh, K	$K \times K$ matrix Hh is $$Hh = \begin{pmatrix} A & B \\ & C \end{pmatrix}$$
A	$IP \times IP$ upper triangular matrix
B	$IP \times (K - IP)$ matrix, which is replaced by $A^{-1}B$ (91)
C	input value is a $(K - IP) \times (K - IP)$ upper triangular matrix. output value is $C^T C$ (92)

10.3 cxhushld

- Complex version of 'hushld'.

10.4 cxsolve

- Complex version of 'solve'.

10.5 fouger

- Executes Fourier transformation by Goertzel method.
- Calls fouge1

10.6 fouge1

- Executes Fourier transformation by Goertzel method.

variable	meaning
G	G(i),i=0,1,...,lg
Af	frequency f
Fc	cosine transform $\Sigma_i G(i) \cos(2\pi f i)$
Fs	sine transform $\Sigma_i G(i) \sin(2\pi f i)$

10.7 posmat

- Executes Cholesky decomposition(31).

variable	meaning
Sqrvar	$\Sigma, \tilde{L}$
Det	$\det \Sigma$
Ierr	$= 0$ if computation is successful

- Together with L, its inverse L^{-1} is given in the compact form

$$(81) \qquad \tilde{L} = \begin{pmatrix} l_{11} & \tilde{l}_{21} & \cdots & \tilde{l}_{k1} \\ l_{21} & l_{22} & \cdots & \tilde{l}_{k2} \\ \vdots & & & \vdots \\ l_{k1} & l_{k2} & \cdots & l_{kk} \end{pmatrix},$$

where

$$(82) \qquad \begin{pmatrix} l_{11}^{-1} & \cdots & \cdots & 0 \\ \tilde{l}_{21} & l_{22}^{-1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \tilde{l}_{k1} & \tilde{l}_{k2} & \cdots & l_{kk}^{-1} \end{pmatrix} \equiv L^{-1}.$$

- The inverse computation, eg. the construction of $\Sigma = L^T L$, can be done.

10.8 invmat

- Executes inversion and determinant computation of a positive definite matrix.

variable	meaning
Sqrvar	Σ, Σ^{-1}
Det	$\det \Sigma$
Ierr	$= 0$ if computation is successful

10.9 rn01

- Generates uniform random number

variable	meaning
rn01	Uniform random number
Ix	> 0 to set initial value
	$= 0$ to generate a random number

10.10 rnor

- Generates normal random number by Box-Müller method

variable	meaning
rnor	$\sim N(0, Std^2)$
Std	standard deviation

11. dall-pack

required sub-packages: areachk-pack lgraph-pack (mpi-pack) (selfcom-pack)

11.1 dall

- Numerically optimizes log likelihoods. See Appendix D.

variable	meaning
Model	name of the subroutine to compute log likelihood
A	parameter vector $\boldsymbol{\theta}$
Ip	dimension of $\boldsymbol{\theta}$
El	log likelihood l
Vd	work area
Endc	end code ($= {}^*$.0 if optimization is successful)
Step	for numerical differentiation by eq.(109)
Limit	NCOUNT limit
Lpsw	out put control
	0: no print
	3: maximum print

11.2 bill

- Davidon's algorithm (see Figure 2 of Appendix D).

11.3 numdif

- Executes numerical differentiation (109).

11.4 innerp

- Computation of the inner product.

11.5 lftint

- Find the largest integer less than or equal to X.

11.6 eval

- Evaluate a function along a line in the parameter space.

12. kalman-pack

required sub-packages: areachk-pack linop-pack (mpi-pack) (selfcom-pack)

12.1 kalman

- Executes Kalman filtering procedure (cf. Appendix C)

variable	meaning
My	dimension of $\boldsymbol{y}_n$ in eq.(21)
Mv	dimension of $\boldsymbol{v}_n$ in eq.(21)
Mz	dimension of $\boldsymbol{z}_n$ in eq.(21)
Mn	length of data
Maxl	work area size for Ψ and Φ
Z	$\boldsymbol{z}_{t\|t}$ in eq.(102)
V	$V_{t\|t}$ in (103)
G	Γ in eq.(21)
Sigv	$\Sigma_V \equiv E\{\boldsymbol{v}_n\boldsymbol{v}_n^T\}$
Sigr	$\Sigma_R \equiv E\{\boldsymbol{r}_n\boldsymbol{r}_n^T\}$
Vf , Irf , Icf , Nf	$\Psi_{\mathrm{Irf(j), Icf(j)}} = \mathrm{Vf(j)} \quad (j = 1, 2, \cdots, \mathrm{Nf})$
Vh , Irh , Ich , Nh	$\Phi_{\mathrm{Irh(j), Ich(j)}} = \mathrm{Vh(j)} \quad (j = 1, 2, \cdots, \mathrm{Nh})$
Y	$\{\boldsymbol{y}_n\}$
Rlim	missing data mark
N0	Initial data length
El	log likelihood l (104)
Lel	> 0 if l is successfully computed.

- Calls kalmaw

12.2 kalmaw

variable	meaning
zt	$\boldsymbol{z}_{t\|t-1}$ in (97)
vt	$V_{t\|t-1}$ in (98)

12.3 both

- Executes computation like $\Psi V_{t-1\|t-1}\Psi^T$ in eq.(98).

12.4 left

- Executes computation like $\Psi \boldsymbol{z}_{t-1\|t-1}$ in eq.(97).

12.5 right

- Executes computation like $V_{t\|t-1}\Phi^T$ in eq.(100).

12.6 clr

- Clears a work array.

12.7 fmatrix

- Shows the structure of the matrix defines in Terms of Vf, Irf, Icf and Nf.

13. baysea-pack

required sub-packages: areachk-pack datard-pack keysave-pack lgraph-pack (mpi-pack) (selfcom-pack)

13.1 sadjust

- Adjusts one dimensional time series data for the seasonality by BAYSEA procedure (Appendix B).

- Calls sadjuw.

13.2 sadjuw

variable	meaning
Lpe	Seasonality length, L
Trinit	$\{T_{-1}, T_0\}$
Vtrend	$\Sigma_{i=3}^{N}(T_i - 2T_{i-1} + T_{i-2})^2/N$
Vseas	$\Sigma_{i=L+1}^{N}(\Sigma_{j=1}^{L} S_{i-j})^2/(N-L)$

13.3 subsea

- Seasonally adjusts given time series

variable	meaning
Ys	$\{y_n\}$
Rlim	missing value mark. See Table 9.
Dmin	DD value that attains the minimum ABIC
Abicm	minimum ABIC (96)
Season	$\{S_n\}$
Trend	$\{T_n\}$
Irreg	$\{I_n\}$

- It is a simplified version of BAYSEA(Akaike and Ishiguro, 1980).
- Hyperparameteres u, v and w are controlled by DD and RIGID through equations:

$$
\begin{aligned}
u &= AF \times DD/RIGID \\
v &= DD \\
w &= DD \times RIGID/\sqrt{L}
\end{aligned}
$$

where $L = 4$ for quarterly data. DD and $RIGID$ are selected minimizing ABIC(Akaike (1980)) under the condition $AF = 1.0$. AF is called "Abuse Factor".

13.4 setd

- Composes a difference operator.

13.5 setdc

- Produces the prior distribution part of the design matrix.

13.6 setx

- Produces the log likelihood part of the design matrix.

13.7 decode

- Interprets the estimated parameter.

13.8 exhsld

- Constructs Householder transformation of a band structure matrix.

13.9 exsolve

- Solves least squares problem.

13.10 init

- Computes initial values of trend component or seasonal component.

14. mpi-pack

required sub-packages: (selfcom-pack)

14.1 mpiinit

- Initializes MPI.

14.2 mpion

- Starts parallel computation.

14.3 mpioff

- Finishes parallel computation.

14.4 mpison

- Allocates parallel jobs to processors.

14.5 mpisoff

- Gathers results from processors.

14.6 mpisres

- Gathers results from processors.

14.7 mpistop

- Stops (parallel) computation.

15. keysave-pack

required sub-packages: (mpi-pack) (selfcom-pack)

15.1 keysave

- Saves and retraces keyboard inputs.

15.2 keympi

- 'keysave' for parallel computation.

15.3 markinp

- Marks keyboard inputs.

15.4 jfrex

- Controls keyboard reading.

16. areachk-pack

required sub-packages: (selfcom-pack)

16.1 areachk

- Checks work area size.

16.2 iw

- Allocates integer type work area in 'iWK' of

```
        common  /comwork/  wK , iWK , limflag
```

16.3 nw

- Allocates real type work area in 'wK' of

```
        common  /comwork/  wK , iWK , limflag
```

17. selfcom-pack

required sub-packages: (mpi-pack)

17.1 selfcom

- Starts subroutine message printing.

17.2 selfend

- Ends subroutine message printing.

17.3 showc

- Prints out character type variable.

17.4 showi

- Prints out integer type variable.

17.5 showr

- Prints out real type variable.

17.6 showw

- Prints out work area size.

17.7 showm

- Prints out memo.

17.8 showm2

- Prints out two line memo.

17.9 showm3

- Prints out three line memo.

17.10 them

- Activates 'self comment' function.

17.11 oml

- Prints out 'Open Market Licence' Terms (cf. Section 5.1.4).

18. bug-pack

required sub-packages: none

18.1 bug

- Read/writes bug.map. See Section F.

18.2 sbuggle

- Smuggles message into the program.

18.3 bugnet

- Shows contents of an array.

18.4 bughtrap

- Watches unexpected discrepancy between processors in parallel computation.

18.5 bugvtrap

- Watches unexpected changes of variables.

18.6 debug

- Enables bug routine to skip some of bug.map lines.

18.7 bugp

- Halts the execution of the program.

5.4 Files

5.4.1 Time series data file

Parameters to be placed on top of time series data are shown in the following table:

Table 9. Time Series Data parameter

key	type	memo
DATASTARTINGLINE		a must. Everything before this line is ignored.
tparam		begin mark, a must.
title	c	name of the dataset
date	i	date of the first record
		in yyyymmdd format
		(for example 19980714 for Jul. 14, 1998 AD)
time	i	time of the first record in hhmmss format
		For example (133206 for 1 o'clock 32 minutes 6 second PM)
longitude	r	future use
latitude	r	future use
altitude	r	future use
length	i	The length of the data, N
		0 if 'length' is to be counted
sampling	r	sampling interval in the unit of "unit"
unit	c	time unit of the sampling interval.
		day, hour, min. sec., etc.
missing	r	missing observation mark. If the value is 0.0,
		no missing observation assumed.
		If the value is greater than 0.0 any value
		greater or equal to this value is treated
		as a missing observation.
channel	i	Number of channels of the data, K
format	c	Format of the data
		'timeorder' for data stored in the format

$$
\begin{array}{cccc}
x_{1,1}, & x_{1,2}, & \cdots, & x_{1,K} \\
x_{2,1}, & x_{2,2}, & \cdots, & x_{2,K} \\
\vdots & \vdots & & \vdots \\
x_{N,1}, & x_{N,2}, & \cdots, & x_{N,K}
\end{array}
$$

'channelwise' for data in the format

$$
\begin{array}{cccc}
x_{1,1}, & x_{2,1}, & \cdots, & x_{N,1} \\
x_{1,2}, & x_{2,2}, & \cdots, & x_{N,2} \\
\vdots & \vdots & & \vdots \\
x_{1,K}, & x_{2,K}, & \cdots, & x_{N,K}
\end{array}
$$

key	type	memo
label	c	'yes' or 'no'
		'yes' if label of each channel is given in the file
calibration	c	'yes' or 'no'
		'yes' if calibration values are given
end		end mark, a must

N.B. type 'c' indicates character type, 'i' integer number, and 'r' real number.

[**Examples**]

[Bicycle Data]

```
ot20
Professor Oya's data
DATASTARTINGLINE
tparam
title= OT20.DAT
sampling= 0.05
```

```
                unit= sec
                channel=5
                length=1000
                format=timeorder
                calibration=yes                                        .......(1)
                label=yes                                              .......(2)
                end
                   'role'  'HA'  'HT'  'ST' 'PT'                       .......(2)
                (*)
                 0.0097888  0.0353988  1.0379724 -1.4243424 1.1069856  .......(1)
                   37          25          7          29          45
                   31          30          19         26          42
                   21          31          25         25          55
                     ........
```

(1) Calibration factors are given.

(2) Labels of channel are given.

(*) Assume that this data set is stored in the file 'bicycle.dat'.

[Economic Data]

```
                toukei suuri example
                DATASTARTINGLINE
                tparam
                title= IWM data
                date = 19680101
                sampling = 0.25                        ...................(1)
                unit = year                            ...................(1)
                length=84
                channel=3
                format= timeorder
                label=yes
                end
                               'IIP-I'     'WPI'     'M2CD'
                (4X,10X,F10.0,F10.0,F10.4)
                6801        33      36.2      50.9      341692
                6802        34      36.7      50.6      354823
                6803        35      37.1      50.8      363018
                ...................
```

(1) It is quarterly data.

(*) Assume that this data set is stored in the file 'IWM.dat'.

5.4.2   AR model file

Parameter set of AR model is summarized in Table 10.

Table 10. AR model parameters

key	type	
DATASTARTINGLINE		a must. Everything before this line is ignored.
tparam		begin mark, a must
title	c	name of the dataset
aic	r	AIC value, if it is available
arfit	c	name of the procedure used to fit the model
archannel	i	number of analyzed channel K of data
arorder	i	order of the AR model M
nparam	i	number of the adjusted parameters by 'arfit'
arlabel	c	'yes' or 'no'
		'yes' if name of each channel is given in the file
end		end mark, a must

A heading containing above information and a data body of the form:

$$
\begin{pmatrix}
\Sigma \\
A_1 \\
\vdots \\
A_M
\end{pmatrix}
$$

makes an *AR model file*. An AR model file is generated whenever model fitting is executed by ARdock. The name 'aaaa.AR' is given to the file if the name of the data file is 'aaaa.dat', for example.

[Example]

```
DATASTARTINGLINE                                    ................(1)
tparam                                              ................(1)
title=IWM data                                      ................(2)
date=  19680101                                     ................(2)
time=          0
longitude=   .00000000D+00
latitude=   .00000000D+00
altitude=   .00000000D+00
length=         84                                  ................(2)
sampling=   .25000000D+00                            ................(2)
unit=year                                           ................(2)
missing=   .00000000D+00
channel=         3                                  ................(2)
format=timeorder                                    ................(2)
label=yes                                           ................(2)
calibration=no
aic=   .12389845D+04                                ................(3)
arfit=e-marts                                       ................(3)
archannel=         3                                ................(1)
arorder=         1                                  ................(1)
nparam=         0
arlabel=yes
reserve=reserved key
end                                                 ................(1)
'IIP-I' 'WPI' 'M2CD'                                ................(2)
(4d20.10)                                           ................(1)
     .1226827258D+01    -.1376259436D+00     .3078189288D+00    .....(1)
    -.1376259436D+00     .1065334082D+01    -.1240825697D+00
     .3078189288D+00    -.1240825697D+00     .1055016054D+00
     .6264084454D+00     .6572288677D+00     .1183363342D-01
    -.1663058558D+00     .9600261510D+00     .1345653910D+00
     .3372700987D-02    -.2477911763D+00     .9559383748D+00
```

(1) These lines are indispensable. Other lines can be omitted. Inversely, prepare a file with these lines then it can be fed to ARdock.

(2) This is the contents of the file 'IWM.AR' which is automatically generated when MARTS model is fitted to 'IWM.dat'. Lines marked with (2) are inherited from 'IWM.dat'.

(3) These lines are obtained as the result of model is fitting.

5.4.3 Graphics data

Suitably scaled graphical presentations are indispensable for correct understanding of a system. ARdock produces file output of graphics data including the scaling information. The heading part of each set of data has the description of the following *graphics parameter*s.

Table 11. Graphics Parameter

key	type	
gfile		begin mark, a must
figtype	c	'timeseries', 'spectrum' or 'contribution'
figtitle	c	title of the graph
length	i	length of the data
channel	i	number of channels
unit	c	x-axis unit
xmin	r	minimum of x-axis
xdiff	r	x-axis spacing
ylabel	c	y-axis label
ymin	r	y-axis minimum
ymax	r	y-axis maximum
ymissing	r	missing value mark. If the value is 0.0, there is no missing value. If the value is greater than 0.0 any value greater or equal to this value should be treated as a missing value.
end		end mark, a must

[Examples]

[timeseries]

```
gfile
figtype=timeseries
figtitle=power building profile$
length= 100
channel= 1
unit=sec
xmin= 0.000000000000000000E+00
xdiff= 0.500000000000000028E-01
ylabel=role
ymin= -1.88751305263359570
ymax= 1.88751305263359570
ymissing= 0.000000000000000000E+00
end
4                                          ..........(1)
(4d20.10)
      .6608869338D-02
      .1629887846D-01
      .2657937117D-01
      ..........
```

(1) Power building profile of variable '4'

[spectrum]

```
gfile
figtype=spectrum
figtitle=Power Spectrum
length= 81
channel= 1
unit=cycle/sec
xmin= 0.000000000000000000E+00
xdiff= 0.125000000000000000
ylabel=role
ymin= 0.000000000000000000E+00
ymax= 13.5973532085583741
ymissing= 0.000000000000000000E+00
end
4                                    ...................(1)
(4d20.10)
      .1359735321D+02
      .1031221301D+02
      .8450967004D+01
      .7156485151D+01
      .7005125301D+01
      ..........
```

(1) Power spectrum of variable '4'

[contribution]

```
gfile
figtype=contribution
figtitle=Power Contribution
length= 81
channel= 3
unit=cycle/sec
xmin= 0.000000000000000000E+00
xdiff= 0.125000000000000000
ylabel=role
ymin= 0.000000000000000000E+00
ymax= 1.00000000000000000
ymissing= 0.000000000000000000E+00
end
1 3 4 5                                    ..........(1)(2)
(4d20.10)
      .2860909689D+00       .8824673263D+00       .8848516672D+00
      .2531022966D+00       .8732733576D+00       .8885339745D+00
      .2498538393D+00       .9220250540D+00       .9333020608D+00
      ........................
```

(1) Cumulative relative power contributions derived assuming the block structure {1,3,4,5}.

(2) Relative power contributions from all blocks add up to the obvious 1.0, which need not be output.

5.4.4 @KEEP.KEY file

User's keyboard inputs can be recorded in this file and referred in retracing mode. See an example in Section 6.6.

5.4.5 E-MARTS file

Model fitting based on the numerical maximization of the log likelihood can be a time consuming task. ARdock stores values of the model parameters at the point of 'DALL?' prompt. See Figure 1b of Section 6. User can resume the optimization where the parameters are saved. A file named 'aaaa.WK' is produced when time series data in the file 'aaaa.dat' is analysed.

5.4.6 Tutorial message file

ARdock is so designed that tutorial help messages are given on user's choice (see Tabel 12a of Section 6). The help messages are stored in the file named TUTORIAL.TXT.

6 User's Manual

6.1 Prompt messages

The prompt messages from ARdock are summarized in Tables 12a, b and c. One ARdock session can be devided into three stages, start-up stage; model fit stage and interactive analysis stage. Prompts in the start-up stage and the analysis stage are collected in Tables 12a and b, respectively. Relations between these prompts are schematically shown in Figure 1a. The relations between prompts in Table 12c. is described in Figures 1b and c.

6.2 Starting up

[Info level selection]

```
SELF COMMENT -------------------------------------
 Subroutines in this program speak for themselves in order of
 their appearances, if THEM(!) is(!) called at the beginning of the program,
 and the iNFO level is set properly. This function is provided to encourage
 those who would like to make modifications.
 .....
 Present info level is  9.
  Make it 0 if you need no COMMENTs.
<<<
9                                        ...................(1)
>>>
......


 self comment -------------------------------------
 jdate/starts
  Converts calendar data to the day count from
  Jun. 1. 1601.AD., and vice versa.
I Ymd2date   =        -->,
I Ymd        =     19680101,        ...................(2)
I Time       =            0,
-------------------------------------------------
 self comment -------------------------------------
 jdate/ended
O Date       =        134350,        ...................(3)
  changed!
O Atime      =    .0000D+00,
O Dayoweek   =            1,         ...................(3)
  changed!
-------------------------------------------------
```

(1) Answer '0', then none of following 'self comments' will appear. Here and following examples, key-inputs are 'parenthesized' by '<<<' and '>>>'.

(2) Calendar date is given. See [Another exampl] in Section 5.3.2.

(3) Julian date and the day-of-week is computed.

[Key-Save option]

```
     KS-file is @KEEP.KEY
     KEY: s)ave, r)etrace, b)atch-mode or n)ullify
      option: function
        s   :  to save key inputs
        n   :  no save ,  no retrace
            :  for tutorial help make it S or N ,  instead of s or n.
        r   :  to retrace saved inputs
        b   :  BACH mode                              ................(3)
   <<<
    s                                                 ..............(1)(2)
   >>>
     SAVE mode
```

(1) Inputs answering prompts are saved in the file named @KEEP.KEY and could be retraced.

(2) To retrace saved inputs, give 'r' instead of 's'.

(3) See section 6.6 for farther details of this option.

[Data selection]

```
    d)ata or m)odel?:
   <<<
   d                                                        ...........(1)
   >>>
     data file:
   <<<
   bicycle.dat                                              ...........(1)
   >>>
    file opener started for mt    18
    ......

    DATASTARTINGLINE
    tparam values:
     1:title = OT20.DAT
     2:date =         0
     3:time =         0
     4:longitude =    .000000D+00
     5:latitude =    .000000D+00
     6:altitude =    .000000D+00
     7:length =       1000
     8:sampling =    .500000D-01
     9:unit = sec
    10:missing =    .000000D+00
    11:channel =         5
    12:format = timeorder
    13:label = yes
    14:calibration = yes                                    .............(2)
    ......
     5 - dimensional Data of length  1000                   .............(3)
    1 role      .370D+02    .310D+02    .210D+02    .130D+02   -.100D+01 ....
    2 HA        .250D+02    .300D+02    .310D+02    .330D+02    .390D+02 ....
    3 HT        .700D+01    .190D+02    .250D+02    .330D+02    .310D+02 ....
    4 ST        .290D+02    .260D+02    .250D+02    .220D+02    .150D+02 ....
    5 PT        .450D+02    .420D+02    .550D+02    .720D+02    .730D+02 ....
     select components:
   <<<
   1 2 3 4 0                                                 .............(4)
   >>>
      4-dimensional data of length  1000                    .............(5)
   role       .362D+00    .303D+00    .206D+00    .127D+00   -.979D-02 ....
   HA         .885D+00    .106D+01    .110D+01    .117D+01    .138D+01 ....
   HT         .727D+01    .197D+02    .259D+02    .343D+02    .322D+02 ....
   ST        -.413D+02   -.370D+02   -.356D+02   -.313D+02   -.214D+02 ....
    Initial data length:
   <<<
    10                                                       .............(6)
   >>>
   E-MARTS file = bicycle.WK                                 .............(7)
```

(1) The 'bicycle data' is chosen.

(2) Calibration factors are supplied.

(3) Before the calibration.

(4) Channels to be analyzed should be chosen here. It is possible to change the ordering. Fill zeros instead of the numbers of excluded channels.

(5) Calibrated values.

(6) Length of initial data N_0. See Section 4.1.

(7) Parameters to specify E-MARTS model is saved here.

Table 12a. Prompt messages in start-up stage

prompt	type	input	action	example[1]
Present info level is 9.	i	9	if self comments of routines are to be printed.	42
Make it 0 if ...		0	if no self comments are required.	
KS-file is ...	c	s	to save key inputs	43, 58
		S	to save key inputs and activate the tutorial help function	
		r	to retrace saved inputs	58–60
		n	no save, no retrace	
		N	no save, no retrace, but requests tutorial helps	
		b	batch mode[2] 'n'	59
		B	batch mode[2] 's'	59
d)ata or m)odel?	c	m	model	51
		d	data	43
AR file:	c	[a-z0-9.]*	file name	51
data file:	c	[a-z0-9.]*	file name	43
select components:	i	$[0-K]^K$	see Section 6.2 [Data selection]	43
Initial data length:	i	[0-9]*	= maximum lag N_0	43

(1) page of ...
(2) Prepare input file with the @KEEP.KEY file format. See Section 6.6.

Table 12b. Prompt messages in interactive analysis stage

prompt	type	input	action	example
ARdock:	c			
		p	power building profile	54
		P	power spectrum (+ relative power contribution)	54
		n	simulation with generated noise	56, 57
		a	evokes autodock procedure	57
		F	frequency band set	53
		B	block boundary set	52
	c i	i k	response to impulse input to k-th variable	55
		s k	response to step input to k-th variable	56
		f k	frequency response to k-th variable	55
		c k	coherency to k-th variable	55
		l k	change looking channel to the k-th cahnnel	56
mask edit: g?r?o?	c i	r $[1-K]^4$	mask operation(1)	52
		o $[1-K]$	mask operation(2)	53
	c	g	end of mask operation	
specify watching set:	i	$[01]^K$	1 in the k-th position indicates the inclusion of the k-th variable in the set	57
frequency band:	r	f_l, f_h	see Section 6.5 [Frequency range]	53
block boundary:	i	$[1-K]$	block boundary (see Section 6.5 [Block structure]	52
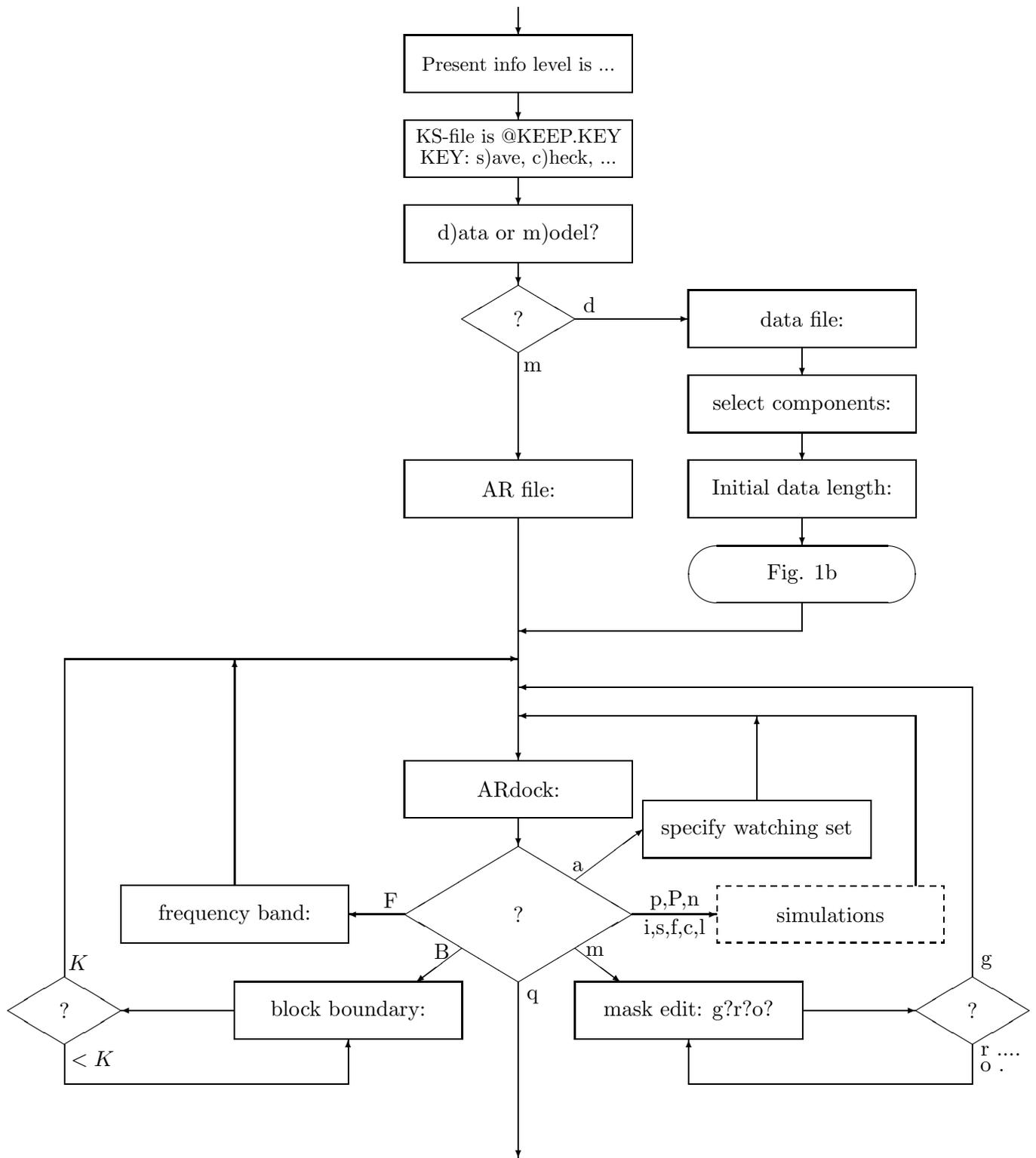
Figure 1a. Prompt messages in start-up stage and interactive analysis stage

Table 12c. Prompt messages in model fit stage

prompt	type	input	action	example
new model?	c	y or n		
E-MARTS file:	c	[a-z0-9]*	file name	
max AR order:	i	M_h	Range of MAICE search	48
mean component:	c	c	constant mean	
		l	linear trend	
		s	seasonal variation and trend	48
	c i	p k	polynomial of order k	
give periods:	i	K	seasonal periods	48
search mode: f?a?m?	c	f[1]	full automatic search for the optimal combination of hyper parameter	48
		a	automatic search for one channel	
		m	interactive manual search	
search mode: a?m?	c	a	automatic	
		m	manual	
abuse-factor:	r	AF		
RIGID: g?c?	c r	c r	r = RIGID	
	c	g	end of the search	
SAR: a?m?n?	c	a	automatic	49
		m	manual	
		n	no SAR structure	
indexm edit: g?r?	c i	r $[1\text{-}K]^4$	index matrix setting. Input four digits i,j,k,l to perform zero-one reversal of rectangular area marked by the (i,j)-th element and (k,l)-th element	
	c	g	end of the operation	
OK?	c	y or n		
TAR controller dim.:	i	K_w	in eq. (25)	50
sight:	r	τ	in eq.(29)	50
v-weights:	r	$u_2, u_3, \ldots, u_K$	in eq.(29)	50
d-weights:	r	$u_1^\Delta, u_2^\Delta, \ldots, u_K^\Delta$	in eq. (29)	50
DALL?	c	y or n	50	

(1) When the computer is a parallel machine, the processing of channels are executed parallely.
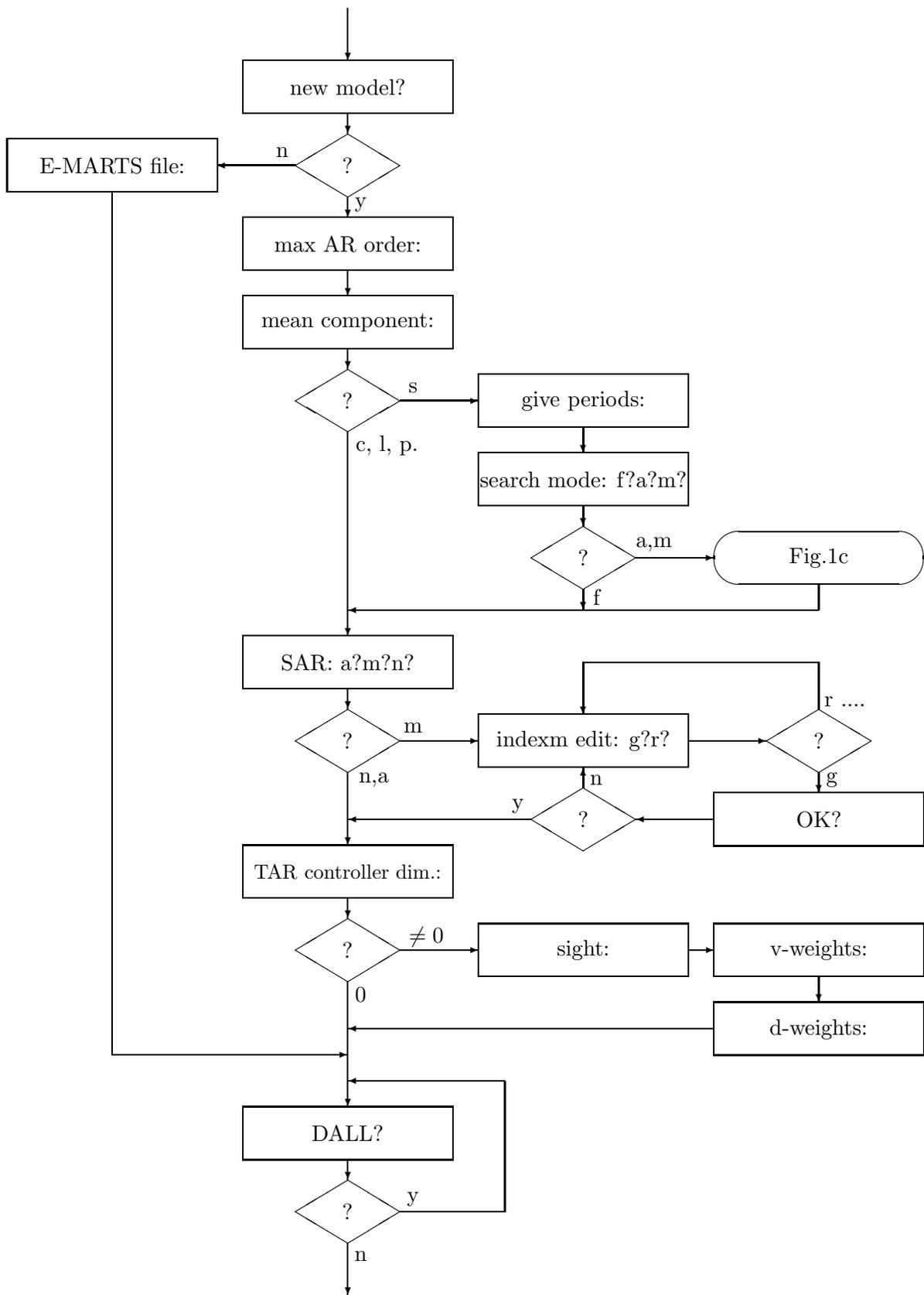
Figure 1b. Model fit stage prompts
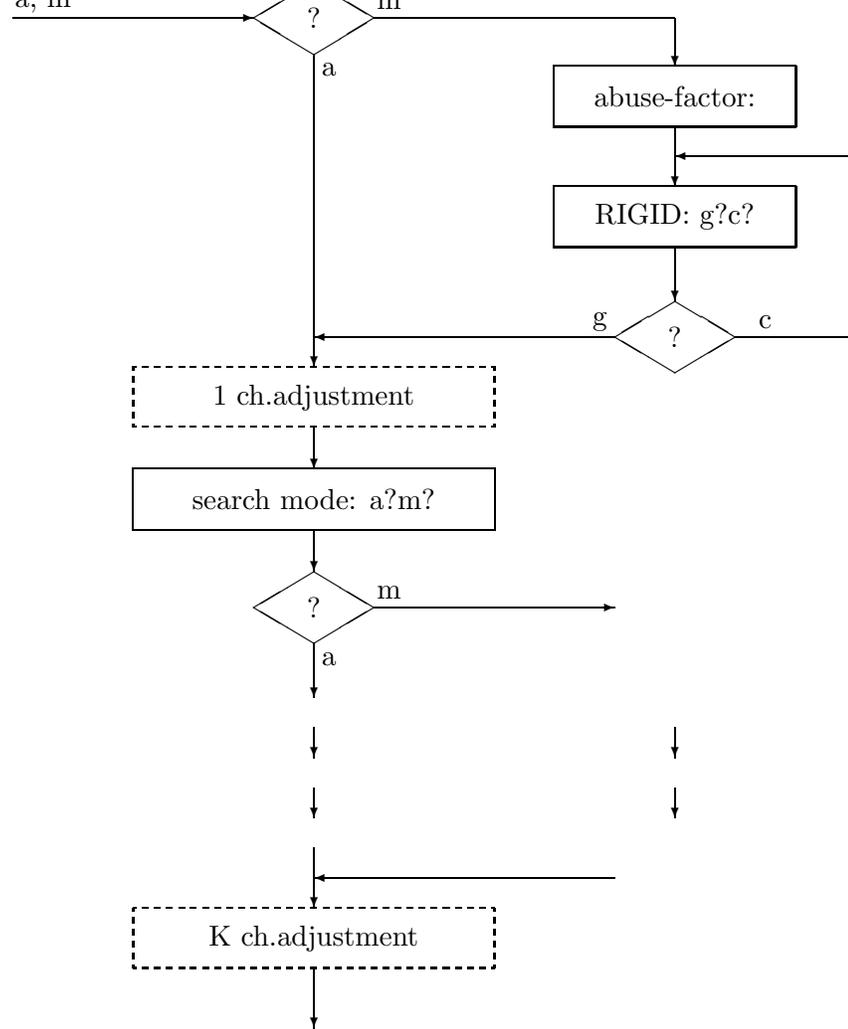
Figure 1c. BAYSEA stage prompts

6.3   Model Assembly

See section 2.6.

[AR order]

```
  max AR order:
<<<
5                                                    ...........(1)
>>>
  mean component: c?l?s?
```

(1) Maximum order for searching MAICE.

[Mean component model]

```
  mean component: c?l?s?
<<<
s                                                    ...........(1)
>>>
  channels are:WPI          M2CD          IIP-I
  give periods:
<<<
1 4 4                                                ...........(2)
>>>
  search mode: f?a?m?
<<<
f                                                    ...........(3)
>>>
.:.....
 icase  =  101
     abic (     .5000000000D+01  )  =       202.02    ..........(4)
     abic (     .5946028086D+01  )  =       211.27
..
..SAR: a?m?n?
```

(1) Select 's' for trend and seasonal adjustment, 'c' for constant, and 'l' for linear.

(2) Specify seasonality period length of each channel.

(3) Select either full-automatic (f), automatic (a) or manual mode(m).

(4) BAYSEA(Appendix B) is used. After the decision of $RIGID$ and DD by minimum ABIC procedure, the 'abuse factor' AF is increased to 30.0 to decompose each channel data into trend, seasonal and irregular component. With a parallel computer, this computation can be executed parallelly.

[SAR Model]

```
  SAR: a?m?n?
  <<<
a                                              . . . . . . . . . . . (1)
  >>>
  order selection
     order      aic
         0     23462.12
         1     11949.24
         2     10994.77
         3     10538.36
         4     10355.07
         5     10295.10
         6     10282.08
         7     10274.50
         8     10233.61
         9     10239.92
        10     10239.80
   lag =   8   AIC =    10233.61                . . . . . . . . . . . (2)
   effective variable search for   1 - th variable
            AIC           #param  Indexv
      -2098.87                35    1111
         :      1702.45       27    0111
         :     -1888.85       27    1011
         :     -2099.17       27    1101
         :     -2068.80       27    1110
      -2099.17                27    1101
         :      1785.09       19    0101
         :     -1819.20       19    1001
         :     -2055.15       19    1100
      -2099.17                27    1101

   effective variable search for   2 - th variable
            AIC           #param  Indexv
       -223.25                36    1111
         :       -81.65       28    0111
         :      1509.78       28    1011
         :         3.56       28    1101
         :      -190.60       28    1110
       -223.25                36    1111
   effective variable search for   3 - th variable
            AIC           #param  Indexv
       7085.77                37    1111
         :      7216.23       29    0111
         :      7172.62       29    1011
         :      7752.66       29    1101
         :      7085.19       29    1110
       7085.19                29    1110
         :      7223.74       21    0110
         :      7193.53       21    1010
         :      7808.70       21    1100
       7085.19                29    1110
   effective variable search for   4 - th variable
            AIC           #param  Indexv
       5470.76                38    1111
         :      5471.15       30    0111
         :      5646.16       30    1011
         :      5681.74       30    1101
         :      7145.71       30    1110
       5470.76                38    1111

   summary
   lag =   8   AIC =    10233.52                . . . . . . . . . . . (3)
           system structure
   role         1 1 0 1
   HA           1 1 1 1
   HT           1 1 1 0
   ST           1 1 1 1                          . . . . . . . . . . . (4)
       causality degree matrix
   role        3884.3   280.0     -.3    44.0
   HA           141.6  1733.0   226.8    32.6
   HT           138.5   108.3   723.5     -.6
   ST             .4    175.4   211.0  1675.0    . . . . . . . . . . . (5)
    TAR controller dim.:
```

(1) SAR model is chosen and automatic search for the index matrix is selected.

(2) MAICE order is 8. AIC value is of the ordinary MAR model(4).

(3) AIC value of the SAR model(23).

(4) The index matrix $\hat{S}$.

(5) The causality degree matrix. See Section 2.4.

[TAR Model]

```
  TAR controller dim.:
  <<<
1                                                    .............(1)
  >>>
  sight:
  <<<
0.9                                                  .............(2)
  >>>
  input 2 weights
  v-weights:
  <<<
1 1                                                  .............(2)
  >>>
  input 3 diff weights
  d-weights:
  <<<
1 1 1                                                .............(2)
  >>>
  sample.WK
  DALL?
```

(1) Select 0 if TAR is not necessary, any other digit for TAR model.

(2) Supply TAR mode parameter initial guesses.

6.4   Model Fit

[DALL]

```
  DALL?
  <<<
y                                                    .............(1)
  >>>
  D1:========== log likelihood maximization =========  .........(2)
  D1:number of processor  =  8
  D1:no Limit for NCOUNT

  D2:Hessian reset count  =   0
  D2:value =          -640.13
  D2:point
          .70707D-01   .61259D-02   .75479D-02   .10735D+02   .63100D-01
          .50231D+02   .49968D+02   .27433D+02   .25162D+02   .32900D+02
          .31163D+02   .10826D+01   .15112D-01   .89175D+00   .19791D+00
          .28376D-01   .11369D+01   .89970D+00   .25218D-01  -.18303D+00
         -.17096D+00   .88657D+00  -.14609D-01   .69909D+00  -.13084D-01
          .64189D+00
  ......

  D2:NCOUNT =     115
  D1:Bill ended with abs ( log LAMBDA) < ramlim.
  D2: ==========   Summary  =========
  D1:value =          -515.41
  D1:point
          .70707D-01   .61259D-02   .75479D-02   .17453D+00   .88408D-01
          .50018D+02   .50067D+02   .27404D+02   .25199D+02   .32864D+02
          .31273D+02   .11789D+01   .12145D+00   .99969D+00  -.19307D+00
          .22485D+00   .13612D+01   .10938D+01   .10106D-01  -.31810D+00
         -.18124D+00   .10125D+01   .64972D-01   .47044D+00   .87715D-01
          .65110D+00
  D1:val. dif. =        124.72
  D2:global profile
  vs.   log likelihood
      -650.00   -625.00   -600.00   -575.00   -550.00   -525.00   -500.00
      .00 !     *X   !        !        !        !        !        !
      .10 !       X* !        !        !        !        !        !
      .20 !        *X!        !        !        !        !        !
      .30 !         !X*X      !        !        !        !        !
      .40 !         !   X*    !        !        !        !        !
      .50 !         !    *X ! !        !        !        !        !
      .60 !         !       X*X        !        !        !        !
      .70 !         !        ! X*X     !        !        !        !
      .80 !         !        !   X*XXXXX        !        !        !
      .90 !         !        !        !   XXXXX*XXXXXX    !        !
     1.00 !         !        !        !        !      XXXXXX*     !
  D1:Hessian reset count  =    3
  D1:end code  =   115.70
  D1:============== end of maximization ===========
  sample.WK
```

```
   DALL?
<<<
 n                                        ............... (1)
>>>

E-MARTS model
   mean component: trend+seasonal
   SAR index matrix
 WPI            1 1 1
 M2CD           1 1 1
 IIP-I          1 1 1
AIC =     1082.81
npar =    26
DALL call count =    1
AR file = IWM.AR                          ............. (3)
```

(1) 'y' or 'n'

(2) DALL(cf. Appendix D) is activated to maximize the log likelihood..

(3) AR part parameters are saved here.

[When AR model file is given]

```
 d)ata or m)odel?
<<<
m                                         ................ (1)
>>>
 AR file:
<<<
IWM.AR                                    ................ (1)
>>>
 file opener started for mt   17
 DATASTARTINGLINE
 tparam values:
  1:title = IWM data
  2:date =   19680101
  3:time =          0
  4:longitude =    .000000D+00
  5:latitude =     .000000D+00
  6:altitude =     .000000D+00
  7:length =         84
  8:sampling =    .250000D+00
  9:unit = year
 10:missing =    .000000D+00
 11:channel =          3
 12:format = timeorder
 13:label = yes
 14:calibration = no
 15:aic =    .108281D+04
 16:arfit = e-marts
 17:archannel =          3
 18:arorder =          1
 19:nparam =         26
 20:arlabel = yes
 21:reserve = reserved key
 22:reserve = reserved key
 channels:WPI      M2CD       IIP-I
```

(1) AR model file 'IWM.AR' is selected.

6.5   Interactive System Analysis

[Finish]

```
  1:WPI              menu: q)uit m)ask F)band B)lock l)ook <1- 3>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
 ARdock:
<<<
q                                         ............... (1)
>>>
wK area usage : 58885                     ............... (2)
wK area limit : 1000000                   ............... (3)
iWK area usage: 142                       ............... (2)
iWK area limit: 10000                     ............... (3)
```

(1) 'q' for quit.

(2) Work area usage in the session.

(3) Present setting of compilation parameters 'nwork' and 'iwork'.

[Variance Covariance Matrix]

```
ARdock
    Variance-covariance/correlation matrix          .......(1)
role              .01      .85      .73    -.21      .......(2)
HA                .03      .15      .90     .26      .......(2)
HT               1.16     6.88   381.73     .29
ST               -.12      .70    38.78   47.43

block diagonal model:  1  2  3  4                   .......(3)
              ( d_AIC  =     4178.36)          .......(4)(5)
```

(1) Elements of the Variance-covariance matrix Σ (6) are given in the diagonal and bottom left part. In the top right part, elements of the correlation matrix are given.

(2) The correlations between the first, second and third variable innovation series are high.

(3) Power spectra and relative power contributions are computed using equations (48) and (49) assuming block diagonal structure(45) of Σ. Present block diagonal structure is specified by {1,2,3,4}. It is equivalent to assuming that Σ is a diagonal matrix.

(4) d_AIC is the difference of AIC(78) and the AIC(71) of the ordinary MAR model(4).

(5) The value 4178.36 indicates the irrelevance of the block structure {1,2,3,4}, and warns that the relative power contribution should be read with some reservation.

[Block structure]

```
block diagonal model:  1  2  3  4
              ( d_AIC  =     4178.36)
.......
  1:role                   menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
<<<
B                                                 ............(1)
>>>
  choose from: 1 2 3 4
   block boundary:
<<<
3
>>>
   block boundary:
<<<
4
>>>
  block diagonal model:  3  4                      ............(2)
              ( d_AIC  =     1241.97)          ............(3)
```

(1) Block structure change is selected.

(2) New block structure is {3,4}.

(3) Some improvement of d_AIC value.

[Masking(1)]

```
  1:role                   menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
<<<                                               ............(1)
m
>>>
   presents mask
role          1111
HA            1111
HT            1111
ST            1111

  mask edit: g?r?o?                               ............(2)
<<<
r 1 1 4 4
>>>
role          1000                                .........(4)(5)
HA            0100
HT            0010
ST            0001
```

```
     mask edit: g?r?o?
 <<<
 r 1 1 2 2                                              ...........(3)
 >>>
 role           1100
 HA             1100
 HT             0010                                    .......(4)(5)
 ST             0001

  mask edit: g?r?o?
 <<<
 g                                                      ...........(6)
 >>>
  1:role              menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
```

(1) Selection of mask operation

(2) Instruction of zero-one reversal of rectangular area marked by the indexes (1,1) and (4,4) of top-left and bottom-right corner elements.

(3) Instruction of zero-one reversal of rectangular area marked by the indexes (1,1) and (2,2) of top-left and bottom-right corner elements.

(4) The operation is successfully carried out.

(5) Diagonal elements are free for mask operation.

(6) Exit from mask operation mode

[Masking(2)]

```
  1:role              menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
 <<<
 m                                                      ............(1)
 >>>

   presents mask
 role           1100
 HA             1100
 HT             0010
 ST             0001/

  mask edit: g?r?o?
 <<<
 o 2                                                    .............(2)
 >>>
 role           1000
 HA             1111                                    .............(3)
 HT             0010
 ST             0001

  mask edit: g?r?o?
 <<<
 g
 >>>
  2:HA        menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext .....(4)
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
```

(1) Selection of mask operation

(2) Selection of the 'open loop response' of the '2'-th variable.

(3) The mask with 1's on the '2'-th row and diagonal is generated.

(4) Note that the 'looking variable' is changed to the '2'-th variable.

[Frequency Range]

```
  3:IIP-I              menu: q)uit m)ask F)band B)lock l)ook <1- 3>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
 <<<
 F                                                      ..............(1)
 >>>
  frequency band:
 <<<
 0,0.6                                                  ..............(2)
 >>>
```

(1) Selection of frequency range change.

(2) Instruction to set a new range 0 cycle/'time unit' $\leq f \leq 0.6$ cycle/'time unit'.

[Power Building Profile]

```
    1:role                    menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
     p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
    ARdock:
    <<<
    P
    >>>                                             .........(1)
    power building profile

    sec                    vs.  role               .........(2)
         -3.00          -2.00       -1.00       .00        1.00      2.00         3.00
         .20 !            !           !          !  *        !         !           !
         .45 !            !           !          !  *XX      !         !           !
         .70 !            !           !          !    XX*    !         !           !
         .95 !            !           !          !      *!   !         !           !
        1.20 !            !           !          !      *!   !         !           !
        1.45 !            !           !          !       *   !         !           !
        1.70 !            !           !          !       *   !         !           !
        1.95 !            !           !          !       *   !         !           !
        2.20 !            !           !          !       *   !         !           !
        2.45 !            !           !          !       *   !         !           !
        2.70 !            !           !          !       *   !         !           !
        2.95 !            !           !          !       *   !         !           !
        3.20 !            !           !          !       *   !         !           !
        3.45 !            !           !          !       *   !         !           !
        3.70 !            !           !          !       *   !         !           !
        3.95 !            !           !          !       *   !         !           !
        4.20 !            !           !          !       *   !         !           !
        4.45 !            !           !          !       *   !         !           !
        4.70 !            !           !          !       *   !         !           !
        4.95 !            !           !          !       *   !         !           !
```

(1) To see the power building profile $P_{kk}^B(t)$ (35).

(2) The 'graph' shows the normalized power building profile $P_{kk}^B(t)/\sigma_k^2$, where the normalizing factor σ_k is of the original (not-masked) AR model.

[Power Spectrum and Relative Power Contribution]

```
    3:IIP-I                    menu: q)uit m)ask F)band B)lock l)ook <1- 3>  N)ext
     p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
    ARdock:
    <<<
    P                                        ...........................(1)
    >>>
               spectrum structure of the  3-th variable    ......(2)(3)(4)
    cycle / year                 :IIP-I
        .00 :         :         :         :         :  123 xx              :
        .03 :         :         :         :         :  12 3 xxx            :
        .06 :         :         :         :         :12   3 xxx            :
        .09 :         :         :         :      : 12   3 xxx             :
        .12 :         :         :         :    12:   3 xxxxx              :
        .16 :         :         :         : 1     3 xxxxxxx               :
        .19 :         :         :    :12         3 xxxxxxxxxxx            :
        .22 :         :         : 12            3 xxxxxxxxxxxxxxxxx        :
        .25 :         :      :12               3 xxxxxxxxxxxxxxx          :
        .28 :         :   :12                  3 xxxxxxxxx               :
        .31 :         :12                      3 xxxxx                   :
        .34 :       1                          3 xxxx                    :
        .38 :    12                            3 xxx                     :
        .41 :  1                               3 xx                      :
        .44 :12                                3 xx                      :
        .47 ::1                                3 x                       :
        .50 :1                                 3 x                       :
```

(1) To see the power spectrum $P_{kk}(f)$ defined by eq.(48) and relative power contribution (49).

(2) Cumulative relative power contribution is shown in the middle area and the power spectrum is shown in the right margin.

(3) Vertical lines shown in relative power contribution area by ':' corresponds to 20%, 40% , ... 100% levels.

(4) Scaling of the power spectrum is chosen so that the head of the 'graph' of the theoretical spectrum of 'not-masked' AR model touches the rightmost ':'-line.

[Coherency]

```
        1:role                menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
          p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
        ARdock:
        <<<
        c 3                                                      .........(1)
        >>>
                coherency of the  1 - th variable
        cycle / sec                  :role
            .00 :           :        3    :           :           : XXXXXXXXXXXXXXXX:
            .16 :           :        3    :           :           : XXXXXXXXXXXXXXX :
            .31 3           :        :           :           :           : XXXXXXXXXXX
            .47 :           :        :           :    3      :           : XXXXXXXX
            .62 :           :        :           :           :        3  : XXXXXXXXX
            .78 :           :        :           :           :         3 XXXXXXXXXXXXXX :
            .94 :           :        :           :           :         3 XXXXX
           1.09 :           :        :           :           :         3: X
           1.25 :           :        :           :           :        3 : X
           1.41 :           :        :           :           :       3  : X
           1.56 :           :        :           :           :      3   : X
           1.72 :           :        :           :           :     3    : X
           1.88 :           :        :           :           :   3      : X
           2.03 :           :        :           :           :  3       : X
           2.19 :           :        :           :           :.3        : X
           2.34 :           :        :           :           :3         : X
           2.50 :           :        :           :           3:         : X
```

(1) To see the coherency (43) between variable 1 and 3.

[Impulse Response]

```
        1:role                menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
          p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
        ARdock:
        <<<
        i 2                                          ....................(1)
        >>>
        response to impulse:HA                        ....................(2)
        sec                    vs.  role
            -3.00      -2.00      -1.00       .00      1.00      2.00      3.00
           .00 |          |          |          *          |          |          |
           .05 |          |          |         *|          |          |          |
           .10 |          |          |         *|          |          |          |
           .15 |          |          |         *|          |          |          |
           .20 |          |          |        * |          |          |          |
           .25 |          |          |       *  |          |          |          |
           .30 |          |          |       *  |          |          |          |
           .35 |          |          |       *  |          |          |          |
           .40 |          |          |      *   |          |          |          |
           .45 |          |          |      *   |          |          |          |
           .50 |          |          |     *    |          |          |          |
           .55 |          |          |    *     |          |          |          |
           .60 |          |          |    *     |          |          |          |
           .65 |          |          |   *      |          |          |          |
           .70 |          |          | | *      |          |          |          |
           .75 |          |          | |*       |          |          |          |
           .80 |          |          | *        |          |          |          |
           .85 |          |          | *        |          |          |          |
           .90 |          |         *|          |          |          |          |
           .95 |          |        * |          |          |          |          |
```

(1) Impulse input(50) to variable '2' is applied.

(2) The normalized response(53). The looking variable is '1'.

[Frequency Response]

```
        1:role                menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
          p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
        ARdock:
        <<<
        f 1                                          .................(1)
        >>>
        frequency response to:role                   .................(2)
        cycle / sec                  vs.  role
            -3.00      -2.00      -1.00       .00      1.00      2.00      3.00
           .00 |          |          |          |          *X         |          |
           .16 |          |          |          |          X*|        |          |
           .31 |          |          |          |          |*X        |          |
           .47 |          |          |          |          |  X*XXX    |          |
           .62 |          |          |          |          |      XXX* |          |
           .78 |          |          |          |          |          |          |
           .94 |          |          |          |          |          |          |
          1.09 |          |          |          |          |  XX*XXXXXX|          |
          1.25 |          |          |          |          | *XX       |          |
          1.41 |          |          |          |          |* |        |          |
          1.56 |          |          |          |          |*|         |          |
          1.72 |          |          |          |          *|          |          |
          1.88 |          |          |          |          *|          |          |
          2.03 |          |          |          |          *|          |          |
          2.19 |          |          |          |          *|          |          |
          2.34 |          |          |          |          *|          |          |
          2.50 |          |          |          |          *|          |          |
```

(1) Command to get the frequency response(56). Note that the result is meaningful only when the summation in the formula converges.

(2) The amplitude factor of the normalized response(57)is shown. The looking variable is '1'.

[Step Response]

```
    2:HA                     menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
      p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
     ARdock:
     <<<
     s 1                                      .........................(1)
     >>>
     response to step:role                    .........................(2)
     sec              vs.  HA
         -3.00       -2.00       -1.00         .00       1.00       2.00       3.00
        .20 !           !           !           !     *     !           !           !
        .45 !           !           !           !       *   !           !           !
        .70 !           !           !           !     *     !           !           !
        .95 !           !           !           !   *       !           !           !
       1.20 !           !           !           !  *        !           !           !
       1.45 !           !           !           !  *        !           !           !
       1.70 !           !           !           !  *        !           !           !
       1.95 !           !           !           !  *        !           !           !
       2.20 !           !           !           !  *        !           !           !
       2.45 !           !           !           !  *        !           !           !
       2.70 !           !           !           !  *        !           !           !
       2.95 !           !           !           !  *        !           !           !
       3.20 !           !           !           !  *        !           !           !
       3.45 !           !           !           !  *        !           !           !
       3.70 !           !           !           !  *        !           !           !
       3.95 !           !           !           !  *        !           !           !
       4.20 !           !           !           !  *        !           !           !
       4.45 !           !           !           !  *        !           !           !
       4.70 !           !           !           !  *        !           !           !
       4.95 !           !           !           !  *        !           !           !
```

(1) Step input(58) to variable '1' is applied.

(2) The normalized response(60). The looking variable is '2'.

[Looking Variable Change]

```
    1:WPI                    menu: q)uit m)ask F)band B)lock l)ook <1- 3>  N)ext
      p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
     ARdock:
     <<<
     l 2                                   ...................(1)
     >>>
```

(1) Instruction to change the looking variable to '2'. Do not mistake the el for one.

[Simulation]

```
    1:role                   menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
      p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
     ARdock:
     <<<
   n                                               .................(1)
     >>>
     simulation with generated noise
     sec              vs.  role
        -15.00      -10.00       -5.00         .00       5.00      10.00      15.00
       2.45 !           !           !         *!          !           !           !
       4.95 !           !           !         *!          !           !           !
       7.45 !           !           !          !*         !           !           !
       9.95 !           !           !        * !          !           !           !
      12.45 !           !           !       *  !          !           !           !
      14.95 !           !           !     *    !          !           !           !
      17.45 !           !           !        *X!          !           !           !
      19.95 !           !           !          !X*        !           !           !
      22.45 !           !           !          !*         !           !           !
      24.95 !           !           !        X*!          !           !           !
      27.45 !           !           !       *X !          !           !           !
      29.95 !           !           !       *  !          !           !           !
      32.45 !           !           !        * !          !           !           !
      34.95 !           !           !         *!          !           !           !
      37.45 !           !           !          *          !           !           !
      39.95 !           !           !        * !          !           !           !
      42.45 !           !           !      *XXX !          !           !           !
      44.95 !           !           !          XXX*        !           !           !
      47.45 !           !           !          !*          !           !           !
      49.95 !           !           !          *           !           !           !
    1:role                   menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
      p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
```

```
  ARdock:
  <<<
n                                        ........................(2)
  >>>
simulation with generated noise
sec              vs.  role
     -15.00      -10.00      -5.00        .00      5.00      10.00      15.00
  2.45 |           |           |          !    *    |           |           |
  4.95 |           |           |          !*        |           |           |
  7.45 |           |           |         X*!        |           |           |
  9.95 |           |           |       *XX!         |           |           |
 12.45 |           |           |          !XX*      |           |           |
 14.95 |           |           |          !*        |           |           |
 17.45 |           |           |          *         |           |           |
 19.95 |           |           |         *X         |           |           |
 22.45 |           |           |          !XX*      |           |           |
 24.95 |           |           |        *XX         |           |           |
 27.45 |           |           |          !X*       |           |           |
 29.95 |           |           |          !   X*    |           |           |
 32.45 |           |           |         *X         |           |           |
 34.95 |           |           |          !*        |           |           |
 37.45 |           |           |        *!          |           |           |
 39.95 |           |           |       * !          |           |           |
 42.45 |           |           |          *         |           |           |
 44.95 |           |           |          !  *      |           |           |
 47.45 |           |           |          ! *       |           |           |
 49.95 |           |           |          !*        |           |           |
```

(1) Output of AR model(5) to the innovation(6) input, which is equivalent to the output of the linear system (1) to the driving innovation (2).

(2) Different innovation sequence is used every time. But the looking variable change keeps the same noise.

[autodock]

```
  4:ST                        menu: q)uit m)ask F)band B)lock l)ook <1- 4>  N)ext
    p)bp  P)ower  n)oise  i)mpulse *  s)tep *  f)req *  c)oherency *
  ARdock:
  <<<
a                                        .........(1)
  >>>
 channels are         :role      HA          HT          ST
 specify watching set:
  <<<
1 1 0 0                                   ......(2)(3)
  >>>

feedback character matrix with respect to the 1 -th variable

role          .0 ******    1.0    1.3
HA         11.4      .0    1.5    1.3
HT        291.6 1067.0     .0    1.0
ST          1.0     1.1    1.2     .0

           1 2 3 4
role     1  -   -  1
HA       2 -   - -  2
HT       3 - -     3
ST       4     -   4                       ........(5)
           1 2 3 4


feedback character matrix with respect to the 2 -th variable

role          .0 ******    1.0     .9
HA           .6      .0     .4    1.3
HT         16.1 1372.8      .0    1.0
ST           .9     1.1    1.3     .0

           1 2 3 4
role     1  -      1
HA       2 +   + -  2
HT       3 - -     3
ST       4   - -   4                       ..........(5)
           1 2 3 4


feedback character matrix with respect to the 3 -th variable

role          .0 ******    1.0    1.0
HA          6.0      .0    1.1    1.3
HT         27.7  867.2      .0    1.0
ST           .9     1.1    1.2     .0

           1 2 3 4
role     1  -      1
HA       2 -     -  2
HT       3 - -     3
ST       4     -   4                       ...........(5)
           1 2 3 4
```

```
         feedback character matrix with respect to the 4 -th variable
         role           .0 ******    1.0     2.1
         HA           59.6      .0    4.5      .8
         HT            6.3   482.3     .0     1.0
         ST            1.0     1.0     .6      .0

                      1 2 3 4
         role       1   -     -   1
         HA         2 -     - + 2
         HT         3 - -       3                  ............(5)
         ST         4       +   4
                      1 2 3 4


         summary                                    ............(3)
         feedback character matrix with respect to the
            watching set: 1 1 0 0
         maximum
         role          1.0 ******    1.0     1.3     ............(4)
         HA           11.4     1.0    1.5     1.3
         HT          291.6  1372.8    1.0     1.0
         ST            1.0     1.1    1.3     1.0
         minimum
         role          1.0 ******    1.0      .9     ............(4)
         HA             .6     1.0     .4     1.3
         HT           16.1  1067.0    1.0     1.0
         ST             .9     1.1    1.2     1.0
         diagram
                      1 2 3 4
         role       1   -     -   1
         HA         2 *     * - 2
         HT         3 - -       3                  ............(5)
         ST         4   - -     4
                      1 2 3 4
```

(1) autodock procedure is activated

(2) inclusion of the first and the second variables to the watching set

(3) 'summary ' ignores the result on the third and fourth variable

(4) The maximum and the minimum of the (1,4)-th element of feedback character matrix are 1.3 and 0.9 respectively.

(5) '+/-' for positive/negative feedback. Feedback character between 0.9 and 1.1 is considered to be neutral. Under the present setting of watching set, the feedback character of effect from variable '1' to variable '2' and '3' to '2'are ambiguous.

6.6   Key save

```
  KS-file is @KEEP.KEY
  KEY: s)ave, r)etrace, b)atch-mode or n)ullify
   option: function
     s   :   to save key inputs
     n   :   no save ,  no retrace
         :   for tutorial help make it S or N ,  instead of s or n.
     r   :   to retrace saved inputs
     b   :   BACH mode
<<<
s                                                 ............(1)
>>>
  SAVE mode
  d)ata or m)odel?
<<<
d                                                 ............(2)
>>>
  data file:
<<<
IWM.dat                                           ............(2)
>>>
....

  select components:
<<<
2 3 1                                             ............(2)
>>>

  Initial data length:
<<<
1                                                 ............(2)
>>>

  new model?
<<<
y                                                 ............(2)
>>>
  max AR order:
<<<
1                                                 ............(2)
>>>
```

```
 mean component: c?l?s?
 <<<
 s                                              ............(2)
 >>>
 channels are:WPI          M2CD          IIP-I
 give periods:
 <<<
 1 4 4                                          ............(2)
 >>>
 search mode: f?a?m?
 <<<
 f                                              ............(2)
 >>>
...
 SAR: a?m?n?
 <<<
 n                                              ............(2)
 >>>
 TAR controller dim.:
 <<<
 0                                              ............(2)
 >>>
 DALL?
 <<<
 y                                              ............(2)
 >>>
.....
```

[@KEEP.KEY file] ....(2)

```
     d)ata, m)odel or g)enerated data ?
     d
     data file:
     IWM.dat
     select components:
      2 3 1
     Initial data length:
      1
     new model?
     y
     max AR order:
      1
     mean component: c?l?s?
      s
     give periods:
      1 4 4
     search mode: f?a?m?
     f
     SAR: a?m?n?
     n
     TAR controller dim.:
      0
     DALL?
     y
     DALL?
     n
     ARdock:
      q
```

[Modified @KEEP.KEY file]

```
     d)ata, m)odel or g)enerated data ?
     d
     data file:
     IWM.dat
     select components:
      2 3 1
     Initial data length:
      1
     new model?
     y
     max AR order:
      1
     mean component: c?l?s?
      s
     --------------------              ............(3)
     give periods:
      1 4 4
     search mode: f?a?m?
     f
     SAR: a?m?n?
     n
     TAR controller dim.:
      0
     =============              ............(4)
     DALL?
     y
     DALL?
     n
     ARdock:
      q
```

(1) save mode

(2) these inputs are saved in the @KEEP.KEY file

(3) Add a line '---------' in the file @KEEP.KEY for one point manual input.

(4) Add a line '=======' in the file to end retracing there.

[Retrace]

```
    KS-file is @KEEP.KEY
    KEY: s)ave, r)etrace, b)atch-mode or n)ullify
     option: function
       s   :   to save key inputs
       n   :   no save ,  no retrace
           :   for tutorial help make it S or N ,  instead of s or n.
       r   :   to retrace saved inputs
       b   :   BACH mode
   <<<
 r                                            .............(1)
  >>>

 ....

   Input!:give periods:                       .............(2)
  <<<
  4 4 4
  >>>

 ....

   DALL?                                       .............(3)
   end of retrace
```

(1) retrace mode is selected

(2) one point manual input

(3) retracing ended here

7 Acknowledgment

References

[1] Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle, *2nd International Symposium on Information Theory* (Petrov, B. N. and Csaki, F. eds.), Akademiai Kiado, Budapest, pp.267–281. (Reproduced in *Breakthroughs in Statistics*, volume 1, S.Kotz and N. L. Johnson, eds., Springer Verlag, New York, (1992).)

[2] Akaike, H. (1980). "Likelihood and the Bayes procedure", *Bayesian Statistics*, Bernardo, J.M., De Groot, M.H., Lindley, D.U. and Smith, A.F.M. eds., University Press, Valencia, Spain.

[3] Akaike, H., Arahata, E. & Ozaki, T. (1975). TIMSAC-74: A time series analysis and control program package - (1) & (2), *Computer Science Monographs* No.5 & No.6, The institute of Statistical Mathematics, Tokyo.

[4] Akaike, H. and Ishiguro, M. (1980). "BAYSEA, a Bayesian seasonal adjustment program", *Computer Science Monographs*, No. 13, The Institute of Statistical Mathematics, Tokyo.

[5] Akaike, H. and Ishiguro, M. (1985). BAYSEA, in TIMSAC-84, Computer Science Monographs(Akaike, H.), Vol.22, pp.1-55, The Institute of Statistical Mathematics.

[6] Akaike, H., Kitagawa, G., Arahata, E. and Tada, F. (1979). " TIMSAC-78", *Computer Science Monographs*, No. 11, The Institute of Statistical Mathematics, Tokyo.

[7] Akaike, H. and Nakagawa, T. (1988). Statistical Analysis and Control of Dynamic Systems, Kluwer Academic Publishers, Dordrecht/Tokyo.

[8] Davidon, W.C. (1968). Variance Algorithm for Minimization, *Computer Journal*, *10*, 406-410.

[9] DiBona, C. Ockman, S. & Stone, M. (1999). *OPENSOURCES*, O'Reilly & Associates, Inc., Tokyo.

[10] Forsythe, G.E. and Moler, C.B. (1969). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Inc.

[11] Golub, G.H. (1965). "Numerical methods for solving linear least square problems", *Numer. Math.*, 7, 206-216.

[12] Ishiguro, M. (1984). Computationally Efficient Implementation of a Bayesian Seasonal Adjustment Procedure, J. Time Series Analysis, Vol.5, No.4, pp.245-253.

[13] Ishiguro, M. (1989). System Analysis with Multivariate AR model, Operations Research, Vol.34, No.10, pp.547-554. (in Japanese)

[14] Ishiguro, M. (1994). System Analysis and Seasonal Adjustment through Model Fitting, in The First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach(Bozdogan, H.), PP.79-91, Kluwer, Netherland.

[15] Ishiguro. M. (1995). DEBB, a debugging bug, manual, ISM Report on Statistical Computing, RSC-026. (in Japanese)

[16] Ishiguro, M. (1998). Teleological Model, Proceedings of the Institute of Statistical Mathematics, Vol.46, No.2, pp.383-399. (in Japanese)

[17] Ishiguro, M. (1999). Statistical Bicycle Model, ESTRELA, No.64, pp.90-93. (in Japanese)

[18] Ishiguro, M. (1999). Technical Aspects of Information Criterion Statistics, Vol.47, No.2, in press. (in Japanese)

[19] Ishiguro, M. and Akaike, H. (1989). DALL: Davidon's Algorithm for Log Likelihood Maximization, Computer Science Monographs, No.25.

[20] Ishiguro, M. and Kato, H (1995). Economic System Analysis, Mathematical Sciences, No.389, pp.50-57. (in Japanese)

[21] Ishiguro, M. and Oya, T (1999). Analysis of Human/2-wheeled-Vehicle System by ARdock, in *Practice of Time series Analysis* (Akaike,H. & Kitagawa, G. eds.), pp.209-228, Springer-Verlag, New York.

[22] Kato, H., Naniwa, S. and Ishiguro, M. (1996). A Bayesian Multivariate Nonstationary Time Series Model for Estimating Mutual Relationships among Variables, Journal of Econometrics, Vol.75, pp.147-161.

[23] Kitagawa, G.(1981). A nonstationary time series model and its fitting by a recursive filter, Journal of Time Series Analysis Vol.2, pp.103-116.

[24] Kitagawa, G. (1993). FORTRAN 77 Time Series Analysis Programing, Iwanami, Tokyo. (in Japanese).

[25] Sakamoto, Y., Ishiguro, M. & Kitagawa, G. (1986). *Akaike Information Criterion Statistics*, D.Reidel Publishing Company, Dordrecht/Tokyo.

Appendix

A   Householder Transformation and Least Squares Method

For any complex column vector $\boldsymbol{w}$ of length 1, U defined by

$$(83) \qquad\qquad U = I - 2\boldsymbol{w}\boldsymbol{w}^*$$

is an unitary matrix, since

$$UU^* = U^*U = I - 4\boldsymbol{w}\boldsymbol{w}^* + 4\boldsymbol{w}\mathbf{w}^*\boldsymbol{w}\mathbf{w}^* = I.$$

Assume $\|\boldsymbol{a}\|^2 = \|\boldsymbol{b}\|^2$ and $\boldsymbol{b}^*\boldsymbol{a}$ is real. Then

1. U derived from

$$(84) \qquad\qquad \boldsymbol{w} = \frac{\boldsymbol{a} - \boldsymbol{b}}{\|\boldsymbol{a} - \boldsymbol{b}\|},$$

    has the property

$$
\begin{aligned}
U\boldsymbol{a} &= \boldsymbol{a} - 2\frac{(\boldsymbol{b} - \boldsymbol{a})(\boldsymbol{b}^* - \boldsymbol{a}^*)^T}{\|\boldsymbol{b} - \boldsymbol{a}\|^2}\boldsymbol{a} \\
&= \boldsymbol{a} - \frac{(\boldsymbol{b} - \boldsymbol{a})(\boldsymbol{b}^* - \boldsymbol{a}^*)^T}{\|\boldsymbol{b} - \boldsymbol{a}\|^2}(\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{a} - \boldsymbol{b}) \\
&= \boldsymbol{b} - \frac{(\boldsymbol{b} - \boldsymbol{a})(\boldsymbol{b}^*\boldsymbol{a} - \boldsymbol{a}^*\boldsymbol{b})}{\|\boldsymbol{b} - \boldsymbol{a}\|^2} \\
&= \boldsymbol{b}.
\end{aligned}
$$

2. If $\boldsymbol{b}$ is of the form $(b + id, 0, \ldots, 0)^T$, and $\|\boldsymbol{b} - \boldsymbol{a}\|$ is to be maximized, b and d are uniquely determined by

$$(85) \qquad\qquad \begin{cases} b = -a_1\ h/\sqrt{a_1^2 + c_1^2} \\ d = -c_1\ h/\sqrt{a_1^2 + c_1^2} \end{cases}$$

    where $\boldsymbol{a} = (a_1 + ic_1, \cdots)$ and $h = \sqrt{\boldsymbol{a}^*\boldsymbol{a}}$.

Transformation U defined by eqs. exp(85),(84) and (83) will be called Householder transformation derived from **a**.

For any $n \times k$ matrix

$$(86) \qquad\qquad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}.$$

Let U_n be the Householder transformation derived from $(x_{11}, x_{21}, \ldots, x_{n1})^T$, then we have

$$(87) \qquad\qquad U_n X = \begin{bmatrix} x_{11}^{(1)} & x_{12}^{(1)} & \cdots & x_{1k}^{(1)} \\ 0 & x_{22}^{(1)} & \cdots & x_{2k}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x_{n2}^{(1)} & \cdots & x_{nk}^{(1)} \end{bmatrix}.$$

Likewise, using U_{n-1} derived from $(x_{22}^{(1)}, x_{32}^{(1)}, \ldots, x_{n2}^{(1)})^T$, we have

$$(88) \qquad \begin{pmatrix} 1, & 0^T \\ 0, & U_{n-1} \end{pmatrix} U_n X = \begin{bmatrix} x_{11}^{(1)} & x_{12}^{(1)} & x_{13}^{(1)} & \cdots & x_{1k}^{(1)} \\ 0 & x_{22}^{(2)} & x_{23}^{(2)} & \cdots & x_{2k}^{(2)} \\ \vdots & 0 & x_{33}^{(2)} & \cdots & x_{3k}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & x_{n3}^{(2)} & \cdots & x_{nk}^{(2)} \end{bmatrix}.$$

We can continue this process to get finally,

$$(89) \qquad \begin{pmatrix} I_{k-1}, & 0 \\ 0, & U_{n-k+1} \end{pmatrix} \cdots \begin{pmatrix} 1, & 0^T \\ 0, & U_{n-1} \end{pmatrix} U_n X = \begin{bmatrix} x_{11}^{(1)} & x_{12}^{(1)} & \cdots & x_{1k}^{(1)} \\ 0 & x_{22}^{(2)} & \cdots & x_{2k}^{(2)} \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & x_{kk}^{(k)} \\ \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

Note that

$$U = \begin{pmatrix} I_{k-1}, & 0 \\ 0, & U_{n-k+1} \end{pmatrix} \cdots \begin{pmatrix} 1, & 0^T \\ 0, & U_{n-1} \end{pmatrix} U_n$$

is an unitary matrix. Let y_j and a_j be m-vector and k-vector respectively, then the set of least square problems can be solved by minimizing

$$(90) \qquad \|y_j - X a_j\|^2 = \|U(y_j - X a_j)\|^2 \equiv \|U y_j - A a_j\|^2.$$

Using the notation

$$\begin{pmatrix} A & B \\ & C \end{pmatrix} \equiv U(X, y_1, y_2, \ldots, y_M),$$

the solutions of the least squares problems are obtained by

$$(91) \qquad (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_M) = A^{-1} B$$

and

$$(92) \qquad C^T C = (y_1 - X\hat{a}_1, y_2 - X\hat{a}_2, \ldots, y_M - X\hat{a}_M)^T$$
$$(y_1 - X\hat{a}_1, y_2 - X\hat{a}_2, \ldots, y_M - X\hat{a}_M).$$

[**Householder transformation and multiple integral**] Using definitions of matrices A and C, a useful multiple integral formula

$$(93) \qquad \int e^{-\|X a_1 - y_1\|^2/2} da_1 = \frac{(2\pi)^{K/2}}{|\det A|^2} e^{-|C_{11}|^2/2}$$

is obtained, where $\det A = \prod_{j=1}^{K} A_{jj}$. Note that $|C_{11}|^2$ is the minimum value of $\|X a_1 - y_1\|^2$.

B BAYSEA

A Bayesian seasonal adjustment program BAYSEA proposed by Akaike and Ishiguro(1985) decomposes the economic time series $\{y_i\}$ into the form

$$y_i = T_i + S_i + I_i \qquad (i = 1, 2, \ldots, N)$$

where T_i, S_i and I_i represent the "trend cycle", "seasonal" and "irregular" components respectively. When the length L of the seasonal cycle is specified, BAYSEA minimizes the cost

$$(94) \qquad \sum_{i=1}^{N}(y_i - T_i - S_i)^2 + u^2\sum_{i=1}^{N}(\Delta^k T_i)^2 + v^2\sum_{i=1}^{N}(\Delta_L^l S_i)^2 + w^2\sum_{i=1}^{N}(\sum_{j=0}^{L-1} S_{i-j})^2$$

in an attempt to get a result which approximately satisfies the conditions:

$$\Delta^k T_i = 0$$
$$\Delta_L^l S_i = 0$$
$$I_i = 0$$

and

$$\sum_{j=0}^{L-1} S_{i-j} = 0,$$

where the difference operators are defined by

$$\Delta T_i = T_i - T_{i-1}$$
$$\Delta_L S_i = S_i - S_{i-L}$$
$$\Delta^0 T_i = T_i, \qquad \Delta_L^0 S_i = S_i$$

and

$$\Delta^k T_i = \Delta(\Delta^{k-1} T_i), \qquad \Delta_L^k S_i = \Delta_L(\Delta_L^{k-1} S_i).$$

To choose weighting factors u^2, v^2 and w^2, a Bayesian interpretation of the method is employed. The minimization of the criterion (94) is equivalent to the maximization of the quantity

$$\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - T_i - S_i)^2\right\} \times \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}[u^2(\Delta^k T_i)^2 + v^2(\Delta_L^l S_i)^2 + w^2(\sum_{j=0}^{L-1} S_{i-j})^2]\right\},$$

which is then equivalent to the maximization of the product of the likelihood of the model for the data $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$

$$f(\boldsymbol{y} \mid \boldsymbol{\theta}, \sigma^2) = \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(y_i - T_i - S_i)^2\right\}$$

with its "prior distribution"

$$(95) \quad \pi(\boldsymbol{\theta} \mid k, l, u^2, v^2, w^2, \sigma^2) = C\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}[u^2(\Delta^k T_i)^2 + v^2(\Delta_L^l S_i)^2 + w^2(\sum_{j=0}^{L-1} S_{i-j})^2]\right\}$$

of the parameter

$$\boldsymbol{\theta} = (T_1, T_2, \ldots, T_N, S_1, S_2, \ldots, S_N)'$$

In this Bayesian framework, the choice of the weighting factor is the choice of the "hyper parameters" of the prior distribution (95). Akaike(1980) proposed the use of

$$(96) \qquad \text{ABIC} = -2\log\int f(y \mid \boldsymbol{\theta}, \sigma^2)\pi(\boldsymbol{\theta} \mid k, l, u^2, v^2, w^2, \sigma^2)d\boldsymbol{\theta}$$

for the choice. Hyper parameter set with smaller ABIC value is considered to be a better choice.

C Kalman Filter

Define notations
$$z_{t|j} = E\{z_t \mid y_1, \ldots, y_j\}$$

and
$$V_{t|j} = E\{(z_t - z_{t|j})(z_t - z_{t|j})^T \mid y_t, \ldots, y_j\}$$

for z_t and y_t of eq.(21) then the one step ahead prediction of state at time $t-1$ is calculated by

(97)
$$z_{t|t-1} = \Psi z_{t-1|t-1}$$

and

(98)
$$V_{t|t-1} = \Psi V_{t-1|t-1} \Psi^T + \Gamma U \Gamma^T,$$

where $U = E\{v_n v_n^T\}$. When observation y_t is obtained at time t, filtering estimate of the state is calculated by

(99)
$$Q_t = \Phi V_{t|t-1} \Phi^T + W$$

(100)
$$K_t = V_{t|t-1} \Phi^T Q_t^{-1}$$

(101)
$$\Delta y_t = y_t - \Phi z_{t|t-1}$$

(102)
$$z_{t|t} = z_{t|t-1} + K_t \Delta y_t$$

(103)
$$V_{t|t} = (I - K_t \Phi) V_{t|t-1}$$

(104)
$$\ell\ell_t = C - \frac{1}{2}(\log \det Q_t + \Delta y_t^T Q_t^{-1} \Delta y_t),$$

where $W = E\{r_n r_n^T\}$. When observations at $t = 1, 2, 3, \ldots, T$ are given the log likelihood of the model is given by

(105)
$$\ell\ell = \sum_{t=N_0+1}^{T} \ell\ell_t.$$

C depends on the dimension of the observation. If the dimension is K, $C = -\frac{K}{2} \log 2\pi$. There are two opposite policy about the setting of initial value. One is to set $V_{1|0} = O$ and estimate $z_{1|0}$ by maximizing the log likelihood. The opposite way is to set $V_{1|0}$ an effectively infinite one and fix $z_{1|0}$.

[**Missing observations**] When some, or possibly all, elements of y_t are missing, we need modify the above procedure. Let us assume that $i(k)-$th elements$(k = 1, 2, \ldots, r_t)$ of y_t are available. y_t in eq.(101), Φ in eqs. (99, 100, 101, 103) and W in eq.(99) are replaced by $M y_t$, $M\Phi$, MWM^T, respectively, where M is the $r_t \times K$ matrix whose k-$i(k)$ elements$(k = 1, 2, \ldots, r)$ are 1 and all other elements are zero.

D   DALL

Read "DALL" so that it rhymes with "call".
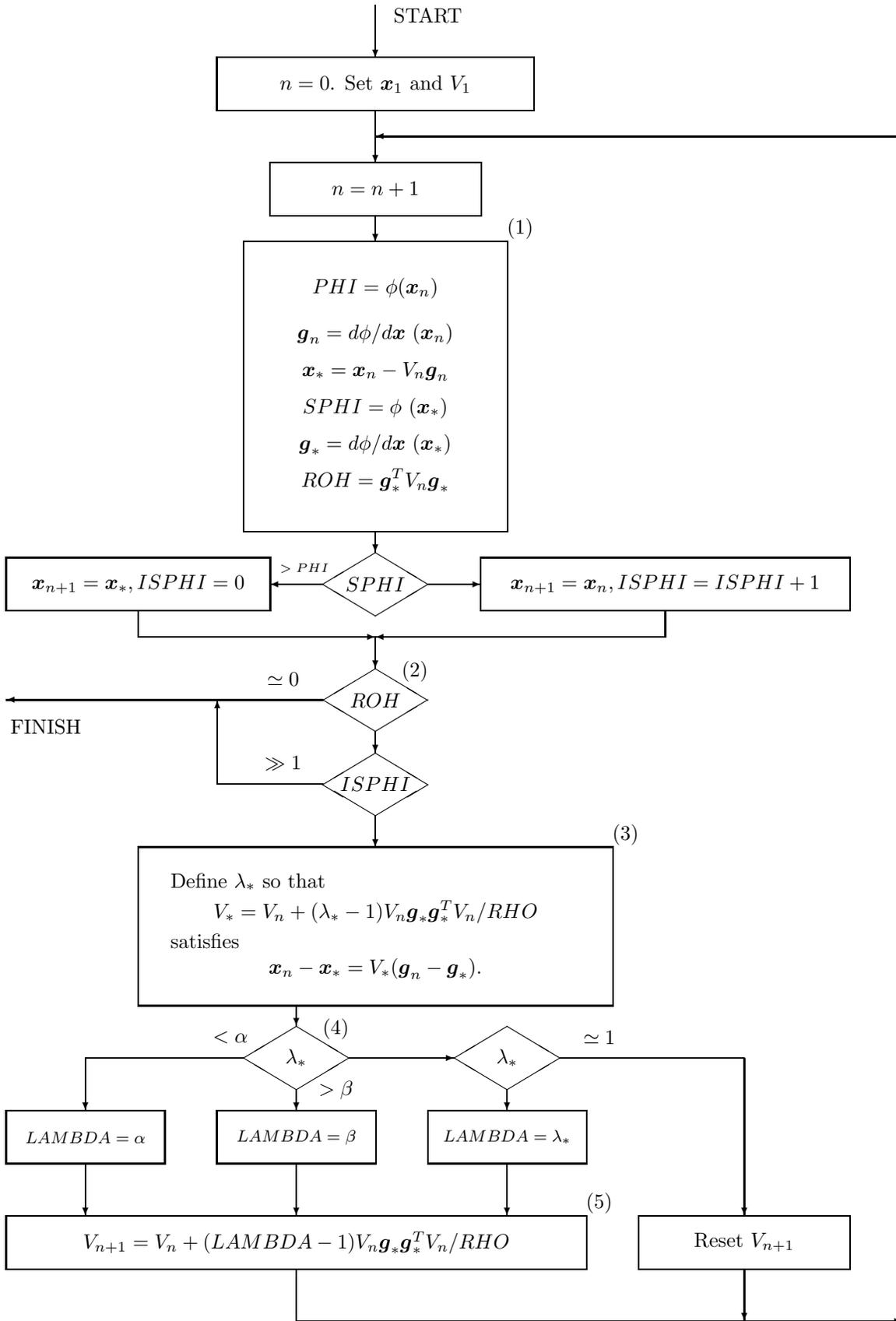
Figure 2. Flow chart of DALL

Assume that a function ϕ has the form defined by a negative definite matrix V_0^{-1}

(106)
$$\phi(x) = \phi_0 + \frac{1}{2}(x - x_0)^T V_0^{-1}(x - x_0),$$

then the relation between the gradient at a point x and the maximizing point is given by

(107)
$$g = V_0^{-1}(x - x_0).$$

That is, if the variance V_0 of the function (106) is known and the gradient at a point is known the maximizing point is readily calculated by

(108)
$$x - V_0 g.$$

On the other hand, if gradients at sufficient number of points are known V_0 can be estimated. Davidon's variance algorithm (Davidon, 1968) is constructed on these two facts.

DALL(Ishiguro & Akaike(1989))'s procedure is as follows

(1) The gradient is computed numerically by

(109)
$$g_i = \frac{\phi\,(x + \text{step}(i)\,e_i) - \phi\,(x - \text{step}(i)\,e_i)}{2\,\text{step}(i)} \quad (i = 1, 2, \ldots, IP)$$

where e_i is the unit vector of the i-th axis and IP is the dimension of the parameter vector θ. This computation can (and should, if possible) be executed parallely.

(2) The difference between $SPHI$ and the maximum value is given by

$$\frac{1}{2}(x_* - x_0)^T V_0^{-1}(x_* - x_0),$$

which is equal to

(110)
$$\frac{1}{2}g_*^T V_0 g_*.$$

RHO is the twice of an estimate of this value. It is implicitly assumed that the variance V_n is sufficiently close to V_0.

(3) Under the present assumptions, if $V_n = V_0$ then $g_* = 0$ and the procedure ends. The fact that $g_* \neq 0$ is an evidence of $V_n \neq V_0$. If (106) is the correct form, V_0 satisfies

(111)
$$x_n - x_* = V_0(g_n - g_*).$$

Define $\gamma = -g_n^T V_n g_*/RHO$, then $\lambda_* = |\gamma/(\gamma + 1)|$.

(4) If $LAMBDA = 1$, V_n is not improved. If $LAMBDA = 1$ is far from 1 the modification might be too drastic, especially if the assumed form (106) of the function is dubious. Now α and β are set at 0.25 and 4.0, respectively.

(5) Generally, the shape of a K-dimensional matrix A is known by calculating $\{Ab_1, Ab_2, Ab_3, \ldots, Ab_K\}$, where $\{b_1, b_2, b_3, \ldots, b_K\}$ is an orthogonal system of K vectors. If we choose the basis so that $b_1 = V_n g_*$, we have relations

(112)
$$V_{n+1} b_j = V_n b_j \quad (j = 2, \ldots, K).$$

E Optimal Control Design

Let us start from an ARX model (25). Let $z_t^l (l = 1, 2, \ldots, M)$ be defined by

$$z_t^l = \sum_{m=L+1}^{M} A_m^{yy} y_{t+l-m} + \sum_{m=L+1}^{M} A_m^{yw} w_{t+l-m} \quad (l = 1, 2, \ldots, M-1)$$

$$z_{t-1}^M = 0$$

then we have

(113)
$$\begin{pmatrix} y_t \\ z_t^1 \\ \vdots \\ z_t^{M-1} \end{pmatrix} = \begin{pmatrix} A_1^{yy} & I & & \\ A_2^{yy} & & \ddots & \\ \vdots & & & I \\ A_M^{yy} & 0 & \cdots & 0 \end{pmatrix}$$

$$\begin{pmatrix} y_{t-1} \\ z_{t-1}^1 \\ \vdots \\ z_{t-1}^{M-1} \end{pmatrix} + \begin{pmatrix} A_1^{yw} \\ A_2^{yw} \\ \vdots \\ A_M^{yw} \end{pmatrix} w_{t-1} + \begin{pmatrix} \zeta_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Define

$$z_t = (y_t^T, z_t^{1^T}, \ldots, z_t^{M-1^T})^T$$

$$u_t = (\zeta_t^T, 0^T, 0^T, \ldots, 0^T)^T$$

and rewrite this equation to:

(114)
$$z_t = \Psi z_{t-1} + \Gamma w_{t-1} + u_t$$

$$y_t = \Phi z_t.$$

Definitions of Ψ, Γ, Φ etc. will be clear from eq.(114).

Let us define the cost of control by

(115) $\quad J = \sum_{t=1}^{N} J_t$

(116) $\quad J_t = E_t \left\{ \begin{pmatrix} z_{t+1} \\ w_t \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} z_{t+1} \\ w_t \end{pmatrix} + (z_{t+1} - z_t)^T Q^\Delta (z_{t+1} - z_t) \right.$

$$\left. + (w_t - w_{t-1})^T R^\Delta (w_t - w_{t-1}) \right\},$$

where Q^Δ, R^Δ and

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}$$

are positive definite. E_t denotes the expectation with respect to z_{t+1} given $w_t, w_{t-1} and z_t$. With the dynamic programming method, the optimal input to $\{w_t\}$ the system(114) to minimize the cost(115) is obtained(Windnall(1968)).

Defining A, B, H, g and f by

(117) $\quad A = -(\Gamma^T (Q + Q^\Delta)\Psi + S^T \Psi - \Gamma^T Q^\Delta)$

(118) $\quad B = \Psi^T Q \Psi + (\Psi - I)^T Q^\Delta (\Psi - I)$

(119) $\quad H = \Gamma^T (Q + Q^\Delta)\Gamma + \Gamma^T S + S^T T\Gamma + R + R^\Delta$

(120) $\quad g = Az_t + R^\Delta x_{t-1}$

$$f = z_t^T B z_t + w_{t-1}^T R^\Delta w_{t-1} + E\left\{ u_{t+1}^T (Q + Q^\Delta) u_{t+1} \right\}$$

Eq.(116) is rewritten into the form

$$(121) \qquad \begin{aligned} J_t &= \boldsymbol{w}_t^T H \boldsymbol{w}_t - 2\boldsymbol{g}^T \boldsymbol{w}_t + f \\ &= (\boldsymbol{w}_t - H^{-1}\boldsymbol{g})^T H (\boldsymbol{w}_t - H^{-1}\boldsymbol{g}) + f - \boldsymbol{g}^T H^{-1} \boldsymbol{g} \end{aligned}$$

Using eq.(121), the optimal $\boldsymbol{w}_t$ is given by

$$(122) \qquad \begin{aligned} \boldsymbol{w}_t = H^{-1}\boldsymbol{g} &= H^{-1} A \boldsymbol{z}_t + H^{-1} R^\Delta \boldsymbol{w}_{t-1} \\ &\equiv G^{T-t} \boldsymbol{z}_t + D^{T-t} \boldsymbol{w}_{t-1} \end{aligned}$$

and the minimum value of the cost(121) is

$$(123) \qquad \hat{J}_t = f - \boldsymbol{g}^T H^{-1} \boldsymbol{g}.$$

(G^{N-t}, D^{N-t}) gives a control law to calculate optimal $\boldsymbol{w}_t$ from $\boldsymbol{z}_t$ and $\boldsymbol{w}_{t-1}$.

Setting $t = N$ in $\hat{J}_t$ Substitute $\hat{J}_N$ for J_N in J to get,

$$(124) \quad \begin{aligned} J &= \begin{pmatrix} \boldsymbol{z}_N \\ \boldsymbol{w}_{N-1} \end{pmatrix}^T \begin{pmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{z}_N \\ \boldsymbol{w}_{N-1} \end{pmatrix} + (\boldsymbol{z}_N - \boldsymbol{z}_{N-1})^T Q^\Delta (\boldsymbol{z}_N - \boldsymbol{z}_{N-1}) \\ &\quad + (\boldsymbol{w}_N - \boldsymbol{w}_{N-2})^T R^\Delta (\boldsymbol{w}_{N-1} - \boldsymbol{w}_{N-2}) + \sum_{t=1}^{N-2} J_t. \end{aligned}$$

This equation is obtained by replacing the first term $\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}$ of eq.(116) by

$$(125) \qquad \begin{aligned} \begin{pmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{pmatrix} &= \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \\ &\quad + \begin{bmatrix} B - A^T H^{-1} A & -A^T H^{-1} R^\Delta \\ -R^\Delta H^{-1} A & R^\Delta - RDH^{-1}R^\Delta \end{bmatrix}, \end{aligned}$$

and N by $N-1$. Repeatedly using eqs.(125), (117) and (118), we get a series of control laws $(G^1, D^1), (G^2, D^2), \ldots$, and (G^N, D^N). Let us assume that this control law converges to a limit as N tends to infinity. Let $\boldsymbol{w}_t$ be computed by $\boldsymbol{w}_t = G^\infty Z_t + D^\infty W_{t-1}$. It is possible to eliminate $\boldsymbol{z}_t$ to get a equation which contains y and w. Let $G^\infty = (G_1, G_2, \ldots, G_M)$, then we have

$$(126) \qquad \boldsymbol{w}_0 = G_1 \boldsymbol{y}_t + \sum_{L=2}^{M} \sum_{k=L}^{M} G_L (A_k^{yy}, A_k^{yw}) \begin{pmatrix} \boldsymbol{y}_{t+L-1-k} \\ \boldsymbol{w}_{t+L-1-k} \end{pmatrix} + D^0 \boldsymbol{w}_{t-1}.$$

Then introducing $L = k - m + 1$ we get

$$\boldsymbol{w} = \sum_{m=1}^{M} \left\{ \sum_{k=m}^{M} G_{k+1-m}(A_k^{yy}, A_k^{yw}) \right\} \begin{pmatrix} \boldsymbol{y}_{t-m} \\ \boldsymbol{w}_{t-m} \end{pmatrix} + G_1 \boldsymbol{\zeta}_t + D^0 \boldsymbol{w}_{t-1}.$$

Define

$$(127) \qquad \begin{aligned} A_m^{wy} &= \sum_{k=m}^{M} G_{k+1-m} A_k^{yy} \\ A_m^{ww} &= \begin{cases} \sum_{k=1}^{M} G_k A_k^{yw} + D^0 & (m = 1) \\ \sum_{k=m}^{M} G_{k+1-m} A_k^{yw} & (m = 2, \ldots, M) \end{cases} \end{aligned}$$

then the dynamics of the whole system under this control is described by eq.(27).

F   Debugging Tool, Bug-Pack

In debugging process it is desirable to have detailed output only when the bug is just coming. This is possible if

1. we can write commands in the output record of the program under concern, and

2. let the program read the modified record.

Bug-pack is a subroutine package to realize this operation. A minimum tutorial course on the use of bug subroutine is given here. The full manual for bug-pack routines will be given elsewhere. The generic form of 'bug' call is

```
   call  bug (
 i Lid , J , J1 , J2 , Cid , Iid , Rid ,
 o Message )
```

1. Find the following part in the main program.

```
        call bug (1,0,0,0,'start',0,0.d0,message)
        call bug (1,0,0,0,'tutorial',511,1999.d0,message)
        if( message .eq. 123 ) write(6,*) 'message is received'
        call sbuggle (jewel)
        if( jewel .eq. 456 ) write(6,*) 'message can be smuggled in'
        do 1 j = 1,10
           call bug (2,j,7,8,'peekaboo',j,0.d0, message )
           call bug (1,j,10,10,'hello',0,0.d0,message)
    1   continue
```

2. Change the first 1 in the first line to 0.

```
        call bug (0,0,0,0,'start',0,0.d0,message)
```

3. Re-compile and run the program. Record the output in a file.

4. The prompt,

```
 start: Bug ? (<Y>es / with <M>ap / <N>o bug )
```

is given. Answer this by 'y' and continue the following computation as usual.

   - 'bug' routine becomes active only after a call with Lid = 0.
   - You will find other bug calls are also activated and make their outputs.

5. Name the output record file 'bug.map'.

6. Edit 'bug.map'. There should be lines,

```
  COM: 0:LOOK
  COM: 0:LETITGO       -10:
  bug.map command list
  LOOK      LETITGO  MESSAGE  LEVEL     BACK      QUIT
  SKIP      DUMMY
  BUG: 0: 1:start    :    1:        0:   .00000000000000000000000000000000000D+00:
  BUG: 0: 1:tutorial:    2:      511:   .19990000000000000000000000000000000D+04:
  BUG: 0: 1:hello   :    3:        0:   .00000000000000000000000000000000000D+00:
```

   - If your machine is a parallel computer, every machine will make print outs.
   - If you are interested in the performance of the machine of 'rank' j, replace every ' COM: 0:' by ' COM: j:' in the bug.map.

Find the line,

```
 BUG: 0: 1:hello   :    3:        0:   .00000000000000000000000000000000000D+00:
```

and add a command line so that you have

```
 COM: 0:QUIT
 BUG: 0: 1:hello   :    3:        0:   .00000000000000000000000000000000000D+00:
```

7. Run the program and answer the prompt,

```
start: Bug ? (<Y>es / with <M>ap / <N>o bug )
```

by 'm' and see what happens.

- This operation is "to start bug with a map".
- Try another answer 'n' sometime.

8. When you start bug with the above map, the execution of the program stops after printing

```
BUG: 0: 1:start   !   1:        0:  .0000000000000000000000000000000000D+00:
BUG: 0: 1:tutorial!   2:      511:  .1999000000000000000000000000000000D+04:
COM: 0:QUIT
```

- 'bug' accepts 'QUIT' command.

9. Edit again 'bug.map' so that you have

```
BUG: 0: 1:start   :   1:        0:  .0000000000000000000000000000000000D+00:
COM: 0:MESSAGE       123
BUG: 0: 1:tutorial:   2:      511:  .1999000000000000000000000000000000D+04:
COM: 0:QUIT
BUG: 0: 1:hello    :   3:        0:  .0000000000000000000000000000000000D+00:
```

run the program, start bug with the map, and you will have

```
BUG: 0: 1:start   !   1:        0:  .0000000000000000000000000000000000D+00:
COM: 0:MESSAGE       123:
BUG: 0: 1:tutorial:   2:      511:  .1999000000000000000000000000000000D+04:
message is received
COM: 0:QUIT
```

- 'bug' accepts 'MESSAGE'.

10. Edit again 'bug.map' and change the line

```
COM: 0:MESSAGE       123
```

to

```
COM: 0:MESSAGE       456
```

and run the program, start bug with the map, then you will have

```
BUG: 0: 1:start   !   1:        0:  .0000000000000000000000000000000000D+00:
COM: 0:MESSAGE       456:
BUG: 0: 1:tutorial!   2:      511:  .1999000000000000000000000000000000D+04:
message can be smuggled in
COM: 0:QUIT
```

- 'MESSAGE' can be smuggled in.

11. Edit again 'bug.map' so that you have

```
BUG: 0: 1:start   :   1:        0:  .0000000000000000000000000000000000D+00:
COM: 0:LEVEL          2
COM: 0:MESSAGE       456
BUG: 0: 1:tutorial:   2:      511:  .1999000000000000000000000000000000D+04:
COM: 0:QUIT
BUG: 0: 1:hello    :   3:        0:  .0000000000000000000000000000000000D+00:
```

run the program, start bug with the map, and you will have

```
BUG: 0: 1:start   !   1:        0:  .0000000000000000000000000000000000D+00:
MAP: 0: 2:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
COM: 0:MESSAGE       456:
BUG: 0: 1:tutorial!   2:      511:  .1999000000000000000000000000000000D+04:
message can be smuggled in
BUG: 0: 2:peekaboo:   3:        7:  .0000000000000000000000000000000000D+00:
BUG: 0: 2:peekaboo:   4:        8:  .0000000000000000000000000000000000D+00:
COM: 0:QUIT
```

- Higher level 'bug' call with higher level 'Lidy' value is activated only after receiving an appropriate 'LEVEL' command.

- 'bug' is active only when the arguments 'J', 'J1' and 'J2' satisfies the relation $J1 \leq J \leq J2$.

12. Edit 'bug.map' and change the line

```
 BUG: 0: 1:tutorial:    2:         511:   .19990000000000000000000000000000000D+04:
```

to

```
 BUG: 0: 1:tutorial:    2:         511:   .20000000000000000000000000000000000D+04:
```

run the program, start bug with the map, and you will have

```
 BUG: 0: 1:start    !   1:          0:   .00000000000000000000000000000000000D+00:
 MAP: 0: 2::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 COM: 0:MESSAGE      456:
 BUG: 0: 1:tutorial!    2:         511:   .19990000000000000000000000000000000D+04:
 mismatch!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 OLD: 0: 1:tutorial:    2:         511:   .20000000000000000000000000000000000D+04:
 DEB: 0: ratio = (new_rid-old_rid)/old_rid :   .500D-03
 DEB: 0:                                  letitgo :      -10
 DEB: 0:QUIT on the mismatch
```

- If a 'bug.map' record does not match, bug can stop the execution of the program.

- With this function, 'bug' can help program modifier, who is trying to get an efficient procedure not changing the results.

13. Edit 'bug.map' and change the line

```
 COM: 0:LETITGO      -10:
```

to make the 'bug.map'

```
 COM: 0:LOOK
 COM: 0:LETITGO      -3:
 bug.map command list
 LOOK      LETITGO  MESSAGE  LEVEL     BACK      QUIT
 SKIP      DUMMY
 BUG: 0: 1:start    :   1:          0:   .00000000000000000000000000000000000D+00:
 COM: 0:LEVEL        2
 COM: 0:MESSAGE      456
 BUG: 0: 1:tutorial:    2:         511:   .20000000000000000000000000000000000D+04:
 COM: 0:QUIT
 BUG: 0: 1:hello    :   3:          0:   .00000000000000000000000000000000000D+00:
```

run the program, start bug with the map, and you will have

```
 COM: 0:LOOK
 COM: 0:LETITGO      -3:
 BUG: 0: 1:start    !   1:          0:   .00000000000000000000000000000000000D+00:
 MAP: 0: 2::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
 COM: 0:MESSAGE      456:
 BUG: 0: 1:tutorial!    2:         511:   .19990000000000000000000000000000000D+04:
 mismatch!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 OLD: 0: 1:tutorial:    2:         511:   .20000000000000000000000000000000000D+04:
 DEB: 0: ratio = (new_rid-old_rid)/old_rid :   .500D-03
 DEB: 0:                                  letitgo :      -3
 message can be smuggled in
 BUG: 0: 2:peekaboo:   3:          7:   .00000000000000000000000000000000000D+00:
 BUG: 0: 2:peekaboo:   4:          8:   .00000000000000000000000000000000000D+00:
 COM: 0:QUIT
```

- Even if 'bug.map' records do not match, bug can allow the continued execution of the program by relaxing the 'LETITGO' value.