IN-35
97237

NASA Technical Memorandum 109039

FREQUENCY DOMAIN IDENTIFICATION TOOLBOX

Lucas G. Horta and **Jer-Nan Juang**
NASA Langley Research Center
Hampton, Virginia 23681-0001

Chung-Wen Chen
North Carolina State University
Raleigh, NC 27695-7910

September 1996

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia

INTRODUCTION

Identification of systems, in most laboratory environments, is performed using spectrum analyzers and a skilled group of engineers and technicians. Classical identification of linear systems for model verification and control design is commonly performed using concepts from spectral analysis. The speed of computers and implementation of fast Fourier transforms algorithms have facilitated manipulation of large sets of data. Once data is acquired, frequency response functions are manipulated to obtain information about the tested system. In many cases, knowledge of the fundamental frequencies of the system is enough information but there are times when mathematical models of the system input/output are needed. Our goal is to provide the user with a convenient and simple to use package to analyzed data given in terms of frequency responses and/or spectral matrices.

Linear time-invariant systems are completely characterized by their impulse response functions or in the case of discrete-time analysis by their pulse responses. If the only information about the system is given in terms of a set of pulse responses, this information must be converted into a compact parametric form for use in analyses. Curve fitting algorithms have been used extensively for this purpose, where a particular model structure is selected and the parameters are evaluated by minimizing the error between the model and estimated pulse responses. A two step approach for the identification of state space models is used in this work. First, frequency data in the form of frequency responses and/or spectral matrices are fitted with a model in matrix polynomial form, and second, smoothed pulse responses computed from the polynomial parameters are used with realization theory for order determination and a state-space realization. One advantage of this approach is the ability to recover state space models from sections of a transfer function with minimum window distortions. Another advantage is the ability to concatenate frequency response functions obtained from multiple tests to recover a single state space model. Cases with different frequency resolution are combined easily.

System identification in general requires knowledge about the system and algorithms being used. The algorithms incorporated in this package are no different. Many parameters are fixed to make it convenient for the casual user to obtain results quickly, however, selected parameters may not be adequate for all possible cases. Exploring the different algorithms and understanding their limitations will help the user get the most out of this package. Figure 1 shows a simple diagram with data flow and main function calls. All intermediate calls are automated in the functions *okid_fqm* and *okid_asf*. The casual user is encourage to use one of these two functions.

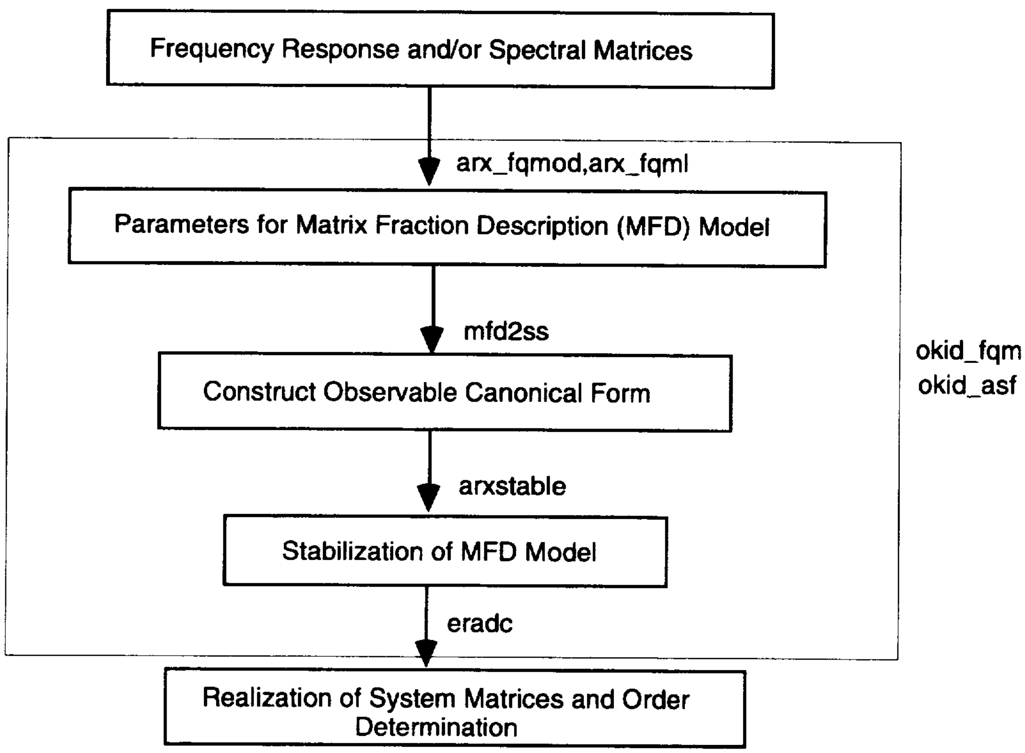


Fig. 1 Road map for identification of state space models from frequency response data

SUBROUTINE LIST

arxstable

Purpose:
Computes a stable state space model from an unstable discrete time model.

Synopsis:

[*A*,*B*,*C*,*D*]=arxstable(*Ao*,*Bo*,*Co*,*Do*,*f*,*dt*,*P*)

Description:
When identifying models from experimental data using parametric approaches, unstable modes appear quite often. This function computes a stable model from an unstable model [*Ao*,*Bo*,*Co*,*Do*] such that their frequency responses are similar. For the unstable system, the frequency response compared is that of the anti-causal system. The model is separated into stable and unstable sub-systems. The unstable sub-system poles are inverted and assigned to the new stabilized model. Unstable poles are assumed to appear in pairs and the desired characteristic polynomial is given by

$$p(z) = \prod_{i=1}^{k}\left(1 - \frac{1}{z_i}z^{-1}\right)\left(1 - \frac{1}{\bar{z}_i}z^{-1}\right)$$

where z_i and $\bar{z}_i$ are unstable pole pairs. A stable representation of the unstable sub-system is written as

$$G_d(z) = \frac{K(z)}{p(z)}$$

Since $G_d(z)$ and $p(z)$ are both known, a least squares problem is formulated to determine the polynomial matrix $K(z)$. With the solution for $K(z)$ one can realized a stable state space model for the unstable part and the append it to the stable portion of the identified model.

The input parameters are the unstable state space model [*Ao*,*Bo*,*Co*,*Do*], a frequency vector *f* in units of Hertz, sample time is defined by *dt*, and the parameter *P* when multiplied by the number of outputs equals the maximum system order.

Algorithm:

See Ref. 1.

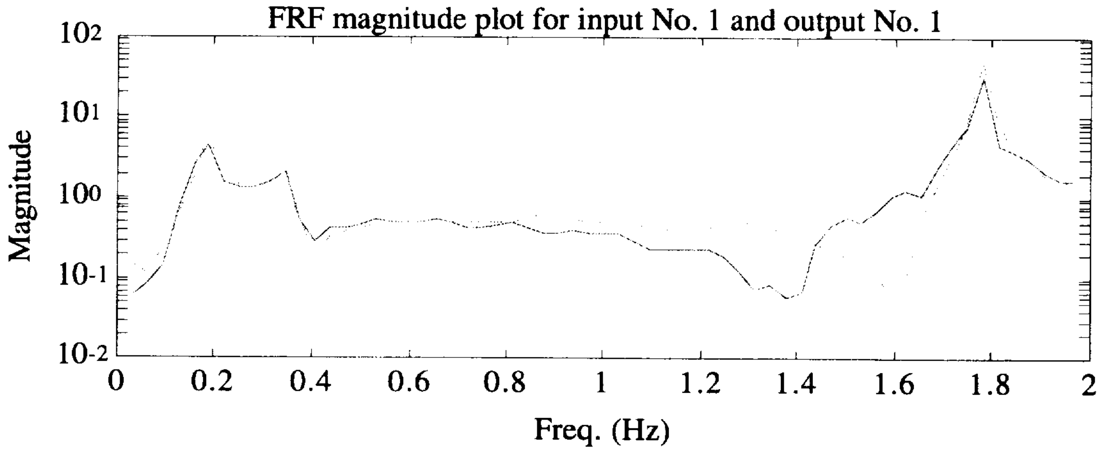
Example:

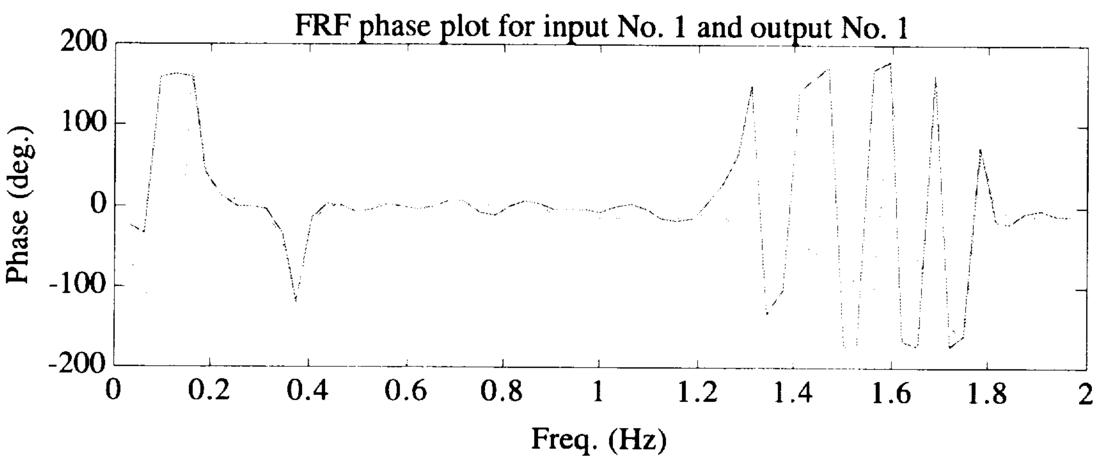
```
load cemdata
[ntot,junk]=size(YU_cross);
r=1;
m=1;
%
%     Skipping frequency points
%
Vs=[1:64];
Gz=Y_frf(Vs,1);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot));
P=20;
[Az,Bz]=arx_fqmod(Gz,f,dt,r,m,P);              % Compute MFD Model
[A,B,C,D]=mfd2ss(Az,Bz);                       % Construct Canonical Form
disp(' Unstable Discrete Eigenvalues')
disp(abs(eig(A)))                              % Poles of unstable model
[As,Bs,Cs,Ds]=arxstable(A,B,C,D,f,dt,P);       % Stable State Space  Model
[Gz_id]=frfm(As,Bs,Cs,Ds,Gz,f,dt,ntot,1);      % Compare solution

 Unstable Discrete Eigenvalues
   1.1539e+00
   1.1539e+00
   1.0678e+00
   1.0678e+00
   9.7717e-01
   9.7717e-01
   9.9158e-01
   9.9158e-01
   1.1182e+00
   1.1182e+00
   8.3122e-01
   8.3122e-01
   4.8006e-01
   1.0923e+00
   1.0923e+00
   9.9254e-01
   9.9254e-01
   8.4916e-01
   8.4916e-01
   9.8934e-01
No. of eigenvalues: 20
No. of unstable eigenvalues: 8
No. of real unstable poles: 0
No. of complex unstable poles: 8
Computing Stabilizing Part: Order should be =< 8
ERADC is used now.
The Hankel matrix size for ERADC is 18 by 194.
```

Maximum Hankel singular value = 5.900084e-01
Minimum Hankel singular value = 2.684231e-07

Damping(%)	Freq(HZ)	Mode SV
4.8630e+00	8.9103e-01	3.5471e-01
4.8630e+00	8.9103e-01	3.5471e-01
5.0898e+00	1.2759e+00	4.2925e-01
5.0898e+00	1.2759e+00	4.2925e-01
3.7805e+00	1.6341e+00	6.5384e-01
3.7805e+00	1.6341e+00	6.5384e-01
2.7379e+00	1.6924e+00	1.0000e+00
2.7379e+00	1.6924e+00	1.0000e+00

Time (min) to reconstruct FRF 0.005983





Reference:

1) Chen, C.-W., Juang J.-N., and Lee, G., "Stable State Space System Identification from Frequency Domain Data," Proceedings of the first IEEE Regional Conference on Aerospace Control Systems, CA., May 25-27, 1993.

arx_fqmod

Purpose:

Computes the left matrix fraction description model given a frequency response function.

Synopsis:

[*Az*,*Bz*]=arx_fqmod(*Gz*,*f*,*dt*,*r*,*m*,*P*);

Description:

The function estimates the denominator and numerator matrix of a frequency response function *Gz*. A left matrix fraction description $Gz = A^{-1}(z)B(z)$ is used to represent the system. The subroutine input matrix *Gz* contains the system frequency response stacked columnwise. For example, a system with *r* inputs and *m* outputs has *Gz* constructed as follows

$$Gz = \begin{bmatrix} g_{11}(\omega_0) & \cdots & g_{m1}(\omega_0) & g_{12}(\omega_0) & \cdots & g_{m2}(\omega_0) & \cdots & g_{1r}(\omega_0) & \cdots & g_{mr}(\omega_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ g_{11}(\omega_f) & \cdots & g_{m1}(\omega_f) & g_{12}(\omega_f) & \cdots & g_{m2}(\omega_f) & \cdots & g_{1r}(\omega_f) & \cdots & g_{mr}(\omega_f) \end{bmatrix}$$

where the starting frequency is ω_0 and the final frequency is ω_f. A corresponding frequency vector is defined f with units of Hertz, sample time is defined by *dt*, and the parameter *P* is the order of the matrix polynomial. The outputs *Az* and *Bz* are the parameters of matrix fraction description *A(z)* and *B(z)*, where

$$A(z) = I + A_1 z^{-1} + \cdots A_p z^{-p} \qquad A_i \in R^{m\times m}$$
$$B(z) = B_0 + B_1 z^{-1} + \cdots B_p z^{-p} \qquad B_i \in R^{m\times r}$$

and $Az = -\begin{bmatrix} A_1 & A_2 & \cdots & A_p \end{bmatrix}^T$, $Bz = \begin{bmatrix} B_0 & B_1 & \cdots & B_p \end{bmatrix}^T$. Both evenly spaced and unevenly spaced frequency response functions can be analyzed.

Algorithm:

A linear least squares problem is formulated, see Ref. 1, and solved using singular value decomposition.

Example:

```
load cemdata
[ntot,junk]=size(YU_cross);
r=1; m=1;
%
%     Skipping frequency points
%
Vs=[1:64 65:2:640];
Gz=Y_frf(Vs,1);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot)); P=20;
[Az,Bz]=arx_fqmod(Gz,f,dt,r,m,P);
Az =2.6818e-01
   9.0337e-02
   7.8337e-01
  -2.4974e-01
   3.9861e-01
  -5.4586e-01
  -3.7194e-01
  -1.1294e-01
   2.6211e-01
  -2.7183e-01
   4.4990e-03
   5.7088e-01
   1.6681e-01
   8.4620e-02
  -6.1663e-02
  -4.7101e-01
  -4.3275e-01
  -2.2067e-03
   6.2649e-01
   2.0486e-01
Bz = 2.1243e+00
  -1.7012e+00
   5.6716e-04
  -1.6684e+00
   1.0146e+00
  -2.3422e-01
   3.5209e-01
   5.6924e-01
   2.0307e-01
  -9.6461e-02
  -8.5128e-02
   5.0896e-01
  -9.9311e-01
  -3.6706e-02
  -1.7887e-01
  -6.0550e-01
   2.0412e-01
   8.8424e-01
   1.0726e+00
  -4.7213e-01
  -8.0487e-01
```

invgz

Purpose:
Computes a pulse response function from a given frequency response.

Synopsis:

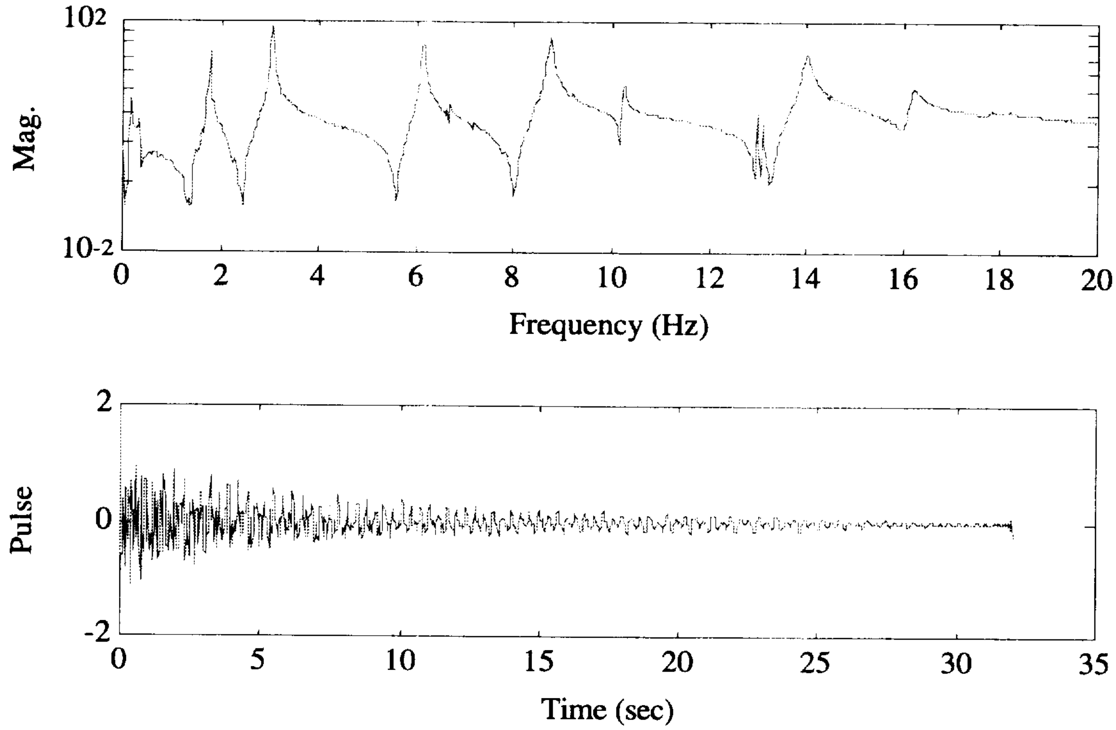
H=invgz(*Gz,m,dt*);

Description:
Given a frequency response function in the matrix *Gz*, stacked as described in the function *arx_fqmod*, the routine computes a pulse response using *ifft*. Before calling *ifft* one must append a mirror image of the frequency response in accordance with the *ifft* function format. The parameter *m* corresponds to the number of outputs and *dt* is the sample time in seconds. Output matrix *H* is stacked by columns.

Algorithm:
MATLAB *ifft* function is called to compute the pulse response.

Example:
The example loads data from the file *cemdata* and proceeds to compute the pulse response using *invgz*. Also shown is the magnitude plot of the frequency response being converted and a plot of the corresponding pulse response.

```
load cemdata
[ntot,junk]=size(YU_cross);
r=1;
m=1;
Vs=[1:640];
Gz=Y_frf(Vs,1);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot));
subplot(211),semilogy(f,abs(Gz))
xlabel('Frequency (Hz)')
ylabel('Mag. ')
H=invgz(Gz,m,dt);
ntot=ntot*2-1;
TIME=dt*[0:ntot-1]';
subplot(212),plot(TIME,H)
xlabel('Time (sec)')
ylabel('Pulse')
```


10^2
Mag.
10^-2
0
2
4
6
8
10
12
14
16
18
20
Frequency (Hz)
2
Pulse
0
-2
0
5
10
15
20
25
30
35
Time (sec)

frfm

Purpose:

Plots a given frequency response function with that obtained from a state space model.

Synopsis:

[*Gz_id*]=frfm(*A,B,C,D,Gz,f,dt,np,flag*);

Description:

This function plots the frequency response of the system given in the discrete state space model [*A,B,C,D*] versus *Gz*. The subroutine input parameters are the frequency vector *f* in units of Hertz, sample time is defined by *dt*, number of frequency points to plot *np*, and the parameter *flag* turns on/off plotting for batch jobs. On output, the frequency response function computed from [*A,B,C,D*] is returned in the variable *Gz_id*.

Algorithm:

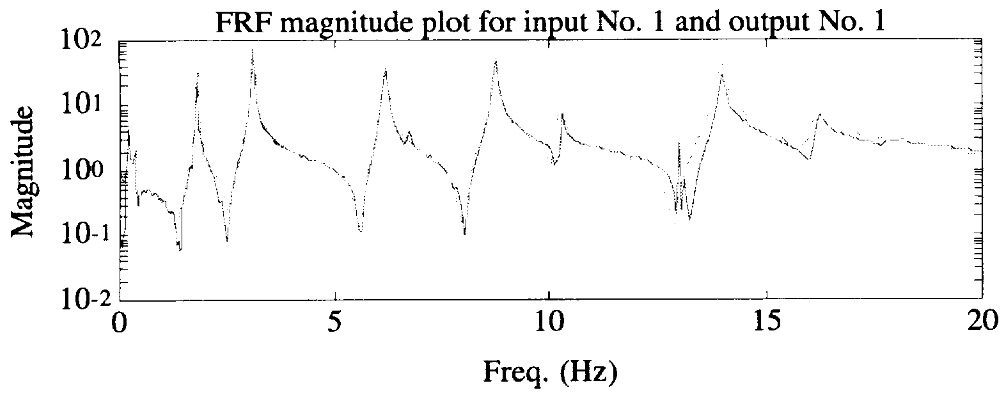
MATLAB *freqresp* function is called to compute the frequency response function for the state space model.

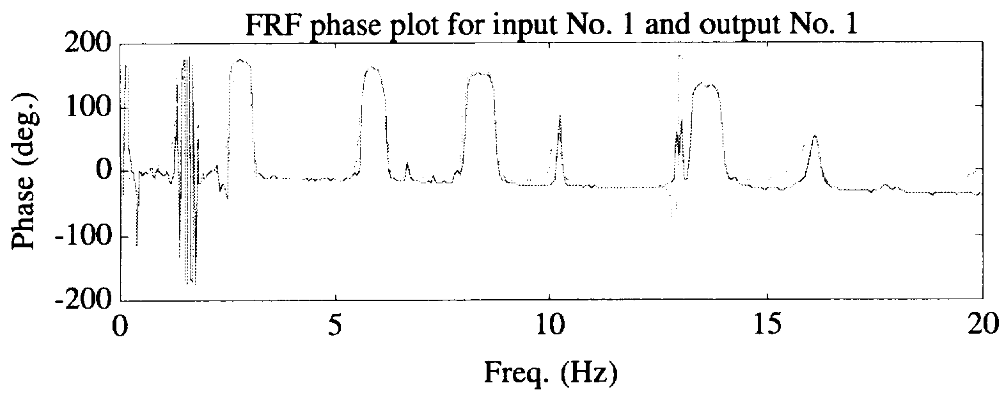
Example:

The example loads data from the file *cemdata* and proceeds to compute the state space model using the function *okid_fqm*. Results from this analysis are then plotted using *frfm*.

```
load cemdata
[ntot,junk]=size(YU_cross);
r=1;
m=1;
%
%     Skipping frequency points
%
Vs=[1:64 65:2:640];
Gz=Y_frf(Vs,1);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot));
```

```
P=20;
[A,B,C,D,Az,Bz]=okid_fqm(r,f,dt,Gz,'batch',P);
[Gz_id]=frfm(A,B,C,D,Gz,f,dt,ntot,1);

batch
 Total number of sample  points = 352
 Number of inputs = 1
 Number of outputs = 1
 Corresponding sampling rate = 39.87 Hz
 Number of desired Markov parameters = 81
No. of eigenvalues: 20
 Time (min) to compute ARX parameters 0.1486
ERADC is used now.
The Hankel matrix size for ERADC is 20 by 60.
Maximum Hankel singular value = 1.366719e+02
Minimum Hankel singular value = 2.714249e-03
  Damping(%)   Freq(HZ)     Mode SV
  1.0000e+02  4.9299e-02  5.7064e-02
  4.9193e+00  1.8667e+01  7.6341e-02
  4.9193e+00  1.8667e+01  7.6341e-02
  3.2112e+00  1.3633e+01  8.0297e-02
  3.2112e+00  1.3633e+01  8.0297e-02
  3.2082e+01  2.1045e+01  1.0955e-01
  3.9617e-01  1.0185e+01  2.6634e-01
  3.9617e-01  1.0185e+01  2.6634e-01
  7.1932e-01  1.6213e+01  2.7279e-01
  7.1932e-01  1.6213e+01  2.7279e-01
  2.9492e-01  6.1385e+00  6.2107e-01
  2.9492e-01  6.1385e+00  6.2107e-01
  4.3469e-01  1.3983e+01  6.2689e-01
  4.3469e-01  1.3983e+01  6.2689e-01
  2.9853e-01  8.7272e+00  7.1157e-01
  2.9853e-01  8.7272e+00  7.1157e-01
  2.0720e-01  1.7783e+00  8.1915e-01
  2.0720e-01  1.7783e+00  8.1915e-01
  2.4028e-01  3.0515e+00  1.0000e+00
  2.4028e-01  3.0515e+00  1.0000e+00
Time (min) to compute the system model 0.03212
Time (min) to reconstruct FRF 0.02942
```





See also:

okid_fqm, frf

frf2ss

Purpose:

Identification of a state space model from a frequency response function.

Synopsis:

[*A,B,C,D,dt*]=frf2ss(*Gz,df,m,norder,npeaks*)

Description:

This routine computes the inverse Fourier Transform of a given frequency response function to determine the system pulse response. Realization theory is used to compute a state space model from a given pulse response. Order determination can be performed interactively after examining the singular values of the Hankel matrix and/or fixed using the input parameter *norder*. The subroutine input parameters are the frequency response function *Gz* (stacked by columns), frequency resolution *df*, number of outputs *m*, desired state space model order *norder*, and an estimated number of observed peaks in the frequency response function *npeaks*. When the system order is not known a priori, the parameter *npeaks* is used to set the dimensions of the Hankel matrix and the system order must be selected interactively. However, if *norder* is not zero the realized system order is equal to *norder*.

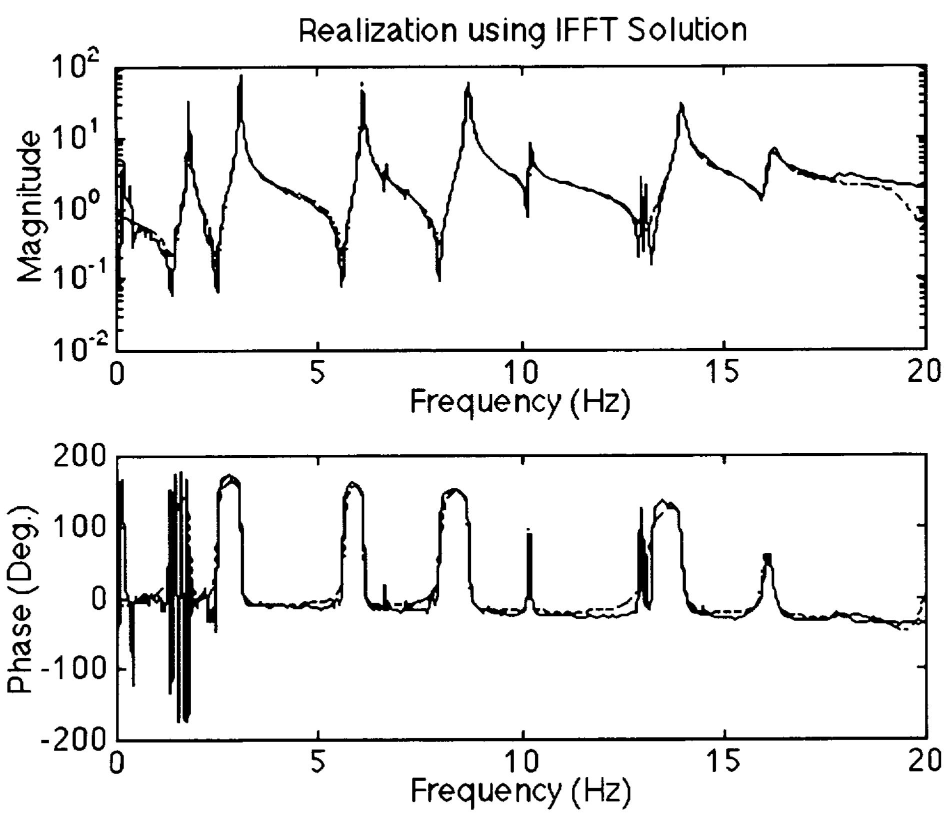
Algorithm:

Matlab routine *ifft* is used in conjunction with the SOCIT routine *eradc*.

Example:

```
load cemdata
[ntot,junk]=size(YU_cross);
f=fhz;
df=fhz(2)-fhz(1);
P=20;m=1;
Gz=Y_frf(1:640,1);
[A,B,C,D,dt]=frf2ss(Gz,df,m,P*m,5);
ERADC is used now.
The Hankel matrix size for ERADC is 20 by 60.
Maximum Hankel singular value = 5.645725e+01
Minimum Hankel singular value = 1.762175e-02
  Damping(%)   Freq(HZ)    Mode SV
  1.0000e+02  1.5350e-01  6.0815e-02
```

9.4207e+00	1.4276e+01	7.8715e-02
9.4207e+00	1.4276e+01	7.8715e-02
3.5310e+00	1.9212e+01	9.6619e-02
3.5310e+00	1.9212e+01	9.6619e-02
2.7946e+01	2.0792e+01	1.2707e-01
6.0418e-01	1.6173e+01	2.3699e-01
6.0418e-01	1.6173e+01	2.3699e-01
2.0623e+00	1.7850e+00	2.6898e-01
2.0623e+00	1.7850e+00	2.6898e-01
1.2189e-01	1.0226e+01	3.4630e-01
1.2189e-01	1.0226e+01	3.4630e-01
3.7105e-01	1.3994e+01	5.5981e-01
3.7105e-01	1.3994e+01	5.5981e-01
3.0739e-01	8.7205e+00	7.8162e-01
3.0739e-01	8.7205e+00	7.8162e-01
1.9688e-01	6.1281e+00	8.3477e-01
1.9688e-01	6.1281e+00	8.3477e-01
3.1895e-01	3.0533e+00	1.0000e+00



mfd2ss

Purpose:

Constructs a state space observable canonical model from a matrix fraction description model.

Synopsis:

[*A*,*B*,*C*,*D*]=mfd2ss(Az,Bz)

Description:

Given the discrete time matrix fraction description representation of the form

$$(I + A_1 z^{-1} + \cdots A_p z^{-p})y(k) = (B_0 + B_1 z^{-1} + \cdots B_p z^{-p})u(k)$$

where $z^{-1}y(k) = y(k-1)$, the observable canonical representation is easily written as

$$\begin{Bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_p(k+1) \end{Bmatrix} = \overbrace{\begin{bmatrix} 0 & \cdots & 0 & -A_p \\ I & \cdots & 0 & -A_{p-1} \\ & \ddots & & \vdots \\ 0 & \cdots & I & -A_1 \end{bmatrix}}^{\mathbf{A}} \begin{Bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{Bmatrix} + \overbrace{\begin{Bmatrix} A_p B_o - B_p \\ A_{p-1} B_o - B_{p-1} \\ \vdots \\ A_1 B_o - B_1 \end{Bmatrix}}^{\mathbf{B}} u(k)$$

$$y(k) = \underbrace{\begin{bmatrix} 0 & 0 & 0 & I \end{bmatrix}}_{\mathbf{C}} \begin{Bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_p(k) \end{Bmatrix} + \underbrace{B_o}_{\mathbf{D}} u(k)$$

On output, the state space matrices are constructed as shown above and the input parameters are arranged as follows

$$Az = -\begin{bmatrix} A_1 & A_2 & \cdots & A_p \end{bmatrix}^T \qquad A_i \in R^{m \times m}$$

$$Bz = \begin{bmatrix} B_0 & B_1 & \cdots & B_p \end{bmatrix}^T \qquad B_i \in R^{m \times r}$$

Example:

```
Az =
  -1  -4
  -3  -5
```

```
   -6   -9
   -7   -8
Bz =
    1    1
    3    6
    7    9
[A,B,C,D]=mfd2ss(Az,Bz)
A =
    0    0   -6   -7
    0    0   -9   -8
    1    0   -1   -3
    0    1   -4   -5
B =
    6
    8
    1
    3
C =
    0    0   -1    0
    0    0    0   -1
D =
     1
     1
```

okid_asf

Purpose:

Identification of a state space model from a given set of spectral matrices.

Synopsis:

$[A,B,C,D,Az,Bz]=\text{okid\_asf}(r,f,dt,Gz,\Phi_{uu},\Phi_{yu},\Phi_{yy},desp,P,iop)$

Description:

State space identification is performed in two phases. First, a matrix fraction description (MFD) model is computed based on experimentally determined spectral matrices. Second, a minimum order state space model is realized using the MFD model parameters. An outline of the approach is as follows.

Given the input/output auto and cross spectral matrices, the system frequency response function can be computed two different ways;

Approach 1: $\Phi_{yu}(k) \cong G(e^{j\omega_k})\Phi_{uu}(k)$

Approach 2: $\Phi_{yy}(k) \cong G(e^{j\omega_k})\Phi_{uy}(k)$

where a solution using approach 1 produces a lower bound estimate of the frequency response function and approach 2 yields an upper bound. Notice that the equations are written in terms of spectral densities. Benefits of estimating the frequency response functions using the spectral densities are reported in Ref. 1. A representation of the frequency response function is given as follows

$$G(z_k) = A(z_k)^{-1}B(z_k)$$

where

$$A(z_k) = I + A_1 z_k^{-1} + \cdots A_p z_k^{-p}$$
$$B(z_k) = B_0 + B_1 z_k^{-1} + \cdots B_p z_k^{-p}$$

and $z_k = e^{j\omega_k}$, $A_i \in R^{m\times m}$, $B_i \in R^{m\times r}$, and the parameter P is the assumed order of the corresponding matrix polynomials. Substituting the above representation into the equations in approach 1 and 2, and multiplying both sides of the resulting equation by $A(z_k)$ yields a linear least squares estimation problem. The estimated matrix polynomials are used to construct an observable canonical state space model. A minimum order model is obtained via realization theory. On output, the minimum order realized system is returned in [*A,B,C,D*] and the polynomial parameters are returned in *Az* and *Bz*. The subroutine input parameters are the number of inputs *r*, the frequency response vector f in units of Hertz, sample time is defined by *dt*, estimated frequency response *Gz* (used only for plotting), spectral matrices Φ_{uu}, Φ_{yu} and Φ_{yy}, a string variable with a test description, matrix fraction polynomial matrix order *P*, and *iop* is a parameter that when set to 1 forces the least squares solution to use approach 1 only.

For multi-input multi-output systems the spectral densities and frequency response function are stacked by columns. For example, a system with 3 outputs and 2 inputs is stacked as follows

$$\begin{bmatrix} \phi_{y_1u_1}(\omega_1) & \phi_{y_1u_2}(\omega_1) \\ \phi_{y_2u_1}(\omega_1) & \phi_{y_2u_2}(\omega_1) \\ \phi_{y_3u_1}(\omega_1) & \phi_{y_3u_2}(\omega_1) \\ \phi_{y_1u_1}(\omega_2) & \phi_{y_1u_2}(\omega_2) \\ \phi_{y_2u_1}(\omega_2) & \phi_{y_2u_2}(\omega_2) \\ \phi_{y_3u_1}(\omega_2) & \phi_{y_3u_2}(\omega_2) \\ \vdots & \end{bmatrix} \xrightarrow{stacked}$$

$$\begin{bmatrix} \phi_{y_1u_1}(\omega_1) & \phi_{y_2u_1}(\omega_1) & \phi_{y_3u_1}(\omega_1) & \phi_{y_1u_2}(\omega_1) & \phi_{y_2u_2}(\omega_1) & \phi_{y_3u_2}(\omega_1) \\ \phi_{y_1u_1}(\omega_2) & \phi_{y_2u_1}(\omega_2) & \phi_{y_3u_1}(\omega_2) & \phi_{y_1u_2}(\omega_2) & \phi_{y_2u_2}(\omega_2) & \phi_{y_3u_2}(\omega_2) \\ & & \vdots & & & \end{bmatrix}$$

The number of rows in the stacked spectral matrix and frequency vector equals the number of frequency response points included in the analysis. One can skip frequency points in areas where the frequency response is dense. When skipping frequency values the frequency vector and corresponding spectral matrices should be decimated in the same way.

Stability of the matrix polynomial representation is not guaranteed. A stabilizing procedure is included based on the work presented in Ref. 2 that automatically computes a stable solution from the unstable model.

Algorithm:

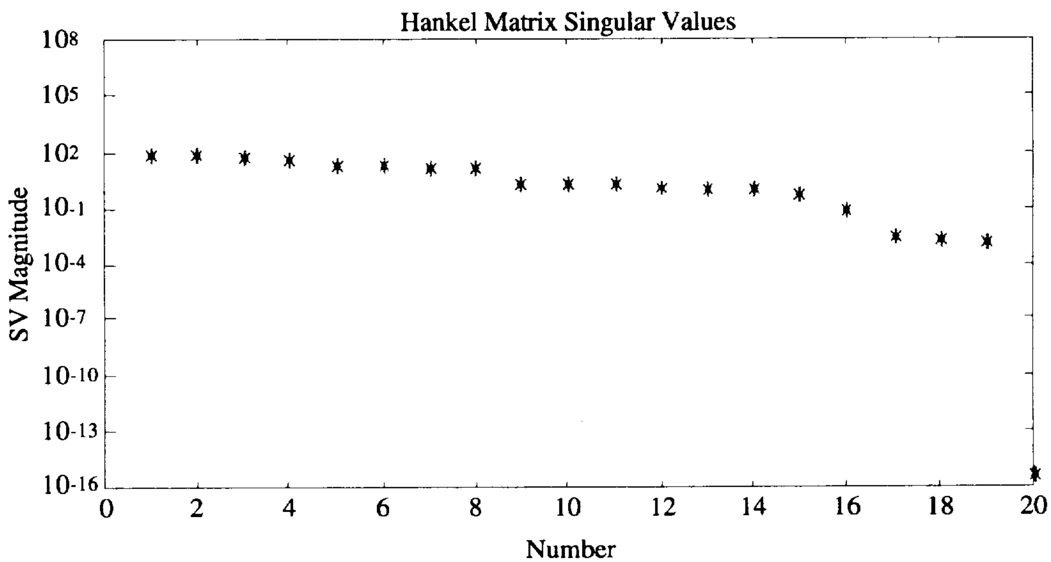
Least squares solution using singular value decomposition for the polynomial matrices.

Example:

This example uses data from a testbed located at NASA Langley. A file named *cemdata* contains experimental data for 2-inputs and 8-outputs. Only the first input/output pair is used in this example.

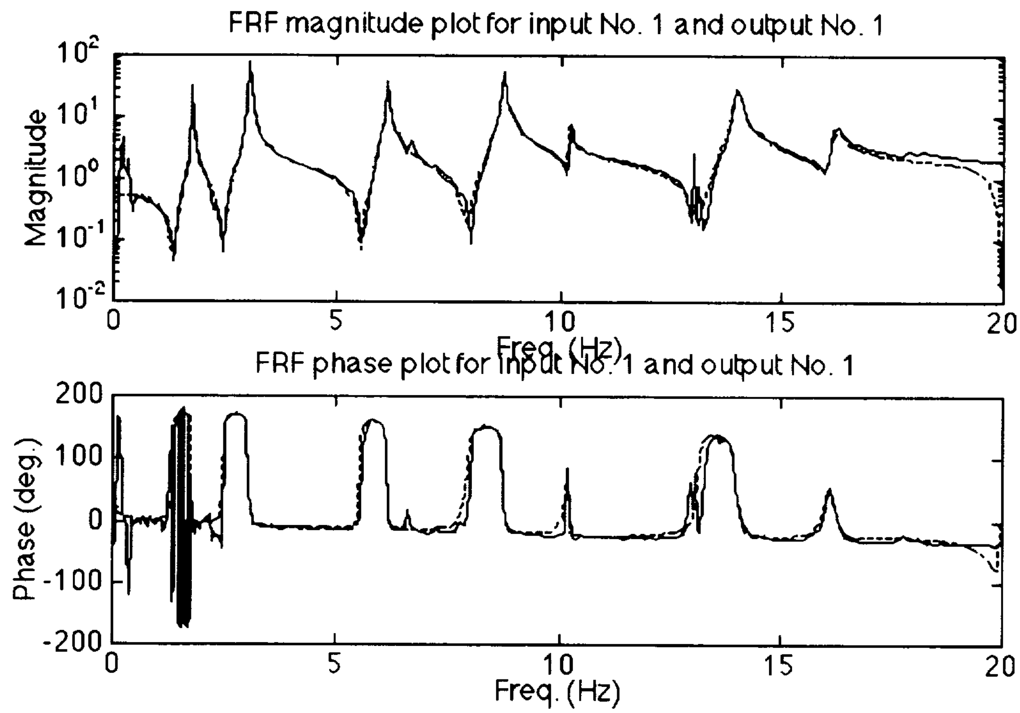
```
load cemdata
[ntot,junk]=size(YU_cross);
f=fhz;
df=fhz(2)-fhz(1);
fny=f(ntot);
dt=1/(2*fny);
r=1; m=1;
SUU=zeros(ntot,r*r); SYY=zeros(ntot,m*m);
%
%       Data from Testbed is incomplete
%       Fill in missing info with zeros
%
IM=1;
Vyy=[];Vyu=[];Vuu=[];
Vyy=[1:m+1:m*m]; Vuu=[1:r+1:r*r];
IM=[1:m];
IR=1;
 for j=1:r
   tm=IM+8*(j-1);
   Vyu=[Vyu tm];
  end
Gz=Y_frf(:,Vyu);
SYU=conj(YU_cross(:,Vyu));
SYY=[];
SUU(:,Vuu)=U_auto(:,1:r);
%
%      Skipping frequency points
%
Vs=[1:64 65:2:640];
SUU=SUU(Vs,:);SYU=SYU(Vs,:);
Gz=Gz(Vs,:);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot));
L=[1:ntot]';
```

```
P=20;
[A1,B1,C1,D1,Az1,Bz1]=okid_asf(r,f,dt,Gz,SUU,SYU,SYY,'OKIDASF ',P,1);
OKIDASF
Total number of sample  points = 352
Number of inputs = 1
Number of outputs = 1
Corresponding sampling rate = 39.87 Hz
Number of desired Markov parameters = 81
Have you run OKID with the same data & P before (1=yes,0=no) ?:= 0
Time (min) to compute ARX parameters 0.1156
No. of eigenvalues: 20
No. of unstable eigenvalues: 1
All unstable poles are real and are discarded
ERADC is used now.
The Hankel matrix size for ERADC is 20 by 60.
Maximum Hankel singular value = 6.428376e+01
Minimum Hankel singular value = 1.998350e-14
```




```
The HSV plot allows you to determine a desired model size.
Besides, further modal reduction may also be desired.
See the option of this function.
Desired Model Order (See HSV plot) (0=stop)=: 20
Model Describes 100 (%) of Test Data
Number of Modes Wanted (See MSV plot) =: 20
  Damping(%)   Freq(HZ)     Mode SV
  1.3819e+00   1.9935e+01   8.8491e-08
  4.8210e+00   1.3266e+01   4.5866e-02
  4.8210e+00   1.3266e+01   4.5866e-02
  6.8664e+01   2.7418e+01   8.9802e-02
  1.0000e+02   2.0803e-02   9.3621e-02
  2.5822e+00   1.9939e+01   1.3214e-01
  6.8636e-01   1.6200e+01   2.1888e-01
  6.8636e-01   1.6200e+01   2.1888e-01
  1.8313e-01   1.0212e+01   3.5822e-01
  1.8313e-01   1.0212e+01   3.5822e-01
```

```
4.9892e-01  1.3989e+01  5.2464e-01
4.9892e-01  1.3989e+01  5.2464e-01
2.9777e-01  6.1372e+00  7.0425e-01
2.9777e-01  6.1372e+00  7.0425e-01
2.5933e-01  1.7743e+00  7.7608e-01
2.5933e-01  1.7743e+00  7.7608e-01
3.3540e-01  8.7337e+00  7.7728e-01
3.3540e-01  8.7337e+00  7.7728e-01
3.0906e-01  3.0536e+00  1.0000e+00
3.0906e-01  3.0536e+00  1.0000e+00
Desired Model Order (See HSV plot) (0=stop)=: 0
Time (min) to compute the system model 0.517
Compare Recons. FRF and Real FRF (1=yes,0=no) ?:= 1
Time (min) to reconstruct FRF 0.0298
```



See also:

arx_fqml,mfd2ss,arxstable,frf2ss

References:

1) Horta, L.G. and Juang, J-N," Frequency Domain System Identification Methods: Matrix Fraction Description Approach," Proceedings of the 1993 Guidance, Navigation, and Control Conference, Monterey, CA, Paper No. 93-3839.

2) Chen, C.-W., Juang J.-N., and Lee, G., "Stable State Space System Identification from Frequency Domain Data," Proceedings of the first IEEE Regional Conference on Aerospace Control Systems, CA., May 25-27, 1993.

okid_fqm

Purpose:

Identification of a state space model from a frequency response function.

Synopsis:

[*A,B,C,D,Az,Bz*]=okid_fqm(*r,f,dt,Gz,desc,P*);

Description:

Identification of state space models from a frequency response function is divided into two steps: First, a matrix fraction description (MFD) is fitted to the frequency response function. Second, a minimum order state space model is realized based on the MFD parameters. The procedure is outlined as follows.

Given an experimentally determined frequency response function, a model representation in terms of matrix polynomials is the following

$$G(z_k) = A(z_k)^{-1} B(z_k)$$

where

$$A(z_k) = I + A_1 z_k^{-1} + \mathrm{L}\; A_p z_k^{-p}$$

$$B(z_k) = B_0 + B_1 z_k^{-1} + \mathrm{L}\; B_p z_k^{-p}$$

and $z_k = e^{j\omega_k}$, $A_i \in R^{m\times m}$, $B_i \in R^{m\times r}$, and the parameter P is the assumed order of the corresponding matrix polynomials. Multiplying both sides of the first equation by $A(z_k)$ yields a linear least squares estimation problem. The estimated matrix polynomials are used to construct an observable canonical state space model. A minimum order model is obtained subsequently via realization theory. On output, the minimum order realized system is returned in [*A,B,C,D*] and the polynomial parameters are returned in *Az* and *Bz*. The subroutine input parameters are the number of inputs *r*, the frequency response vector *f* in units of Hertz, sample time is defined by *dt*, estimated frequency response *Gz*, a string variable *desc* with a test description, and the number of terms in the matrix polynomial representation *P*. The number of rows in the frequency response matrix and corresponding frequency vector determines the number of frequency response points included in the analysis. One can skip frequency points in areas where the frequency response is dense. When skipping frequency points the frequency vector and corresponding spectral matrices should be decimated in the same way. For multi-input multi-output systems the matrix *Gz* is stacked by columns. An example for proper stacking is shown in *okid_asf*. For more details, see algorithm discussion in Ref. 1.

Stability of the matrix polynomial representation is not guaranteed. A stabilizing procedure is included based on the work presented in Ref. 2 that automatically computes a stable solution from the unstable model.

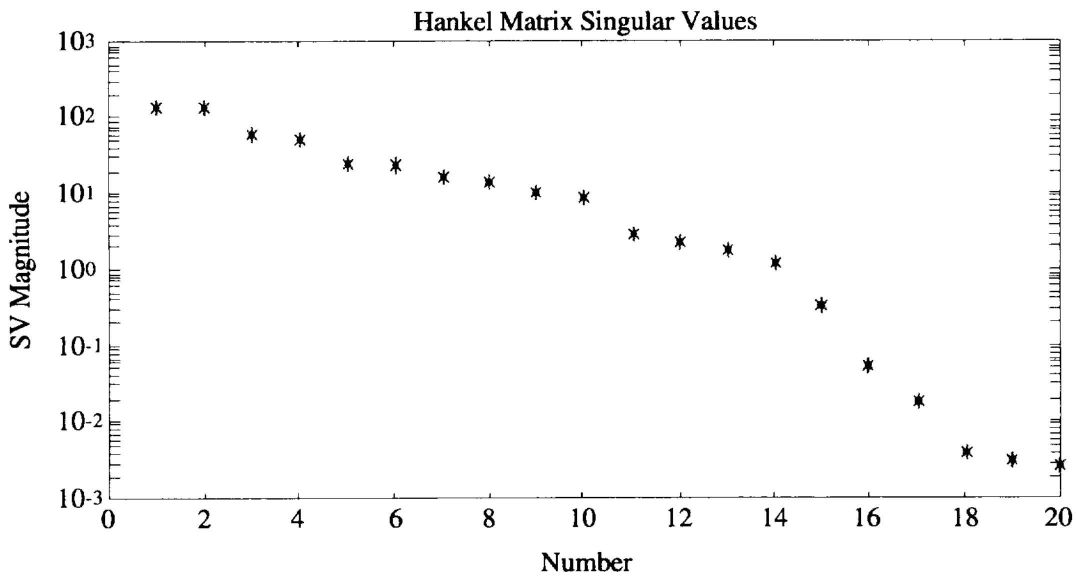
Approach:

First, a least squares solution for MFD parameters is obtained using singular value decomposition. The MFD model provides the system pulse response used with the function *eradc* to recover a minimum order state space model.

Example:

```
load cemdata
[ntot,junk]=size(YU_cross);
f=fhz;
df=fhz(2)-fhz(1);
fny=f(ntot);
r=1;
m=1;
%
%       Select columns from frequency responses
%
IM=1;
Vyu=[];
IM=[1:m];
 for j=1:r
   tm=IM+8*(j-1);
   Vyu=[Vyu tm];
  end
Gz=Y_frf(:,Vyu);
%
%      Skipping frequency points
%
Vs=[1:64 65:2:640];
Gz=Gz(Vs,:);f=fhz(Vs);
[ntot,jj]=size(Gz);
dt=1/(2*f(ntot));
P=20;
[A,B,C,D,Az,Bz]=okid_fqm(r,f,dt,Gz,' OKIDFQM',P);

OKIDFQM
 Total number of sample  points = 352
```

Number of inputs = 1
Number of outputs = 1
Corresponding sampling rate = 39.87 Hz
Number of desired Markov parameters = 81
Have you run OKID with the same data & P before (1=yes,0=no) ?:= 0
No. of eigenvalues: 20
Time (min) to compute ARX parameters 0.149
ERADC is used now.
The Hankel matrix size for ERADC is 20 by 60.
Maximum Hankel singular value = 1.366719e+02
Minimum Hankel singular value = 2.714249e-03

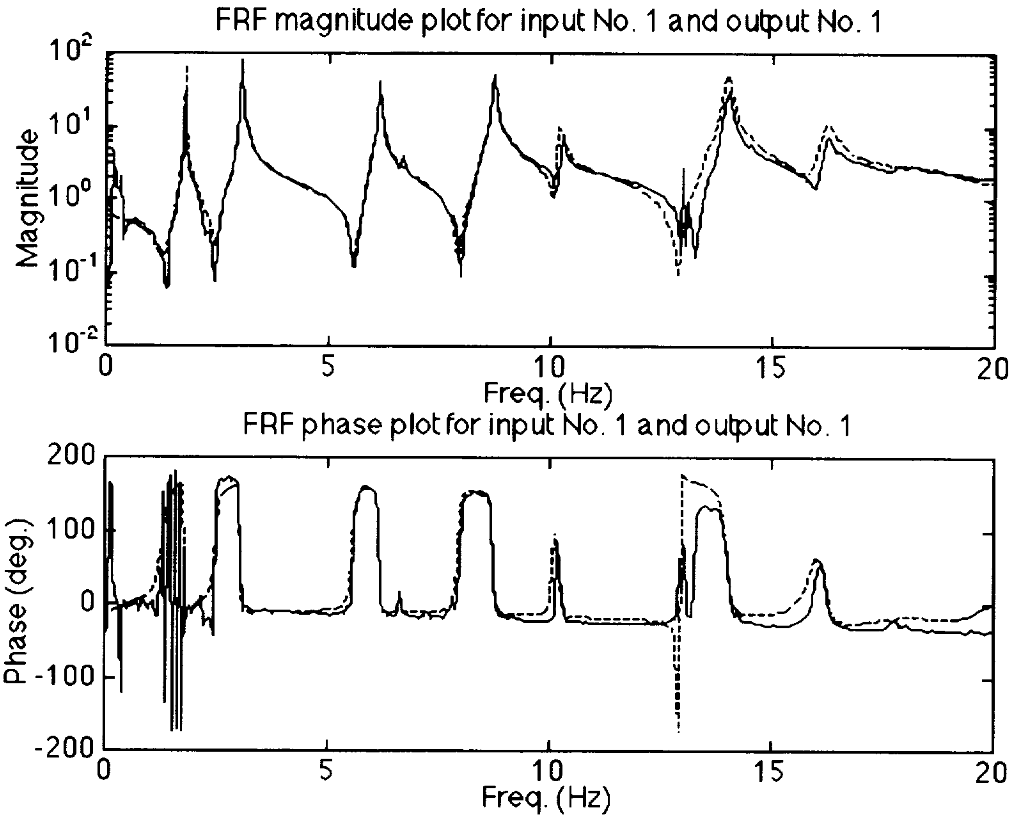


The HSV plot allows you to determine a desired model size.
Besides, further modal reduction may also be desired.
See the option of this function.
Desired Model Order (See HSV plot) (0=stop)=: 20
Model Describes 100 (%) of Test Data
Number of Modes Wanted (See MSV plot) =: 20

Damping(%)	Freq(HZ)	Mode SV
1.0000e+02	4.9299e-02	5.7064e-02
4.9193e+00	1.8667e+01	7.6341e-02
4.9193e+00	1.8667e+01	7.6341e-02
3.2112e+00	1.3633e+01	8.0297e-02
3.2112e+00	1.3633e+01	8.0297e-02
3.2082e+01	2.1045e+01	1.0955e-01
3.9617e-01	1.0185e+01	2.6634e-01
3.9617e-01	1.0185e+01	2.6634e-01
7.1932e-01	1.6213e+01	2.7279e-01
7.1932e-01	1.6213e+01	2.7279e-01
2.9492e-01	6.1385e+00	6.2107e-01
2.9492e-01	6.1385e+00	6.2107e-01
4.3469e-01	1.3983e+01	6.2689e-01
4.3469e-01	1.3983e+01	6.2689e-01
2.9853e-01	8.7272e+00	7.1157e-01
2.9853e-01	8.7272e+00	7.1157e-01

```
2.0720e-01  1.7783e+00  8.1915e-01
2.0720e-01  1.7783e+00  8.1915e-01
2.4028e-01  3.0515e+00  1.0000e+00
2.4028e-01  3.0515e+00  1.0000e+00
```

```
Desired Model Order (See HSV plot) (0=stop)=: 0
Time (min) to compute the system model 0.2922
Compare Recons. FRF and Real FRF (1=yes,0=no) ?:= 1
Time (min) to reconstruct FRF 0.03077
```



See also:

arx_fqmod,okid_asf,arxstable,frf2ss

References:

1) Chen, C.-W., Juang J.-N., and Lee, G., " Frequency Domain State-Space System Identification," NASA Technical Memorandum, 107659, July 1992.

2) Chen, C.-W., Juang J.-N., and Lee, G., "Stable State Space System Identification from Frequency Domain Data," Proceedings of the first IEEE Regional Conference on Aerospace Control Systems, CA., May 25-27, 1993.

stab

Purpose:

Computes a stable state space model given a desired characteristic polynomial and the unstable portion of the frequency response. Supporting function for *arxstable*.

Synopsis:

[*A*,*B*,*C*,*D*,*Gzns*,*pulse*]=stab(*Gzn*,*pns*,*r*,*m*,*f*,*dt*,*P*)

Description:

This function is used by *arxstable* to determine a stable state space representation of the unstable portion of a frequency response. The input parameters are the unstable frequency response *Gzn*, the desired stable characteristic polynomial *pns*, number of inputs *r*, number of outputs *m*, frequency vector *f* in units of Hertz, sample time *dt*, and the parameter *P* when multiplied by *m* equals the system order.

Algorithm:

See Ref. 1

Example:

See function *arxstable*

Reference:

1) Chen, C.-W., Juang J.-N., and Lee, G., "Stable State Space System Identification from Frequency Domain Data," Proceedings of the first IEEE Regional Conference on Aerospace Control Systems, CA., May 25-27, 1993.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Technical Memorandum

4. TITLE AND SUBTITLE
Frequency Domain Identification Toolbox

5. FUNDING NUMBERS
233-10-14-03

6. AUTHOR(S)
Lucas G. Horta, Jer-Nan Juang, and Chung-Wen Chen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM-109039

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 35

Availability: NASA CASI, (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)
This report documents software written in MATLAB programming language for performing identification of systems from frequency response functions. MATLAB is a commercial software environment which allows easy manipulation of data matrices and provide other instrinsic matrix functions capabilities. Algorithms programmed in this collection of subroutines have been documented elsewhere but all references are provided in this document. A main feature of this software is the use of matrix fraction descriptions and system realization theory to identify state space models directly from test data. All subroutines have templates for the user to use as guidelines .

14. SUBJECT TERMS
Identification, vibration, state space identification, MATLAB software

15. NUMBER OF PAGES
28

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102