MATLAB Model and Data Structures

August 30, 2018

Contents

Chapter 1

Calling Sequence

```
u = idinput(1024,'PRBS',[0  1/20],[-1  1])
a = [1  0.5];b = [0  2  3];
model = idpoly(a,b,'Ts',0.1)
y = sim(u,model) + rand(length(u),1)
plantData = iddata(y,u,0.1)
sys = armax(plantData,[2,2,1,1])
```

Chapter 2

Model structure (idpoly)

2.1 Model structure output [1]

1. Type of model : discrete-time[1]
2. Type of estimation method
3. Equation showing relation between polynomials
4. Model representation (A,B,C polynomials in increasing power of z^{-1})
5. Sample time
6. Parameterization : Polynomial orders, Number of free parameters
7. Hint for functions that can be used for extra information like uncertainties in polynomial values and their covariance
8. Information about whether the model is constructed with fixed parameters or estimated from data
9. Information about the data used to estimate the model
10. Fit percentage, Final Prediction Error(FPE), Mean Squared Error(MSE)
11. Focus - Error to be minimized ('prediction' or 'simulation')

[1]This option is also available in GUI (called using 'ident')

```
sys =
Discrete-time ARMAX model:  A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 + 0.4372 z^-1 - 0.1452 z^-2

  B(z) = 1.717 z^-1 + 2.555 z^-2

  C(z) = 1 + 0.7938 z^-1

Sample time: 0.1 seconds

Parameterization:
   Polynomial orders:   na=2   nb=2   nc=1   nk=1
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "plantData".
Fit to estimation data: 84.6% (prediction focus)
FPE: 0.2617, MSE: 0.2591
```

Figure 2.1: Model Output

2.2 Model structure attributes

1. Polynomial coefficients vector
2. Addition of integrators in noise channel [1]
3. Variable('z^{-1}') used in polynomials
4. Transport delays for each input/output pair using IODelay
5. Following information about each polynomial :
   (a) Parameter values
   (b) Upper and lower bounds on values
   (c) Property of each coefficient - Tunable or not
   (d) Scaling factor used to normalize the parameter value
   (e) Unit and label for each parameter
6. Value of variance of noise channel
7. Detailed report with following information :
   (a) Status : model is constructed or estimated
   (b) Estimation method used
   (c) Handling of initial conditions during estimation, specified as one of the following values : 'zero', 'estimate', 'backcast', 'auto' [1]

(d) Values of Mean Squared Error(MSE), Final Prediction Error(FPE), Fit Percentage, Raw Akaike's Information Criterion(AIC), Small sample-size corrected (AICc), Normalized AIC(nAIC), Bayesian information criteria(BIC), Loss function

(e) Information about parameters (free parameters vector, covariance, and label) [1]

(f) Following options are available in OptionsUsed structure :

i. Initial condition

ii. Specifying whether to display the estimation progress ('off' or 'on') [1]

iii. Input and output offset

iv. Controlling whether parameter covariance data is generated [1]

v. Regularized estimation of model parameters using a constant that determines the bias versus variance tradeoff [1]

vi. Following search methods available for iterative parameter estimation :

A. Subspace Gauss-Newton least squares search

B. Adaptive subspace Gauss-Newton search

C. Levenberg-Marquardt least squares search

D. Steepest descent least squares search

E. Non-linear least squares solver

F. Constrained non-linear solvers

vii. Following search options are available :

A. Tolerance

B. Maximum iterations

C. Step tolerance, function tolerance

viii. Focus - Error to be minimized ('prediction' or 'simulation') [1]

ix. Following options for weighting filter are available

A. Passbands Specify a row vector or matrix containing frequency values that define desired passbands

B. SISO filter

x. Controlling whether to enforce stability of estimated model (estimated model must be stable) [1]

xi. Following advanced options are available :

A. Error threshold to specify when to adjust the weight of large errors from quadratic to linear

B. Max Size to specify maximum number of elements in a segment when input-output data is split into segments
C. AutoInitThreshold to specify when to automatically estimate the initial condition
D. StabilityThreshold to specify thresholds for stability tests

(g) Controlling random number generation at the start of estimation

(h) Information about data used for estimation - type of data, length, sample time, offset in input or output, input intersample behaviour

(i) Following termination conditions for the iterative search used for prediction error minimization :

i. Reason
ii. Number of iterations performed
iii. Infinity norm of the gradient search vector when the search algorithm terminates
iv. Number of function call
v. Norm of the gradient search vector in the last iteration
vi. Criterion improvement in the last iteration
vii. Algorithm used by 'lsqnonlin' or 'fmincon' search method

8. Input Delay [1]
9. Output Delay
10. Sample time
11. Time unit
12. Name of input data vector (If not specified, default name 'u1')
13. Input unit
14. Input channel groups to assign input channels of MIMO systems into groups and refer to each group by name
15. Name of output data vector (If not specified, default name 'y1')
16. Output unit
17. Output channel groups to assign output channels of MIMO systems into groups and refer to each group by name

18. Notes in the form of string (An arbitrary field to store extra information)
19. Any other user data/comments
20. Any arbitrary name for model
21. Sampling grid for model arrays, specified as a data structure

```
get(sys)
               A: [1 0.4372 -0.1452]
               B: [0 1.7175 2.5552]
               C: [1 0.7938]
               D: 1
               F: 1
 IntegrateNoise: 0
       Variable: 'z^-1'
        IODelay: 0
      Structure: [1×1 pmodel.polynomial]
  NoiseVariance: 0.2604
         Report: [1×1 idresults.polyest]
     InputDelay: 0
    OutputDelay: 0
             Ts: 0.1000
       TimeUnit: 'seconds'
      InputName: {'u1'}
      InputUnit: {''}
     InputGroup: [1×1 struct]
     OutputName: {'y1'}
     OutputUnit: {''}
    OutputGroup: [1×1 struct]
          Notes: [0×1 string]
       UserData: []
           Name: ''
   SamplingGrid: [1×1 struct]
```

Figure 2.2: Model attributes

```
>> sys_armax.Structure

ans =
                A: [1×1 param.Continuous]
                B: [1×1 param.Continuous]
                C: [1×1 param.Continuous]
          IODelay: 0
    IntegrateNoise: 0

ARMAX model structure.
>> sys_armax.Structure.A

ans =

       Name: 'A'
      Value: [1 -1.0712 0.1001]
    Minimum: [-Inf -Inf -Inf]
    Maximum: [Inf Inf Inf]
       Free: [0 1 1]
      Scale: [1 1 1]
       Info: [1×3 struct]
```

Figure 2.3: sys.Structure

```
ans =

            Status: 'Estimated using ARMAX with prediction focus'
            Method: 'ARMAX'
   InitialCondition: 'zero'
               Fit: [1×1 struct]
        Parameters: [1×1 struct]
       OptionsUsed: [1×1 idoptions.polyest]
         RandState: [1×1 struct]
          DataUsed: [1×1 struct]
       Termination: [1×1 struct]
```

Figure 2.4: sys.Report

```
>> sys_armax.Report.OptionsUsed
Option set for the armax command:

      InitialCondition: 'auto'
               Display: 'off'
           InputOffset: []
          OutputOffset: []
    EstimateCovariance: 1
        Regularization: [1×1 struct]
          SearchMethod: 'auto'
         SearchOptions: [1×1 idoptions.search.identsolver]
                 Focus: 'prediction'
       WeightingFilter: []
      EnforceStability: 0
              Advanced: [1×1 struct]
```

Figure 2.5: sys.Report.OptionsUsed

Chapter 3

Data structure (iddata)

3.1 Data structure output

1. Domain of data : Time or frequency
2. Number of samples
3. Sample time
4. Output vector name
5. Output unit
6. Input vector name
7. Input unit

3.2 Data structure attributes

1. Domain of data : Time or frequency
2. Any arbitrary name for data
3. Output data vector stored in OutputData
4. Output data vector stored in y
5. Name of output vector which is displayed on iddata call (If not specified, default names, 'y1';'y2';...)
6. Output unit

7. Input data vector stored in InputData
8. Input data vector stored in u
9. Name of input vector which is displayed on iddata call (If not specified, default names, 'u1';'u2';...)
10. Input unit
11. Period of input data
12. Inter sample behaviour of input : 'zoh', 'foh', or 'bl', depending on whether data is piecewise constant, piecewise linear, or band limited
13. Sample time
14. Starting time instant stored in Tstart
15. Time instants stored in a vector
16. Time unit
17. Name of experiment (iddata can store data of multiple experiments)
18. Notes in the form of string (An arbitrary field to store extra information)
19. Any other user data/comments

```
plantData =

Time domain data set with 1024 samples.
Sample time: 0.1 seconds

Outputs      Unit (if specified)
   y1

Inputs       Unit (if specified)
   u1
```

Figure 3.1: Data Output

```
>> get(plantData)

ans =

  struct with fields:

              Domain: 'Time'
                Name: ''
          OutputData: [1024×1 double]
                   y: 'Same as OutputData'
          OutputName: {'y1'}
          OutputUnit: {''}
           InputData: [1024×1 double]
                   u: 'Same as InputData'
           InputName: {'u1'}
           InputUnit: {''}
              Period: Inf
         InterSample: 'zoh'
                  Ts: 0.1000
              Tstart: []
    SamplingInstants: [1024×0 double]
            TimeUnit: 'seconds'
      ExperimentName: 'Exp1'
               Notes: {}
            UserData: []
```

Figure 3.2: Data attributes