On Parametric Model Estimation

LUMINITA GIURGIU, MIRCEA POPA
Technical Sciences Department
“Nicolae Balcescu” Land Forces Academy
Revolutiei no. 3-5 Street, Sibiu
ROMANIA
http://www.armyacademy.ro

Abstract: - Building mathematical models of a dynamic system based on measured data, by adjusting parameters within a given model until its output coincides with the measured output is one of the means ideas in the black box modelling - system identification technique. Picking a model, from the model structure, that provides the best one-step ahead predictions in terms of the smallest expected squared error between observed outputs and predictions – the estimation process- is the first important step in the identification procedure. In many real-world situations, it is too difficult to describe a system using known physical laws. This article exploits the MatLab System Identification Toolbox objects, methods, and functions to perform black-box modelling. Based on measured data sets obtained by simulation of linear models, considerations and comments are made upon the results of the SIT techniques application.

Key-Words: - modelling, estimation, parametric models, prediction error, linear model structure

1 Introduction

A black-box model is a flexible structure that is capable of describing many different systems and its parameters might not have any physical interpretation. Black-box modeling has the advantage that can estimate many model structures and compare them to choose the best one.

Prediction error approach is the method where the parameters of the model are chosen so that the difference between the model predicted output and the measured output is minimized. The availability of this method covers all model structures.

2 Linear Model Structures

According to Ljung [1] a system is called linear if it is possible to describe it by a model like this:

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

where *G* and *H* are transfer functions in the time delay operator. Equation (1) said that the measured output *y(t)* is a sum of two contributions: first comes from the measured input *u(t)* and second comes from the noise *He*.

The objective of the identification procedure is to determine good estimates of the two transfer functions, meaning the model’s ability to produce one-step ahead predictions with errors of low variance. The predictor form of the model or the minimum variance (one-step ahead) prediction is given by:

$$\begin{gathered}\hat{y}(t\,|\,t-1)=\\ H^{-1}(q^{-1})G^{-1}(q^{-1})u(t)+[1-H^{-1}(q^{-1})]\,y(t)\end{gathered} \tag{2}$$

The model structure is a parameterized set *M* of candidate models:

$$\begin{gathered}M:\{G(q^{-1},\theta),H(q^{-1},\theta)\,|\,\theta\in D_m\},\\ y(t)=G(q^{-1},\theta)u(t)+H(q^{-1},\theta)e(t),\end{gathered} \tag{3}$$

where θ signifies the adjustable parameters and D_m is a subset in R^p inside which the search for a model should be carried out.

The inclusion of θ as an argument in predictor form implies that the model structure represents a set of models. The model structure can be written in the alternative form:

$$\hat{y}(t\,|\,\theta)=\varphi^T(t)\theta \tag{4}$$

where θ is the parameter vector and φ is the regression vector, which contains past inputs, past outputs, or signals derived from the inputs and outputs.

A particular choice of parameter vector $\theta=\hat{\theta}$ represents the model. Different assumptions about the spectral density of the noise and how the noise is assumed to enter the system will determine simplifications and consequently the different types of model structures. The general parametric model structure is represented in fig.1 and written as:

$$A(q^{-1})y(t) = q^{-d}\frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t) \quad (5)$$

where *A*, *B*, *C*, *D*, *F* are monic polynomials written in time delay operator, *d* represents the number of delays between input and output and *e* is a white noise sequence with zero mean value.

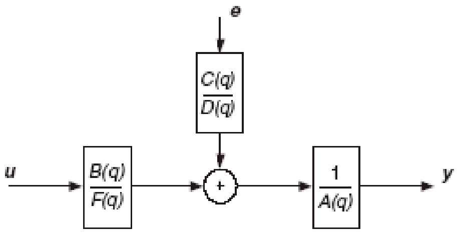

Fig. 1 The general parametric model structure

Special forms of (1), where one or more polynomial are set to identity are known as: AR, ARX, ARMAX, BJ, OE, models. System Identification Toolbox handles models structures defined in a flexible way by specifying the orders and delays in corresponding estimation routines such as arx, iv4, oe, bj, armax, pem.

3 Model Estimation

System identification is the procedure of deriving a model from data and model estimation is the procedure of fitting a model with a specific model structure. We have linear models and parametric models of a specific structure—e.g., physical models, ARMAX models. Parametric models have a well-defined mathematical structure, and this structure is fit to the input-output data by adjusting the coefficient values (model parameters).

Parametric identification methods use numerical search to find the parameter values that correspond to the best agreement between simulated and measured output. The prediction error in (1) can be computed as:

$$e(t) = H^{-1}(q)[y(t) - G(q)u(t)] \quad (6)$$

It means that for given data *y* and *u*, these errors are functions of *G* and *H*. The parametric identification method (prediction error approach) determines estimates of *G* and *H* by minimizing the criterion:

$$V_N(G,H) = \sum_{t=1}^{N} e^2(t) \quad (7)$$

that is:

$$[\hat{G}_N, \hat{H}_N] = min\sum_{t=1}^{N} e^2(t) \quad (8)$$

The SIT's functions for parametric model estimation share the same command structure: *model* = *pem* (*data*, *modelstructure*), where *data* is an iddata object and contains the output and input data sequences, the variable *modelstructure* specifies the particular structure (orders and delays) of the model to be estimated and *pem* represents the estimation routine. The resulting estimated model is contained in *model*, coded in theta format and colecting information about the model structure and orders, delays, parameters and estimated covariances of estimated parameters. Models of basically any structure can be constructed based on the prediction error method.

These parameter estimation routines require an itterative search for the minimum of the function (7): this search uses a startup procedure based on least squares and instrumental variables [2]. From the initial estimate, a Gauss-Newton minimization procedure is carried out until the norm of the Gauss-Newton direction is less than a certain tolerance [3]. The itterations also stops when reaching the maximum number initially established by the routine or when none decrease of the criterion is observed along the search direction.

Observations: while the search is typically initialized by the built-in startup procedure giving just orders and delays, the ability to force specific initial conditions is useful in many contexts. The iterative search procedure in pem, armax, oe, bj routines lead to theta values corresponding to a local minima of the criterion function, but nothing guarantees that this local minima is also a global one. Even if the startup procedure for black-box models in these routines is reasonably efficient in giving initial estimates that lead to the global minimum, it is a good idea to try starting the minimization at several different initial conditions to see if a smaller value of the loss function can be found. The estimated covariance matrix of the estimated parameter vector is returned by these routines, as part of the *model*. This reflects the reliability of the estimates.

4 Simulation and estimation case study

The case under study takes for simulating synthetic data an ARMAX system given initially by the linear difference equation, see Listing 1.

The steps made are as follows:

- *model1* contains the *A*, *B*, *C* original parameters polynomials;
- based on the obtained data set, an identification procedure is applied, see Listing 2;

- the results are the *th* model which contains this time the *A*, *B*, *C* estimated parameters polynomials, the value of the criterion (Loss function) and the final prediction error (FPE);
- the minimization algorithm is modified by decreasing the admitted values of the norm search vector (Tolerance) and estimated residual standard deviation (LimitError-robustification), and applied again, in order to better achievement results, see Listing 3;
- the results are the new model *model2* which contains the new *A*, *B*, *C* estimated parameters polynomials and the new values of Loss function and FPE;
- the three models *model1*, *th* and *model2* are compared, see Table1 where are centralized the values of original parameters, first estimated parameters, second estimated parameters, Loss function and FPE.

Listing 1

```
model1=poly2th([1 -1.5 0.7],[0 1 0.5],[1 -1 0.2]);
u=idinput(400,'rs',[0 1]);
e=randn(400,1);
y=idsim([u e],model1);
```

Listing 2

```
z=[y u]
th=armax(z, [2 2 2 1], 'trace')
Initial Estimate:
   Current loss: 0.914        ParVector
                               -1.4945
                               0.6917
                               1.0098
                               0.5102
                               -1.0585
                              0.2277
Iteration # 1:
   Search direction: gns
   Bisected search vector 0 times
   Current loss: 0.80863   Previous loss: 0.81338
   New par   prev. par   gn-dir
   -1.5052   -1.4945   -0.0107
    0.7024    0.6917    0.0106
    0.9974    1.0098   -0.0123
    0.4912    0.5102   -0.0189
   -1.0827   -1.0585   -0.0241
    0.2240    0.2277   -0.0036
   Norm of gn-vector: 0.036525
   Expected improvement: 0.56548 %
   Achieved improvement: 0.58378 %
Iteration # 2:
   Search direction: gns
   Search direction: lm
   Search direction: grad
   Bisected search vector 6 times
   Current loss: 0.80862   Previous loss: 0.80863
   New par   prev. par   gn-dir
   -1.5047   -1.5052    0.0016
    0.7021    0.7024   -0.0011
    0.9976    0.9974   -0.0026
    0.4916    0.4912    0.0074
   -1.0834   -1.0827    0.0009
    0.2232    0.2240   -0.0031
   Norm of gn-vector: 0.0086718
   Expected improvement: 0.0083665 %
   Achieved improvement: 0.00043643 %
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + C(q)e(t)
A(q) = 1 - 1.505 q^-1 + 0.7021 q^-2
B(q) = 0.9976 q^-1 + 0.4916 q^-2
C(q) = 1 - 1.083 q^-1 + 0.2232 q^-2
Estimated using ARMAX from data set z
Loss function 0.912001 and FPE 0.94028
```

4.1 The algorithm

An iterative Gauss-Newton algorithm is used to minimize a robustified quadratic prediction error criterion. Each iteration the Gauss-Newton vector is bisected up to 10 times until a lower value of the criterion is found: Current loss: 0.80863 ≤ Previous loss: 0.81338 (Iteration #1). If no such value is found, a Levenberg-Marquardt (lm) or gradient (grad) search direction is used instead and the procedure is repeated (Iteration #2).

The minimization information furnished includes current (New par) and previous (prev par) parameter estimates, values of the criterion function (Current loss), Gauss-Newton vector (gn-dir) and its norm (Norm of gn-vector), number of times the search vector has been bisected.

Listing 3

```
m=armax(z,[2 2 2 1],'tol',0.001,'lim',1,'trace','full');
modified_alg=m.alg
model2=armax(z,[2 2 2 1], 'alg', modified_alg)
modified_alg =
Approach: 'Pem'
Focus: 'Prediction'
MaxIter: 20
Tolerance: 0.0010
LimitError: 1
MaxSize: 'Auto'
SearchDirection: 'Auto'
FixedParameter: []
Trace: 'Full'
N4Weight: 'Auto'
N4Horizon: 'Auto'
Advanced: [1x1 struct]
Initial Estimate:
   Current loss: 0.914        ParVector
                               -1.4945
                               0.6917
                              1.0098
                               0.5102
                               -1.0585
                              0.2277
Iteration # 1:
   Search direction: gns
   Bisected search vector 0 times
   Current loss: 0.92562   Previous loss: 0.93392
   New par   prev. par   gn-dir
   -1.5036   -1.4945   -0.0092
    0.7023    0.6917    0.0105
    0.9901    1.0098   -0.0196
    0.5014    0.5102   -0.0088
   -1.0851   -1.0585   -0.0265
    0.2178    0.2277   -0.0098
   Norm of gn-vector: 0.038203
   Expected improvement: 0.84686 %
   Achieved improvement: 0.88878 %
Iteration # 2:
   Search direction: gns
   Bisected search vector 0 times
   Current loss: 0.92538   Previous loss: 0.92562
   New par   prev. par   gn-dir
   -1.5008   -1.5036    0.0028
    0.6996    0.7023   -0.0027
    0.9897    0.9901   -0.0004
    0.5090    0.5014    0.0076
   -1.0830   -1.0851    0.0021
    0.2137    0.2178   -0.0041
   Norm of gn-vector: 0.0097605
   Expected improvement: 0.024797 %
   Achieved improvement: 0.026487 %
Iteration # 3:
   Search direction: gns
   Bisected search vector 0 times
   Current loss: 0.92537   Previous loss: 0.92538
   New par   prev. par   gn-dir
   -1.5008   -1.5008    0.0000
    0.6995    0.6996   -0.0000
    0.9899    0.9897    0.0001
    0.5088    0.5090   -0.0002
   -1.0832   -1.0830   -0.0002
    0.2128    0.2137   -0.0009
   Norm of gn-vector: 0.00094901
   Expected improvement: 0.00056232 %
   Achieved improvement: 0.00053764 %
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + C(q)e(t)
A(q) = 1 - 1.501 q^-1 + 0.6995 q^-2
B(q) = 0.9899 q^-1 + 0.5088 q^-2
C(q) = 1 - 1.083 q^-1 + 0.2128 q^-2
Estimated using ARMAX from data set z
Loss function 0.911245 and FPE 0.939501
```

Table 1

	model1	*th*	*model2*
a_0	1	1	1
a_1	- 1.5	-1.505	-1,501
a_2	0.7	0.721	0.6995
b_0	0	0	0
b_1	1	0.9976	0.9899
b_2	0.5	0.4916	0.5088
c_0	1	1	1
c_1	-1	-1.083	-1.083
c_2	0.2	0.2232	0.2128
Current loss		0.912001	0.911245
FPE		0.94028	0.93950

4.2 Refining the estimates

The case under study takes a data set obtained in an experiment. To refine the model estimates, two ways are followed:

1. the set splits in two parts: *set1* and *set2*; the first is used to estimate an ARMAX model *model1* with the default *armax* algorithm properties and the second is used to refine the initial model with *pem* routine and with modified algorithm settings: increased number of maximum iterations and smaller tolerance;
2. for initial parameter guesses a model object is created first, using a constructor method and setting the initial parameter values in the model properties; this initial model is provided next as input to *pem* routine.

Figure fig.2 shows the compared results of the original data, initial model and refined model.

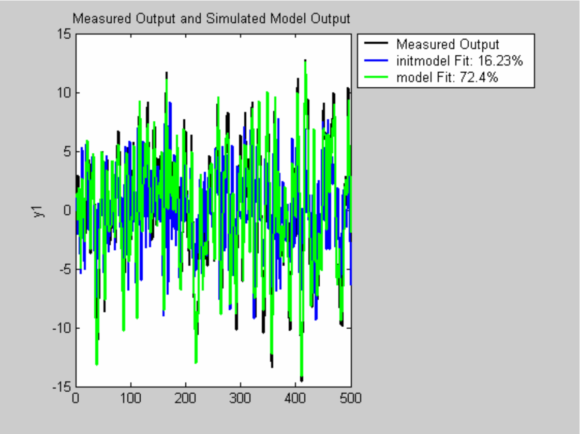

Fig. 2 Original data, initial and refined model

5 Conclusion

It is important to state the purpose of the model as a first step in the system identification procedure.

There are a huge variety of model applications, for example, the model could be used for control, prediction, signal processing, error detection or simulation. The purpose of the model affects the choice of identification methods and the experimental conditions, and it should therefore be clearly stated (for example, it is important to have an accurate model around the desired crossover frequency, if the model is used for control design).

This article has shown that the estimation process, as the first important step in the identification procedure, can be improved by modifying the minimization algorithm and by refining the estimates.

Estimating polynomial models means providing input delays and model orders. When known insights into the physics of the system exist, there is the possibility to guess the number of poles and zeros. However, in most cases, we do not know the model orders in advance.

Any single-output polynomial model can be estimated using the iterative prediction-error estimation method *pem*. For Gaussian disturbances, this method gives the maximum likelihood estimate that minimizes the prediction errors to obtain maximum-likelihood values. The resulting models are stored as idpoly model objects. This routine can also be used to refine parameter estimates of an existing polynomial model.

It is very important to note that validating the estimated model is the next step in determining how robust the estimation was. Even if the estimation data set returns very good results, without validation of the model, there is no way of telling if the estimated parameters were over-fitted for a particular data set. Using additional data sets for validation shows how the model responds to a variety of different inputs and if the original estimation was appropriate.

References:

[1] Ljung L. and T. Glad, Modeling of Dynamic Systems, Prentice Hall, Englewood Cliffs, N.J., 1994

[2] Raol J. R., Girija G. and Singh J., Modelling and Parameter Estimation of Dynamic Systems, The Institution of Engineering and Technology, 2005

[3] Ferdinand van der Heijden, Robert Duin, Dick de Ridder, David M. J. Tax, Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB, Willey, 2004

[4] Englezos P., Kalogerakis N., Englezos E., Applied Parameter Estimation for Chemical Engineers, Marcel Dekker Ltd., 2001

[5] Ljung L., System Identification Toolbox– For use with MatLab, The MathWorks, Inc., 1997

[6] Katayama T., Subspace Methods for System Identification, Springer, 2005

[7] Yucai Z., Multivariable System Identification: for Process Control, Eindhoven University of Technology, Pergamon Pres, 2001