We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

System Identification Using Orthonormal Basis Filter

Lemma D. Tufa and M. Ramasamy
Chemical Engineering Department,
Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, Perak,
Malaysia

1. Introduction

Models are extensively used in the design and implementation of advanced process control systems. In model predictive control (MPC), model of the plant is used to predict the future output of the plant using the current and future optimal inputs and past outputs. Therefore, the design of MPC, essentially, includes the development of an effective plant model that can be used for predicting the future output of the plant with good accuracy (Camacho & Bordon, 2004; Rawlings, 2000). Models can be developed either from purely theoretical analysis (conservation principles, thermodynamics, etc.) or from experimental data or somewhere in between. The process of model development from experimental data is known as system identification. The identification test can be conducted either in open-loop (open-loop identification) or while the plant is under feedback control (closed-loop identification).

The theory of linear system identification is well developed and there are already numerous literatures. The pioneering work in system identification was done by Ljung (1999) and his book provides detailed theoretical foundation for system identification. The book by Nelles (2001) is also a very practical book and highly recommended for practitioners both on linear and non-liear system identification. Heuberger, *et al.*, (2005) authored a very comprehensive book on modeling and identification using rational orthonormal basis functions, though current developments in application of OBF for MPC, closed-loop identification, etc., were not included.

There are several linear dynamic model structures that are commonly used in control relevant problems. They have two general forms, *i.e.*, the state space and input-output forms. In this chapter, we deal with the latter form also called transfer function. The most common linear input-output model structures can be derived from one general structure (1). The general linear structure consists of a deterministic component, *i.e.*, the plant input , $u(k)$, filtered by a linear filter and a noise component, *i.e.*, a white noise, $e(k)$, filtered by a corresponding linear filter.

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}e(k) \tag{1}$$

The q in (1) is the forward shift operator defined as $q\,u(t) = u(t + 1)$ and q^{-1} is the delay (backward shift) operator, $q^{-1}u(t) = u(t - 1)$.

The various commonly used structures can be easily derived from the general model structure by making some assumptions. The ARX model can be derived from (1) by assuming $F(q) = D(q) = C(q) = 1$. Therefore, the ARX model structure has the form

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}e(k) \tag{2}$$

The Auto Regressive Moving Average with Exogenous Input (ARMAX) can be derived from (1) by assuming $F(q) = D(q) = 1$.

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}e(k) \tag{3}$$

Other linear model structures are listed below:

Box Jenkins (BJ):

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k) \tag{4}$$

Output Error (OE):

$$y(k) = \frac{B(q)}{F(q)}u(k) + e(k) \tag{5}$$

Finite Impulse Response (FIR):

$$y(k) = B(q)u(k) + e(k) \tag{6}$$

It should be noted that in FIR model structures the filters are simple delays. Equation (6) can be expanded into

$$y(k) = (b_1 q^{-1} + b_2 q^{-2} + ... + b_m q^{-m})u(k) + e(k) \tag{7}$$

The selection of the appropriate model structure for a specific purpose, among other factors, depends on the consistency of the model parameters, the number of parameters required to describe a system with acceptable degree of accuracy and the computational load in estimating the model parameters. The optimality of model parameters is generally related to the bias and consistency of the model. Bias is the systematic deviation of the model parameters from their optimal value and inconsistency refers to the fact that the bias does not approach zero as the number of data points approach infinity (Nelles, 2001). The most widely used linear models are Step Response, ARX and FIR models (Ljung, 1999; Nelles, 2001). Their popularity is due to the simplicity in estimating the model parameters using the popular linear least square method. However, it is known that all of these three model structures have serious drawbacks in application. The ARX model structure leads to inconsistent parameters for most open-loop identification problems and the FIR and step

response model need very large number of parameters to capture the dynamics of a system with acceptable accuracy. The inconsistency in the ARX and also in ARMAX model parameters is caused by the assumption of common denominator dynamics for both the input and noise transfer functions given by $1/A(q)$, which implies that the plant model and the noise model are correlated. In reality, this is rarely the case for open loop identification problems. The output error (OE) and the Box Jenkins (BJ) model structures assume independent transfer function and noise models, and hence they allow consistent parameter estimation. However, determination of the model parameters in both cases involves nonlinear optimization. In addition, in case of BJ, because of the large number of parameters involved in the equation, it is rarely used in practice, especially, in MIMO systems.

Orthonormal Basis Filter (OBF) models have several advantages over the conventional linear models. They are consistent in parameters for most practical open-loop systems and the recently developed ARX-OBF and OBF-ARMAX structures lead to consistent parameters for closed loop identification also. They require relatively a fewer numbers of parameters to capture the dynamics of linear systems (parsimonious in parameters) and the model parameters can be easily estimated using linear least square method (Heuberger, et al., 2005; Heuberger, et al., 1995; Ninness & Gustafsson, 1997; Van den Hof, et al., 1995). MIMO systems can be easily handled using OBF and OBF based structures. In addition, recent works by Lemma and Ramasamy (Lemma & Ramasamy, 2011) prove that OBF based structures show superior performance for multi-step ahead prediction of systems with uncertain time delays compared to most conventional model structures.

Among the earliest works on rational orthonormal bases was contributed by Takenaka (1925) in the 1920's in relation to approximation via interpolation, with the subsequent implications for generalized quadrature formula. In subsequent works, in the 1960s, Walsh (1975) contributed extensively in the applications of orthonormal bases for approximation, both in discrete time and continuous time analysis. In similar periods, Wiener (Wiener, 1949) examined applications of continuous time Laguerre networks for the purpose of building optimal predictor. Van den Hof, *et al.*, (1995) introduced the generalized orthonormal basis filters. They showed that pulse, Laguerre and Kautz filters can be generated from inner functions and their minimal balanced realizations. Ninness and Gustafsson (1997) unified the construction of orthonormal basis filters. Lemma, *et al.*, (2011) proposed an improved method for development of OBF models where the poles and time delays of the system can be estimated and used to develop a parsimonious OBF model. On another work (Lemma, et al., 2010) it was shown that BJ type OBF models can be easily developed by combing structures with AR and ARMA noise model. Some works on closed-loop identification using OBF based structures have also been presented (Badwe, et al., 2011; Gáspár, et al., 1999; Lemma, et al., 2009; Lemma & Ramasamy, 2011).

2. Development of conventional OBF models

Consider a discrete time linear system

$$y(k) = G(q)u(k) \tag{8}$$

where $G(q)$ = transfer function of the system. A stable system, $G(q)$, can be approximately represented by a finite–length generalized Fourier series expansion as:

$$G(q) = \sum_{i=1}^{n} l_i f_i(q) \qquad (9)$$

where $\{l_i\}$, i =1, 2, …, n are the model parameters, n is the number of parameters, and $f_i(q)$ are the orthonormal basis filters for the system $G(q)$. Orthonormal basis functions can be considered a generalization of the finite length fourier series expansion. Two filters f_1 and f_2 are said to be orthonormal if they satisfy the properties:

$$\langle f_1(q), f_2(q) \rangle = 0 \qquad (10)$$

$$\|f_1(q)\| = \|f_2(q)\| = 1 \qquad (11)$$

2.1 Common orthonormal basis filters

There are several orthonormal basis filters that can be used for development of linear OBF models. The selection of the appropriate type of filter depends on the dynamic behaviour of the system to be modelled.

Laguerre filter

The Laguerre filters are first-order lag filters with one real pole. They are, therefore, more appropriate for well damped processes. The Laguerre filters are given by

$$f_i = \sqrt{(1 - p^2)} \frac{(1 - pq)^{i-1}}{(q - p)^i}, \quad |p| < 1 \qquad (12)$$

where p is the estimated pole which is related to the time constant, τ, and the sampling interval T_s of the system by

$$p = e^{-(T_s/\tau)} \qquad (13)$$

Kautz filter

Kautz filters allow the incorporation of a pair of conjugate complex poles. They are, therefore, effective for modeling weakly damped processes. The Kautz filters are defined by

$$f_{2i-1} = \frac{\sqrt{(1 - a^2)(1 - b^2)}}{q^2 + a(b - 1)q - b} g(a, b, q, i) \qquad (14)$$

$$f_{2i} = \frac{\sqrt{(1 - b^2)}(q - a)}{q^2 + a(b - 1)q - b} g(a, b, q, i) \qquad (15)$$

where

$$g(a,b,q,i) = \left(\frac{-bq^2 + a(b-1)q + 1}{q^2 + a(b-1) - b} \right)^{i-1} \tag{16}$$

$-1 < a < 1$ and $-1 < b < 1$ $n = 1, 2, \ldots$

Generalized orthonormal basis filter

Van den Hof, *et al.*, (1995) introduced the generalized orthonormal basis filters and showed the existence of orthogonal functions that, in a natural way, are generated by stable linear dynamic systems and that form an orthonormal basis for the linear signal space l_2^n. Ninness & Gustafsson (1997) unified the construction of orthonormal basis filters. The GOBF filters are formulated as

$$f_i(q,p) = \frac{\sqrt{1 - |p_i|^2}}{(q - p_i)} \prod_{j=1}^{i-1} \frac{(1 - p_j^* q)}{(q - p_j)} \tag{17}$$

where $p \equiv \{p_j : j = 1, 2, 3, \ldots\}$ is an arbitrary sequence of poles inside the unit circle appearing in complex conjugate pairs.

Markov-OBF

When a system involves a time delay and an estimate of the time delay is available, Markov-OBF can be used. The time delay in Markov-OBF is included by placing some of the poles at the origin (Heuberger, et al., 1995). For a SISO system with time delay equal to d samples, the basis function can be selected as:

$$f_i = z^{-i} \text{ for } i = 1, 2, \ldots, d \tag{18}$$

$$f_{i+d}(q, p) = \frac{\sqrt{1 - |p_i|^2}}{(q - p_i)} \prod_{j=1}^{i-1} \frac{(1 - p_j^* q)}{(q - p_j)} z^{-d} \text{ for } i = 1, 2, \ldots, N \tag{19}$$

Patwardhan and Shah (2005) presented a two-step method for estimating time delays from step response of GOBF models. In the first step, the time delays in all input-output channels are assumed zero and the model is identified with GOBF. In GOBF models, the time delay is approximated by a non-minimum phase zero and the corresponding step response is an inverse response. The time delay is then estimated from a tangent drawn at the point of inflection.

2.2 Estimation of GOBF poles

Finding an appropriate estimate of the poles for the filters is an important step in estimating the parameters of the OBF models. Arbitrary choice of poles may lead to a non-parsimonious model. Van den Hof, *et al.*, (2000) showed that for a SISO system with poles $\{a_j : |a_j| < 1 \text{ for } j = 1, 2, \ldots, n\}$, the rate of convergence of the model parameters is determined by the magnitude of the slowest Eigen value.

$$\rho = \max_j \prod_{k=1}^{n} \left| \frac{a_j - p_k}{1 - \bar{p}_k a_j} \right| \tag{20}$$

where p_k = arbitrary poles.

Therefore, a good approximation by a small number of parameters can be obtained by choosing a basis for which ρ is small. It is shown that the poles determined by Van den Hof *et al.* method closely match the dominant poles of the system (Heuberger, et al., 2005; Wahlberg, 1991). Lemma, *et al.*, (2011) proposed a systematic way to estimate the dominant poles and time delays of a system from the input-output identification test data. An OBF model is first developed with randomly chosen real poles and generalized orthonormal basis filters with 10-12 terms. The model is simulated to get a noise free step response of the system. One or two poles of the system are estimated from the resulting noise free step response of the OBF model and it is also observed whether the system is weakly damped or not. This process can be repeated until some convergence criterion is fulfilled. The procedure normally converges after two or three iterations. The procedure is iiterations and is illustrated in Example 1.

2.3 Model parameter estimation

In OBF models, the output can be expressed as a linear combination of the input sequence filtered by the respective filters. For a finite number of parameters, from (9) we get

$$\hat{y}(k) = l_1 f_1(q) u(k) + l_2 f_2(q) u(k) + \ldots + l_n f_n(q) u(k) \tag{21}$$

Equation (21) is not linear in its parameters and therefore estimation of parameters using linear least square method is impossible. However, it can be modified such that it is linear in parameters, as

$$\hat{y}(k) = l_1 u_{f1}(k) + l_2 u_{f2}(k) + \ldots + l_n u_{fn}(k) \tag{22}$$

where $u_{fi}(k)$ is the filtered input given by

$$u_{fi}(k) = f_i(q) u(k) \tag{23}$$

Once the dominant poles of the system and the types of filters are chosen, the filters $f_1, f_2, \ldots, f_n$ are fixed. The filtered inputs, u_{fi}, are determined by filtering the input sequence with the corresponding filter. For an OBF model with n parameters, the prediction can be started from the nth instant in time. Equation (22) can be expanded and written in matrix form as

$$\begin{bmatrix} \hat{y}_{n+1} \\ \hat{y}_{n+2} \\ . \\ . \\ . \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} u_{f1}(n) & u_{f2}(n-1) & \ldots & u_{fn}(1) \\ u_{f1}(n+1) & u_{f2}(n) & \ldots & u_{fn}(2) \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ u_{f1}(N-1) & u_{f2}(N-2) & \ldots & u_{fn}(N-n) \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ . \\ . \\ . \\ l_n \end{bmatrix} \tag{24}$$

where N is the future time instant.

Equation (24) in vector-matrix notation is given by

$$\hat{y} = X\theta \tag{25}$$

where $\theta = [l_1, l_2, ..., l_n]^T$ is the parameter vector, $\hat{y}$ is the output vector $\hat{y} = [y_{n+1}, y_{n+2}, ..., y_N]$ and X is the regressor matrix given by

$$X = \begin{bmatrix} u_{f1}(n) & u_{f2}(n-1) & \cdots & u_{fm}(1) \\ u_{f1}(n+1) & u_{f2}(n) & \cdots & u_{fm}(2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ u_{f1}(N-1) & u_{f2}(N-2) & \cdots & u_{fm}(N-n) \end{bmatrix} \tag{26}$$

Since (25) is linear in parameters, the model parameters can be estimated using linear least square formula (27).

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{27}$$

Algorithm 1

1. Use GOBF structure and two randomly selected stable poles and develop (6 to 12) sequence of GOBF filters
2. Develop the regressor matrix (26) using the filters developed at step (1) and the input sequence u(k)
3. Use the linear least square formula (27) to estimate the model parameters
4. Make a better estimate of the poles of the system from the step response of the GOBF model
5. Repeat steps 1 to 4 with the new pole until a convergence criterion is satisfied

The Percentage Prediction Error (PPE) can be a good convergence criterion.

Example 1

An open loop identification test for SISO system is carried out and the input-output data shown in Figure 1 is obtained. A total of 4000 data points are collected at one minute sampling interval with the intention of using 3000 of them for modelling and 1000 for validation. Develop a parsimonious OBF model using the data. No information is available about the pole of the system.

Since there is no information about the poles of the system, two poles: 0.3679 and 0.9672 are arbitrarily chosen for the first iteration of the model. A GOBF model with six terms (you can choose other numbers and compare the accuracy if you need) is first developed with these two poles alternating. Note that, once the poles, type of filter, *i.e.,* GOBF and the number of terms is fixed the filters are fixed and the only remaining value to determine the model is the model parameters.
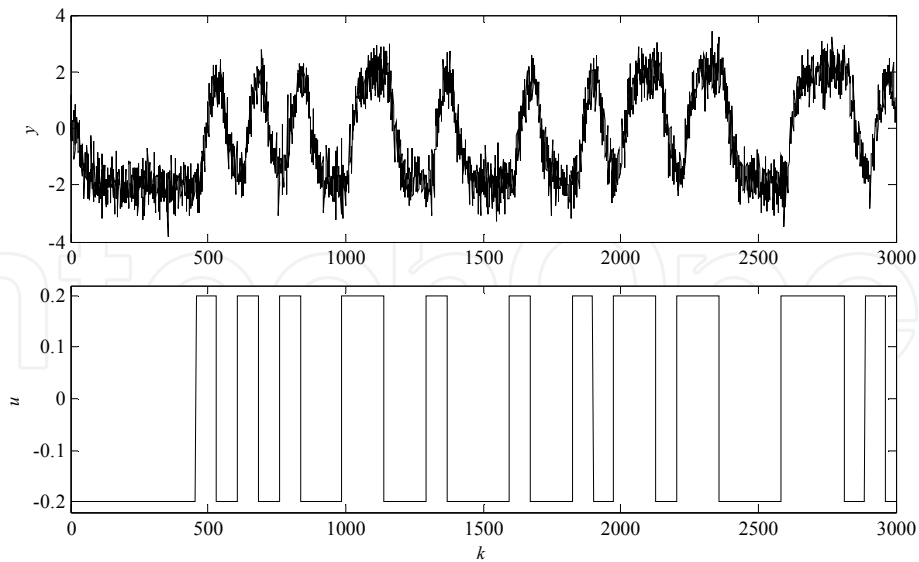
Fig. 1. Input-output data used for identification

To estimate the model parameters the regressor matrix is developed and used together with the plant measured output $y(k)$ in the least square formula (27) to find the model parameters:

[-0.2327   0.8733   -0.2521   0.8854   -0.8767   -0.2357]

The percentage prediction error (PPE) is found to be 9.7357. For the second iteration, the poles of the system are estimated from the noise free step-response of the GOBF model shown in Figure 2.
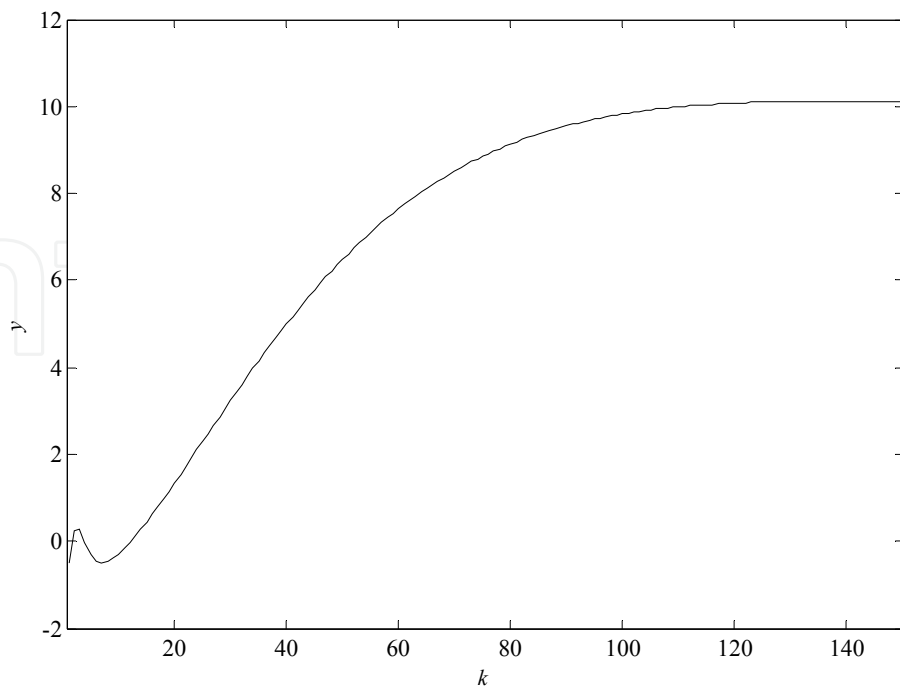

Fig. 2. Step response of the noise free OBF model developed in the first iteration

The OBF model parameters, the PPE and poles estimated at each iteration are presented in Table 1.

Iterations	PPEs	Poles	Model Parameters
1	9.7357	[0.3679 0.9672]	[0.2327 0.8733 -0.2521 0.8854 -0.8767 -0.2357]
2	9.5166	[0.9467 0.9467]	[0.7268 0.7718 0.4069 -0.5214 0.0273 0.0274]
3	9.5149	[0.9499 0.9306]	[0.4992 0.9781 0.4723 -0.3377 0.1387 -0.0305]

Table 1. The results of the OBF iterative identification method

Note that the parameters in the last iteration together with the OBF filters determine the model of the plant. The model accuracy is judged by cross validation. Figure 3 shows the measured output data for sampling instants 3001 to 4000 (this data is not used for modelling) and the result of the OBF simulation for the plant input for the instants 3001 to 4000.

3. BJ- Type models by combining OBF with conventional noise model structures

In Example 1, we developed a GOBF model to capture the deterministic component of the plant. The residual of the model however was just discarded. In reality, this residual may contain useful information about the plant. However, as it is already noted, conventional OBF models do not include noise models. Patwardhan and Shah (2005) showed that the regulatory performance of MPC system improves significantly by including a noise model to the OBF simulation model. In their work, the residual of the OBF model is whitened with
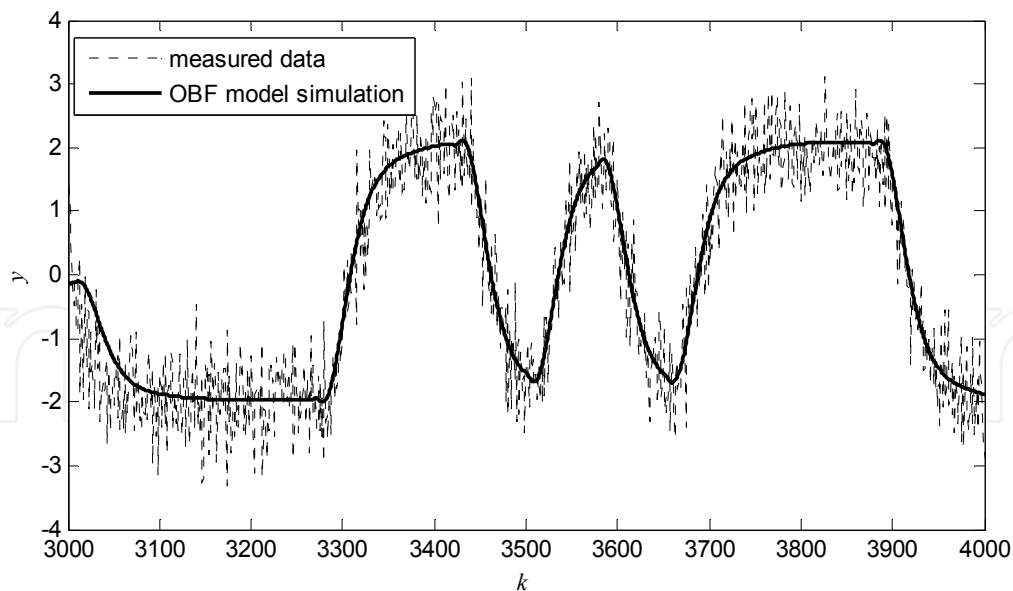

Fig. 3. Validation of the final GOBF model with 6 parameters.

Auto Regressive (AR) noise model. The AR noise model is parameterized in terms of OBF parameters and a minimal order state space model was realized. In this section, an integrated approach for developing BJ models with an OBF plant model and AR or ARMA noise model is presented.

3.1 Model structures

The BJ model structure is known to be the most flexible and comprehensive structure of the conventional linear models(Box & Jenkins, 1970).

$$y(k) = \frac{B(q)}{F(q)} u(k) + \frac{C(q)}{D(q)} e(k) \tag{28}$$

In (28) $B(q)/F(q)$ describes the plant model whereas $C(q)/D(q)$ describes the noise model. The BJ-type model structure proposed by Lemma, *et al.*, (2010) is obtained by replacing the plant model structure with OBF model structure. First, the OBF-AR structure, *i.e.*, with $C(q)$=1 is discussed then the OBF-ARMA structure is discussed.

The OBF-AR model structure assumes an OBF and AR structures for the plant and noise transfer functions, respectively.

$$y(k) = G_{OBF}(q)u(k) + \frac{1}{D(q)} e(k) \tag{29}$$

The OBF-ARMA structure has more flexible noise model than the OBF-AR structure as given by (30).

$$y(k) = G_{OBF}(q)u(k) + \frac{C(q)}{D(q)} e(k) \tag{30}$$

3.2 Estimation of model parameters

The model parameters of both OBF-AR and OBF-ARMA structures are estimated based on the prediction error method as explained below.

Estimation of parameters of OBF-AR model

The prediction error $e(k)$ is defined as

$$e(k) = y(k) - \hat{y}(k \mid k-1) \tag{31}$$

Introducing the prediction error (31) in (29) and rearranging leads to

$$\hat{y}(k \mid k - 1) = D(q)G_{OBF}(q)u(k) + \left(1 - D(q)\right) y(k) \tag{32}$$

Assuming that the noise sequence is uncorrelated to the input sequence, the parameters of the OBF model can be estimated separately. These parameters can then be used to calculate the OBF simulation model output using (32).

$$y_{obf}(k) = G_{OBF}(q)u(k) \tag{33}$$

Inserting (33) in (32)

$$\hat{y}(k \mid k - 1) = D(q)y_{obf}(k) + \left(1 - D(q)\right) y(k) \tag{34}$$

Equation (34) is linear in parameters since $y_{obf}(k)$ is already known. With $D(q)$ monic, (34) can be expanded and rearranged to yield

$$\hat{y}(k\,|\,k-1) = y_{obf}(k) - d_1 r(k-1) - d_2 r(k-2) - \ldots - d_n r(k-n)$$ (35)

where

$n\,is$ the order of the polynomial $D(q)$

$r(i) = y(i) - y_{obf}(i)$

Note that $r(i)$ represents the residual sequence of the output sequence $y(k)$ of the system from the OBF model output $y_{obf}(k)$. The model parameters in (35) can be calculated by the linear least square formula (27) with the regressor matrix given by (36).

$$X = \begin{bmatrix} y_{obf}(n) & -r(n-1) & -r(n-2) - & \ldots & -r(1) \\ y_{obf}(n+1) & -r(n) & -r(n-1) - & \ldots & -r(2) \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ y_{obf}(N) & -r(N-1) & -r(N-2) - & \ldots & -r(N-n) \end{bmatrix}$$ (36)

where $n = n_D$.

The step-by-step procedure for estimating the OBF-AR model parameters, explained above, is outlined in Algorithm 2.

Algorithm 2

1. Develop a parsimonious OBF model
2. Determine the output sequence of the OBF model $y_{obf}(k)$ for the corresponding input sequence $u(k)$
3. Determine the residuals of the OBF model $r(k) = y(k) - y_{obf}(k)$
4. Develop the regression matrix X given by (36)
5. Determine the parameters of the noise model using (27) enforcing monic condition, i.e., $d_0 = 1$.

Estimation of parameters of OBF-ARMA model

The OBF-ARMA structure is given by (28)

$$y(k) = G_{OBF}(q)u(k) + \frac{C(q)}{D(q)}e(k)$$ (28)

Substituting the prediction error (31) in (28) and rearranging yields

$$C(q)\hat{y}(k\,|\,k-1) = D(q)G_{OBF}(q)u(k) - D(q)y(k) + C(q)y(k)$$ (37)

As in the case of OBF-AR model, if the noise sequence is uncorrelated with the input sequence, the OBF model parameters can be calculated separately and be used to calculate the simulation model output $y_{obf}(k)$ using (33).

Introducing (33) in (37) results in

$$C(q)\hat{y}(k \mid k-1) = D(q)y_{obf}(k) - D(q)y(k) + C(q)y(k) \tag{38}$$

Expanding and rearranging (37) we get

$$\hat{y}(k \mid k-1) = y_{obf}(k) - d_1 r(k-1) - d_2 r(k-2) - \ldots - d_m r(k-m) + \\ c_1 e(k-1) + c_2 e(k-2) + \ldots + c_n e(k-n) \tag{39}$$

The parameter vector and the regressor matrix are derived from (39) and are given by (40) and (41)

$$\theta = [d_1\ d_2 \ldots d_m c_1\ c_2\ \ldots\ c_n]^T \tag{40}$$

where $n = n_C$, the order of the polynomial $C(q)$

$m = n_D$, the order of the polynomial $D(q)$

$mx = max\ (m,\ n)+1$

$$X = \begin{bmatrix} y_{obf}(mx) & -r(mx-1) & -r(mx-2) - \ldots - r(mx-n) & e(mx-1) & e(mx-2) \ldots e(mx-m) \\ y_{obf}(mx+1) & -r(mx) & -r(mx-1) - \ldots - r(mx-n+1) & e(mx) & e(mx-1) \ldots e(mx-m+1) \\ \cdot & \cdot & \cdot & \cdot & \cdot \quad \cdot \quad \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \quad \cdot \quad \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \quad \cdot \quad \cdot \\ y_{obf}(N) & -r(N-1) & -r(N-2) - \ldots & -r(N-n+1)\ e(N-1)\ e(N-2) \ldots & e(N-m+1) \end{bmatrix} \tag{41}$$

$$y = [y(mx)\quad y(mx+1)\ \ldots\ y(N)]^T \tag{42}$$

Equation (39) in the form shown above appears a linear regression. However, since the prediction error sequence, $e(k\text{-}i)$, itself is a function of the model parameters, it is nonlinear in parameters. To emphasize the significance of these two facts such structures are commonly known as pseudo-linear(Ljung, 1999; Nelles, 2001). The model parameters can be estimated by either a nonlinear optimization method or an extended least square method (Nelles, 2001). The extended least square method is an iterative method where the prediction error sequence is estimated and updated at each iteration using the prediction error of OBF-ARMA model. A good initial estimate of the prediction error sequence is obtained from the OBF-AR model. The parameters for the noise model are estimated using the linear least square method with (40) and (41) as parameters vector and regressor matrix, respectively. From the derivation, it should be remembered that all the poles and zeros of the noise models should be inside the unit circle and both the numerator and denominator polynomials should be monic. If an OBF-AR model with a high-order noise model can be developed, the residuals of the OBF-AR model will generally be close to white noise. In such

cases, the noise model parameters of the OBF-ARMA model can be estimated using linear least square method in one step. The step-by-step procedure for estimating OBF-ARMA model parameters is outlined in Algorithm 3.

Algorithm 3

1. Develop a parsimonious OBF model
2. Determine the OBF simulation model output $y_{\text{obf}}(k)$ for the corresponding input sequence $u(k)$
3. Determine the residual of the simulation model $r(k) = y(k) - y_{\text{obf}}(k)$
4. Develop OBF-AR prediction model
5. Determine the residual of the OBF-AR model, $\hat{e}(k)$
6. Use $y_{\text{obf}}(k)$, $r(k)$ and $e(k) \approx \hat{e}(k)$ to develop the regressor matrix (40)
7. Use the linear least square formula (27) to estimate the parameters of the OBF ARMA model
8. Re-estimate the prediction error $e(k) = y(k) - \hat{y}(k)$ from the residual of OBF-ARMA model developed in step 7
9. Repeat steps 6 to 8 until convergence is achieved

Convergence criteria

The percentage prediction error (PPE) can be used as convergence criteria, *i.e.,* stop the iteration when the percentage prediction error improvement is small enough.

$$PPE = \frac{\sum_{k=1}^{n}(y(k) - \hat{y}(k))^2}{\sum_{k=1}^{n}\left((y(k) - \overline{y}(k))\right)^2} \times 100$$

where $\overline{y}$ represents the mean value of measurements $\{y(k)\}$ and $\hat{y}(k)$ predicted value of $y(k)$.

3.3 Multi-step ahead prediction

Multi-step ahead predictions are required in several applications such as model predictive control. In this section multi-step ahead prediction equation and related procedures for both OBF-AR and OBF-ARMA are derived.

Multi-step ahead prediction using OBF-AR model

Using (33) in (29) the OBF-AR equation becomes

$$y(k) = y_{obf}(k) + \frac{1}{D(q)}e(k) \tag{43}$$

i-step ahead prediction is obtained by replacing k with $k + i$

$$y(k + i) = y_{obf}(k + i) + \frac{1}{D(q)}e(k + i) \tag{44}$$

To calculate the *i*-step ahead prediction, the error term should be divided into current and future parts as shown in (45).

$$y(k+i) = y_{obf}(k+i) + \frac{F_i(q)}{D(q)}e(k) + E_i(q)e(k+i)$$ (45)

The last term in (45) contains only the future error sequence which is not known. However, since *e(k)* is assumed to be a white noise with mean zero, (45) can be simplified to

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i) + \frac{F_i(q)}{D(q)}e(k)$$ (46)

F_i and E_i are determined by solving the Diophantine equation (47) which is obtained by comparing (44) and (45)

$$\frac{1}{D(q)} = E_i(q) + \frac{q^{-i}F_i(q)}{D(q)}$$ (47)

Equation (46) could be taken as the final form of the *i*-step ahead prediction equation. However, in application, since *e(k)* is not measured the equation cannot be directly used. The next steps are added to solve this problem.

Rearranging (43) to get

$$\frac{1}{D(q)}e(k) = y(k) - y_{obf}(k)$$ (48)

Using (48) in (46) to eliminate *e(k)*

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i) + F_i(q)(y(k) - y_{obf}(k))$$ (49)

Rearranging (49)

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i)(1 - F_i(q)q^{-i}) + F_i(q)y(k)$$ (50)

Rearranging the Diophantine equation (47)

$$\left(1 - q^{-i}F_i(q)\right) = D(q)E_i(q)$$ (51)

Using (51) in (50)

$$\hat{y}(k+i\,|\,k) = E_i(q)D(q)y_{obf}(k+i) + F_i(q)y(k)$$ (52)

Equation (52) is the usable form of the multi-step ahead prediction equation for the OBF-AR model. Given an OBF-AR model, the solution of the Diophantine equation to get E_i and F_i and the prediction equation (52) forms the procedure for *i*-step ahead prediction of the OBF-AR model.

Multi-step ahead prediction using OBF-ARMA model

Using (33) in (30) the OBF-ARMA equation becomes

$$y(k) = y_{obf}(k) + \frac{C(q)}{D(q)}e(k) \tag{53}$$

i-step ahead prediction is obtained by replacing k with $k + i$

$$y(k+i) = y_{obf}(k+i) + \frac{C(q)}{D(q)}e(k+i) \tag{54}$$

To calculate the i-step ahead prediction, the error term should be divided into current and future parts.

$$y(k+i) = y_{obf}(k+i) + \frac{F_i(q)}{D(q)}e(k) + E_i(q)e(k+i) \tag{55}$$

Since $e(k)$ is assumed to be a white noise with mean zero, the mean of $E_i(q)\,e(k+i)$ is equal to zero, and therefore (55) can be simplified to

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i) + \frac{F_i(q)}{D(q)}e(k) \tag{56}$$

F_i and E_i are determined by solving the Diophantine equation (57) which is obtained by comparing (54) and (56)

$$\frac{C(q)}{D(q)} = E_i(q) + \frac{q^{-i}F_i(q)}{D(q)} \tag{57}$$

Rearranging (57)

$$\frac{1}{D(q)}e(k) = \frac{1}{C(q)}\Big(y(k) - y_{obf}(k)\Big) \tag{58}$$

Using (58) in (56) to eliminate $e(k)$

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i) + \frac{F_i(q)}{C(q)}\Big(y(k) - y_{obf}(k)\Big) \tag{59}$$

Rearranging (59)

$$\hat{y}(k+i\,|\,k) = y_{obf}(k+i)\left(1 - \frac{F_i(q)q^{-i}}{C(q)}\right) + \frac{F_i(q)}{C(q)}y(k) \tag{60}$$

Rearranging the Diophantine equation (60)

$$\left(1 - \frac{q^{-i}F_i(q)}{C(q)}\right) = \frac{D(q)E_i(q)}{C(q)} \tag{61}$$

Using (61) in (60) results in the final usable form of the *i*-step ahead prediction for OBF-ARMA model.

$$\hat{y}(k+i \mid k) = \frac{E_i(q)D(q)}{C(q)} y_{obf}(k+i) + \frac{F_i(q)}{C(q)} y(k)$$ (62)

Since $y_{obf}(k+i)$ is the output sequence of the simulation OBF model, if the OBF model parameters are determined its value depends only on the input sequence $u(k+i)$. Therefore, the *i*-step ahead prediction according to (62) depends on the input sequence up to instant $k+i$ and the output sequence up to instant k.

Multiple-Input Multiple-Output (MIMO) systems

The procedures for estimating the model parameters and *i*-step ahead prediction can be easily extended to MIMO systems by using multiple-MISO models. First, a MISO OBF model is developed for each output using the input sequences and the corresponding orthonormal basis filters. Then, AR model is developed using $y_{obf}(k)$ and the residual of the OBF simulation model. The OBF-ARMA model is developed in a similar manner, with an OBF model relating each output with all the relevant inputs and one ARMA noise model for each output using Algorithm (Lemma, et al., 2010).

Example 2

In this simulation case study, OBF-AR and OBF-ARMA models are developed for a well damped system that has a Box-Jenkins structure. They are developed with various orders and compared within themselves and with each other. The system is represented by (63). Note that both the numerator and denominator polynomials of the noise model are monic and their roots are located inside the unit circle.

$$y(k) = q^{-6} \frac{1 - 1.3q^{-1} + 0.42q^{-2}}{1 - 2.55q^{-1} + 2.165q^{-2} - 0.612q^{-3}} u(k) + \frac{1 + 0.6q^{-1}}{1 - 1.15q^{-1} + 0.58q^{-2}} e(k)$$ (63)

An identification test is simulated on the system using MATLAB and the input–output sequences shown in Figure 4 is obtained.
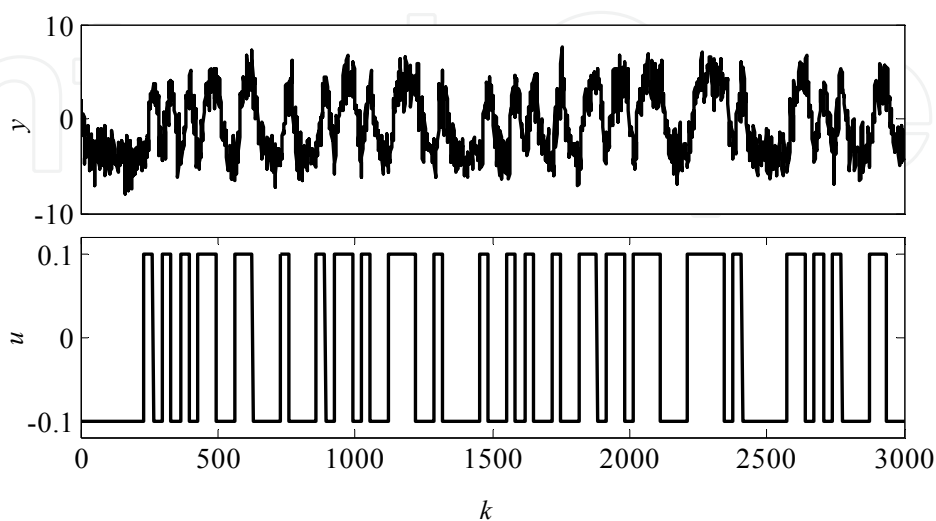

Fig. 4. Input-output data sequence generated by simulation of (63)

The mean and standard deviations of the white noise, $e(k)$, added to the system are 0.0123 and 0.4971, respectively, and the signal to noise ratio (SNR) is 6.6323 . The input signal is a pseudo random binary signal (PRBS) of 4000 data points generated using the *'idinput'* function in MATLAB with band [0 0.03] and levels [-0.1 0.1]. Three thousand of the data points are used for model development and the remaining 1000 for validation. The corresponding output sequence of the system is generated using SIMULINK with a sampling interval of 1 time unit.

OBF-AR model

First a GOBF model with 6 parameters and poles 0.9114 and 0.8465 is developed and the model parameters are estimated to be [3.7273 5.6910 1.0981 -0.9955 0.3692 -0.2252] using Algorithm 1. The AR noise model developed with seven parameters is given by:

$$\frac{1}{D(q)} = \frac{1}{1 - 1.7646q^{-1} + 1.6685q^{-2} - 1.0119q^{-3} + 0.5880q^{-4} - 0.3154q^{-5} + 0.1435q^{-6} - 0.0356q^{-7}} \quad (64)$$

The spectrum of the noise model of the system compared to the spectrum of the model for 3, 5 and 7 parameters is shown in Figure 5. The percentage predication errors of the spectrums of the three noise models compared to spectrum of the noise model in the system is given in Table 2.

n_D	PPE
3	54.3378
5	1.5137
7	0.9104

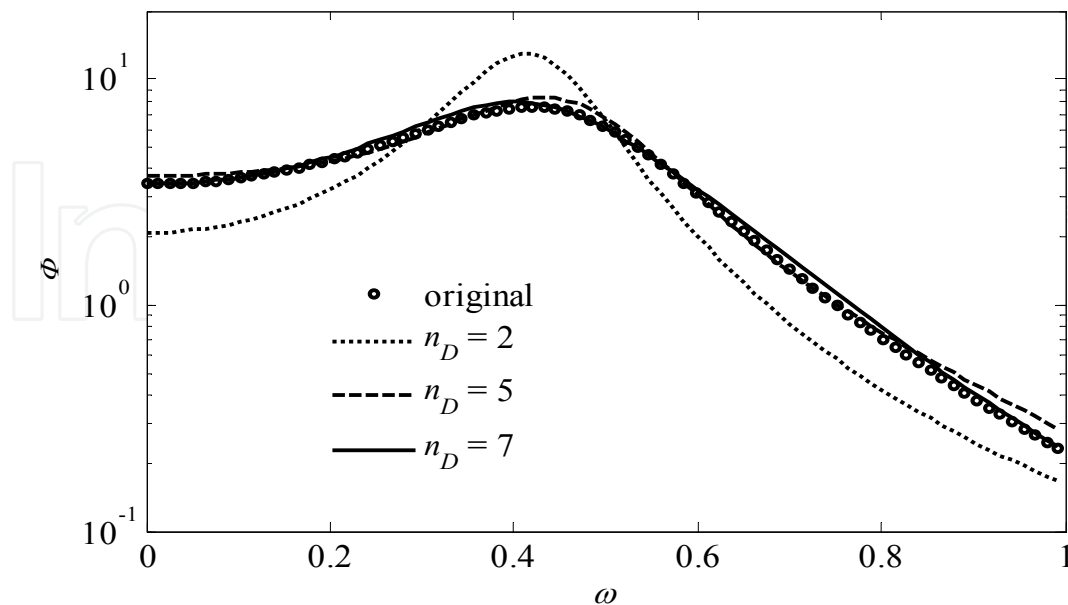Table 2. PPE of the three AR noise models of system

Fig. 5. Spectrums of the AR noise models for n_D = 2, 5 and 7 compared to the noise transfer function of system

It is obvious from both Figure 4 and Table 2 that the noise model with $n_D = 7$ is the closest to the noise transfer function of the system. Therefore, this noise model together with the GOBF model described earlier form the OBF-AR model that represent the system.

4. Closed loop identification using OBF-ARX and OBF-ARMAX structures

When a system identification test is carried out in open loop, in general, the input sequence is not correlated to the noise sequence and OBF model identification is carried out in a straight forward manner. However, when the system identification test is carried out in closed loop the input sequence is correlated to the noise sequence and conventional OBF model development procedures fail to provide consistent model parameters.

The motivation for the structures proposed in this section is the problem of closed-loop identification of open-loop unstable processes. Closed-loop identification of open-loop unstable processes requires that any unstable poles of the plant model should be shared by the noise model $H(q)$ otherwise the predictor will not be stable. It is indicated by both Ljung (1999) and Nelles (2001) that if this requirement is satisfied closed-loop identification of open-loop unstable processes can be handled without problem. In this section, two different linear structures that satisfy these requirements and which are based on OBF structure are proposed. While the proposed models are, specially, effective for developing prediction model for open-loop unstable process that are stabilized by feedback controller, they can be used for open-loop stable process also. These two linear model structures are OBF-ARX and OBF- ARMAX structures.

4.1 Closed–loop identification using OBF-ARX model

Consider an OBF model with ARX structure given by (65)

$$y(k) = \frac{G_{OBF}(q)}{A(q)}u(k) + \frac{1}{A(q)}e(k) \tag{65}$$

Rearranging (65)

$$\hat{y}(k \mid k-1) = G_{OBF}(q) - (1 - A(q))y(k) \tag{66}$$

With $A(q)$ monic (66) can be expanded to

$$\hat{y}(k \mid k-1) = G_{OBF}(q) - a_1 y(k-1) - a_2 y(k-2) - a_m y(k-m) \tag{67}$$

Note that, (67) can be further expanded to

$$\begin{aligned}\hat{y}(k \mid k-1) = l_1 u_{f1}(k) + l_2 u_{f2}(k) + ... + l_m u_{fm}(k) - \\ a_1 y(k-1) - a_2 y(k-2) - ... - a_n y(k-n)\end{aligned} \tag{68}$$

Therefore, the regressor matrix for the OBF-ARX structure is given by

$$X = \begin{bmatrix} u_{f1}(mx) & u_{f2}(mx-1)...u_{fm}(mx-m) & -y(mx-1) & -y(nx-2)...-y(mx-n) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ u_{f1}(N) & u_{f2}(N-1)... u_{fm}(N-m) & -y(N-1) & -y(N-2)...-y(N-n) \end{bmatrix} \quad (69)$$

where m = order of the OBF model

n = order of $A(q)$

$mx = \max(n, m) + 1$

u_{fi} = input u filtered by the corresponding OBF filter f_i

The parameters are estimated using (69) in the least square equation (27). Note that in using (27) the size of y must be from mx to N.

4.2 Multi-step ahead prediction using OBF-ARX model

Consider the OBF-ARX model

$$y(k) = \frac{y_{obf}(k)}{A(q)} + \frac{1}{A(q)} e(k) \quad (70)$$

i-step ahead prediction is obtained by replacing k with $k + i$

$$y(k+i) = \frac{y_{obf}(k+i)}{A(q)} + \frac{1}{A(q)} e(k+i) \quad (71)$$

To calculate the i-step ahead prediction, the noise term can be divided into current and future parts.

$$y(k+i) = \frac{y_{obf}(k+i)}{A(q)} + \frac{F_i(q)}{A(q)} e(k) + E_i(q)e(k+i) \quad (72)$$

Since $e(k)$ is assumed to be a white noise with mean zero, the mean of $E_i(q)\,e(k+i)$ is equal to zero (72) can be simplified to

$$\hat{y}(k+i \mid k) = \frac{y_{obf}(k+i)}{A(q)} + \frac{F_i(q)}{A(q)} e(k) \quad (73)$$

On the other hand rearranging (71)

$$y(k+i) = \frac{y_{obf}(k+i)}{A(q)} + e(k+i)\left(\frac{q^{-i}F_i(q)}{A(q)} + E_i(q) \right) \quad (74)$$

Comparing (70) and (73), F_i and E_i can be calculated by solving the Diophantine equation.

$$\frac{1}{A(q)} = E_i(q) + \frac{q^{-i}F_i(q)}{A(q)} \tag{75}$$

Rearranging (70)

$$\frac{1}{A(q)}e(k) = y(k) - \frac{y_{obf}(k)}{A(q)} \tag{76}$$

Using (76) in (73) to eliminate $e(k)$

$$\hat{y}(k+i\,|\,k) = \frac{y_{obf}(k+i)}{A(q)} + F_i(q)\left(y(k) - \frac{y_{obf}(k)}{A(q)}\right)$$

$$= y_{obf}(k+i)\left(\frac{1}{A(q)} - \frac{q^{-i}F_i(q)}{A(q)}\right) + F_i(q)y(k) \tag{77}$$

Rearranging the Diophantine equation (76)

$$E_i(q) = \frac{1}{A(q)} - \frac{q^{-i}F_i(q)}{A(q)} \tag{78}$$

Finally using (78) in (77), the usable form of the *i*-step ahead prediction formula, (79), is obtained.

$$\hat{y}(k+i\,|\,k) = E_i(q)y_{obf}(k+i) + F_i(q)y(k) \tag{79}$$

Note that in (79), there is no any denominator polynomial and hence no unstable pole. Therefore, the predictor is stable regardless of the presence of unstable poles in the OBF-ARX model. It should also be noted that, since $y_{obf}(k+i)$ is the output sequence of the simulation OBF model, once the OBF model parameters are determined its value depends only on the input sequence $u(k+i)$. Therefore, the *i*-step ahead prediction according to (79) depends on the input sequence up to instant $k+i$ and the output sequence up to instant k.

4.3 Closed–loop identification using OBF-ARMAX model

Consider the OBF model with ARMAX structure

$$y(k) = \frac{G_{OBF}(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}e(k) \tag{80}$$

Rearranging (80)

$$\hat{y}(k\,|\,k-1) = G_{OBF}(q) - (1 - A(q))y(k) + (C(q) - 1)e(k) \tag{81}$$

With $A(q)$ and $C(q)$ monic, expanding (74)

$$\hat{y}(k \mid k-1) = l_1 u_{f1}(k) + l_2 u_{f2}(k) + ... + l_m u_{fm}(k) +$$
$$- a_1 y(k-1) - a_2 y(k-2) - ... - a_n y(k-n) +$$
$$c_1 e(k-1) + c_2 e(k-2) + ... + c_n e(k-n) \qquad (82)$$

From (83) the regressor matrix is formulated for orders m, n, p

$$X = \begin{bmatrix} u_{f1}(mx) & u_{f2}(mx-1)...u_{fm}(mx-m) & -y(mx-1) & -y(nx-2)...-y(mx-n) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ u_{f1}(N) & u_{f2}(N-1)...\,u_{fm}(N-m) & -y(N-1) & -y(N-2)...-y(N-n) \end{bmatrix}$$

$$\begin{bmatrix} -e(mx-1) - e(mx-2)...-e(mx-p) \\ . & . & . \\ . & . & . \\ . & . & . \\ -e(N-1) & -e(N-2)...-e(N-p) \end{bmatrix} \qquad (83)$$

where m = order of the OBF model

n = order of the $A(q)$

p = order of $C(q)$

mx = max (n, m, p) + 1

u_{fi}= input u filtered by the corresponding OBF filter f_i

$e(i)$ = the prediction error

To develop an OBF-ARMAX model, first an OBF-ARX model with high $A(q)$ order is developed. The prediction error is estimated from this OBF-ARX model and used to form the regressor matrix (83). The parameters of the OBF-ARMAX model are, then, estimated using (83) in (27). The prediction error, and consequently the OBF-ARMAX parameters can be improved by estimating the parameters of the OBF-ARMAX model iteratively.

Multi-step ahead prediction using OBF-ARMAX model

A similar analysis to the OBF-ARX case leads to a multi-step ahead prediction relation given by

$$\hat{y}(k+i \mid k) = \frac{E_i(q)}{C(q)} y_{obf}(k+i) + \frac{F_i(q)}{C(q)} y(k) \qquad (84)$$

where F_i and E_i are calculated by solving the Diophantine equation

$$\frac{C(q)}{A(q)} = E_i(q) + \frac{q^{-i} F_i(q)}{A(q)} \qquad (85)$$

When OBF-ARMAX model is used for modeling open-loop unstable processes that are stabilized by a feedback controller, the common denominator $A(q)$ that contains the unstable pole does not appear in the predictor equation, (84). Therefore, the predictor is stable regardless of the presence of unstable poles in the OBF-ARMAX model, as long as the noise model is invertible. Invertiblity is required because $C(q)$ appears in the denominator. It should also be noted that, since $y_{obf}(k+i)$ is the output sequence of the OBF simulation model, once the OBF model parameters are determined its value depends only on the input sequence $u(k+i)$. Therefore, the i-step ahead prediction according to (84) depends on the input sequence up to instant $k+i$ and the output sequence only up to instant k.

5. Conclusion

OBF models have several characteristics that make them very promising for control relevant system identification compared to most classical linear models. They are parsimonious compared to most conventional linear structures. Their parameters can be easily calculated using linear least square method. They are consistent in their parameters for most practical open-loop identification problems. They can be used both for open-loop and closed-loop identifications. They are effective for modeling system with uncertain time delays. While the theory of linear OBF models seems getting matured, the current research direction is in OBF based non-linear system identification and their application in predictive control scenario.

6. Acknowledgement

7. References

Akçay, H. (1999). Orthonormal basis functions for modelling continuous-time systems, *Signal Processing,* pp. 261–274.

Badwe, A. S. , Patwardhan, S. C. & Gudi, R. D. (2011). Closed-loop identification using direct approach and high order ARX/GOBF-ARX models, *Journal of Process Control,* Vol. 21, pp. 1056– 1071.

Box, G.E.P. , Jenkins, G.M. (1970). *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.

Camacho, E.F. & Bordon, C. (2004). *Model Predictive Control*, Springer Verlag Limited, London.

Dewilde, P., Vieira, A.C. & Kailath, T. On a generalised Szegö–Levinson realization algorithm for optimal linear predictors based on a network synthesis approach, *IEEE Transactions on Circuits and Systems,Vol.* CAS-25 , No. 9, 663–675.

Finn, C.K., Wahlberg, B. & Ydstie, B.E. (1993). Constrained predictive control using orthogonal expansions, *AIChE Journal*, vol. 39 pp. 1810–1826.

Gáspár, P., Szabó, Z. & Bokor, J. (1999). Closed-loop identification using generalized orthonormal basis functions, *Proceedings of the 38th Conference on Decision & Control,* Phoenix, Arizona USA, December 1999.

Heuberger, P. S. C. , Van den Hof, P. M. J. & Bosgra, O. H. (1995). A Generalized Orthonormal Basis for Linear DynamicalSystems, *IEEE Transactions On Automatic Control,* vol. 40, pp. 451-465.

Heuberger, P. S. C., Van den Hof, P. M. J. & Wahlberg, B. (2005). *Modeling and Identification with Rational Orthogonal Basis Functions*, Springer-Verlag Limited, London.

Lemma, D. T., Ramasamy (2011), M., Closed-loop identification of systems with uncertain time delays using ARX–OBF structure, *Journal of Process control*, Vol. 21, pp. 1148–1154.

Lemma, D.T. , Ramasamy, M. , Shuhaimi, M. (2011). Improved Method for Development of Parsimonious Otrthonormal Basis Filter Models, *Journal of Process Control, Vol* 21, pp. 36-45.

Lemma, D.T. , Ramasamy, M., & Shuhaimi, M. (2009). Closed Loop Identification Using Orthonormal Basis Filter (OBF) and Noise Models, *Proceedings of AIChE Annual Meeting*, Nasville, USA, November 2009.

Lemma, D.T. , Ramasamy, M., Patwardhan, S.C., Shuhaimi, M.(2010). Development of Box-Jenkins type time series models by combining conventional and orthonormal basis filter approaches, *Journal of Process Control, Vol.* 20, pp. 108–120.

Ljung, L. (1999). *System Identification: Theory for the User,* Prentice Hall PTR, New Jersey.

Merched, R. & Sayed, A. H.(2001). RLS-Laguerre lattice adaptive filtering: Error feedback, normalized, and array-based algorithms, *IEEE Transactions on Signal Processing,* vol. 49, pp. 2565–2576, 2001.

Nelles, O. (2001). *Nonlinear System Identification*. Springer-Verlag, Berlin Heidel Berg.

Ninness, B. M., and Gustafsson, F. (1997). A unifying construction of orthonormal bases for system identification, *IEEE Transactions on Automatic Control,* vol. 42, pp. 515– 521.

Patwardhan, S. C. , Manuja , S., Narasimhan , S.S. & Shah, L. (2006). From data to diagnosis and control using generalized orthonormal basis filters. Part II: Model predictive and fault tolerant control, Journal of Process Control, vol. 16, pp. 157-175.

Patwardhan, S. C., & Shah, S. L. (2005). From data to diagnosis and control using generalized orthonormal basis filters, Part I: Development of state observers, *Journal of Process Control,* vol. 15, pp. 819-835.

Rawlings, J.B. (2000). Tutorial overview of model predictive control. *IEEE Contr. Syst. Mag.* 38–52.

Takenaka, S. (1925). On the orthogonal functions and a new formula of interpolation, *Japanese Journal of Mathematics,* pp. 129–145.

Van den Hof, P.M.J. , Heuberger, P.S.C., Bokor, J. (1995). System identification with generalized orthonormal basis functions, *Automatica,* Vol. 31, pp. 1821–1834.

Van den Hof, P. M. J. , Walhberg, B., Heurberger, P. S. C., Ninness, B., Bokor, J., & Oliver e Silva, T. (2000). Modeling and identification with rational orthonormal basis functions, *Proceedings of IFAC SYSID,* Santa Barbara, California, 2000.

Wahlberg, B. (1991). System Identification using Laguerre filters, *IEEE Transactions on Automatic Control,* vol. 36, pp. 551-562.

Walsh, J. L. (1975). Interpolation and Approximation by Rational Functions in the Complex Domain, *American Mathematical Society Colloquium Publications,* vol. XX.

Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of Stationary Time Series*: M.I.T.-
    Press, Cambridge, MA 1949.