

KING SAUD UNIVERISTY
COLLEGE OF ENGINEERING
RESEARCH CENTER

Final Research Report No. 42/425

Identification of a liquid phase chemical reactor

By

Dr. Emad Ali

9 1425 H
10 2004 G

Table of Contents

<u>Content</u>	<u>Page</u>
1. Introduction	1
2. Physical Process flow sheet and setup	2
3. Implementation Issues	4
3.1 Instrumentation calibration	4
3.2 Interpretation of Results	6
4. System Identification	8
4.1 Data Preprocessing	9
4.2 Model Structure	10
4.3 Identification Analysis	12
4.4 Model Order Selection	14
5. Experimental Tests	17
5.1 ARX model	17
5.2 ARMAX model	20
5.3 OEM Model	20
Conclusion	22
Nomenclature	23
References	24
Appendix	25

List of Tables

Page

Table 1: NAOH flow stream

4

Table 2: Ethyl acetate flow stream

4

Table 3: concentration calibration

6

List of Figures	Page
Figure 1: Schematic of the experimental set up	34
Figure 2: NaOH flow calibration	35
Figure 3: Ethyl acetate flow calibration	35
Figure 4: Conductivity-concentration calibration	36
Figure 5: Experiment data for $F_{\text{NaOH}} = F_{\text{ethyleacetete}} = 300 \text{ ccm}$	36
Figure 6a: ARX model using $n = 2$ and $m = 2$	37
Figure 6b: Zoom-in into Figure 6	37
Figure 7: The model accuracy metrics versus the model order	38
Figure 8: ARX model using $n = 5$ and $m = 5$	38
Figure 8b: zoom-in into Figure 8	39
Figure 9: model residuals; (a) $n = m = 2$, (b) $n = m = 5$	39
Figure 10: model residual versus process output measurement;	40
Figure 11: model residual vs input measurement	40
Figure 12: model residual auto-correlation; (a) $n = m = 2$; (b) $n = m = 5$	41
Figure 13: model residual process input cross-correlation	41
Figure 14: Probability density of the model residuals; dashed line	42
Figure 15: Impulse response of the model: $n = m = 2$	42
Figure 16: Impulse response of the model: $n = m = 5$	43
Figure 17: Step response of the model: $n = m = 2$	43
Figure 18: Step response of the model: $n = m = 5$	44
Figure 19: Poles of the model	44
Figure 20: Impulse response for ARX model with $n = 10, m = 2$	45
Figure 21: Step response of ARX model with $n = 10, m = 2$	45
Figure 22: Experimental Data at Feed flow rate of 500 ccm (dashed); 700 ccm (solid)	46
Figure 23: ARX Model validation with different set of data	46
Figure 24: ARMAX model accuracy measures versus the model order	47
Figure 25: ARMAX model prediction; (a) $n = m = 1 = 5$, (b) $n = m = 1 = 15$	47
Figure 26: poles of the ARMAX models	48
Figure 27: OE model using $n = m = 5$	48
Figure 28: validating OE model using original data, (a) flow = 300 ccm; (b) flow = 700 ccm.	49

Acknowledgment

The investigator would like to thank the general directorate of research center at the college of engineering for their support, continuous follow up and patience throughout the project duration. I would like to emphasize the benefit and advantage of this program, which enables the scientist and scholars to manifest their research ideas into realistic application.

ABSTRACT

This work dealt with fitting empirical models to experimental data. The experimental data is obtained from a liquid phase chemical reaction taking place in a CSTR. The chemical reaction is between sodium hydroxide and ethyl acetate to form sodium acetate and ethanol. The collected experimental data consists of molar concentration of sodium hydroxide and sodium acetate which is obtained from the conductivity measurements and the feed flow rates. The feed flow rates are obtained from the flow meter readings mounted on the feed lines. Both signals, i.e. concentration and flow rates, are sampled every second and transmitted to PC via data acquisition system. The measured NaOH concentration and feed flow rates are fitted to three types of empirical models namely, ARX, ARMAX and OEM. The ARX model was found to fit well to the experimental data even for low order. However, the dynamic properties of the resulted model may not be satisfactory especially for control application. Higher order ARX-model was found to provide better dynamic properties but delivered aggressive response during cross validation. Although ARMAX models are believed to provide better model prediction particularly in the presence of noise, it is found not useful for our case because the resulted model suffers from instability. To obtain a stable ARMAX model careful selection of the model order is necessary. Furthermore, it is found that OEM model can not be fitted to the experimental data because the input signals, i.e. the feed flow rates, are not persistently excited. This means that the input signals should be in the form of periodical waves whereas simple step changes were used in the experiments.

الملخص

هذا البحث يتطرق الى دراسة تمثيل بيانات معملية بنموذج رياضي تجريبي. تستمد البيانات المعملية من عملية تفاعل كيميائي في الطور السائل تتم داخل مفاعل الخلط المستمر حيث يتفاعل كل من هيدروكسيد الصوديوم و اثيل اسيتايت لإنتاج أسيتات الصوديوم و الكحول الإيثيلي. البيانات المعملية تحتوي على التركيز المولاري لهيدروكسيد الصوديوم و اسيتات الصوديوم و التي تستنبط من قراءات التوصيل الأيوني و كذلك على قياس معدل التدفق للقيم. تعمل الأجهزة الكهربيه على تجميع القراءات المعملية بمعدل كل ثانيه و ارساله الى الحاسب الآلي عبر كرت تجميع المعلومات. يركز البحث على تمثيل العلاقة بين التركيز المولاري لهيدروكسيد الصوديوم مع معدل التدفق باستخدام ثلاثة انواع من النماذج الرياضيه التجريبيه المعروفة. و قد اُخت النتائج الى ان النموذج (ARX) الأفضل في تمثيل التركيز المولاري و لكن خواصه الديناميه سيئه و غير مقبوله في حاله إستخدامها في عمليات التحكم. و يمكن تحسين الخواص الديناميه بزيادة حجم هيكل النموذج المقترح و لكن أوضحت دراسة التحقق المعاكس الى أن هذا النموذج قد يؤدي الى استجابته ديناميه غير مستقرة. كما اوضحت الدراسة الى ان استخدام النموذج (ARMAX) غير مجدي بسبب عدم استقراريه الأستجابة الزمنية و للحصول على استقراريه لابد من دقة ضبط معاملات حجم الهيكل. أما بالنسبه للنموذج (OEM) فقد وجد أنه غير مناسب لتمثيل البيانات المعملية و ذلك لأن قراءات تدفق المدخلات غير منشطة بشكل كافي و ذلك يعني أن السلوك الدينامي للمدخلات لابد أن تكون على شكل موجات دوريه من أجل تنشيط الأستجابته الزمنية للمخرجات بحيث تجعل القراءات المعملية غنيه بالمعلومات.

1. INTRODUCTION

The objective of this work is to identify the dynamic behavior of a CSTR reactor using measured data. The concept of fitting a model to experimental data is well known as system identification. The field of system identification is well developed [1]-[6] and has several applications in the chemical engineering industry. Among these applications is process where most of the advanced control algorithms such as Model Predictive Control (MPC) are based on step response models that are derived directly from empirical dynamic models. Therefore, the success of such advanced control systems depends on the accuracy of such models [7]. The developing of successful models for chemical industrial processes is a challenging task because of the cross interaction and multivariable nature of the chemical processes. In control community, system (process) identification became one of the most active branches in the last three decades. An astonishing fact is that most of the identification results developed in the last 30 years are not used by industrial control engineers, although there is an urgent need for efficient and effective identification methods in process control industry. For this reason, there is a renewed interest in closed-loop identification and control-relevant identification [8]. Therefore, this project aims at developing a procedure and understanding of identifying black-box (empirical) models from experimental data. In particular, model structures such as AutoRegressive with eXternal input (ARX), AutoRegressive Moving Average with eXternal input (ARMAX), and Output Error Models (OEM) are considered in this study because they are the widely used ones in practice. The challenging part of identifying such model is determining the model structure and model order. There are several techniques for selecting the model order however it is still a challenging task that involves eminent experience and extensive trial and error procedure [9].

2. PHYSICAL PROCESS FLOWSHEET and SETUP

The existing process is shown in fig. 1. As shown by the figure, the process consists of two feed tanks (reservoir) which contain the solution needed for the chemical reaction. The solution is pumped to head tanks from which it is fed to the reactor through proper tubing. The head tanks are used to eliminate fluctuations in feed flow rate caused by direct pumping to the reactor. The flow rate is adjusted by very elementary flow meters. Homogeneity of the mixture inside the reactor is maintained via electrical stirrer. An analog elementary conductivity meter is used to measure the solution concentration at suitable intervals of time. Typically a chemical reaction between ethyl acetate and sodium hydroxide to produce ethyl alcohol and sodium acetate is carried out in the reactor.



The proposed upgrade of this process involves installing the following instrumentation:

- Two digital flow sensors
- Two electrical control valves
- One digital conductivity meter
- One Type-k Thermocouple
- Data acquisition system
- Personal computer

The function of the flow sensor, conductivity meter, and thermocouple is to continuously and accurately measure the transient progress of the major process variables such as inlet flow rates, concentration of the product and reaction temperature respectively. The measured data is processed by the data acquisition system and transmitted to the personal computer. At the computer, the transmitted data can then be graphically visualized, analyzed and stored for future manipulations. These features will dramatically facilitate the experiment operation and will help the user to easily manipulate and analyze the results.

A multi channel data acquisition system is used as interface between the various process sensors and the computer. The main function of the data acquisition is to process all measured data and transfer them between the computer and the process actuators back and forth. A 330 MHz Pentium II personal computer is used in this experiment. The personal computer is for data logging, processing, analysis and manipulation. The measured variables in this experiment are the feed flow rates and the conductivity. These recorded data will be utilized later for the process identification.

3. IMPLEMENTATION ISSUES

3.1 INSTRUMENT CALIBRATION

The basic instrument used in the experiment transmits the measured variable in terms of voltage signals. The signal range is 0-5V. This is the common communication media between the data acquisition card and digital computers. For more scientific application, it is more useful to convert these signals into suitable engineering units. In this experiment, the fluid flow measurement, temperature measurement and the salt concentration measurement should be converted (calibrated) into their corresponding units.

3.1.1 Flow Sensor Calibration:

First the fluid flow measurement is calibrated. This was achieved through recording the voltage reading produced by the flow sensor and the corresponding flow rate reading provided by the regular flow meter. Different readings were taken to cover the available span for the proportional valve opening. The valve opening is proportional to the analog input voltage applied to it. Table 1 shows the flow calibration for the NaOH stream. Table 2 shows the flow calibration for the ethyl acetate stream.

Table 1: NAOH flow stream

NaOH Flow		
Volt	High (ml/s)	Low (ml/s)
0	0	0
1	1.3899	0.16
2	10.9649	6.0606
3	12.6904	9.5831
4	12.7226	12.1507
5	12.9366	12.8617

Table 2: Ethyl acetate flow stream

Ethyl Acetate Flow		
Volt	High (ml/s)	Low (ml/s)
0	0	0
1	0.1942	0.0952
2	14.8699	0.5359
3	20	1.487
4	20.02	16.8067
5	20.5285	18.1818

The first column in these tables denotes the voltage applied to the control valve, which is proportional to the valve opening. The other columns correspond to the calculated flow rate in ml/sec. The "high" column denotes the calculated flow rate when the valve opening starts from maximum value of 5 volt and goes down to the minimum of 0 volt. The "Low" column corresponds to the other situation, i.e. starting from 0 volts and goes up to 5 volt. It is interesting that the flow rate differs with the opening scheme, i.e. upward (Low) or downward (high). In addition, the flow rate responds nonlinearly to the valve opening (voltage). This is also clear from the plot of the flow rate versus the voltage as shown in Fig. 2 and 3. It is also obvious that the valve for the ethyl acetate delivers more flow than that of the NaOH. This can be attributed to many factors such as the hardware settings of each valve, flow line resistance, pump strength, fluid density and other practical issues.

Since the flow-volt readings do not seem to be linear, a linear fitting cannot be approximated. However, this does not create problem for the open loop tests because the flow rate of each stream will be fixed during the experiments. The calibrated flow sensors are as follows:

$$F_{\text{NaOH}} \text{ (ml/L)} = 0.334S \text{ (Volt)} + 2.917 \quad (1)$$

$$F_{\text{Ethyl acetate}} \text{ (ml/L)} = 4.42S \text{ (Volt)} - 1.65 \quad (2)$$

3.1.2 Salt Concentration Calibration:

The dissolved salt conductivity is measured by a bench-top conductivity meter. The conductivity is given in the unit of Siemens. The measured conductivity of the salt should be thus converted into concentration units of gmole/L. Different sample of brine solution with known concentration are used to calibrate the conductivity meter. The conductivity of each sample is measured as listed in Table 3 and a plot for the measured conductivity versus the salt concentration is shown in Fig. 4. Clearly the concentration-conductivity relation is linear. Therefore, the salt concentration can be inferred directly from the measured conductivity according to:

$$C \text{ (gmole/L)} = 0.0046G \text{ (ms/cm)} - 0.0005 \quad (3)$$

Table 3: concentration calibration

mS/cm	C (gmole/l)
0.0116	0
3.06	0.01
5.23	0.025
10.08	0.05
16.9	0.075
22.4	0.1
27	0.125

3.2 INTERPRETATION OF RESULTS

Having used the bench-top conductivity meter to record the conductivity of the contents of the reactor over the period of reaction, the conductivity measurements must be translated into degree of conversion of the constituents.

Both sodium hydroxide and sodium acetate contribute conductance to the reaction solution whilst ethyl acetate and ethyl alcohol do not. The conductivity of a sodium hydroxide at a given concentration and temperature however, is not the same as that of a sodium acetate solution at the same molarities and temperature and a relationship has been established allowing conversion to be inferred from conductivity.

The sodium hydroxide concentration, A and sodium acetate concentration, C can be calculated alongside the readings of conductivity using the following equations [10]-[12]:

$$C_A = (C_{A\infty} - C_{Ao}) \left[\frac{\Lambda_o - \Lambda}{\Lambda_o - \Lambda_\infty} \right] + C_{Ao} \quad (4)$$

$$C_C = C_{C\infty} \left[\frac{\Lambda_o - \Lambda}{\Lambda_o - \Lambda_\infty} \right], \text{ for } C_{Co} = 0 \quad (5)$$

Consequently, the reaction conversion at any time can be then calculated as follows:

$$X_a = \frac{C_{A0} - C_A}{C_{A0}} \quad (6)$$

$$X_C = \frac{C_C}{C_{C\infty}} \quad (7)$$

4. SYSTEM IDENTIFICATION

System identification is the experimental approach to process modeling. Basically, process identification involves fitting the experimental input-output data into empirical model of predefined structure. In industrial application models developed by identification is preferred over the first-principle model, i.e. models developed using the mass and energy balances. For example, the empirical models can be easily developed by applying the well-known regression analysis. Moreover, it does not require detailed knowledge of the process physical parameters.

System identification includes the following steps

- ***Experiment design***: its purpose is to obtain good experimental data, and it includes the choice of the measured variables and of the character of the input signals. Usually, input signals should be designed such that it provides enough process excitation. Collected data should be filtered to remove noise and de-trended to remove nonlinearity.
- ***Selection of model structure***: A suitable model structure is chosen using prior knowledge and trial and error. The latter will be used in this work.
- ***Choice of the criterion to fit***: A suitable cost function is chosen, which reflects how well the model fits the experimental data. Instead, the
- ***Parameter estimation***: The above cost function is solved using optimization software to obtain the numerical values of the model parameters. MATLAB is used here to solve the optimization problem.
- ***Model validation***: The model is tested in order to reveal any inadequacies.

4.1 DATA PREPROCESSING

Collected plant data may not be ready for immediate use because:

- High frequency disturbances (noise)
- Occasional burst and outliers (spikes) and missing data.
- Drift and offset and low frequency disturbances

In this case, processing of Data is an important prerequisite for the estimation phase. It may involve repair of the data in terms of replacing missing or obviously wrong data as well as merging disjoint data sets. This stage may involve data polishing by removing undesired disturbance features in the data.

Data Detrending

It is customary to deal with drift and offset by averaging the data:

One way to use deviation variables:

$$y(t) = y^m(t) - \bar{y} \quad (8)$$

$$u(t) = u^m(t) - \bar{u} \quad (9)$$

$$\bar{y} = \frac{1}{N} \sum y^m(t) \quad (10)$$

$$\bar{u} = \frac{1}{N} \sum u^m(t) \quad (11)$$

Another way is to use difference variable:

$$y(t) = y^m(t) - y^m(t-1) \quad (12)$$

Data differencing will be used in this work and it also helps in removing low frequency noise.

High frequency noise can be removed prefiltering. However, this is not considered here because filtering usually tends to attenuate the model gain, and in extreme cases can change the dynamics. The resulting filtered model will not match the real plant.

Moreover, the data collected for the output is free of noise because RS232 connection is used to interface the conductivity meter with the PC.

4.2 MODEL STRUCTURE

We will try three different model structures in this work. These model structures are well known in the field of model identification and are discussed in the following.

4.2.1 ARX Models (AutoRegressive with eXternal input)

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n) = b_1u(k-1) + \dots + b_mu(k-m) + e(k) \quad (13)$$

$$(1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n})y(k) = (b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m})u(k) + e(k) \quad (14)$$

$$A(z)y(k) = B(z)u(k) + e(k) \quad (15)$$

$$A = 1 + a_1 + a_2 + \dots + a_n \quad (16)$$

$$B = b_1 + b_2 + \dots + b_m \quad (17)$$

The idea is to find the model parameter a 's and b 's using the experimental measurements. The parameter estimation can be easily solved because it can be formulated as least squares problem.

4.2.2. ARMAX Models (AutoRegressive Moving Average with eXogenous input)

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n) = b_1u(k-1) + \dots + b_mu(k-m) + e(k) + c_1e(k-1) + \dots + c_l e(k-l) \quad (18)$$

$$(1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n})y(k) = (b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m})u(k) + (1 + c_1z^{-1} + c_2z^{-2} + \dots + c_lz^{-l})e(k) \quad (19)$$

$$A(z)y(k) = B(z)u(k) + C(z)e(k) \quad (20)$$

$$A = 1 + a_1 + a_2 + \dots + a_n \quad (21)$$

$$B = b_1 + b_2 + \dots + b_m \quad (22)$$

$$C = 1 + c_1 + c_2 + \dots + c_l \quad (23)$$

The idea is to find the model parameter a 's, b 's and c 's using the experimental measurements. Because ARMAX possesses an extended noise filter $C(z)$, it is more

flexible than the ARX model when noise and unmeasured disturbance are present in the process. Note that ARMAX simplifies to ARX when $C(z) = 1$. However, the model is nonlinear in its parameters which require using nonlinear optimization to determine its parameter values. Convergence problem may arise

4.2.3 OEM Models (Output Error Models)

$$y(k) = \frac{B(z)}{F(z)}u(k) + e(k) \quad (24)$$

$$\hat{y}(k/k-1) + f_1\hat{y}(k-1) + f_2\hat{y}(k-2) + \dots + f_n\hat{y}(k-n) = b_1u(k-1) + b_2u(k-2) + \dots + b_mu(k-m) \quad (25)$$

$$\hat{y}(k/k-1) = -f_1\hat{y}(k-1) - f_2\hat{y}(k-2) - \dots - f_n\hat{y}(k-n) + b_1u(k-1) + b_2u(k-2) + \dots + b_mu(k-m) \quad (26)$$

Which can be summarized as:

$$\hat{y}(k/k-1) = \varphi(k)^T \theta \quad (27)$$

With

$$\varphi(k) = [-\hat{y}(k-1) \quad -\hat{y}(k-2) \quad \dots \quad -\hat{y}(k-n) \quad u(k-1) \quad u(k-2) \quad \dots \quad u(k-m)]^T$$

$$\theta = [f_1 \quad f_2 \quad \dots \quad f_n \quad b_1 \quad b_2 \quad \dots \quad b_m]$$

The OEM model is nonlinear in its parameters which require using nonlinear optimization to determine its parameter values. Convergence and local minima problems may arise.

Comparison between ARX and OEM models:

- The ARX model depends on the past measurements of the inputs and outputs.
- The OEM model depends on the past measurement of the inputs only.
- The ARX always remain close to the process output because it is based on the process output and thus can not run-away from the process.
- The price that ARX model pays is the introduction of the disturbance into the prediction since the process output is corrupted by noise.
- The OEM model requires initial condition of the past n simulated outputs. Usually they are set to zero.
- OEM are good choice when disturbance models are not to be fitted,

4.3 IDENTIFICATION ANALYSIS (model validation)

After the model has been developed, it is necessary to assess the accuracy of the obtained model. Several possible methods are available as discussed in the following.

4.3.1 Validation

- Validation is a graphical evaluation of the accuracy of the estimated model.
- It shows how well the model is capable of reproducing the plant data.

Using Training data

The data used for estimating the model parameters is known as the training data and can be used again to simulate the model. The predicted model output is then plotted against the actual data. This is known as *validation*. (fitting)

Using Test data

The test data is another set of data that is used for measuring the performance of the obtained model; i.e. evaluating the quality of the model. Simulating the model using the test data and plotting the result against the actual data is known as *cross-validation*. (validation)

4.3.2 Measure of Accuracy

Alternative to validation, a numerical evaluation of the accuracy of the estimated model can also be investigated.

Sum of squares

Summing the squared deviation of the predicted output from the actual output is a measure of the amount of departure from the true model:

$$MSE = \sqrt{\frac{\sum_{i=1}^N (y(i) - \hat{y}(i))^2}{N - d}} \quad (28)$$

The smaller the MSE, the higher is the closeness of the fit to the true model and vice versa.

Coefficient of determination

The coefficient of determination is the ratio of the squared deviation of the model from the output mean to the total sum of squares:

$$R^2 = \frac{\sum_{i=1}^N (\hat{y}(i) - \bar{y})^2}{\sum_{i=1}^N (y(i) - \bar{y})^2} \quad (29)$$

- As R^2 approach 1, the estimated model approaches the true system.
- The coefficient of determination (R^2) is scalar accuracy test.

4.3.3 Analysis of the residuals

The performance metric used in item 4.2.2 above uses the average of the residuals, which provides a single measure of how well the identified model fit the data. The residual is the deviation of the predicted output from the true output:

$$e_i = y(i) - \hat{y}(i) \quad (30)$$

- It is regarded as the observation error in contrast to the unknown true error ε_i in the regression model:

$$\varepsilon_i = y(i) - E[\hat{y}(i)] \quad (31)$$

- The observation error, e , should then reflect the properties of the true error. This is the basic idea of *residual analysis*.
- The residual will be analyzed through trend plots, auto correlation, and cross correlation

4.4 MODEL ORDER SELECTION

The order selection, i.e. determining n , m , and l of a parametric model can have a significant impact on the identification results.

Techniques for selecting the optimal model order:

- Preliminary data analysis for order selection.
- Alternatively one can optimize the bias/variance tradeoff.
- Training, validating and testing :
 - Split the plant data into three sets namely; training data, validation data and test data.
 - Train say T different complex models with the training data.
 - Choose the model with the lowest error on validation data.
 - Finally the quality of the chosen model is evaluated on a fresh test data.
 - It is also possible to split the plant data into S parts. The first $S - 1$ parts can be used to train the different models, while the last set to select the best model.

- Statistical Test:

- The t-test or F-test can be used to differentiate between two possible models with different complexity.
- First formulate the hypothesis:

$$H_o : \theta_{add} = 0 \quad \text{simple model Ok}$$

$$H_a : \theta_{add} \neq 0 \quad \text{simple model not Ok}$$

- The θ_{add} denotes the vector of additional parameters in the complex model compared to the simple one.
- If the null hypothesis H_o is correct the complex model does not make any use of the additional parameters and thus the simple model should be selected.
- A statistical test such as the chi-squared distribution (χ^2) can be used to accept or reject the hypothesis.

- Correlation-Based Methods:

- Another strategy for testing whether a model is appropriate is to check whether it captures all information contained in the data.
- The correlation between the model error and the model input can be used:

$$corr\{u_i, e_i\} = 0$$

- In practice, the correlation is not exactly zero, and a statistical test can reveal whether the above hypothesis can be accepted.
- However, trial-and-error along with available information criterion is commonly used:

- AIC (Akaike Information Theoretic Criterion) technique:

$$J = \frac{1 + (d/N)}{1 - (d/N)} \frac{1}{N} \sum_{k=1}^N e^2(k) \quad (32)$$

- MDL (Minimum Description Length) technique:

$$J = \left(1 + \frac{\log N}{N}\right) d \frac{1}{N} \sum_{k=1}^N e^2(k) \quad (33)$$

- (BIC) Bayesian Information Criterion:

$$J = N \ln \left[\frac{1}{N} \sum_{k=1}^N e^2(k) \right] + \ln(N)d \quad (34)$$

- LILC Khinchin Law of Iterated Algorithm Criterion:

$$J = N \ln \left[\frac{1}{N} \sum_{k=1}^N e^2(k) \right] + 2\rho \ln(\ln(N))d \quad (35)$$

- FPE Final Prediction Error Criterion:

$$J = N \ln \left[\frac{1}{N} \sum_{k=1}^N e^2(k) \right] + N \ln \left(\frac{N+d}{N-d} \right) \quad (36)$$

- AIK Akaike Information Criterion:

$$J = N \ln \left[\frac{1}{N} \sum_{k=1}^N e^2(k) \right] + \rho d \quad (37)$$

Here $d = \dim \theta$, ρ is an adjustable parameter.

<i>Method</i>	<i>Drawback</i>
Bias/variance tradeoff	These errors can not be estimated easily
Multi-objective optimization	Computationally demanding
Training, validation and testing	Extensive effort
Statistical tests	Test only two models at one time. Not constructive
Information Criteria	Repeated solution, adjustable parameter.
Correlation based methods	Similar to statistical test, they only tell if the model is adequate but not how to change the model in order to make it adequate.

5. EXPERIMENTAL TESTS

5.1 ARX MODEL

To generate necessary process data, the reactor was operated at feed flow rates of 300 ccm. The reactor was initially empty. The two reactor feeds were turned on at fixed flow rate and the conductivity of the solution inside the reactor is continuously measured and sent directly to the PC. The recorded data is shown in Fig. 5. The figure shows the concentration in moles/l of the reactant NaOH and the product Sodium Acetate. Note that concentration of the reactant and product are generated from the measured conductivity using (4) and (5). Fig. 5 also shows the recorded flow rate over 600 samples. It is obvious that as the reactor fills up with NaOH, the conductivity of NaOH increases. However, in the mean while the sodium hydroxide starts reacting with the Ethyle acetate which results in reduction of the Sodium hydroxide concentration and propagation of the sodium acetate until both of them reach equilibrium as indicated by steady state.

The data in Fig. 5 is used to develop an ARX model. The concentration of NaOH is taken as the output and the feed flow rates as the inputs. Therefore, we need to specify a value for n , the size of past outputs to be included in the model, and two values of m , the size of each past inputs to be included in the model described by (13) & (14). As a start we take $n = 2$ and $m = 2$ for both inputs. The model development, i.e. estimating the parameter values for a and b in (13) & (14) is carried out using MATLAB. Comparison of the developed model with the actual data is shown in Fig. 6a,b. Note that NaOH profile shown in the figures is in difference format (see (12)) while that shown in Fig. 5 is in full format. Fig. 6a shows excellent overall agreement between the model prediction and the actual data, while Fig. 6b indicates that less model accuracy is observed during the early stages of dynamic. Tow types of scalar accuracy measure is also given in Fig. 6a. The first one is the fitness value, i.e. MSE in (28). This metric measures the average error between the model and the true data. The smaller the fitness value is the more accurate the model is. Another metric is the coefficient of determination which measures how close the model to the data. A value close to one indicates excellent model prediction. The AIC value which calculates the average error with respect to the model order is evaluated for the result in Fig. 6.

To determine the optimal model order, several methods are available as discussed in section 4.4. In this work we will rely on the AIC value at different values for the model order. Fig. 7 shows the fitness value, coefficient of determination and AIC value for various values of the model order, N_p . Note that $N_p = n + 2m$. The solid line is for increment of step 10 in the model order, while the dashed line for increment of step 1 in the model order. It is clear the both MSE and AIC have a minima and R has a maxima between $N_p = 10$ and $N_p = 20$. The dashed line indicates that the minima or the maxima can be extended to $N_p > 20$. Fig. 8 illustrates the result for ARX model with $n = m = 5$, i.e. $N_p = 15$. The model at $N_p = 15$ looks spikier at steady state than that at $N_p = 6$, i.e. Fig. 6, whereas the former model should be more accurate than the latter because the model at $N_p = 15$ has smaller values for MSE and AIC. It is evident that increasing n & m would improve the model quality. However, increasing the value of m makes the model prediction noisy because of the noise incorporated in the flow measurement as shown in Fig. 8.

To further inspect the model quality, residual analysis can also be considered. Fig. 9 demonstrates the trend of model-data residuals which provides visual inspection of the model error over all data points in contrast to the MSE which provides only a simple average error. Nevertheless, the plot does not show any particular difference between the models of different order. Similarly, fig. 10 & 11 demonstrates the error-output trend and error-input trend, respectively. Sometimes these plots show particular trends such as concave, convex, inclined/declined line, or drift. When such remarkable trends exist, one can extract some conclusion about the model quality, however, in our case, the residual plots have almost the same shape. Fig. 12 & 13 show the auto-correlation of the error (residuals) and the input-error cross-correlation. Again these type of plots can be helpful as it may indicate bad modeling when the correlation violates the upper or the lower bounds. The upper and lower bounds are the confidence interval usually determined as sigma value corresponding to 95% in the normal distribution. Fig. 14 depicts the probability distribution of the normalized residuals. As the model accuracy increases, the probability distribution should become narrower and taller. We can observe that increasing the model order from 2 to 5 indicated insignificant improvement.

The final decision on the goodness of the model depends on its utilization. If the model to be used for control purposes, then further model properties should be investigated. For example, the model poles, impulse response and step response should be examined. Fig. 15 & 16 show the model impulse response for $N_p = 6$ and $N_p = 15$. The low order model behaves like a process with a lead dynamics which does not truly represent the process dynamics, while the impulse response for the higher model order seems to be more appropriate. Fig. 17 & 18 represent the step response of the model for the same two cases. The step response is useful to determine the process steady state gain and time constant. If prior information is available about the gain and time constant, sound judgment on the model accuracy can be made. However, only the gain direction is known. Accordingly, the model at with $N_p = 15$ is more reliable because it possesses the expected gain direction for the process. Fig. 19 shows the poles of the developed models. For $n = 2$, there are two poles and for $n = 5$ there are five poles. Fortunately, all poles are inside the unit circle which ensures the stability of the generated models. The latter is an essential model property for control application.

For ARX model with noisy input measurement, it might be favorable to increase n to capture the full dynamic and reduce m to eliminate the effect of noisy input. Fig. 20-21 illustrate the modeling result for $n = 10$ and $m = 2$. The resultant model seems to provide reasonable model dynamic characteristics as shown by the impulse and step responses in Fig. 20 and 21. If the process gain and time constant are known precisely, one can further optimize the model order, i.e. the value of n and m .

As discussed in section 4, the model faithfulness can also be checked by cross validation. This means that the model prediction is compared to a fresh set of data other than the training data, i.e. data used for developing the model. For this reason the experiment was conducted at different operating conditions to generate new experimental data. Fig. 22 shows the experimental data at feed flow rate of 500 and 700 ccm, respectively which will be used as validation data. It is obvious from the figure that at higher flow rate, the residence time becomes smaller which prevents the reaction from propagating. This situation of lower reaction conversion leads to higher NaOH concentration and lower Sodium Acetate concentration. Fig. 23 demonstrates how the model prediction is compared to the validation data in difference format. Visual inspection of fig. 23 indicates that all models can reproduce the experimental data

reasonably. However, increasing the model order, i.e. n , the number of model poles increases leading to more aggressive response.

5.2 ARMAX MODEL

In this section, an ARMAX model will be considered. The ARMAX model is described by (18)-(23). The ARMAX model is more useful when noisy measurement is the case. However, this requires determination of an additional design parameter which is the size of past error, l , to be included in the model. Furthermore, the model becomes nonlinear in the parameter as the past errors involve the past measured outputs in nonlinear fashion. Figure 24 show how MSE, R and AIC varies with n . Note that we take $l = m = n$ and consequently the model order becomes $N_p = n + l + 2m$. It is evident from the figure that the model quality varies nonlinearly with the model order. There is no clear trend and hence no conclusion about the suitable model order can be deduced.

Inclusion of the past errors in the model might be risky as unstable model can result. Fig. 25 show the model prediction for $n = m = l = 5$ and $n = m = l = 15$. The figure shows how the model prediction becomes highly oscillatory at these values of model order. For further confirmation, the poles of these two models are calculated and plotted in Fig. 26. It is obvious that for the low model order, the poles are at the edge of stability, i.e. very close to unity. For the high model order, some poles are clearly outside the unit circle indicating unstable model dynamics. Therefore, one must be very careful when designing an ARMAX model. Several simulation and analysis were considered for ARMAX models. We found that ARMAX models provide smoother prediction especially at steady state, however, the size of error, l , must be less than $n/3$ to guarantee stability.

5.3 OEM model

In this section, we try to fit an OEM model for the NaOH concentration as a function of the feed flow rates using the experimental data. The modeling equation is given by (24)-(27). In this case, we need to specify n and m and the model order, therefore, becomes $N_p = 2n + 2m$. Fig. 27 illustrates the OEM model for $n = m = 5$ using experimental data generated at feed flow rate equal 300 ccm. There is no doubt that the OE model can not reproduce the measured output. Different values for the model order were tried, but the OE model was still found to perform poorly. Note that according to (24)-(27), the OE

model depends only on the past *model outputs* and the measured past inputs. This means that, unlike the ARX and ARMAX models, the OEM model does not have a feedback, i.e. it does not involve the *measured output* in the model structure. Despite this fact, the OEM model still can model any process dynamic if the input signal is **persistently** excited. In our case, the input signals are not excited enough. This point needs further investigation. Alternatively, we resort to the original experimental signals instead of the differenced signals. Using the original input-output data, i.e. without differencing, generates improved OEM model as shown in Fig. 28. It should be noted that $n = 1$ is used to avoid incorporating past model outputs and $m = 30$ to involve more past inputs in the model. It is found that larger values for n may not provide good model. Moreover, the model dynamic properties may not necessarily be good. The lower part of Fig. 28 checks the model validity with the experimental data at feed flow rate of 700 ccm. It is obvious that the model performs very poorly because the OEM model responds only to the input transient behavior.

6. CONCLUSIONS

In this work, the measured signal of a CSTR is used to model the process dynamics. The measured signals are; two input flow rates and the reactant concentration, NaOH. Three types of models were considered. The models were developed using the measured data along with MATLAB software. The crucial point in developing these models is to specify appropriate model order. It is found that n (size of past outputs) should be at least 5 in order to obtain a faithful model. Different model validation methods were used to examine the model accuracy among which, cross validation, numerical scalar metrics, and residual analysis. Although these methods were useful, they were not informative as much as the impulse and step responses of the generated model. The latter provided insightful knowledge about the dynamic properties of the model and lead to the founding mentioned above. It is also found that for ARX model, smaller m (the size of past inputs) is preferable to avoid the noise effect. For ARMAX models, care should be taken when determining the model order because wrong choice may lead to unstable model dynamics. The OEM model was found to be the worst among the other models because it failed to predict the even the training data, i.e. data used for generating the model. This failure can be attributed to the lack of output feedback in the model structure. The latter should not create a problem if the input signals were persistently excited which was not the case in our experiment. Further work on this line is needed. For example, we need to study how to create persistently exciting signals such as sine wave or pseudo random binary sequence and applied to the process in order to obtain process data that is rich in dynamic information. In general, ARX model presented promising results for identifying the process dynamic while OEM seems to be sufficient for steady normal operation as it unable to adapt changes in the process.

NOMENCLATURE

a 's	Model coefficient for output measurements
b 's	Model coefficients for input
C_i	Concentration of species i , mole/l
C_{A_0}, C_{A^∞}	Initial and final concentration of A
C_{C_0}, C_{C^∞}	Initial and final concentration of C
d	Number of model parameters
e	Model error
f 's	Model coefficients for output in the OEM model
F_1	Feed flow rate of pure component A , l/min
F_2	Feed flow rate of pure component B , l/min
k	Sampling time
l	Number of past error to be included in the model
m	Number of past inputs to be included in the model
n	Number of past outputs to be included in the model
R	Coefficient of determination
t	Time, min
u	Input signals
X	Reaction conversion
y	Output measurement
\hat{y}	Model output
z	z-domain
Λ	Conductivity measurement at any time
$\Lambda_0, \Lambda^\infty$	Initial and final value for conductivity

References

- [1] D. Montgomery and G. Runger; *Applied Statistics and Probability for Engineers*; Wiley, 1999.
- [2] L. Ljung; *System Identification, Theory for the User*; Prentice-Hall, 1999.
- [3] A. Constantinides and N. Mostoufi; *Numerical methods for Chemical Engineers with MATLAB Applications*, Prentice-Hall, 1999.
- [4] Y.C. Zhu, and T. Backx *Identification of Multivariable Industrial Processes: for Simulation, Diagnosis and Control*. Springer-Verlag, London. 1993.
- [5] O. Nelles, *Nonlinear System Identification*; Springer Verlag, 2001.
- [6] J. P. Norton, *An Introduction to Identification*, Academic Press, 1986.
- [7] Y.C. Zhu, Multivariable process identification for MPC: the asymptotic method and its applications. *Journal of Process Control*, Vol. 8, No. 2, 101-115, 1998.
- [8] Van den Hof, P.M.J. Closed-loop issues in system identification. In *Proceedings of the 11th IFAC Symposium on System Identification*, Fukuoka, Japan, Vol. 4, 1651-1664, 1997.
- [9] Rahul Bindlish, Model order selection for process identification applied to an industrial ethylene furnace, *Journal of Process Control* 13 569–577, 2003.
- [10] A.M. Abu Khalaf, and Ali, E., "Automatic Start-up of A Non-Isothermal CSTR", *Chem. Eng. Ed.*, 34(3), 246-251, 2000.
- [11] A. M. Abu-Khalaf, "Start-Up of a Non-Isothermal CSTR: Mathematical Modeling," *Chem. Eng. Ed.*, 31(4), 250, 1997.
- [12] A.M. Abu-Khalaf, "Mathematical Modeling of an Experimental Reaction System," *Chem. Eng. Ed.*, 28(1), 48, 1994.

APPENDIX

In this section the MATLAB program used for calculation and analysis is included. The programs used are as follows:

Identarx.m

This is the main program which reads the saved rperimantal data, develop the model, perform statistical analysis, and plot the result. The statistical analysis is carried out in a separate function called *statistic.m*. The program assumes the experimental data to be saved in the same directory. When run, the program requests the user to enter the values of the model order, i.e. m , n and l .

Statistic.m

This is an auxiliary function called from *Identarx* mfunction to perform statistical analysis such as residual analysis, correlation, probability distribution, etc.

Arx2step.m

Another auxiliary function that is called from *Identarx* mfunction to create step response, impulse response, calculates poles and zeros, steady state gain, etc.

Validate.m

Another auxiliary function that is used to validate he created model against another set of validation data.

All above program use some built-in MATLAB functions.

%Program Identarx.m

```
clear all
close all

global ny nu
global nout na nb nk nc N st
global th Ts
global nameu namey namee

load y300.dat
y=y300;
load u300.dat
u=u300(:, :);

[Ny ny]=size(y);
[Nu nu]=size(u);

if Nu~=Ny
    error('data length conflict !!!')
end

N=250;
nout=1;
nys=1; ny=1;

na=input('input na = ');
```

```

nb=input('input nb = ');
nk=zeros(1,nu);
Ts=0.1;

np=sum(na)+sum(nb);
dfi=N-np;
df=N-nys*np;
np=np*nys;

namey=cell(1);
nameu=cell(1);
namee=cell(1);

    for i=1:ny
        namey(i)={[ 'y',int2str(i)]};
        namee(i)={[ 'e',int2str(i)]};
    end
    for i=1:nu
        nameu(i)={[ 'u',int2str(i)]};
    end

y=diff(y(:,nout)); y=[zeros(1,ny); y];
u=diff(u); u=[zeros(1,nu);u];
z=[y(1:N,:) u(1:N,:)];

gc=0; pc=0; fc=1; sc=1;
st=[1 1 1 1 1 1]*0;

% Estimate the model parameter
nny=ones(nys,1)*na;
nnu=ones(nys,1)*nb;
nnk=ones(nys,1)*nk;
nn=[nny nnu nnk];

th=arx(z,nn);

% claculate the residuals
[yh]=predict(z,th,1);
for i=1:nys
    e(1:N,i)=yh(1:N,i)-y(1:N,i);
    ss(i)=e(1:N,i)'*e(1:N,i);
    fit(i)=sqrt(ss(i)/dfi);
    et(1:N,i)=y(1:N,i)-mean(y(1:N,i));
    sst(i)=et(1:N,i)'*et(1:N,i);
    ey(1:N,i)=yh(1:N,i)-mean(y(1:N,i));
    ssy(i)=ey(1:N,i)'*ey(1:N,i);
    r(i)=sqrt(max(0,(1-ss(i)/sst(i))));
    r(i)=sqrt(ssy/sst);
end

% AKiak analysis
AIC=(1+np/N)*ss/(1-np/N);
BIC=N*log(ss/N)+log(N)*np;
MDL=(1+log(N)/N)*np*ss/N;
FPE=N*log(ss/N)+N*log((N+np)/(N-np));

% covariance analysis
[PAR,P,LAM]=th2par(th);

```

```

% validate the mdoel with the regression data
figure(3)
loc=nys*100+10;
for i=1:nys
    subplot(loc+i); plot([yh(:,i) z(:,i)]);
    if i==1; title('ARX-fitting; yellow:[model], magenta:[actual]');
end;
if i==nys; xlabel('Sample point'); end;
ylabel(namey(i)); v=axis;
y3=v(3)+0.55*(v(4)-v(3)); x2=v(1)+0.5*(v(2)-v(1));
y1=v(3)+0.95*(v(4)-v(3)); y2=v(3)+0.1*(v(4)-v(3));
text(x2,y1,['fitness value = ',num2str(fit(i))],'Color','k');
text(x2,y2,['Akaik value = ',num2str(AIC(i))],'color','k');
text(x2,y3,['Coef. determination = ',num2str(r(i))],'color','k');
end

% statistical Analysis
[tnsd,tb,stdb,CL]=statistic(e,u(1:N,:),yh,dfi,fit,P,PAR,st,namee,nameu,
,namey);

% additonal options
if max([gc pc fc scl])==1
    arx2step(gc,pc,fc,sc);
end

save modelarx102 th

```

%Program Statistic.m

```

function
[tnsd,tb,stdb,CL]=statistic(e,u,y,df,fit,P,PAR,st,errr,namu,name)

[mn nu]=size(u);
[N ny]=size(e);
[np]=size(PAR,1);
tnsd=1.713;
a=95;
%if df>120; df=120; end
% Calculating confidence interval
%alpha = 1 - a/100;
%infinity = 30;
%t0 = 1; t = 2;
%while abs(t - t0) > 1e-6
%    t0 = t;
%    t = t + (quad('stud',t,infinity,[],[],df) - alpha/2) / stud(t,df);
%end
%tnsd=t;

% Parameter standard deviation and confidence limit
[m mg]=size(P);
sme=mean(fit);
if mg>1;
    stdb=sme*sqrt(diag(P));
else
    stdb=sme*sqrt(P);
end
CL=tnsd*stdb;
tb=PAR'./stdb;

```

```

if max(st)==1

% residual correlation analysis
for i=1:ny
    mu=mean(e(1:N,i))*0;
    ss=cov(e(1:N,i));
    cr(i,1:2)=[mu+tnsd*ss/sqrt(N) mu-tnsd*ss/sqrt(N)];
    Re(i,1:N)=covf(e(:,i),N);
end

% cross correlation of residual against inputs
k=0; ccw=[];
for i=1:ny
    for j=1:nu
        k=k+1;
        ss=cov(e(1:N,i))*cov(u(1:N,i));
        mu=mean(e(1:N,i).*u(1:N,i))*0;
        conf1=mu+tnsd*ss/sqrt(N);
        conf2=mu-tnsd*ss/sqrt(N);
        crr=[ones(N,1)*conf1 ones(N,1)*conf2];
        ccw=[ccw crr];
        dummy=xcov([e(1:N,i),u(1:N,j)], 'biased');
        Reu(1:N,k)=dummy(N:2*N-1,1);
    end
end

for i=1:ny
    se(:,i)=sort(e(1:N,i));
    ss(i)=se(:,i)'*se(:,i)/(N-np);
    Ps(:,i)=exp(-se(:,i).*se(:,i)/2/ss(i))/sqrt(2*pi*ss(i));
end

% plot some result
if st(1)==1;
    figure(4)
    set(4,'name','Residual Plot')
    t=0:N-1; t=t'; loc=ny*100+10;
    for i=1:ny
        subplot(loc+i); plot(t,zeros(N,1),t,e(1:N,i),'.');
ylabel(errr(i));
        v=axis; axis([10 N v(3)/10 v(4)/10]); if i==ny; xlabel('No.
of sample');end
        if i==1; title('Residuals'); end; axis([10 N -4e-1 4e-1]);
    end
end

if st(2)==1;
    figure(5)
    set(5,'name','Correlaion of residuals')
    for i=1:ny
        loc=ny*100+10;
        subplot(loc+i); plot([Re(i,:)' ones(N,1)*cr(i,1)
ones(N,1)*cr(i,2)]);
        v=axis; axis([0 N 2*cr(i,2) 2*cr(i,1)]); if i==ny;
xlabel('No. of sample');end
        ylabel(['corr ',char(errr(i))]); if i==1;title('correlation
of residulas');end
    end
end
end

```

```

if st(3)==1;
figure(6)
set(6,'name','Probability Distribution')
loc=ny*100+10;
for k=1:ny
subplot(loc+k); plot(se(:,k)/sqrt(ss(k)),Ps(:,k))
if k==1; title('Probability distribution of residuals'); end;
ylabel(errr(k)); if k==ny; xlabel('mean'); end
end
end

if st(4)==1;
figure(7)
set(7,'name','Input vs Residual'); kk=0;
for k=1:ny
for j=1:nu
kk=kk+1;
subplot(max(ny,nu),min(ny,nu),kk); plot(u(:,j),e(:,k),'.')
grid on; v=axis; ylabel(errr(k));
%text(0.5*(v(1)+v(2)),v(3)+0.9*(v(4)-v(3)),namu(j));
xlabel(namu(j));
end
end
end

if st(5)==1;
figure(8)
set(8,'name','Output vs Residual');
loc=ny*100+10;
for k=1:ny
subplot(loc+k); plot(y(:,k),e(:,k),'.')
grid on; v=axis; ylabel(errr(k));
%text(0.5*(v(1)+v(2)),v(3)+0.9*(v(4)-v(3)),name(k))
xlabel(name(k));
end
end

if st(6)==1;
figure(9)
set(9,'name','Input-Residual correlation Plot');
k=0; loc=nu*100+ny*10;
for i=1:ny
for j=1:nu
k=k+1;
subplot(nu,ny,k); plot([Reu(:,k) ccw(:,2*k-1:2*k)]);
axis([0 N 2*ccw(1,2*k) 2*ccw(1,2*k-1)]);
ylabel(errr(i)); v=axis;
text(0.5*N,v(3)+0.9*(v(4)-v(3)),namu(j))
end
end
end

end % empty(st) loop

return

%Program arx2step.m
function arx2step(gcal,pcal,fcalscal)

```

```

%(1) Creates the step response model from the ARX model')
%(2) Calculates the steady state gain, poles and zeros')

global ny nu
global nout na nb nk nc N st
global th Ts
global nameu namey namee

n=N; t=0:n-1; t=t';
[A,B]=th2arx(th);

nys=ny;
%
% calculate the gain
%
if gcal==1 | scal==1;
    for i=1:nys
        for k=1:na(i)+1
            a(i,k)=A(i,i+(k-1)*nys);
        end
        for j=1:nu
            for k=1:nb(j)
                b(j,k)=B(i,j+(k-1)*nu);
            end
            gain(i,j)=sum(b(j,:))/sum(a(i,:));
        end
    end
    if gcal==1;
        figure(14); axis('off'); axis([0 1 0 1]);

        set(14,'Color','k','Resize','on','Name','Gain','Units','normal',...
            'menubar','none','numbertitle','off','Position',[0.15 0.25
0.5 0.4]);
        printm=uimenu(14,'Label','Print','Callback','printdlg(gcf)');
        msg={'Steady State Gain Matrix'};

        text(0.25,1.0,msg,'Color','w','Fontweight','bold','FontSize',10,'fo
ntname','Times');

        z=510-100-(nu-1)*5;
        z=fix(z/nu);
        yup=240;
        y=80;
        for j=1:nu
            Box11(j)= uicontrol('Style','text','position',[y yup z 15],...
                'BackgroundColor',[1,0.6,0.8],'string',nameu(j));
            y=y+z+5;
        end
        s=yup-25;
        for i=1:nys
            Box10(i)= uicontrol('Style','text','position',[15 s 50
20],...
                'BackgroundColor',[1,0.6,0.8],'string',namey(i));
            yy=80;
            for j=1:nu
                Box20(i,j)= uicontrol('Style','edit','position',[yy s z
20],...
                    'BackgroundColor',[1,0.8,0.8],'string',num2str(gain(i,j)));
                yy=yy+z+5;
            end
        end
    end
end

```

```

        end
        s=s-25;
    end
end
save fgain.dat gain /ascii
end

%
% creat step response by:
% (1)stepping the ARX-model
% (2)long division of ARX-model
%
HH=[];yy=[];
if scal==1;
    for i=1:nu
        u=zeros(n,nu);
        u(1:n,i)=ones(n,1);
        x=idsim(u,th); yy=[yy x];
        for j=1:nys
            BB=B(j,i:nu:nu*nb(i));
            [h]=impz(BB,a(j,:),n,1);
            HH=[HH h];
        end
    end
    % rearrange step response crated by long division
    for i=1:nys*nu
        S(1,i)=HH(1,i);
    end
    for j=2:n
        for i=1:nu*nys
            S(j,i)=sum(HH(1:j,i));
        end
    end
    % plotting the result
    for i=1:nys
        figure(15+i); loc=nu*100+10;
        for j=1:nu
            subplot(loc+j); plot(t,S(:,i+(j-1)*nys),'-',t,yy(:,i+(j-
1)*nys),'--');
            grid off; axis tight; ylabel(namey(i)); ax=axis;
            xt=ax(1)+0.2*(ax(2)-ax(1)); yt=ax(3)+0.8*abs((ax(4)-ax(3)));
            text(xt, yt, [char(nameu(j)),' solid: long division, dashed:
step test']);
        end
    end
    save fstep.dat yy /ascii
end

%
% zero-pole
%
if pcal==1
    [spole,szero]=pandz(th,nys);
    save fpoles.dat spole /ascii
    save fzeros.dat szero /ascii
end

%
% frequency response
%
if fcal==1

```

```

w=(0:n-1)*2*pi/n;
[A,B,C,D,K,x0]=th2ss(th);
sys=ss(A,B,C,0,Ts);
[mag,phase]=bode(sys,w);
for i=1:nys
    figure(15+nys+2+i);
    for j=1:nu
        clear ym; clear yp;
        ym(:,1)=mag(i,j,:); yp(:,1)=phase(i,j,:);
        subplot(2,2,j); semilogx(w,ym); grid on;
        ax=axis; xt=ax(1)+0.2*(ax(2)-ax(1)); yt=ax(3)+0.8*abs((ax(4)-
ax(3)));
        text(xt,yt,[char(namey(i)),'-',char(nameu(j))]);
        ylabel('magnitude');
        subplot(2,2,j+2); semilogx(w,yp); grid on;
        ax=axis; xt=ax(1)+0.2*(ax(2)-ax(1)); yt=ax(3)+0.8*abs((ax(4)-
ax(3)));
        text(xt,yt,[char(namey(i)),'-',char(nameu(j))]);
        ylabel('phase'); xlabel('frequency');
    end
end
end
end

```

%Program validate.m

```

%
% load validation data
%
load y500.dat
load u500.dat
load y700.dat
load u700.dat

figure(1)
N=400;
subplot(221)
plot(y700(1:N,1),'-'); hold on;
plot(y500(1:N,1),'::'); ylabel('NaOH');
subplot(222)
plot(y700(1:N,2),'-'); hold on;
plot(y500(1:N,2),'::'); ylabel('Sodium Acetate')
subplot(223)
plot(u700(1:N,1),'-'); hold on
plot(u500(1:N,1),'::');
ylabel('Flow, NaOH'); xlabel('Sample')
subplot(224)
plot(u700(1:N,2),'-'); hold on
plot(u500(1:N,2),'::');
ylabel('Flow, Ethyle Acetate'); xlabel('Sample')

N=50;
y=diff(y700(1:N,1)); y=[zeros(1,1); y];
u=diff(u700(1:N,:)); u=[zeros(1,2);u];
[z1]=[y u];
y=diff(y500(1:N,1)); y=[zeros(1,1); y];
u=diff(u500(1:N,:)); u=[zeros(1,2);u];
[z2]=[y u];

figure(2)
t=0:N-1;

```

```
load modelarx2
```

```
[yh]=predict(z1,th,1);  
subplot(321)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
ylabel('NaOH'); text(10, 0.03, 'n = m = 2, u = 500 ccm');  
[yh]=predict(z2,th,1);  
subplot(322)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
ylabel('NaOH'); text(10, 0.05, 'n = m = 2, u = 700 ccm');
```

```
load modelarx5
```

```
[yh]=predict(z1,th,1);  
subplot(323)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
text(10, 0.03, 'n = m = 5, u = 500 ccm'); ylabel('NaOH');  
[yh]=predict(z2,th,1);  
subplot(324)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
text(10, 0.05, 'n = m = 5, u = 700 ccm'); ylabel('NaOH');
```

```
load modelarx102
```

```
[yh]=predict(z1,th,1);  
subplot(325)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
text(5, 0.03, 'n = 10, m = 2, u = 500 ccm'); ylabel('NaOH');  
xlabel('sample')  
[yh]=predict(z2,th,1);  
subplot(326)  
plot(t,yh(1:N,1),t,z1(1:N,1),':');  
text(5, 0.05, 'n = 10, m = 2, u = 700 ccm'); ylabel('NaOH');  
xlabel('sample')
```

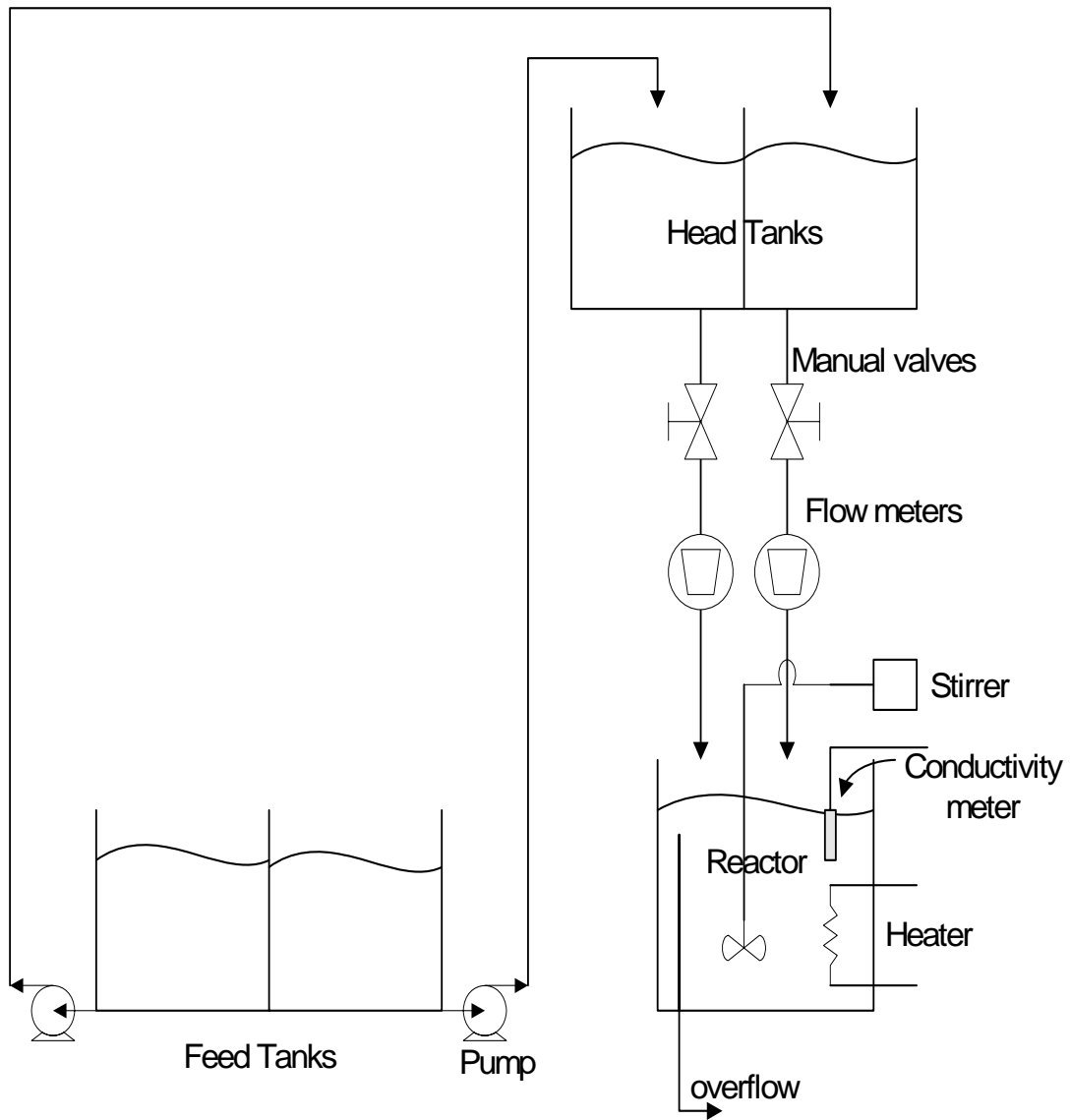


Fig. 1: Schematic of the experimental set up

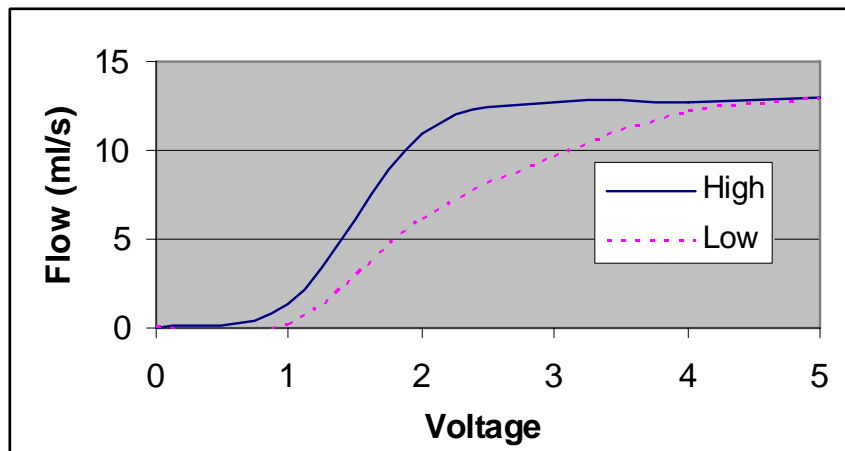


Fig. 2: NaOH flow calibration

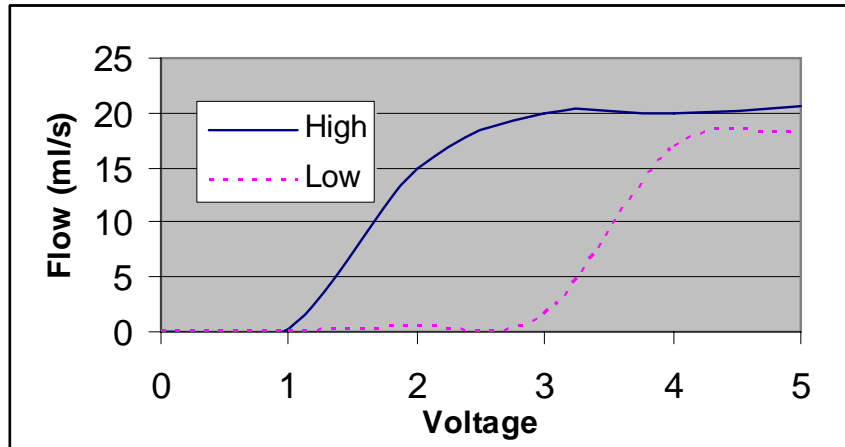


Fig. 3: Ethyl acetate flow calibration

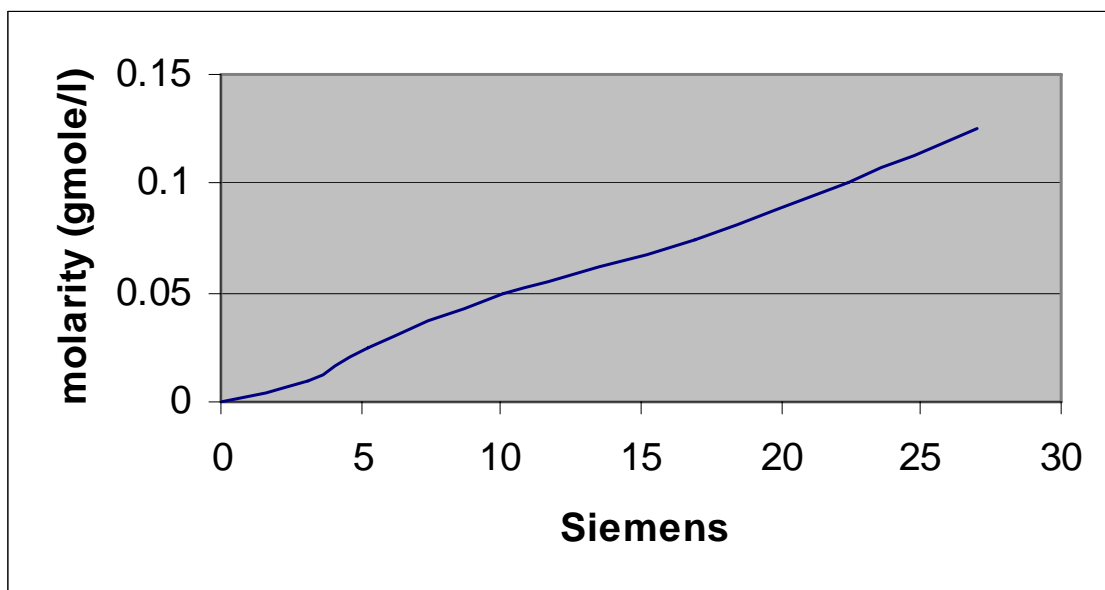


Fig. 4: Conductivity-concentration calibration

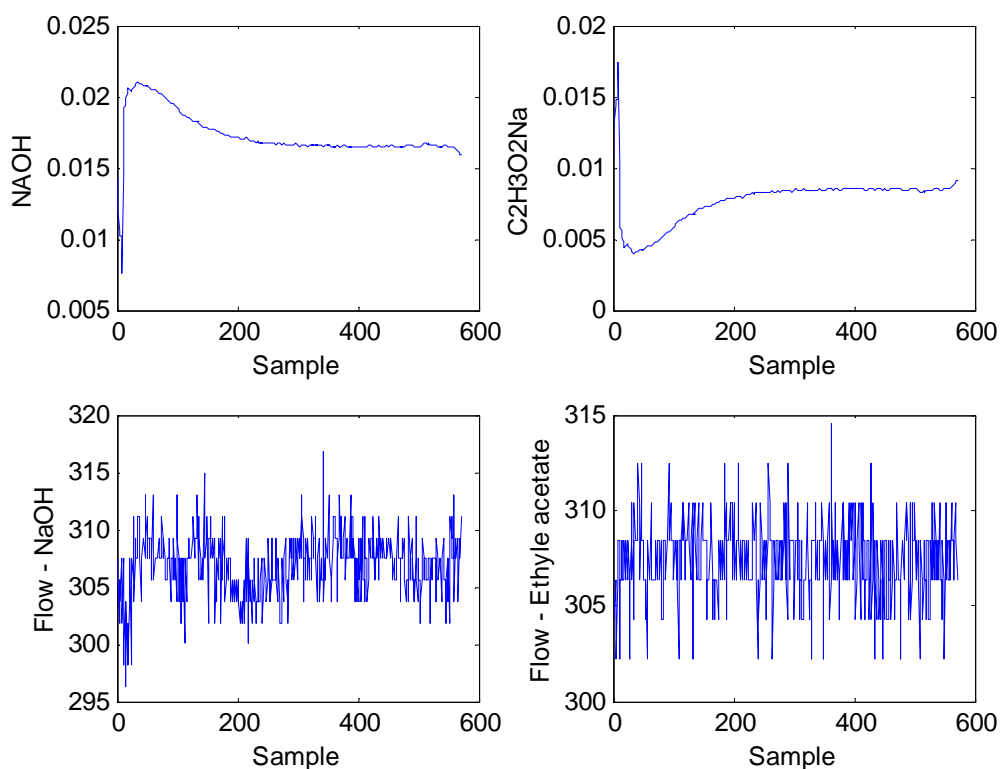


Fig. 5: Experiment data for $F_{\text{NaOH}} = F_{\text{ethyleacetate}} = 300 \text{ ccm}$

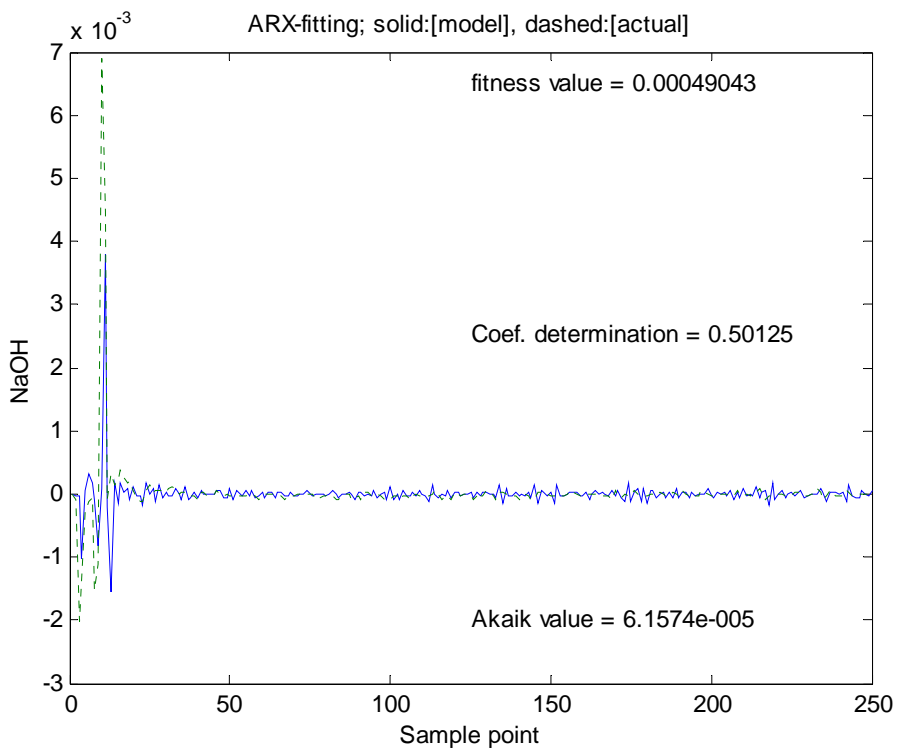


Fig. 6a: ARX model using $n = 2$ and $m = 2$

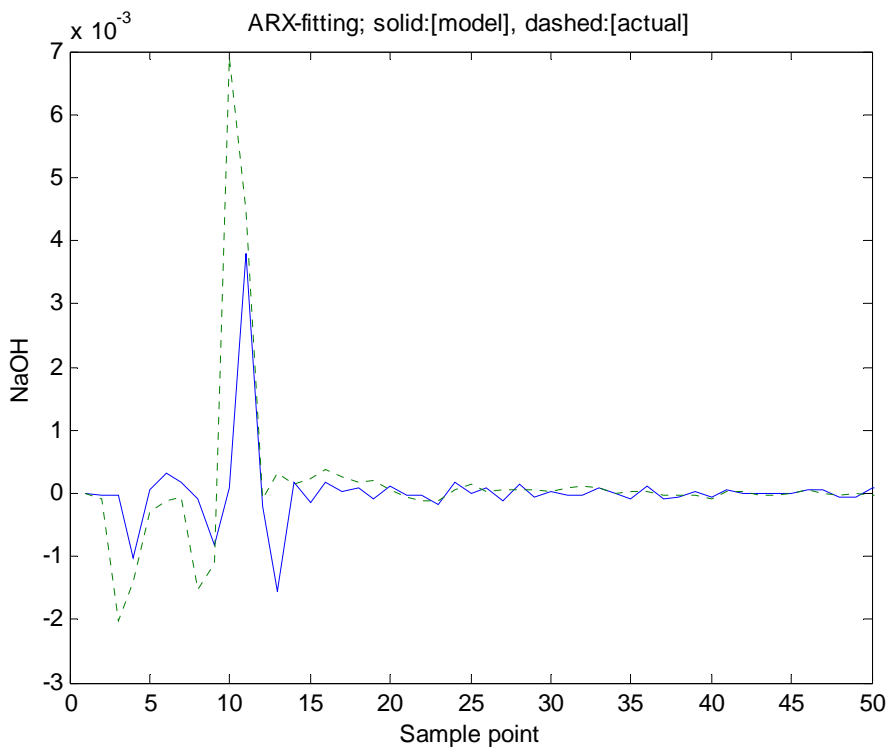


Fig. 6b: Zoom-in into Figure 6

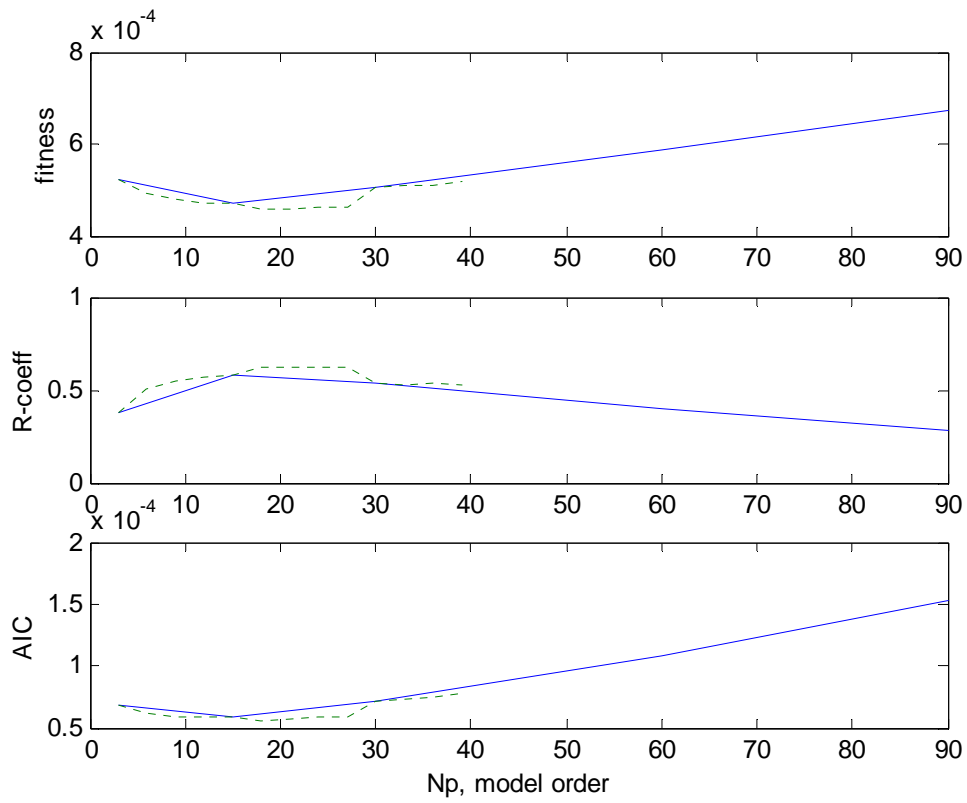


Fig. 7: The model accuracy metrics versus the model order

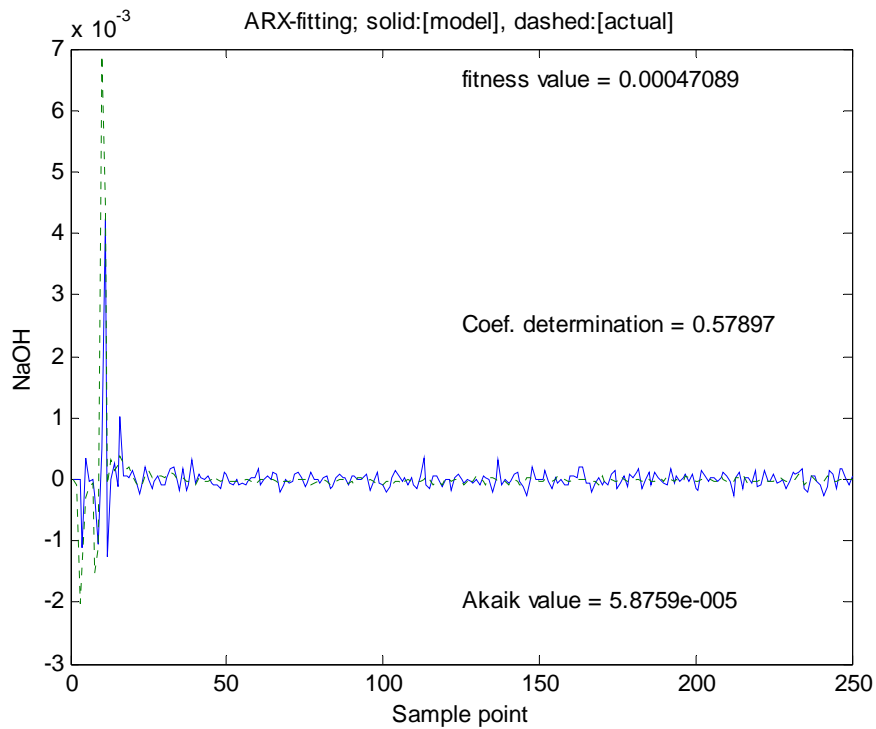


Fig. 8: ARX model using $n = 5$ and $m = 5$

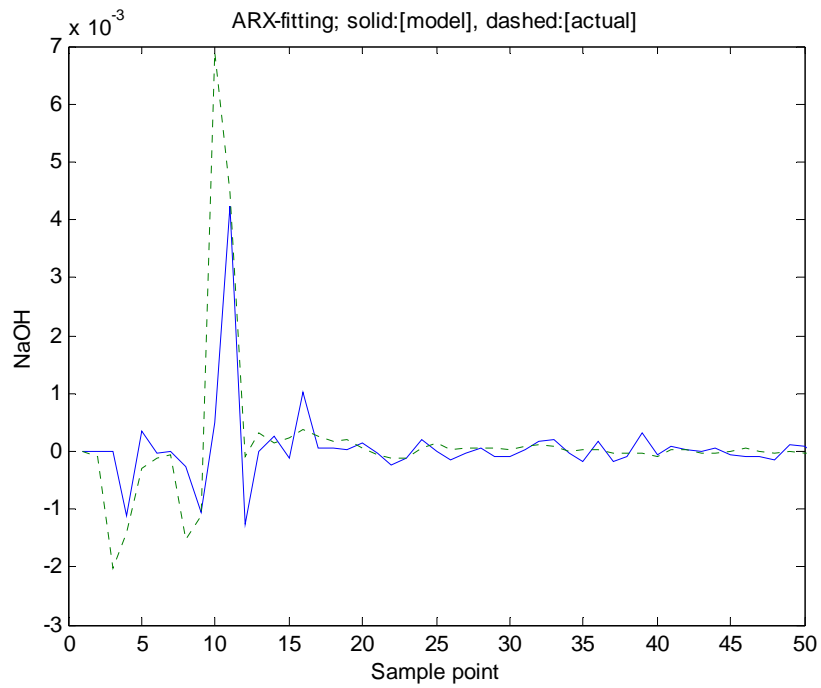


Fig. 8b: zoom-in into Figure 8.

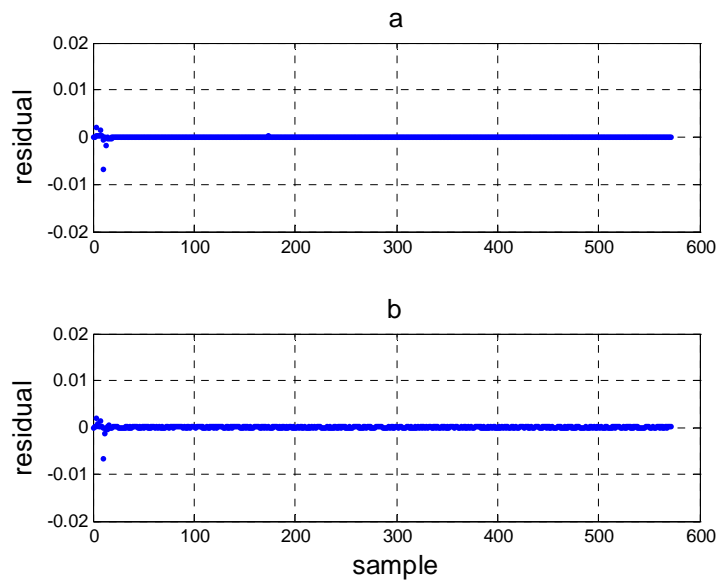


Fig. 9: model residuals; (a) $n = m = 2$, (b) $n = m = 5$

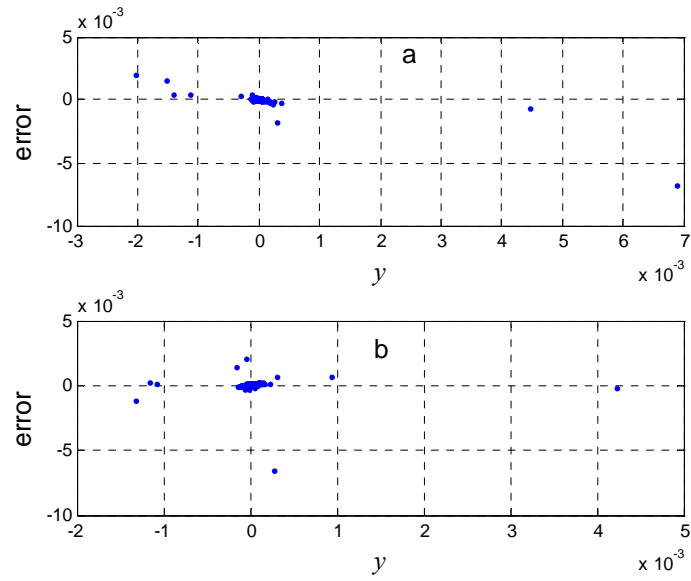


Fig. 10: model residual versus process output measurement; (a) $n = m = 2$; (b) $n = m = 5$

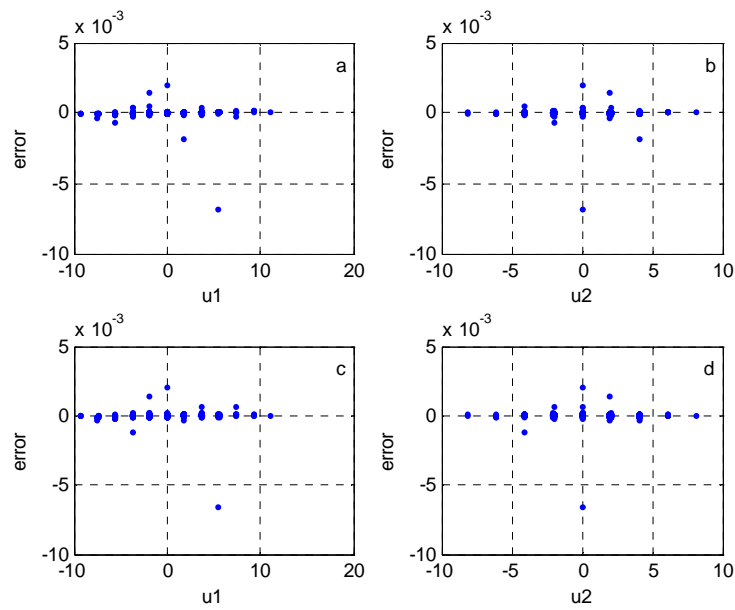


Fig. 11: model residual vs input measurement; (a,b) $n = m = 2$; (c,d) $n = m = 5$

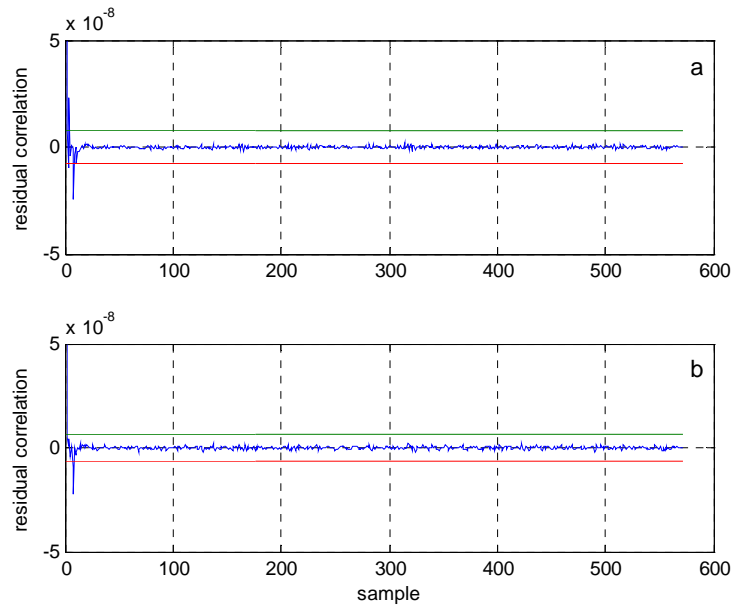


Fig. 12: model residual auto-correlation; (a) $n = m = 2$; (b) $n = m = 5$

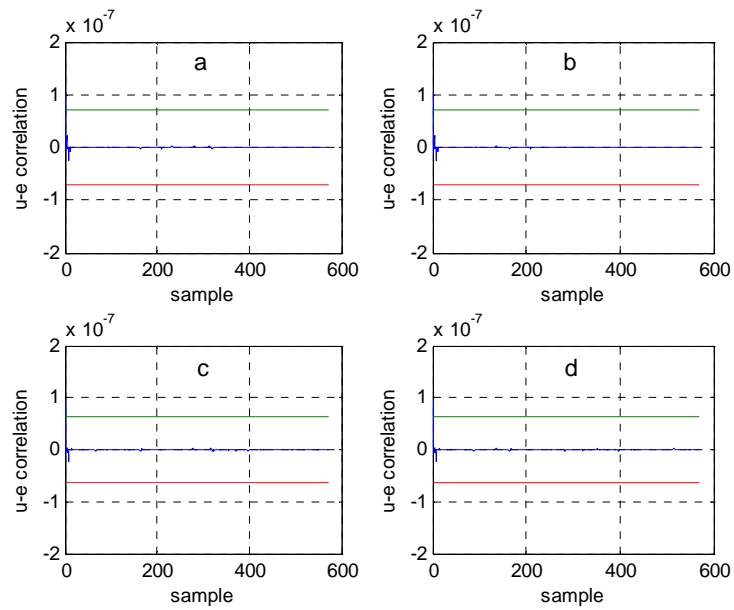


Fig. 13: model residual process input cross-correlation; (a,b) $n = m = 2$; (c,d) $n = m = 5$

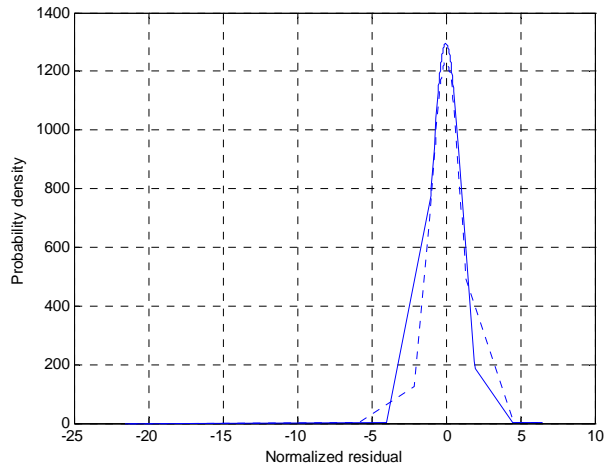


Fig. 14: Probability density of the model residuals; dashed line: $n = m = 2$, solid line: $n = m = 5$

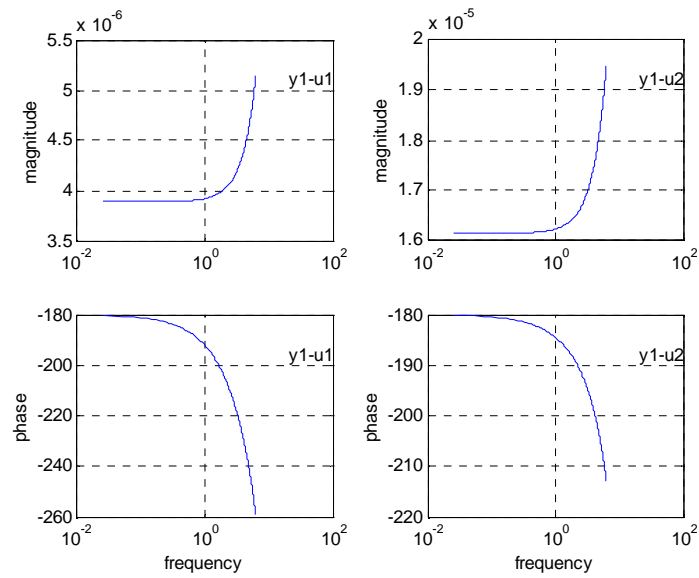


Fig. 15: Impulse response of the model: $n = m = 2$

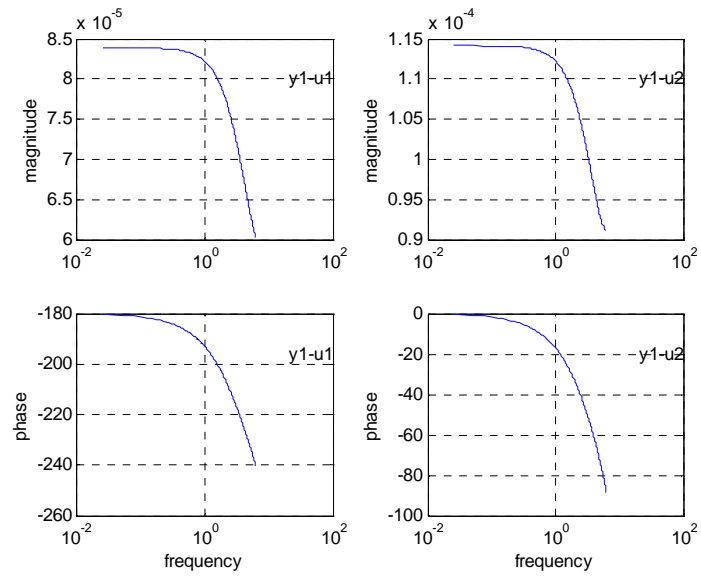


Fig. 16: Impulse response of the model: $n = m = 5$

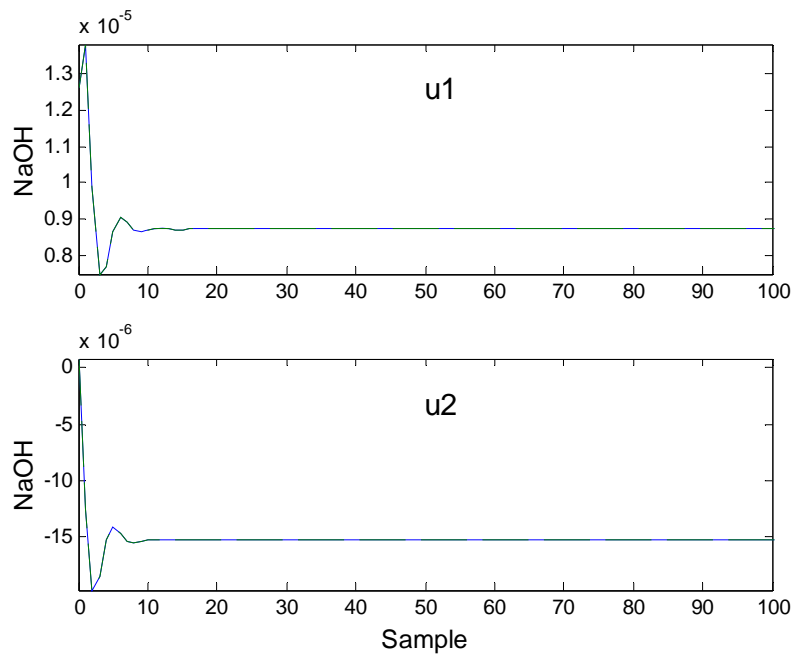


Fig. 17: Step response of the model: $n = m = 2$

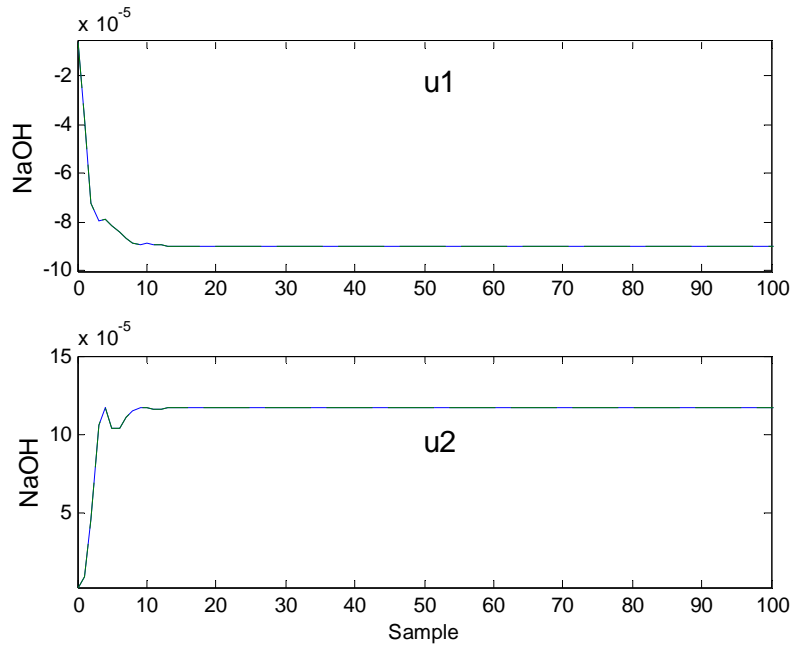


Fig. 18: Step response of the model: $n = m = 5$

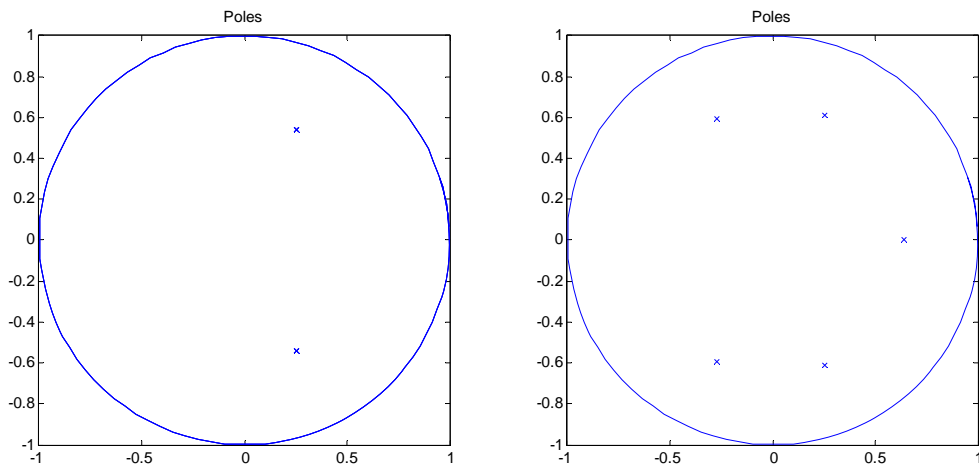


Fig. 19: Poles of the model

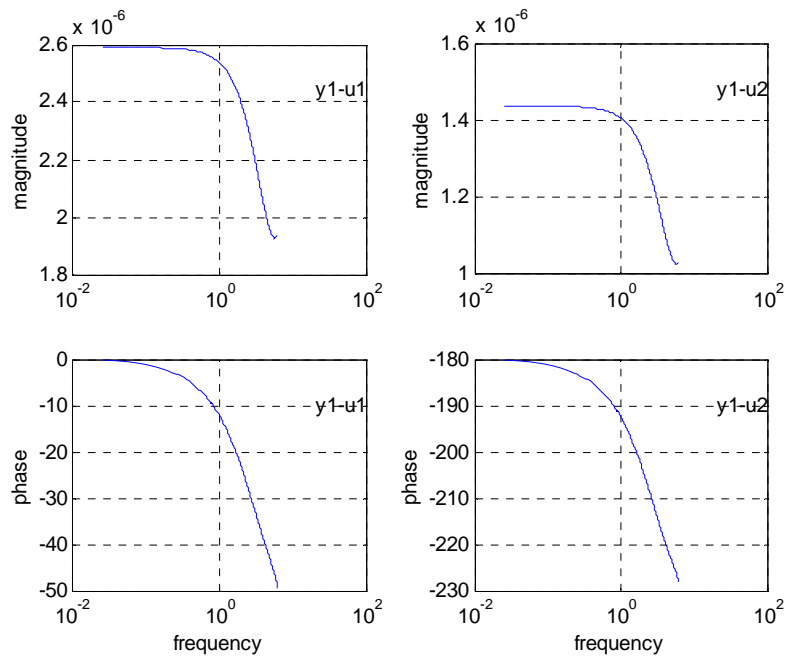


Fig. 20: Impulse response for ARX model with $n = 10$, $m = 2$

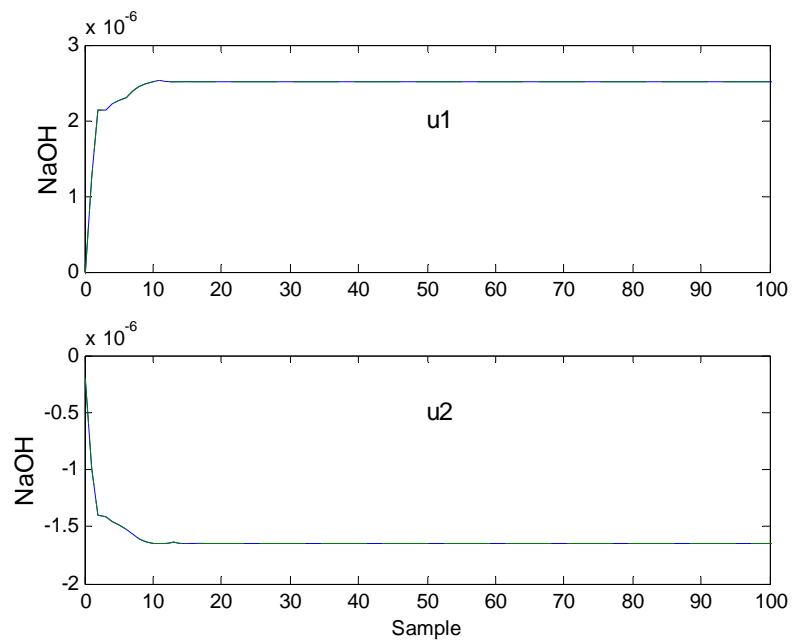


Fig. 21: Step response of ARX model with $n = 10$, $m = 2$

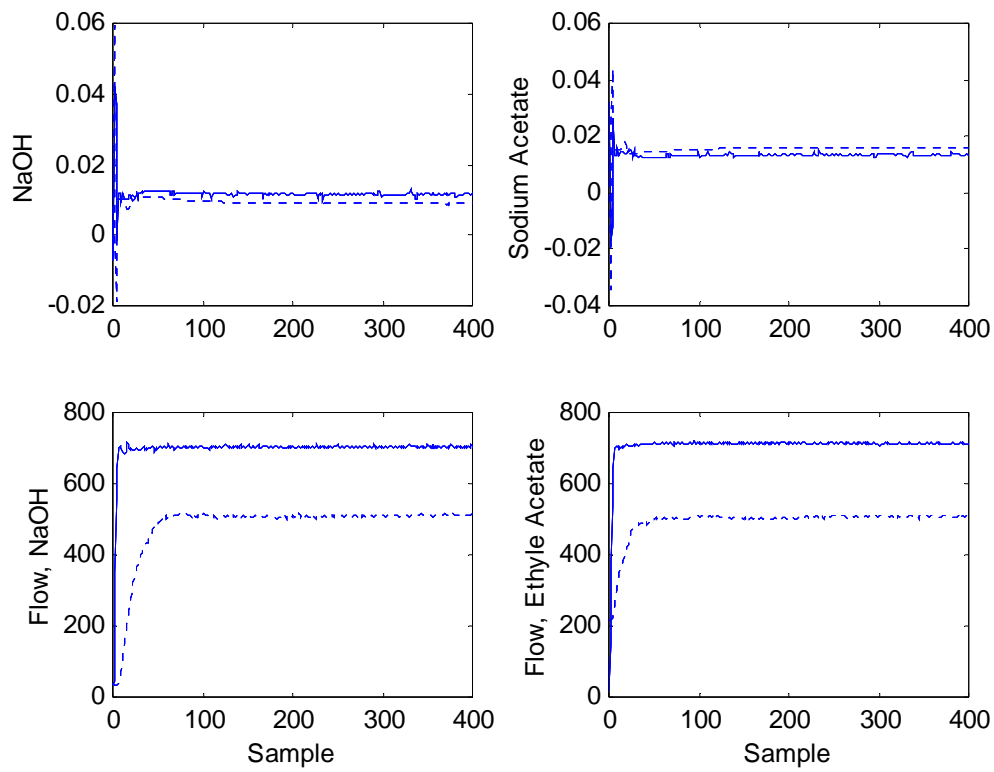


Fig. 22: Experimental Data at Feed flow rate of 500 ccm (dashed); 700 ccm (solid)

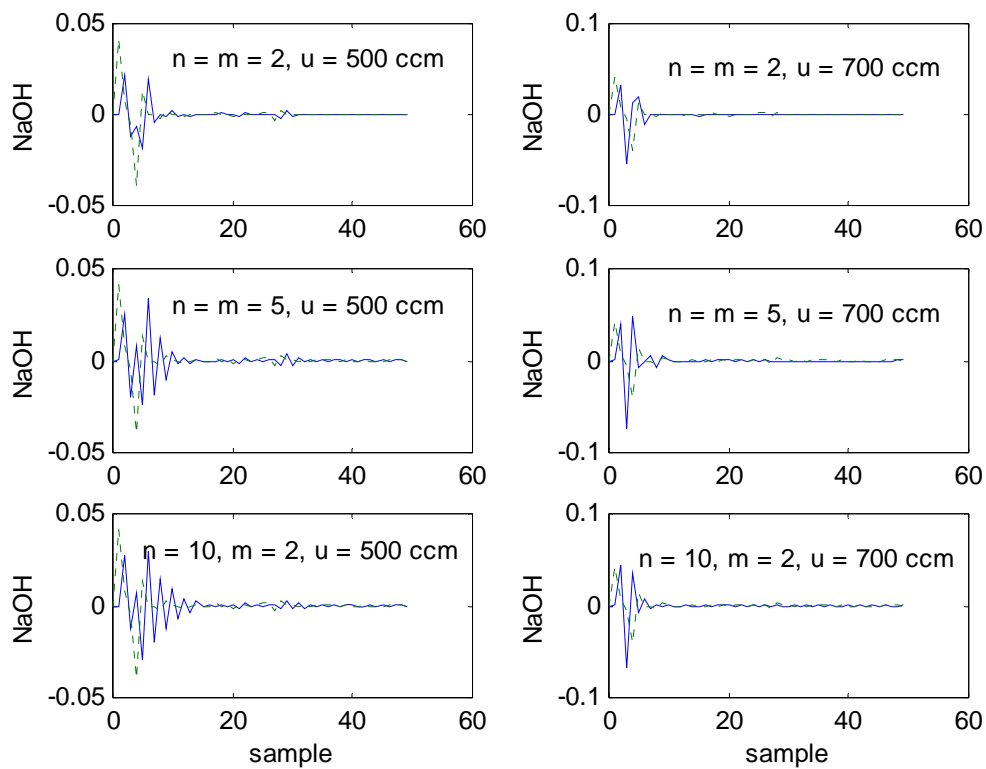


Fig. 23: ARX Model validation with different set of data

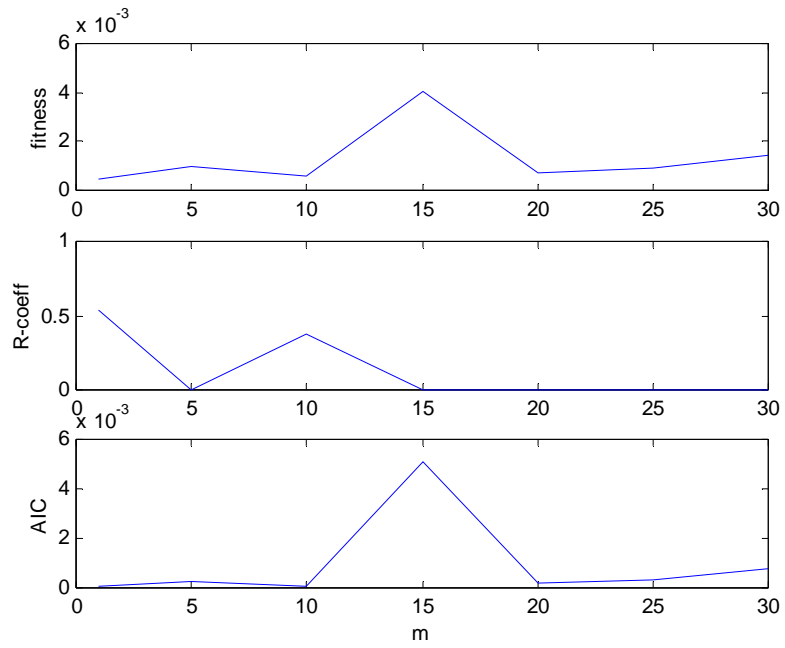


Fig. 24: ARMAX model accuracy measures versus the model order

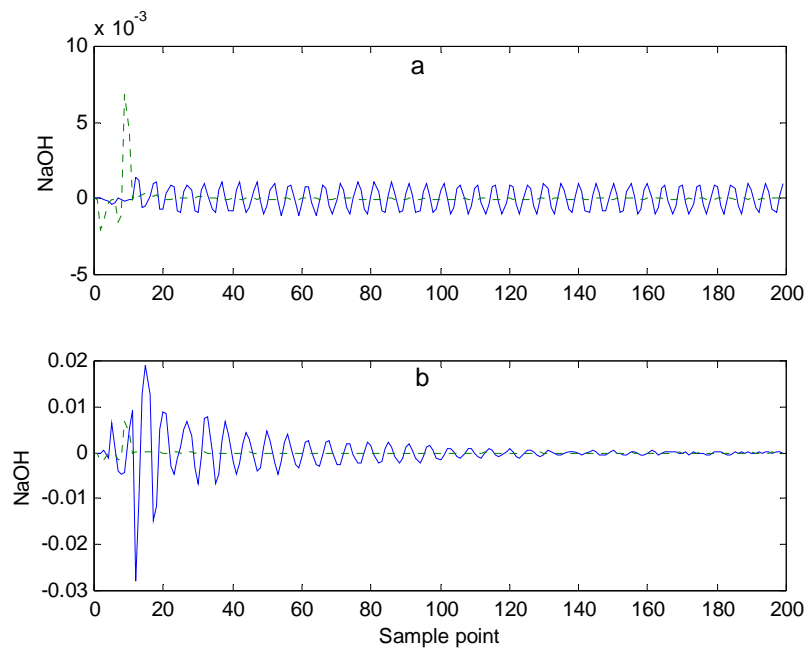


Fig. 25: ARMAX model prediction; (a) $n = m = l = 5$, (b) $n = m = l = 15$.

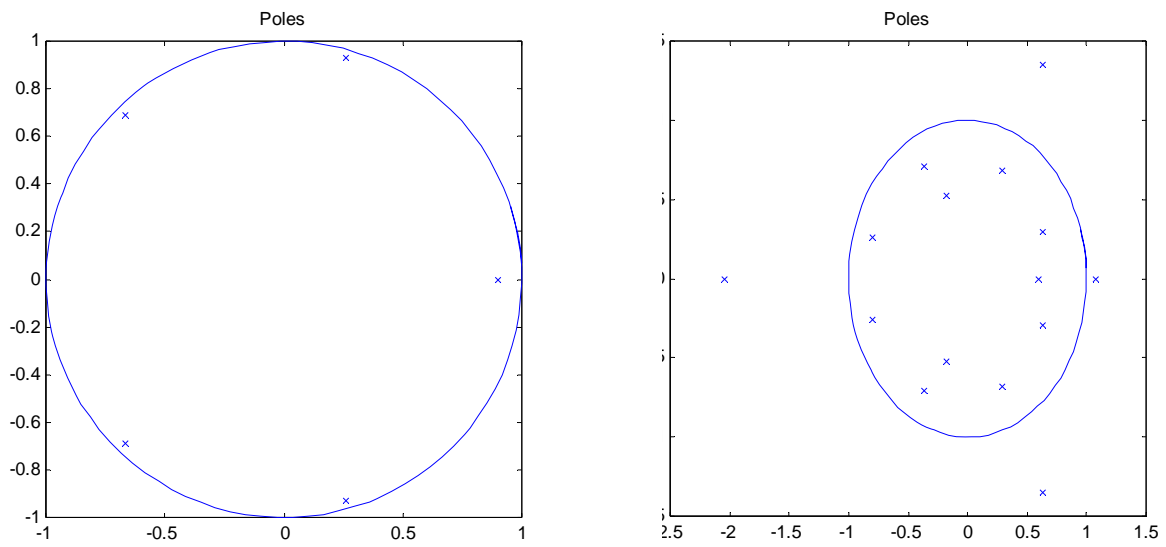


Fig. 26: poles of the ARMAX models

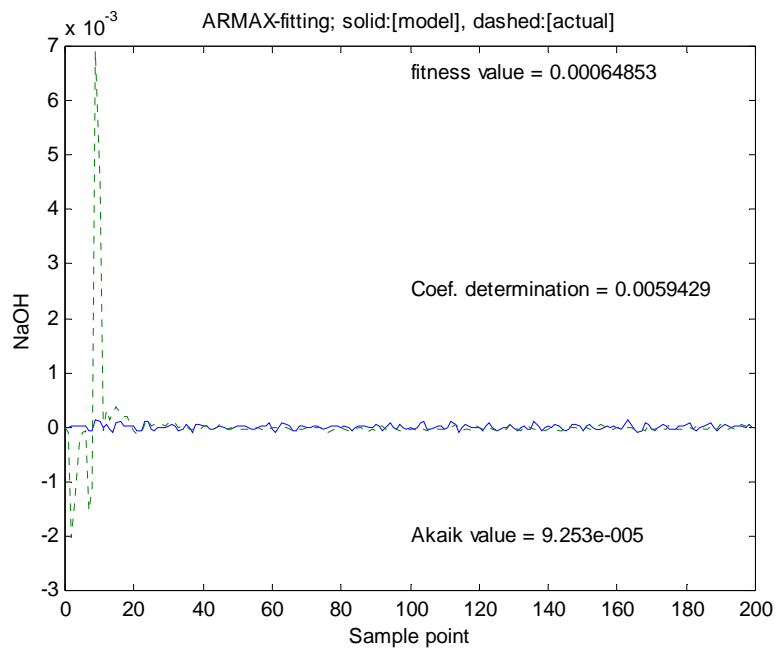


Fig. 27: OE model using $n = m = 5$.

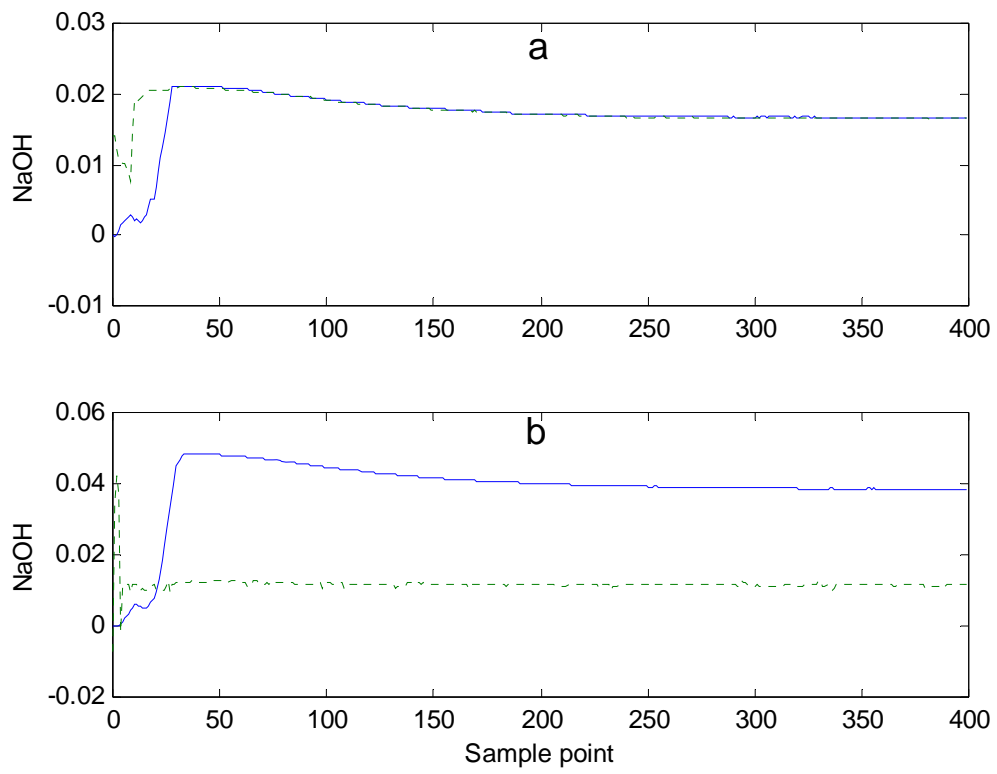


Fig. 28: validating OE model using original data, (a) flow = 300 ccm; (b) flow = 700 ccm.