# ModelID, an Interactive Program for Identification of MPC Relevant State-Space Models.

**Jørgen K. H. Knudsen** *

* *2-control Aps, Frimodtsvej 11, DK-2900 Hellerup, Denmark*
*(e-mail: JoeK@2-control.dk)*

**Abstract:** This paper describes a practical work flow during identification of linear time invariant state space models for MPC controllers using the new ModelID system identification program. ModelID is designed for process engineering practitioners, who want to develop models without having a detailed knowledge of system identification theory or computer programming. After preprocessing of process data, the user is guided through a set of MISO identifications. After inspection of the impulse responses from the MISO models, the MISO models are combined into a MIMO state space model, using SVD decomposition of Hankel matrices. Finally the state space model is tested in a MPC control loop.
Estimation of time lags in the process are demonstrated using simple (known) processes with different levels of process and measuring noise. The same processes are used to illustrate the problems encountered, when the processes cannot be described as ARX processes. The use of instrumental variable methods and linear filtering provides a solution to these problems.
The identification cycle is supported by many graphical outputs, providing valuable information about the process and the evaluated model.
ModelID is developed for the windows platform, using the C#/.NET based library MPCMath.

*Keywords:* system identification,instrumental variable methods, state-space models, MPC

## INTRODUCTION

The most time consuming task in implementation of MPC control is the development of the model, required for the controller. In many cases linear time invariant models are used for the controller, and typically the models are derived from plant data. Plant data can be historical data from data logging systems, or data generated from carefully planned experiments on the plant. ModelID is an identification program aimed at assisting the development of models. ModelID is designed for people who does not have a deep theoretical background in system identification or computer programming.

ModelD is a GUI based tool, developed for the windows platform, using the C#/.NET based library MPCMath (Knudsen, 2010b).

The purpose of this paper is to illustrate the practical work flow during system identification using ModelID. The paper demonstrates how time delays in the system can be estimated from whitening filtered cross correlations or from impulse responses calculated during the identification cycle. Finally the paper demonstrates the use of instrumental variable algorithms, in cases where the system cannot be adequately described by ARX models

ModelID consist of a set of tools running in a graphical portal. These tools are invoked sequentially during the identification work flow.

The steps and tools in the work flow are:

| Tool | Tasks |
|---|---|
| Data tool | Read data from file and perform initial data treatment. |
| Model tool | Set model dimensions and process delays.<br>Perform MISO identifications. |
| Impulse tool | Set impulse response length and calculate impulse responses.<br>Evaluate model quality from impulse responses. |
| Reduction tool | Perform SVD reduction. Select dimension of state space model and calculate MIMO state space models in innovation form. |
| MPC tool | Initial tuning and test of MPC controller. |

## DATA TOOL

Process data, read from a comma separated file, is displayed for removal of outliers and removal of fall-outs.

The four tank process is used to illustrate the work-flow. The four tank system, illustrated in Fig. 1, was introduced by Johanson (2000) as a benchmark for control design. Fig. 2 illustrates simulated raw data for this process.

The Controlled variables are the water levels in the four tanks, H1, H2, H3 and H4. The manipulated variables are the two inflows, F1 and F2.

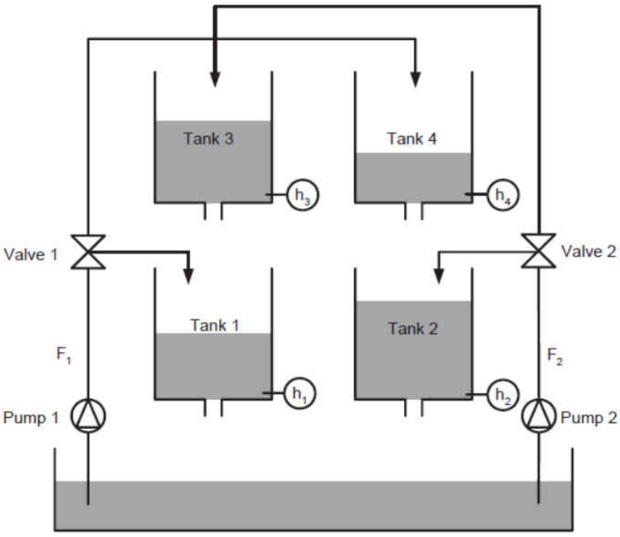


Fig. 1. Four tank process used to illustrate process identification work flow

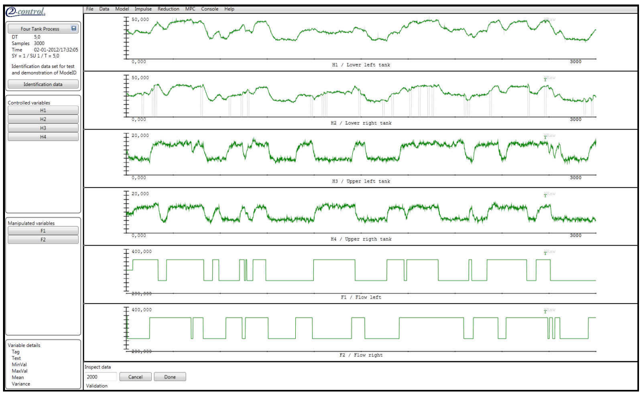

Fig. 2. Simulated raw process data for the four tank process.

The left panel of ModelID's GUI have buttons for the controlled variables and the manipulated variables. These are used to display and edit detailed information for the variables. Fig. 3 illustrates the removal of spikes and fall outs by selection of a threshold parameter for spikes and the value for a fall out situation. Finan et al. (2010) describes the handling of these problems.

After removal of spikes and fall outs the data set is divided into a part used for identification and a part used for validation

## MODEL TOOL

### *The identification procedure*

The linear time invariant system is given by

$$Y(t) = G(q)U(t) + H(q)E(t) \tag{1}$$

where

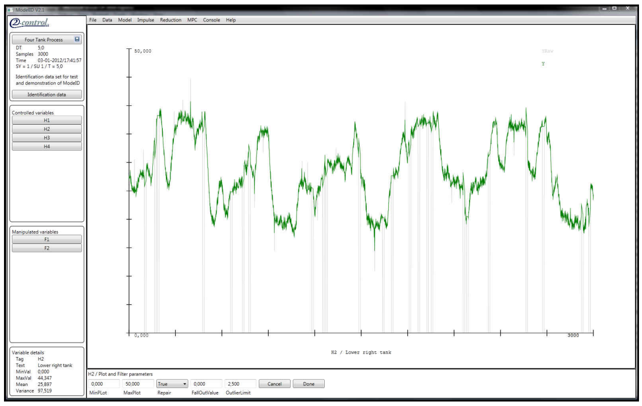

Fig. 3. Spikes and fall outs removal.

| | | |
|---|---|---|
| $Y(t)$ | $\in \mathbb{R}^{n_y}$ | Controlled variables. |
| $U(t)$ | $\in \mathbb{R}^{n_u}$ | Manipulated variables. |
| $E(t)$ | $\sim N_{iid}(0, I)$ | White noise, with $I$ being the identity matrix. |
| $G(q)$ | $\sim \mathbb{R}^{n_y \times n_u}$ | Deterministic transfer function between the controlled variables and the manipulated variables. |
| $H(q)$ | $\sim \mathbb{R}^{n_y \times n_y}$ | Transfer function of the disturbance model. |
| $q$ | | The time shift operator $qx(t) = x(t+1)$ and $q^{-1}x(t) = x(t-1)$. |

The controlled variables, $Y(t)$, are split into a deterministic part $Y_d(t)$ and a stochastic part $W(t)$, with $Y(t) = Y_d(t) + W(t)$. The transfer functions in (1) are assumed to have the structure

$$G(q) = \frac{B(q)}{A(q)} \tag{2}$$

$$H(q) = \frac{\Lambda}{D(q)} \tag{3}$$

where the polynomials are

$$A(q) = I - \sum_{j=1}^{sy} A_j q^{-j} \qquad A_j \in \mathbb{R}^{n_y \times n_y} \tag{4}$$

$$B(q) = \sum_{j=1}^{su} B_j q^{-j} \qquad B_j \in \mathbb{R}^{n_y \times n_u} \tag{5}$$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{n_y} \end{pmatrix} \tag{6}$$

$$D(q) = I - \sum_{j=1}^{sd} D_j q^{-j} \qquad D_j \in \mathbb{R}^{n_y \times n_y} \tag{7}$$

The deterministic one step predictor for this system is

$$\hat{Y}(t|t-1) = \sum_{j=1}^{sy} A_j Y(t-j) + \sum_{j=1}^{su} B_j U(t-j) \tag{8}$$

The deterministic predictor for the individual controlled variable is

$$\hat{y}_i(t|t-1) = \sum_{j=1}^{sy} a_{i,j} Y(t-j) + \sum_{j=1}^{su} b_{i,j} U(t-j) \qquad (9)$$

where $a_{i,j}$ and $b_{i,j}$ are the $i$ rows of $A_j$ and $B_j$ , $1 \leq i < n_y$.

The prediction errors for the individual controlled variables are defined by

$$\epsilon_i(t) = y_i(t) - \hat{y}_i(t) \qquad (10)$$

Having $n$ samples of $Y(t)$ and $U(t)$ , $0 \leq t < n$ , estimated $\hat{A}(q)$ and $\hat{B}(q)$ can be determined minimizing $n_y$ MISO problems

$$V_i = \sum_{t=1}^{n} \ell_i(F_i(q)\epsilon_i(t)) \qquad (11)$$

where $\ell_i$ are suitable norm functions. In ModelID the $\ell_2, \ell_1, \ell_\infty$ and $\ell_{Huber}$ norms are provided.

$F_i(q)$ are linear low pass filters reducing the effect of high frequency noise signals, defined by

$$F(q) = \frac{1-f}{1-fq^{-1}} \qquad 0 \leq f < 1 \qquad (12)$$

No filtering is obtained by setting $f = 0$. Selecting $f \to 1$ blocks all information.

Having identified the deterministic part $\hat{G}(q)$ as the combined result of $n_y$ MISO identifications, the stochastic noise signal can be estimated

$$W(t) = Y(t) - \hat{Y}(t) = Y(t) - \hat{G}(q)U(t) \qquad (13)$$

with the one step predictor:

$$\hat{W}(t|t-1) = \sum_{j=1}^{sd} D_j W(t-j) + \Lambda E(t) \qquad (14)$$

*Cross correlations and model dimensions*

The cross correlations between the manipulated variables and the controlled variables reveal important information about model structure and time delays in the process. Pure cross correlations, as shown in Fig. 4, show that there is no or limited interaction between the pairs F1-H3 and F2-H4.

Filtering the $u(t)$ and $y(t)$ with a filter $F_{wh}(q)$, which tries to make the manipulated variable $u(t)$ as white as possible, increases the information content (as well as the noise level).

$$\begin{aligned} u_F(t) &= F_{wh}(q)u(t) \\ y_F(t) &= F_{wh}(q)y(t) \end{aligned} \qquad (15)$$

The whitening filter $F_{wh}(q)$ is determined by modelling the manipulated variable as an AR-process with dimension 10.

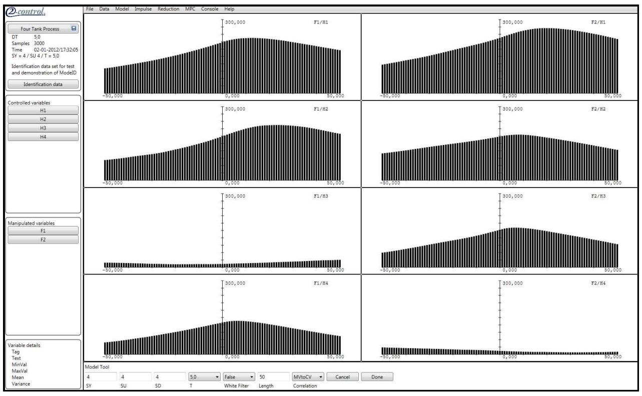

Fig. 4. Model tool: Four tank process correlations

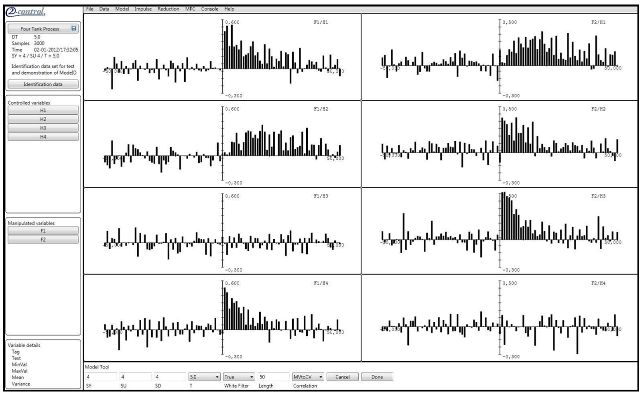

Fig. 5. Model tool: Correlations with white filter option. The correlations show first order dynamic between the pairs F1-H1, F1-H4. F2-H2, F2-H3 and higher order dynamics between F1-H2, F2-H1.

$$F_{wh}(q)u(t) = e(t) \qquad (16)$$

Fig. 5 shows the correlation plot using the white filter option. The correlations indicates that the relations between F1-H1, F1-H4, F2-H2 and F2-H3 could be first order dynamics, and the relations between F1-H2 and F2-H1 second order or higher dynamics. The correlation plot also indicates that there are no pure time delays involved in the plant dynamics.

At this stage the user has to enter estimated delays for the manipulated variables and the dimensions $sy$, $su$ and $sd$ in equations (8) and (14). This is an iterative process, where the user returns from the subsequent tools to this point, until a satisfactory result is obtained.

Having selected the model dimensions, the results of the individual MISO identifications are displayed by clicking the controlled variable buttons at the left side of the GUI panel. In Fig. 6, the upper left graph shows the plant measurements and the the output of the one step predictor (9). The lower left graph shows the prediction error $\epsilon_i$ (10). The upper right graph shows the distribution of $\epsilon_i$, the normal distribution curve corresponding to $\epsilon_i$ and the penalty function corresponding to the selected norm.

The lower right quadrant displays the obtained MISO model for the deterministic and stochastic part of the MISO model. At this point the $D_j$ in (14) are assumed

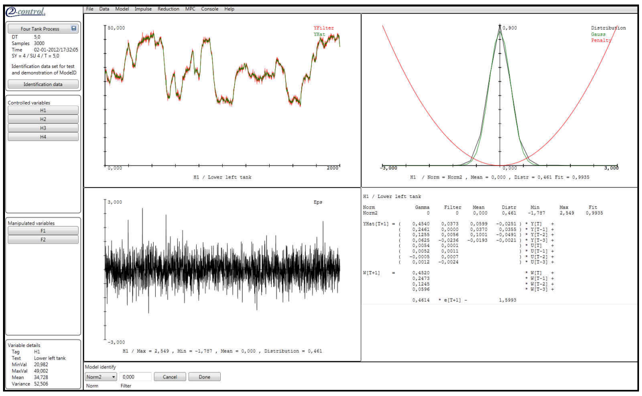

Fig. 6. Model tool: MISO identification

to be diagonal matrices, giving a "MISO" estimate of the stochastic noise.

At this point the user selects the desired norms and filters for the MISO identifications. Application of the different norms are described in Finan et al. (2010).

The $\ell_2$ norm is the standard norm for system identification tasks. The problem can be solved efficiently using either QR or Cholesky factorization algorithms. The disadvantage of the $\ell_2$ norm is the sensitivity to outliers. The $\ell_1$ norm is less sensitive to outliers and leads to the so called robust identification algorithms.

The Huber Norm, $\ell_{Huber}$, defined by (17), is a compromise between $\ell_2$ and $\ell_1$ norms

$$\ell_{Huber}(\epsilon_i) = \begin{cases} \frac{1}{2}\epsilon_i^2 & |\epsilon_i| \leq \gamma \\ \gamma|\epsilon_i| - \frac{1}{2}\gamma^2 & |\epsilon_i| > \gamma \end{cases} \tag{17}$$

$\ell_1$ and $\ell_\infty$ norms problems are solved as Linear Programming problems. The problems with a Huber-norm are solved as Quadratic Programming problems. Even though parameter estimation using these norms requires more computer resources than least-squares parameter estimation, such problem are solvable in acceptable time on standard laptop computers.

When satisfactory result has been obtained for all the $ny$ identifications, a "MIMO" model for the stochastic noise (3) is calculated before calling the impulse tool. The stochastic model is determined by regression of the predictor equation (14), assuming $D_i$ to be general matrices.

## IMPULSE TOOL

The impulse responses for the deterministic part of the identified model are calculated using (8) with

$$\begin{array}{lll} u_i(t) = 1 & t = 0 & 1 \leq i \leq n_u \\ u_i(t) = 0 & t > 0 & \end{array}$$

A sufficiently long length of the calculated impulse responses, $M$, must be entered, ensuring that they end close to zero.

If the model dimensions $sy$ and $su$ are chosen too high, the impulse responses will start with oscillating components.

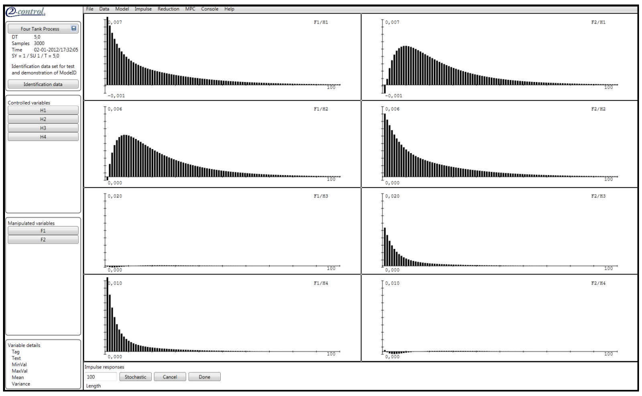

Fig. 7. Impulse responses for the deterministic model.

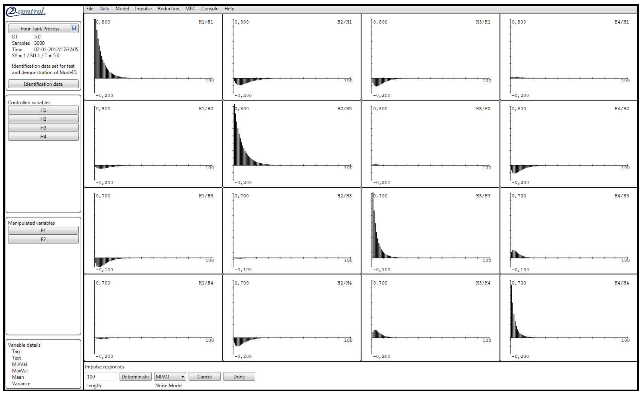

Fig. 8. The impulse responses for the MIMO disturbance model.

The impulse responses will also display pure delays between the controlled variables and the manipulated variables. Here the user can return to the Model tool and modify the model dimensions and delays for the manipulated variables.

The impulse responses for the stochastic part are calculated using (14) with

$$\begin{array}{lll} w_i(t) = 1 & t = 0 & 1 \leq i \leq n_y \\ w_i(t) = 0 & t > 0 & \\ E(t) = 0 & t \geq 0 & \end{array}$$

The user can select the desired structure for the stochastic part of the model ($D_{MIMO}, D_{MISO}$ or $D_{ARX}$). The $D_{ARX}$ option sets $D_i = A_i$ resulting in the noise model of an ARX process. This option is useful if the plant data hos no or negligible stochastic components.

The stochastic model is interesting for two reasons. Primarily it shows to what degree the processes can be described as ARX processes. If the process is not well represented by an ARX structure, the $\ell_2$, $\ell_1$ and $\ell_{Huber}$ norm produces biased estimates. The bias increases with the noise level. Secondly, the stochastic model is used to calculate the Kalman gain, K, of the final state space model (18). The Kalman gain is required for tracking between the physical plant and the internal model in the MPC controller. If neither the "MIMO", MISO" or the "ARX" stochastic models are applicable, the Kalman gain must be specified using other methods.

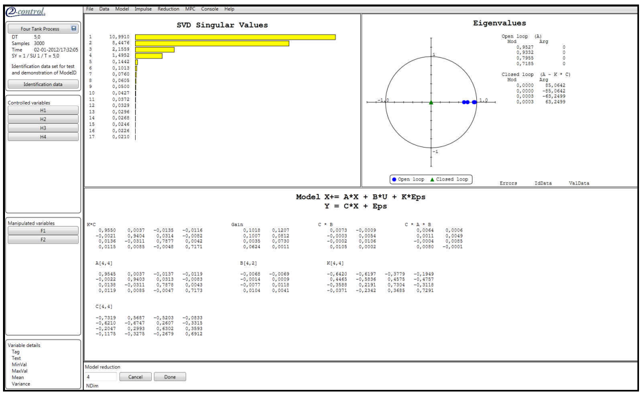


Fig. 9. The reduction tool creates state space models in innovation form.

## REDUCTION TOOL AND STATE-SPACE MODEL

In the reduction tool a block-Hankel matrix is constructed from the impulse responses. The Hankel matrix is factorized using a singular value decomposition (SVD)algorithm. As described in Appendix A, the state space model in innovation form can be realized by model reduction in balanced form using the SVD of the Hankel matrix. The state space model in innovation form is

$$X(t+1) = AX(t) + BU(t) + KE(t) \tag{18a}$$
$$Y(t) = CX(t) + E(t) \tag{18b}$$

The rank of the Hankel matrix is equal to the minimal rank for a state-space system representing the process. For a NDim system, the first NDim singular values are non zero and the subsequent singular values close to zero.

The result of the SVD reduction is shown in Fig. 9. The upper left graph shows the singular values from the SVD. The values show that the system can be represented by a state-space model of dimension NDim = 4. The user enters the desired dimension, NDim, and ModelID calculates the state space model as shown in the lower half of Fig. 9.

The upper right part of the diagram show the open and closed loop eigenvalues ($A - C * K$ in eq. (18))for the state space model.

If the length of the impulse responses is chosen too short, the singular values will decrease gradually, making the selection of the state-space model dimension, NDim, difficult.

## MPC TOOL

ModelID includes a MPC module, where tuning of normal MPC and soft constrained MPC (Prasath et al., 2010; Knudsen, 2010a) for the derived State-Space model can be tested simulating the MPC control loop.

## ESTIMATING TIME DELAYS

Proper estimation of time delays between the manipulated variables and the controlled variables is important in order to minimize the dimension of the identified state space model. Data from first order and second order

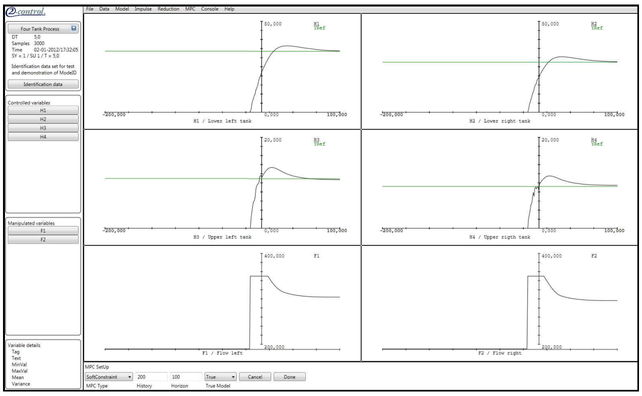

Fig. 10. MPC tool: Controlling the four tank process

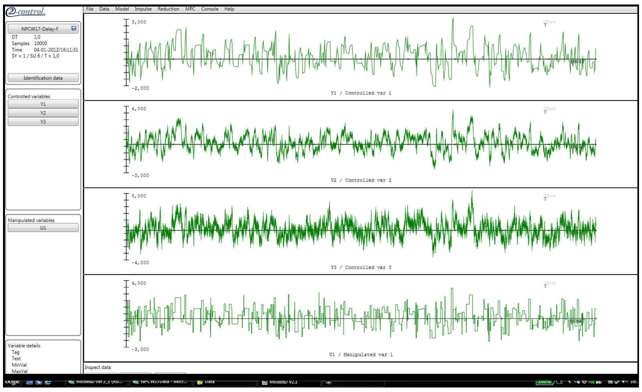

Fig. 11. First order ARX processes signal with increasing noise levels.

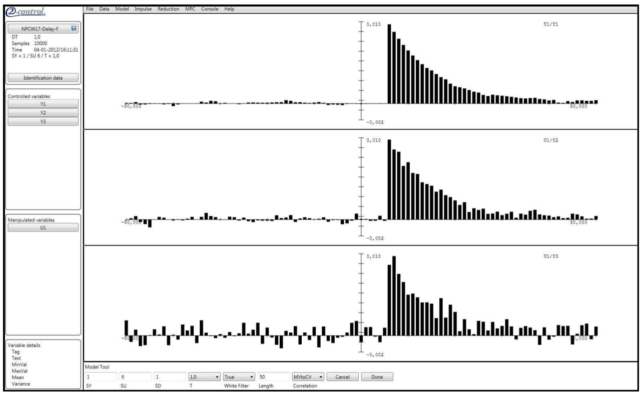

Fig. 12. First order ARX processes cross correlations.

processes with pure time delays will be used to illustrate the problem.

The first order system, with a delay of 5 second and a time constant on 10 seconds, is given by

$$y(t) = 0.9048y(t-1) + 0.0952u(t-6) + \sigma * e(t) \tag{19}$$

Fig. 11 shows three responses from (19) with noise variance $\sigma^2 = (0.0, 0.01, 0.1)$. With these noise levels, the time delay of 5 second is shown on all three response. Fig. 13 shows the impulse responses with MISO model dimension $sy = 1$ , $su = 6$. All the three responses clearly shows the delay. The user should return to the Model tool at set $su = 1$ and set a delay of 5 seconds for the manipulated variable.

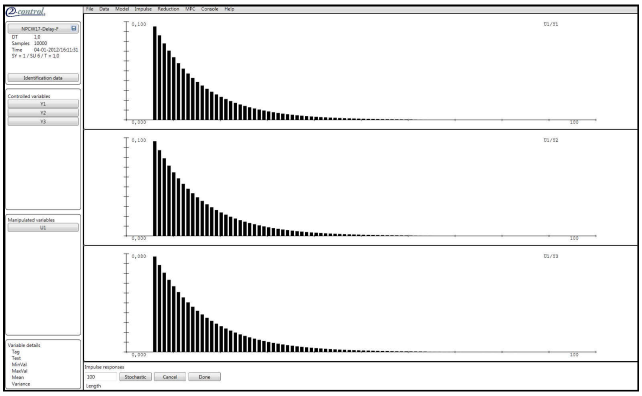

Fig. 13. First order ARX processes impulse responses.

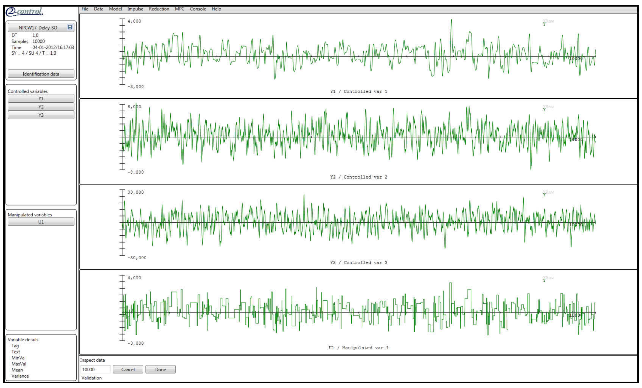

Fig. 14. Second order ARX processes signal with increasing noise levels.

The second order ARX system, with with a time delay of 5 second, a time constant of 10.0 second and damping of 0.5, is given by

$$y(t) = 1.8953y(t-1) - 0.9048y(t-2) \\ + 0.0047u(t-6) + 0.0047u(t-7) + \sigma e(t) \quad (20)$$

Fig. 14 shows three responses from (20) with noise variance $\sigma^2 = (0.0, 0.01, 0.1)$. With increasing noise level, the proper time delay cannot be determined from the correlation plots on Fig. 15. The impulse responses obtained setting $sy = 2$ and $su = 7$ is shown on Fig. 16. It is evident from Fig. 16 that the proper delay is obtained for all three noise levels. The user should return to the model tool at set $su = 2$ and set a delay of 5 seconds for the manipulated variable.

## INSTRUMENTAL VARIABLE METHODS

Regressions using the the $\ell_2, \ell_1, \ell_\infty$ and $\ell_{Huber}$ norms delivers unbiased estimates if we are dealing with ARX processes. If the process cannot be properly described as an ARX process, the estimate will be biased.

An example is data from a second order Output Error process with a time constant of 10.0 sec and a damping of 1.5. This process is described by

$$y(t) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 - a_1 q^{-1} - a_2 q^{-2}} u(t) + \sigma e(t) \quad (21)$$

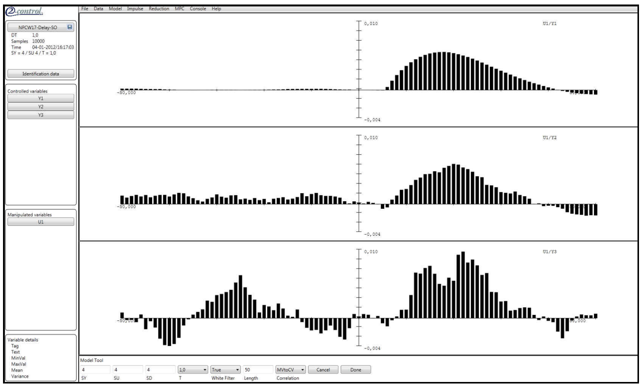

Fig. 15. Second order ARX processes cross correlations.

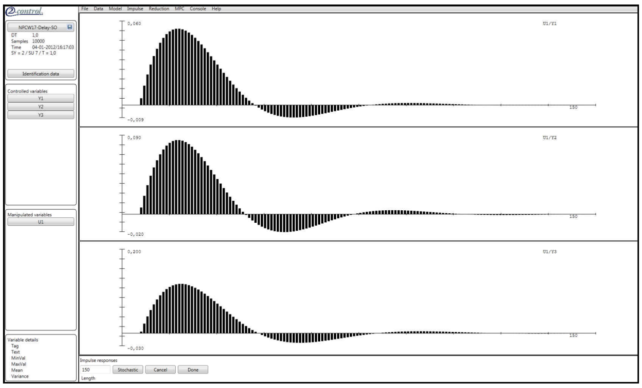

Fig. 16. Second order ARX processes impulse responses.

with $a_1 = 1.7322$ , $a_2 = -0.7408$, $b_1 = 0.0045$ and $b_2 = 0.0041$

Regression results with the $\ell_2$ norm, using ModelID are shown if Table 1. The parameters for $\sigma^2 = 0.0$ are the correct ones. The results obtained with $\sigma^2$= 0.01 and 0.1 are very biased.

Table 1. $\ell_2$ norm estimates

| $\sigma^2$ | $a_1$ | $a_2$ | $b_1$ | $b_2$ |
|---|---|---|---|---|
| 0.0 | 1.7322 | -0.7408 | 0.0045 | 0,0041 |
| 0.01 | 0.5515 | 0.3382 | 0.0028 | 0,0730 |
| 0.1 | 0.4376 | 0.4282 | 0.0262 | 0,0954 |

The Instrumental Variable methods (Ljung, 1999; Söderström, 2000) are a possible solutions to his problem. ModelID has implemented the IV4 algorithm (Ljung, 1999) , which can be selected during the MISO identifications with the Model tool. The results obtained with the IV4 algorithm are shown if Table 2.

Table 2. IV4 estimates with Filter = 0.5

| $\sigma^2$ | $a_1$ | $a_2$ | $b_1$ | $b_2$ |
|---|---|---|---|---|
| 0.0 | 1.7322 | -0.7408 | 0.0045 | 0,0041 |
| 0.01 | 1.8953 | -0.9048 | 0.0126 | 0.0031 |
| 0.1 | 1.9390 | -0.9447 | 0.0126 | 0,0069 |

The IV4 algorithm gives a much better estimate. In some cases the IV4 algorithm gives an unstable predictor, which luckily is very clearly seen. The predictor can be stabilised using the filter option. If stable, the prediction are rather insensitive to the selected value of the filter.

## CONCLUSION

The ModelId is a convenient tool to estimate linear time invariant model for MPC controllers. The aim is to present a tool which is relatively simple to use for the practical user.

In he future it would be natural to include sub-space methods, and support for Identification of closed loop data. Further investigations of Instrumental Variable algorithms is another interesting field.

## REFERENCES


Finan, D.A., Jørgensen, J.B., Poulsen, N.K., and Madsen, H. (2010). Robust model identification applied to type 1 diabetes. In *American Control Conference, Marriott Waterfront, Baltimore, MD, USA*.

Johanson, K.H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on control systems technology*, 8(3), 456–465.

Knudsen, J.K.H. (2010a). Implementing model predictive control in the csharp/.net environment. In *Model Based Control Conference. DTU*.

Knudsen, J.K.H. (2010b). Introduction to mpcmath. "`http://www.2-control.dk`".

Ljung, L. (1999). *System Identification, Theory for The User*. Prentice Hall, second edition.

Maciejowski, J. (2002). *Predictive Control with constraint*. Prentice Hall.

Prasath, G., Recke, B., Chidambaram, M., and Jørgensen, J. (2010). Application of soft constrained mpc to a cement mill circuit. In *9th International Symposium on Dynamics and Control of Process Systems,DYCOPS 2010*.

Söderström, T. (2000). *Instrumental variable methods for system identification*. Springer.


## Appendix A. GENERATING THE STATE SPACE MODEL

The Markov parameters for the process can be formed from the impulse responses (Maciejowski, 2002).

$$H(t) = \begin{pmatrix} h_{11}(t) & h_{11}(t) & \dots & h_{1n_u}(t) & h^e_{11}(t & \dots & h^e_{1n_y}(t) \\ h_{21}(t) & h_{21}(t) & \dots & h_{2n_u}(t) & h^e_{21}(t) & \dots & h^e_{2n_y}(t) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_{n_y1}(t) & h_{n_y1}(t) & \dots & h_{n_yn_u}(t) & h^e_{n_y1}(t) & \dots & h^e_{n_yn_y}(t) \end{pmatrix} \tag{A.1}$$

where $h_{ij}(t)$ is the deterministic impulse response for Controlled variable $i$ to Manipulated variable $j$ at time $t$ and $h^e_{ij}(t)$ is the stochastic impulse response for Controlled variable $i$ to Manipulated variable $j$ at time $t$

From this the Hankel matrices $\mathcal{H}_{M,M}$ and $\bar{\mathcal{H}}_{M+1,M+1}$ can be formed:

$$\mathcal{H}_{M,M} = \begin{pmatrix} H(1) & H(2) & \dots & H(M) \\ H(2) & H(3) & \dots & H(M+1) \\ \vdots & \vdots & \ddots & \vdots \\ H(M) & H(M+1) & \dots & H(2M-1) \end{pmatrix} \tag{A.2}$$

$$\bar{\mathcal{H}}_{M+1,M+1} = \begin{pmatrix} H(2) & H(3) & \dots & H(M+1) \\ H(3) & H(4) & \dots & H(M+2) \\ \vdots & \vdots & \ddots & \vdots \\ H(M+1) & H(M+2) & \dots & H(2M) \end{pmatrix} \tag{A.3}$$

Singular Value Decomposition, SVD, gives

$$\mathcal{H}_{M,M} = [K_1 K_2]] \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} [L_1 L_2]' \approx K_1 * \Lambda_1 L_1' \tag{A.4}$$

From this the matrices for the State-Space model in innovation form can be calculated

$$X(t+1) = AX(t) + BU(t) + KE(t) \tag{A.5a}$$
$$Y(t) = CX(t) + E(t) \tag{A.5b}$$

where

$$A = \Lambda_1^{-1/2} * K_1' \bar{\mathcal{H}}_{M+1,M+1} L_1 \Lambda_1^{-1/2} \tag{A.6a}$$
$$\tilde{B} = \Lambda_1^{1/2} [(L_1)_{1:n_u+n_y,:}]' \tag{A.6b}$$
$$B = \tilde{B}_{1:n_u,:} \tag{A.6c}$$
$$K = \tilde{B}_{n_u:n_u+n_y,:} \tag{A.6d}$$