# EMPIRICAL MODELING: APPROXIMATING THE DSR E SUB-SPACE SYSTEM IDENTIFICATION ALGORITHM BY A TWO-STEP ARX ALGORITHM

**Magnus Komperød[1], Tor Anders Hauge[2], David Di Ruscio[3], Bernt Lie[3]**

[1]Østfold University College, Faculty of Engineering
Halden, Norway

[2]Xstrata Nikkelverk
Kristiansand, Norway

[3]Telemark University College, Faculty of Technology,
Department of Electrical Engineering,
Information Technology and Cybernetics
Porsgrunn, Norway

*Bernt.Lie@hit.no (Bernt Lie)*

## Abstract

DSR E is a sub-space system identification algorithm for solving deterministic / stochastic linear time-invariant (LTI) system identification problems. DSR E consists of two steps: (i) The innovation process is identified. (ii) The deterministic / stochastic system identification problem is reduced to a deterministic problem by considering the identified innovation process as a deterministic input. In the DSR E algorithm presented in [1], this deterministic problem is solved using a deterministic sub-space system identification method. ARX is a system identification algorithm based on the ordinary least squares (OLS) method. In addition to the basic, single-step ARX algorithm, there are various multi-step ARX algorithms. In this paper it is shown that the first step of DSR E is mathematically identical to the single-step ARX algorithm. The second step, which is a deterministic problem, may also be solved by single-step ARX. Hence, each of the two steps of the DSR E algorithm may be replaced by single-step ARX, allowing DSR E to be approximated by a two-step ARX algorithm. DSR E and its two-step ARX approximation are compared by modeling a section of the copper refining process at Xstrata Nikkelverk, Kristiansand, Norway.

## Keywords

ARX; DSR E; Sub-space system identification; System identification.

## 1 Introduction

Modeling of dynamic systems is a most important part of today's science and engineering. Dynamic models serve many purposes, for example: (i) Training of process operators, pilots and astronauts. (ii) Exploring systems in a different time scale than physical time. (iii) Testing systems by simulations before they are manufactured, for example ships, airplanes, missiles, and sub-sea oil installations. (iv) Model-based control, such as LQG control, model-based predictive control (MPC), linear and nonlinear decouplers, Smith predictors, etc.

One of the most commonly used approaches for modeling dynamic systems is to develop models from equations of science. This is referred to as mechanistic modeling or first principle modeling. Another approach that may be based on laws of science and / or knowledge of the systems to be modeled is linguistic modeling (fuzzy modeling). Empirical modeling is another commonly used approach for developing dynamic as well as static models: Models are developed directly from observations of the systems. This is also referred to as black-box modeling. Empirical modeling used to build *dynamic* models is referred to as system identification. System identification is commonly used for developing models for model-based control. System identification may be used for systems that are too complex to be modeled by mechanistic modeling and where parameters in the mechanistic models are unknown.

A general introduction to system identification is given in [2]. Both linear and nonlinear system identification are considered. Also practical issues are discussed,

such as experiment design and data preprocessing. The DSR sub-space system identification algorithm is presented in [3]. In [1, 4] the DSR E sub-space system identification algorithm for use in closed loops is presented. DSR and DSR E have been developed by David Di Ruscio. Simulations comparing DSR E to other system identification algorithms, including N4SID, DSR, and the MATLAB implementation `pem` (Prediction Error Method), are presented in [1].

The main contribution of this paper is to show that the DSR E algorithm can be approximated by a two-step ARX algorithm. This approximation will in this paper be referred to as DARX.

The mathematical derivation of the DSR E algorithm, and inherently also the DARX algorithm, requires $N \to \infty$ and $J \to \infty$. Here $N$ is the number of samples and $J$ is a parameter to the DSR E and DARX algorithms. The $J$ parameter is the order of the model to be identified in the first step of the DARX algorithm. These requirements can not be met in any practical system identification problems. A study of how finite values of $N$ and $J$ influence the first step of DSR E and DARX is presented in this paper.

## 2 Notation and Definitions

The inputs to a system are collected in the input column vector, $u \in \mathbb{R}^{r\times 1}$, where $r$ is the number of inputs. The outputs from a system are collected in the output column vector, $y \in \mathbb{R}^{m\times 1}$, where $m$ is the number of outputs. A sub-script to these vectors, for example $y_k$, refers to the sampling number.

The limit notation of Eq. (1) is simplified as shown in Eq. (2).

$$\lim_{x\to z, y\to z} f(x,y) \tag{1}$$

$$\lim_{x,y\to z} f(x,y) \tag{2}$$

**Def. 1** (Innovation Process)**.** The system output, $y_k$, may be decomposed into two components: (i) The component of $y_k$ that can be predicted from previous inputs, $u_{-\infty}, \ldots, u_{k-1}$, and previous outputs, $y_{-\infty}, \ldots, y_{k-1}$, assuming no model errors. For bi-proper systems, i.e. systems having direct feed-through from the input, $u_k$, to the output, $y_k$, the current input, $u_k$, is also included in the prediction of $y_k$. This predictable component of $y_k$ is referred to as $\bar{y}_k$. (ii) The complement of $\bar{y}_k$, i.e. the component of $y_k$ that can *not* be predicted from previous inputs and previous outputs. This is referred to as the innovation process, $\varepsilon_k$. Hence, $\varepsilon_k = y_k - \bar{y}_k$.

The symbol $\varepsilon$ is used for the true innovation process, which in general is unknown. The symbol $\epsilon$ is used for the identified innovation process. The identified innovation process is in general not exactly identical to the true innovation process.

**Def. 2** (State Space Model Form)**.** The discrete state space model form used by the DSR E algorithm is as shown in Eq. (3) and Eq. (4) [5].

$$\begin{aligned} \bar{x}_{k+1} &= \tilde{A}\bar{x}_k + \tilde{B}u_k + \tilde{C}e_k && (3)\\ y_k &= \tilde{D}\bar{x}_k + \tilde{E}u_k + \tilde{F}e_k && (4) \end{aligned}$$

In Eq. (3) and Eq. (4) the tilde symbol is used to avoid confusion with the polynomials of ARMAX and ARX models. $\bar{x} \in \mathbb{R}^{n\times 1}$ is the estimate of the system state vector, $x$, where $n$ is the number of system states. $e \in \mathbb{R}^{m\times 1}$ is white noise with covariance matrix $E(e_k e_k^T) = I_m$. For invertible $\tilde{F}$, the system can be written on innovation form as presented in Eq. (5) and Eq. (6) [5].

$$\begin{aligned} \bar{x}_{k+1} &= \tilde{A}\bar{x}_k + \tilde{B}u_k + \tilde{K}\varepsilon_k && (5)\\ y_k &= \tilde{D}\bar{x}_k + \tilde{E}u_k + \varepsilon_k && (6) \end{aligned}$$

In Eq. (6), $\varepsilon_k = \tilde{F}e_k$ is the innovation process and $\tilde{K} = \tilde{C}\tilde{F}^{-1}$ is the Kalman filter gain matrix. In this paper, only strictly proper systems will be considered, i.e. $\tilde{E} = 0_{m\times r}$.

**Def. 3** (ARMAX Model Form)**.** Eq. (7) defines the general form of ARMAX models [2].

$$A(q)y_k = B(q)u_k + C(q)\varepsilon_k \tag{7}$$

In Eq. (7), $q$ is the time-shift operator of the Z-transform, i.e. $q^{-1}\, y_k = y_{k-1}$. Symbol $q$ is commonly used within the subject of system identification. Symbol $z$ is used in many other contexts. $A(q)$, $B(q)$, and $C(q)$ are polynomials. $n_A$, $n_B$, and $n_C$ are the number of coefficients in these polynomials that in general are different from 1. The $A(q)$ and $C(q)$ polynomials are monic polynomials, i.e. the coefficient of their highest order term is 1.

**Def. 4** (ARX Model Form)**.** Eq. (8) defines the general form of ARX models [2].

$$A(q)y_k = B(q)u_k + \varepsilon_k \tag{8}$$

The $A(q)$ polynomial is monic.

**Def. 5** (Orthogonal Projection)**.** The orthogonal projection of matrix $G$ onto matrix $H$, $G/H$, is defined as in Eq. (9) [5].

$$G/H \stackrel{\text{def}}{=} GH^T(HH^T)^\dagger H \tag{9}$$

In Eq. (9), the super-script $\dagger$ refers to the Moore-Penrose pseudo-inverse.

**Def. 6** (Complement of Orthogonal Projection)**.** The complement of the orthogonal projection of matrix $G$ onto matrix $H$, $GH^\perp$, is defined as in Eq. (10) [5].

$$GH^\perp \stackrel{\text{def}}{=} G - G/H \stackrel{\text{def}}{=} G - GH^T(HH^T)^\dagger H \tag{10}$$

**Def. 7** (Hankel Matrix)**.** Let $s_t \in \mathbb{R}^{n_r \times n_c}$ be a matrix of data sampled at timestep $t$. The Hankel matrix $S_{t_0|L}$, organizing timeseries of $s_t$ starting at timestep $t_0$, i.e. $s_{t_0}, s_{t_0+1}, \ldots$, is defined as in Eq. (11).

$$S_{t_0|L} \stackrel{\text{def}}{=} \begin{bmatrix} s_{t_0} & s_{t_0+1} & \cdots & s_{t_0+K-1} \\ s_{t_0+1} & s_{t_0+2} & \cdots & s_{t_0+K} \\ \vdots & \vdots & \ddots & \vdots \\ s_{t_0+L-1} & s_{t_0+L} & \cdots & s_{t_0+L+K-2} \end{bmatrix} \in \mathbb{R}^{Ln_r \times Kn_c} \tag{11}$$

In Eq. (11), $L$ is the number of block rows in $S_{t_0|L}$ and $K$ is the number of block columns in $S_{t_0|L}$ [5].

**Def. 8** (Lower Block Triangular Toeplitz Matrix for the Quadruple $(\tilde{D}, \tilde{A}, \tilde{C}, \tilde{F})$)**.** The lower block triangular Toeplitz matrix for the quadruple $(\tilde{D}, \tilde{A}, \tilde{C}, \tilde{F})$ is defined as in Eq. (12) [5].

$$H_L^s \stackrel{\text{def}}{=} \begin{bmatrix} \tilde{F} & 0_{m\times m} & 0_{m\times m} & \cdots & 0_{m\times m} \\ \tilde{D}\tilde{C} & \tilde{F} & 0_{m\times m} & \cdots & 0_{m\times m} \\ \tilde{D}\tilde{A}\tilde{C} & \tilde{D}\tilde{C} & \tilde{F} & \cdots & 0_{m\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{D}\tilde{A}^{L-2}\tilde{C} & \tilde{D}\tilde{A}^{L-3}\tilde{C} & \tilde{D}\tilde{A}^{L-4}\tilde{C} & \cdots & \tilde{F} \end{bmatrix} \in \mathbb{R}^{Lm\times Lm} \tag{12}$$

In Eq. (12), $L$ is the number of block rows and block columns in $H_L^s$.

## 3 The DSR E Algorithm

This section gives a brief derivation of the DSR E subspace system identification algorithm. A comprehensive presentation of DSR E is provided by [1]. DSR E is also presented in [4].

In [5] it is proved that for linear time-invariant (LTI) systems the innovation process, $\varepsilon_k$, can be identified directly from previous inputs, $u_{-\infty}, \ldots, u_{k-1}$, and previous outputs, $y_{-\infty}, \ldots, y_{k-1}$, without relying on models. The DSR E algorithm presented in [1, 4] is based on this proof. A MATLAB implementation of the DSR E algorithm is available in the DSR Toolbox for MATLAB ([6]). The DSR E algorithm consists of two steps:

1. The innovation process, $\varepsilon_k$, is identified by orthogonal projection of the current output, $y_k$, onto inputs and outputs from the $J$ preceding samples, i.e. $u_{k-J}, \ldots, u_{k-1}$ and $y_{k-J}, \ldots, y_{k-1}$. Here $J$ is a parameter to the DSR E algorithm. The complement of this orthogonal projection is the identified innovation process, $\epsilon_k$ [1, 4]. Please refer to Subsec. 3.1 for details.
2. The identified innovation process, $\epsilon_k$, is considered as a known deterministic input. Hence, the deterministic / stochastic system identification problem is reduced to a deterministic system identification problem [1, 4]. In the DSR E algorithm presented in [1], this deterministic problem is solved by a deterministic sub-space system identification algorithm.

The the following parameters are most important with respect to the DSR E algorithm and its derivation [5, 1, 4]:

1. $L$ - the number of block rows in the Toeplitz matrices and some of the Hankel matrices to be used in the second step of the DSR E algorithm, i.e. $L$ has the same meaning as in Eq. (11) and Eq. (12).
2. $g$ - if the system is strictly proper, i.e. $E = 0_{m\times r}$, then $g$ is set to 0. Otherwise $g$ is set to 1.
3. $J$ - the number of preceding inputs and outputs used to identify the innovation process. Please refer to Subsec. 3.1 for details.

In addition to parameters $L$, $g$, and $J$, the model order, $n$, of the model to be identified by DSR E has to be specified to the DSR E implementation of the DSR Toolbox for MATLAB ([6]).

### 3.1 Step 1 of the DSR E Algorithm

Let $U_{0|J}$, $U_{J|L+g-1}$, $Y_{0|J}$, $Y_{J|L}$, and $E_{J|L}$ be Hankel matrices according to Def. 7. The input vector $u \in \mathbb{R}^{r\times 1}$ is the block elements of the matrices $U_{0|J}$ and $U_{J|L+g-1}$. The output vector $y \in \mathbb{R}^{m\times 1}$ is the block elements of the matrices $Y_{0|J}$ and $Y_{J|L}$. The white noise vector $e \in \mathbb{R}^{m\times 1}$ is the block elements of matrix $E_{J|L}$. In [5], Eq. (13) is proved. This proof will not be repeated in here.

$$\lim_{J,K\to\infty} Y_{J|L} - Y_{J|L} / \begin{bmatrix} U_{J|L+g-1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} = \lim_{J,K\to\infty} H_L^s E_{J|L} \tag{13}$$

Please note that the mathematical derivation of Eq. (13) requires that $J \to \infty$ and $K \to \infty$, where $K$ is the number of columns in the Hankel matrices of Eq. (13). In Sec. 7 the consequences of finite values of $J$ and

$K$ are considered. Choosing $L = 1$ and $g = 0$ gives Eq. (14) [1, 4].

$$\begin{bmatrix} U_{J|L+g-1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} = \begin{bmatrix} U_{J|0} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} = \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} \quad (14)$$

The equality between the middle and the rightmost terms of Eq. (14) is due to $U_{J|0} \in \mathbb{R}^{0\times K}$. For $L = 1$ the Toeplitz matrix of Def. 8 reduces to $H_1^s \in \mathbb{R}^{m\times m} \Rightarrow H_1^s = \tilde{F}$. Hence, Eq. (13) can be written as Eq. (15) [1, 4].

$$\begin{aligned} \lim_{J,K\to\infty} Y_{J|1} - Y_{J|1}/\begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} &= \lim_{J,K\to\infty} \tilde{F}E_{J|1} \\ &= \lim_{J,K\to\infty} \tilde{F}\begin{bmatrix} e_J & e_{J+1} & \dots & e_{J+K-1} \end{bmatrix} \\ &= \lim_{J,K\to\infty} \begin{bmatrix} \varepsilon_J & \varepsilon_{J+1} & \dots & \varepsilon_{J+K-1} \end{bmatrix} \\ &= \lim_{J,K\to\infty} \varepsilon_{J|1} \end{aligned} \quad (15)$$

In Eq. (15), $\varepsilon_{J|1}$ is the Hankel matrix of which block elements are the innovation process $\varepsilon \in \mathbb{R}^{m\times 1}$.

### 3.2 Step 2 of the DSR E Algorithm

As the innovation process is identified in Eq. (15), the deterministic / stochastic system identification problem of Eq. (5) and Eq. (6) reduces to a deterministic problem. For strictly proper systems, i.e. $\tilde{E} = 0_{m\times r}$, the system identification problem is now on the form of Eq. (16) and Eq. (17) [1, 4].

$$x_{k+1} = \tilde{A}x_k + \begin{bmatrix} \tilde{B} & \tilde{K} \end{bmatrix}\begin{bmatrix} u_k \\ \epsilon_k \end{bmatrix} \quad (16)$$

$$y_k - \epsilon_k = \tilde{D}x_k \quad (17)$$

Step 2 of the DSR E algorithm as presented in [1] is to solve the deterministic system identification problem of Eq. (16) and Eq. (17) using a deterministic sub-space system identification algorithm.

## 4 Relating Orthogonal Projection to the Ordinary Least Squares Method

The key to understand that the first step of the DSR E algorithm is mathematically identical to the single-step ARX algorithm is the relation between orthogonal projection (Def. 5) and the ordinary least squares (OLS) method. Consider the linear regression problem of Eq. (18).

$$Y = XB + E \quad (18)$$

In Eq. (18), the elements of $X \in \mathbb{R}^{N\times b}$ and $Y \in \mathbb{R}^{N\times a}$ are known data. The regression matrix $B \in \mathbb{R}^{b\times a}$ is to be identified. $E \in \mathbb{R}^{N\times a}$ is the residual of the linear regression. Assume that $N \geq b$ and that $\mathrm{rank}(X) = b$. Then $\mathrm{rank}(X^TX) = b$, which is full rank. For a quadratic matrix of full rank, i.e. an invertible matrix, the Moore-Penrose pseudo inverse is equivalent to the inverse, i.e. $(X^TX)^\dagger = (X^TX)^{-1}$. Transposing Eq. (18) gives Eq. (19).

$$Y^T = B^TX^T + E^T \quad (19)$$

Solving the linear regression problem of Eq. (19) using OLS gives Eq. (20).

$$B^T = Y^TX(X^TX)^{-1} \quad (20)$$

Inserting Eq. (20) into Eq. (19) gives Eq. (21).

$$Y^T = \underbrace{Y^TX(X^TX)^{-1}}_{B}X^T + E^T \quad (21)$$

Because $(X^TX)^\dagger = (X^TX)^{-1}$, the first term on the right hand side of Eq. (21) is identical to the right hand side of Eq. (9), where $Y^T = G$ and $X^T = H$. Hence, according to Def. 5, Eq. (21) can be written as Eq. (22).

$$Y^T = Y^T/X^T + E^T \quad (22)$$

Further, using Def. 6 gives Eq. (23).

$$Y^TX^{T\perp} = Y^T - Y^T/X^T = E^T \quad (23)$$

Conclusions: (i) The orthogonal projection $Y^T/X^T$ is equivalent to the part of $Y$ that can be explained by an OLS regression of $Y$ onto $X$. (ii) The complement of the orthogonal projection, $Y^TX^{T\perp}$, is equivalent to the residual of this OLS regression, $E$.

## 5 The ARX Algorithm

The term ARX may refer to (i) the model form of Eq. (8) or (ii) a system identification algorithm used to identify models on the form of Eq. (8). This section gives a brief introduction to the ARX algorithm.

For single input, single output (SISO), strictly proper systems, Eq. (8) can be written on the form of Eq. (24). It has here been used that $q^{-1}y_k = y_{k-1}$, where $q$ is the time shift operator of the Z transform. Eq. (24) can be rewritten as Eq. (25).

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + \ldots + a_{n_A} y_{k-n_A} = b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_{n_B} u_{k-n_B} + \varepsilon_k \tag{24}$$

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \ldots - a_{n_A} y_{k-n_A} + b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_{n_B} u_{k-n_B} + \varepsilon_k \tag{25}$$

The polynomial coefficients, $a_1, a_2, \ldots, a_{n_A}$ and $b_1, b_2, \ldots, b_{n_B}$, can be estimated by stacking timeseries of $y$ and $u$ in the $Y$ and $X$ matrices of Eq. (18) as shown in Eq. (26), and then solve this OLS problem with respect to the ARX parameter vector, $\theta$, which corresponds to the regression matrix, $B$, of Eq. (18).

$$\underbrace{\begin{bmatrix} \vdots \\ y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -y_{k-1} & -y_{k-2} & \cdots & -y_{k-n_A} & u_{k-1} & u_{k-2} & \cdots & u_{k-n_B} \\ -y_k & -y_{k-1} & \cdots & -y_{k-n_A+1} & u_k & u_{k-1} & \cdots & u_{k-n_B+1} \\ -y_{k+1} & -y_k & \cdots & -y_{k-n_A+2} & u_{k+1} & u_k & \cdots & u_{k-n_B+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_A} \\ b_1 \\ b_2 \\ \vdots \\ b_{n_B} \end{bmatrix}}_{\theta} + \underbrace{\begin{bmatrix} \vdots \\ \varepsilon_k \\ \varepsilon_{k+1} \\ \varepsilon_{k+2} \\ \vdots \end{bmatrix}}_{E} \tag{26}$$

In Eq. (26), $Y \in \mathbb{R}^{P\times 1}$, $X \in \mathbb{R}^{P\times(n_A+n_B)}$, $\theta \in \mathbb{R}^{(n_A+n_B)\times 1}$, and $E \in \mathbb{R}^{P\times 1}$, where $P$ is the number of rows in $Y$, $X$, and $E$.

The ARX system identification problem of Eq. (26) can be generalized to multiple input, multiple output (MIMO) systems as shown in Eq. (27).

$$\underbrace{\begin{bmatrix} \vdots \\ y_k^T \\ y_{k+1}^T \\ y_{k+2}^T \\ \vdots \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -y_{k-1}^T & -y_{k-2}^T & \cdots & -y_{k-n_A}^T & u_{k-1}^T & u_{k-2}^T & \cdots & u_{k-n_B}^T \\ -y_k^T & -y_{k-1}^T & \cdots & -y_{k-n_A+1}^T & u_k^T & u_{k-1}^T & \cdots & u_{k-n_B+1}^T \\ -y_{k+1}^T & -y_k^T & \cdots & -y_{k-n_A+2}^T & u_{k+1}^T & u_k^T & \cdots & u_{k-n_B+2}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}}_{X} \Theta + \underbrace{\begin{bmatrix} \vdots \\ \varepsilon_k^T \\ \varepsilon_{k+1}^T \\ \varepsilon_{k+2}^T \\ \vdots \end{bmatrix}}_{E} \tag{27}$$

In Eq. (27), $Y \in \mathbb{R}^{P \times m}$, $X \in \mathbb{R}^{P \times (mn_A + rn_B)}$, $\Theta \in \mathbb{R}^{(mn_A+rn_B) \times m}$, and $E \in \mathbb{R}^{P \times m}$, where $P$ is the number of rows in $Y$, $X$, and $E$. The notation $\Theta$ is used for the ARX parameter matrix, which is to be identified by the OLS method.

Solving the linear regression problem of Eq. (26) or Eq. (27) with respect to $\theta$ or $\Theta$, respectively, is the system identification algorithm referred to as single-step ARX. There also exist various multi-step ARX algorithms. These algorithms involve solving several single-step ARX problems. A two-step ARX identification algorithm is derived in Sec. 6.

## 6 Approximating the DSR E Algorithm Using a Two-Step ARX Algorithm

This section shows that the DSR E sub-space system identification algorithm can be approximated by a two-step ARX algorithm.

### 6.1 Approximating Step 1 of the DSR E Algorithm

Consider Eq. (15), using the final right hand side term: As the requirements $J \to \infty$ and $K \to \infty$ can not be met in any practical system identification problems, the limit notation is removed and the true innovation process, $\varepsilon$, is replaced by the identified (in general not exact) innovation process, $\epsilon$. Further, Eq. (15) is rearranged by: (i) The last term on the left hand side is rewritten using Def. 5 and moved to the right hand side. (ii) The equation is transposed. Eq. (15) is then written as Eq. (28). The underbraces of Eq. (28) is on the form of Eq. (18), where the regression matrix, $B$, is replaced by the ARX parameter matrix, $\Theta$. The structures of the regression matrices of Eq. (28) are shown in Eq. (29) and Eq. (30).

$$\underbrace{Y_{J|1}^T}_{Y} = \underbrace{\begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}^T}_{X} \underbrace{\left( \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}^T \right)^{-1} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} Y_{J|1}^T}_{\Theta} + \underbrace{\epsilon_{J|1}^T}_{E} \tag{28}$$

$$Y_{J|1}^T = \begin{bmatrix} y_J^T \\ y_{J+1}^T \\ \vdots \\ y_{J+K-1}^T \end{bmatrix} \in \mathbb{R}^{K \times m} \tag{29}$$

$$\left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right]^T = \left[\begin{array}{cccccccc} u_0^T, & u_1^T, & \ldots, & u_{J-1}^T, & y_0^T, & y_1^T, & \ldots, & y_{J-1}^T \\ u_1^T, & u_2^T, & \ldots, & u_J^T, & y_1^T, & y_2^T, & \ldots, & y_J^T \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{K-1}^T, & u_K^T, & \ldots, & u_{K+J-2}^T, & y_{K-1}^T, & y_K^T, & \ldots, & y_{K+J-2}^T \end{array}\right] \in \mathbb{R}^{K\times(r+m)J} \quad (30)$$

The right hand sides of Eq. (29) and Eq. (30) are recognized as the matrices for linear regression of a strictly proper ARX model where $n_A = n_B = J$: Choosing $n_A = n_B = J$ in Eq. (27) gives that $Y$ and $X$ as underbraced in Eq. (27) are identical to Eq. (29) and Eq. (30) respectively. Comparing $X$ as underbraced in Eq. (27) to Eq. (30) shows that: (i) The arrangement (order) of the block columns is different and (ii) the signs (plus or minus) of the $y$ block elements are different. However, this will not affect the orthogonal projection as it does not affect the information available in each row of the matrices. Hence, Eq. (28) is identical to a strictly proper ARX model written on OLS regression form. This regression can be written as Eq. (31).

$$Y_{J|1}^T = \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right]^T \Theta + \epsilon_{J|1}^T \quad (31)$$

The main point of this derivation is obtained by rewriting Eq. (31) as Eq. (32).

$$\epsilon_{J|1}^T = Y_{J|1}^T - \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right]^T \Theta \quad (32)$$

Eq. (32) proves that the innovation process identified by the DSR E method, $\epsilon$, is mathematically identical to the residual of a strictly proper ARX model where $n_A = n_B = J$. This residual can also be expressed as the one-step-ahead prediction errors in a simulation running the ARX model on its own training dataset. It has now been proved that the first step of the DSR E algorithm can be replaced by the single-step ARX algorithm.

### 6.2 Approximating Step 2 of the DSR E Algorithm

Similar to the DSR E algorithm, the DARX algorithm considers the identified innovation process, $\epsilon$, as a known deterministic input. The deterministic / stochastic system identification problem has then been reduced to a deterministic system identification problem. This deterministic system can be written on the ARMAX form, Eq. (7), replacing the true (but unknown) innovation process, $\varepsilon$, by the identified innovation process, $\epsilon$, which in general is not exactly identical to the true innovation process. The model is then on the form of Eq. (33). Assume for simplicity that a single input, single output (SISO) system is to be modeled using a strictly proper ARMAX model of which $n_A = n_B = n_C = n$. Hence, the ARMAX polynomials are given by Eq. (34) to Eq. (36).

$$A(q)y_k = B(q)u_k + C(q)\epsilon_k \quad (33)$$

$$A(q) = 1 + a_1q^{-1} + \ldots + a_nq^{-n} \quad (34)$$

$$B(q) = b_1q^{-1} + \ldots + b_nq^{-n} \quad (35)$$

$$C(q) = 1 + c_1q^{-1} + \ldots + c_nq^{-n} \quad (36)$$

Inserting Eq. (34) to Eq. (36) into Eq. (33) and using that $q^{-1}y_k = y_{k-1}$ gives Eq. (37). Eq. (37) can be rewritten as Eq. (38). Writing Eq. (38) on linear regression form gives Eq. (39). The number of equations in this linear regression problem is $K - n$, where $K$ is the number of rows in Eq. (29) and Eq. (30).

$$y_k + a_1y_{k-1} + \ldots + a_ny_{k-n} = b_1u_{k-1} + \ldots + b_nu_{k-n} + \epsilon_k + c_1\epsilon_{k-1} + \ldots + c_n\epsilon_{k-n} \quad (37)$$

$$y_k - \epsilon_k = -a_1y_{k-1} - \ldots - a_ny_{k-n} + b_1u_{k-1} + \ldots + b_nu_{k-n} + c_1\epsilon_{k-1} + \ldots + c_n\epsilon_{k-n} \quad (38)$$

$$\begin{bmatrix} y_{J+n} & - & \epsilon_{J+n} \\ y_{J+n+1} & - & \epsilon_{J+n+1} \\ & \vdots & \\ y_{K+J-1} & - & \epsilon_{K+J-1} \end{bmatrix} \tag{39}$$

$$= \begin{bmatrix} -y_{J+n-1}, & \dots, & -y_J, & u_{J+n-1}, & \dots, & u_J, & \epsilon_{J+n-1}, & \dots, & \epsilon_J \\ -y_{J+n}, & \dots, & -y_{J+1}, & u_{J+n}, & \dots, & u_{J+1}, & \epsilon_{J+n}, & \dots, & \epsilon_{J+1} \\ \vdots & \ddots, & \vdots & \vdots & \ddots, & \vdots & \vdots & \ddots, & \vdots \\ -y_{K+J-2}, & \dots, & -y_{K+J-n-1}, & u_{K+J-2}, & \dots, & u_{K+J-n-1}, & \epsilon_{K+J-2}, & \dots, & \epsilon_{K+J-n-1} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

The ARX regression problem of Eq. (39) identifies not only the $A(q)$ and $B(q)$ polynomials, but also the $C(q)$ polynomial. Hence, the identified model is an ARMAX model on the form of Eq. (33). An ARMAX model can be converted to a state space model on the form of Eq. (5) and Eq. (6) and vice versa. Hence, the ARMAX model identified by solving the ARX regression problem of Eq. (39) can be converted to a state space model as generated by the DSR E algorithm presented in [1].

It has now been shown that: (i) The innovation process identified by the first step of the DSR E algorithm, $\epsilon$, is mathematically identical to the residual of an ARX identification where $n_A = n_B = J$. (ii) The second step of the DSR E algorithm, as presented in [1], can be replaced by ARX identification. From (i) and (ii) it is concluded that the DSR E algorithm can be approximated by a two-step ARX algorithm. The model identified by this approximation is an ARMAX model.

## 7 The Influence of Number of Samples, $N$, and the Parameter $J$

Consider Eq. (29) and Eq. (30): The lowest sample index used is 0 (in $u_0$ and $y_0$) and the highest sample index used is $J+K-1$ (in $y_{J+K-1}$). Hence, the number of samples, $N$, used to identify the innovation process is given by $N = K + J$.

As the mathematical derivation from [5] requires that $J \to \infty$ and $K \to \infty$, it is also implicitly requires that $N \to \infty$. These requirements can not be met in any practical system identification problems. In practical problems the number of samples, $N$, is given by the dataset. It is then a consideration to choose a proper value of $J$. $K$ is then given by $K = N - J$.

Consider a single input, single output (SISO) system. Then the OLS regression problem of Eq. (28) has $2J+K$ unknown values: (i) A SISO ARX model where $n_A = n_B = J$ has $2J$ unknown parameters, i.e. $J$ coefficients in the $A(q)$ polynomial and $J$ coefficients in the $B(q)$ polynomial. (ii) Unknown innovation processes for $K$ samples. The number of equations in the OLS problem is $K$. By choosing $J$ too large, the identified innovation process, $\epsilon$, will be smaller (absolute value) than the true innovation process, $\varepsilon$. This can be illustrated by an extreme choice of $J$: Choosing $J = N/3$. Then the ARX model will have $2J = 2N/3$ parameters. The number of equations in the OLS regression problem will be $K = N - J = N - N/3 = 2N/3$. Assuming that these $2N/3$ equations are linearly independent, the number of parameters to be identified in the OLS regression is equal to the number of linearly independent equations. Hence, the OLS problem is reduced to a deterministic set of linear equations. Then the residual, i.e. the identified innovation process, $\epsilon$, will be zero regardless of the true innovation process, $\varepsilon$. On the other hand, choosing $J$ too low will also conflict the derivation of [5].

In order to quantify the fit of the identified innovation process, $\epsilon$, to the true innovation process, $\varepsilon$, the fit criterion $W(N, J)$ of Eq. (40) has been defined.

$$W(N, J) \stackrel{\text{def}}{=} \frac{1}{N - J} \sum_{k=J+1}^{N} (\varepsilon_k - \epsilon_k(N, J))^2 \tag{40}$$

Two datasets were generated by simulations using a SISO ARMAX model where $n_A = n_B = n_C = 6$. During the simulations, a pseudo random binary signal (PRBS) was applied to the deterministic input, $u$. A uniformly distributed random number sequence was applied for simulating of the innovation process, $\varepsilon$. Different amplitude of the innovation process, $\varepsilon$, was applied for dataset no. 2 compared to dataset no. 1. Otherwise the datasets were generated using identical conditions.

For both datasets, $W(N, J)$ was plotted as function of $J$ for $N = 350$, $N = 1,000$, $N = 3,000$, and $N = 10,000$, i.e. total eight $J$ versus $W(N, J)$ plots. These plots are shown in Fig. 1 and Fig. 2. In Fig. 1 the innovation process, $\varepsilon$, is uniformly distributed in the interval $[-0.05, 0.05]$. In Fig. 2 the innovation process is

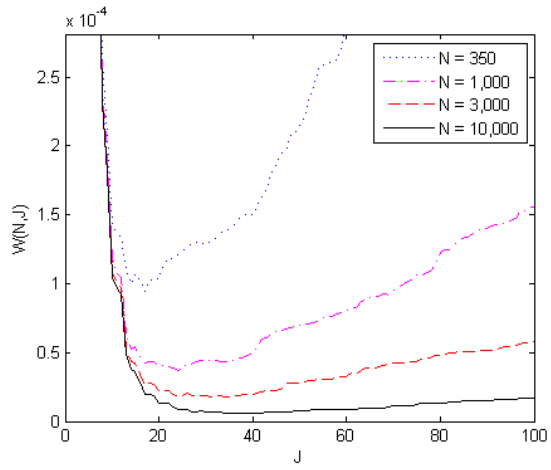


Fig. 1 The figure shows $W(N, J)$ plotted as function of $J$ for $N = 350$, $N = 1,000$, $N = 3,000$ and $N = 10,000$. The innovation process, $\varepsilon$, is uniformly distributed in the interval $[-0.05, 0.05]$.

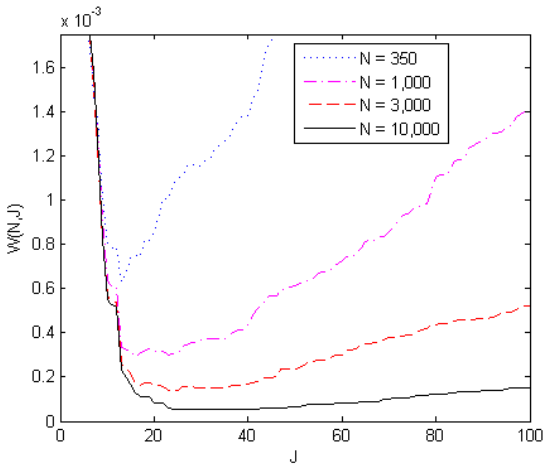


Fig. 2 The figure shows $W(N, J)$ plotted as function of $J$ for $N = 350$, $N = 1,000$, $N = 3,000$ and $N = 10,000$. The innovation process, $\varepsilon$, is uniformly distributed in the interval $[-0.15, 0.15]$.

uniformly distributed in the interval $[-0.15, 0.15]$. In other words: In Fig. 2 the amplitude of the innovation process is three times higher than in Fig. 1. Please note that the figures have different scaling of their respective Y-axes.

Based on the plots shown in Fig. 1 and Fig. 2, it seems reasonable to draw the following conclusions:

1. Increased value of $N$ gives better match between $\varepsilon$ and $\epsilon$ for the optimal choice of $J$, i.e. the value of $J$ giving the lowest $W(N, J)$. This is to be expected because the derivation from [5] assumes $N \to \infty$.
2. When ignoring some high frequency variations ("noise") on the curves, it seems (but can not be stated) that: (i) There is exact one minimum on each curve. (ii) The curves are strictly increasing as moving away from these minimums. This seems reasonable according to the discussion for choice of $J$ above.
3. As $N$ increases the optimal choice of $J$, i.e. the value of $J$ giving the minimum of $W(N, J)$, also increases. This is reasonable: As $N$ increases, the number of equations in the OLS problem of Eq. (28) also increases. Hence, the number of coefficients in the ARX model, $2J$, may increase without over-fitting the model.
4. As the amplitude of the innovation process, $\varepsilon$, increases, the fit criterion, $W(N, J)$, also increases. This is reasonable: When $\varepsilon$ and $\epsilon$ in general have larger values, also the difference between these values will be larger.

## 8 Comparing DSR E and DARX on Industrial Data

The DSR E and DARX algorithms have been compared on experimental data from the copper refining process of Xstrata Nikkelverk, Kristiansand, Norway. A single input, single output (SISO) system was modeled using DSR E and DARX. The input, $u$, is the mass flow from the roasting furnace to the copper leaching process. The output, $y$, is the concentration of sulphuric acid, $H_2SO_4$, in the flow from the copper leaching process to the electro winning. Before system identification, the input and output dataseries were preprocessed by (i) removing outliers, (ii) subtracting mean value, (iii) dividing by standard deviation, and (iv) compensating for the time delay form the input, $u$, to the output, $y$, by shifting the input dataseries with respect to the output dataseries. For DSR E, the parameters were chosen as $L = n = 20$, $g = 0$, and $J = 30$. For DARX, the parameters were chosen as $J = 30$ and $n_A = n_B = n_C = 20$. The DSR E implementation of the DSR Toolbox for MATLAB ([6]) was used for identifying the DSR E model. For identifying the DARX model, a MATLAB implementation of the DARX algorithm was written.

Based on the models identified by DSR E and DARX, ballistic simulations were run. Fig. 3 compares these simulations to the actual measured sulphuric acid concentration. The ballistic simulations fit the measured data poorly. This is to be expected because there are several other factors influencing the concentration of sulphuric acid that are not included in the models. By comparing the identified models to the model properties expected based on process knowledge, it has been verified that the modeled transfer functions have correct signs (in this case minus) and correct stability properties (in this case integrating).

The main point of this experiment is to show that DSR E and DARX give identical models: Fig. 3 shows that the model responses are identical when applying the mass flow signal, $u$, to the model inputs. Even though the values of the ballistic simulations vary within approximately $\pm 1.3$, the maximum difference between the

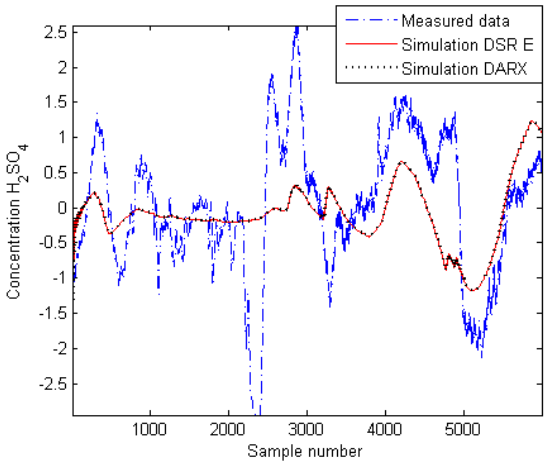


Fig. 3 The figure compares ballistic simulations using the models identified by DSR E and DARX. The actual measured sulphuric acid concentration, after being preprocessed, is also shown.

DSR E based simulation and the DARX based simulation is only $3.3 \times 10^{-11}$. This is a very strong indication that the models are identical.

To verify that the models actually are identical, the pole / zero plots and the step responses were also compared. Both the deterministic models, i.e. the transfer functions from $u$ to $y$, and the noise models, i.e. the transfer functions from $\varepsilon$ to $y$, were compared by pole / zero plots and step response plots. These plots are *not* shown in this paper. To the resolution of the plots, it was not possible to distinguish the models, neither by the pole / zero plots nor by the step response plots. It is then concluded that the models are identical beyond the model accuracy that can be expected from such modeling techniques.

## 9 Conclusions

The DSR E sub-space system identification algorithm can be approximated by a two-step ARX algorithm. The first step of the DSR E algorithm is mathematically identical to the identification of a strictly proper ARX model of which $n_A = n_B = J$: The innovation process identified by DSR E is identical to the residual of the ARX identification. During the second step of the DSR E algorithm the deterministic / stochastic system identification problem is reduced to a deterministic problem by considering the identified innovation process as a known deterministic input. In the DSR E algorithm, as presented in [1], this deterministic problem is solved by a deterministic sub-space system identification algorithm. However, this problem may instead be solved using the single-step ARX algorithm. Hence, each of the two steps of the DSR E algorithm may be replaced by the single-step ARX algorithm. This allows DSR E to be approximated by a two-step ARX algorithm.

The mathematical derivation of DSR E requires that $N \to \infty$ and $J \to \infty$, where $N$ is the number of samples and $J$ is as presented above. These requirements can not be met in any practical system identification problems. Simulations presented in this paper show that the innovation process is identified more accurately as $N$ increase. The optimal choice of $J$, i.e. the value of $J$ that gives the most accurate identification, increases as $N$ increases.

The DSR E algorithm and its two-step ARX approximation were compared by modeling a section of the copper refining process at Xstrata Nikkelverk, Kristiansand, Norway. The model identified by the DSR E algorithm and the model identified by the approximation are identical beyond the model accuracy that can be expected from such modeling techniques.

## 10 Acknowledgments

The authors are most grateful to Xstrata Nikkelverk, Kristiansand, Norway for providing datasets from the copper refining process and for allowing these data to be used in this paper.

## 11 References

[1] GW. Nilsen. *Topics in Open and Closed Loop Subspace System Identification: Finite Data-Based Methods, Ph.D. Thesis*. Norwegian University of Science and Technology / Telemark University College, Trondheim / Porsgrunn, Norway, 2006.

[2] L. Ljung. *System Identification - Theory for the User, 2nd Edition*. Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[3] D. Di Ruscio. Combined deterministic and stochastic system identification and realization: DSR - a subspace approach based on observations. *Modeling, Identification and Control*, 17, 1996.

[4] D. Di Ruscio. Subspace system identification of the Kalman filter: Open and closed loop systems. In *The 6th international conference on Computing, Communication and Control Technologies: CCCT 2008*, Orlando, Florida, June 29th - July 2rd 2008.

[5] D. Di Ruscio. Subspace system identification of the Kalman filter. *Modeling, Identification and Control*, 24:125–157, 2003.

[6] D. Di Ruscio. *DSR Toolbox for MATLAB*. Telemark University College, Porsgrunn, Norway, 2005.