Initialization Approach for Decoupling Polynomial NARX Models Using Tensor Decomposition [★]

Kiana Karami [*] **David Westwick** [**]

[*] *Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary, Alberta, T2N 1N4, Canada. (kiana.karami@ucalgary.ca).*
[**] *Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary, Alberta, T2N 1N4, Canada. (dwestwic@ucalgary.ca)*

Abstract: The Nonlinear Auto-regressive eXogenous input (NARX) model has been widely used in nonlinear system identification. It's chief disadvantages are that it is a black-box model that suffers from the curse of dimensionality, in that the number of parameters increases rapidly with the nonlinearity degree. One approach to dealing with these problems involves decoupling the nonlinearity, but this requires solving a non-convex optimization problem. Solving non-convex optimization problems has always been challenging due to the possibility of getting trapped in a sub-optimal local optima. As a result, these kinds of optimization problems are sensitive to the initial solution. Providing an appropriate initial solution can increase the likelihood of finding the globally optimal solution. In this paper, an initialization technique that uses the polynomial coefficients in a full, albeit low order, NARX model is proposed. This technique generates a tensor from the coefficients in the from full polynomial NARX model and applies a tensor factorization in order to generate an appropriate starting point for decoupled polynomial NARX model optimization problem. The proposed technique is applied to nonlinear benchmark problem and the results are promising.

Keywords: Nonconvex optimization, Decoupled polynomial, System identification, NARX model, Nonlinear system, Tensor, Waring Decomposition.

1. INTRODUCTION

The field of system identification has been a very active area since the late 1950's (Zadeh (1962); Åström and Eykhoff (1971)). In the beginning, linear system identification was the center of attention. But in some cases, linear models are not satisfying enough for the user, since most of the systems around us possess nonlinear behavior. Nonlinear system identification plays an important role when linear models are not precise enough to handle the real system's nonlinearity (Ljung (1999); Pintelon and Schoukens (2012)). The area of nonlinear system identification is an active area of study and research (Noël and Kerschen (2017); Schoukens and Ljung (2019)).

Strong nonlinear behaviors such as moving resonances, hysteresis etc., can only be modeled using internal dynamics models such as: block-oriented models with feedback, nonlinear state-space models (NLSS) or nonlinear equation error models, such as the NARX or NARMAX models (Schoukens and Ljung (2019); Billings (2013)). The common aspect of these models is that the poles of their best linear approximation (BLA) (Schoukens (2015)) can move depending on the experimental condition. The major advantage of the NARX model as compared to the other mentioned methods is, provided the nonlinearity is modelled using an expansion onto a known basis, it is linear in its parameters, so the optimization problem will be convex. This model was developed by Billings (2013) and has been used in the literature extensively (Caswell (2014); Chan et al. (2015); Ruiz et al. (2016); Alcan et al. (2019)).

In the NARX model, the current output of the system is related to the past inputs and past outputs:

$$\begin{aligned} y(t) = f(&y(t-1), \cdots, y(t-n_y), \\ &u(t), u(t-1), \cdots, u(t-n_u)) + e(t) \end{aligned} \tag{1}$$

where t is the discrete-time index, $u(t)$ is the input, $y(t)$ is the output, $e(t)$ is the equation error, n_u is the number of past input terms and n_y is the number of output terms.

When the nonlinear function f is a linear combination of known basis functions (Sjöberg et al. (1995)) such as a polynomial, the output is a linear function of the parameters which results in a linear regression. Since no more iterative optimization is needed, the global optimal solution is guaranteed. This is a reason why NARX models are accepted by industry and have been successfully used in industrial problems.

[★] This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN/06464-2015.

Function f can be expanded using various basis functions such as a polynomial (Jones and Billings (1989)), sigmoidal neural network (Alcan et al. (2019)), spline (Shmilovici and Aguilar-Martin (1999)) etc. For example, using the polynomial basis, so that f is a polynomial function of its arguments, the result is the polynomial NARX model:

$$\begin{aligned} & f(y(t-1), \cdots, y(t-n_y), \\ & \quad u(t), u(t-1), \cdots, u(t-n_u)) \\ & = \sum_{m=0}^{M} \sum_{p=0}^{m} \sum_{k_1=1}^{n_y} \cdots \sum_{k_p=k_p-1}^{n_y} \sum_{k_p+1=1}^{n_u} \cdots \sum_{k_m=k_m-1}^{n_u} \\ & \left[c_{p,q}(k_1, k_2, \cdots, k_m) \prod_{i=a}^{p} y(t-k_i) \prod_{i=p+1}^{m} u(t-k_i) \right] \end{aligned} \tag{2}$$

where $c_{p,q}(k_1, k_2, \cdots, k_m)$ are the m'th degree polynomial coefficients that corresponding to various combinations of the products of p delayed outputs and $q = m - p$ delayed inputs.

Although the NARX model is used commonly, it still has some drawbacks, for example, it is a black box model and very difficult to interpret, which means that there is no physical insight into the model. Also the number of parameters grows really quickly with the order, $n = n_u + n_y$ and M is the maximum degree. In the case of P-NARX models, assuming we are interested in a M degree nonlinearities, the number of parameters is calculated using (3)

$$P_F = \sum_{i=0}^{M} \binom{(n+i-1)}{i} = 1 + n + \frac{n(n+1)}{2} + \cdots \tag{3}$$

As mentioned before, one of the major disadvantages of polynomial NARX models is the rapid growth in the number of parameters. In order to tackle this problem, the full polynomial can be approximated using the structure in Fig. 1. Then the number of parameters will be given by:

$$P_D = ((n + M) \times r) + 1 \tag{4}$$

note that r is the number of branches in decoupled model. Provided r is relatively small, this will be less for the full model – and the polynomial order can be increased since the number of parameters grows linearly (or less) with the polynomial order.

This decoupling approach was proposed for multi-input multi-output (MIMO) systems by Dreesen et al. (2015a). The decoupling method has been used in many applications (Dreesen et al. (2015b); Esfahani et al. (2018); Karami et al. (2019a). in order to simplifies the MIMO polynomial, the decoupling algorithm replaces the multivariate polynomial with a linear mixing matrix, a bank of single-input single-output (SISO) polynomials followed by another linear mixing matrix.

We developed a similar approach for MISO systems (Westwick et al. (2018)) as shown in Fig. 1, the multivariate function, $f(.)$, will be replaced by a mixing matrix, V and a bank of univariate polynomials, $g(.)$, whose outputs are summed together. In that paper, it is shown that the decoupling method can reduce the number of parameters significantly.

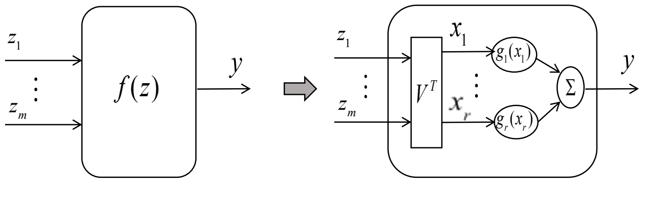


Fig. 1. Decoupling a MISO Nonlinearity Into SISO Polynomial Branches and a Mixing Matrix - in P-NARX model, z_i could be either a past input or a past output (see (2))

Despite the fact that the decoupling approach reduces the number of parameters significantly, there is still a major drawback with this method. By decoupling the polynomial NARX model, the new model is no longer linear the in parameters. The decoupled model is nonlinear in the elements of the mixing matrix, V, in this situation an iterative optimization procedure is needed. Since the optimization is non-convex and solved using an iterative search, the solution may converge to a sub-optimal local minimum. Thus, it is critically important to provide the iterative optimization with a good initial solution to maximize the chances that it converges to the global optimum.

In a previous paper (Karami et al. (2019b)), two different approaches for this initialization were proposed: A 3-way tensor was constructed by evaluating the Hessian of the polynomial at each sample in the identification data and stacking the resulting matrices. This tensor was factored using the Canonical Polyadic Decomposition (Vervliet et al. (2016)) (CPD) (either unstructured, or with the imposition of a constraint that guaranteed the underlying polynomial structure). In this paper a new initialization technique based on factoring a tensor constructed directly from the polynomial coefficients is proposed. The advantages of this technique are: 1) no differentiation is required (and hence no loss of information) and 2) the tensor size does not vary with data length (and is generally quite small). This paper is balanced as follows, Section 2 summarizes the iterative optimization used in decoupling polynomial NARX models. Section 3 introduces the new initialization technique. In Section 4 the results of applying the proposed technique to a nonlinear benchmark, the Silver-Box are shown. Lastly, Section 5 summarizes the paper.

1.1 Notation

Lower and upper case letters in a regular type-face, a, A, will refer to scalars, bold faced lower case letters, $\mathbf{a}$, refer to vectors, matrices are indicated by bold faced upper case letters, $\mathbf{A}$, and bold faced calligraphic script will be used for tensors, $\boldsymbol{\mathcal{A}}$. $\otimes$ and $\times_n$ will denote the Kronecker product and mode-n tensor product, respectively.

2. DECOUPLED POLYNOMIAL NARX MODEL

Dreesen et al. (2015a) and Westwick et al. (2018) developed decoupling techniques for MIMO and MISO polynomials respectively. In both cases, the multivariate polynomial was replaced by a bank consisting of several univariate polynomials as well as mixing matrices before, and in the MIMO case after, the bank of univariate polynomials.

In the case of a polynomial NARX model, the inputs to the model are past inputs and outputs, so the vector $\mathbf{z} = [z_1 \cdots z_m]^T$ in Fig. 1 becomes $\mathbf{z}(t) = [y(t-1) \cdots y(t-n_y)\, u(t-n_k)\, u(t-n_k-1) \cdots u(t-n_k-n_u)]^T$ where n_u and n_y are the number of past input and output samples used in the model and n_k is the input delay which is always greater than or equal to zero.

The output of a P-NARX model can be written in the form of a decoupled polynomial:

$$\hat{y}(t) = c_0 + \sum_{i=1}^{r} g_i(\mathbf{v}_i{}^T \mathbf{z}(t)) \tag{5}$$

where the $g_i(x)$ are univarite polynomials given by:

$$g_i(x_i(t)) = \sum_{j=1}^{M} c_{j,i} x_i^j(t) \tag{6}$$

In the decoupled representation in (5) and (6), $c_{j,i}$ represents the degree j polynomial coefficient in the i'th branch. The $\mathbf{v}_i$ are the columns of the mixing matrix where $\mathbf{V} = [\,\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_r\,]$. The intermediate signal, $x_i(k) = \mathbf{v}_i{}^T\mathbf{z}(k)$ is bridge that connects the inputs and the output of the model. Collecting all the intermediate signals for all branches at all measurement points we can rewrite the estimate output equation as:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{c} \tag{7}$$

where

$$\mathbf{X} = [\,\mathbf{1}_N\ \mathbf{X}_1\ \mathbf{X}_2\ \ldots\ \mathbf{X}_r\,] \tag{8}$$

$$\mathbf{X_i} = \left[\,\mathbf{x}_i\ \mathbf{x}_i^2\ \ldots\ \mathbf{x}_i^n\,\right] \tag{9}$$

$$\mathbf{x}_i = [\,x_i(1)\ x_i(2)\ \ldots\ x_i(N)\,]^T \tag{10}$$

The ultimate goal is to to find the optimal $\mathbf{V}$ and $\mathbf{c} = [\,c_0\ c_{1,1}\ \ldots\ c_{M,1}\ c_{2,1}\ \ldots\ c_{M,r}\,]^T$ that minimize the error between the measured output and the model output. The cost function of interest is:

$$V_N(\mathbf{V}, \mathbf{c}) = ||\mathbf{y} - \hat{\mathbf{y}}(\mathbf{V}, \mathbf{c})||_2^2 \tag{11}$$

This cost function is non-convex since the estimated output is nonlinear in the elements of the mixing matrix, $\mathbf{V}$. However, the optimization problem can be solved using separable least square (SLS) optimization. The elements of the vector $\mathbf{c}$ can be written as functions of the parameters of mixing matrix, $\mathbf{V}$, by solving for $\mathbf{c}$ in (7) via linear regression. Then we will end up with the separated cost function:

$$\hat{\mathbf{V}} = \arg\min_{\mathbf{V}} V_N(\mathbf{V}, \mathbf{c}(\mathbf{V})) \tag{12}$$

The optimization problem in (12) can be solved using using a quasi-Newton algorithm (Ljung (1999)). Since this is an iterative optimization over a non-convex error surface, a good initial solution for the optimization is necessary. Different approaches for initialization were assessed in (Karami et al. (2019b)). Those approaches involved computing the Hessian of the estimated output, $\hat{\mathbf{y}}$, with respect to the inputs, $\mathbf{z}(k)$. These methods involved heavy computation. A new approach for generating an initial estimate is proposed in Section 3.

3. OPTIMIZATION INITIALIZATION

As mentioned before the optimization problem needs a good initial estimate for the variables in the mixing matrix, $\mathbf{V}$. The polynomial coefficients, $\mathbf{c}$, are treated as closed-form functions of V, so they do not need to be initialized. This initialization is based on the Waring decomposition. The Waring problem has been extensively discussed in literature (Comon and Mourrain (1996); Oeding and Ottaviani (2013)). The classical Waring problem for polynomials indicates that a general homogeneous polynomial of n variables and degree kd can be represented as the sum of terms that are each a sum of degree d terms raised to the power k (Fröberg et al. (2012)). Such representations are often called Waring decomposition. The relationships between the Waring decomposition and various tensor decomposition have become an important research area (Comon and Mourrain (1996); Kolda and Bader (2009); Brachat et al. (2010); Schoukens and Rolain (2012)).

Let $y(\mathbf{z})$ be the output of a MISO polynomial, This function includes m generic inputs, $z_i, i = 1..m$, and that in a P-NARX model these can be either past outputs, or current or past inputs.

$$\begin{aligned} y(\mathbf{z}) =& y_0 + \sum_{i_1=1}^{m} \alpha_{i_1} z_{i_1} + \sum_{i_1=1}^{m}\sum_{i_2=i_1}^{m} \alpha_{i_1,i_2} z_{i_1} z_{i_2} \\ &+ \sum_{i_1=1}^{m} \cdots \sum_{i_n=i_{n-1}}^{m} \alpha_{i_1,\cdots,i_n} z_{i_1} \cdots z_{i_n} \end{aligned} \tag{13}$$

For the purpose of illustration, let $y(\mathbf{z})$ be a polynomial function with 3 inputs z_1, z_2 and z_3 of $3rd$ degree. The First order (linear) term in (13) is given by:

$$f_1(z) = \sum_{i_1=1}^{3} \alpha_{i_1} z_{i_1} = T_1^T \mathbf{z} \tag{14}$$

where T_1 includes the first order information in a $1st$ order tensor (vector):

$$T_1 = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{15}$$

For the second degree term:

$$f_2(z) = \sum_{i_1=1}^{3}\sum_{i_2=i_1}^{3} \alpha_{i_1,i_2} z_{i_1} z_{i_2} = \mathbf{z}^T T_2 \mathbf{z} \tag{16}$$

where T_2 collects the second order information in a $2nd$ order tensor (matrix):

$$T_2 = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} \end{bmatrix} \tag{17}$$

(16) can also be written in forms of tensor product as $f_2(z) = T_2 \times_1 \mathbf{z} \times_2 \mathbf{z}$. Then we will compute the SVD of T_2 as it is shown in Fig. 2. Let $T_2 = VSV^T$ be the SVD of T_2, then (16) can be written as $f_2(z) = (V^T\mathbf{z})^T SV^T\mathbf{z}$.

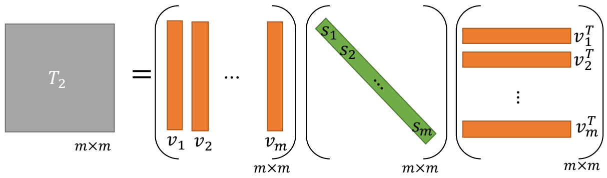


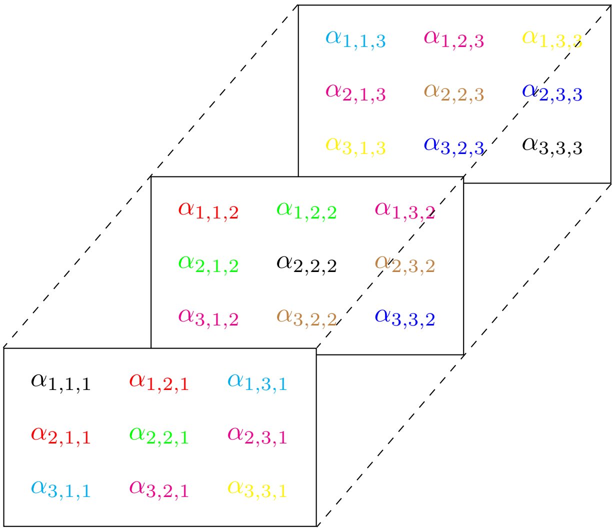
Fig. 2. Schematic Representation of Single Value Decomposition

Finally, for the third degree term:

$$f_3(z) = \sum_{i_1=1}^{3} \sum_{i_2=i_1}^{3} \sum_{i_3=i_2}^{3} \alpha_{i_1,i_2,i_3} z_{i_1} z_{i_2} z_{i_3}$$
$$= T_3 \times_1 \mathbf{z} \times_2 \mathbf{z} \times_3 \mathbf{z} \tag{18}$$

T_3 stacks up the third order information in a $3rd$ order tensor.

$T_3 =$


$\alpha_{1,1,3}$ $\alpha_{1,2,3}$ $\alpha_{1,3,3}$
$\alpha_{2,1,3}$ $\alpha_{2,2,3}$ $\alpha_{2,3,3}$
$\alpha_{3,1,3}$ $\alpha_{3,2,3}$ $\alpha_{3,3,3}$
$\alpha_{1,1,2}$ $\alpha_{1,2,2}$ $\alpha_{1,3,2}$
$\alpha_{2,1,2}$ $\alpha_{2,2,2}$ $\alpha_{2,3,2}$
$\alpha_{3,1,2}$ $\alpha_{3,2,2}$ $\alpha_{3,3,2}$
$\alpha_{1,1,1}$ $\alpha_{1,2,1}$ $\alpha_{1,3,1}$
$\alpha_{2,1,1}$ $\alpha_{2,2,1}$ $\alpha_{2,3,1}$
$\alpha_{3,1,1}$ $\alpha_{3,2,1}$ $\alpha_{3,3,1}$

(19)

After decomposing the tensor using CPD, it results in the tensor product of 3 copies of $V^T\mathbf{z}$ where V is the CPD factor.

$$T_3 = \sum_{i=1}^{r} \mathbf{v}_i \otimes \mathbf{v}_i \otimes \mathbf{v}_i \tag{20}$$

where $\mathbf{v}_i$'s are columns of the mixing matrix. Note that this could be expanded to higher order polynomial degrees. In general dimension of T_1 is $m \times 1$, T_2 is $m \times m$ and T_3 is $m \times m \times m$.

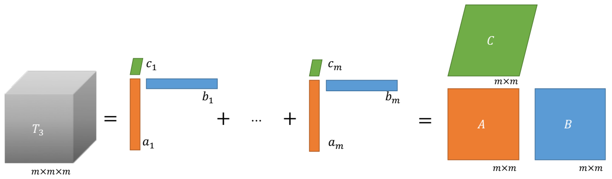


Fig. 3. Schematic Representation of Canonical Polyadic Decomposition

We will use the polynomial information to find an initial solution for the optimization problem in (12). It is possible to use all first, second and third order information of the polynomial by combining the first order vector, eigen vectors from the singular value decomposition (SVD) of the second order matrix (Fig 2) and CPD factors from canonical polyadic decomposition (CPD) of the third order tensor (Fig 3). Note that since T_2 and T_3 are symmetric, the decomposition factors are equal.

The other approach is to only use the third order information in T_3. This way we will get rid of redundant information from T_1 and T_2. we will use canonical polyadic decomposition approach as implemented in Tensorlab (Vervliet et al. (2016)) to decompose the tensor, T_3, and use its factors as an initial solution to the optimization problem.

Comparing this method to previous proposed initial solution (Karami et al. (2019b)) which involves computing the Hessian of the polynomial NARX model and evaluating it at all measurement points, this method is much less computationally expensive. Assuming that you have N data points, and n past inputs/outputs, the Hessian tensor will be $n \times n \times N$. On the other hand, the proposed method requires the CPD of a $n \times n \times n$ tensor. The quality of the optimization will be compared in next section.

4. BENCHMARK RESULTS

The proposed algorithm is tested on a nonlinear benchmark problem: the Silver-Box measurement data set (Wigren and Schoukens (2013)). More information can be found on the nonlinear benchmark website (`www.nonlinearbenchmark.org`).

4.1 Silver-Box

The Silver-Box benchmark represents a variety of mechanical oscillating processes which are an important set of nonlinear systems. Examples of these include the suspensions in motor vehicles. The circuitry in the Silver-Box represents a nonlinear mechanical resonating system using a mass, viscous damping and a nonlinear spring. Figure 4 shows a schematic representation of the Silver-Box differential equation:

$$m\frac{d^2y(t)}{dt} + d\frac{dy(t)}{dt} + k(y(t))y(t) = u(t) \tag{21}$$
$$k(y(t)) = a + by^2(t)$$

where m is the mass, d is the damping constant and $k(.)$ is the nonlinear spring. $u(t)$ is the input (force) and $y(t)$ is the output (displacement).

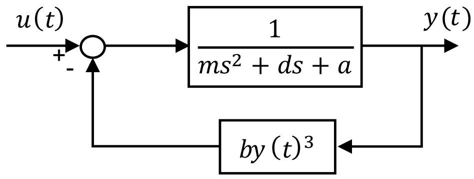


Fig. 4. Schematic Representation of Silver-Box (Schoukens et al. (2016))

The data set provided by Wigren and Schoukens (2013) for the Silver-Box includes 2 parts: The first part is a white Gaussian noise sequence filtered by a $9th$ order discrete time Butterworth filter with a 200Hz cut-off frequency, this part of the signal is used for validation of the model. The second part consists of 10 realizations of a random phase odd multi-sine, this part of the signal is used for identification and testing proposes. The accuracy of the identified model will be examined using the RMS fit:

$$FIT = 100 \times \left(1 - \sqrt{\frac{\sum_{t=1}^{N_t}(y(t) - \hat{y}(t))^2}{\sum_{t=1}^{N_t}(y(t) - y_{avg})^2}}\right) \tag{22}$$

The full 3rd order P-NARX model identified by Karami et al. (2019b) was used to generate the initial estimates using 3 past inputs and 3 past outputs. Separable least squares optimizations, using the method described in

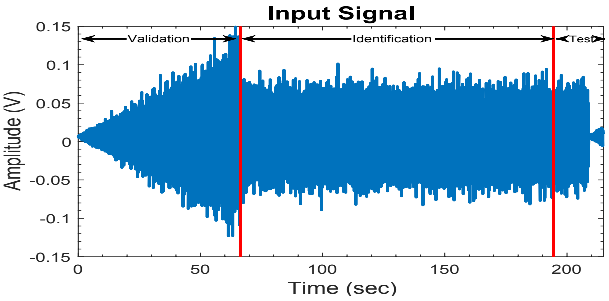


Fig. 5. Silver-Box Reference Signal (Input Signal).

(Karami et al. (2019b)) were initialized using a variety of starting points.

Figure 6 Shows the cost function of the optimization problem using two different starting points: the black line is when we used the proposed approach in this paper as an initial solution versus the magenta line which used the CPD of the Hessian as starting point. Although it takes longer for black line to settle, but it settled at slightly lower cost function. Both of these methods obtained a 99.81% fit accuracy in identification data. It is important to note that even though it took more iterations for the cost function to reach the optimal value for the proposed technique in this paper, the mathematical computation for finding the starting point is much cheaper than what was proposed in (Karami et al. (2019b)).

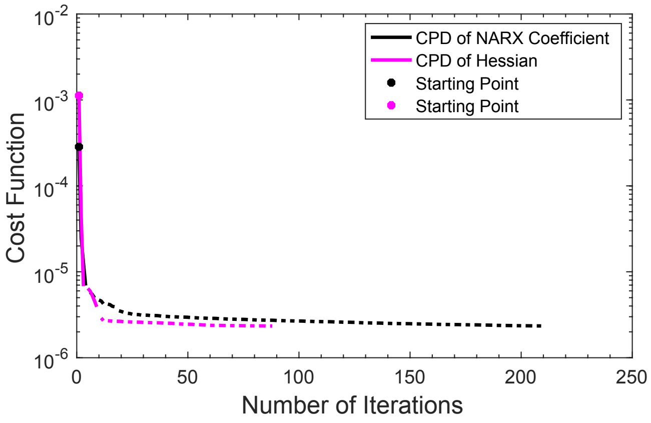


Fig. 6. Cost Function During Optimization of Decoupled Polynomial NARX Model of Silver-Box Benchmark Using Two Different Initial Solution

Table 1 compares the columns of the mixing matrices among the different initializations, reported as the normalized mean squared error between the matrices. The two Hessian-based solutions are almost exactly the same, and are very similar to the polynomial coefficient initialization, but there is little similarity with any of the random initializations. Table 2 computes the same statistic for the final solutions. Similar trends were noted among the final solutions.

		Initial Value			
		Random Average	Hessian	Hessian w/struc.	NARX Coeff.
Initial Value	Random Average	-	0.9060	0.9060	0.8994
	Hessian	0.9528	-	0.0000	0.0417
	Hessian w/struc.	0.8782	0.0000	-	0.0436
	NARX Coeff.	0.8709	0.0391	0.0391	-

Table 1. Comparison of the Initial Mixing Matrices, **V**, for Different Initialization Technique Using Normalized Mean Square Error

		Final Value			
		Random Average	Hessian	Hessian w/struc.	NARX Coeff.
Final Value	Random Average	-	0.7572	0.7572	0.7921
	Hessian	0.7572	-	0.0000	0.2188
	Hessian w/struc.	0.7572	0.0000	-	0.2188
	NARX Coeff.	0.7921	0.2188	0.2188	-

Table 2. Comparison of the Final Mixing Matrices, **V**, for Different Initialization Technique Using Normalized Mean Square Error

Table 3 shows the accuracy of the identified model in the triangular validation part of the signal for both prediction and simulation. Also, Fig. 7 demonstrate the prediction error on triangular validation signal. These results show that, using this technique we can achieve a model as accurate as other proposed initialization methods while the computational cost is very inexpensive.

	Triangle Validation Fit %	
	CPD of Hessian (Karami et al. (2019b))	CPD of NARX Coeff. (This Paper)
Prediction	99.77%	99.76%
Simulation	99.11%	98.77%

Table 3. Accuracy of Validation Data Set of 1-Step Ahead Prediction and Simulation for Silver-Box Data

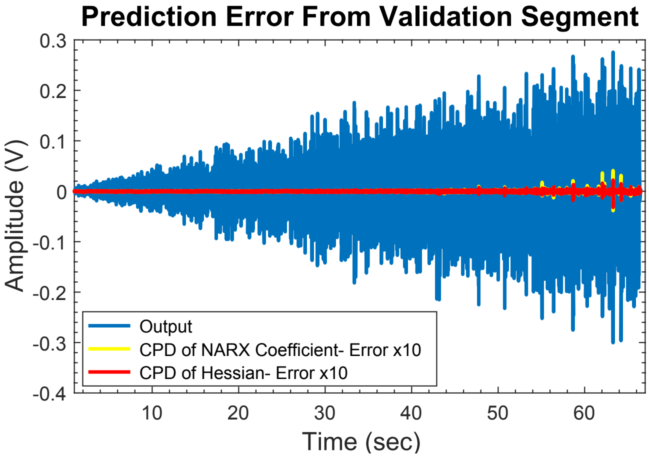


Fig. 7. Prediction Errors on the Validation Data for Silver-Box Using Two Different Initial Solution. Errors Are Scaled Up 10x for Better Visibility

5. CONCLUSION

We have proposed an initial solution for the decoupling of a P-NARX model. This technique collects the coefficients of the full P-NARX model into a 3-way tensor and then decomposes the tensor using the CPD. It then uses one of the three identical decomposition factors as the initial mixing matrix. The technique is applied to the Silver-Box benchmark problem. It leads to similar results (post optimization) as two previously described Hessian based approaches, but it is much cheaper in computation.

ACKNOWLEDGEMENTS

We thank Dr. Johan Schoukens (Vrije Universteit Brussel) for providing critical and valuable discussions regarding the ideas developed in this manuscript.

REFERENCES

Alcan, G., Unel, M., Aran, V., Yilmaz, M., Gurel, C., and Koprubasi, K. (2019). Predicting nox emissions in diesel engines via sigmoid narx models using a new experiment design for combustion identification. *Measurement*, 137, 71–81.

Åström, K.J. and Eykhoff, P. (1971). System identification—a survey. *Automatica*, 7(2), 123–162.

Billings, S.A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Brachat, J., Comon, P., Mourrain, B., and Tsigaridas, E. (2010). Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11-12), 1851–1872.

Caswell, J.M. (2014). A nonlinear autoregressive approach to statistical prediction of disturbance storm time geomagnetic fluctuations using solar data. *Journal of Signal and Information Processing*, 5(02), 42.

Chan, R.W., Yuen, J.K., Lee, E.W., and Arashpour, M. (2015). Application of nonlinear-autoregressive-exogenous model to predict the hysteretic behaviour of passive control systems. *Engineering Structures*, 85, 1–10.

Comon, P. and Mourrain, B. (1996). Decomposition of quantics in sums of powers of linear forms. *Signal Processing*, 53(2-3), 93–107.

Dreesen, P., Ishteva, M., and Schoukens, J. (2015a). Decoupling multivariate polynomials using first-order information and tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 36(2), 864–879.

Dreesen, P., Ishteva, M., and Schoukens, J. (2015b). Recovering wiener-hammerstein nonlinear state-space models using linear algebra. *IFAC-PapersOnLine*, 48(28), 951–956.

Esfahani, A.F., Dreesen, P., Tiels, K., Noël, J.P., and Schoukens, J. (2018). Parameter reduction in nonlinear state-space identification of hysteresis. *Mechanical Systems and Signal Processing*, 104, 884–895.

Fröberg, R., Ottaviani, G., and Shapiro, B. (2012). On the waring problem for polynomial rings. *Proceedings of the National Academy of Sciences*, 109(15), 5600–5602.

Jones, J.P. and Billings, S. (1989). Recursive algorithm for computing the frequency response of a class of nonlinear difference equation models. *International Journal of Control*, 50(5), 1925–1940.

Karami, K., Westwick, D., and Schoukens, J. (2019a). Identification of decoupled polynomial narx model using simulation error minimization. In *2019 American Control Conference (ACC)*, 4362–4367. IEEE.

Karami, K., Westwick, D.T., and Schoukens, J. (2019b). Applying polynomial decoupling methods to the polynomial narx model. *Mechanical Systems and Signal Processing (Under Revision)*.

Kolda, T.G. and Bader, B.W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3), 455–500.

Ljung, L. (1999). System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*.

Noël, J.P. and Kerschen, G. (2017). Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing*, 83, 2–35.

Oeding, L. and Ottaviani, G. (2013). Eigenvectors of tensors and algorithms for waring decomposition. *Journal of Symbolic Computation*, 54, 9–35.

Pintelon, R. and Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

Ruiz, L., Cuéllar, M., Calvo-Flores, M., and Jiménez, M. (2016). An application of non-linear autoregressive neural networks to predict energy consumption in public buildings. *Energies*, 9(9), 684.

Schoukens, J. and Ljung, L. (2019). Nonlinear system identification: A user-oriented roadmap.

Schoukens, J., Vaes, M., and Pintelon, R. (2016). Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Systems*, 36(3), 38–69.

Schoukens, M. (2015). *Identification of Parallel Block-Oriented Models starting from the Best Linear Approximation*. Ph.D. thesis, VRIJE UNIVERSITEIT BRUSSEL.

Schoukens, M. and Rolain, Y. (2012). Cross-term elimination in parallel wiener systems using a linear input transformation. *IEEE Transactions on Instrumentation and Measurement*, 61(3), 845–847.

Shmilovici, A. and Aguilar-Martin, J. (1999). Adaptive matching pursuit of narx models with spline basis functions. *International journal of systems science*, 30(8), 879–888.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.Y., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12), 1691–1724.

Vervliet, N., Debals, O., Sorber, L., Van Barel, M., and De Lathauwer, L. (2016). Tensorlab user guide. *Available on: http://www. tensorlab. net*.

Westwick, D.T., Hollander, G., Karami, K., and Schoukens, J. (2018). Using decoupling methods to reduce polynomial narx models. *IFAC-PapersOnLine*, 51(15), 796–801.

Wigren, T. and Schoukens, J. (2013). Three free data sets for development and benchmarking in nonlinear system identification. In *Proc. 2013 Eur. Control Conf.(ECC2013)*, 17–19.

Zadeh, L.A. (1962). From circuit theory to system theory. *Proceedings of the IRE*, 50(5), 856–865.