21th IMEKO TC4 International Symposium on Understanding
the World through Electrical and Electronic Measurement,
and 19th International Workshop on ADC Modelling and Testing
Budapest, Hungary, September 7-9, 2016

A polynomial nonlinear state-space toolbox for Matlab

Koen Tiels

Vrije Universiteit Brussel, Dept. ELEC, Pleinlaan 2, B-1050 Brussels, Belgium,
Koen.Tiels@vub.ac.be, +32 (0)2 629 36 65, skype: Koen Tiels

***Abstract* – A polynomial nonlinear state-space (PNLSS) model can capture many nonlinear dynamic behaviors and has been successfully applied in a large range of applications. The model structure can easily deal with multiple input multiple output (MIMO) systems. This paper presents PNLSS 1.0 [1], a toolbox to simulate and estimate PNLSS models in Matlab.**

***Keywords*—** System identification, State-space, Nonlinear systems, Matlab toolbox

I. INTRODUCTION

All real-life systems behave nonlinearly to a certain extent. Nevertheless, linear models are often used to describe their behavior since linear models can be easily interpreted, the linear system theory is mature, and there are many tools available. When the nonlinear distortions become too large, however, the system's behavior cannot be captured accurately enough by a linear model, and a nonlinear model is required.

Many different nonlinear model structures have been proposed in the past, e.g. Volterra series [15], block-oriented models [5], NARMAX models [1], nonlinear state-space models [16], neural networks [9], etc. This paper opts for the polynomial nonlinear state-space (PNLSS) framework [11].

A PNLSS model is fairly flexible in the sense that it can exhibit many nonlinear dynamic behaviors, like hysteresis [10], subharmonics, and chaotic behavior. Moreover, this modeling framework has been successfully applied in the past on a large range of applications (e.g. mechanical [11], electronic [12], hydrostatic [2, 12]). The model is nonlinear in its parameters, but an initialization with a linear model often turns out to be sufficient. Furthermore, the model structure can easily deal with multiple input multiple output (MIMO) systems.

This paper presents a Matlab toolbox that helps the user with the identification of a PNLSS model.

The rest of this paper is organized as follows. Section ii. describes the PNLSS model structure and its identification, as well as the main functions in the PNLSS 1.0 Matlab toolbox that implement this identification procedure. Section iii. provides an overview of all the functions in the toolbox. Section iv. presents the contributions of this paper and Section v. presents future work.

[1]The toolbox will be available online at http://homepages.vub.ac.be/∼ktiels/pnlss.html

II. DESCRIPTION OF THE METHOD

This section first describes the polynomial nonlinear state-space model structure. Next, it discusses the identification procedure for the model. This procedure consists of three simple steps: estimation of a nonparametric linear model, estimation of a parametric subspace model, and nonlinear optimization of the model parameters.

A. PNLSS model

A polynomial nonlinear state-space model [11] is a natural extension of a discrete-time linear state-space model. The state and output equation of the model are given by

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{\zeta}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \quad (1)$$

and

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{F}\boldsymbol{\eta}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad (2)$$

respectively. Here, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the input vector at time t, $\boldsymbol{y}(t) \in \mathbb{R}^p$ is the corresponding output vector, and $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector. Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the state matrix, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ is the input matrix, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{p\times m}$ is the feed-through matrix. The multivariate functions $\boldsymbol{\zeta} : \mathbb{R}^{n+m} \to \mathbb{R}^{n_\zeta}$ and $\boldsymbol{\eta} : \mathbb{R}^{n+m} \to \mathbb{R}^{n_\eta}$ are monomials with user-chosen degrees larger than one. The matrices $\boldsymbol{E} \in \mathbb{R}^{n\times n_\zeta}$ and $\boldsymbol{F} \in \mathbb{R}^{p\times n_\eta}$ contain the corresponding monomial coefficients.

The state equation captures the dynamics of the model as it describes how the states of the system evolve as a function of the current states and inputs. By including the nonlinear terms $\boldsymbol{E}\boldsymbol{\zeta}$ and $\boldsymbol{F}\boldsymbol{\eta}$, nonlinear dynamics can be captured.

The parameters in the PNLSS model are the elements of the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{E}$, and $\boldsymbol{F}$, i.e. the parameter vector $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta}^T = [vec(\boldsymbol{A})^T\ vec(\boldsymbol{B})^T\ vec(\boldsymbol{C})^T\ vec(\boldsymbol{D})^T\ vec(\boldsymbol{E})^T\ vec(\boldsymbol{F})^T]. \quad (3)$$

Note that the PNLSS model is a black-box model. In general, this means that no physical interpretation can be given to the model parameters. Moreover, no information of the internal structure of the system is used when identifying the PNLSS model, only measured inputs and outputs are used.

B. *Identification procedure*

The identification of a PNLSS model follows an output-error framework. The goal is thus to identify the model parameters $\boldsymbol{\theta}$ such that the difference between the measured output $\boldsymbol{y}_m(t)$ and the modeled output $\boldsymbol{y}(t)$ is minimized in (weighted) least-squares sense. The cost function is either formulated in the time domain:

$$K_{\text{time}}(\boldsymbol{\theta}) = \sum_{t=1}^{N} \left\| \boldsymbol{W}_{\text{time}}^{T}(t)(\boldsymbol{y}(t,\boldsymbol{\theta}) - \boldsymbol{y}_m(t)) \right\|^2 \quad (4)$$

or in the frequency domain:

$$K_{\text{freq}}(\boldsymbol{\theta}) = \sum_{k=1}^{N_F} \boldsymbol{\epsilon}^H(k,\boldsymbol{\theta}) \boldsymbol{W}_{\text{freq}}(k) \boldsymbol{\epsilon}(k,\boldsymbol{\theta}), \quad (5)$$

where $\boldsymbol{W}_{\text{time}} \in \mathbb{R}^{p\times N}$ is the weighting matrix in the time domain,

$$\boldsymbol{\epsilon}(k,\boldsymbol{\theta}) = \boldsymbol{Y}(k,\boldsymbol{\theta}) - \boldsymbol{Y}(k) \quad (6)$$

is the spectrum of the output error at frequency line k, N_F is the number of excited frequencies in the positive half of the frequency band, and $\boldsymbol{W}_{\text{freq}}(k) \in \mathbb{C}^{p\times p}$ is a user-chosen frequency domain weighting matrix. Typically, $\boldsymbol{W}_{\text{freq}}(k)$ is the inverse of the covariance of the output spectrum, which can be easily calculated by the toolbox when using periodic excitation signals. The frequency domain weighting also allows the user to select the frequency band of interest.

The modeled output is nonlinear in its parameters, so that the identification of a PNLSS model is a nonlinear optimization problem. This requires good initial estimates to converge to at least a good local minimum. Here, the PNLSS model will be initialized from the best linear approximation (BLA) [14]. Next, a linear state-space model will be estimated on the nonparametric BLA using a frequency domain subspace method [8, 13]. Finally, the model parameters are optimized with a Levenberg-Marquardt algorithm [7, 14]. These steps are briefly detailed in the rest of this section, together with the main functions to perform these steps with the toolbox.

B..1 Estimation of the best linear approximation

To keep the notation simple, this section explains the best linear approximation of a single input single output (SISO) system. The extension of the estimation of the BLA for MIMO systems is explained in [3, 4].

The BLA [14] of a system with input $u(t)$ and output $y(t)$ is defined for a class of input signals as the linear system whose output approximates the system's output best in mean-square sense around the operating point, i.e.

$$G_{\text{BLA}}(k) = \arg\min_{G(k)} E_u \left\{ \left\| \tilde{Y}_m(k) - G(k)\tilde{U}(k) \right\|^2 \right\} \quad (7)$$

where $\tilde{Y}_m(k)$ and $\tilde{U}(k)$ are the spectra of the measured output and the input from which the means are removed. The expected value $E_u\{\cdot\}$ is the ensemble average over the given class of input signals.

This paper considers the input signals to be random-phase multisines:

$$u(t) = \sum_{k=-N_F/2+1}^{N_F/2} U_k e^{j2\pi \frac{k}{N} t}, \quad (8)$$

for $t = 1, \ldots, N$, and where the Fourier coefficients $U_k = {U_{-k}}^* = |U_k| e^{j\phi_k}$ are either zero (the harmonic is not excited) or have a normalized amplitude $|U_k| = \frac{1}{\sqrt{N}} \check{U}(\frac{k}{N})$, where $\check{U}(\frac{\omega}{2\pi}) \in \mathbb{R}^+$ is a uniformly bounded function ($\check{U}(\frac{\omega}{2\pi}) \leq C_U/\sqrt{2} < \infty$) with a countable number of discontinuities. These signals belong the the class of extended Gaussian signals, which amongst others also include Gaussian noise. The periodic nature of the random-phase multisine signal, however, enables a separation of the noise and nonlinear distortion.

The toolbox offers a function to generate random-phase multisines with a flat amplitude spectrum:

```
>> u = fMultisine(N,kind,M,R);
```

where `N` is the number of samples per period, `kind` is either a string that determines which frequency lines are excited (all, odd, even, ...) or is a vector with the excited frequency lines, `M` is the last excited frequency line and `R` is the number of phase realizations. The toolbox also offers a function to estimate the frequency response function (FRF) of the BLA, the total (sum of noise and nonlinear distortion) and noise distortion:

```
>> [G,covGtotal,covGnoise] = fCovarFrf(U,Y);
```

by applying the robust method [14] to the input and output spectra `U` and `Y`. This nonparametric analysis helps the user in the decision of whether or not to spend time in estimating a nonlinear model, since the difference between the total and noise distortion level tells the user how much can be gained by going from a linear to a nonlinear model.

B..2 Estimation of a subspace model

The next step is to estimate a parametric discrete-time linear state-space model on the estimated nonparametric BLA. To do that, the frequency domain subspace identification method of [8] is used with the frequency domain

weighting in [13]. This already provides a good initial estimate, but optionally, a Levenberg-Marquardt optimization on the linear model parameters can be performed. The subspace method and the optimization are combined as

```
>> models = fLoopSubSpace(freq,G,covGtotal,
nAll,maxr,maxIter,forcestable,showfigs,fs);
```

where `freq` is the vector of excited frequencies, `nAll` is a vector with model orders that are scanned, `maxr` is the maximum value of the subspace dimension parameter `r` over which a scan is done, `maxIter` is the maximum number of iterations of the Levenberg-Marquardt optimization, `forcestable` is a boolean that indicates whether or not to force a stable parametric model (with pole reflection), `showfigs` is a boolean that indicates whether or not to display the comparison of the BLA and the parametric models, and `fs` is the sampling frequency. The variable `models` contains the state-space matrices of the best models found for each scanned model order. The state-space matrices for model order `n` can be selected with

```
>> [A,B,C,D] = models{n}{:};
```

B..3 Optimization of the model parameters

Next, these linear state-space matrices are put in a PNLSS model form with

```
>> model_init = fCreateNLSSmodel(A,B,C,D,nx,
ny,T1,T2,0);
```

where `model_init` is a structure that contains the properties of the PNLSS model, `nx` and `ny` are vectors with the selected nonlinear degrees in $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$ respectively, `T1` and `T2` are transient settings, and the last parameter is obsolete and put to zero. The toolbox also allows to select which polynomial coefficients in $\boldsymbol{E}$ and $\boldsymbol{F}$ are free for optimization.

Finally, the parameters of the initial model are optimized with a Levenberg-Marquardt algorithm:

```
>> [model,y_mod,models] = fLMnlssWeighted(u,y
,model_init,nIter,W);
```

where `u` and `y` contain the estimation data and `nIter` is the number of iterations. Depending on the dimensions of the weighting matrix `W`, frequency domain weighting (if `W` is a $p \times p \times \frac{N}{2}$ array), time domain weighting (if `W` is a $N \times p$ array) or no weighting (`W = []`) is applied. We advice the user to start the modeling with uniform weighting (= no weighting) due to the possible presence of dominant model errors. The structure `model` is the best model on the estimation data and `y_mod` is its modeled output. The models after each successful iteration are stored in `models`. This allows the user to select the best model on a validation data set, which is recommended to avoid overfitting. The output of a model on a validation data set can be calculated with

```
>> yval = fFilterNLSS(models(i),uval);
```

Optionally, the initial states $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ and the initial inputs $\boldsymbol{u}_0 = \boldsymbol{u}(0)$ can also be estimated.

III. OVERVIEW OF THE TOOLBOX

The toolbox consists of a number of Matlab functions that implement the methods described in Section ii.. In particular, there are functions available to generate multisine signals and do transient handling, to estimate nonparametric frequency response functions and the total and noise distortion levels, to estimate parametric subspace models, to calculate the Jacobians of the subspace and PNLSS models, to do Levenberg-Marquardt optimization on the model parameters, and to construct and simulate the PNLSS model. There are also some utility functions available for plotting, etc. Furthermore, there is a tutorial example that explains the work-flow of estimating a PNLSS model.

IV. NOVELTIES IN THE PAPER

The PNLSS modeling approach has been successfully applied on a large number of applications in the past. This paper presented PNLSS 1.0, a Matlab toolbox that makes the PNLSS modeling approach publicly available and easily accessible.

The source code of the toolbox is available. Each of the functions is documented and most functions contain one or more examples on how to use them. Like this, besides being a tool for modeling the nonlinear dynamic input/output behavior of a system, the toolbox can also be a tool for learning about nonlinear system identification techniques.

V. FUTURE WORK

The successor of PNLSS 1.0 will have an object-oriented implementation in Matlab. This will make it easier to include (external) modules to perform specific tasks (initialization of the model, specification of the cost function, calculation of the Jacobian, optimization of the model parameters). The toolbox is fully implemented in Matlab, but with the possibility to include external modules, speed-ups of the time-critical parts will be realized.

Currently, only polynomial nonlinearities are implemented in the toolbox. The functionality of the toolbox will be extended by also allowing for non-polynomial nonlinearities [6].

A third improvement will be the ability to impose structure on the state-space matrices. This will allow the user to obtain some physical interpretation of the model.

VI. ACKNOWLEDGMENT

Many members of the nonlinear system identification team of the ELEC Department have contributed to this software. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement n. 320378.

REFERENCES

[1] Sheng Chen and Stephen A. Billings. Representations of non-linear systems: the NARMAX model. *International Journal of Control*, 49(3):1013–1032, 1989.

[2] Tom Coen, Johan Paduart, Jan Anthonis, Johan Schoukens, and Josse De Baerdemaeker. Nonlinear system identification on a combine harvester. In *Proceedings of the 2006 American Control Conference*, Minneapolis, MN, USA, June 2006.

[3] Tadeusz P. Dobrowiecki and Johan Schoukens. Linear approximation of weakly nonlinear MIMO systems. *IEEE Transactions on Instrumentation and Measurement*, 56:887–894, 2007.

[4] Tadeusz P. Dobrowiecki, Johan Schoukens, and Patrick Guillaume. Optimized excitation signals for MIMO frequency response function measurements. *IEEE Transactions on Instrumentation and Measurement*, 55:2072–2079, 2006.

[5] Fouad Giri and Er-Wei Bai, editors. *Block-oriented Nonlinear System Identification*. Springer, first edition, 2010.

[6] Anna Marconato, Jonas Sjöberg, Johan A.K. Suykens, and Johan Schoukens. Improved initialization for nonlinear state-space modeling. *IEEE Transactions on Instrumentation and Measurement*, 63:972–980, 2014.

[7] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1963.

[8] Tomas McKelvey, Hüseyin Akçay, and Lennart Ljung. Subspace-based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41:960–979, 1996.

[9] Oliver Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, 2001.

[10] Jean-Philippe Noël, Alireza Fakhrizadeh Esfahani, Gaëtan Kerschen, and Johan Schoukens. Hysteresis identification using nonlinear state-space models. In *Presented at the 34th International Modal Analysis Conference (IMAC)*, Orlando, FL, USA, January 2016. 18 pages.

[11] Johan Paduart, Lieve Lauwers, Jan Swevers, Kris Smolders, Johan Schoukens, and Rik Pintelon. Identification of nonlinear systems using Polynomial Nonlinear State Space models. *Automatica*, 46:647–656, 2010.

[12] Johan Paduart, Johan Schoukens, Rik Pintelon, and Tom Coen. Nonlinear state space modelling of multivariable systems. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.

[13] Rik Pintelon. Frequency-domain subspace system identification using non-parametric noise models. *Automatica*, 38:1295–1311, 2002.

[14] Rik Pintelon and Johan Schoukens. *System Identification: A Frequency Domain Approach*. Wiley-IEEE Press, second edition, 2012.

[15] Martin Schetzen. *The Volterra & Wiener Theories of Nonlinear Systems*. Krieger Publishing Company, Malabar, Florida, 2006.

[16] Vincent Verdult. *Nonlinear System Identification: A State-Space Approach*. PhD thesis, University of Twente, 2002.