Message Passing-based System Identification for NARMAX Models

Albert Podusenko*, Semih Akbayrak*, İsmail Şenöz*, Maarten Schoukens and Wouter M. Kouw

***Abstract*— We present a variational Bayesian identification procedure for polynomial NARMAX models based on message passing on a factor graph. Message passing allows us to obtain full posterior distributions for regression coefficients, precision parameters and noise instances by means of local computations distributed according to the factorization of the dynamic model. The posterior distributions are important to shaping the predictive distribution for outputs, and ultimately lead to superior model performance during 1-step ahead prediction and simulation.**

I. Introduction

Noise models (e.g. NOE, NAR(MA)X) are important components of nonlinear system identification techniques, for instance to capture impacts by external disturbances or variations in system behaviour [1]. Techniques such as Extended Least-Squares [1, Sec. 2.2.2.2] treat noise as a point-wise prediction error, i.e., the difference between the observation and the model's prediction. Here, we regard the noise instances as latent variables and infer full posterior distributions simultaneously with model parameters through variational Bayes. This means more information is included in the marginalization to obtain the posterior predictive distribution for future outputs, which increases performance.

In earlier work on Bayesian identification for NARMAX models, we obtained point estimates for the noise instances by collapsing our predictive distribution to its most probable value (*maximum a posteriori*) [2]. However, collapsing a distribution means disregarding uncertainty, which entails loss of - potentially valuable - information. This was already recognized by Fujimoto & Takaki who considered a latent variable approach to noise instance estimation in ARMAX models [3]. Later work extended this to non-Gaussian noise using sparse finite and infinite mixture models [4]. We generalize this approach to the NARMAX setting. This is not trivial because nonlinearities lead to analytically intractable posterior densities [5, Ch. 5]. We use the unscented transform to tackle this problem. Other approaches to Bayesian NARMAX identification achieved exciting results using (reversible-jump) Markov Chain Monte Carlo techniques [6]. However, Monte Carlo sampling is typically computationally much heavier than variational Bayes [7].

The proposed approach is implemented as a Forney-style Factor Graph (FFG) of the probabilistic NARMAX model and uses message passing to compute posterior distributions [8]. The advantage of FFGs is that their modular, plug-in type structure allows one to extend models without re-deriving earlier update equations. Furthermore, the computation can be distributed along nodes and edges by exploiting the factorization of the probabilistic model, producing an efficient algorithm [8], [9]. Our implementation is based on ReactiveMP.jl, a software package for automatic message passing in factor graphs [10].

Our main contributions are:

- We formulate a probabilistic polynomial NARMAX model with noise as latent random variables.
- We propose an inference algorithm with which coefficients, noise instances and noise precision parameters may be estimated simultaneously.
- We derive a posterior predictive distribution for simulating future outputs.

The proposed NARMAX identification algorithm is evaluated on both a 1-step ahead and a simulation experiment, demonstrating excellent performance.

II. Notation

We denote scalars with lowercase letters x, vectors with lowercase boldface letters $\mathbf{x}$, and matrices with uppercase boldface letters $\mathbf{X}$. Functions with multivariate outputs are denoted with boldface letters as well. Subscripts are used to index variables and functions, e.g. x_k, while superscripts denote association with variables, such as $\mathbf{m}^{\mathbf{x}}$ denoting the mean of $\mathbf{x}$. We denote expectations of a function $f(x)$ with respect to $p(x)$ as $\mathbb{E}_{p(x)}[f(x)]$, expectations of a multivariate function $f(\mathbf{x}, \mathbf{y})$ with respect to all variables except $\mathbf{x}$ as $\mathbb{E}_{\setminus p(\mathbf{x})}[f(\mathbf{x}, \mathbf{y})]$, and integration of $f(\mathbf{x}, \mathbf{y})$ with respect to all variables but $\mathbf{x}$ as $\int f(\mathbf{x}, \mathbf{y}) \mathrm{d} \setminus \mathbf{x}$.

III. System

Consider a discrete-time dynamical system where $u_k \in \mathbb{R}$ is a measured input signal, $y_k \in \mathbb{R}$ is a measured output signal and $e_k \in \mathbb{R}$ is noise. In a NARMAX system, the output y_k is generated as a function of the current input u_k, a series of previous inputs u_{k-1}, .. u_{k-n_a}, previous outputs y_{k-1}, .. y_{k-n_b}, previous noise instances e_{k-1}, .. e_{k-n_e} and the current noise instance e_k. Let us define two collections:

$$\mathbf{x}_k \triangleq [\, u_k,\, u_{k-1},\, \ldots\, u_{k-n_a},\, y_{k-1},\, \ldots\, y_{k-n_b}\,]^\top \quad (1)$$

$$\mathbf{h}_k \triangleq [\, e_{k-1},\, \ldots\, e_{k-n_e}\,]^\top. \quad (2)$$

Based on (1) and (2) a NARMAX system can be written as:

$$y_k = f(\mathbf{x}_k, \mathbf{h}_k) + e_k \text{ , where } \quad e_k \sim \mathcal{N}(0, \gamma^{-1}) \quad (3)$$

with precision parameter γ. All e_k are independent and identically distributed. The function f is assumed to be time-invariant, non-linear, and continuous everywhere.

* AP, SA and IS contributed equally. All authors are part of the dept. of Electrical Engineering, TU Eindhoven, in Eindhoven, the Netherlands.

IV. Model specification

We start our model construction with the evolution of the noise vector $\mathbf{h}_k$:

$$\mathbf{h}_{k+1} = \mathbf{S}\,\mathbf{h}_k + \mathbf{c}e_k\,, \tag{4}$$

where $\mathbf{S}$ is a shift operator that pushes all elements down (dropping the last one) and $\mathbf{c}$ is a vector that adds the most recent instance as the new first element:

$$\mathbf{S} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathrm{I}_{n_e\text{-}1} & \mathbf{0} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^\top. \tag{5}$$

Note that although the vector $\mathbf{h}_k$ evolves over time, the individual noise instances remain independent of each other. We describe the relationship between the output y_k and the concatenation of data and noise $[\mathbf{x}_k\ \mathbf{h}_k]$ with a polynomial regression model:

$$y_k = \boldsymbol{\theta}^\top \boldsymbol{\phi}([\mathbf{x}_k\ \mathbf{h}_k]) + e_k\,. \tag{6}$$

The multivariate polynomial $\boldsymbol{\phi}$ of degree d performs a basis expansion $\boldsymbol{\phi} : \mathbb{R}^M \to \mathbb{R}^D$, where $M = 1 + n_a + n_b + n_e$ and D is a binomial coefficient choosing d elements from a set of size $M+d$. Writing (6) and (4) in probabilistic state-space model form gives:

$$p(e_k \mid \gamma) = \mathcal{N}(e_k \mid 0, \gamma^{-1}) \tag{7}$$

$$p(\mathbf{h}_{k+1} \mid \mathbf{h}_k, e_k) = \delta\big(\mathbf{h}_{k+1} - (\mathbf{S}\,\mathbf{h}_k + \mathbf{c}e_k)\big) \tag{8}$$

$$p(z_k \mid \boldsymbol{\theta}, \chi, \mathbf{x}_k, \mathbf{h}_k) = \mathcal{N}\left(z_k \mid \boldsymbol{\theta}^\top \boldsymbol{\phi}\left([\mathbf{x}_k\,\mathbf{h}_k]\right), \chi^{-1}\right) \tag{9}$$

$$p(y_k \mid z_k, e_k) = \delta(y_k - (z_k + e_k))\,, \tag{10}$$

where δ refers to a Dirac delta distribution used to characterize deterministic mappings [5]. The variable z_k is an auxiliary state based on a "soft dot product" for the polynomial regression function, which is necessary because products of Gaussian random variables require approximation (see Sec. V-A). The final likelihood (10) is the deterministic sum of the auxiliary state and the current noise instance.

We specify the following prior distributions for regression coefficients $\boldsymbol{\theta}$, overall noise precision γ, soft product precision χ and the initial noise vector $\mathbf{h}_1$:

$$\begin{aligned} p(\boldsymbol{\theta}) &= \mathcal{N}\left(\boldsymbol{\theta} | \mathbf{m}^{\boldsymbol{\theta}}, \mathbf{P}^{\boldsymbol{\theta}}\right), & p(\mathbf{h}_1) &= \mathcal{N}\left(\mathbf{h}_1 | \mathbf{m}^{\mathbf{h}_1}, \mathbf{P}^{\mathbf{h}_1}\right) \\ p(\gamma) &= \Gamma\left(\gamma | \alpha^\gamma, \beta^\gamma\right), & p(\chi) &= \Gamma\left(\chi | \alpha^\chi, \beta^\chi\right), \end{aligned} \tag{11}$$

where $\mathbf{m}^{\mathbf{h}_1}$ would typically be all zeros. Let $\mathbf{y} \triangleq y_{1:T}$ denote the collection of observations, $\mathbf{z} \triangleq z_{1:T}$ the auxiliary states, $\boldsymbol{\eta} \triangleq [\boldsymbol{\theta}\ \gamma\ \chi]$ the collection of time-invariant parameters, $\mathbf{x} \triangleq \mathbf{x}_{1:T}$ the set of all inputs and outputs, $\mathbf{e} \triangleq e_{1:T}$ the set of all noise instances and $\mathbf{h} \triangleq \mathbf{h}_{1:T+1}$ the set of all noise vectors. We collect the entire states and parameters into $\boldsymbol{\psi} \triangleq [\mathbf{x}, \mathbf{e}, \mathbf{h}, \mathbf{z}, \boldsymbol{\eta}]$ and form our generative model:

$$\begin{aligned} &p(\mathbf{y}, \boldsymbol{\psi}) = p(\boldsymbol{\theta})p(\mathbf{h}_1)p(\gamma)p(\chi) \\ &\prod_{k=1}^{T} p(y_k|z_k, e_k)p(z_k|\boldsymbol{\theta}, \chi, \mathbf{x}_k, \mathbf{h}_k)p(\mathbf{h}_{k+1}|\mathbf{h}_k, e_k)p(e_k|\gamma)\,. \end{aligned} \tag{12}$$

An FFG corresponding to one time slice of the model (12) is given in Figure 1, where the composite node Π represents (9). Note that the definition in (8) manifests itself as a loop in the FFG and will make inference challenging.

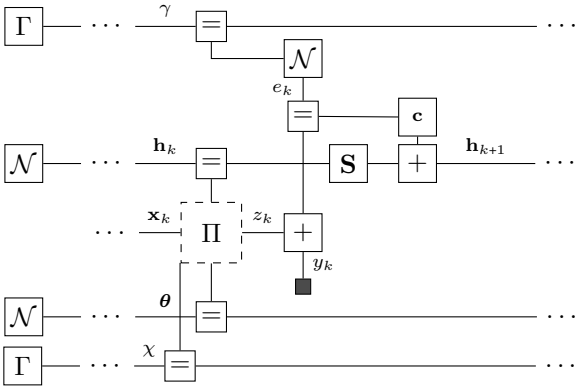

Fig. 1. Forney factor graph of latent noise NARMAX model. Black nodes represent data constraints on the marginals, i.e., observations of inputs and outputs. Terminal nodes (left) represent prior distributions, solid nodes are standard operations (c.f. [8]) and the dotted node marked Π is custom (see Fig. 2).

V. Inference procedure

Having specified a NARMAX model in the form of a joint distribution, Bayesian inference is concerned with obtaining an exact posterior after T observations via Bayes rule:

$$p(\boldsymbol{\psi} \mid \mathbf{y}) = \frac{p(\mathbf{y}, \boldsymbol{\psi})}{p(\mathbf{y})}\,, \tag{13}$$

where the denominator term is the marginalization over all unobserved variables $p(\mathbf{y}) = \int p(\mathbf{y}, \boldsymbol{\psi}) \mathrm{d}\boldsymbol{\psi}$. Unfortunately, we cannot obtain the posterior distribution exactly because the required marginalization has no known analytical solution. Below we describe how we perform approximate inference for this model.

Our treatment is inspired by [3] and follows the derivations of [11]. Variational optimization approximates an intractable posterior with a simpler distribution $q(\boldsymbol{\psi}) \approx p(\boldsymbol{\psi}|\mathbf{y})$, dubbed the *variational posterior*. A variational objective (14), which upper bounds the negative log model evidence, describes the quality of the approximate posterior [12]:

$$F[q, p] = \int q(\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\psi})}{p(\mathbf{y}, \boldsymbol{\psi})} \mathrm{d}\boldsymbol{\psi} \geq -\log p(\mathbf{y})\,. \tag{14}$$

The variational objective may be decomposed into a differential entropy term $H[q]$ and an "average energy" term $U[q, p]$:

$$H[q] = -\int q(\boldsymbol{\psi}) \log q(\boldsymbol{\psi}) \mathrm{d}\boldsymbol{\psi} \tag{15}$$

$$U[q, p] = -\int q(\boldsymbol{\psi}) \log p(\mathbf{y}, \boldsymbol{\psi}) \mathrm{d}\boldsymbol{\psi}, \tag{16}$$

Computing (14) becomes intractable because the entropy (15) term does not factorize according to (12) [12]. To circumvent intractability, we will be minimizing Bethe free energy and introduce constraints on $q(\boldsymbol{\psi})$ [11]. The factorization for the variational posterior is constrained to be structured as follows:

$$q(\boldsymbol{\psi}) = q(\boldsymbol{\eta}) \prod_{k=1}^{T} q(\mathbf{x}_k, \mathbf{h}_k, z_k) q(\mathbf{h}_{k+1}, \mathbf{h}_k, e_k) q(z_k, e_k) \tag{17}$$

$$q(\mathbf{h}_{k+1}, \mathbf{h}_k, e_k) \triangleq p(\mathbf{h}_{k+1}|\mathbf{h}_k, e_k)\, q(\mathbf{h}_k, e_k)\,, \tag{18}$$

where $q(\boldsymbol{\eta}) = q(\gamma)q(\chi)q(\boldsymbol{\theta})$, and (18) is possible due to [11, Thm. 8]. The Bethe free energy $F_B[q,p]$ for the NARMAX model (12) then becomes:

$$\begin{aligned}
F_B[q,p] &= U[q(\boldsymbol{\theta}),p(\boldsymbol{\theta})] + U[q(\chi),p(\chi)] + U[q(\gamma),p(\gamma)] \\
&\quad + U[q(\mathbf{h}_1),p(\mathbf{h}_1)] + \sum_{k=1}^{T} U[q(\gamma)q(e_k),p(e_k|\gamma)] \\
&\quad + \sum_{k=1}^{T} U[q(\mathbf{x}_k,\mathbf{h}_k,z_k)q(\boldsymbol{\theta})q(\chi),p(z_k|\boldsymbol{\theta},\chi,\mathbf{x}_k,\mathbf{h}_k)] \\
&\quad - \sum_{k=1}^{T} H[q(z_k,e_k)] + H[q(\mathbf{h}_k,e_k)] + H[q(\mathbf{x}_k,\mathbf{h}_k,z_k)] \\
&\quad + \sum_{k=1}^{T} H[q(z_k)] + H[q(\mathbf{h}_k)] + H[q(e_k)] + H[q(\mathbf{x}_k)] \\
&\quad - H[q(\boldsymbol{\theta})] - H[q(\chi)] - H[q(\gamma)],. \qquad (19)
\end{aligned}$$

Moreover, we impose marginalization and normalization constraints on the joint factors:

$$\int q(\mathbf{x}_k,\mathbf{h}_k,z_k)\mathrm{d}\backslash\mathbf{h}_k = q(\mathbf{h}_k) \text{ and } \int q(\mathbf{h}_k)\mathrm{d}\mathbf{h}_k = 1\,. \quad (20)$$

Augmenting the Bethe free energy (19) with the marginalization and normalization constraints (20), produces the following Lagrangian:

$$\begin{aligned}
\mathcal{L}[q,\lambda] &= F_B[q,p] \\
&+ \sum_{k=1}^{T} \int \lambda_1(\mathbf{h}_k) \left(q(\mathbf{h}_k) - \int q(\mathbf{h}_k,\mathbf{x}_k,z_k)\mathrm{d}\backslash\mathbf{h}_k \right) \mathrm{d}\mathbf{h}_k \\
&+ \sum_{k=1}^{T} \int \lambda_2(\mathbf{h}_k) \left(q(\mathbf{h}_k) - \int q(\mathbf{h}_k,e_k)\mathrm{d}\backslash\mathbf{h}_k \right) \mathrm{d}\mathbf{h}_k \\
&+ \sum_{k=1}^{T} \int \lambda(\mathbf{x}_k) \left(q(\mathbf{x}_k) - \int q(\mathbf{h}_k,\mathbf{x}_k,z_k)\mathrm{d}\backslash\mathbf{x}_k \right) \mathrm{d}\mathbf{x}_k \\
&+ \sum_{k=1}^{T} \int \lambda_1(z_k) \left(q(z_k) - \int q(\mathbf{h}_k,\mathbf{x}_k,z_k)\mathrm{d}\backslash z_k \right) \mathrm{d}z_k \\
&+ \sum_{k=1}^{T} \int \lambda_2(z_k) \left(q(z_k) - \int q(z_k,e_k)\mathrm{d}e_k \right) \mathrm{d}z_k \\
&+ \sum_{k=1}^{T} \int \lambda_1(e_k) \left(q(e_k) - \int q(z_k,e_k)\mathrm{d}z_k \right) \mathrm{d}e_k \\
&+ \sum_{k=1}^{T} \int \lambda_2(e_k) \left(q(e_k) - \int q(\mathbf{h}_k,e_k)\mathrm{d}\backslash e_k \right) \mathrm{d}e_k \\
&+ \sum_{k=1}^{T} \lambda_k^{\mathbf{z}} \left(\int q(z_k)\mathrm{d}z_k - 1 \right) + \lambda_k^{\mathbf{x}} \left(\int q(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k - 1 \right) \\
&+ \sum_{k=1}^{T} \lambda_k^{\mathbf{h}} \left(\int q(\mathbf{h}_k)\mathrm{d}\mathbf{h}_k - 1 \right) + \lambda^{\boldsymbol{\theta}} \left(\int q(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} - 1 \right) \\
&+ \lambda^{\gamma} \left(\int q(\gamma)\mathrm{d}\gamma - 1 \right) + \lambda^{\chi} \left(\int q(\chi)\mathrm{d}\chi - 1 \right), \qquad (21)
\end{aligned}$$

where λ's denote Lagrangian multipliers and functions. We include a marginalization over the data vector $\mathbf{x}_k$ to account for the fact that in a simulation setting some of the elements of $\mathbf{x}_k$ will not be observed. When $\mathbf{x}_k$ is observed, the posterior is $\delta(\mathbf{x}_k - \hat{\mathbf{x}}_k)$ and marginalization is automatically fulfilled.

Based on the Lagrangian (21) and weak duality [13, Ch 5] we aim to solve the following constraint optimization problem:

$$\min_q F_B[q,p] = \min_q \max_\lambda \mathcal{L}[q,\lambda]\,. \qquad (22)$$

A set of necessary conditions for optimality is given [3] by the following set of conditions on the variational posteriors:

$$\begin{aligned}
&\frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\mathbf{h}_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(z_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(e_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\mathbf{x}_k)} = 0\,, \\
&\frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\boldsymbol{\theta})} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\gamma)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\chi)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\mathbf{h}_k,e_k)} = 0\,, \\
&\frac{\delta\mathcal{L}[q,\lambda]}{\delta q(\mathbf{x}_k,\mathbf{h}_k,z_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta q(z_k,e_k)} = 0\,, \qquad (23)
\end{aligned}$$

where $\delta\mathcal{L}[q,\lambda]/\delta q(\cdot)$ denotes a variational derivative of the Lagrangian. Additionally, the following optimality conditions on the Lagrangian multipliers and functions are necessary:

$$\begin{aligned}
&\frac{\delta\mathcal{L}[q,\lambda]}{\delta\lambda_i(z_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta\lambda(\mathbf{x}_k)} = 0\,,\ \frac{\delta\mathcal{L}[q,\lambda]}{\delta\lambda_i(\mathbf{h}_k)} = 0\,, \frac{\delta\mathcal{L}[q,\lambda]}{\delta\lambda_i(e_k)} = 0\,, \\
&\frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_\chi} = 0\,,\ \frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_\gamma} = 0\,,\ \frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_{\mathbf{z}}} = 0\,, \\
&\frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_{\mathbf{x}}} = 0\,,\ \frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_{\mathbf{h}}} = 0\,,\ \frac{\partial\mathcal{L}[q,\lambda]}{\partial\lambda_{\boldsymbol{\theta}}} = 0\,, \qquad (24)
\end{aligned}$$

where $i = 1, 2$. In order to efficiently solve for the optimality conditions we leverage the fact that most of the computations can be distributed along the model's factorization structure (12). Solutions can then efficiently be computed by message passing on factor graphs [12], [11].

A. Message passing on Forney-style factor graphs

A factor graph is a graphical representation of a factorized function, where nodes represent factors and edges correspond to variables [8]. A *Forney* factor graph (FFG) imposes the constraint that variables may only be attached to two nodes at most; any variable that is an argument for three factors must be attached to a node enforcing equality between them [8]. Figure 1 shows the factor graph corresponding to a single time-step of our generative model and Figure 2 shows the factor graph inside the composite node responsible for the nonlinearity given by (9).

Solutions to the optimality conditions (23) and (24) for a generic factor graph under the specified constraints are derived in [11, Thm 2]. For instance, the update for $\boldsymbol{\theta}$ is:

$$q(\boldsymbol{\theta}) = \exp\left(\mathbb{E}_{\backslash q(\boldsymbol{\theta})}[\log p(\boldsymbol{\Psi})] - \lambda_{\boldsymbol{\theta}} \right)\,, \qquad (25)$$

$$\lambda_{\boldsymbol{\theta}} = \log \int \exp\left(\mathbb{E}_{\backslash q(\boldsymbol{\theta})}[\log p(\boldsymbol{\Psi})] \right) \mathrm{d}\boldsymbol{\theta}\,. \qquad (26)$$

The computation of (25) and (26) localizes further due to factorized model structure as shown in [12], [11]. The results

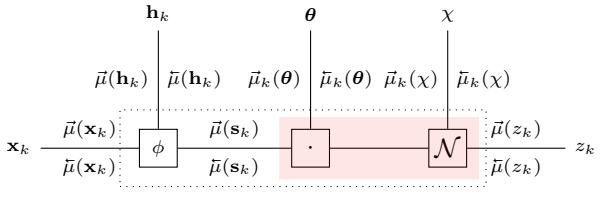

Fig. 2. Internals of the composite factor node denoted with Π from Figure 1, including associated messages. The red shaded box is treated as a composite *soft dot product* node.

of the local computations are called messages. For example, (25) can be written as:

$$q(\boldsymbol{\theta}) = \frac{\vec{\mu}_k(\boldsymbol{\theta})\overleftarrow{\mu}_k(\boldsymbol{\theta})}{\int \vec{\mu}_k(\boldsymbol{\theta})\overleftarrow{\mu}_k(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}\,, \tag{27}$$

where the messages are

$$\overleftarrow{\mu}_k(\boldsymbol{\theta}) = \exp\left(\mathbb{E}_{\setminus q(\boldsymbol{\theta})}\left[\log p(z_k|\boldsymbol{\theta},\chi,\mathbf{x}_k,\mathbf{h}_k)\right]\right) \tag{28}$$

$$\vec{\mu}_k(\boldsymbol{\theta}) = p(\boldsymbol{\theta})\prod_{j\neq k}\exp\left(\mathbb{E}_{\setminus q(\boldsymbol{\theta})}\left[\log p(z_j|\boldsymbol{\theta},\chi,\mathbf{x}_j,\mathbf{h}_j)\right]\right). \tag{29}$$

We put arrows over the messages in (28) and (29) to distinguish between edges. For time-varying parameters, messages are marked with time instance subscripts.

Similarly, we can compute approximate posterior marginals for the time-varying states. For example, the variational posterior over the latent noise vector $\mathbf{h}_k$ is:

$$q(\mathbf{h}_k) = \exp\left(\lambda_1(\mathbf{h}_k) + \lambda_2(\mathbf{h}_k) + \lambda_k^{\mathbf{h}}\right). \tag{30}$$

The Lagrange functions and multipliers are determined as:

$$\lambda_1(\mathbf{h}_k) = \log\overleftarrow{\mu}(\mathbf{h}_k)\,, \quad \lambda_2(\mathbf{h}_k) = \log\vec{\mu}(\mathbf{h}_k)\,, \tag{31}$$

$$\lambda_k^{\mathbf{h}} = -\log\int\exp\left(\lambda_1(\mathbf{h}_k) + \lambda_2(\mathbf{h}_k)\right)\mathrm{d}\mathbf{h}_k\,, \tag{32}$$

with the leftward message defined as

$$\overleftarrow{\mu}(\mathbf{h}_k) \triangleq \int \vec{\mu}(\mathbf{x}_k)\overleftarrow{\mu}(\mathbf{s}_k)\delta(\mathbf{s}_k - \phi([\mathbf{x}_k\ \mathbf{h}_k]))\mathrm{d}\mathbf{s}_k\mathrm{d}\mathbf{x}_k\,, \tag{33}$$

$$\overleftarrow{\mu}(\mathbf{s}_k) \triangleq \int \overleftarrow{\mu}(z_k)\exp\left(\mathbb{E}_{q(\boldsymbol{\theta})q(\chi)}\left[\log p(z_k|\boldsymbol{\theta},\chi,\mathbf{s}_k)\right]\right)\mathrm{d}z_k\,,$$

and the rightward message defined as

$$\vec{\mu}(\mathbf{h}_k) \triangleq \vec{\nu}(\mathbf{h}_k)\overleftarrow{\nu}(\mathbf{h}_k) \tag{34}$$

$$\overleftarrow{\nu}(\mathbf{h}_k) \triangleq \int \overleftarrow{\mu}(e_k)\overleftarrow{\mu}(\mathbf{h}_{k+1})p(\mathbf{h}_{k+1}|\mathbf{h}_k,e_k)\mathrm{d}\mathbf{h}_{k+1}\mathrm{d}e_k$$

$$\vec{\nu}(\mathbf{h}_k) \triangleq \int \vec{\mu}(e_{k\text{-}1})\vec{\mu}(\mathbf{h}_{k\text{-}1})p(\mathbf{h}_k|\mathbf{h}_{k\text{-}1},e_{k\text{-}1})\mathrm{d}\mathbf{h}_{k\text{-}1}\mathrm{d}e_{k\text{-}1}.$$

The distribution $p(z_k \mid \boldsymbol{\theta},\chi,\mathbf{s}_k) = \mathcal{N}(z_k \mid \boldsymbol{\theta}^\top\mathbf{s}_k, \chi^{-1})$ is the composite soft dot product shaded red in Figure 2, dependent on intermediate variable $\mathbf{s}_k = \phi([\mathbf{x}_k\ \mathbf{h}_k])$.

The marginal for $\mathbf{h}_k$ can be computed as the normalized product of messages $\vec{\mu}(\mathbf{h}_k)$ and $\overleftarrow{\mu}(\mathbf{h}_k)$, similar to (30). However, the functional forms of these messages are analytically intractable due to the non-invertible non-linear transformation [5, Ch. 5]. In Section V-B, we detail how we approximate message computations around deterministic non-linear state transitions to compute $q(\mathbf{h}_k)$.

B. Approximating messages with the Unscented Transform

The polynomial ϕ performs a nonlinear transformation on the Gaussian distributed noise instances in $\mathbf{h}_k$. However, random variables are not closed under nonlinear transformation, i.e., the outcome of ϕ is no longer Gaussian distributed [5, Ch. 5]. We thus have no general analytic solution to the integrations required to compute (variational) messages and marginals involving $\boldsymbol{\phi}([\mathbf{x}_k\mathbf{h}_k])$. In this sub-section we discuss Unscented transform based approximation [5], [14].

Consider the non-linear function $\mathbf{s}_k = \boldsymbol{\phi}([\mathbf{x}_k\ \mathbf{h}_k])$ as part of (9) and let $\tilde{\mathbf{x}}_k \triangleq [\mathbf{x}_k\ \mathbf{h}_k]$ be an augmented state with dimensionality M. We are interested in obtaining a Gaussian approximation to $\overleftarrow{\mu}(\tilde{\mathbf{x}}_k)$, which will consequently assist approximating $\overleftarrow{\mu}(\mathbf{h}_k)$. Given the messages,

$$\vec{\mu}(\mathbf{h}_k) = \mathcal{N}(\mathbf{h}_k \mid \vec{\mathbf{m}}_k^{\mathbf{h}}, \vec{\mathbf{P}}_k^{\mathbf{h}}) \tag{35}$$

$$\vec{\mu}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k \mid \vec{\mathbf{m}}_k^{\mathbf{x}}, \vec{\mathbf{P}}_k^{\mathbf{x}}) \tag{36}$$

$$\overleftarrow{\mu}(\mathbf{s}_k) = \mathcal{N}\left(\mathbf{s}_k \mid \overleftarrow{\mathbf{m}}_k^{\mathbf{x}}, \overleftarrow{\mathbf{P}}_k^{\mathbf{x}}\right), \tag{37}$$

we write the augmented message as

$$\vec{\mu}(\tilde{\mathbf{x}}_k) = \mathcal{N}\left(\tilde{\mathbf{x}}_k \,\middle|\, \vec{\mathbf{m}}_k^{\tilde{\mathbf{x}}}, \vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}}\right) \tag{38}$$

$$\vec{\mathbf{m}}_k^{\tilde{\mathbf{x}}} \triangleq \begin{bmatrix}\vec{\mathbf{m}}_k^{\mathbf{x}} \\ \vec{\mathbf{m}}_k^{\mathbf{h}}\end{bmatrix},\ \vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}} \triangleq \begin{bmatrix}\vec{\mathbf{P}}_k^{\mathbf{x}} & \mathbf{0} \\ \mathbf{0} & \vec{\mathbf{P}}_k^{\mathbf{h}}\end{bmatrix}. \tag{39}$$

In the Unscented Transform, the covariance matrix $\vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}}$ is decomposed into its singular values [14]:

$$\vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}} = \vec{\mathbf{U}}_k\vec{\boldsymbol{\Sigma}}_k\vec{\mathbf{V}}_k^\top = \sum_{i=1}^{M}\zeta_k^{(i)}\vec{\mathbf{u}}_k^{(i)}\vec{\mathbf{v}}_k^{(i)\top}. \tag{40}$$

Based on (40), we compute the following weights and sigma points:

$$\tilde{\mathbf{x}}_k^{(\pm i)} = \vec{\mathbf{m}}_k^{\tilde{\mathbf{x}}} \pm \sqrt{(M+\nu)\zeta_k^{(i)}}\vec{\mathbf{u}}_k^{(i)}, \quad i = 1,\ldots,M \tag{41}$$

$$\omega^{(\pm i)} = \frac{1}{2(M+\upsilon)}, i = 1,\ldots,M \tag{42}$$

$$\tilde{\mathbf{x}}_k^{(0)} = \vec{\mathbf{m}}_k^{\tilde{\mathbf{x}}}, \quad \omega^{(0)} = \frac{\upsilon}{\upsilon+M} + (1-\vartheta^2+\epsilon)\,, \tag{43}$$

where $\upsilon = \vartheta^2(M+\kappa) - M$ such that ϑ, ϵ and κ are free parameters. The definition of the weights $\omega^{(i)}$ ensures proper normalization. For the interpretation of these parameters, we refer to [14].

Then we transform the points $\tilde{\mathbf{x}}_k^{(\pm i)}$ through the non-linearity and define the auxiliary transformed points as $\boldsymbol{\xi}_k^{(i)} = \boldsymbol{\phi}\left(\tilde{\mathbf{x}}_k^{(i)}\right)$. Then the first two moments of $\vec{\mu}(\mathbf{s}_k) = \mathcal{N}(\mathbf{s}_k|\vec{\mathbf{m}}_k^{\mathbf{s}}, \vec{\mathbf{P}}_k^{\mathbf{s}})$ and the cross-covariance between $\tilde{\mathbf{x}}_k$ and $\mathbf{s}_k$ are determined as

$$\vec{\mathbf{m}}_k^{\mathbf{s}} = \sum_{i=-M}^{M}\omega^{(i)}\boldsymbol{\xi}_k^{(i)} \tag{44}$$

$$\vec{\mathbf{P}}_k^{\mathbf{s}} = \sum_{i=-M}^{M}\omega^{(i)}\left(\boldsymbol{\xi}_k^{(i)} - \vec{\mathbf{m}}_k^{\mathbf{s}}\right)\left(\boldsymbol{\xi}_k^{(i)} - \vec{\mathbf{m}}_k^{\mathbf{s}}\right)^\top \tag{45}$$

$$\mathbf{C}_k = \sum_{i=-M}^{M}\omega^{(i)}\left(\tilde{\mathbf{x}}_k^{(i)} - \vec{m}_k^{\tilde{\mathbf{x}}}\right)\left(\boldsymbol{\xi}_k^{(i)} - \vec{\mathbf{m}}_k^{\mathbf{s}}\right)^\top. \tag{46}$$

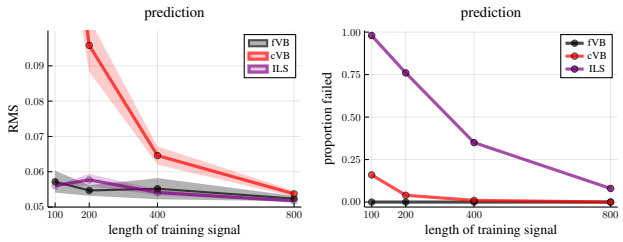

Fig. 3. 1-step ahead prediction result. (Left) Average RMS (standard errors as ribbons) by the length of training signal for the 1-step ahead prediction error in case of a 0.05 noise std. dev. (Right) The proportion of experiments failed due to diverging parameter estimates.

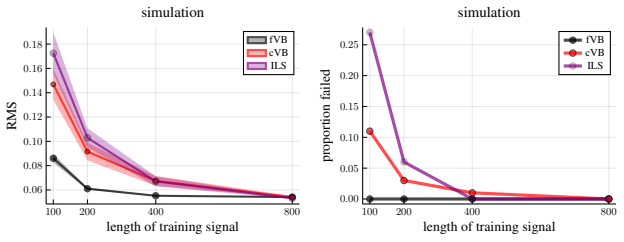

Fig. 4. Simulation results. (Left) Average RMS (standard errors as ribbons) by the length of training signal for the simulation error in case of a 0.05 noise std. dev. (Right) The proportion of experiments failed due to diverging parameter estimates.

Using the predicted statistics (44),(45) and (46) we can compute the approximate mean and covariance of the backwards message $\overleftarrow{\mu}(\tilde{\mathbf{x}}_k)$ according to [5, Page 151, eq.9.18] as

$$\overleftarrow{\mathbf{m}}_k^{\tilde{\mathbf{x}}} = \vec{\mathbf{m}}_k^{\tilde{\mathbf{x}}} + \mathbf{G}_k \left(\overleftarrow{\mathbf{m}}_k^{\mathbf{s}} - \vec{\mathbf{m}}_k^{\mathbf{s}} \right), \quad \mathbf{G}_k = \mathbf{C}_k \left(\vec{\mathbf{P}}_k^{\mathbf{s}} \right)^{-1} \tag{47}$$

$$\overleftarrow{\mathbf{P}}_k^{\tilde{\mathbf{x}}} = \vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}} + \mathbf{G}_k \left(\overleftarrow{\mathbf{P}}_k^{\mathbf{s}} - \vec{\mathbf{P}}_k^{\mathbf{s}} \right) \mathbf{G}_k^{\top} . \tag{48}$$

Using (47) and (48) we can form the Gaussian approximation $\overleftarrow{\mu}(\tilde{\mathbf{x}}_k) \approx \mathcal{N}\left(\tilde{\mathbf{x}}_k \mid \overleftarrow{\mathbf{m}}_k^{\tilde{\mathbf{x}}}, \overleftarrow{\mathbf{P}}_k^{\tilde{\mathbf{x}}}\right)$. Then, we can determine the approximate marginal $q(\tilde{\mathbf{x}}_k)$ as

$$q(\tilde{\mathbf{x}}_k) \propto \vec{\mu}(\tilde{\mathbf{x}}_k) \overleftarrow{\mu}(\tilde{\mathbf{x}}_k) = \mathcal{N}\left(\tilde{\mathbf{x}}_k \mid \mathbf{m}_k^{\tilde{\mathbf{x}}}, \mathbf{P}_k^{\tilde{\mathbf{x}}}\right) , \tag{49}$$

with parameters:

$$\mathbf{P}_k^{\tilde{\mathbf{x}}} = \left(\left(\vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}} \right)^{-1} + \left(\overleftarrow{\mathbf{P}}_k^{\tilde{\mathbf{x}}} \right)^{-1} \right)^{-1} \tag{50}$$

$$\mathbf{m}_k^{\tilde{\mathbf{x}}} = \mathbf{P}_k^{\tilde{x}} \left(\left(\vec{\mathbf{P}}_k^{\tilde{\mathbf{x}}} \right)^{-1} \vec{\mathbf{m}}_k^{\tilde{x}} + \left(\overleftarrow{\mathbf{P}}_k^{\tilde{\mathbf{x}}} \right)^{-1} \overleftarrow{\mathbf{m}}_k^{\tilde{x}} \right) . \tag{51}$$

Since $\tilde{\mathbf{x}}_k$ is a concatenation of $\mathbf{h}_k$ and $\mathbf{x}_k$, we can write the following block structure for the moments of $\tilde{\mathbf{x}}_k$

$$\mathbf{m}_k^{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{m}_k^{\mathbf{x}} \\ \mathbf{m}_k^{\mathbf{h}} \end{bmatrix} , \quad \mathbf{P}_k^{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{P}_k^{\mathbf{x}} & \times \\ \times & \mathbf{P}_k^{\mathbf{h}} \end{bmatrix} , \tag{52}$$

such that we can write

$$q(\mathbf{h}_k) \approx \mathcal{N}(\mathbf{h}_k \mid \mathbf{m}_k^{\mathbf{h}}, \mathbf{P}_k^{\mathbf{h}}) . \tag{53}$$

The outgoing message towards z_k relies on the dot product between Gaussian distributed coefficients $\boldsymbol{\theta}$ and the approximation to the polynomial $\mathbf{s}_k$, which is also Gaussian distributed. Unfortunately, products of the Gaussian variables are no longer Gaussian distributed. We employ the rules tabulated in [Table 1][15] to approximate the moments of the outgoing message $\vec{\mu}(z_k) \propto \mathcal{N}(z_k \mid \vec{m}_k^z, \vec{v}_k^z)$.

VI. Prediction and Simulation

We form a posterior predictive distribution for future output using the messages available at time $t > k$:

$$p(y_t \mid \mathbf{x}_t) \approx \int \vec{\mu}(z_t) \vec{\mu}(e_t) p(y_t | z_t, e_t) \, \mathrm{d}z_t \mathrm{d}e_t \tag{54}$$

$$= \int \vec{\mu}(z_t) \mathcal{N}\left(y_t \,\middle|\, z_t, \frac{\beta^\gamma}{\alpha^\gamma} \right) \mathrm{d}z_t \tag{55}$$

$$\propto \mathcal{N}\left(y_t \,\middle|\, \vec{m}_t^z, \frac{\beta^\gamma}{\alpha^\gamma} + \vec{v}_t^z \right) . \tag{56}$$

Above, the approximately equal is the result of using the variational message $\vec{\mu}(e_t) \propto \exp(\mathbb{E}_{q(\gamma)}[\log p(e_t|\gamma)])$. In 1-step ahead prediction, the elements of $\mathbf{h}_t$ will have been shaped by both the prior and the likelihood $\mathbf{x}_t$, and will have a non-zero mean. In a simulation, $\mathbf{h}_t$ will only be shaped by the priors and will be zero-mean (this is reminiscent of zero-padding [16]).

During simulation, the elements of $\mathbf{x}_t$ will be shaped by previous posterior predictive distributions. The approximation from Section V-B is already equipped to handle the replacement of observed outputs with random variables.

VII. Experiments

We evaluate the proposed model and the inference algorithm in a verification experiment[1] involving 100 Monte Carlo samples of a polynomial NARMAX system (see Figure 6 for an example). We chose a polynomial of degree 3 with no DC component and no noise cross terms, producing a dimensionality of $D = 22$. The input is an odd random phase multi-sine signal with a minimum excited frequency of 0 and a maximum of 100, generated under a sampling frequency of 1 kHz. The coefficients are pseudo-randomly generated: those responsible for linear terms of the polynomial inherit their values from a Butterworth filter, and the others are generated according to a uniform distribution on $\mathcal{U}(-0.01, 0.01)$. The system's noise standard deviation was set to 0.05.

We compared the proposed *fully Variational Bayes* (fVB) estimator with a collapsed (i.e., noise instances are computed as differences between observations and MAP estimates of the posterior predictive distribution [2]) variational Bayes (cVB) and a classic Iterative Least-Squares (ILS) estimator [1, Section 3.6] estimator. We varied the length of the training signal $L \in [100, 200, 400, 800]$ and computed the Root Mean Square (RMS) over a test signal of length 1000. Figure 4 demonstrates the simulation errors of the estimators as a function of training samples. For small training sets (< 500 samples), fVB outperforms both the cVB and the ILS estimators. For the training length of 1000 all three algorithms perform equally well, yielding 0.0528, 0.0532, 0.0523 for fVB, cVB and ILS, respectively. For the simulation experiment, fVB provides the lowest number of failed runs, that is, 5 against 12 (cVB) and 20 (ILS). A run is considered 'failed' when the identified model produces

[1]Code at: `https://github.com/biaslab/CDC-2022`.

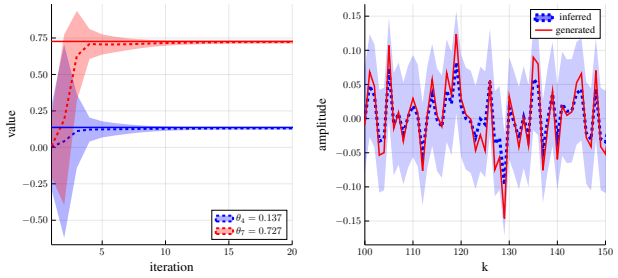

Fig. 5. (Left) Inference results for coefficients $q(\boldsymbol{\theta}_k \mid \boldsymbol{y})$ (only 2 out of 22 coefficients are shown to avoid clutter). The solid lines correspond to the true underlying values. The dashed lines correspond to the expected values of the posterior estimates, with the shaded regions corresponding to the inferred standard deviation of the approximate posterior distributions. (Right) Inferred approximate posterior distributions of the noise variables $q(e_k \mid \boldsymbol{y})$. The expected values track the true noise instances well. Note also the differences in the posterior variance over time.

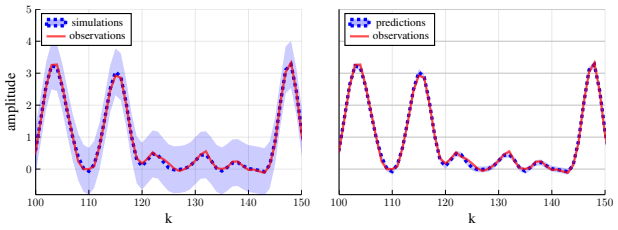

Fig. 6. Example simulation (left) and 1-step ahead prediction (right) for $L = 50$. The red solid line corresponds to outputs and the blue ribbon corresponds to the posterior predictive distribution ($\pm$ standard deviation), with the dashed line being the most probable value.

diverging predictions on the test set. Figure 3 shows the 1-step ahead prediction errors as a function of training samples. As can be seen, for $L = 1000$, all estimators provide similar RMS values: 0.0520 (fVB), 0.0528 (cVB), and 0.0517 (ILS). However, for $L < 1000$ fVB outperforms both the cVB and the ILS estimators. For small training sizes ($L < 1000$), ILS had the highest number of failed runs. Remarkably, fVB had 0 failed runs throughout all 400 runs. We highlight the inference results of the proposed algorithms in Figures 5-6.

VIII. Discussion

The factor graph of our proposed model contains loops, which may cause convergence issues in message passing procedures [17]. However, some results suggest loopy belief propagation with variational messages converges to the fixed points of the Bethe free energy [18].

In our experimental setup, the proposed estimator *fVB* outperforms cVB and ILS in terms of RMS. However, RMS as a proxy for model comparison is undesirable as it ignores important properties of Bayesian estimators, such as uncertainty estimation. We used RMS solely to comply with the absence of a posterior predictive distribution in ILS.

An important future direction is to incorporate richer classes of function approximators, such as Gaussian processes or neural networks. Recent work has performed Bayesian inference for dynamical systems using long short-term memory recurrent neural networks [19].

IX. Conclusion

This paper introduced a Bayesian NARMAX identification approach where model parameters and individual noise instances are estimated simultaneously. Inference was phrased as optimizing a variational posterior through message passing on a factor graph. It was shown to outperform a classical NARMAX identification approach, especially when sample size is small.

X. Acknowledgements

Partly funded by research program ZERO (no. P15-06), Netherlands Organisation for Scientific Research (NWO).

References

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

[2] W. M. Kouw, A. Podusenko, M. T. Koudahl, and M. Schoukens, "Variational message passing for online polynomial NARMAX identification," in *American Control Conference*, 2022, pp. 1–6.

[3] K. Fujimoto and Y. Takaki, "On system identification for ARMAX models based on the variational Bayesian method," in *IEEE Conference on Decision and Control*, 2016, pp. 1217–1222.

[4] J. Dahlin, A. Wills, and B. Ninness, "Sparse Bayesian ARX models with flexible noise distributions," *IFAC-PapersOnLine*, vol. 51, no. 15, pp. 25–30, 2018.

[5] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[6] T. Baldacchino, S. R. Anderson, and V. Kadirkamanathan, "Computational system identification for Bayesian NARMAX modelling," *Automatica*, vol. 49, no. 9, pp. 2641–2651, 2013.

[7] J. Courts, J. Hendriks, A. Wills, T. B. Schön, and B. Ninness, "Variational state and parameter estimation," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 732–737, 2021.

[8] S. Korl, "A factor graph approach to signal modelling, system identification and filtering," Ph.D. dissertation, ETH Zurich, 2005.

[9] J. Dauwels, "On variational message passing on factor graphs," in *IEEE International Symposium on Information Theory*, 2007, pp. 2546–2550.

[10] D. Bagaev and B. de Vries, "Reactive message passing for scalable Bayesian inference," *arXiv:2112.13251*, 2021.

[11] İ. Şenöz, T. van de Laar, D. Bagaev, and B. de Vries, "Variational message passing and local constraint manipulation in factor graphs," *Entropy*, vol. 23, no. 7, p. 807, 2021.

[12] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] F. Gustafsson and G. Hendeby, "Some relations between Extended and Unscented Kalman filters," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 545–555, 2012.

[15] A. Podusenko, W. M. Kouw, and B. de Vries, "Message passing-based inference for time-varying autoregressive models," *Entropy*, vol. 23, no. 6, p. 683, 2021.

[16] D. Khandelwal, M. Schoukens, and R. Tóth, "On the simulation of polynomial NARMAX models," in *IEEE Conference on Decision and Control*, 2018, pp. 1445–1450.

[17] A. T. Ihler, J. W. Fisher III, A. S. Willsky, and D. M. Chickering, "Loopy belief propagation: convergence and effects of message errors." *Journal of Machine Learning Research*, vol. 6, no. 5, 2005.

[18] T. Heskes, "Stable fixed points of loopy belief propagation are local minima of the Bethe free energy," *Advances in Neural Information Processing Systems*, vol. 15, 2002.

[19] A. Brusaferri, M. Matteucci, P. Portolani, and S. Spinelli, "Nonlinear system identification using a recurrent network in a Bayesian framework," in *IEEE International Conference on Industrial Informatics*, vol. 1, 2019, pp. 319–324.