Non-linear State-space Model Identification from Video Data using Deep Encoders

Citation for published version (APA):

Document license: CC BY

Document status and date:
Published: 01/01/2021

Document Version:
Author’s version before peer-review

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Non-linear State-space Model Identification from Video Data using Deep Encoders

Gerben I. Beintema ∗ Roland Toth ∗∗ Maarten Schoukens ∗

∗ Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB, Eindhoven, The Netherlands (e-mails: g.i.beitema@tue.nl, r.toth@tue.nl, m.schoukens@tue.nl).

∂ Systems and Control Laboratory, Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary.

Abstract: Identifying systems with high-dimensional inputs and outputs, such as systems measured by video streams, is a challenging problem with numerous applications in robotics, autonomous vehicles, and medical imaging. In this paper, we propose a novel non-linear state-space identification method starting from high-dimensional input and output data. Multiple computational and conceptual advances are combined to handle the high-dimensional nature of the data. An encoder function, represented by a neural network, is introduced to learn a reconstructability map to estimate the model states from past inputs and outputs. This encoder function is jointly learned with the dynamics. Furthermore, multiple computational improvements, such as an improved reformulation of multiple shooting and batch optimization, are proposed to keep the computational time under control when dealing with high-dimensional and large datasets. We apply the proposed method to a video stream of a simulated environment of a controllable ball in a unit box. The simulation study shows low simulation error with excellent long term prediction for the obtained model using the proposed method.

Keywords: Non-linear State-Space Modelling, Deep Learning, Pixels, Multiple Shooting.

1. INTRODUCTION

Systems with high dimensional inputs and outputs (i.e., large-scale systems) are ever more prevalent due to the increased presence of, for instance, high-resolution video cameras, PDE simulations, system networks, and medical imaging devices. Hence, the identification of flexible models and methods for modelling nonlinear large-scale systems is of the essence. However, currently, this is a challenging task due to the curse of dimensionality and the difficulty of modelling nonlinearities that are encountered in these systems (Moerland et al., 2020). There is extensive literature available for linear state-space model identification for large-scale systems such as subspace methods (Van Overschee and De Moor, 2012), expectation-maximization (Gibson and Ninness, 2005), and PCA or CCA (Katayama, 2006). However, non-linear state-space identification for large-scale systems is currently an open problem.

Recent results for non-linear state-space identification present considerable advances in, polynomial state-space models (Decuyper et al., 2020) and artificial neural networks based state-space models (Schoukens and Toth, 2020; Masti and Bemporad, 2018). Furthermore, parameter estimation methods for non-linear state-space models have improved considerably by the introduction of the multiple shooting method with considerable theoretical (Ribeiro et al., 2019) and practical results (Decuyper et al., 2020). These models and estimation methods have yet to be analysed and developed for large-scale systems. One successful approach to identify non-linear large-scale systems combines a non-linear autoencoder for dimension reduction with a multiple input multiple output (MIMO) NARX model (Wahlström et al., 2015b) (this approach will be referred to as “IO autoencoder” within this paper). The IO autoencoder approach outperforms linear identification methods and allows for model predictive control (Wahlström et al., 2015a). However, a MIMO NARX model is considerably more difficult to interpret and perform controller design with than non-linear state-space models. The complexity of a MIMO NARX model also rapidly increases for growing dynamical complexity. Furthermore, the NARX model structure often degrades in performance when used for simulation.

The aim of this paper is to develop an encoder-informed non-linear state-space identification approach that can efficiently process high-dimensional input-output data. To this end this paper combines \( i) \) non-linear state-space models parameterized as artificial neural networks, \( ii) \) a non-linear encoder together with \( iii) \) an improved formulation of the multiple shooting method utilizing batch optimization. Here the non-linear encoder enables the identification of large-scale systems. The proposed method only requires a single loss function, obtains state-of-the-art results using randomly initialized model parameters and allows for simulation error minimization.

The paper is structured as follows: Section 2 provides an overview of the proposed method, Section 3 shows...

1 Code available at https://github.com/GerbenBeintema/SS-encoder-video
We aim to estimate the following discrete-time state-space model:

\[ \hat{x}_{t+1} = f_\theta(\hat{x}_t, u_t), \quad (1a) \]
\[ \tilde{y}_t = h_\theta(\hat{x}_t), \quad (1b) \]

with \( t \in \mathbb{Z} \) the time index, \( \hat{y}_t \in \mathbb{R}^{nx \times ny} \) the model output, \( \tilde{y}_t \) the system output, \( \hat{x}_t \in \mathbb{R}^n \) the internal model state, \( u_t \in \mathbb{R}^m \) the input, \( \theta \) the model parameters and \( f_\theta \) and \( h_\theta \) the state and output function. Artificial Neural Networks (ANN) are used to represent \( f_\theta \) and \( h_\theta \) as they have excellent approximation properties for high-dimensional functions (Barron, 1993). We assume that the measured data is generated by a system contained within this model class: \( y_t = h_\theta(x_t, u_t) + v_t \) and \( x_{t+1} = f_\theta(x_t, u_t) \), where a possibly coloured, additive zero-mean finite-variance noise source \( v_t \in \mathbb{R}^n \) is assumed to be present at the system output.

### 2.2 Parameter estimation

Most commonly, non-linear state-space models with an OE noise structure are estimated by minimizing the simulation loss (i.e. \( V_{\text{sim}}(\theta) \sim \sum_t \|h_\theta(x_t) - y_t\|^2 \)), however, the computational cost scales linearly with the number of samples \( O(N_{\text{samples}}) \).

To improve the scalability of the proposed method with the length of the dataset, the proposed loss function is constructed by summing over \( N \) sub-sections with starting indices \( t_i \) and length \( T + k_0 + 1 \), similar to the multiple shooting method (Ribeiro et al., 2019), as:

\[ V_{\text{encoder}}(\theta) = \frac{1}{2N(T+1)} \sum_{i=1}^N \sum_{k=k_0}^{T+k_0} \| \tilde{y}_{t_i \rightarrow t_i+k} - y_{t_i+k} \|^2, \quad (2a) \]

\[ \tilde{y}_{t_i \rightarrow t_i+k} := h_\theta(\hat{x}_{t_i \rightarrow t_i+k}), \quad (2b) \]
\[ \hat{x}_{t_i \rightarrow t_i+k+1} := f_\theta(\hat{x}_{t_i \rightarrow t_i+k}, u_{t_i+k}), \quad (2c) \]
\[ \hat{x}_{t_i \rightarrow t_i+k} := e_\theta(y_{t_i-n_i\rightarrow t_i-k-1} - u_{t_i-n_i\rightarrow t_i-k-1}), \quad (2d) \]

where \( x_{t_i \rightarrow t_i+k} \) indicates \( k \) recursive uses of \( f_\theta \) to calculate the state as:

\[ \hat{x}_{t_i \rightarrow t_i+k} = f_\theta(f_\theta(...f_\theta(\hat{x}_{t_i \rightarrow t_i}, u_{t_i}), ..., u_{t_i+k-2}), u_{t_i+k-1}). \]

The initial state \( \hat{x}_{t_i \rightarrow t_i} \) is given by an encoder function \( e_\theta \) based on the previous input and output samples \( u_{t_i-n_i\rightarrow t_i-1} \in \mathbb{R}^{n_i} \) and \( y_{t_i-n_i\rightarrow t_i-1} \in \mathbb{R}^{nx \times ny} \). We call the proposed method the state-space encoder method. It is graphically presented in Figure 1. Just like the state and output functions \( f_\theta \) and \( h_\theta \), the encoder function \( e_\theta \) is represented as an ANN to ensure excellent approximation properties for when dealing with high-dimensional data (Barron, 1993).

The proposed method resolves some of the shortcomings of the parametric start method (Decuyper et al., 2020) (i.e. \( \hat{x}_{t_i \rightarrow t_i} \) is introduced as a parameter of the model) for it has a fixed model complexity whereas the parametric start scales linearly with the number of sections. Moreover, the encoder can generalize to unseen data sets, jump-starting simulations with the correct internal state and aid in control.

Furthermore, due to the independence of the loss function on each section, the proposed method allows for i) computational speedup by utilizing modern parallelization methods and ii) the utilization of batch optimization methods. The batch formulation of the state-space encoder method is obtained by summing not over all sections, but only a subset \( B \) as:

\[ V_{\text{batch}}(\theta) = \frac{1}{2N_{\text{batch}}(T+1)} \sum_{i \in B} \sum_{k=k_0}^{T+k_0} \| \tilde{y}_{t_i \rightarrow t_i+k} - y_{t_i+k} \|^2, \quad (3a) \]

\[ B \subset \{1, 2, ..., N\}. \quad (3b) \]

This reformulation can utilize modern powerful batch optimization algorithms developed by the machine learning community (e.g. Adam (Kingma and Ba, 2014)). Furthermore, utilizing the batch formulation only requires the data to be partially loaded which can be necessary for large data sets of large-scale systems where memory constraints play an essential role.
To calculate the pixel intensity at a position for the actuation. using the RK4 scheme and the zero-order hold condition time by integrating the system with the time step \( \Delta p \)

The following non-linear equation \( \beta \) external forces bounded by \( p \) where \((x, y)\) are sampled from a uniform distribution \( U \). We perform three separate experiments where the inputs are kept with the constraints. The three experiments are i) a noisy training set of size 30,000, ii) a noisy validation set of size 5000 and iii) a noiseless test set of size 5000 frames. Multiple instances of the same training and validation data are created with different amplitudes of added Gaussian noise as \( v \sim N(0, \sigma_v) \) over each image where \( a \) is the noise to signal ratio and \( \sigma_v = 0.204 \) the standard deviation of the test set.

3. NUMERICAL EXPERIMENTS

3.1 Simulated Environment

As an example of a large-scale system, we consider a simulated environment with a camera of 25 by 25 pixels pointed at a ball which can move within a unit box graphically shown in Figure 2. This environment can be interpreted physically as a ball in a concave square cup pointed at a ball which can move within a unit box simulated environment with a camera of 25 by 25 pixels. As an example of a large-scale system, we consider a

3.2 State-space encoder approach

We implement the state-space encoder as follows; we utilize the same feedforward neural network structure for \( e_\theta, f_\theta \) and \( h_\theta \) consisting of a two hidden layer neural network, 64 nodes per layer, tanh activation and a linear bypass. The parameters are initialized by sampling the uniform distribution \( U(\sqrt{k}, \sqrt{k}) \) with \( k = 1/\sqrt{n_{in}} \) where \( n_{in} \) is the number of inputs to the layer. The images are flattened into 25·25 = 625 vectors for both the input of the encoder and the output of \( h_\theta \).

We choose the following state-space encoder hyperparameters of \( n_x = 6, k_0 = 0, T = 50, n_u = n_b = 5 \). The state dimension \( n_s \) is chosen deliberately a bit larger than the minimal state dimension of 4 for we noticed that the optimization is significantly faster (by approximately 25\%) and more consistent. We suspect the mechanism behind this is similar to immersion which reduces the complexity of the required functions by increasing the state dimension (Ohtsuka, 2005). Furthermore, the multiple shooting starting indices \( \ell_i \) are allowed to be every possible value within the range of the training set. For training, the Adam batch optimization method (Kingma and Ba, 2014) is used with the default learning rate of \( \alpha = 10^{-4} \) and a batch size of 256. After each epoch, the simulation error is calculated on the validation set and the current model is saved if a new lowest simulation loss has been achieved. Furthermore, we normalize both the inputs and outputs.
IO autoencoder

40
14.04% 9.40%
4000
20
10
7.61% 7.07%
2000
3000
12.20% 7.90%
50
8.56% 7.55%
1000
5000
70
60
80
n
-step NRMS error
RMS

IO autoencoder

Wahlström et al., 2015b

Table 1. The NRMS simulation error on the noiseless test set for different levels of noisy training data.

Noise Level $a$ | IO autoencoder $b$ | State-space Encoder

<table>
<thead>
<tr>
<th>Noise Level $a$</th>
<th>0%</th>
<th>5%</th>
<th>20%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRMS $b$</td>
<td>7.35%</td>
<td>7.61%</td>
<td>8.56%</td>
<td>12.20%</td>
<td>14.04%</td>
</tr>
<tr>
<td>NRMS scaled $b$</td>
<td>6.57%</td>
<td>7.07%</td>
<td>7.55%</td>
<td>7.90%</td>
<td>9.40%</td>
</tr>
</tbody>
</table>

by subtracting the mean and dividing by the standard deviation, this improves performance and training time.

The obtained results of the method proposed in this paper are compared with a reproduction of the IO autoencoder (Wahlström et al., 2015b) approach which is implemented utilizing the same neural network structure and as similar as possible hyperparameters (i.e. $n_z = n_x = 6$, $n = n_a = n_b = 5$).

3.3 Results

We applied the state-space encoder method on the ball-in-box simulated camera system as mentioned in the previous section. The performance of the state space encoder can be compared visually by observing the output frames of a simulation run on the test set as seen in Figure 3 for the zero noise case. The difference is minimal for most instances of time which suggest that our proposed method can accurately find a non-linear state-space model of order 6 for this system. Moreover, For the case with additive measurement noise the proposed method is still able to successfully model the system as shown in Figure 4.

Quantitatively the proposed method has an NRMS simulation error lower than the IO autoencoder method as can be seen in Table 1 (the NRMS error is obtained by dividing the RMS error by the standard deviation of the test set $\sigma_y = 0.204$). In the same table, it can be seen that even relatively high values of noise $a$ only marginally decrease the performance of the proposed method whereas the IO autoencoder degrades significantly faster. We suspect that this difference is due to the OE noise structure present in the considered system.

However, looking closely at the simulations on the test set shows instances of time where the error becomes relatively high. This can also be observed in the root mean square error of a single output frame over time as in Figure 5 where there are spikes of high simulation error error go up to ten times the average. After close inspection, we found that these spikes appeared when the model was used in regions where there was little to no training data available and hence requiring extrapolation. This can be resolved by having larger training sets or using input design methods which aim to fill the state-space more efficiently (Heinz and Nelke, 2017).

Lastly, to measure the effectiveness of the encoder and validate the chosen hyperparameters of the encoder method we use the $n$-step NRMS. This is the expected normalized $n$-steps ahead prediction error after the encoder presumed state estimate. More formally:

$$\text{NRMS}_n = \sqrt{\sum_{i=1}^{M} \frac{(\hat{y}_{t \rightarrow t+n} - y_{t+n})^2}{\sigma_y}}.$$  (6)

This quantity is shown in Figure 6 where the average is taken over all the possible starting indices $t_i$ of the test set. For the state-space encoder method, the $n$-step NRMS is almost a straight line which indicates that the encoder is able to estimate the initial state accurately with these hyperparameters, whereas, the IO autoencoder degrades slightly in performance for increasing $n$ which is typical for the NARX model structure as it minimizes the one step ahead prediction error.

4. DISCUSSION

This paper proposed a novel method to effectively identify non-linear state-space models for large-scale systems. The proposed method utilizes a state-space encoder, a reformulation of the multiple shooting method which allows for the batch optimization to be used. This results in an optimization approach that scales well for increasing input-output dimensionality and data set length. The method was able to identify with high accuracy a numerical environment using a video stream of a ball with non-linear dynamics while being minimally affected by increased levels of noise. This was achieved without explicitly modelling the dynamics of the ball and purely learning from pixels and force inputs. Applications of this method to physical systems and a detailed theoretical analysis are subject to future work.
REFERENCES


