

# Computing Semilinear Sparse Models for Approximately Eventually Periodic Signals

Fredy Vides \*

\* *Scientific Computing Innovation Center, Universidad Nacional Autónoma de Honduras, Tegucigalpa, Honduras, (e-mail: fredy.vides@unah.edu.hn)*

**Abstract:** Some elements of the theory and algorithmics corresponding to the computation of semilinear sparse models for discrete-time signals are presented. In this study, we will focus on approximately eventually periodic discrete-time signals, that is, signals that can exhibit an aperiodic behavior for an initial amount of time, and then become approximately periodic afterwards. The semilinear models considered in this study are obtained by combining sparse representation methods, linear autoregressive models and GRU neural network models, initially fitting each block model independently using some reference data corresponding to some signal under consideration, and then fitting some mixing parameters that are used to obtain a signal model consisting of a linear combination of the previously fitted blocks using the aforementioned reference data, computing sparse representations of some of the matrix parameters of the resulting model along the process. Some prototypical computational implementations are presented as well.

**Keywords:** Autoregressive models, neural-network models, parameter identification, least-squares approximation, time-series analysis.

## 1. INTRODUCTION

In this document, some elements of the theory and algorithmics corresponding to the computation of semilinear sparse models for discrete-time signals are presented. The study reported in this document is focused on approximately eventually periodic discrete-time signals, that is, signals that can exhibit an aperiodic behavior for an initial amount of time, and then become approximately periodic afterwards.

The main contribution of the work reported in this document is the application of a *colaborative scheme* involving sparse representation methods, linear autoregressive models and GRU neural network models, where each block model is first fitted independently using some reference data corresponding to some given signal. Subsequently, some mixing parameters are fitted, in order to obtain a signal model consisting of a linear combination of the previously fitted blocks, using the aforementioned reference data to fit the mixing coefficients. Along the process, some of the matrices of parameters of the resulting model are fitted using sparse representation methods. Some theoretical aspects of the aforementioned process are described in §3. As a byproduct of the work reported in this document, a toolset of Python programs for semilinear sparse signal model computation based on the ideas presented in §3 and §4 has been developed, and is available in Vides (2021a).

The applications of the sparse signal model identification technology developed as part of the work reported in this document, range from numerical simulation for predictive maintenance of industrial equipment and structures, to geological data analysis.

A prototypical algorithm for the computation of sparse semilinear autoregressors based on the ideas presented in §3, is presented in §4. Some illustrative computational implementations of the prototypical algorithm presented in §4 are documented in §5.

## 2. PRELIMINARIES AND NOTATION

Given  $\delta > 0$ , let us consider the function defined by the expression

$$H_\delta(x) = \begin{cases} 1, & x > \delta \\ 0, & x \leq \delta \end{cases}.$$

Given a matrix  $A \in \mathbb{C}^{m \times n}$  with singular values denoted by the expressions  $s_j(A)$  for  $j = 1, \dots, \min\{m, n\}$ . We will write  $\text{rk}_\delta(A)$  to denote the number

$$\text{rk}_\delta(A) = \sum_{j=1}^{\min\{m, n\}} H_\delta(s_j(A)).$$

Given a time series  $\Sigma = \{x_t\}_{t \geq 1} \subset \mathbb{C}$ , a positive integer  $L$  and any  $t \geq L$ , we will write  $\mathbf{x}_L(t)$  to denote the vector

$$\mathbf{x}_L(t) = [x_{t-L+1} \ x_{t-L+2} \ \cdots \ x_{t-1} \ x_t]^\top \in \mathbb{C}^L.$$

Given an ordered sample  $\Sigma_N = \{x_t\}_{t=1}^N \subset \Sigma$  from a time series  $\Sigma = \{x_t\}_{t \geq 1}$ , we will write  $\mathcal{H}_L(\Sigma_N)$  to denote the Hankel type trajectory matrix corresponding to  $\Sigma_N$ , that is determined by the following expression.

$$\mathcal{H}_L(\Sigma_N) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{N-L+1} \\ x_2 & x_3 & x_4 & \cdots & x_{N-L+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_N \end{bmatrix}$$

We will write  $I_n$  to denote the identity matrix in  $\mathbb{C}^{n \times n}$ , and we will write  $\hat{e}_{j,n}$  to denote the matrices in  $\mathbb{C}^{n \times 1}$  representing the canonical basis of  $\mathbb{C}^n$  (each  $\hat{e}_{j,n}$  equals the  $j$ -column of  $I_n$ ).

We will write  $\mathbf{S}^1$  to denote the set  $\{z \in \mathbb{C} : |z| = 1\}$ . Given any matrix  $X \in \mathbb{C}^{m \times n}$ , we will write  $X^*$  to denote the conjugate transpose  $\bar{X}^\top \in \mathbb{C}^{n \times m}$  of  $X$ . A matrix  $P \in \mathbb{C}^{n \times n}$  will be called an orthogonal projector whenever  $P^2 = P = P^*$ . Given any matrix  $A \in \mathbb{C}^{n \times n}$ , we will write  $\sigma(A)$  to denote the spectrum of  $A$ , that is, the set of eigenvalues of  $A$ .

### 3. SEMILINEAR MODELING OF APPROXIMATELY EVENTUALLY PERIODIC SIGNALS

A discrete-time signal represented by a times series  $\Sigma = \{x_t\}_{t \geq 1}$  is said to be approximately eventually periodic (**AEP**) if it can be aperiodic for an initial amount of time, and then becomes approximately periodic afterwards. In other words, there are  $\varepsilon > 0$  and two integers  $T, S > 0$  such that  $|x_{t+kT} - x_t| \leq \varepsilon$  for each  $t \geq S$  and each integer  $k \geq 0$ . The integer  $T$  will be called an approximate period of  $\Sigma$ .

*Remark 1.* Based on the notion of approximately eventually periodic signal considered on this study, it can be seen that given an AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$ , there is a positive integer  $S$  such that the tail  $\{x_t\}_{t \geq S}$  of  $\Sigma$  is approximately periodic.

#### 3.1 Semilinear Sparse Autoregressors

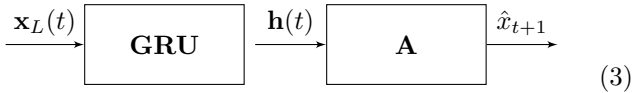
Given an AEP signal  $\Sigma = \{x_t\}_{t \geq 1} \subset \mathbb{C}$  and a lag value  $L > 0$ . For the study reported in this document we will consider semilinear signal models of the form:

$$x_{t+1} = \mathcal{S}(\mathbf{x}_L(t)) = \mathcal{L}(\mathbf{x}_L(t)) + \mathcal{G}(\mathbf{x}_L(t)) + \mathcal{E}(\mathbf{x}_L(t)), t \geq L, \quad (1)$$

where  $\mathcal{L}(\mathbf{x}_L(t))$  denotes a linear operation defined by the expression

$$\mathcal{L}(\mathbf{x}_L(t)) = c_1 x_t + c_2 x_{t-1} + c_3 x_{t-2} + \dots + c_L x_{t-L+1}, \quad (2)$$

the term  $\mathcal{G}(\mathbf{x}_L(t))$  represents a linear combination of neural networks based on gated recurrent unit (**GRU**) layers, whose structure is described by the block diagram



and  $\mathcal{E}(\mathbf{x}_L(t))$  represents some suitable error term. For some given integer  $m > 0$ , each GRU cell  $\mathbf{G}_j$  in the GRU block in (3) is described for each  $j = 1, \dots, m$  by the following equations:

$$\begin{aligned} r_j(t) &= \sigma(\hat{e}_{j,m}^\top (W_{ir} \mathbf{x}_L(t) + W_{hr} \mathbf{h}(t-1) + b_r)) \\ z_j(t) &= \sigma(\hat{e}_{j,m}^\top (W_{iz} \mathbf{x}_L(t) + W_{hz} \mathbf{h}(t-1) + b_z)) \\ n_j(t) &= \tanh(\hat{e}_{j,m}^\top (W_{in} \mathbf{x}_L(t) + b_n) \\ &\quad + r_j(t) \hat{e}_{j,m}^\top (W_{hn} \mathbf{h}(t-1))) \\ h_j(t) &= (1 - z_j(t)) n_j(t) + z_j(t) h_j(t-1) \end{aligned} \quad (4)$$

with  $\mathbf{h}(t) = [h_1(t) \dots h_m(t)]^\top$ , and where  $\sigma$  denotes the sigmoid function. The configuration considered in (3) is based on a GRU layer in order to prevent vanishing gradients, by taking advantage of the GRU structure

presented in Cho et al. (2014). Although LSTM networks can be used to prevent vanishing gradients as well, for the study reported in this document GRU networks were chosen instead of LSTM networks, as they have fewer trainable parameters. The affine layer  $\mathbf{A}$  of the neural network described in (3) is determined by the expression

$$\mathbf{A}(\mathbf{h}(t)) = \mathbf{w}_A^\top \mathbf{h}(t) + b_A.$$

In order to compute models of the form (1), we can combine sparse autoregressive models of the form (2) that can be computed using the methods presented in Vides (2021b), with GRU neural network models of the form (3) that can be computed using the computational tools provided as part of TensorFlow, Keras and PyTorch, that are described as part of Chollet et al. (2015) and Paszke et al. (2019).

An approximate representation

$$\tilde{\mathcal{L}}(\mathbf{x}_L(t)) = \tilde{c}_1 x_t + \tilde{c}_2 x_{t-1} + \tilde{c}_3 x_{t-2} + \dots + \tilde{c}_L x_{t-L+1},$$

of the linear component of (1) such that

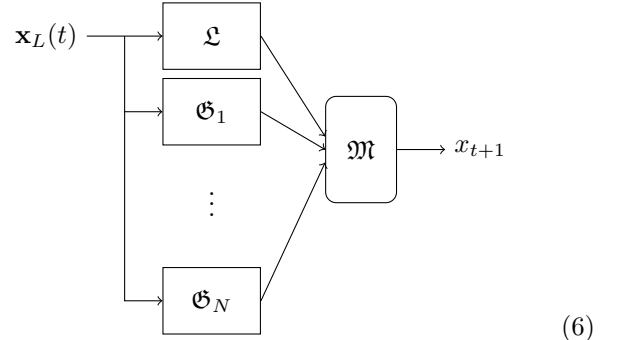
$$\mathcal{L}(\mathbf{x}_L(t)) \approx \tilde{\mathcal{L}}(\mathbf{x}_L(t)), t \geq L$$

can be computed using some sample  $\Sigma_N = \{x_t\}_{t=1}^N$  and a corresponding subsample  $\Sigma_0 = \{x_t\}_{t=1}^{N-1} \subset \Sigma_N$  for some suitable  $N > L$ , by *approximately solving* the matrix equation

$$\mathcal{H}_L(\Sigma_0)^\top \begin{bmatrix} c_L \\ c_{L-1} \\ \vdots \\ c_2 \\ c_1 \end{bmatrix} = \begin{bmatrix} x_{L+1} \\ x_{L+2} \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}, \quad (5)$$

using sparse least-squares approximation methods.

Schematically, the semilinear autoregressors considered in this study can be described by a block diagram of the form,



where the block  $\mathcal{L}$  is represented by (2), each block  $\mathcal{G}_j$  is represented by (3), and where the block  $\mathcal{M}$  is a *mixing* block defined by the expression

$$\mathcal{M}(y_1(t), \dots, y_{N+1}(t)) = \sum_{j=1}^{N+1} w_j y_j(t),$$

for some coefficients  $w_j$  to be determined and some given  $N$ , with  $y_1(t) = \mathcal{L}(\mathbf{x}_L(t))$  and  $y_{k+1}(t) = \mathcal{G}_k(\mathbf{x}_L(t))$  for each  $k = 1, \dots, N$  and each  $t \geq L$ .

The details of the computation of the neural network blocks of model (1) will be omitted for brevity, for details on the theory and computation of the GRU neural network models considered for this study the reader is kindly referred to Cho et al. (2014), Chollet et al. (2015), Paszke et al. (2019) and Vides (2021a).

Several interesting papers have been written on the subject of hybrid time series models that combine ARIMA and ANN models, two important references on this matter are Zhang (2003) and Khandelwal et al. (2015). Besides using sparse AR models instead of ARIMA models, another important distinctive aspect of the modeling approach reported in this document, is that instead of using the recurrent neural network components of (6) represented by  $\mathcal{G}$  in (1) to approximate the residuals  $r_t = x_{t+1} - \mathcal{L}(\mathbf{x}_L(t))$ . Using some suitable training subsets  $\Sigma_I, \Sigma_M$  of a given data sample  $\Sigma_N$  from an arbitrary AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$  under consideration, first the parameters of the blocks  $\mathfrak{L}, \mathfrak{G}_1, \dots, \mathfrak{G}_N$  of (6) are fitted independently using  $\Sigma_I$ , and then the coefficients of the block  $\mathfrak{M}$  of (6) are fitted using  $\Sigma_M$  and some corresponding predicted values generated with  $\mathfrak{L}, \mathfrak{G}_1, \dots, \mathfrak{G}_N$ . Computing sparse representations of some of the matrix parameters of the resulting model along the process.

### 3.2 An Operator Theoretic Approach to the Computation of Linear Components of Semilinear Sparse Autoregressors

Given an AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$  whose behavior can be approximately described by a model of the form (2), that can be computed by approximately solving an equation of the form (5) for some suitable integers  $N, L > 0$  with  $N > L$ , and given some sample  $\Sigma_0 = \{x_t\}_{t=1}^{N-1}$ , if we consider any sample  $\tilde{\Sigma}_0 = \{\tilde{x}_t\}_{t=1}^{N-1} \subset \Sigma$ , such that for some positive integer  $S$  the states in  $\tilde{\Sigma}_0$  satisfy the conditions  $\tilde{x}_t = x_{t+S}$ , for each  $t = 1, \dots, N-1$ . We will have that the matrix

$$C_L = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \\ c_L & c_{L-1} & \cdots & \cdots & c_2 & c_1 \end{bmatrix} \in \mathbb{C}^{L \times L} \quad (7)$$

will *approximately* satisfy the condition

$$\mathcal{H}_L(\Sigma_0)^\top (C_L^S)^\top = \mathcal{H}_L(\tilde{\Sigma}_0)^\top. \quad (8)$$

Using matrices of the form (7) one can express linear models of the form (2) as follows.

$$\mathcal{L}(\mathbf{x}_L(t)) = \hat{e}_{L,L}^\top C_L \mathbf{x}_L(t) \quad (9)$$

One can observe that to each model of the form (2), there corresponds a matrix of the form (7). From here on, a matrix that satisfies the previous conditions will be called the matrix form of a linear model  $\mathcal{L}$  of the form (2).

Given  $\delta > 0$ , and two matrices  $A \in \mathbb{C}^{m \times n}$  and  $Y \in \mathbb{C}^{m \times p}$ , let us write  $AX \approx_\delta Y$  to represent the problem of finding  $X \in \mathbb{C}^{n \times p}$ ,  $\alpha, \beta \geq 0$  and an orthogonal projector  $Q$  such that  $\|AX - Y\|_F \leq \alpha\delta + \beta\|(I_m - Q)Y\|_F$ . The matrix  $X$  will be called a solution to the problem  $AX \approx_\delta Y$ .

**Theorem 2.** Given  $\delta > 0$ , two integers  $L, M > 0$ , a sample  $\Sigma_N = \{x_t\}_{t=1}^N$  from an AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$  with  $N > L$ , and a matrix  $A \in \mathbb{C}^{L \times M}$ . If  $r = \text{rk}_\delta(\mathcal{H}_L(\Sigma_N)) > 0$ , then there is a sparse matrix  $\hat{A} \in \mathbb{C}^{L \times M}$  with at most  $Mr$  nonzero entries such that  $\mathcal{H}_L(\Sigma_N)^\top \hat{A} \approx_\delta \mathcal{H}_L(\Sigma_N)^\top A$ .

**Proof.** Since  $\text{rk}_\delta(\mathcal{H}_L(\Sigma_N)^\top) = \text{rk}_\delta(\mathcal{H}_L(\Sigma_N)) > 0$  by (Vides, 2021b, Lemma 3.2). This result is a consequence of the application of (Vides, 2021b, Theorem 3.6) to the problem  $\mathcal{H}_L(\Sigma_N)^\top \hat{A} \approx_\delta \mathcal{H}_L(\Sigma_N)^\top A$ .

Given an AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$  with approximate period  $T$  and a signal model  $\mathcal{S}$  for  $\Sigma$  of the form (1), if for the linear component  $\mathcal{L}$  of  $\mathcal{S}$  the corresponding residuals  $r_t = |x_{t+1} - \mathcal{L}(\mathbf{x}_L(t))|$  are small, then the significative contribution of  $\mathcal{L}$  to the modeling process of  $\Sigma$ , will be beneficial for interpretability purposes.

Let us consider an AEP signal  $\Sigma$  with approximate period  $T$  and approximately periodic tail  $\tilde{\Sigma} = \{x_t\}_{t \geq S}$ . Given some suitable integer lag value  $L > 0$  such that there is an approximate linear model  $\mathcal{L}$  for  $\Sigma$  of the form (2) with small residuals  $r_t = |x_{t+1} - \mathcal{L}(\mathbf{x}_L(t))|$ . Let  $C_L$  be the matrix form of  $\mathcal{L}$  and let  $s = S + L - 1$ . By applying a Krylov subspace approach along the lines presented in (Saad, 2011, §6.1), and as a consequence of (Vides, 2021b, Theorem 4.3.), one can find a matrix  $W_k \in \mathbb{C}^{L \times k}$  whose columns form an orthonormal basis of the subspace  $\mathcal{K}_T \subset \mathbb{C}^L$  determined by the expression

$\mathcal{K}_T = \text{span}(\{\mathbf{x}_L(s), C_L \mathbf{x}_L(s), C_L^2 \mathbf{x}_L(s), \dots, C_L^{T-1} \mathbf{x}_L(s)\})$ , such that each  $z \in \sigma(W_k^* C_L W_k)$  satisfies the relation  $|z^T - 1| \leq \varepsilon$ , for some  $\varepsilon > 0$ . The matrix  $W_k^* C_L W_k$  will be called an approximately periodic (**AP**)  $\Sigma$ -section of  $C_L$  and will be denoted by  $C_L|_{\Sigma}^{AP}$ .

**Remark 3.** Based on the previous considerations, when a given AEP signal  $\Sigma = \{x_t\}_{t \geq 1}$  with approximate period  $T$  is *well explained* by the linear component of a semilinear model  $\mathcal{S}$  of the form (1), that is, when the corresponding residuals are relatively small, the matrix  $C_L|_{\Sigma}^{AP}$  corresponding to the model should *mimic* the approximate periodicity of the approximately periodic tail  $\{x_t\}_{t \geq S}$  of  $\Sigma$ , in the sense that the number  $\|(C_L|_{\Sigma}^{AP})^T - I_k\|$  should be relatively small for some suitable matrix norm  $\|\cdot\|$  (in the sense of (Saad, 2011, §1.5)). Consequently, the points in  $\sigma((C_L|_{\Sigma}^{AP})^T)$  should cluster around 1.

## 4. ALGORITHMS

The ideas in section §3 can be translated into a prototypical algorithm represented by algorithm 1, that relies on (Vides, 2021b, Algorithm 1), Theorem 2 and (Vides, 2021b, Theorem 4.3.).

We can apply algorithm 1 to compute the model parameters needed for the computation of signal models of the form (6).

## 5. NUMERICAL EXPERIMENTS

In this section, some computational implementations of the methods reported in this document are presented. The experimental results documented in this section can be replicated using the function `NumericalExperiment.py` or the Jupyter notebook `SLSpAARModelsDemo.ipynb`, that are available in Vides (2021a). The configuration required to replicate the results in this section is available as part of the aforementioned programs.

From here on, we will refer to the sparse semilinear models proposed in this document as **SpARS** models. The signal

---

**Algorithm 1: SpARSMoel:** Semilinear Sparse Model Parameters Computation

---

**Data:**  $\Sigma_N = \{x_t\}_{t=1}^N \subset \mathbb{C}$

**Result:**  $\mathbf{c}, \mathfrak{G}_1, \dots, \mathfrak{G}_j, \mathbf{w}_M = \text{SpARSMoel}(\Sigma_N)$

- 0: Estimate the lag value  $L$  using auto-correlation function based methods;
- 1: Approximately solve (5) for  $\mathbf{c}$  using the data in  $\Sigma_N$  and applying (Vides, 2021b, Algorithm 1);
- 2: Fit the model blocks  $\mathfrak{G}_j$  of (6) using the data in  $\Sigma_N$ .
- 3: For the GRU layers of each  $\mathfrak{G}_j$ , compute sparse representations of the corresponding input weights  $W_{ir}, W_{iz}, W_{in}$  in (4) when appropriate, applying Theorem 2.
- 4: Compute the coefficients  $\mathbf{w}_M = (w_1, \dots, w_{N+1})$  of the mixing block  $\mathfrak{M}$  of (6) using the data in  $\Sigma_N$  and (Vides, 2021b, Algorithm 1);

**return**  $\mathbf{c}, \mathfrak{G}_1, \dots, \mathfrak{G}_j, \mathbf{w}_M$

---

approximations computed using the SpARS models presented in this document are compared with the approximations obtained using standard AR models, and the standard AR models are computed using the Python program **Autoreg** included as part of the **statsmodels** module. In this section we will write  $nz$  to denote nonzero elements. For the experiments documented in this section two GRU RNN blocks were used, the block  $\mathfrak{G}_1$  was computed using TensorFlow 2.6.0 and its input weights were replaced by their corresponding sparse representations, that were computed using (Vides, 2021b, Algorithm 1) along the lines of Theorem 2, and the block  $\mathfrak{G}_2$  was computed using PyTorch 1.9.1+cpu and its input weights were left unchanged.

### 5.1 Numerical Experiment 1

In this section, algorithm 1 is applied to compute a SpARS model for the signal data sample recorded in the csv file **AlmostPeriodicSignal.csv** in the DataSets folder in Vides (2021a). The graphic representations of the results produced by the command line

```
>>> NumericalExperiment(1)
```

are shown in figures 1 and 2, respectively.

In every figure like figure 2, the red dots represent the points in each considered spectrum, the blue lines represent  $\mathbf{S}^1$ , and the number  $T$  represents the estimated approximate period for each signal considered.

### 5.2 Numerical Experiment 2

In this section, algorithm 1 is applied to compute a SpARS model for the signal data sample recorded in the csv file **NLOscillatorSignal.csv** in the DataSets folder in Vides (2021a). The graphic representations of the results produced by the command line

```
>>> NumericalExperiment(2)
```

are shown in figures 3 and 4, respectively.

### 5.3 Numerical Experiment 3

In this section, algorithm 1 is applied to compute a SpARS model for the signal data sample recorded in the csv files:

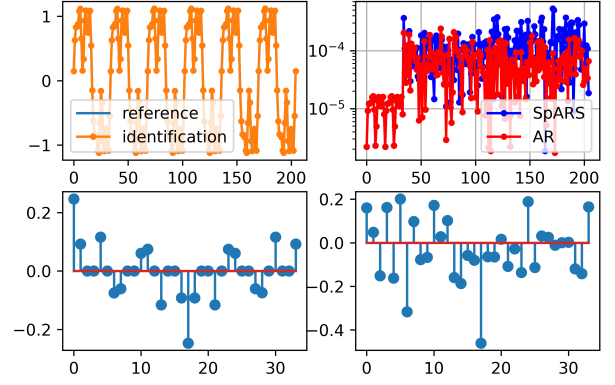


Fig. 1. Reference and identified signals for SpARS model (top left). Approximation errors (top right). Linear component parameters of the SpARS model with 18  $nz$  (bottom left). Linear component parameters of the standard AR model with 34  $nz$  (bottom right).

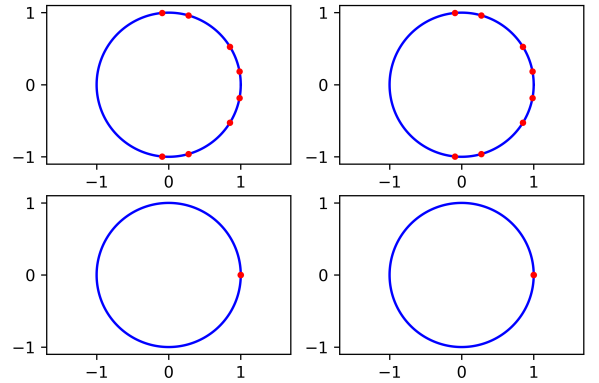


Fig. 2.  $\sigma(C_L|_{\Sigma}^{AP})$  for the linear component of the SpARS model (top left).  $\sigma(C_L|_{\Sigma}^{AP})$  for the linear component of the standard AR model (top right).  $\sigma((C_L|_{\Sigma}^{AP})^T)$  for the linear component of the SpARS model (bottom left).  $\sigma((C_L|_{\Sigma}^{AP})^T)$  for the linear component of the standard AR model (bottom right).

- **art\_daily\_no\_noise.csv**
- **art\_daily\_small\_noise.csv**

that are included as part of the datasets described in Ahmad et al. (2017). The graphic representations of the results produced by the command line

```
>>> NumericalExperiment(3.1)
```

for the periodic signal without noise are shown in figures 5 and 6, respectively. The graphic representations of the results produced by the command line

```
>>> NumericalExperiment(3.2)
```

for the periodic signal with noise are shown in figures 7 and 8.

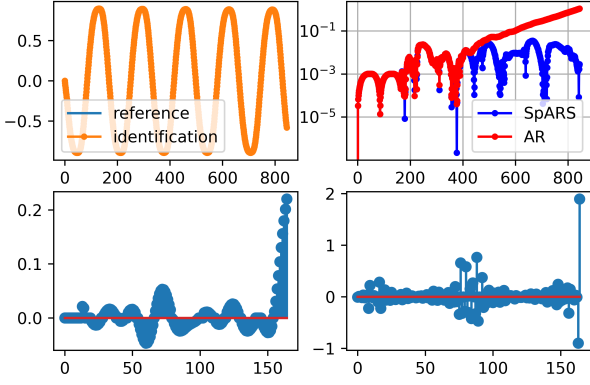


Fig. 3. Reference and identified signals for SpARS model (top left). Approximation errors (top right). Linear component parameters of the SpARS model with 118  $nz$  (bottom left). Linear component parameters of the standard AR model with 165  $nz$  (bottom right).

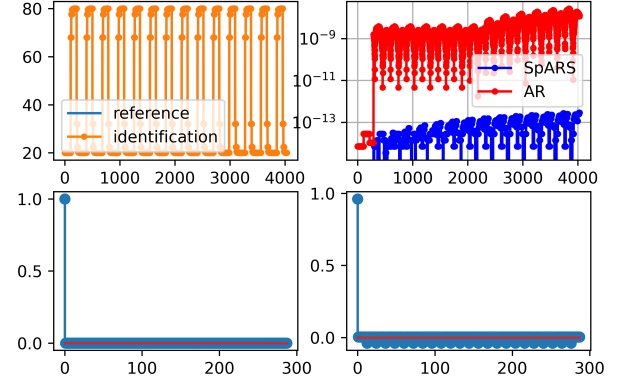


Fig. 5. Reference and identified signals for SpARS model (top left). Approximation errors (top right). Linear component parameters of the SpARS model with 8  $nz$  (bottom left). Linear component parameters of the standard AR model with 288  $nz$  (bottom right).

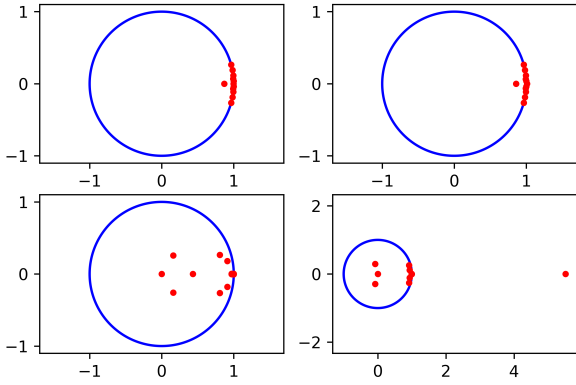


Fig. 4.  $\sigma(C_L|\Sigma^{AP})$  for the linear component of the SpARS model (top left).  $\sigma(C_L|\Sigma^{AP})$  for the linear component of the standard AR model (top right).  $\sigma((C_L|\Sigma^{AP})^T)$  for the linear component of the SpARS model (bottom left).  $\sigma((C_L|\Sigma^{AP})^T)$  for the linear component of the standard AR model (bottom right).

#### 5.4 Approximation Errors

The approximation root mean square errors are summarized in table 1.

Table 1. RMSE

Experiments	SpARS Model	AR Model
Experiment 1	0.0001603580	0.0001450925
Experiment 2	0.0129667794	0.3100591516
Experiment 3.1	0.0000000000	0.0000000074
Experiment 3.2	3.6981093180	4.0939437825

It is appropriate to mention that the root mean square errors can present little fluctuations as one performs several numerical simulations, due primarily to the nature of the neural-network models, as the linear components tend to present very low or no variability from simulation to simulation.

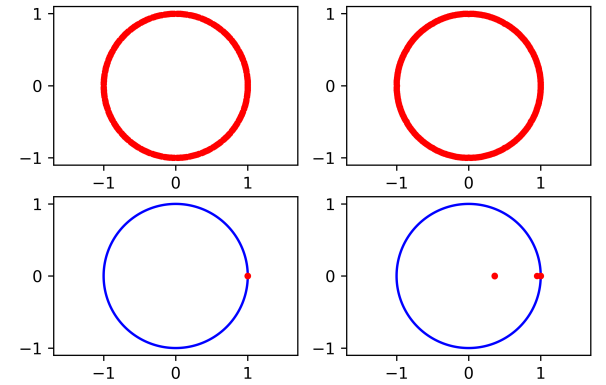


Fig. 6.  $\sigma(C_L|\Sigma^{AP})$  for the linear component of the SpARS model (top left).  $\sigma(C_L|\Sigma^{AP})$  for the linear component of the standard AR model (top right).  $\sigma((C_L|\Sigma^{AP})^T)$  for the linear component of the SpARS model (bottom left).  $\sigma((C_L|\Sigma^{AP})^T)$  for the linear component of the standard AR model (bottom right).

#### 5.5 Data Availability

The Python programs that support the findings of this study are openly available in the SPAAR repository, reference number Vides (2021a). The time series data used for the experiments 1 and 2 documented in §5.1 and §5.2, respectively, are available as part of Vides (2021a). The time series data used for experiment 3 in §5.3 are available as part of the Numenta Anomaly Benchmark (NAB) described in Ahmad et al. (2017).

## 6. CONCLUSION

Although in some experiments in §5 the root mean square errors corresponding to the AR and SpARS models are similar, the AP  $\Sigma$ -sections corresponding to the SpARS models exhibit a better *mimetic approximately periodic*

## ACKNOWLEDGEMENTS

The numerical simulations documented in this paper were performed with computational resources from the Scientific Computing Innovation Center of UNAH, as part of the research project PI-063-DICIHT. Some experiments were performed on Google Colab and IBM Quantum Lab as well.

## REFERENCES

- Ahmad, S., Lavin, A., Purdy, S., and Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262, 134–147. doi: <https://doi.org/10.1016/j.neucom.2017.04.070>. Online Real-Time Learning Strategies for Data Streams.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Association for Computational Linguistics, Doha, Qatar. doi:10.3115/v1/D14-1179. URL <https://aclanthology.org/D14-1179>.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Khandelwal, I., Adhikari, R., and Verma, G. (2015). Time series forecasting using hybrid arima and ann models based on dwt decomposition. *Procedia Computer Science*, 48, 173–179. doi: <https://doi.org/10.1016/j.procs.2015.04.167>. International Conference on Computer, Communication and Convergence (ICCC 2015).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.
- Saad, Y. (2011). *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611970739.
- Vides, F. (2021a). Spaar: Sparse signal identification python toolkit. URL <https://github.com/FredyVides/SPAAR>.
- Vides, F. (2021b). Sparse system identification by low-rank approximation. *CoRR*, abs/2105.07522. URL <https://arxiv.org/abs/2105.07522>.
- Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50, 159–175. doi: [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0).

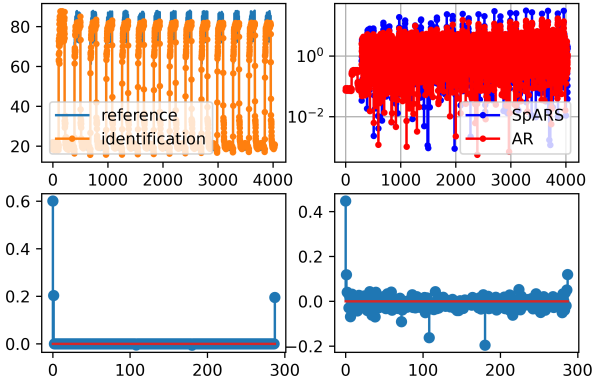


Fig. 7. Reference and identified signals for SpARS model (top left). Approximation errors (top right). Linear component parameters of the SpARS model with 5  $nz$  (bottom left). Linear component parameters of the standard AR model with 288  $nz$  (bottom right).

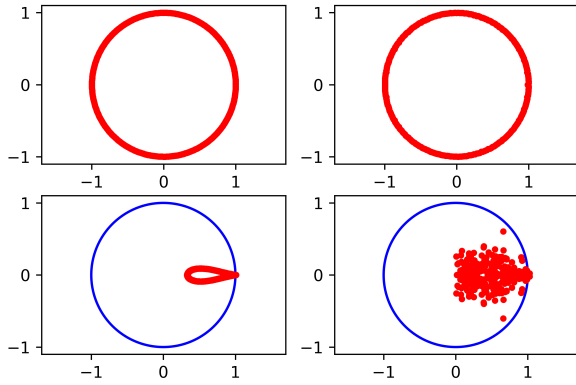


Fig. 8.  $\sigma(C_L|_{\Sigma}^{AP})$  for the linear component of the SpARS model (top left).  $\sigma(C_L|_{\Sigma}^{AP})$  for the linear component of the standard AR model (top right).  $\sigma((C_L|_{\Sigma}^{AP})^T)$  for the linear component of the SpARS model (bottom left).  $\sigma((C_L|_{\Sigma}^{AP})^T)$  for the linear component of the standard AR model (bottom right).

behavior in the sense of remark 3, as it can be visualized in figures 2, 4, 6 and 8. This mimetic behavior is interesting not just from a theoretical point of view, as it provides a criterion for how well the linear component of a given model *mimics or captures* the eventual approximate periodic behavior of the signal under study, but also for practical computational reasons, as long term predictions or simulations can be affected when the eigenvalues of the AP  $\Sigma$ -section of the matrix form corresponding to the linear component of a signal model, lie outside the set  $\mathbf{D}^1 = \{z \in \mathbb{C} : |z| \leq 1\}$ , as one can observe in figures 3 and 4. Another advantage of the matrix parameters sparsity of the linear component of the SpARS models is the reduction of the computational complexity of the corresponding linear component, when compared to a nonsparse linear model based on the same reference data.