Weight identification through global optimization in a new hysteretic neural network model

Elie Leroy, Arthur Marmin, Marc Castella, Laurent Duval

WEIGHT IDENTIFICATION THROUGH GLOBAL OPTIMIZATION IN A NEW HYSTERETIC NEURAL NETWORK MODEL

Elie Leroy[†*] *Arthur Marmin*[†] *Marc Castella*[‡] *Laurent Duval*[⋆]

[†]Université Paris-Saclay, CentraleSupélec, Inria, Center for Visual Computing, Gif-sur-Yvette, France
[‡]Samovar, Télécom SudParis, Institut Polytechnique de Paris, France
[⋆] ESIEE Paris, LIGM, Université Gustave-Eiffel and IFP Energies nouvelles, France

ABSTRACT

Unlike their biological counterparts, simple artificial neural networks are unable to retain information from their past state to influence their behavior. In this contribution, we propose to consider new nonlinear activation functions, whose outputs depend both from the current and past inputs through a hysteresis effect. This hysteresis model is developed in the framework of convolutional neural networks. We then show that, by choosing the nonlinearity in the vast class of rational functions, the identification of the weights amounts to solving a rational optimization problem. For the latter, recent methods are applicable that come with global optimality guarantee, contrary to most optimization methods used in the neural network community. Finally, simulations show that such hysteresis nonlinear activation functions cannot be approximated by traditional ones and illustrate the effectiveness of our weight identification method.

Index Terms— Convolutional neural networks (CNN), hysteresis, polynomial and global optimization

1. INTRODUCTION

Over the last decade, thanks to the increasing computational power of GPUs, Artificial Neural Networks (ANN) have seen their use thrive in handling big data and artificial intelligence problems. Conceptually inspired by biological neural networks, ANN connect neurons, identified as a whole by a set of input weights and nonlinear activation functions. A quite successful neural network architecture in signal and image learning tasks is the Convolutional Neural Network (CNN) which can handle data processing objectives from image content classification [1] to natural language processing [2] by using the same set of weights — called a convolutional filter — for multiple neurons in the same layer.

However, unlike their biological inspiration, most current ANN can be described as "memoryless", disregarding the order and temporal relations of the input data fed to them. This always leads to a loss of information which can be harmful to the overall performance of the network when the output depends on the temporal correlation of the input data. While some models have taken a structural approach to the memory problem in ANN such as Recurrent Neural Nets [3], we focus on how to embed memory into the activation function.

Our approach consists in introducing a hysteresis (dependence of the system's state on its history) into the activation function, to keep track of the input past states and to adapt present and future behaviors accordingly [4, 5, 6]. These models rely on a known activation function, to create their own by introducing two hysteresis branches. The first branch is activated when the input signal is decreasing and the second when it is increasing. The model presented in [4] is a binary hysteresis neural network, based on a binary step while the hysteretic Hopfield neural network presented in [5] is continuous, relying on a hyperbolic tangent. Both use gradient-based resolution methods during training, which comes with some complications when handling the two distinct branches. In the present work, we apply the same methodology to create a new memoryful model called the Softsign Hysteresis Neural Network (SHNN), based on the Softsign activation function [7, 8].

Most ANN rely heavily on local algorithms during their training, usually gradient-based, to find the optimal set of weights for their neurons. If the underlying optimization problem is nonconvex, which is common, those methods always face the risk of getting trapped near a local minimum and can be slow when used on activation functions with vanishing gradient. Global optimization methods ensure convergence to the best feasible weights regardless of existing local minimima and of gradient's shape. One such methods which is specifically designed for rational (or polynomial) loss functions is the moment-SoS (Sum of Squares) relaxation also known as the Lasserre relaxation [9]. It has recently been considered in the signal processing community, providing good results on sparse signal reconstruction problems [10, 11].

The SHNN proposed in this article is rational and the associated optimization problem can therefore be solved glob-

[*]This research was supported by DATAIA convergence institute as part of the "Programme d'Investissement d'Avenir", (ANR- 17-CONV-0003) operated by CentraleSupelec and IFP Energies nouvelles.

ally using Lasserre relaxation. This genuine combination of a polynomial model and rational nonlinearity allowed us to successfully identify the filter coefficients. It can adapt easily to any semi-algebraic function such as the Rectified Linear Unit (ReLU) and is able to approximate with an arbitrary precision many other activation functions from the sigmoid to the hyperbolic tangent (or S-curve). Finally, note that similar hysteresis models and identification problems may appear in many applications of signal processing to physics, control, electronics (see e.g. [12] and references therein).

Our paper is organized as follows: Section 2 presents our model of hysteresis CNN. Section 3 sets the optimization problem to identify the weights of the network. The Lasserre relaxation is also briefly sketched. Simulations results are discussed in Section 4. Finally some concluding remarks are drawn in Section 5.

2. PROPOSED HYSTERESIS MODEL

2.1. Observation model

Our work focuses on the identification of a single layer CNN. If $\mathbf{x} \in \mathbb{R}^T$ and $\mathbf{y} \in \mathbb{R}^T$ denote respectively the input and output of the network, we assume that the following relation holds:

$$\mathbf{y} = \bar{\mathbf{y}} + \mathbf{n} = \Phi(\bar{\mathbf{w}} * \mathbf{x}) + \mathbf{n} \,, \tag{1}$$

where $\bar{\mathbf{y}} = \Phi(\bar{\mathbf{w}} * \mathbf{x})$ is the noiseless network output, $\mathbf{n} \in \mathbb{R}^T$ is an additive Gaussian noise, and $\bar{\mathbf{w}} \in \mathbb{R}^m$ is the weight vector of the convolutional filter which is identical for all neurons. The notation $\bar{\mathbf{w}} * \mathbf{x}$ stands for the usual convolution of $\bar{\mathbf{w}}$ and $\mathbf{x}$. It is defined as the vector in $\mathbb{R}^T$ whose elements are $(\bar{\mathbf{w}} * \mathbf{x})_t = \sum_{i=1}^{m} \bar{w}_i x_{t+1-i}$, where the elements of $\mathbf{x}$ with negative indices are zero by convention. Finally, the function $\Phi : \mathbb{R}^T \to \mathbb{R}^T$ represents the nonlinear activations of the neurons of the layer. It is defined as the component-wise function $\Phi = \left(\Phi^{(t)}\right)_{1 \leq t \leq T}$, with, for any t in $\{1, \ldots, T\}$, $\Phi^{(t)} : \mathbb{R} \to \mathbb{R}$ being the activation function of the t^{th} neuron.

2.2. Hysteresis activation function

We extend here the nonlinearity model classically used for the activation function to hysteretic ones by creating two branches. In other words, writing φ for a nonlinear S-shape activation function, we define our new model by introducing a fixed delay parameter $\theta > 0$ to separate two branches:

$$\Phi^{(t)}(u_t|u_{t-1}) = \begin{cases} \varphi(u_t + \theta) \text{ if } u_t \leq u_{t-1} \\ \varphi(u_t - \theta) \text{ if } u_t > u_{t-1} \,. \end{cases} \tag{2}$$

This hysteresis activation function for the t^{th} neuron is illustrated in Figure 1 with the lower (respectively higher) branch being activated when the argument is increasing (respectively decreasing).

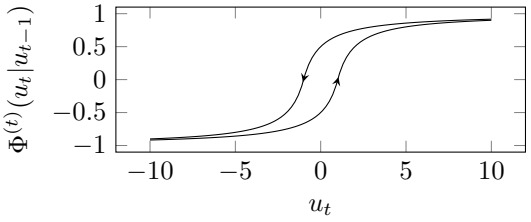

Fig. 1. Hysteresis Softsign activation function. The parameters $\theta = 1$, $\delta = 1$ are the ones used in the simulations of Section 4.

2.3. Hysteresis Neural Network Training

Given the observation model of Section 2.1, and for a given nonlinear function φ, our goal and main contribution consist in proposing a supervised learning method which is able to estimate the weights $\bar{\mathbf{w}}$. For a given training input signal $\mathbf{x}_{\text{train}}$ and its associated output $\mathbf{y}_{\text{train}}$, a classical approach consists in minimizing an ℓ_2 error term, which yields the problem of finding

$$\hat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^m}{\operatorname{argmin}} \, \|\mathbf{y}_{\text{train}} - \Phi(\mathbf{w} * \mathbf{x}_{\text{train}})\|_2^2 \,.$$

The reconstructed output $\hat{\mathbf{y}} = \Phi(\hat{\mathbf{w}} * \mathbf{x})$ gives an estimation of the expected output $\bar{\mathbf{y}}$. Considering the hysteresis model from (2), we obtain the optimization problem

$$\underset{\mathbf{w} \in \mathbb{R}^m}{\text{minimize}} \quad \sum_{t=1}^{T} \left(y_t - \Phi^{(t)}((\mathbf{w} * \mathbf{x})_t | (\mathbf{w} * \mathbf{x})_{t-1}) \right)^2 \tag{3}$$

Problem (3) involves mixed discrete and continuous variables, to handle respectively the two branches and the continuous nonlinearity. As such, it is nonconvex, which is a potentially challenging task. We now illustrate specific choices for the activation function φ that allows one to use an advanced global optimization approach.

3. GLOBAL WEIGHT IDENTIFICATION

3.1. Softsign Hysteresis

Several common activation functions [13] can be parametrized by rational or piecewise rational functions, i.e. a quotient of polynomials. Among them, the ReLU (Rectified Linear Unit) and the Square Nonlinearity (SQNL) [14] can be explicitly modeled through polynomial functions and constraints. More generally, any activation function, such as the sigmoid or the arctangent, can be tightly approximated by a suitable rational function [15].

Our work was performed on a Softsign hysteresis, based on the Softsign defined as $\text{soft}_\delta : u \mapsto \frac{u}{\delta + |u|}$ where $\delta > 0$ is a fixed parameter. Consequently, choosing soft_δ for the function φ in (2), our activation function is defined by:

$$\Phi^{(t)}(u_t|u_{t-1}) = \text{soft}_\delta(u_t - \text{sign}(u_t - u_{t-1})\theta) \tag{4}$$

This activation function can be expressed by rational functions and constraints. The absolute value can be expressed using polynomial constraints as shown in [10], by noting that $|u|$ can be represented by the unique element in the set $\{a \in \mathbb{R} \mid a \geq 0,\, a^2 = u^2\}$. Similarly, $\operatorname{sign}(u_t - u_{t-1})$ has a polynomial formulation by introducing the extra real variables ξ_t such that $\xi_t^2 = 1$ and $(u_t - u_{t-1})\xi_t \geq 0$.

This will be especially important during the training of our network, since it will be possible to rely on global optimization methods suited for polynomial, rational and more generally semi-algebraic functions.

3.2. Softsign Hysteresis Neural Network

Using the above activation function, we construct a single layer CNN that we call the Softsign Hysteresis Neural Network. Based on the rational/polynomial modeling of the activation function, we obtain the important feature that the training of our hysteresis model, similarly to the method in [16], translates to the minimization of a rational function under polynomial constraints. More precisely, training our network boils down to solving the problem:

$$\begin{aligned} \underset{(\mathbf{w},\mathbf{a},\boldsymbol{\xi})\in\mathbb{R}^m\times\mathbb{R}^T\times\mathbb{R}^T}{\text{minimize}} \quad & \sum_{t=1}^{T}\left(y_t - \frac{(\mathbf{w}*\mathbf{x})_t - \xi_t\theta}{\delta + a_t}\right)^2 \\ \text{s.t.} \quad & \begin{cases} a_t \geq 0 \\ a_t^2 = \left((\mathbf{w}*\mathbf{x})_t + \xi_t\theta\right)^2 \\ \xi_t^2 = 1 \\ \left((\mathbf{w}*\mathbf{x})_t - (\mathbf{w}*\mathbf{x})_{t-1}\right)\xi_t \geq 0\ . \end{cases} \end{aligned} \tag{5}$$

Problem (5) is rational and therefore is well adapted to the Lasserre relaxation which is sketched out in the next section.

3.3. Lasserre relaxation for global optimization

In order to solve Problem (5), we use the framework of Lasserre [9] which is suited for finding global optima of polynomial optimization problems. It has been extended to the minimization of a sum of rational fractions in [17], which corresponds to our criterion.

Lasserre's approach translates a polynomial or rational problem on a compact set of $\mathbb{R}^N$ into a moment problem. Truncating the moment sequence at a given relaxation order yields a hierarchy of convex semidefinite programming (SDP) relaxations. Solving the latter results in a sequence of non-decreasing lower bounds converging to the global optimum of the initial problem. It is known that convergence occurs generically at a finite order [18] and the minimizers can be extracted [19]. Equality of the lower-bound and the criterion value also ensures global optimality.

	$\left\|\hat{\mathbf{y}}_{\text{train}} - \mathbf{y}_{\text{train}}\right\|^2 / T_{\text{train}}$	$\left\|\hat{\mathbf{y}}_{\text{test}} - \bar{\mathbf{y}}_{\text{test}}\right\|^2 / T_{\text{test}}$	$\left\|\hat{\mathbf{w}} - \bar{\mathbf{w}}\right\| / \left\|\bar{\mathbf{w}}\right\|$
Hysteresis	0.018	0.012	$5.79\,\%$
Memoryless	0.268	0.261	$79.69\,\%$

Table 1. Comparison of the performance of hysteresis and memoryless Softsign models on data reconstruction and filter estimation for $T = 250$, $m = 4$.

4. NUMERICAL EXPERIMENTS

4.1. Implementation

Simulations were operated on MATLAB R2020a using packages GloptiPoly [20] to model the polynomial optimization problem and extract the global optima from the SDP solution, YALMIP [21] to manage the SDP solver called for the relaxation, and SDPT3 [22] as the actual SDP solver. These simulations were run without parallelization on an Intel Xeon W-2135 CPU at 3.70 GHz with 6 cores and 12 threads using up to 64 GB of memory.

4.2. Experimental framework

The input $\mathbf{x}$ was sampled i.i.d. with standard normal distributions, $20\,\%$ of these data were selected as the training set $\mathbf{x}_{\text{train}}$ and the remaining $80\,\%$ were used as the testing set $\mathbf{x}_{\text{test}}$. The ground truth convolutional filter weights in $\bar{\mathbf{w}}$ were sampled i.i.d. with a uniform distribution over $[0, 1]$. The noise $\mathbf{n}$ was sampled i.i.d. with a centered normal distribution of standard deviation $\sigma_0 \frac{\|\Phi(\bar{\mathbf{w}}*\mathbf{x})\|_2}{\sqrt{T}}$ where the noise level was set to $\sigma_0 = 30\,\%$. The noisy output $\mathbf{y}$ was generated following Equation (1), using the Softsign hysteresis activation function in Equation (4) with parameters set to $\theta = 1$ and $\delta = 1$. Small changes in these values did not modify our results. The relaxation order in the Lasserre hierarchy was set to 3.

4.3. Results

We compared the two models SHNN and memoryless Softsign based neural network. The filter weights were randomly generated $\bar{\mathbf{w}} = [0.5247, 0.6412, 0.0162, 0.8369]$ and data of size $T = 250$ was generated with a time dependency using the same hysteresis technique given by Equations (1) and (4). The estimation and prediction results are displayed in Table 1. As one could expect, usual activation functions are not adapted to reconstruct data with a strong dependency to the past.

To illustrate performance, we generated 20 Monte Carlo samples of input data $\mathbf{x}$ of size $T = 250$ and filters $\bar{\mathbf{w}}$ of size $m = 3$.

The training was performed on a subdataset $\mathbf{x}_{\text{train}}$, and the associated $\mathbf{y}_{\text{train}}$, of size $T_{\text{train}} = 50$, solving problem (5). An estimated filter $\hat{\mathbf{w}}$ was reconstructed for each Monte Carlo sample. Figure 2 represents the relative error of $\hat{\mathbf{w}}$ compared to $\bar{\mathbf{w}}$ ordered increasingly, along with the associated training loss and its lower bound returned by the Lasserre hierarchy.

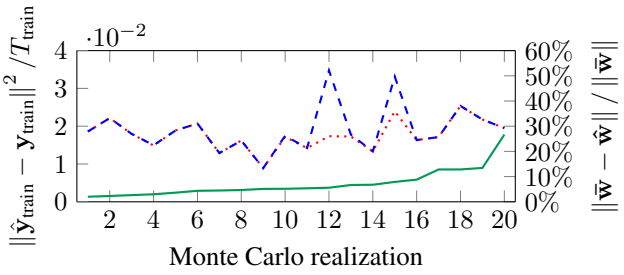

Fig. 2. Training phase : Relative error of the estimated filter (— solid green) ordered increasingly reported on the right axis. Training loss (- - - dashed blue) and lower bound returned in the Lasserre relaxation ($\cdots$ dotted red) reported on the left axis.

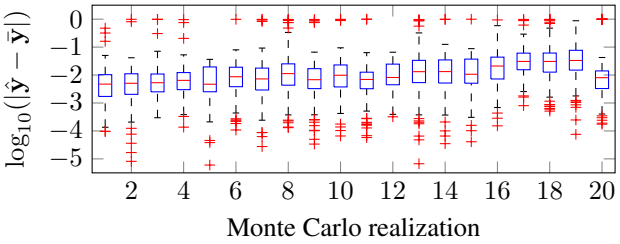

Fig. 3. Testing phase: noiseless testing error distributions over the 20 Monte Carlo samplings.

When the loss and lower bound are equal, up to numerical precision, the Lasserre hierarchy has converged and global optimality is certified. This may not be always the case, as illustrated by the 12th and 15th Monte Carlo samplings, when an "hysteresis leap" occurs during training: when two consecutive values $(\bar{\mathbf{w}} * \mathbf{x})_t$ and $(\bar{\mathbf{w}} * \mathbf{x})_{t-1}$ are too close, a small error in $\hat{\mathbf{w}}$ may cause a change in monotony and a switch of value for the associated ξ_t (this can be seen on Figure 4 at the vertical dotted lines). This results in a leap from one hysteresis branch to another, which greatly increases the loss. However, in all cases and even when global optimality cannot be certified, the relative error of the estimated filter remains low, which illustrates that our method is successful and robust for estimation of $\bar{\mathbf{w}}$.

The learnt filters were tested on the remaining data of size $T_{\text{test}} = 200$, comparing the noiseless model target $\bar{\mathbf{y}} = \Phi(\bar{\mathbf{w}} * \mathbf{x}_{\text{test}})$ and the reconstructed $\hat{\mathbf{y}} = \Phi(\hat{\mathbf{w}} * \mathbf{x}_{\text{test}})$. In Figure 3, the box plots give an overview of the distribution of the logarithmic error along the same 20 Monte Carlo samplings ordered similarly to Figure 3. The central red bar indicates the median, the blue box delimits the span between the 25th and 75th percentiles and the whiskers extend to the most extreme data points not considered outliers, which fall out of the interquartile range by a deviation of more than 1.5 its span, and are identified by red marks.

These distributions show that the error remains consistent across all Monte Carlo realization. The upper outliers on the error with values close to 1 are caused by the same

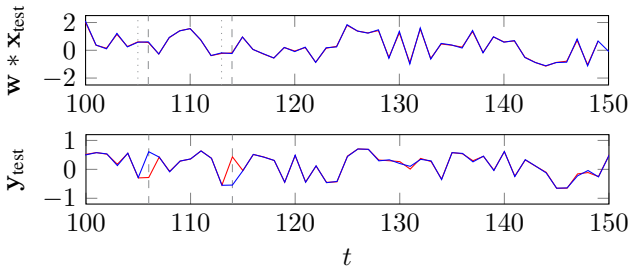

Fig. 4. Zoom on the ground truth (blue) and reconstructed (red) convolution vectors (top) and on the noiseless (blue) and reconstructed (red) output (bottom) for the 10th Monte Carlo realization.

	$\|\hat{\mathbf{y}}_{\text{test}} - \bar{\mathbf{y}}_{\text{test}}\|^2 / T_{\text{test}}$	$\|\hat{\mathbf{y}}_{\text{test}} - \mathbf{y}_{\text{test}}\|^2 / T_{\text{test}}$	$\|\bar{\mathbf{w}} - \hat{\mathbf{w}}\| / \|\bar{\mathbf{w}}\|$
T=100	3.2×10^{-3}	33.0×10^{-3}	11.67 %
T=250	0.8×10^{-3}	12.6×10^{-3}	7.21 %
T=500	0.3×10^{-3}	7.3×10^{-3}	4.32 %

Table 2. Comparison of the performance on data reconstruction and filter estimation for $m = 3$, $T = 100/250/500$.

phenomenon of hysteresis leaps observed during the training phase. As shown in the temporal reconstructions from Figure 4, the reconstructed convolutions are virtually identical. Yet when two consecutive convolution values are too close (highlighted by vertical dotted lines), a small error in $\hat{\mathbf{w}}$ results in a big error in $\hat{\mathbf{y}}$ due to the discrete nature of the hysteresis parameter ξ_t. This hurts common quantitative metrics such as the signal-to-noise ratio or mean relative error, while not impacting much the overall quality of most of the reconstructed points in the signal.

We also compared the performance for $m = 3$ and different sizes of data $T = 100, 250, 500$ still using 20 % of data as training set to compare reconstruction quality and filter precision. The results of these simulations are presented in Table 2. As expected, the bigger the data set, the more precise the filter reconstructed.

5. CONCLUSION

We proposed an extension of usual ANN models to include memory in their handling of data. It is based on hysteresis activation functions, which makes the history of the data become an important feature in the training of the network. This approach is different and complementary to architecture based methods such as RNN and LSTM. By translating our model to polynomial and rational equations, we were able to perform a global optimization of the problem associated to the filter weights identification, thus avoiding the drawback of possible local minima in traditional gradient-based training. Our tests on simulated data illustrate the pertinence of our hysteresis model and its good overall performance to reconstruct various filters.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[2] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jun. 2014, pp. 655–665, Association for Computational Linguistics.

[3] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, "Recurrent neural network regularization," in *Int. Conf. Learning Representations*, May 7-9, 2015.

[4] Yoshiyasu Takefuji and K. C. Lee, "An artificial hysteresis binary neuron: a model suppressing the oscillatory behaviors of neural dynamics," *Biol. Cybern.*, vol. 64, no. 5, pp. 353–356, Mar. 1991.

[5] Sunil Bharitkar and Jerry M. Mendel, "The hysteretic Hopfield neural network," *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 879–888, July 2000.

[6] Chunbo Xiu and Yuxia Liu, "Hysteresis response neural network and its applications," in *Proc. ISECS International Colloquium on Computing, Communication, Control, and Management*, Aug. 8-9, 2009.

[7] David L. Elliott, "A better activation function for artificial neural networks," Tech. Rep. 93-8, Institute for Systems Research, University of Maryland, 1993.

[8] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[9] Jean-Bernard Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, Jan. 2001.

[10] Marc Castella, Jean-Christophe Pesquet, and Arthur Marmin, "Rational optimization for nonlinear reconstruction with approximate ℓ_0 penalization," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1407–1417, Mar. 2019.

[11] Arthur Marmin, Marc Castella, Jean-Christophe Pesquet, and Laurent Duval, "Sparse signal reconstruction for nonlinear models via piecewise rational optimization," *Signal Process.*, vol. 179, pp. 107835, Feb. 2021.

[12] Dan Schonfeld and Gary Friedman, "On the optimality of hysteresis operators in signal processing and communication systems," *J. Franklin Inst.*, vol. 342, no. 7, pp. 749–759, 2005.

[13] Prajit Ramachandran, Barret Zoph, and Quoc V. Le, "Searching for activation functions," in *Int. Conf. Learning Representations*, Apr. 30-May 3, 2018.

[14] Adedamola Wuraola and Nitish Patel, "SQNL: A new computationally efficient activation function," in *Proc. Int. Joint Conf. Neur. Netw.*, Jul. 8-13, 2018.

[15] Germund Dahlquist and Åke Björck, *Numerical Methods in Scientific Computing, Volume I*, Society for Industrial and Applied Mathematics, Jan. 2008.

[16] Arthur Marmin, Marc Castella, and Jean-Christophe Pesquet, "How to globally solve non-convex optimization problems involving an approximate ℓ_0 penalization," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, May 12-17, 2019, pp. 5601–5605.

[17] Florian Bugarin, Didier Henrion, and Jean-Bernard Lasserre, "Minimizing the sum of many rational functions," *Math. Program. Comput.*, vol. 8, no. 1, pp. 83–111, Aug. 2015.

[18] Jiawang Nie, "Optimality conditions and finite convergence of Lasserre's hierarchy," *Math. Programm.*, vol. 146, no. 1-2, pp. 97–121, Aug. 2013.

[19] Didier Henrion and Jean-Bernard Lasserre, "Detecting global optimality and extracting solutions in GloptiPoly," in *Positive Polynomials in Control*, vol. 312, pp. 293–310. Springer Berlin Heidelberg, 2005.

[20] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg, "GloptiPoly 3: moments, optimization and semidefinite programming," *Optim. Methods Softw.*, vol. 24, no. 4-5, pp. 761–779, Oct. 2009.

[21] Johan Löfberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *Proc. IEEE International Conference on Robotics and Automation*, Sep. 2-4, 2004.

[22] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü, "SDPT3 — a Matlab software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, no. 1-4, pp. 545–581, Jan. 1999.