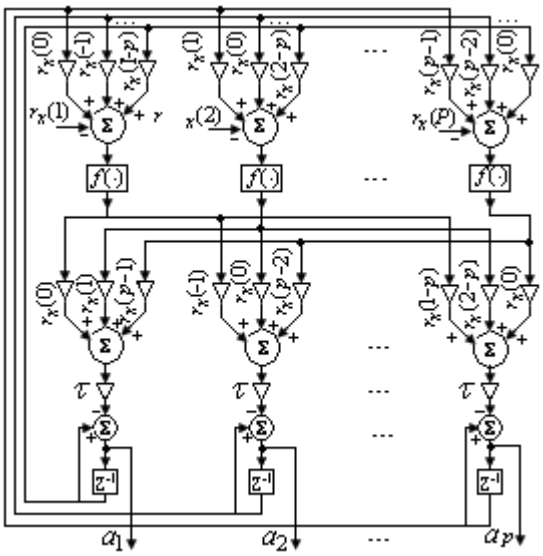

Fig. 1. Architecture of DGNN model

A discrete model of GNN can be obtained by the use of the forward-difference rule to compute $\dot{a}(t)$.

$$\dot{a}(t=kh) \approx (a((k+1)h) - a(kh))/h \tag{17}$$

where h denotes the sampling interval. For presentation convenience, we use $a_k = a(t=kh)$. Thus the presented DGNN model (14) can be reformulated as:

$$a_{k+1} = a_k - \tau R^T f(Ra_k - r) \tag{18}$$

with $\tau = \gamma h$. Figure (1) shows the architecture of the discrete neural network. As we can see, we have $2p^2$ weighting function, p adders of p elements, p adders of $p+1$ elements and p time-delays.

B. Spectrum estimation using DGNN model and the FFT algorithm

In this subsection, we describe the spectrum estimation procedure using both the DGNN model and the super-fast algorithm proposed in [5]. As mentioned below, in the exression (4), the terms $A(f)$ and $B(f)$ may be evaluated by the use of the FFT algorithm. Instead of using the Levinson-Durbin algorithm to compute the AR parameters, we see that the DGNN constitutes a good alternative to compute these parameters. So, the proposed spectrum estimation algorithm can be described as follow: ones the AR parameters are computed using the DGNN model, we compute the two terms $A(f)$ and $B(f)$ using the FFT algorithm and finally we need some algebric operators to compute the spectrum. The figure (2) shows the diagram of the proposed scheme.

IV. Compact architecture of the DGNN

By settig $z_1 = Ra_k$, $z_2 = f(z_1 - r)$ and $z_3 = R^T z_2$, the dynamic of the neural network can be rewritten in a compact form as:

$$a_{k+1} = a_k - \gamma h z_3 \tag{19}$$

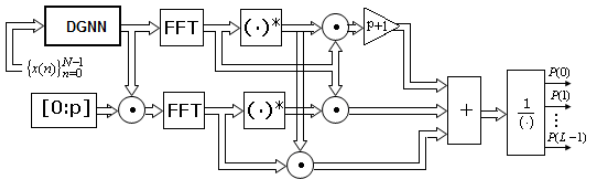

Fig. 2. Diagramm of the proposed model

This equation consists of two Toeplitz matrix-vector products $z_1 = Ra_k$ and $z_3 = R^T z_2$ which can be computed efficiently using the following algorithm.

A. Fast matrix-vector product computation

Since R is a Toeplitz matrix which is fully defined by its first column and first row, thus it depends only on $2p-1$ parameters rather than p^2 . To compute the product z_1 , the Toeplitz matrix R can be first embedded into a circulant matrix $C \in \mathcal{R}^{2p \times 2p}$ as [16], [17]:

$$C = \begin{bmatrix} R & S \\ S & R \end{bmatrix} \tag{20}$$

where

$$S = \begin{bmatrix} 0 & r_x(p-1) & \cdots & r_x(1) \\ r_x(1-p) & 0 & \cdots & r_x(2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(-1) & r_x(-2) & \cdots & 0 \end{bmatrix} \tag{21}$$

The matrix S never needs to be formed explicitly as C is simply a Toeplitz matrix with columns described by a circular-shift of the first column of the matrix given by:

$$c_1 = [r_x(0) r_x(1) \cdots r_x(p-1) 0 r_x(1-p) r_x(2-p) \cdots r_x(-1)] \tag{22}$$

Now we form a new matrix-vector product as:

$$C \cdot \begin{bmatrix} a_k \\ 0_n \end{bmatrix} = \begin{bmatrix} R & S \\ S & R \end{bmatrix} \cdot \begin{bmatrix} a_k \\ 0_n \end{bmatrix} = \begin{bmatrix} Ra_k \\ Sa_k \end{bmatrix} \tag{23}$$

Note that the vector $\begin{bmatrix} a_k & 0_n \end{bmatrix}^T$ is simply the vector a_k zero padded to the length of c_1 and will be noted x_p . Then the equation (21) will be rewritten as:

$$C \cdot x_p = \begin{bmatrix} Ra_k \\ Sa_k \end{bmatrix} \tag{24}$$

The product Ra_k can be computed efficiently using the following algorithm [16]:
- Compute $X_p = FFT(x_p)$.
- Compute $w = FFT(c_1)$.
- Compute the element wise vector-vector product $H = X_p \odot w$.
- Compute $z = IFFT(H)$.

The p first elements of the vector z constitute the product Ra_k. since the FFT algorithm can be done in $O(plogp)$ operations, the product Ra_k can be also obtained in $O(plogp)$ operations [16], [17]. Figure (3) shows the block diagram illustrating the fast matrix-vector multiplication.

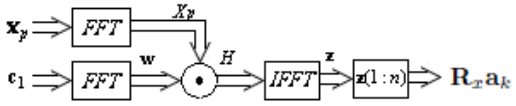

Fig. 3. Block diagram illustrating the fast matrix-vector multiplication

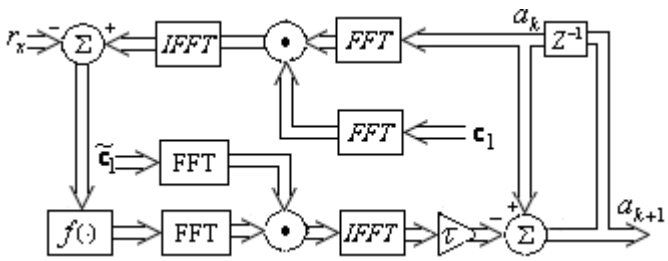

Fig. 4. Block diagrams of the CDGNN model

B. Proposed architecture of CDGNN

To compute the product $z_3 = R^T z_2$, we just replace R by R^T and a_k by z_2 . We note here that the matrix R^T is Toeplitz, then it can be generated only by its first row and first column. Furthermore the Teoplitz matrix R^T can be embedded into a circulant matrix C^T. Let $\tilde{c}_1$ be the first column of the matrix C^T and $\tilde{w}$ its Fourier transform. The block diagram realizing and the detailed architecture of the proposed neural network are shown in the figures (4) and (5) respectively. As we can see, the FFTs of the column c_1 and $\tilde{c}_1$ constitutes the connection weighting of the neural network. So we have just $4p$ weighting function instead of $2p^2$ in the original DGNN. The entire circuit contains 4 blocks FFT/IFFTs, $2p$ adders of 2 elements, p time-delays, and $4p$ weighted connections.

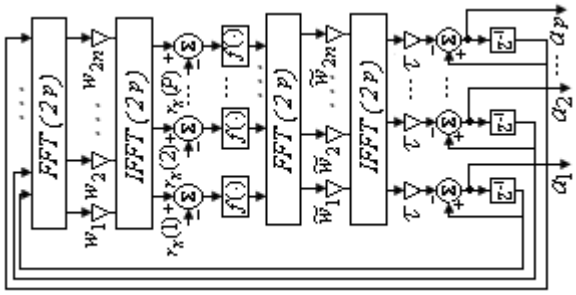

Fig. 5. Architecture of the proposed neural network

C. Complexity and Comparison

The complexity of a neural network is defined as the total number of multiplications per iteration. Since the FFT/IFFT algorithm using p points requires $0.5plogp$ multiplications, then it can be seen that the proposed neural network model requires $5p + 4plog2p$ multiplications per iteration instead of $2p^2 + p$ for the original DGNN model. As result, the computational complexity of the proposed neural network is reduced to $O(plogp)$ multiplications.
Concerning the memory storage, in addition to $O(plogp)$ memory required for the FFT/IFFT blocks, we need to store the $6n$ elements of the vectors c_1,$\tilde{c}_1$,b , and the outputs, thus only $O(plogp)$ elements need storage in the proposed CDGNN model instead of $O(p^2)$ elements in the original model.

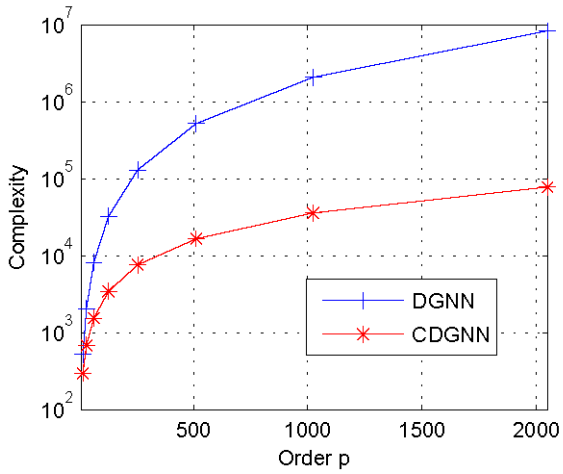

Fig. 6. Computational complexity of the two networks

V. Computer Simulations

Computer simulations have been performed to assess the performance of the proposed method in term of accuracy and computational complexity. In the first experiment,we will show the effectiveness of the CDGNN model for the AR parameters estimation. While in the second experiment,we will consider the estimation of a stationary process consisting of a mixture of two complex sinusoids corrupted by an additive zero-mean Gaussian noise.

A. AR parameters estimation

Let $x(n)$ an AR(4) process given by

$$x(n) = 2.0371x((n-1) - 2.4332x(n-2) + 1.7832x(n-3) - 0.7019x(n-4) + e(n) \quad (25)$$

where the input process $e(n)$ is a white Gaussian process with zero mean and variance σ_e^2. In this example, we let the number of process samples $N = 64$, and $h = 10^{-4}$ and $\gamma = 195000$. Figures (7)and (8) show the convergence behaviour of the original network and its simplified version. As we can see from figure (7), starting from a random initial state a_0 , the two networks converge in the same manner to the true parameters which implies the correctness of the proposed scheme. In the figure (8), we show the convergence error of the two networks, the residual error for the two networks is about 10^{-14} after convergence.

B. Spectrum estimation

The data used in this example consists of two sinusoids embedded in noise.

$$x(n) = \sum_{k=1}^{2} A_k \cos(2\pi f_k n) + e(n), n = 1, 2, \cdots, N \quad (26)$$

ISSN: 2692-5079

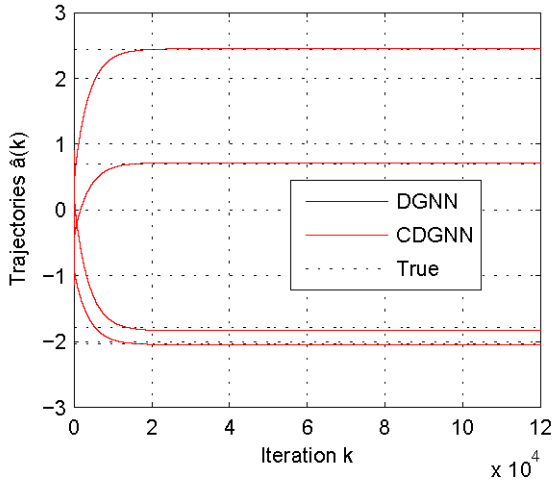

Fig. 7. Convergence behaviour of the state trajectory.

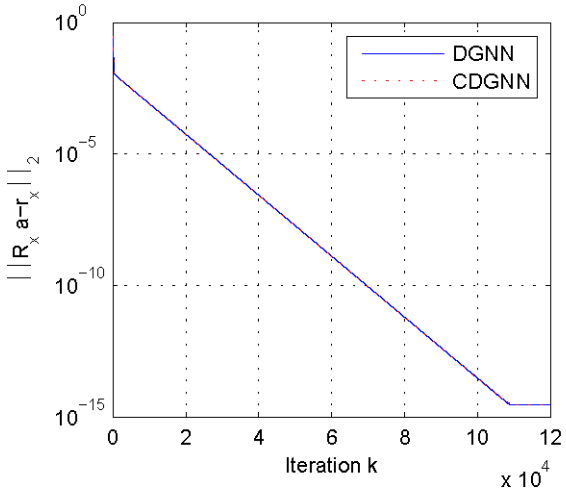

Fig. 8. Convergence error of the two neural networks.

The added noise $e(n)$ is white Gauusian. The amplitudes and frequencies of sinusoids A_1, A_2, $f_1$andf_2 were chosen as 1,1.5,0.1 and 0.3 respectively. The data samples we used in this simulation was 256. The neural network parameters were chosen as $h = 10^{-4}$,$\gamma = 500$, the model order $p = 8$ and the number of the FFT points is 512. Figure (9) shows the power spectral densities obtained by the use of the original method cited in [5] and the proposed one. As we can see, the two Methods lead to a similar results.

VI. Conclusion

Recurrent neural networks are very useful as computational models for solving computationally intensive problems due to their inherent nature of parallel and distributed information processing. The Toeplitz structure of the correlation matrix allows us to design a compact implementation of the gradient based neural network for solving Toeplitz systems using the fast Fourier transform to compute the neural network activation. The proposed reduced neural network is very suitable for estimating the AR parameters of models with large order.

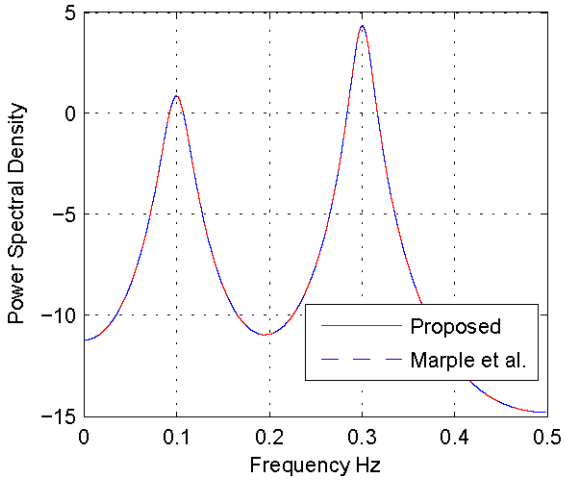

Fig. 9. Spectral density estimation using the proposed scheme

Computer simulations show that the proposed algorithm is very suitable for AR parameters and spectral density estimation.

References

[1] S. M. Kay, editors, *Modern Spectral Estimation: Theory ad Application*, NJ: Prentice-Hall,Englewood Cliffs, 1988.
[2] S.L. Marple, editors, *Digital Spectral Analysis with Applications*, NJ: Prentice-Hall,Englewood Cliffs, 1987.
[3] J. Capon, High-resolution frequency-wavenumber spectrum analysis, *Proc. IEEE*, 57, 1408–1418, 1969.
[4] B. R. Musicus, Fast MLM Power Spectrum Estimation from Uniformly Spaced Correlations, *IEEE Trans. Acoustics, Speech and Signal Processing* , 33(4), 1333-1335, 1985.
[5] S. L. Marple, M. Adeli and H. Liu, Super-Fast Algorithm for Minimum Variance (Capon) Spectral Estimation, *Conf. Signals, Systems and Computers* , 1832-1836, 2010.
[6] S.K. Park, Hopfield Neural Network for AR Spectral Estimator, *in Proc. IEEE* , 487-490, 1990.
[7] Y.Xia and M.S. Kamel, A Cooperative Recurrent Neural Network Algorithm For Parameter Estimation of Autoregressive Signals,*International Joint Conference on Neural Networks*,2516- 2522, 2006.
[8] - A. Benchabane, A. Bennia, F. Charif and A. Ahmed-Taleb, Multi-Dimensional Capon Spectral Estimation Using Zhang Neural Networks,*Multidimensional systems and signals processing*, 24,583-598, 2013.
[9] P.S.Stanimirović ,M. D.Petković,Gradient neural dynamics for solving matrix equations and their applications, *Neurocomputing*, 306, 200-212, 2018.
[10] P.S.Stanimirović ,M.D.Petković and D. Gerontitis ,Gradient neural network with nonlinear activation for computing inner inverses and the Drazin inverse, *Neural Process. Lett.* , 48, 109–133, 2018.
[11] P.S.Stanimirović ,M. D.Petković,Improved GNN Models for Constant Matrix Inversion, *Neural Process. Lett.*, doi.org/10.1007/s11063-019-10025-9, 2019.
[12] Y. Zhang, K. Chen and W. Ma, MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying equations,*proceedings of International Conference on Life System Modeling and Simulation*, 450-454, 2007.
[13] C. Yi and Y. Zhang, Analogue recurrent neural network for linear algebraic equation solving, *Electronics Letters*, 44(18), 1078-1079, 2008.
[14] L.Xia, W.Meng,R.Lu and L. Ding, Advances in Neural Networks, *Springer International Publishing Switzerland* ,444-451, 2015
[15] Y. Zhang, Z.H Chen and K. Chen, Convergence properties aalysis of gradient neural network for solving online liear equations, *Acta Automatica Sinica*, 35(8), 1136-1139,2009.
[16] V. Pan, editors, *Structured matrices and polynomials: Unified superfast algorithms*, Birkhauser, Boston ,2001.

ISSN: 2692-5079

[17] T. Kailath and A. H. Sayed, editors, *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, PA, 1999.

Abderrazak Benchabane received the B.Sc. degree in automatics from Annaba University, Algeria, in 1993, the M.Sc. degree in electronics from the University of Constantine, Algeria, in 2003. and the Ph.D. degree on the spectral estimation using neural networks at the University of Constantine in 2015. His research interests include signal processing and neural networks.

Fella Charif received the B.Sc. , M.Sc. and the Ph.D. degrees in electronics from the University of Biskra, Algeria, in 1994, 2006, and 2015 respectively. Her current research interests include signal and image and neural networks.