

Received March 28, 2020, accepted April 19, 2020, date of publication April 22, 2020, date of current version May 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989507

ADMM-Net: A Deep Learning Approach for Parameter Estimation of Chirp Signals Under Sub-Nyquist Sampling

HANNING SU¹, QINGLONG BAO¹, AND ZENGPING CHEN²

¹College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China

²College of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510000, China

Corresponding author: Qinglong Bao (cbpest@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61401489.

ABSTRACT Parameter estimation of chirp signals plays an important role in the field of radar countermeasures. Compressed sensing (CS) based sub-Nyquist sampling and parameter estimation methods alleviate the pressure on hardware systems to acquire and process chirp signals with large time-bandwidths. In this paper, a framework based on the fractional Fourier transform (FrFT) and alternating direction method of multipliers network (ADMM-Net) is proposed to realize chirp signal parameter estimation under sub-Nyquist sampling. The whole framework is composed of multiple parallel ADMM-Nets, where each ADMM-Net is defined over a data flow graph, which is derived from the iterative procedures of the ADMM algorithm for optimizing a CS-based p -order FrFT spectral estimation model. The chirp rate and central frequency of chirp signals are obtained through a two-dimensional search on the spectrum image output by the network group. Experiments demonstrate that the proposed ADMM-Net-based method can achieve higher estimation accuracy and computational efficiency at lower signal-to-noise ratios and sampling ratios than traditional CS methods. We also demonstrate that the proposed ADMM-Net-based framework has strong generalization ability for multi-component chirp signals. Furthermore, we further generalize ADMM-Net to GADMM-Net, in which the activation function is data-driven instead of model-driven. Experiments demonstrate that GADMM-Net significantly improves on the basic ADMM-Net and achieves higher spectral resolution with faster computation speed.

INDEX TERMS Alternating direction method of multipliers, chirp signals, compressed sensing, deep learning, fractional Fourier transform, parameter estimation, sub-Nyquist sampling, unrolling method.

I. INTRODUCTION

Linear frequency-modulated (LFM) signals, which are also called chirp signals, are widely employed in various applications, such as radar [1], sonar [2], ultrasonics [3], and telecommunication [4]. Accurate estimation of the parameters of chirp signals, i.e., central frequency f_0 and chirp rate k , is essential in passive detection technology [5]. However, in the field of radar countermeasures, chirp signals are usually accompanied by a large time-bandwidth, and the estimation of its modulation parameters requires an extremely high sampling rate, which puts tremendous strain on the hardware systems used for signal acquisition, transmission, and processing. Although some scholars have proposed the

method of under-sampling combined with dechirping [6], which reduces the sampling frequency requirement, there is also frequency ambiguity [7]. Therefore, how to find a new signal parameter estimation algorithm to reduce the pressure caused by large time-bandwidth is an urgent problem to be solved.

Compressed sensing (CS) has brought new ideas to chirp signal processing. Most applications of CS theory for signal processing are for accurate reconstruction of signals [8]–[11]. However, in the application of CS theory for signal parameter estimation, with the help of a specific parameter matching dictionary, the estimation task can be achieved without complete reconstruction of signals [12]–[17].

CS has been a popular technique to acquire a decomposition expression of the signal from sampled measurements by solving an under-determined linear system [18].

The associate editor coordinating the review of this manuscript and approving it for publication was Pietro Savazzi¹.

CS incorporates data compression by a measurement matrix $\Phi \in \mathbb{C}^{M \times N}$ ($M \ll N$) and the solution of the coefficient vector $\theta \in \mathbb{C}^N$ of signal $x \in \mathbb{C}^N$ on the sparse basis $\Psi \in \mathbb{C}^{N \times N}$ from measurements $y \in \mathbb{C}^M$ by a non-linear mapping f with lowest calculation error:

$$E = \inf_{\Phi, \Psi, f} \sup \|x - \Psi f(\Phi x)\|, \quad y = \Phi x, \quad x = \Psi \theta \quad (1)$$

The measurement matrix Φ , sparse basis Ψ , and non-linear mapping $f(\cdot)$ are three key factors in CS. In this work, we focus on learning a sparse basis and non-linear mapping.

A. MEASUREMENT MATRIX

The measurement matrix is commonly required to satisfy some conditions, such as small autocorrelation coefficients [19] or small restricted isometry constant (RIC) [20], [21], to avoid aliasing effects by under-sampling. Random Gaussian matrices, partial Fourier matrices, randomly permuted coded diffraction, and Walsh-Hadamard operators are popular measurement matrices for different practical tasks.

B. SPARSE BASIS

The traditional chirp signal sparse representation is to decompose the signal in some specific transformation domain and express it with the known basis function in the form of an atom-dictionary or transform matrix. Dictionary-based bases [12]–[15] usually have local sparsity, which can represent the local time-frequency information of chirp signals. However, as the super-resolution estimation of parameters greatly increases the number of atoms in the dictionary, a high computational complexity is required; besides, because of the sparsity of single-frequency signals and narrow-band chirp signals in the frequency domain, the inverse Fourier transform matrix [16] can be selected as the sparse basis, whereas it is not suitable for wide-band chirp signals. Based on the energy concentration property of chirp signals in the proper order Fractional Fourier transform (FrFT) domain [22], a sparse dictionary is proposed in Ref. [17], where the discrete inverse FrFT (DIFRFT) matrices of different orders are adopted as the atoms. The solution of θ under each atom is essentially a p -order FrFT spectrum estimation model, which provides access for the detection and parameter estimation of chirp signals based on the FrFT spectrum [23]. Besides, the scale of the DIFRFT dictionary is much smaller than that of the time-frequency dictionary.

C. NON-LINEAR RECONSTRUCTION MAPPING

The constrained standard CS model [18] is

$$\hat{\theta} = f(y) = \arg \min_{\theta} \frac{1}{2} \|\Phi \Psi \theta - y\|_2^2 + \mathcal{P}(\theta) \quad (2)$$

where $\mathcal{P}(\theta)$ is a penalty term enforcing the transform sparsity of the signal, which is a regularization function derived from the data prior, e.g., ℓ_q -norm ($0 \leq q \leq 1$) for a sparse prior.

A wide variety algorithms have been developed to optimize the model, such as convex relaxation algorithms based on

the ℓ_1 -norm [24], [25], greedy algorithms based on the ℓ_0 -norm [26], [27], and non-convex optimization algorithms [28], [29]. Although they are effective with many theoretical guarantees, most of them lack computational efficiency, and how to determine the numerous uncertainties in the model, such as regularization parameters and the parameters involved in the optimization algorithm, is still a challenging task.

Recently, deep neural network (DNN)-based supervised learning methods have shown success in signal parameter estimation, such as direction of arrival estimation [30], [31], and channel estimation [32], [33], where a DNN reconstructs a line spectral estimation model based on measurement datasets. The basic idea of these works is taking a DNN to directly model the non-linear mapping by learning network parameters from massive training data. However, although conventional data-driven DNNs are computationally efficient, they have restrictive receptive fields. They use conventional network architectures as black boxes without explicitly modeling or investigating the domain knowledge, such as transformation model or signal sparsity. Therefore, it is difficult for a conventional black box network to automatically learn the transform sparsity.

The unrolling method is a discriminative learning method that unfolds an iterative optimization algorithm to a hierarchical architecture. This category of methods links the conventional model-based approach to the deep learning approach. Model-based networks focus on learning a deep cascade of multiple sub-networks by introducing a data consistency (DC) layer. Each sub-network generates a prediction using the predicted result of the preceding sub-network. These unrolling networks are mainly used in the field of image reconstruction, e.g., [34] unfolded an AMP algorithm [11] for CS image recovery, where the denoising operator in each AMP iteration was replaced with a very deep CNN. Sun *et al.* [36] and Yang *et al.* [37] presented an ADMM-Net derived from the iterative procedure of alternating direction method of multipliers (ADMM) algorithm [37] to solve a CS-MRI problem. Grefor and LeCun [38] proposed learning an iterative soft thresholding algorithm (LISTA) to approximate t sparse codes of the ℓ_1 sparse regularization problem in fixed steps. Wang *et al.* [39] and Xin *et al.* [40] further unfolded the iterative hard thresholding algorithm to feed-forward neural networks for an ℓ_0 regularized sparse coding model to obtain a sparser solution.

Inspired by the excellent performances of unrolling methods in image reconstruction, we propose an effective model-based deep learning method for parameter estimation of chirp signals under sub-Nyquist sampling. The proposed method is based on a CS model of the ℓ_1 -norm with the DIFRFT dictionary as a sparse basis. Then, we design an ADMM iterative algorithm to solve the above CS model. By unrolling and the ADMM iterative procedures, we derive a corresponding version of ADMM-Net, and the FrFT of signals under each p -order inverse FrFT matrix are obtained. Finally, the optimal transformation order is obtained through

a two-dimensional search on the FrFT spectrum image, and estimations of the chirp rate and central frequency of the chirp signal are obtained.

Contributions: The main contributions of this work can be summarized as follows:

- 1) We propose a novel deep learning-based framework for chirp signal parameter estimation under sub-Nyquist sampling. The whole framework is composed of multiple parallel ADMM-Nets, where each ADMM-Net is defined over a data flow graph, which is derived from the iterative procedures in the ADMM algorithm for optimizing a CS-based p -order FrFT spectral estimation model.
- 2) Extensive experiments demonstrate that the proposed ADMM-Net-based method can achieve higher estimation accuracy and computational efficiency at lower signal-to-noise ratios (SNRs) and sampling ratios compared with traditional CS methods. We also demonstrate that the proposed ADMM-Net-based framework has strong generalization ability for multi-component chirp signals.
- 3) We further unroll the ADMM-Net to a generalized one (GADMM-Net), where the activation function is data-driven instead of model-driven. Experiments demonstrate that GADMM-Net significantly improves the basic ADMM-Net and achieves higher spectral resolution with higher computational speed.

The remainder of the paper consists of five parts: Section II introduces the sparse representation based on FrFT. Section III presents the details of unrolling ADMM. Section IV presents an application of the proposed framework in parameter estimation under sub-Nyquist sampling. Section V presents the results of simulations to evaluate the performance of the proposed framework with respect to conventional ones, and then, a GADMM-Net structure is proposed to improve the performance of the algorithm. Finally, Section VI concludes the paper.

II. SPARSE REPRESENTATION BASED ON FrFT

In this section, we first study the sparsity of chirp signals in the FrFT domain and then introduce the DIFRFT dictionary.

A. FrFT SPARSITY

A mono-component chirp signal $s(t)$ with amplitude 1 and duty ratio 1 can be expressed as

$$s(t) = \exp(j2\pi f_0 t + j\pi k t^2) \quad t \in [-T/2, T/2] \quad (3)$$

where T is pulse width. Referring to the Fourier transform expression, the FrFT expression of $s(t)$ can be defined as

$$S_p(u) = FrFT^p [s(t)] = \int_{-\infty}^{\infty} s(t) K_\alpha(t, \mu) dt \quad (4)$$

where p is the transformation order of the FrFT, α is the rotation angle of the signal on the time-frequency plane, $\alpha = p\pi/2$, and $FrFT[\cdot]$ is the FrFT operator; $K_\alpha(t, \mu)$

represents the kernel function of the FrFT, which is expressed as

$$K_\alpha(t, u) = \begin{cases} A_\alpha \exp[j\pi (\mu^2 \cot \alpha - 2\mu t \csc \alpha + t^2 \cot \alpha)], & \alpha \neq n\pi \\ \delta(t - \mu), & \alpha = 2n\pi \\ \delta(t + \mu), & \alpha = (2n \pm 1)\pi \end{cases} \quad (5)$$

where $A_\alpha = \sqrt{|1 - j \cot \alpha|}$. By substituting Eqn. (3) into Eqn. (5), the FrFT expression of the signal can be written as

$$S_p(u) = \int_{-\infty}^{+\infty} \exp(j2\pi f_0 t + \pi k t^2) K_\alpha(t, u) dt \quad (6)$$

Let rotation angle $\alpha = \alpha_0$, $\cot \alpha_0 = -k$, and $\alpha_0 = \frac{\pi p_0}{2}$; then, the FrFT expression of a chirp signal at rotation angle α_0 can be calculated as

$$\begin{aligned} S_{p_0}(u) &= FrFT^{p_0} [s(t)] \\ &= \int_{-\infty}^{+\infty} A_{\alpha_0} \exp[j\pi (u^2 \cot \alpha_0 - 2ut \csc \alpha_0 + t^2 \cot \alpha_0)] \\ &\quad \cdot \exp(j2\pi f_0 t + j\pi k t^2) dt \\ &= 2\pi A_{\alpha_0} \exp(-j\pi u^2 k) \\ &\quad \cdot \frac{1}{2\pi} \int_{-\infty}^{+\infty} \exp[-j2\pi (u \csc \alpha_0 - f_0) t] dt \\ &= A_{\alpha_0} \exp(-j\pi u^2 k) \delta(\csc \alpha_0 - f_0) \end{aligned} \quad (7)$$

Therefore, the FrFT of a chirp signal behaves as an impulse function when rotated at an angle of $\alpha = \alpha_0 = -\arccot(k)$, and the impulse peak position is $\mu = \mu_0 = f_0 \sin \alpha_0$. Accordingly, the estimation process of chirp rate k and central frequency f_0 of the chirp signal can be described as

$$\begin{cases} \hat{k} = -\cot(\alpha_0) \\ \hat{f}_0 = \mu_0 \csc(\alpha_0) \end{cases} \quad (8)$$

B. DIFRFT DICTIONARY

Sampling a chirp signal $s(t)$ by interval $T_N = 1/F_s$, we have $s[n] = s(nT_N)$, $n = 1, \dots, N$.

Pei [41] derived a sampling discretization-type DFRFT computation algorithm. Let $\Delta\mu$ denote the sampling interval in the FrFT domain. The sampling discretization-type DFRFT of $s[n]$ can be expressed as

$$S_p[n] = \sum_{n=-N}^N K_\alpha[n, n] s[n] \quad (9)$$

where the default length of $S_p[n]$ is N , and $K_\alpha[n, n]$ is defined as

$$K_\alpha[n, n] = \begin{cases} B_\alpha \exp[j\pi (n^2 \Delta u^2 \cot \alpha - 2n^2 \Delta u T_N \csc \alpha + n^2 T_N^2 \cot \alpha)] & \alpha \neq n\pi \\ \delta(nT_N - n\Delta u) & \alpha = 2n\pi \\ \delta(nT_N + n\Delta u) & \alpha = (2n \pm 1)\pi \end{cases} \quad (10)$$

where $B_\alpha = \sqrt{S(\sin \alpha - j \cos \alpha)/N}$ with $S = \text{sgn}(\sin \alpha)$. Eqn. (9) can be written in the form of a transformation matrix

$$\mathbf{S}_p = \mathbf{K}_\alpha \mathbf{s} \quad (11)$$

where

$$\begin{aligned} \mathbf{S}_p &= [S_p[1], \dots, S_p[N]]^T \\ \mathbf{s} &= [s[1], \dots, s[N]]^T \\ \mathbf{K}_\alpha &= \begin{bmatrix} K_\alpha[1, 1] & \dots & K_\alpha[1, N] \\ \vdots & \ddots & \vdots \\ K_\alpha[N, 1] & \dots & K_\alpha[N, N] \end{bmatrix} \end{aligned} \quad (12)$$

The kernel functions of the FrFT and inverse FrFT (iFrFT) have the following characteristics:

$$K_{-\alpha}(u, t) = K_\alpha^*(u, t) \quad (13)$$

$$\int_{-\infty}^{+\infty} K_\alpha(u, t) K_\alpha^*(u', t) dt = \delta(u - u') \quad (14)$$

Therefore, the iFrFT of a chirp signal can be understood as the linear expansion of chirp signal $s(t)$ in the space with the inverse transformation kernel $K_{-\alpha}(t, \mu)$ as the basis function:

$$\mathbf{s} = \mathbf{K}_\alpha^H \mathbf{S}_p = \mathbf{K}_{-\alpha} \mathbf{S}_p \quad (15)$$

where $\mathbf{K}_{-\alpha}$ is still an orthogonal basis for the μ domain. At this point, we can construct a dictionary based on \mathbf{K}_α^H , and obviously, in the case where N is determined, the complexity of the dictionary $O(n)$ is uniquely determined by the density of p , which is much smaller than the dictionary with parameter matching $O(n^r)$, $r \geq 2$.

In this work, we use the DIFRFT dictionary as the sparse basis for CS methods.

III. ADMM-NET FOR CS MODEL

Alternating direction method of multipliers (ADMM) [37] is a widely utilized variable splitting algorithm for solving the energy minimization problem. It adopts the augmented Lagrangian function of the given model and splits variables into sub-groups, which can be alternately optimized by solving a few simple subproblems.

A. ADMM SOLVER

Tibshirani [42] proposed the least absolute shrinkage and selection operator (LASSO) model in 1996, adding the ℓ_1 norm regularization term on the basis of the original fidelity term, which is more conducive to selecting the ideal solution. The LASSO model is

$$\hat{\theta} = \arg \min_{\theta} \left\{ \frac{1}{2} \|\Phi \Psi \theta - \mathbf{y}\|_2^2 + \lambda \|\theta\|_1 \right\} \quad (16)$$

where $\lambda > 0$ usually, and by adjusting λ , the sparsity of the sparse solution can be adjusted, and good noise suppression performance can be achieved.

In 2012, Boyd *et al.* [37] systematically proposed the operation steps to solve the LASSO problem based on the ADMM method and combined the dual decomposition and augmented Lagrange multi-multiplier methods to solve the

optimization problem step by step, and they improved the computational efficiency and obtained robust convergence performance. The advantage of the ADMM method is that a complex optimization problem is decomposed into several sub-problems, and the global performance can be harmonized while distributed optimization is realize to get closer to the optimal solution. According to ADMM algorithm theory, LASSO can be decomposed into two subproblems: $f(\theta) = \frac{1}{2} \|\Phi \Psi \theta - \mathbf{y}\|_2^2$ and $g(\mathbf{z}) = \lambda \|\mathbf{z}\|_1$, where $\Phi \Psi$ is the observation matrix \mathbf{A} .

The optimization problem can be solved efficiently by introducing auxiliary variables $\mathbf{z} = [z[1], z[2], \dots, z[N]]^T$ to the transform domains, making Eqn. (16) equivalent to

$$\min_{\theta} \{f(\theta) + g(\mathbf{z})\}, \quad \text{s.t. } \mathbf{z} = \theta. \quad (17)$$

Hence, an optimization equation with augmented Lagrange terms is established:

$$\mathcal{L}_\rho(\theta, \mathbf{z}, \alpha) = f(\theta) + g(\mathbf{z}) + \alpha^H (\theta - \mathbf{z}) + \frac{\rho}{2} \|\theta - \mathbf{z}\|_2^2 \quad (18)$$

where α represents Lagrangian multipliers and ρ is the Lagrange multiplier coefficient. The introduced Lagrange multiplier term adds an equality constraint to the optimization of Eqn. (19), and the objective solution is limited to the corresponding feasible region, which improves the solution efficiency, i.e., the robustness of the algorithm. In this case, the ADMM algorithm alternately optimizes $\{\theta, \mathbf{z}, \alpha\}$ by solving the following subproblems:

$$\begin{cases} \theta^{(k+1)} = \arg \min_{\theta} \{f(\theta) + \frac{\rho}{2} \|\theta - \mathbf{z}^{(k)} + \alpha^{(k)}\|_2^2\} \\ \mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z}} \{g(\mathbf{z}) + \frac{\rho}{2} \|\theta^{(k+1)} - \mathbf{z} + \alpha^{(k)}\|_2^2\} \\ \alpha^{(k+1)} = \alpha^{(k)} + \rho (\theta^{(k+1)} - \mathbf{z}^{(k+1)}) \end{cases} \quad (19)$$

where k denotes the k -th iteration. θ and \mathbf{z} are iteratively updated in alternate directions to achieve joint minimization, while the dual variable α is updated by the iteration values of θ and \mathbf{z} . In the solution process, the Lagrange multiplier acts on each of θ and \mathbf{z} and then indirectly acts on α through the iteration of θ and \mathbf{z} , thus improving the convergence rate and robustness. The following three steps for Eqn. (19) are explained as follows:

1) RIDGE REGRESSION

According to the first equation in Eqn. (19), the optimization problem of variable θ is equivalent to the joint minimization problem of the fidelity term and Lagrangian term, which can be understood as a ridge regression problem, i.e.,

$$\begin{aligned} \theta^{(k+1)} &= \arg \min_{\theta} \left\{ \frac{1}{2} \text{Tr} \left[(\mathbf{y} - \mathbf{A}\theta)^H (\mathbf{y} - \mathbf{A}\theta) \right] \right. \\ &\quad \left. + \frac{\rho}{2} \text{Tr} \left[\left(\theta - \mathbf{z}^{(k)} + \alpha^{(k)} \right)^H \left(\theta - \mathbf{z}^{(k)} + \alpha^{(k)} \right) \right] \right\} \end{aligned} \quad (20)$$

where $\text{Tr}[\cdot]$ represents the trace operator. It can be seen from Eqn. (20) that the minimized objective function is a convex

function, so the optimal value can be obtained by taking the derivative of the objective function

$$\boldsymbol{\theta}^{(k+1)} = (\mathbf{A}^H \mathbf{A} + \rho \mathbf{I})^{-1} [\mathbf{A}^H \mathbf{y} + \rho (\mathbf{z}^{(k)} - \boldsymbol{\alpha}^{(k)})] \quad (21)$$

2) ℓ_1 NORM REGULARIZATION

According to the second equation in Eqn. (19), the optimization problem of variable \mathbf{z} is equivalent to the regularization problem of the ℓ_1 norm, which is different from the LASSO problem of Eqn. (16), where the regularization is oriented to the augmented Lagrange term. The optimization problem for \mathbf{z} is the same as for $\boldsymbol{\theta}$:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \arg \min_{\mathbf{z}} \left\{ \lambda \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\boldsymbol{\theta}^{(k+1)} - \mathbf{z} + \boldsymbol{\alpha}^{(k)}\|_2^2 \right\} \\ &= S_{\frac{\lambda}{\rho}} \left(\boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right) \end{aligned} \quad (22)$$

where $S_{\frac{\lambda}{\rho}}(x_i)$ represents the soft thresholding of x_i , which is any element of \mathbf{x} . When $\mathbf{x} \subseteq \mathbb{R}$, we can obtain

$$\mathbf{z}^{(k+1)} = \max \left(\left| \boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right| - \frac{\lambda}{\rho}, 0 \right) \cdot \text{sign} \left(\boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right) \quad (23)$$

where $\text{sign}(\cdot)$ denotes the sign function. A schematic diagram of the soft threshold operator $S_{\frac{\lambda}{\rho}}(x_R)$ for a real value x_R is shown in Fig. 1(a). The real soft threshold can be regarded as a piecewise linear filter that zeroizes smaller values of $|x_R| \leq |\frac{\lambda}{\rho}|$ and linearly shrinks x_R when $|x_R| > |\frac{\lambda}{\rho}|$.

Based on the complex value characteristics of chirp signals in the time domain and transformation domain, we improve the existing real-valued soft threshold operator to its complex form [43], where any element z_i in the complex vector \mathbf{z} is decomposed into a real part $\Re(z_i)$ and imaginary part $\Im(z_i)$, and the same complex decomposition is performed for $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$. The complex form of Eqn. (23) is obtained by taking derivatives of $\Re(\mathbf{z})$ and $\Im(\mathbf{z})$:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \Re(\mathbf{z}^{(k+1)}) + j\Im(\mathbf{z}^{(k+1)}) \\ &= S_{\frac{\lambda}{\rho}} \left(\Re \left(\boldsymbol{\theta}^{(k+1)} \right) + \Re \left(\boldsymbol{\alpha}^{(k)} \right) \right) \\ &\quad + j S_{\frac{\lambda}{\rho}} \left(\Im \left(\boldsymbol{\theta}^{(k+1)} \right) + \Im \left(\boldsymbol{\alpha}^{(k)} \right) \right) \\ &= \max \left(\left| \boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right| - \frac{\lambda}{\rho}, 0 \right) \cdot \frac{\left(\boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right)}{\left| \boldsymbol{\theta}^{(k+1)} + \boldsymbol{\alpha}^{(k)} \right|} \end{aligned} \quad (24)$$

According to Eqn. (24), the complex soft threshold has the same phase retention as the real soft threshold. For convenience, the polar coordinate scheme of the complex x_C soft threshold operator $S_{\frac{\lambda}{\rho}}(x_C)$ is shown in Fig. 1(b).

In Fig.1(b), the magnitude of x_C is $|x_C|$, compound angle $\angle x_C \in [0, 2\pi)$, and the compound angle of complex $S_{\frac{\lambda}{\rho}}$ is also $\angle x_C$. $|x|$, and $\angle x$ are sufficient to represent all the values of x_C . The real-valued soft threshold can also be considered as a special case of the complex-valued soft threshold at $\angle x = 0$.

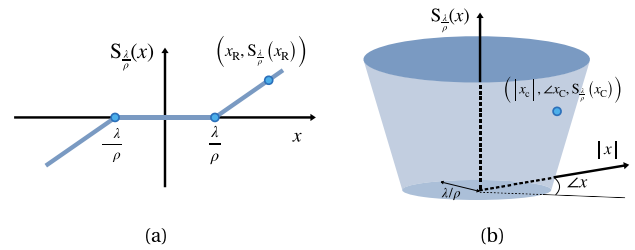


FIGURE 1. (a) Real-valued soft threshold operator and (b) complex-valued soft threshold operator.

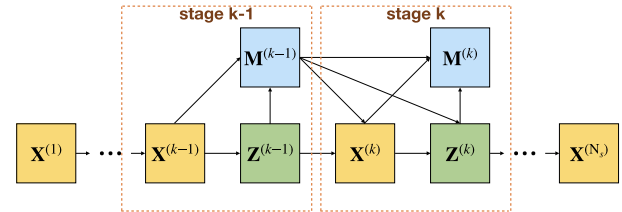


FIGURE 2. Data flow graph of an ADMM-Net.

3) DUAL VARIABLE UPDATE

The updating of $\boldsymbol{\theta}$ and \mathbf{z} is the process of minimizing their combined values. In the complex-valued ADMM algorithm, the iteration of $\boldsymbol{\alpha}$ is also the update process of the joint dual variables of $\boldsymbol{\theta}$ and \mathbf{z} , i.e.,

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \rho \left(\boldsymbol{\theta}^{(k+1)} - \mathbf{z}^{(k+1)} \right) \quad (25)$$

The process is iterated through Gauss-Seidel until the stop criterion is met, i.e., the set number of iterations or accuracy $\|\mathbf{r}^{(k)}\|_2 \leq \varepsilon$ is reached, where $\mathbf{r}^{(k)} = \boldsymbol{\theta}^{(k)} - \mathbf{z}^{(k)}$ denotes the main residual, ε is the precision threshold set based on experience, and the algorithm terminates when the stop criterion is met.

B. ADMM-NET

To design our deep ADMM-Net, we first map the ADMM iterative procedures in Eqn. (19) to a data flow graph. As shown in Fig. 2, this graph is comprised of nodes corresponding to different operations in ADMM, and directed edges correspond to the data flows between operations. In this case, the k -th stage of the data flow graph corresponds to the $(N_s + k)$ -th iteration of the ADMM algorithm, where N_s is the number of stages. The whole data flow graph is multiple repetitions of the above stages corresponding to successive iterations in ADMM. In the k -th stage of the graph, there are three types of nodes mapped from three types of operations in ADMM, i.e., coefficient solving layer ($\mathbf{X}^{(k)}$), nonlinear transform layer ($\mathbf{Z}^{(k)}$), and multiplier update layer ($\mathbf{M}^{(k)}$) in Eqn. (19). Given an under-sampled signal, it flows over the graph and finally outputs a spectrum.

1) COEFFICIENT SOLVING LAYER $\mathbf{X}^{(k)}$

This layer solves the sparse vector $\boldsymbol{\theta}^{(k)}$ according to Eqn. (21). Given $\boldsymbol{\alpha}^{(k-1)}$ and $\mathbf{z}^{(k-1)}$, which are outputs of previous layers

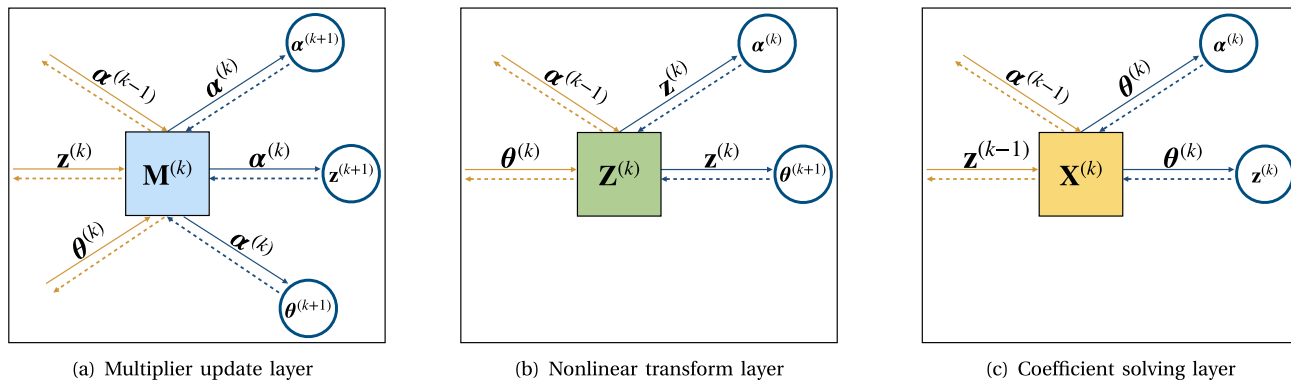


FIGURE 3. Illustrations of three types of graph nodes (i.e., layers) and their data flows in the k -th stage. The rectangular box represents the layer of interest, and the circles represent the layers that have data communication (i.e., connections) with that layer. The solid arrow indicates the data flow in the forward pass and dashed arrow indicates the backward flow in the backward pass when computing gradients in back-propagation.

in stage $k - 1$, the output of this layer is defined as

$$\theta^{(k)} = \left(\mathbf{A}^H \mathbf{A} + \rho^{(k)} \mathbf{I} \right)^{-1} \left(\rho^{(k)} \mathbf{z}^{(k-1)} + \mathbf{A}^H \mathbf{y} - \rho^{(k)} \alpha^{(k-1)} \right) \quad (26)$$

where \mathbf{y} represents the input measurements and $\rho^{(k)}$ is a learnable penalty parameter in the k -th stage. We do not constrain the parameters to be the same in different stages to increase the network capacity. The output $\theta^{(k)}$ in this layer is the input for subsequent multiplier update layer ($\alpha^{(k)}$) and nonlinear transform layer ($\mathbf{z}^{(k)}$) in stage k . In the first stage (i.e., $k = 1$), $\theta^{(1)} = \left(\mathbf{A}^H \mathbf{A} + \rho^{(1)} \mathbf{I} \right)^{-1} \left(\mathbf{A}^H \mathbf{y} \right)$.

2) NONLINEAR TRANSFORM LAYER $\mathbf{z}^{(k)}$

This layer performs a nonlinear transform inspired by a complex-valued soft threshold operator $S_\tau(\cdot)$. Given $\theta^{(k)}$ and $\alpha^{(k-1)}$, the output of this layer is defined as

$$\mathbf{z}^{(k)} = \max \left(\left| \theta^{(k)} + \alpha^{(k-1)} \right| - \tau^{(k)}, 0 \right) \cdot \frac{\left(\theta^{(k)} + \alpha^{(k-1)} \right)}{\left| \theta^{(k)} + \alpha^{(k-1)} \right|} \quad (27)$$

where $\tau^{(k)}$ is a learnable threshold parameter in the k -th stage.

3) MULTIPLIER UPDATE LAYER $\mathbf{M}^{(k)}$

This layer is defined by the Lagrangian multiplier updating procedure $\alpha^{(k)}$ in Eqn. (25). Given the three inputs $\alpha^{(k-1)}$, $\theta^{(k)}$, and $\mathbf{z}^{(k)}$, the output of this layer in stage k is defined as

$$\alpha^{(k)} = \eta_1^{(k)} \alpha^{(k-1)} + \eta_2^{(k)} \theta^{(k)} - \eta_3^{(k)} \mathbf{z}^{(k)} \quad (28)$$

where $\eta_1^{(k)}$, $\eta_2^{(k)}$, and $\eta_3^{(k)}$ are learnable update rate parameters in the k -th stage.

4) NETWORK PARAMETERS

These layers are organized in the data flow graph shown in Fig. 2. In the deep architecture, we aim to learn the following parameters: $\rho^{(k)}$ in the coefficient solving layer, $\tau^{(k)}$ in the nonlinear transform layer, $\eta_1^{(k)}$, $\eta_2^{(k)}$, and $\eta_3^{(k)}$ in the multiplier update layer, where $k = 1, 2, \dots, N_s$ are the indexes for

each stage. All of these parameters are taken as the network parameters to be learned. Obviously, the data set size of the CS model-based network is much smaller than that of the data-driven network.

C. NETWORK TRAINING

1) LOSS FUNCTION

The sparse vectors from fully sampled data are taken as the ground-truth labels θ^{gt} and the under-sampled measurement data \mathbf{y} as the input. Then, a training set with size Γ is constructed containing pairs of under-sampled data and ground-truth labels. We choose averaged normalized root mean square error (NRMSE) as the loss function to train the networks. Given pairs of training data, the loss between the network output and ground-truth is defined as

$$E(\Theta) = \frac{1}{|\Gamma|} \sum_{(\mathbf{y}, \theta^{gt}) \in \Gamma} \frac{\sqrt{\left\| \hat{\theta}(\mathbf{y}, \Theta) - \theta^{gt} \right\|_2^2}}{\sqrt{\left\| \theta^{gt} \right\|_2^2}} \quad (29)$$

where $\hat{\theta}(\mathbf{y}, \Theta)$ is the network output based on network parameter Θ and under-sampled measurement data \mathbf{y} . We learn the parameters $\Theta = \left\{ \rho^{(k)}, \tau^{(k)}, \eta_1^{(k)}, \eta_2^{(k)}, \eta_3^{(k)} \right\}_{k=1}^{N_s-1} \cup \left\{ \rho^{(N_s)} \right\}$ by minimizing the loss w.r.t. them using L-BFGS [44] and learn the gradients of the loss function $E(\Theta)$ w.r.t. parameters Θ using backpropagation over the network.

2) BACKWARD PROPAGATION

In the forward pass, the data of the k -th stage is processed in the order of $\mathbf{X}^{(k)}$, $\mathbf{Z}^{(k)}$, and $\mathbf{M}^{(k)}$. In the backward pass, the gradients are computed in an inverse order. For stage k , Fig. 3 shows three types of nodes (i.e., network layers) and the data flow over them. Each node has multiple inputs and (or) outputs. We next introduce the gradient computation for each layer in a typical stage k ($k < N_s$).

a: MULTIPLIER UPDATE LAYER $\mathbf{M}^{(k)}$

As shown in Fig. 3(a), this layer has three sets of inputs: $\alpha^{(k-1)}$, $\mathbf{z}^{(k)}$, and $\theta^{(k)}$. Its output $\alpha^{(k)}$ is the input to compute

$\alpha^{(kC1)}$, $\mathbf{z}^{(k+1)}$, and $\theta^{(kC1)}$. The parameters of this layer are $\eta_1^{(k)}$, $\eta_2^{(k)}$, and $\eta_3^{(k)}$. The gradients of loss w.r.t. the parameters can be computed as

$$\frac{\partial E}{\partial \eta_l^{(k)}} = \frac{\partial E}{\partial \alpha^{(k)}} \frac{\partial \alpha^{(k)}}{\partial \eta_l^{(k)}}, \quad (30)$$

where $l = 1, 2, 3$, and

$$\frac{\partial E}{\partial \alpha^{(k)}} = \begin{cases} \frac{\partial E}{\partial \alpha^{(k+1)}} \frac{\partial \alpha^{(k+1)}}{\partial \alpha^{(k)}} + \frac{\partial E}{\partial \mathbf{z}^{(k+1)}} \frac{\partial \mathbf{z}^{(k+1)}}{\partial \alpha^{(k)}} \\ + \frac{\partial E}{\partial \theta^{(k+1)}} \frac{\partial \theta^{(k+1)}}{\partial \alpha^{(k)}}, & k \leq N_s - 2 \\ \frac{\partial E}{\partial \alpha^{(k)}} = \frac{\partial E}{\partial \theta^{(k+1)}} \frac{\partial \theta^{(k+1)}}{\partial \alpha^{(k)}}, & k = N_s - 1 \end{cases} \quad (31)$$

where $\frac{\partial E}{\partial \alpha^{(k)}}$ is the summation of gradients along the three dashed blue arrows in Fig. 3(a). The gradients of the output in this layer w.r.t. its inputs, $\frac{\partial \alpha^{(k)}}{\partial \alpha^{(k-1)}}$, $\frac{\partial \alpha^{(k)}}{\partial \mathbf{z}^{(k)}}$, $\frac{\partial \alpha^{(k)}}{\partial \theta^{(k)}}$, and parameters, $\frac{\partial \alpha^{(k)}}{\partial \eta_l^{(k)}}$, can be computed from the partial derivative of Eqn. (28).

b: NONLINEAR TRANSFORM LAYER $\mathbf{z}^{(k)}$

As shown in Fig. 3(b), this layer has two sets of inputs: $\alpha^{(k-1)}$ and $\theta^{(k)}$. Its output $\mathbf{z}^{(k)}$ is the input for computing $\alpha^{(k)}$ and $\theta^{(k+1)}$ in the next stage. The parameter of this layer is $\tau^{(k)}$. The gradient of loss w.r.t. parameters can be computed as

$$\frac{\partial E}{\partial \tau^{(k)}} = \frac{\partial E}{\partial \mathbf{z}^{(k)}} \frac{\partial \mathbf{z}^{(k)}}{\partial \tau^{(k)}}, \quad (32)$$

where

$$\frac{\partial E}{\partial \mathbf{z}^{(k)}} = \frac{\partial E}{\partial \alpha^{(k)}} \frac{\partial \alpha^{(k)}}{\partial \mathbf{z}^{(k)}} + \frac{\partial E}{\partial \theta^{(k+1)}} \frac{\partial \theta^{(k+1)}}{\partial \mathbf{z}^{(k)}}, \quad k \leq N_s - 1 \quad (33)$$

Similarly, we can also compute the gradients of layer output to its inputs, $\frac{\partial \mathbf{z}^{(k)}}{\partial \alpha^{(k-1)}}$, $\frac{\partial \mathbf{z}^{(k)}}{\partial \theta^{(k)}}$, and parameters $\frac{\partial \mathbf{z}^{(k)}}{\partial \tau^{(k)}}$ from Eqn. (27).

c: COEFFICIENT SOLVING LAYER $\mathbf{x}^{(k)}$

As shown in Fig. 3(c), this layer has two sets of inputs: $\mathbf{z}^{(k-1)}$ and $\alpha^{(k-1)}$, and the output $\theta^{(k)}$ is the input to compute $\mathbf{z}^{(k)}$ and $\alpha^{(k)}$ in the subsequent layer. The parameter of this layer is $\rho^{(k)}$. The gradients of loss w.r.t. parameters are computed as

$$\frac{\partial E}{\partial \rho^{(k)}} = \frac{\partial E}{\partial \theta^{(k)}} \frac{\partial \theta^{(k)}}{\partial \rho^{(k)}}, \quad (34)$$

where

$$\frac{\partial E}{\partial \theta^{(k)}} = \frac{\partial E}{\partial \alpha^{(k)}} \frac{\partial \alpha^{(k)}}{\partial \theta^{(k)}} + \frac{\partial E}{\partial \mathbf{z}^{(k)}} \frac{\partial \mathbf{z}^{(k)}}{\partial \theta^{(k)}}, \quad k \leq N_s - 1 \quad (35)$$

Similarly, we can also compute the gradients of a layer output to its inputs, $\frac{\partial \theta^{(k)}}{\partial \alpha^{(k-1)}}$, $\frac{\partial \theta^{(k)}}{\partial \mathbf{z}^{(k-1)}}$, and parameters $\frac{\partial \theta^{(k)}}{\partial \rho^{(k)}}$ from Eqn. (26).

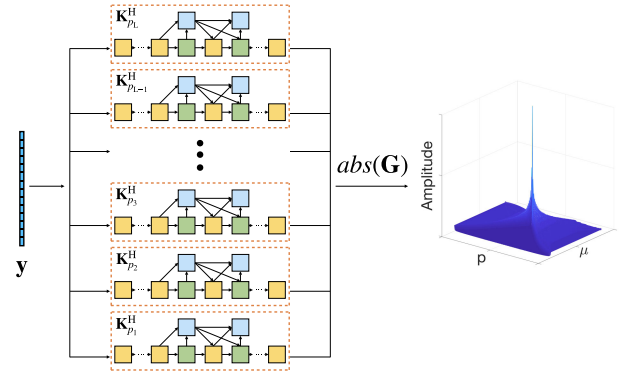


FIGURE 4. Data flow of the framework.

IV. ADMM-NET APPLIED TO CHIRP SIGNAL PARAMETER ESTIMATION

In this section, we use ADMM-Net to output the FrFT spectrum of chirp signals under sub-Nyquist sampling. First, the data flow diagram of the whole framework is introduced, and then the training details of ADMM-Net are introduced.

A. DATA FLOW

Eqn. (15) shows the decomposition form of a chirp signal on the p -order DIFRFT matrix $\mathbf{K}_p^H(\mathbf{K}_p^H)$, i.e., $\mathbf{s} = \mathbf{K}_p^H \mathbf{S}_p$. Therefore, we can obtain a p -order FrFT spectrum estimation model based on CS:

$$\hat{\mathbf{S}}_p = \arg \min_{\mathbf{S}_p} \left\{ \frac{1}{2} \|\Phi \mathbf{K}_p^H \mathbf{S}_p - \mathbf{y}\|_2^2 + \lambda \|\mathbf{S}_p\|_1 \right\}, \quad (36)$$

where Φ is a random Gaussian matrix. Obviously, the essence of the above CS model based on the p -order DIFRFT matrix is to obtain the p -order FrFT spectrum of the sequence under fully-Nyquist sampling through the measurement sequence under sub-Nyquist sampling. Therefore, each $\hat{\mathbf{S}}_p$ under \mathbf{K}_p^H must be solved by a p -order ADMM-Net, $p = p_1, p_2, \dots, p_L$. Then, a network framework consisting of L parallel ADMM-Nets sharing input \mathbf{y} is obtained. The data flow diagram based on ADMM-Nets is shown in Fig. 4. All $\hat{\mathbf{S}}_p$ constitute the spectrum image of $(p, \mu) \in \mathbf{G} = [\hat{\mathbf{S}}_{p_1}, \dots, \hat{\mathbf{S}}_{p_L}]^T$, $\mathbf{G} \in \mathbb{C}^{L \times N}$, and then, a two-dimensional extreme value search is conducted on $abs(\mathbf{G})$. The coordinate information of the peak can be substituted into Eqn. (8) to obtain the parameters of the reference chirp signal.

Different from the ADMM-Net [36] used for MRI image recovery, the ADMM-Net in our work is used to solve θ rather than restore \mathbf{x} . In [36], ADMM-Net is further generalized, where sparse bases are learned by neural networks through data. This is because the contour of the reconstructed image is known; however, in parameter estimation, it is difficult to highlight the parameter characteristics of the signal without defining a standard without a known transformation domain. Instead, we use the predefined FrFT spectrum estimation model (Eqn. (36)) to complete the task of parameter estimation. Besides, formally, though, the output of ADMM-Net in

our work looks like an image, it is a matrix of coefficient vectors $G = [\hat{S}_{p_1}, \dots, \hat{S}_{p_L}]^T$, which is completely different from an RGB image. Therefore, what we propose is in fact a “network group” structure as shown in Fig. 4, in which each sub ADMM-Net is used to solve a \hat{S}_p .

The structure of the network framework is determined by parameters $\{N, M, N_s, L\}$. Obviously, $\{N, L\}$ affects the resolution of the output spectrum image, M represents the sampling ratio, and N_s reflects the depth of the network. Because the resolution of the spectral image is determined by the properties of the FrFT, which represents the upper limit of the estimation accuracy, we fix $\{N, L\}$ as a constant and only set $\{M, N_s\}$ as variables in the following sections.

B. NETWORK TRAINING

In this subsection, we train a basic ADMM-Net to realize parameter estimation of chirp signals.

1) DATA SETS

Set $N = 512, L = 200$, fully sampled frequency $F_s = 2\text{ MHz}$, and construct the DIFRFT matrix \mathbf{K}_p^H with step size 0.01 in the interval $[0.01, 2]$.

We take the $S_p[n]$ from fully sampled $s[n]$ as the label and the under-sampled measurement data $\mathbf{y}[m] = \Phi s[n]$ as the input. Then, a training data set is constructed containing pairs of $\mathbf{y}[m]$ and $S_p[n]$.

The $\mathbf{y}[m]$ in the data sets is taken from the mono-component chirp signals $s[n]$ within the specified frequency range. For each chirp signal s of the training example, the central frequency f_0 is in the range of 10 KHz to 1000 KHz with intervals of $\Delta f_0 = 10\text{ KHz}$, while the chirp rate k is in the range of 0 to 3.9 GHz/s with intervals of $\Delta k = 0.1\text{ GHz/s}$. The testing examples have intermediate values of parameters, i.e., parameters are separated from values in the training examples by $0.5 \cdot \Delta$. By not having overlapping values in the training and testing sets, one can ensure that the network is not overfitting, i.e., memorizing only the inputs shown to it without learning to generalize to new inputs. Furthermore, to satisfy Nyquist’s sampling law, a constraint was added to $\{f_0, k\}$: $\frac{kT}{2} \leq f_0 \leq \frac{F_s}{2} - \frac{kT}{2}$. The number of parameter templates that satisfy this constraint is 1904 for training and 1954 for testing. The distribution of the modulation parameter of $s[n]$ for 3858 templates from the training and testing examples is shown in Fig. 5.

A sequence of measurement data $\mathbf{y}[m]$ is generated under each training template, which constitutes the final training set, containing all 1,904 sets of reference parameters. Accordingly, the testing set consists of 1,954 templates, of which 10% are used as the validation set.

Note that all data sets are noiseless. Considering that Gaussian white noise does not show energy accumulation in any FrFT domain, we can obtain the output with excellent noise resistance without adding additional noise resistance processing to the data sets or network.

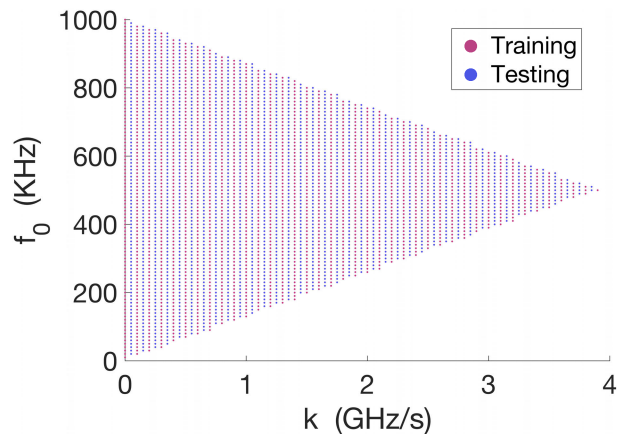


FIGURE 5. Parameter distribution of $s[n]$.

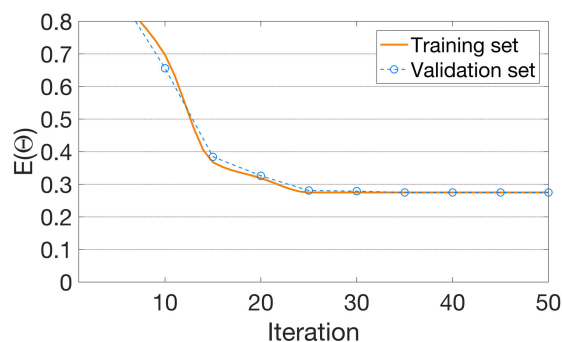


FIGURE 6. $E(\Theta^{(1.20)})$ for increasing L-BFGS iterations.

2) TRAINING

We learn the parameters $\{\Theta^{(p_1)}, \dots, \Theta^{(p_L)}\}$ of the parallel ADMM-Nets group by minimizing the loss $E(\Theta^{(p)})$ using gradient-based algorithm L-BFGS. We implement the ADMM-Nets using Tensorflow, and all experiments were conducted in Python (3.7) on a macbook with an 2.6 GHz Intel Core i7 CPU and a GTX1080 GPU. It should be noted that M and N_s are predefined and determine the structure of the parallel ADMM-Nets group. Once M and N_s are determined, these networks have similar loss curves. Fig. 6 shows the loss curve with L-BFGS iterations of an ADMM-Net corresponding to $p = 1.20$ under the settings of 30% sampling ratio, i.e., $M = 153$, and $N_s = 7$. It is satisfactory that $E(\Theta^{(p)})$ converges to the ideal value after 22 L-BFGS iterations.

V. SIMULATION AND RESULTS

In this section, we report several quantitative experiments on the estimation performance of the proposed framework. Based on the DIFRFT dictionary, we first compare the sensitivity of our deep ADMM-Net with conventional CS methods in terms of sampling ratio and SNR. Then, we further study the generalization ability of the framework on multi-component chirp signals. Finally, we observe the

influence of different network structures on the results and propose a more generalized network model.

A. PERFORMANCE COMPARISON

In this subsection, we compare the sensitivity of our deep ADMM-Net with conventional CS methods in terms of sampling ratio and SNR. Conventional methods include convex relaxation algorithms based on ℓ_1 , basis pursuit denosing (BPDN) [24], fast iterative shrinkage thresholding (FIST) [25], greedy algorithms, orthogonal matching Pursuit (OMP) [26], and iterative hard thresholding (IHT) [27] and non-convex optimization algorithm, iterative reweighted least squares (IRLS) [28], Bayesian compressive sensing (BCS) [29]. In the following experiment, we continue with the ADMM-Nets group trained in Section IV-B. The DIFRFT dictionary is used as the sparse basis $\Psi \in \mathbb{C}^{N \times N}$, and the random Gaussian matrix is used as the measurement matrix $\Phi \in \mathbb{C}^{M \times N}$ for all CS methods in this work.

The experimental sample in **Experiment 1** is a discrete complex-valued mono-component chirp sequence with white Gaussian noise with signal length $N = 512$, sampling frequency $F_s = 2 \text{ MHz}$, chirp rate $k = 1.17 \text{ GHz/s}$, and central frequency $f_0 = 0.505 \text{ MHz}$. Under different sampling ratios and SNRs, the above CS methods are used to solve the coefficient vectors $\theta^{(p)} \in \mathbb{C}^N$ of the signal under each $\mathbf{K}_p^H \in \mathbb{C}^{N \times N}$. Then, the coordinates of the peak can be obtained by two-dimensional peak search on $\mathbf{G} = [\theta^{(p_1)}, \dots, \theta^{(p_L)}]^T$, $L = 200$, $p = 0.01 : 0.01 : 2$, and \hat{k} and \hat{f}_0 are obtained through Eqn. (8). The evaluation criterion is the mean absolute percentage error as follows:

$$\delta_\varepsilon = \frac{100}{M_t} \sum_1^{M_t} \frac{|\hat{\varepsilon} - \varepsilon|}{\varepsilon}, \quad \varepsilon \in \{f_0, k\} \quad (37)$$

where M_t denotes the trials of a Monte Carlo simulation (MCS).

Experiment 1: We first compare the sensitivity of our deep ADMM-Nets group with conventional CS methods in terms of sampling ratio under $\text{SNR} = 10 \text{ dB}$. The compression sampling ratio was changed from 5% to 30% in steps of 5%. An MCS with 200 trials is conducted at each sampling ratio. Because parameter estimation based on the DIFRFT dictionary is a spectral peak estimation, there is a synchronous detection threshold before the peak is overwhelmed by clutter. Thus, the estimation error of \hat{k} and \hat{f}_0 of a mono-component chirp signal has the same variable threshold. Therefore, we can compare the performance of these CS methods only through the performance of δ_k .

Fig. 7(a) shows the quantitative results of the CS methods under different sampling ratios with $\text{SNR} = 10 \text{ dB}$, and Tab. 1 shows the average time taken by these CS methods to complete a trial.

The results show that when the sampling ratio is greater than or equal to 20%, all CS methods used in this work can achieve the estimation accuracy of the fully sampled FrFT.

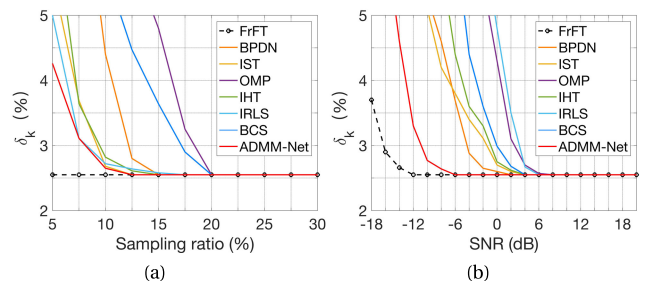


FIGURE 7. (a) δ_k curves with sampling ratio and (b) δ_k curves with SNR. The dotted line represents the estimated accuracy of the fully sampled FrFT, which is used as a baseline reference.

TABLE 1. Average runtime of a trial.

Method	Time Consumed	
	CPU	GPU
BPDN	11.13 s	--
IST	8.25 s	--
OMP	12.67 s	--
IHT	8.61 s	--
IRLS	4.03 s	--
BCS	16.39 s	--
ADMM-Net	1.22 s	0.45 s

The main difference of these error curves is that they have different sampling ratio thresholds, which directly reflect the resolution of the output FrFT spectrum image. Compared with the conventional methods, our proposed ADMM-Net produces the highest estimation accuracy under all sampling ratios. In average, ADMM-Net achieves comparable estimation accuracy (FrFT baseline) using sampling ratios in the range of 2.5% ~ 7.5% less than conventional methods. Further, it achieves an approximate 20 times speed up in running time (GPU time) compared with IST and IHT. Under a 15% sampling ratio, ADMM-Net outperforms BCS by 1% and runs more than 36 times faster.

Further, we also compare the sensitivity of our deep ADMM-Net with conventional CS methods in terms of SNR. Similarly, the above sample is still used for the experiment. SNR is increased from -18 dB to 20 dB in steps of 2 dB, and an MCS with 200 trials is performed for each SNR. Fig. 7(b) shows the quantitative results of the CS methods under different SNRs, where the sampling ratio is 30%.

The results show that when $\text{SNR} \geq 6 \text{ dB}$, all CS methods used in this work can achieve the estimation accuracy of the fully sampled FrFT. ADMM-Net has the best noise resistance compared with other CS methods, which is embodied in its lowest SNR threshold. With the comparable estimation accuracy (FrFT base-line), ADMM-Net is 6 dB ~ 12 dB better than BPDN and other conventional CS methods. We conclude that even in the training phase of the network, the data sets are noiseless, but this does not affect the noise resistance of the network, which is determined by the properties of the FrFT. This gives CS model-driven networks a natural advantage over data-driven networks.

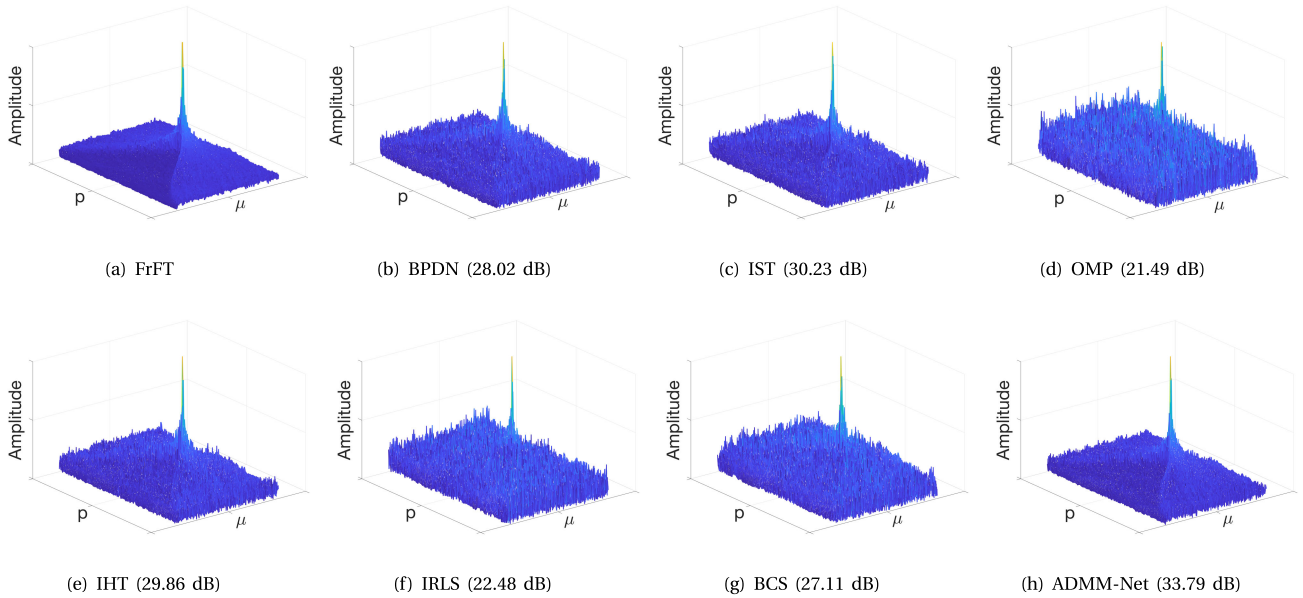


FIGURE 8. Examples of reconstructed FrFT spectrum images on the sample chirp signal using 30% sampling ratio and SNR=10 dB. (a) FrFT spectrum image under full sampling. (b)-(h) Reconstructed FrFT spectrum images based on BPDN, IST, OMP, IHT, IRLS, BCS, and ADMM-Nets. Their PSNRs are 28.02 dB, 30.23 dB, 21.49 dB, 29.86 dB, 22.48 dB, 27.11 dB, and 33.79 dB, respectively.

The visual comparisons of reconstructed $\hat{\mathbf{G}} = [\hat{\mathbf{S}}_{p_1}, \dots, \hat{\mathbf{S}}_{p_L}]^T$ using 30% sampling ratio and SNR = 10 dB are shown in Fig. 8. Peak SNR (PSNR) is introduced to represent the quality of $\hat{\mathbf{G}}$:

$$PSNR(\hat{\mathbf{G}}) = 10 * \log \left(\frac{MAX_{\hat{\mathbf{G}}}^2}{MSE} \right) \quad (38)$$

$$MSE = \frac{1}{L \times N} \sum_{i=0}^{L-1} \sum_{j=0}^{N-1} \left\| \hat{\mathbf{G}}(i, j) - \mathbf{G}(i, j) \right\|^2 \quad (39)$$

where $MAX_{\hat{\mathbf{G}}}$ indicates the maximum magnitude of the elements in $\hat{\mathbf{G}}$.

The visualization results show that ADMM-Net has the highest FrFT spectrum image resolution.

B. DISCUSSION

In the above experiment, ADMM-Net produced the best results for various SNRs and sampling ratios. In this subsection, we further discuss the generalization ability of ADMM-Net to multi-component chirp signals. Then, the effects of different network architectures are evaluated, and a generalized ADMM-Net (GADMM-Net) with improved performance is proposed.

1) GENERALIZATION ABILITY

One advantage of model-driven networks is their ability to generalize. In previous experiments, the proposed ADMM-Net was fed with a mono-component chirp signal dataset. Therefore, we test the generalization ability of ADMM-Net by testing the learned networks group from mono-component data to multi-component data.

TABLE 2. Performance of ADMM-Nets for estimating a three-component chirp signal.

Signal	Parameters						Threshold	
	k (GHZ/s)	\hat{k} (GHZ/s)	δ_k (%)	f_0 (MHz)	\hat{f}_0 (MHz)	δ_{f_0} (%)	Sampling ratio (%)	SNR (dB)
s_1	2.5391	2.5384	0.03	0.5050	0.5802	0.64	10.0	-8
s_2	1.5625	1.5867	1.55	0.7125	0.7135	0.13	12.5	-6
s_3	0.7812	0.7620	2.47	0.4525	0.4541	0.34	10.0	-8

Experiment 2 A three-component chirp signal is used as the input of the trained networks group, and the SNRs and sampling ratios are used as the variables to conduct quantitative experiments. An MCS with 200 trials under each variable unit is performed. Tab. 2 shows the results of the three-component chirp signal as the input of the network. In the experiment, we observed the robust estimation error of each chirp component and recorded their SNR and sampling ratio mutation thresholds.

Obviously, ADMM-Net has strong generalization ability for the three-component chirp signal, and the spectral peak of each chirp component in the plane (p, μ) can be accurately detected. The difference in the estimation accuracy between components is caused by the difference in modulation parameters, which is attributed to Eqn. (8). The threshold value is nearly the same as that of the mono-component chirp signal in **Experiment 1**, which indicates that ADMM-Net still maintains high noise resistance and information recovery ability to multi-component signals.

Fig. 9 shows the visualization results of the three-component chirp signal processed by BPDN and ADMM-Net when the sampling ratio is 30% and SNR is 10 dB.

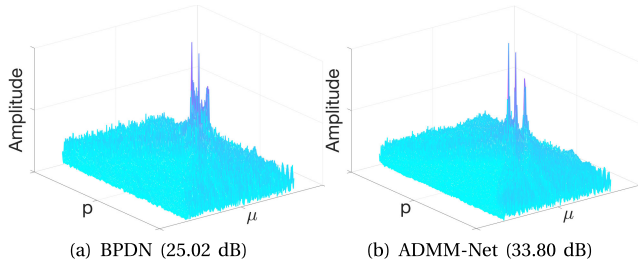


FIGURE 9. Visualization results of three-component chirp signals processed by (a) BPDN and (b) ADMM-Nets. Their PSNRs are 25.02 dB and 33.80 dB, respectively.

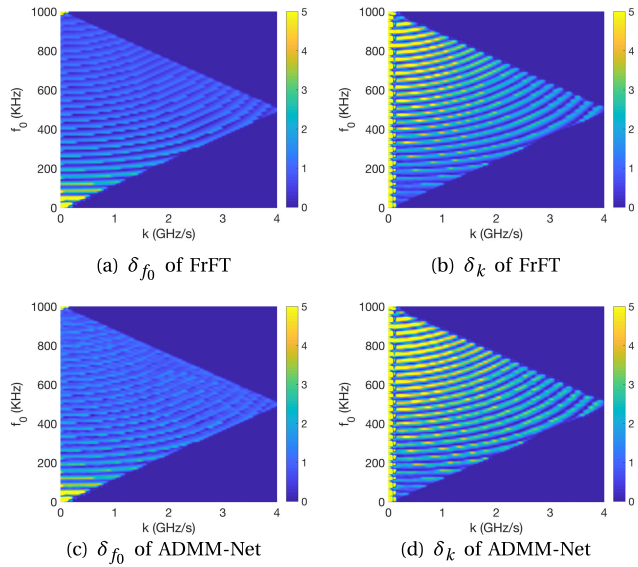


FIGURE 10. Sensitivity of FrFT and ADMM-Net to chirp signal modulated parameters.

The results clearly show that in the (p, μ) plane based on BPDN, there is strong pseudo-spectral interference between components, while the spectral peaks of the (p, μ) plane based on ADMM-Net are cleaner, indicating that ADMM-Net has better spectral resolution than BPDN.

Experiment 3 To verify the effectiveness of the method, an experiment on the sensitivity of modulation parameters was conducted to observe the performance of ADMM-Net on other modulation parameters. Under each test template (the red dot shown in Fig. 5, a mono-component chirp signal with SNR = 10dB was generated as a sample. All the samples made up the sample set. Then, each sample was processed in an MCS with 200 trials using FrFT under full sampling and an ADMM-Net with 30% sampling ratio.

We continue to use mean absolute percentage error as the degree of deviation:

$$\delta_\varepsilon = \frac{100}{M_t} \sum_1^{M_t} \frac{|\hat{\varepsilon} - \varepsilon|}{\varepsilon}, \quad \varepsilon \in \{f_0, k\} \quad (40)$$

Fig. 10 shows the error matrix obtained by FrFT and ADMM-Net. The triangular region represents the parameter

distribution of the sample set, which is same as the testing set's parameter distribution in Fig. 5. Different colors represent the magnitude of δ_ε , whose upper bound shown in the figure is 5%, i.e., for each point $\delta_\varepsilon = 5\%$ in the figure, its true $\delta_\varepsilon \geq 5\%$. Fig. 10(a) and 10(c) respectively show the error matrix amplitude diagram of FrFT and ADMM-Net estimating f_0 on the sample set, whereas Fig. 10(b) and 10(d) show the results for k .

The simulation shows that the sensitivity of ADMM-Net and FrFT to chirp signal parameters are nearly the same. It is not difficult to find that the estimation accuracy of chirp signal oscillates with the parameter changes, which confirms the reliability of the results in Tab. 2. Obviously, ADMM-Net with iFrF matrix as sparse basis inherits the spectrum characteristics of FrFT, which proves the effectiveness of the proposed method for chirp signal parameter estimation.

2) EVALUATION FOR DIFFERENT NETWORK ARCHITECTURES

Generally, a common problem faced by neural network-based methods is how to determine the network structure containing the optimal solution. We next evaluate the performance of our proposed ADMM-Net with varying network architectures.

Experiment 4: The depth of an ADMM-Net depends on the number of stages (i.e., N_s) corresponding to the iterations in the ADMM algorithm. To test the effect of the number of stages, we train deeper networks by adding one stage at each time with other parameters fixed, i.e., sampling ratio 30%. Then, we select a mono-component chirp signal testing set at SNR = 10 dB to conduct the experiment. Fig. 11(a) shows the average testing PSNR values of ADMM-Nets using an increasing number of stages.

The PSNR increases fast when $N_s \leq 7$ and marginally increases when further increasing the number of stages. Therefore, in this work, $N_s = 7$ is optimal from the perspective of calculation efficiency and accuracy.

Experiment 5: In the nonlinear transformation layer, ADMM-Net performs nonlinear transform inspired by the shrinkage function $S(\cdot)$ defined by $\mathbf{z}^{(k)}$ in Eqn. (24). Obviously, like ReLU, when $|x| \leq |\frac{\lambda}{\rho}|$, the negative gradient will be set to zero, which may lead to neuronal necrosis. Therefore, instead of setting it to be a determined shrinkage function, we aim to learn a more general function using a piecewise linear function. Given $\theta^{(k)}$ and $\alpha^{(k-1)}$, the output of this layer is defined as

$$\mathbf{z}^{(k)} = S_{PLF} \left(\theta^{(k)} + \alpha^{(k-1)}; \left\{ p_i, q_i^{(k)} \right\}_{i=1}^{N_c} \right) \quad (41)$$

where $S_{PLF}(\cdot)$ is a piecewise linear function determined by a set of control points $\left\{ p_i, q_i^{(k)} \right\}_{i=1}^{N_c}$:

$$S_{PLF}(a) = \begin{cases} a + q_1^{(n)} - p_1, & a < p_1 \\ a + q_{N_c}^{(n)} - p_{N_c}, & a > p_{N_c} \\ q_r^{(n)} + \frac{(a - p_r)(q_{r+1}^{(n)} - q_r^{(n)})}{p_2 - p_1}, & p_1 \leq a \leq p_{N_c} \end{cases} \quad (42)$$

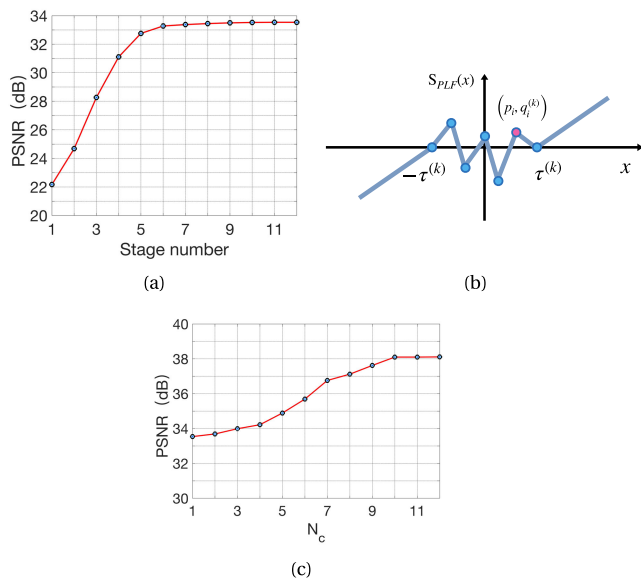


FIGURE 11. (a) Average testing PSNRs using different numbers of stages with 30% sampling ratio and SNR = 10 dB. (b) An example of an adaptive piecewise function with $N_c = 5$. (c) Average testing PSNRs using different N_c , with $N_s = 7$, with 30% sampling ratio and SNR = 10 dB.

where $r = \lfloor \frac{a-p_1}{p_2-p_1} \rfloor$ and $\{p_i\}_{i=1}^{N_c}$ are the uniform positions located within $[-\tau^{(k)}, \tau^{(k)}]$, and $\{q_i^{(k)}\}_{i=1}^{N_c}$ are the values at these positions in the k -th stage. Fig. 11(b) presents an illustrative example function. Thus, we get a more generalized ADMM-Net (GADMM-Net). Because a piecewise linear function can approximate any function, we can learn a flexible nonlinear transform function from data beyond the off-the-shelf hard or soft thresholding. Both real and imaginary parts share the same piecewise linear function,

$$\mathbf{z}^{(k)} = S_{PLF} \left(\Re \left(\boldsymbol{\theta}^{(k)} + \boldsymbol{\alpha}^{(k-1)} \right) \right) + j S_{PLF} \left(\Im \left(\boldsymbol{\theta}^{(k)} + \boldsymbol{\alpha}^{(k-1)} \right) \right) \quad (43)$$

Fig. 11(c) shows the average testing PSNR values of ADMM-Nets using an increasing number of N_c . The PSNR increases fast when $N_c \leq 10$ and marginally increases when further increasing the number of stages. To compare the estimation performance of GADMM-Net against that of ADMM-Net, we train and test a GADMM-Net and ADMM-Net with the data sets in **Experiment 1**, with the network parameters set as $N_s = 7$ and $N_c = 10$. Similarly, the sample signal in **Experiment 1** is used to test the performances of the networks. Because the estimation accuracies of \hat{k} and \hat{f}_0 of a mono-component chirp signal have the same detection threshold, let us compare the performances of the networks through the performance of δ_{f_0} . Fig. 12(a) shows the quantitative results of the two networks under different sampling ratios with SNR = 10 dB, and Fig. 12(b) shows the quantitative results under different SNRs with sampling ratio 30%.

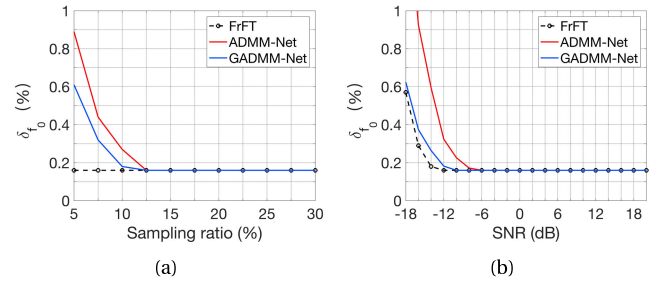


FIGURE 12. (a) δ_{f_0} curves with sampling ratio and (b) δ_{f_0} curves with SNR.

The computational efficiencies of the two networks are similar; the average testing times (GPU) of one trial are 0.47 s and 0.45 s, respectively. The results in Fig. 12 show that GADMM-Net has better estimation performance than ADMM-Net, with a 2.5% reduction in the demand for sampling ratio and better noise resistance. The SNR detection threshold has been optimized to approximately 4 dB, which is close to the estimation performance of the FrFT with full sampling.

VI. CONCLUSION

In this paper, a deep learning-based framework was proposed for chirp signal parameter estimation under sub-Nyquist sampling. The framework was expanded into a neural network group by a series of parallel ADMM algorithms, and its forward propagation is an FrFT spectrum estimation model. The framework can convert the measurement data under sub-Nyquist sampling to a fully sampled FrFT spectrum image, which can be used for chirp signal detection and parameter estimation. The simulation results show that the FrFT spectrum image obtained based on ADMM-Net has higher spectral resolution with faster computing speed than those obtained by other CS methods, which gives the parameter estimation of chirp signals better estimation accuracy at low SNR and low sampling ratio.

The framework proposed in this paper is the application of the CS unrolling method in chirp signal parameter estimation. Although it has better robustness and generalization capabilities than traditional data-driven neural networks, it is limited by the selection of sparse bases. For example, the FrFT basis used in this paper cannot directly estimate the parameters of chirp signals with duty ratio $\eta < 1$. In much of the literature on CS methods for image enhancement, the unrolling method is further generalized, where sparse bases are learned by neural networks through data. This is because the contour of the reconstructed image is known; however, in parameter estimation, it is difficult to highlight the parameter characteristics of the signal without defining a standard without a known transformation domain. Although we further generalized ADMM-Net in the activation layer in this work to improve its performance, we are still unable to deal with $\eta < 1$ adaptively, which needs to be further considered in future work.

REFERENCES

- [1] A. G. Stove, "Linear FMCW radar techniques," *IEE Proc. F, Radar Signal Process.*, vol. 139, no. 5, pp. 343–350, 1992.
- [2] P. R. Atkins, T. Collins, and K. G. Foote, "Transmit-signal design and processing strategies for sonar target phase measurement," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 1, pp. 91–104, Jun. 2007.
- [3] T. Misaridis and J. A. Jensen, "Use of modulated excitation signals in medical ultrasound. Part I: Basic concepts and expected benefits," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 52, no. 2, pp. 177–191, Feb. 2005.
- [4] M. Martone, "A multicarrier system based on the fractional Fourier transform for time-frequency-selective channels," *IEEE Trans. Commun.*, vol. 49, no. 6, pp. 1011–1020, Jun. 2001.
- [5] J. E. Palmer, H. A. Harms, S. J. Searle, and L. Davis, "DVB-T passive radar signal processing," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 2116–2126, Apr. 2013.
- [6] Y.-Y. Huang, J.-Y. Wang, and J.-Y. Chen, "Wide-band frequency estimation with sub-Nyquist sampling," *Chin. J. Radio Sci.*, vol. 2, p. 29, Feb. 2001.
- [7] X.-X. Shen, R.-Q. Ye, B. Tang, and J.-Y. Yang, "An algorithm for estimation of wideband LFM signal parameters based on subsampling," *Chin. J. Radio Sci.*, vol. 22, no. 1, pp. 43–46, 2007.
- [8] J. Martinez, J. Mejia, B. Mederos, A. Ochoa, O. Cruz-Mejía, and J. A. Marmolejo-Saucedo, "Compress sensing algorithm for estimation of signals in sensor networks," *Wireless Netw.*, pp. 1–8, Jun. 2019, doi: 10.1007/s11276-019-02031-5.
- [9] D. Zhang, Q. Zhang, B. Song, G. Zheng, and Y. Zhang, "Super resolution DOA estimation with incomplete single snapshot data via matrix completion and compress sensing," in *Proc. MATEC Web Conf.*, vol. 176, 2018, p. 01018.
- [10] A. D. Maio, Y. C. Eldar, and A. M. Haimovich, *Compressed Sensing in Radar Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2019.
- [11] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.
- [12] S. Kafle, T. Wimalajeewa, and P. K. Varshney, "Bayesian sparse signal detection exploiting laplace prior," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4259–4263.
- [13] T. Wimalajeewa and P. K. Varshney, "Sparse signal detection with compressive measurements via partial support set estimation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 46–60, Mar. 2017.
- [14] S. Jian, D. Changping, K. Ying, and C. Chen, "Compression sensing signal detection algorithm based on orthogonal matching pursuit," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 2, pp. 271–279, Jun. 2020.
- [15] C. Zhou, Y. Gu, Y. D. Zhang, Z. Shi, T. Jin, and X. Wu, "Compressive sensing-based coprime array direction-of-arrival estimation," *IET Commun.*, vol. 11, no. 11, pp. 1719–1724, Aug. 2017.
- [16] B. Liu and P. Fu, "Parameter estimation of LFM signal with narrowband jamming based on CS," in *Proc. 1st Int. Conf. Pervas. Comput., Signal Process. Appl.*, Sep. 2010, pp. 487–490.
- [17] H. Yan, C. Dong, and G. Zhao, "Parameter estimation of LFM signal based on compressed sensing," *Chin. J. Radio Sci.*, vol. 30, no. 3, pp. 449–456, 2015.
- [18] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [19] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.
- [20] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [21] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.
- [22] L. Qi, R. Tao, S. Zhou, and Y. Wang, "Detection and parameter estimation of multicomponent LFM signal based on the fractional Fourier transform," *Sci. China F, Inf. Sci.*, vol. 47, no. 2, p. 184, 2004.
- [23] R. Tao, B. Deng, and Y. Wang, "Research progress of the fractional Fourier transform in signal processing," *Sci. China F*, vol. 49, no. 1, pp. 1–25, Jan. 2006.
- [24] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.
- [25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.
- [26] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, 1993, pp. 40–44.
- [27] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 629–654, Dec. 2008.
- [28] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 3869–3872.
- [29] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
- [30] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [31] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [32] H. Su, Q. Bao, and Z. Chen, "Parameter estimation processor for chirp signals based on a complex-valued deep neural network," *IEEE Access*, vol. 7, pp. 176278–176290, 2019.
- [33] D. George and E. A. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Phys. Rev. D, Part. Fields*, vol. 97, no. 4, Feb. 2018, Art. no. 044039.
- [34] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1772–1783.
- [35] J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
- [36] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.
- [37] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
- [38] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
- [39] Z. Wang, Q. Ling, and T. S. Huang, "Learning deep l_0 -norm encoders," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2194–2200.
- [40] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4340–4348.
- [41] S.-C. Pei and J.-J. Ding, "Closed-form discrete fractional and affine Fourier transforms," *IEEE Trans. Signal Process.*, vol. 48, no. 5, pp. 1338–1353, May 2000.
- [42] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. Roy. Stat. Soc., B, Stat. Methodol.*, vol. 73, no. 3, pp. 273–282, Jun. 2011.
- [43] P. Li, Y. Lei, and H. Li, "Robust and efficient sparse-feature enhancement for generalized SAR imagery," *J. Electron. Inf. Technol.*, vol. 41, no. 12, pp. 2826–2835, 2019.
- [44] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, Sep. 1995.



HANNING SU received the B.S. degree in electronic engineering from Xidian University, in 2018. He is currently pursuing the M.A. degree with the ATR Laboratory, National University of Defense Technology. His current research interests include array signal processing, deep learning, and passive radar.



QINGLONG BAO received the B.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2003 and 2010, respectively. He is currently an Associate Professor with the National University of Defense Technology. His current research interests include radar data acquisition and signal processing.



ZENGPING CHEN received the B.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1987 and 1994, respectively. He is currently a Professor and a Ph.D. Supervisor with the Sun Yat-sen University. His current research interests include signal processing, radar systems, and automatic target recognition.

...