Parameter Estimation of 2-D Cubic Phase Signal Using Cubic Phase Function with Genetic Algorithm

Igor Djurović, Pu Wang, Cornel Ioana

Abstract— **This paper presents a generalization of cubic phase function (CPF) for two-dimensional (2-D) cubic phase polynomial phase signals (PPS). Since a straightforward application of the CPF to the 2-D PPS leads to a demanding three-dimensional (3-D) search an efficient implementation is proposed by using genetic algorithms. Simulation results demonstrate that the proposed approach outperforms the classical Francos-Friedlander technique in terms of lower SNR threshold.**

I. Introduction

Polynomial-phase signals (PPS) are important in practice and their applications include radar signal processing, sonar processing, communications, speech, etc [1]-[4]. For example, 2-D PPS signals are frequently recorded in applications involving dual-channel and interferometric synthetic aperture radars [5]-[8]. Phase differencing is a popular technique for parameter estimation of the higher-order PPSs [9]-[11]. Friedlander and Francos have generalized the phase differencing for parameter estimation of the 2-D PPSs [5], [6], [12]-[14]. Specifically, their approach requires two consecutive phase differencing steps for estimating the highest-order parameters of a 2-D cubic phase PPS (CP-PPS). Then, the lower order coefficients are estimated after dechirping procedure in a straightforward manner of the 1-D case. However, dechirping causes the error-propagation effect on estimation of lower-order parameters. Recently, O'Shea and co-workers have proposed a novel technique, cubic phase function (CPF), for estimation of a one-dimensional (1-D) CP-PPS with one difference only [15], [16]. As a result, the reduced non-linearity allows improvement of the robustness to a high amount of additive noise and, in addition, this technique avoids the error-propagation effect for estimating the second-order parameters in the CP-PPS. Numerous refinements and generalizations for the higher-order PPS have been presented in [15]-[20].

In this paper, we consider a generalization of the CPF for the 2-D CP PPS. Straightforward application of the CPF leads to a three-dimensional (3-D) search. A genetic algorithm is introduced to facilitate the search over the 3-D coordinates. Numerical results show that the genetic algorithm based proposed technique outperforms the Friedlander-Francos (FF) approach in term of lower estimation threshold for 4-5dB.

The manuscript is organized as follows: The signal model and the FF approach are described in Section II. The proposed technique is presented in Section III. Asymptotic accuracy study is carried out in Section IV. The genetic algorithm used for performing the 3-D search is described in Section V. Numerical examples are provided in Section VI. Conclusions and discussions are given in Section VII.

II. Theoretical background

A. Signal model

Consider the following 2-D CP-PPS model:

$$y(n,m) = x(n,m) + \nu(n,m),$$

$$(n,m) \in [-N/2, N/2) \times [-M/2, M/2), \qquad (1)$$

where

$$x(n,m) = Ae^{(j\phi(n,m))} =$$

$$= Ae^{\left(j\sum_{p=0}^{3}\sum_{q=0}^{3-p} c(p,q)n^p m^q\right)}, \qquad (2)$$

and $\nu(n,m)$ is a white complex Gaussian noise with zero-mean and variance σ^2, i.e., $E\{\nu(n,m)\}=0$ and $E\{\nu(n,m)\nu^*(n_1,m_1)\}=\sigma^2\delta(n-n_1,m-m_1)$. In (2), A is a constant amplitude, $\phi(n,m)$ is a polynomial phase with total order up to 3, and $c(p,q)$ is the $(p+q)$-layer parameter. The 2-D model in (2) is called the 2-D triangular form; see [5], [6], [12]-[14] and references. The signal support region is $N\times M$. In this paper, our goal is to estimate the second-order partial derivatives of the signal phase:

$$\begin{bmatrix} \frac{\partial^2\phi(n,m)}{\partial n^2} \\ \frac{\partial^2\phi(n,m)}{\partial n\partial m} \\ \frac{\partial^2\phi(n,m)}{\partial m^2} \end{bmatrix} = \begin{bmatrix} 2c(2,0)+2c(2,1)m+6c(3,0)n \\ c(1,1)+2c(2,1)n+2c(1,2)m \\ 2c(0,2)+2c(1,2)n+6c(0,3)m \end{bmatrix}, \quad (3)$$

and, based on the estimates of the above derivatives, to estimate signal parameters $\{c(p,q)|p\in[0,P]$ and $q\in[0,Q],\ P+Q\le 3\}$ and A in a more accurate manner than the FF algorithm used as the benchmark [5], [6], [12].

B. FF approach

For the 2-D CP-PPS, the FF approach uses three phase differences to estimate the highest-layer parameters as

$$\begin{aligned} &PD_{0,2}[y(n,m)] \\ &= y(n,m)[y^*(n,m+\tau_m)]^2 y(n,m+2\tau_m), \\ &PD_{1,1}[y(n,m)] \\ &= y(n,m)y^*(n+\tau_n,m)\times \\ &y^*(n,m+\tau_m)y(n+\tau_n,m+\tau_m), \\ &PD_{2,0}[y(n,m)] \\ &= y(n,m)[y^*(n+\tau_n,m)]^2 y(n+2\tau_n,m), \quad (4) \end{aligned}$$

where $*$ denotes complex conjugation, and τ_n and τ_m are two lag coefficients in the n and m axes. It should be noted that each phase difference involves a fourth-order non-linearity. If the phase differences are calculated for a noise-free signal, the resulting phases of the above differences are given by (constant phase terms independent on n and m are removed since they do not change position of the maximum):

$$\begin{aligned} \phi_{0,2}(n,m) &= \text{angle}\{PD_{0,2}[y(n,m)]\} \\ &= 2\tau_m^2 c(1,2)n + 6\tau_m^2 c(0,3)m \\ \phi_{1,1}(n,m) &= \text{angle}\{PD_{1,1}[y(n,m)]\} \\ &= 2\tau_n\tau_m c(2,1)n + 2\tau_n\tau_m c(1,2)m \\ \phi_{2,0}(n,m) &= \text{angle}\{PD_{2,0}[y(n,m)]\} \\ &= 6\tau_n^2 c(3,0)n + 2\tau_n^2 c(2,1)m. \quad (5) \end{aligned}$$

It is observed that the phase differences in (4) are 2-D complex sinusoids with frequency proportional to the highest-layer parameters $c(3,0)$, $c(2,1)$, $c(1,2)$ and $c(0,3)$. These parameters can be estimated as positions of the corresponding 2-D FT (FT_{2D}) maximum. The estimates of the highest-layer parameters can be obtained by locating the peaks of corresponding 2-D Fourier spectra. For example:

$$(\hat{\omega}_n,\hat{\omega}_m) = \arg\max_{(\omega_n,\omega_m)} |FT_{2D}[PD_{0,2}[y(n,m)]]|, \quad (6)$$

where FT_{2D} denotes the 2-D Fourier transform, and the parameters $c(1,2)$ and $c(0,3)$ can be estimated as

$$(\hat{c}(2,1),\hat{c}(0,3)) = \left(\frac{\hat{\omega}_n}{2\tau_m^2},\frac{\hat{\omega}_m}{6\tau_m^2}\right). \quad (7)$$

Other phase parameters can be estimated by dechirping the original signal with the obtained highest-layer estimates in a similar procedure as in the 1-D case.

The main problem here is that the FF approach employs the fourth-order non-linearity limiting the accuracy of the highest-order estimates. Furthermore, the current estimate errors propagate to the subsequent estimates after the dechirping.

III. Proposed approach

The proposed approach is based on the CPF, which is recently introduced by O'Shea and co-workers for the chirp-rate estimation of the CP-PPS [15], [16]. Generalizations of this approach for the higher-order PPS can be found in [17], [18]. In this paper, a new phase differencing operator, named the chirp differencing, is introduced as a generalization of the CPF for the case of the 2-D CP-PPS.

The proposed chirp difference is defined as

$$r_y(n,m;\tau_n,\tau_m)$$
$$= y(n+\tau_n, m+\tau_m)y(n-\tau_n, m-\tau_m). \quad (8)$$

Compared with the phase differences in (4), the chirp difference involves only a second-order non-linearity. This property benefits the accuracy of the estimates, i.e., it lowers the SNR threshold. Following evaluation of the chirp difference, the magnitude of the 2-D CPF is given as:

$$f_y(n,m;\boldsymbol{\Psi}) = |\mathbf{g_y}(\mathbf{n},\mathbf{m};\boldsymbol{\Psi})|^{\mathbf{2}}$$
$$= |g_y(n,m;\psi_n,\psi_{nm},\psi_m)|^2 =$$
$$\Bigg| \sum_{\tau_n=-\min(N/2-n-1,N/2+n)}^{\min(N/2-n-1,N/2+n)} \sum_{\tau_m=-\min(M/2-m-1,M/2+m)}^{\min(M/2-m-1,M/2+m)} r_y(n,m;\tau_n,\tau_m)$$
$$\times e^{(-j\psi_n\tau_n^2 - j\psi_m\tau_m^2 - j2\psi_{nm}\tau_n\tau_m)}\Bigg|^2, \quad (9)$$

where $\boldsymbol{\Psi} = [\boldsymbol{\psi_n}, \boldsymbol{\psi_{nm}}, \boldsymbol{\psi_m}]$ is a vector in the space used for estimation of the second-order derivatives of the signal phase:

$$\hat{\boldsymbol{\Omega}}(\mathbf{n},\mathbf{m}) = [\hat{\boldsymbol{\Omega}}_{\mathbf{n}}(\mathbf{n},\mathbf{m}), \hat{\boldsymbol{\Omega}}_{\mathbf{nm}}(\mathbf{n},\mathbf{m}), \hat{\boldsymbol{\Omega}}_{\mathbf{m}}(\mathbf{n},\mathbf{m})]$$
$$= \arg\max_{\boldsymbol{\Psi}} f_y(n,m;\boldsymbol{\Psi}).$$

In the absence of noise, the 2-D CPF achieves maxima at:

$$\hat{\Omega}_n(n,m) = \Omega_n(n,m)$$
$$= 2c(2,0) + 2c(2,1)m + 6c(3,0)n, \quad (10)$$
$$\hat{\Omega}_m(n,m) = \Omega_m(n,m)$$
$$= 2c(0,2) + 2c(1,2)n + 6c(0,3)m, \quad (11)$$
$$\hat{\Omega}_{nm}(n,m) = \Omega_{nm}(n,m)$$
$$= 2c(2,1)n + 2c(1,2)m + c(1,1), \quad (12)$$

where $\boldsymbol{\Omega}(\mathbf{n},\mathbf{m}) = [\boldsymbol{\Omega}_{\mathbf{n}}(\mathbf{n},\mathbf{m}), \Omega_{nm}(n,m), \Omega_m(n,m)]$ are exact values of the second-order partial derivatives of the 2-D PPS. (10)-(12) suggest that the proposed 2-D CPF can be used to estimate the second-order partial derivatives of the signal phase even in the presence of a high noise.

Based on the estimates of $\hat{\boldsymbol{\Omega}}(\mathbf{n},\mathbf{m})$ the relevant phase parameters in (3) can be estimated as follows

1) Choose three instants pairs, i.e., (n_i, m_i), $i = 1,2,3$;

2) Estimate the corresponding $\hat{\boldsymbol{\Omega}}(\mathbf{n_i},\mathbf{m_i}) = [\hat{\boldsymbol{\Omega}}_{\mathbf{n}}(\mathbf{n_i},\mathbf{m_i}),\ \hat{\Omega}_{nm}(n_i,m_i),\ \hat{\Omega}_m(n_i,m_i)]$ $i = 1,2,3$, by searching for the maxima of (9);

3) Estimate seven phase parameters including the four third-layer ones $\{c(3,0),\ c(2,1),\ c(1,2),\ c(0,3)\}$ and the three second-layer ones $\{c(2,0),\ c(1,1),\ c(0,2)\}$ using

$$\begin{bmatrix} \hat{c}(2,0) \\ \hat{c}(3,0) \\ \hat{c}'(2,1) \end{bmatrix} = \begin{bmatrix} 2 & 6n_1 & 2m_1 \\ 2 & 6n_2 & 2m_2 \\ 2 & 6n_3 & 2m_3 \end{bmatrix}^{-1} \begin{bmatrix} \hat{\Omega}_n(n_1,m_1) \\ \hat{\Omega}_n(n_2,m_2) \\ \hat{\Omega}_n(n_3,m_3) \end{bmatrix},$$
$$\begin{bmatrix} \hat{c}(0,2) \\ \hat{c}(0,3) \\ \hat{c}'(1,2) \end{bmatrix} = \begin{bmatrix} 2 & 6m_1 & 2n_1 \\ 2 & 6m_2 & 2n_2 \\ 2 & 6m_3 & 2n_3 \end{bmatrix}^{-1} \begin{bmatrix} \hat{\Omega}_m(n_1,m_1) \\ \hat{\Omega}_m(n_2,m_2) \\ \hat{\Omega}_m(n_3,m_3) \end{bmatrix},$$
$$\begin{bmatrix} \hat{c}(1,1) \\ \hat{c}''(2,1) \\ \hat{c}''(1,2) \end{bmatrix} = \begin{bmatrix} 1 & 2n_1 & 2m_1 \\ 1 & 2n_2 & 2m_2 \\ 1 & 2n_3 & 2m_3 \end{bmatrix}^{-1} \begin{bmatrix} \hat{\Omega}_{nm}(n_1,m_1) \\ \hat{\Omega}_{nm}(n_2,m_2) \\ \hat{\Omega}_{nm}(n_3,m_3) \end{bmatrix}. \quad (13)$$

It is observed that the above equations produce two estimates of "mixed" the third-layer parameters $c(1,2)$ and $c(2,1)$, i.e., $(\{\hat{c}'(2,1),\ \hat{c}''(2,1)\}$ and $\{\hat{c}'(1,2),\ \hat{c}''(1,2)\})$. Depending on the problem at hand, the final estimates of $c(1,2)$ and $c(2,1)$, can be obtained by either choosing one of the twice estimates or averaging them.

After finding the above estimates, the lower-layer phase parameters and the amplitude can be estimated in a straightforward manner as in [10]. Note that the dechirping technique is used again here to estimate the zero-layer phase parameter $c(0,0)$, the first-layer phase parameters $c(0,1)$ and $c(1,0)$, and the amplitude. Therefore, these estimates undergo the error-propagation effects from the third-layer and second-layer parameter estimation. Nevertheless, the second-layer parameter estimation is free of the error-propagation effects, while the FF approach introduces the propagated error to the second-layer parameter estimates.

Since the 2-D CPF results in a 3-D function of $[\psi_n, \psi_{nm}, \psi_m]$ for a fixed instant pair (n,m), a 3-D search is the required to locate the maxima in the 2-D CPF; see (9). However, the main problem in this approach is required 3-D search for second-order derivatives. The statistical performance analysis of the proposed technique is given in the next section while a technique for the 3-D search employing the genetic algorithm is described in Section V.

IV. Statistical performance

The proposed estimator is unbiased, i.e., $E\{\hat{c}(i,j)\} = c(i,j)$ for the second and third-layer coefficients $i+j \geq 2$. The estimator of the second layer coefficient is asymptotically efficient, i.e., the variance of these parameters estimate for high SNR is approaching the Cramer Rao lower bound (CRLB):

$$E\{[c(2,0) - \hat{c}(2,0)]^2\}$$

$$= E\{(\delta c(2,0))^2\} = \frac{90\left(1+\frac{1}{2SNR}\right)}{SNR\ N^5 M}$$

$$E\{(\delta c(0,2))^2\} = \frac{90\left(1+\frac{1}{2SNR}\right)}{SNR\ M^5 N}$$

$$E\{(\delta c(1,1))^2\} = \frac{72\left(1+\frac{1}{2SNR}\right)}{SNR\ M^3 N^3}. \qquad (14)$$

For high SNR we can neglect the term SNR^{-2} with respect to SNR^{-1} and the variance is approaching the CRLB [6]:

$$CRLB\{c(2,0)\} = \frac{90}{SNR\ N^5 M}$$

$$CRLB\{c(0,2)\} = \frac{90}{SNR\ M^5 N}$$

$$CRLB\{c(1,1)\} = \frac{72}{SNR\ M^3 N^3}. \qquad (15)$$

The estimator of the third-layer coefficients achieves the following variances:

$$E\left\{(\delta c(3,0))^2\right\} = \frac{2036.03 + \frac{1844.46}{SNR}}{SNR\ N^7 M}$$

$$E\left\{(\delta c(0,3))^2\right\} = \frac{2036.03 + \frac{1844.46}{SNR}}{SNR\ N M^7}$$

$$E\{(\delta c(2,1))^2\} = \frac{1440 + \frac{2160}{SNR}}{SNR\ N^5 M^3}$$

$$E\{(\delta c(1,2))^2\} = \frac{1440 + \frac{2160}{SNR}}{SNR\ N^3 M^5} \qquad (16)$$

while the corresponding CRLBs are

$$CRLB\{c(3,0)\} = \frac{1400}{SNR\ N^7 M}$$

$$CRLB\{c(0,3)\} = \frac{1400}{SNR\ N M^7}$$

$$CRLB\{c(2,1)\} = \frac{1080}{SNR\ N^5 M^3}$$

$$CRLB\{c(2,1)\} = \frac{1080}{SNR\ N^3 M^5}. \qquad (17)$$

It can be seen that for high SNR when the term SNR^{-2} can be neglected with respect to SNR^{-1}, the proposed estimator produces variance 1.63dB higher than the CRLB for parameters $c(3,0)$ and $c(0,3)$. In the case of the mixed parameters $c(2,1)$ and $c(1,2)$, the estimator variance is higher than the CRLB for only 1.25dB. Note that the related derivations are tedious; the main steps are outlined in the Appendix. In addition, the accuracy of third order-coefficients depends on the positions (n_i, m_i), $i = 1,2,3$, in (12). This point is also discussed within the Appendix.

V. Genetic algorithm

A. GA background

Genetic algorithms have found applications in diverse fields as possible solutions for multi-parameter and multi-modal optimization problems, especially for the case of large dimensions [21], [22]. A nice overview of the genetic algorithms and their applications in signal processing is given in [23]. The application of these algorithms in signal processing has significantly increased, especially in the fields where multiparameter optimization is required [22].

B. 2-D CPF with GA implementation

Since there is no firm proof of the convergence of the genetic algorithms for the considered problem, i.e., maximization of function (9), we turn to experimental trials for an appropriate setup of the genetic algorithm. With extensive experimental trials, it is observed that single population algorithms do not produce accurate results. Therefore multi-population algorithms are used in order to achieve more accurate estimates.

In this paper, we use the following setup of the genetic algorithm. Each parameter of $\mathbf{\Psi} = [\boldsymbol{\psi}_{\mathbf{n}}, \psi_{nm}, \psi_m]$ is represented with a 14-bit string with a grid of $1.22 \cdot 10^{-7}$. A population of 100 chromosomes, i.e., combinations of $\mathbf{\Psi} = [\boldsymbol{\psi}_{\mathbf{n}}, \boldsymbol{\psi}_{\mathbf{nm}}, \boldsymbol{\psi}_{\mathbf{m}}]$, is randomly selected, and then divided into 10 subgroups of 10 chromosomes. In each generation (fact iteration) 80% of the previous population survives, and probability of recombination, i.e., probability of combining string containing parameters $\mathbf{\Psi}$, is set to 1. Furthermore, the genetic algorithm assumes 10% of mutations (random changes in bits representing chromosomes) in one generation. The maximal number of the generations is 150 and, after 5 generations, 20% of the chromosomes move between subgroups (subpopulation).

The above setup represents an aggressive genetic algorithm due to a large percentage of mutations, the percentage of chromosomes moved between subgroups, etc. However, it still does not give us satisfactory results since sometimes we have obtained results trapped within local optimum. Specifically, the experiment results show that, around the SNR threshold, 3% of outliers in the estimation of the second-order partial derivatives are caused by the divergence of the genetic algorithm. To mitigate the effects of the divergence, an approximate bound of the divergence of the genetic algorithm is proposed. The approximate bound of the divergence is found as

$$B_{N,\mathrm{div}} = [2\arctan(SNR[\mathrm{dB}]/2+8)/\pi]^{2.5}$$

$$B_{\mathrm{div}} = A^2\Pi\,[2\arctan(SNR[\mathrm{dB}]/2+8)/\pi]^{2.5}, \tag{18}$$

where $B_{N,\mathrm{div}}$ and B_{div} denote the approximate bounds with and without normalization with respect to $A^2\Pi$, respectively, and Π is the number of terms in sum (9). Fig. 1 shows experimental results of the normalized $|g_y(n,m;\mathbf{\Psi})|$ at and away from the true second-order derivatives ($\mathbf{\Psi} = \mathbf{\Omega}(\mathbf{n},\mathbf{m})$), and compares these results with the proposed divergence bound. In this experiment, the number of instants was 8000. In Fig. 1, the thin line represents the maximal values of the $|g_y(n,m;\mathbf{\Psi})|$ deviating from the exact second-order derivatives, the thick line corresponds to the minimal values of the $|g_y(n,m;\mathbf{\Psi})|$ at the exact locations, and the dashed line denotes the approximate bound of the divergence. To set the threshold, we used the following two constraints: 1) values of $|g_y(n,m;\mathbf{\Psi})|$ at the exact position of the second-order derivatives are larger than the threshold; 2) values of this function away from the exact position are smaller than the threshold. Using these constraints we would know whether we reached the exact values for the second order derivatives or we are trapped within a local maximum. The approximate bound (18) can be used to verify whether the genetic algorithm produces an outlier. For example, if the obtained values of (9) are smaller than the bound after 150 generations, suggests that the estimate might be an outlier with high probability and thus we repeat the genetic algorithm with new starting population of chromosomes. To limit the computation time, the maximal number of the algorithm runs is set to 20. Fig. 1 clearly demonstrates that the second-order partial derivative in (10)-(12) can be accurately estimated for $SNR \geq -10$dB.

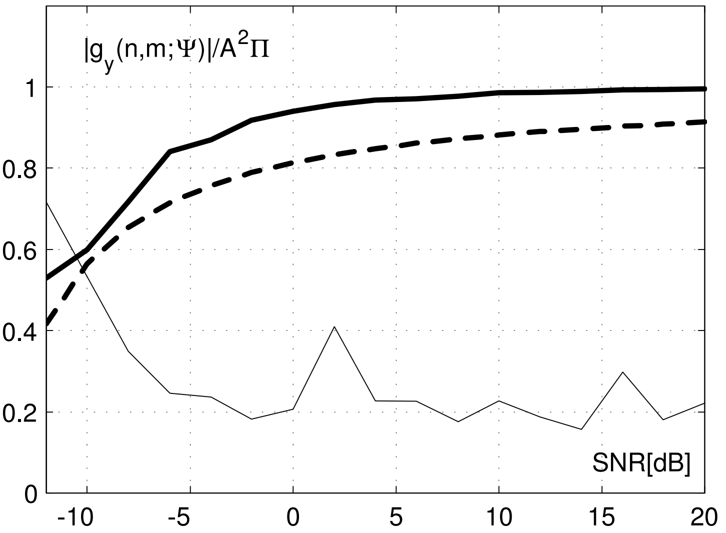

Fig. 1. Determination of the divergence bound of the genetic algorithm: thick line - minimal value of the function $|g_y(n,m;\mathbf{\Psi})|$ for the exact values of the second-order partial derivatives; thin line - the maximal values of the function $|g_y(n,m;\mathbf{\Psi})|$ for positions outside of the secondorder partial derivatives; dashed line - adopted threshold.

Hence, it can be assumed that the SNR threshold of the algorithm is about -10dB. Moreover, the proposed bound provides a good estimate of the values of the $|g_y(n,m;\mathbf{\Psi})|$ at the exact locations.

Since the approximate bound (18) requires the knowledge of A as well as the SNR, an efficient estimate of A is given as [24]:

$$\hat{A}^2 = \sqrt{|2E_2^2 - E_4|} \quad (19)$$

$$\hat{\sigma}^2 = |E_2 - \hat{A}^2| \quad (20)$$

where

$$E_i = E\{y^i(n,m)\}. \quad (21)$$

Subsequently, the SNR can be estimated as

$$\widehat{SNR} = \frac{\hat{A}^2}{\hat{\sigma}^2}. \quad (22)$$

C. Computational Issue

Before proceeding with numerical examples, we need to briefly elaborate the usage of the genetic algorithm in this application. Assume the number of discrete grids for each parameter, i.e., $[\psi_n, \psi_{nm}, \psi_m]$ is 200. In the case of the direct search, we need to search over a 3-D space with about $200^3 = 8 \cdot 10^6$ elements. We should keep in mind that all operations are performed on the 2-D data and that the 3-D search can be more demanding. To refine these estimates, additional interpolation may be required around the obtained estimates. On the other hand, the genetic algorithm requires 150 generations with 100 members and, in the worst case, 20 runs of the algorithm. This implementation leads to about $3 \cdot 10^5$ computations. As a result, the genetic algorithm is faster at least 25 times (it can be up to 500 times) than the direct search.

Note that the FF approach is less demanding than the proposed technique. Namely, the overall complexity of the FF approach is of the order of magnitude $O(NM \log_2 NM)$ since all functions (4) could be evaluated by using the 2-D FFT algorithms. The complexity of the evaluation of (9) for a single triplet $[\psi_n, \psi_{nm}, \psi_m]$ is $O(NM)$. As previously explained, we need to evaluate this function for thousands points. It means that the complexity is significantly higher than for the case of the FF approach, i.e., $O(RNM)$, where R is a total number of elements used in the genetic algorithm evaluation (between $15 \cdot 10^3$ and $3 \cdot 10^5$ in our implementation). Note that the FF approach commonly requires an additional interpolation of estimates in order to get precise results close to the CRLB.

VI. Numerical example

In this section, we numerically examine the proposed approach. First, we generated the signal given by (2) with parameters $A = 1$, $c(0,0) = 1$, $c(1,0) = 4.5 \cdot 10^{-1}$, $c(0,1) = 8.2 \cdot 10^{-2}$, $c(2,0) = -1.5 \cdot 10^{-3}$, $c(1,1) = 6 \cdot 10^{-3}$, $c(0,2) = -2.2 \cdot 10^{-3}$, $c(3,0) = 1.7 \cdot 10^{-5}$, $c(2,1) = 4 \cdot 10^{-5}$, $c(1,2) = 3.73 \cdot 10^{-5}$, $c(0,3) = -1.35 \cdot 10^{-5}$, $N = 100$ and $M = 100$. The FF approach is used as the benchmark [12]. The relevant coefficients for the FF approach are chosen as $\tau_n = \tau_m = 33$ (4), [12] and the corresponding search is performed over a 2-D space with 512×512 elements for all three functions (4). Additional interpolations are performed around initial estimates by a factor of 100. The 2-D CPF is evaluated at the instants $(50,50)$, $(50,40)$ and $(40,50)$ using the genetic algorithm setup described in previous section and the 3-D search procedure for optimization of (9). Numerical results are given in Fig. 2, where the MSEs for four characteristic higher-order parameters of the 2-D CP-PPS are depicted. Results are obtained with 200 runs of the Monte-Carlo simulation. Thin solid lines represent the MSEs achieved by the FF approach, the thick solid lines correspond to the proposed approach 2-D CPF evaluated by the genetic algorithm, the thick dashed lines depict the MSEs of the 2-D CPF with 3-D search, while the thin dashed lines are for the corresponding CRLB. Note that the MSE is calculated for each coefficient as:

$$MSE_{(p,q)} = \frac{1}{\#}\sum_{i=1}^{\#}(c(p,q) - \hat{c}(p,q))^2, \quad (23)$$

where $c(p,q)$ and $\hat{c}(p,q)$ are the considered coefficient and the corresponding estimate, while # is number of trials in the Monte-Carlo simulation.

The proposed approach with the 3-D search procedure outperforms the FF approach both in the terms of lower MSE and lower SNR threshold ≈7dB. However, the complexity of this approach limits its practical application. Due to the estimation bias, the genetic algorithm achieves higher MSEs than the FF approach for SNR≥-2dB. Generally, the estimation bias can be caused for various reasons including a small number of generations, discretization in parameter space, etc. Nevertheless, the SNR threshold of this algorithm is shown in Fig. 2 to be about 5dB lower than that of the FF approach. This is a significant advantage of the proposed approach since in the range of at least 5dB it significantly outperforms the FF technique. The threshold of the proposed genetic algorithm based approach is 2dB higher than in the case of direct search; however it significantly reduced the computation burden. The convergence of the genetic algorithm in the above setup is relatively good as described below. For SNR≥0dB, it always produces satisfactory results after a single algorithm run; for SNR =-4dB it requires an average 1.03 runs of the algorithm; for SNR =-6dB on average 1.26 runs are required; while, for SNR =-8dB, we require 3.2 runs of the algorithm. The maximal number of the algorithm runs in all trials was 13.

The numerical results imply that for high SNR it is better to use the FF approach, while below the SNR threshold of the FF technique we can apply the proposed approach with genetic algorithm. Below the threshold of the genetic algorithm we can try with the 3-D search. Note that the low accuracy of the genetic algorithm for high SNR can be avoided with the direct search in a narrow range about the obtained estimates or with some alternative search procedure (for example the least mean squares (LMS) algorithms). These topics remains outside of the current research.

VII. Conclusion and discussion

The 2-D CPF is proposed for estimation of parameters of the 2-D CP-PPS. The 3-D search for the estimates is performed using the genetic algorithm. Parameters of the genetic algorithm are described in details. Asymptotic accuracy study of the proposed technique is performed with optimal selection of coefficients (n_i, m_i) in (9). In our opinion this research is just the first step in this direction, since we will consider several important di-

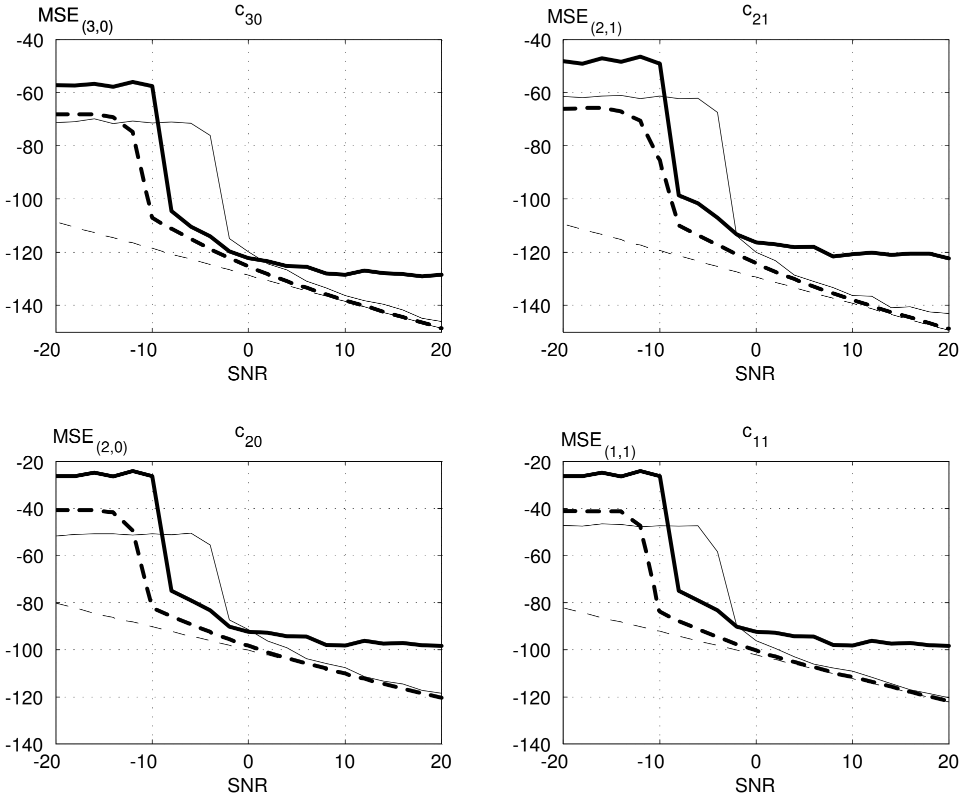

Fig. 2. MSE for the proposed genetic algorithm-based approach (thick solid line), direct 3D search with the 2D CPF (thick dashed line), the FF approach (thin solid line), and the CRLB (thin dashed line) for the estimates of (c(3, 0), c(2, 1), c(2, 0), and c(1, 1)).

rections for generalization to higher-order 2-D PPS and for multidimensional PPS. In addition, the genetic algorithm can be combined with the LMS algorithms as it was done in [23] to further reduce the computation burden. Finally, the most important direction is to reduce the dimension of the problem from the 3-D toward 2-D or even to 1-D.

VIII. Appendix

A. *Model*

Consider a noisy signal (1). The 2-D CPF is $g_y(n,m;\mathbf{\Psi}) = \mathbf{g_x(n,m;\Psi)} + \boldsymbol{\delta}\mathbf{g(n,m;\Psi)}$ where signal component is given as

$$g_x(n,m;\mathbf{\Psi}) =$$

$$\times \sum_{\tau_n}\sum_{\tau_m} r_x(n,m;\tau_n,\tau_m)$$

$$e^{-j\psi_n\tau_n^2 - j\psi_m\tau_m^2 - j2\psi_{nm}\tau_n\tau_m} \tag{24}$$

while a component introduced by interferences is:

$$\delta g(n,m;\mathbf{\Psi}) =$$

$$\sum_{\tau_n}\sum_{\tau_m} z_{x\nu}(n,m,\tau_n,\tau_m)$$

$$\times e^{-j\psi_n\tau_n^2 - j\psi_m\tau_m^2 - j2\psi_{nm}\tau_n\tau_m} \tag{25}$$

where

$$z_{x\nu}(n,m,\tau_n,\tau_m)$$

$$= x(n+\tau_n, m+\tau_m)\nu(n-\tau_n, m-\tau_m) +$$

$$\nu(n+\tau_n, m+\tau_m)x(n-\tau_n, m-\tau_m) +$$

$$\nu(n+\tau_n, m+\tau_m)\nu(n-\tau_n, m-\tau_m). \tag{26}$$

Function $|g_x(n,m;\mathbf{\Psi})|^{\mathbf{2}}$ achieves the maximum for $\mathbf{\Psi} = \mathbf{\Omega(n,m)}$= $[\mathbf{\Omega_n(n,m)}, \Omega_{nm}(n,m),$ $\Omega_m(n,m)$= $\left[\frac{\partial^2\boldsymbol{\phi}\mathbf{(n,m)}}{\partial\mathbf{n^2}}\right.$, $\frac{\partial^2\phi(n,m)}{\partial n\partial m}$, $\left.\frac{\partial^2\phi(n,m)}{\partial m^2}\right]$.

Due to the noisy term $\delta g(n,m;\mathbf{\Psi})$ the maximum of $f_y(n,m;\mathbf{\Psi})$ moves to $\mathbf{\Omega}+\boldsymbol{\delta\Omega}=[\mathbf{\Omega_n}+\boldsymbol{\delta\Omega}_\mathbf{n},\ \Omega_{nm}+\delta\Omega_{nm},\ \Omega_m+\delta\Omega_m]$. Hereafter, due to brevity we remove the dependency of the second-order derivatives of the signal phase on position (n,m).

Then, it follows:

$$\left[\frac{\partial f_x(n,m;\mathbf{\Psi})}{\partial\psi_i}+\frac{\partial\delta f(n,m;\mathbf{\Psi})}{\partial\psi_i}\right]|_{\mathbf{\Psi}=\mathbf{\Omega}+\boldsymbol{\delta\Omega}}$$

$$=0,\ i=1,2,3, \tag{27}$$

where ψ_i, $i=1,2,3$ are corresponding elements of the vector $\mathbf{\Psi}$, $\psi_1=\psi_n$, $\psi_2=\psi_{nm}$, $\psi_3=\psi_m$.

Perturbation $\delta f(n,m;\mathbf{\Psi})$ can be approximated by $\delta g(n,m;\mathbf{\Psi})$ under the assumption that the SNR is relatively large resulting in neglecting of the term $|\delta g^*(n,m;\mathbf{\Psi})|^2$

$$\delta f(n,m;\mathbf{\Psi})$$

$$\approx 2\,\mathrm{Re}\left\{g_x(n,m;\mathbf{\Psi})\boldsymbol{\delta}\mathbf{g}^*(\mathbf{n},\mathbf{m};\mathbf{\Psi})\right\}. \tag{28}$$

We expand (27) around $\mathbf{\Omega}$ for high SNR into the Taylor expansion up to the second term:

$$\frac{\partial f_x(n,m;\mathbf{\Psi})}{\partial\psi_i}|_{\mathbf{\Psi}=\mathbf{\Omega}}+\frac{\partial\delta f(n,m;\mathbf{\Psi})}{\partial\psi_i}|_{\mathbf{\Psi}=\mathbf{\Omega}}+$$

$$\sum_{l=1}^{3}\delta\Omega_l\frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_i\partial\psi_l}|_{\mathbf{\Psi}=\mathbf{\Omega}}=0,\quad i=1,2,3. \tag{29}$$

This can be written in the matrix form:

$$\boldsymbol{\delta}\mathbf{F_1}+\mathbf{F_2}\boldsymbol{\delta\Omega}=\mathbf{0}, \tag{30}$$

where

$$\boldsymbol{\delta}\mathbf{F_1}$$

$$=\left[\begin{array}{ccc}\frac{\partial\delta f(n,m;\mathbf{\Psi})}{\partial\psi_1} & \frac{\partial\delta f(n,m;\mathbf{\Psi})}{\partial\psi_2} & \frac{\partial\delta f(n,m;\mathbf{\Psi})}{\partial\psi_3}\end{array}\right]^T_{|\mathbf{\Psi}=\mathbf{\Omega}},$$

$$\boldsymbol{\delta\Omega}=\left[\begin{array}{ccc}\delta\Omega_1 & \delta\Omega_2 & \delta\Omega_3\end{array}\right]^{\mathbf{T}}$$

and

$$\mathbf{F_2}=$$

$$\left[\begin{array}{ccc}\frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial^2\psi_1} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_1\partial\psi_2} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_1\partial\psi_3}\\ \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_2\partial\psi_1} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_2^2} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_2\partial\psi_3}\\ \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_3\partial\psi_1} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_3\partial\psi_2} & \frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_3^2}\end{array}\right]_{|\mathbf{\Psi}=\mathbf{\Omega}}. \tag{31}$$

Elements of $\mathbf{F_2}$ are given as:

$$[\mathbf{F_2}]_{il}=\frac{\partial^2 f_x(n,m;\mathbf{\Psi})}{\partial\psi_i\partial\psi_l}|_{\mathbf{\Psi}=\mathbf{\Omega}}$$

$$=2\,\mathrm{Re}\left\{\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_i\partial\psi_l}g_x^*(n,m;\mathbf{\Omega})+\right.$$

$$\left.\frac{\partial g_x(n,m;\mathbf{\Omega})}{\partial\psi_i}\frac{\partial g_x^*(n,m;\mathbf{\Omega})}{\partial\psi_l}\right\}. \tag{32}$$

Note that the matrix $\mathbf{F_2}$ is symmetric since $[\mathbf{F_2}]_{il}=[\mathbf{F_2}]_{li}$.

Similarly, elements of $\delta\mathbf{F_1}$ follow from (28) as:

$$[\delta\mathbf{F_1}]_\mathbf{i}$$

$$=2\,\mathrm{Re}\left\{\frac{\partial g_x(n,m;\mathbf{\Omega})}{\partial\psi_i}\delta g^*(n,m;\mathbf{\Omega})+\right.$$

$$\left.g_x(n,m;\mathbf{\Omega})\frac{\partial\boldsymbol{\delta}\mathbf{g}^*(\mathbf{n},\mathbf{m};\mathbf{\Omega})}{\partial\boldsymbol{\psi}_\mathbf{i}}\right\}. \tag{33}$$

B. *Asymptotic MSE of the second-layer parameter estimates*

Since the function $f_x(n,m;\mathbf{\Psi})$ is noise-free and $\delta f(n,m;\mathbf{\Psi})$ is random perturbation, the matrix $\mathbf{F_2}$ is deterministic and the vector $\delta\mathbf{F_1}$ is random. Then the estimation error can be expressed as

$$\delta\mathbf{\Omega}=-\mathbf{F_2^{-1}}\boldsymbol{\delta}\mathbf{F_1}. \tag{34}$$

Taking the expectation with respect to $\delta\mathbf{\Omega}$ the bias is:

$$E\{\delta\mathbf{\Omega}\}=-\mathbf{F_2^{-1}}\mathbf{E}\{\boldsymbol{\delta}\mathbf{F_1}\} \tag{35}$$

and the covariance matrix of $\delta\mathbf{\Omega}$ is:

$$E\{(\delta\mathbf{\Omega})(\boldsymbol{\delta\Omega})^\mathbf{T}\}=\mathbf{F_2^{-1}}\mathbf{C}_{\boldsymbol{\delta}\mathbf{F_1}}\mathbf{F_2^{-1}} \tag{36}$$

where $C_{\delta\mathbf{F_1}}=E\{(\delta\mathbf{F_1})(\boldsymbol{\delta}\mathbf{F_1})^\mathbf{T}\}$. Finally, the variances for the estimate errors are given as:

$$E\{(\delta\Omega_i)^2\}=[\mathbf{F_2^{-1}}\mathbf{C}_{\boldsymbol{\delta}\mathbf{F_1}}\mathbf{F_2^{-1}}]_\mathbf{ii}. \tag{37}$$

We can use the following intermediate results:

$$g_y^*(n,m;\mathbf{\Psi})|_{\mathbf{\Psi}=\mathbf{\Omega}}=\mathbf{g_y^*}(\mathbf{n},\mathbf{m};\mathbf{\Omega})$$

$$= \sum_{\tau_n=-K}^{K} \sum_{\tau_m=-L}^{L} A^2 e^{-j2\phi(n,m)}$$

$$\times e^{j(\Omega_n-\psi_n)\tau_n^2+j(\Omega_m-\psi_m)\tau_m^2+j2(\Omega_{nm}-\psi_{nm})\tau_n\tau_m}$$

$$|_{\mathbf{\Psi}=\mathbf{\Omega}} = \mathbf{A^2 e^{-j2\phi(n,m)}(2K+1)(2L+1)}. \quad (38)$$

Using the following approximation that holds for even k and $K \gg k$:

$$\sum_{m=-K}^{K} m^k \approx 2K^{k+1}/(k+1) \quad (39)$$

we can evaluate derivatives of $g_x(n,m;\mathbf{\Psi})$ for $\mathbf{\Psi}=\mathbf{\Omega}$ as

$$\frac{\partial g_x(n,m;\mathbf{\Omega})}{\partial\psi_1} \approx -jA^2 e^{j2\phi(n,m)}\frac{2K^3}{3}(2L+1) \quad (40)$$

$$\frac{\partial g_x(n,m;\mathbf{\Omega})}{\partial\psi_3} \approx -jA^2 e^{j2\phi(n,m)}\frac{2L^3}{3}(2K+1) \quad (41)$$

$$\frac{\partial g_x(n,m;\mathbf{\Omega})}{\partial\psi_2} = 0, \quad (42)$$

$$\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_1^2} \approx -A^2 e^{j2\phi(n,m)}\frac{2K^5}{5}(2L+1) \quad (43)$$

$$\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_1\partial\psi_3} \approx -A^2 e^{j2\phi(n,m)}\frac{2K^3}{3}\frac{2L^3}{3} \quad (44)$$

$$\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_1\partial\psi_2} = \frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_2\partial\psi_3} = 0 \quad (45)$$

$$\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_3^2} \approx -A^2 e^{j2\phi(n,m)}\frac{2L^5}{5}(2K+1) \quad (46)$$

$$\frac{\partial^2 g_x(n,m;\mathbf{\Omega})}{\partial\psi_2^2} \approx -4A^2 e^{j2\phi(n,m)}\frac{2K^3}{3}\frac{2L^3}{3} \quad (47)$$

where $(2K+1)\times(2L+1)$ is the window size.

Then the matrix $\mathbf{F_2}$ is given as:

$$\mathbf{F_2}$$

$$= -\frac{128}{9}A^4K^2L^2 \begin{bmatrix} \frac{1}{5}K^4 & 0 & 0 \\ 0 & K^2L^2 & 0 \\ 0 & 0 & \frac{1}{5}L^4 \end{bmatrix}. \quad (48)$$

To determine the random term $\delta\mathbf{F_1}$ (33) by using (25) we need to evaluate $\partial\delta g^*(n,m;\mathbf{\Psi})/\partial\psi_i|_{\mathbf{\Psi}=\mathbf{\Omega}}$, $i=1,2,3$. This term can be written as

$$\frac{\partial\delta g^*(n,m;\mathbf{\Omega})}{\partial\psi_i}$$

$$= j\sum_{\tau_n=-M}^{M}\sum_{\tau_m=-L}^{L}\xi_i(\tau_n,\tau_m)$$

$$\times z_{x\nu}^*(n,m;\tau_n,\tau_m)e^{j\psi_n\tau_n^2+j\psi_m\tau_m^2+j2\psi_{nm}\tau_n\tau_m} \quad (49)$$

where

$$\xi_i(\tau_n,\tau_m) = \begin{cases} \tau_n^2 & i=1 \\ 2\tau_n\tau_m & i=2 \\ \tau_m^2 & i=3. \end{cases} \quad (50)$$

After tedious but straightforward derivations it follows:

$$[\delta\mathbf{F_1}]_\mathbf{i} = -\mathbf{8A^2KL}\,\mathrm{Im}\{\mathbf{\Gamma(i,K,L)}\}, \quad (51)$$

where

$$\Gamma(i,K,L)$$

$$= e^{j2\phi(n,m)}\sum_{\tau_n=-K}^{K}\sum_{\tau_m=-L}^{L}\lambda_i(\tau_n,\tau_m)\times$$

$$z_{x\nu}^*(n,m;\tau_n,\tau_m)e^{j\psi_n\tau_n^2+j\psi_m\tau_m^2+j2\psi_{nm}\tau_n\tau_m} \quad (52)$$

and

$$\lambda_i(\tau_n,\tau_m) = \begin{cases} \tau_n^2-\frac{K^2}{3} & i=1 \\ 2\tau_n\tau_m & i=2 \\ \tau_m^2-\frac{L^2}{3} & i=3. \end{cases} \quad (53)$$

Again, for the sake of brevity we removed explicit dependence of $\Gamma(i,K,L)$ on the position (n,m). Since $E\{\delta\mathbf{F_1}\}=\mathbf{0}$ then $E\{\delta\mathbf{\Omega}\}=\mathbf{0}$. In other words, the estimator is unbiased. In order to determine variances, we need to determine $C_{\delta\mathbf{F_1}} = E\{(\delta\mathbf{F_1})(\boldsymbol{\delta}\mathbf{F_1})^\mathbf{T}\}$, whose the elements are:

$$[C_{\delta\mathbf{F_1}}]_{il} = \left[E\{(\delta\mathbf{F_1})(\boldsymbol{\delta}\mathbf{F_1})^\mathbf{T}\}\right]_{il} =$$

$$64A^4K^2L^2E\{\mathrm{Im}[\Gamma(i,K,L)]\,\mathrm{Im}[\Gamma(l,K,L)]\}. \quad (54)$$

Here, the expectation can be rewritten as

$$E\{\mathrm{Im}[\Gamma(i,K,L)]\,\mathrm{Im}[\Gamma(l,K,L)]\}$$

$$= -\frac{1}{4}E\{[\Gamma^*(i,K,L)$$

$$-\Gamma(i,K,L)[\Gamma^*(l,K,L)-\Gamma(l,K,L]\},$$

where

$$E\{\Gamma(i,K,L)\Gamma(l,K,L)\}$$
$$=E\{\Gamma^*(i,K,L)\Gamma^*(l,K,L)\}=0$$

and

$$E\{\Gamma(i,K,L)\Gamma^*(l,K,L)\}\approx 0 \quad \text{for } i\neq l.$$

Then:

$$[C_{\delta\mathbf{F_1}}]_{il} \approx \begin{cases} 32A^4K^2L^2E\{|\Gamma(i,K,L)|^2\} & i=l \\ 0 & i\neq l. \end{cases} \tag{55}$$

By using (52) and (39) we obtain

$$E\{|\Gamma(i,K,L)|^2\} = \begin{cases} \frac{32}{45}(2A^2\sigma^2+\sigma^4)K^5L & i=1 \\ \frac{32}{9}(2A^2\sigma^2+\sigma^4)K^3L^3 & i=2 \\ \frac{32}{45}(2A^2\sigma^2+\sigma^4)KL^5 & i=3. \end{cases} \tag{56}$$

Then, it follows:

$$C_{\delta\mathbf{F_1}} = \frac{1024}{9}(2A^2\sigma^2+\sigma^4)A^4K^3L^3 \times \begin{bmatrix} \frac{1}{5}K^4 & 0 & 0 \\ 0 & K^2L^2 & 0 \\ 0 & 0 & \frac{1}{5}L^4 \end{bmatrix}. \tag{57}$$

It can be noticed that $C_{\delta\mathbf{F_1}} = -8KL(2A^2\sigma^2+\sigma^4)\mathbf{F_2}$. Now, we have:

$$E\{(\delta\Omega_i)^2\} = [\mathbf{F_2^{-1}C_{\delta F_1}F_2^{-1}}]_{\mathbf{ii}} = -8KL(2A^2\sigma^2+\sigma^4)[\mathbf{F_2^{-1}}]_{\mathbf{ii}}, \tag{58}$$

where

$$\mathbf{F_2^{-1}} = -\frac{\mathbf{9}}{\mathbf{128A^4K^2L^2}} \begin{bmatrix} \frac{5}{K^4} & 0 & 0 \\ 0 & \frac{1}{K^2L^2} & 0 \\ 0 & 0 & \frac{5}{L^4} \end{bmatrix} \tag{59}$$

and

$$E\{(\delta\Omega_1)^2\} = E\{(\delta\Omega_n)^2\} = \frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\ K^5L} \tag{60}$$

$$E\{(\delta\Omega_2)^2\} = E\{(\delta\Omega_{nm})^2\} = \frac{9\left(2+\frac{1}{SNR}\right)}{16SNR\ K^3L^3}$$

$$E\{(\delta\Omega_3)^2\} = E\{(\delta\Omega_m)^2\} = \frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\ KL^5}. \tag{61}$$

To estimate the second-layer parameters we assume that the evaluation is performed for central instant of considered domain. Then, $\tau_n \in [-(N-1)/2,(N-1)/2]$ and $\tau_m \in [-(M-1)/2,(M-1)/2]$ and we have

$$E\{(\delta c(2,0))^2\} = \frac{E\{(\delta\Omega_n)^2\}|_{\substack{n=0\\m=0}}}{4} = \frac{90\left(1+\frac{1}{2SNR}\right)}{SNR\ N^5M} \tag{62}$$

$$E\{(\delta c(0,2))^2\} = \frac{E\{(\delta\Omega_m)^2\}|_{\substack{n=0\\m=0}}}{4} = \frac{90\left(1+\frac{1}{2SNR}\right)}{SNR\ M^5N}$$

$$E\{(\delta c(1,1))^2\} = E\{(\delta\Omega_{nm})^2\}|_{\substack{n=0\\m=0}} = \frac{72\left(1+\frac{1}{2SNR}\right)}{SNR\ M^3N^3}. \tag{63}$$

This confirms that the proposed estimator is **asymptotically efficient** for the second-layer parameters (neglecting the terms containing SNR^{-2}).

C. *Asymptotic MSE of the third-layer parameter estimates*

Here, we will consider the asymptotic MSEs of parameters $c(2,1)$ and $c(3,0)$ while similar derivations hold for parameters $c(1,2)$ and $c(0,3)$. To simplify the analysis we estimate Ω_n for three points $(n=0,\ m=0)$, $(n,m=0)$ and $(n=0,\ m)$. The estimates can be written as:

$$\begin{bmatrix} \hat{c}(2,0) \\ \hat{c}(3,0) \\ \hat{c}(2,1) \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 6n & 0 \\ 2 & 0 & 6m \end{bmatrix}^{-1} \begin{bmatrix} \Omega_n(0,0) \\ \Omega_n(n,0) \\ \Omega_n(0,m) \end{bmatrix}, \tag{64}$$

while the estimation error for parameters is linearly related to errors in estimation of phase parameters.

$$\begin{bmatrix} \delta\hat{c}(2,0) \\ \delta\hat{c}(3,0) \\ \delta\hat{c}(2,1) \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 2 & 6n & 0 \\ 2 & 0 & 6m \end{bmatrix}^{-1} \begin{bmatrix} \delta\Omega_n(0,0) \\ \delta\Omega_n(n,0) \\ \delta\Omega_n(0,m) \end{bmatrix}, \quad (65)$$

The covariance matrix exhibits

$$\mathbf{C_1} = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 6n & 0 \\ 2 & 0 & 6m \end{bmatrix}^{-\mathbf{1}}$$
$$\times E\left\{ \begin{bmatrix} \delta\Omega_n(0,0) \\ \delta\Omega_n(n,0) \\ \delta\Omega_n(0,m) \end{bmatrix} \begin{bmatrix} \delta\Omega_n(0,0) \\ \delta\Omega_n(n,0) \\ \delta\Omega_n(0,m) \end{bmatrix}^T \right\}$$
$$\times \begin{bmatrix} 2 & 0 & 0 \\ 2 & 6n & 0 \\ 2 & 0 & 6m \end{bmatrix}^{-1}. \quad (66)$$

The diagonal elements of $\mathbf{C_1}$ give the MSE of the three estimates $c(2,0)$, $c(3,0)$ and $c(2,1)$. Here, we will consider just the third-order parameters:

$$E\{(\delta c(3,0))^2\} = \frac{E\{(\delta\Omega_n(0,0))^2\}}{36n^2}$$
$$+ \frac{E\{(\delta\Omega_n(n,0))^2\} - 2E\{\delta\Omega_n(0,0)\delta\Omega_n(n,0)\}}{36n^2}$$
$$E((\delta c(2,1))^2\} = \frac{E\{(\delta\Omega_n(0,0))^2\}}{4m^2} +$$
$$\frac{E\{(\delta\Omega_n(0,m))^2\} - 2E\{\delta\Omega_n(0,0)\delta\Omega_n(0,m)\}}{4m^2}. \quad (67)$$

C.1 Asymptotic MSEs of the $c(3,0)$ estimates

To calculate $E\{(\delta c(3,0))^2\}$ we need to determine $E\{\delta\Omega_n(0,0)\delta\Omega_n(n,0)\}$ while $E\{(\delta\Omega_n(0,0))^2\}$ and $E\ \{(\delta\Omega_n(n,0))^2\}$ follows from (60) for $K = N/2$ and $L = M/2$ for $(0,0)$ and $K = N/2-n$ and $L = M/2$ for $(n,0)$. This quantity is equal to

$$E\{\delta\Omega_n(0,0)\delta\Omega_n(n,0)\}$$
$$= \frac{45^2}{16^2 A^4 \left(\frac{N}{2}\right)^5 \left(\frac{N}{2}-n\right)^5 \left(\frac{M}{2}\right)^2} \times$$
$$E\{\mathrm{Im}\{\Gamma(1,N/2,M/2)\}$$
$$\mathrm{Im}\{\Gamma(1,N/2-n,M/2)\}\} =$$
$$\frac{45^2 E\{\Gamma(1,N/2,M/2)\Gamma^*(1,N/2-n,M/2)\}}{512A^4 \left(\frac{N}{2}\right)^5 \left(\frac{N}{2}-n\right)^5 \left(\frac{M}{2}\right)^2} \quad (68)$$

where

$$E\{\Gamma(1,N/2,M/2)\Gamma^*(1,N/2-n,M/2)\}$$
$$\approx 4A^2\sigma^2 \frac{(N-2n)^5 M}{180} \quad (69)$$
$$E\{\Gamma(1,N/2,M/2)\Gamma(1,N/2-n,M/2)\} = 0. \quad (70)$$

As a result

$$E\{(\delta c(3,0))^2\} = \frac{E\{(\delta\Omega_n(0,0))^2\}}{36n^2} +$$
$$\frac{E\{(\delta\Omega_n(n,0))^2\} - 2E\{\delta\Omega_n(0,0)\delta\Omega_n(n,0)\}}{36n^2}$$
$$\approx \frac{\frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\left(\frac{N}{2}\right)^5\left(\frac{M}{2}\right)}}{36n^2}$$
$$+ \frac{\frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\left(\frac{N}{2}-n\right)^5\left(\frac{M}{2}\right)} - \frac{720}{SNR\ N^5 M}}{36n^2}. \quad (71)$$

Numerical results show that $n = 0.1114N$ or $n \approx 0.11N$ produces the minimum MSE for a high SNR (e.g. $SNR = 20$dB). By substituting $n \approx 0.11N$ into the above expression we obtained the MSE for the $c(3,0)$ estimate as:

$$E\left\{(\delta c(3,0))^2\right\} = \frac{2036.03 + \frac{1844.46}{SNR}}{SNR\ \ N^7 M}. \quad (72)$$

Similarly we can derive the asymptotic accuracy for $c(0,3)$ as:

$$E\left\{(\delta c(0,3))^2\right\} = \frac{2036.03 + \frac{1844.46}{SNR}}{SNR\ \ NM^7}. \quad (73)$$

Remark: It is interesting to note that, the accuracy in estimation of the $c(3,0)$ and $c(0,3)$ is similar to the 1-D case [16]. In the 1-D case, $n = 0.11N$ gives MSE of the third-order parameter:

$$E\left\{(\delta a_3)^2\right\} = \frac{2038 + \frac{1844}{SNR}}{SNR\ \ N^7}. \quad (74)$$

C.2 Asymptotic MSEs of the $c(2,1)$ estimates

Again, from (61) we have to determine $E\{\delta\Omega_n(0,0)\delta\Omega_n(0,m)\}$ since $E\{(\delta\Omega_n(0,0))^2\}$ is given with (62) while $E\{(\delta\Omega_n(0,m))^2\}$ follows from (60) as:

$$E\{(\delta\Omega_n(0,m))^2\} = \frac{180\left(2+\frac{1}{SNR}\right)}{SNR\ N^5\,(M-2m)}. \quad (75)$$

After a tedious but straightforward derivations it follows that

$$E\{\delta\Omega_n(0,0)\delta\Omega_n(0,m)\} = \frac{45^2 E\left\{\Gamma(1,N/2,M/2)\Gamma^*(1,N/2,M/2-m)\right\}}{512A^4\left(\frac{N}{2}\right)^{10}\left(\frac{M}{2}\right)\left(\frac{M}{2}-m\right)} \tag{76}$$

where

$$E\left\{\Gamma(1,N/2,M/2)\Gamma^*(1,N/2,M/2-m)\right\} \approx 4A^2\sigma^2\frac{N^5(M-2m)}{180}. \tag{77}$$

Then, the MSE for parameter $c(2,1)$ is

$$E\{(\delta c(2,1))^2\} \approx \frac{\frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\left(\frac{N}{2}\right)^5\left(\frac{M}{2}\right)}}{4m^2} + \frac{\frac{45\left(2+\frac{1}{SNR}\right)}{16SNR\left(\frac{N}{2}\right)^5\left(\frac{M}{2}-m\right)} - \frac{720}{SNR\ N^5M}}{4m^2}. \tag{78}$$

Numerical results show that $m = 0.2502M$ or $m \approx 0.25M$ gives minimum MSE for high SNR (e.g. $SNR = 20$dB). By inserting $m \approx 0.25M$ into the above expression we obtain the MSE for the $c(2,1)$ estimate as

$$E\{(\delta c(2,1))^2\} = \frac{1440 + \frac{2160}{SNR}}{SNR\ \ N^5M^3}. \tag{79}$$

Similarly, the asymptotic accuracy for $c(1,2)$ is equal to:

$$E\{(\delta c(1,2))^2\} = \frac{1440 + \frac{2160}{SNR}}{SNR\ \ N^3M^5}. \tag{80}$$

IX. Acknowledgment

The work of Igor Djurović was realized at the INP Grenoble, France, and supported by the CNRS, under contract No. 180 089 013 00387. This research is supported in part by the Ministry of Science and Education of Montenegro. The work of Pu Wang was supported in part by the National Natural Science Foundation of China under Grant 60802062.

References

[1] A. W. Rihaczek, *Principles of high-resolution radar*, McGraw-Hill, New York, 1969.

[2] D. R. Wehner, *High-Resolution Radar*, Artech House, Norwood, MA, 1995.

[3] J. C. Curlandar and R. N. McDonough, *Synthetic Aperture Radar - System and Signal Processing*, John Wiley & Sons, New York, 1991.

[4] B. Porat, *Digital processing of random signals: theory and methods*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[5] B. Friedlander and J. M. Francos, "Model based phase unwrapping of 2-D signals," *IEEE Trans. Signal Processing*, Vol. 44, No. 12, pp. 2999-3007, Dec. 1996.

[6] J. M. Francos and B. Friedlander, "Two-dimensional polynomial phase signals: parameter estimation and bounds," *Multidimensional Systems and Signal Processing*, Vol. 9, pp. 173-205, 1998.

[7] J. J. Sharma, C. H. Gierull, and M. J. Collins, "The influence of target acceleration on velocity estimation in dual-channel SAR-GMTI," *IEEE Trans. Geoscience and Remote Sensing*, Vol. 44, No. 1, pp. 134–147, Jan. 2006.

[8] J. J. Sharma, C. H. Gierull, and M. J. Collins, "Compensating the effects of target acceleration in dual-channel SARCGMTI," *IEE Proc.-Radar Sonar Navigation*, vol. 153, no. 1, pp. 53–62, Feb. 2006.

[9] B. Porat and B. Friedlander, "Asymptotic statistical analysis of the high-order ambiguity function for parameter estimation of polynomial-phase signals," *IEEE Trans. Information Theory*, Vol. 42, No. 3, pp. 995–1001, May 1996.

[10] S. Peleg and B. Porat, "Linear FM signal parameter estimation from discrete-time observations," *IEEE Trans. Aerospace and Electronics Systems*, Vol. 27, No. 4, pp. 607–614, July 1991.

[11] S. Barbarossa, A. Scaglione and G. B. Giannakis, "Product high-order ambiguity function for multicomponent polynomial-phase signal modeling", *IEEE Trans. Signal Processing*, Vol. 46, No. 3, pp. 691-708, Mar. 1998.

[12] B. Friedlander and J. M. Francos, "An estimation for 2-D polynomial phase signals," *IEEE Trans. Image Processing*, Vol. 5, No. 6, pp. 1084-1087, June 1996.

[13] J. M. Francos and B. Friedlander, "Optimal parameter selection in the phase differencing algorithm for 2-D phase estimation," *IEEE Trans. Signal Processing*, Vol. 47, No. 1, pp. 273-279, Jan. 1999.

[14] J. M. Francos and B. Friedlander, "Parameter estimation of 2-D random amplitude polynomial-phase signals," *IEEE Trans. Signal Processing*, Vol. 47, No. 7, pp. 1795-1810, July 1999.

[15] P. O'Shea, "A new technique for instantaneous frequency rate estimation," *IEEE Signal Processing Letters*, Vol. 9, No. 8, pp. 251-252, Aug. 2002.

[16] P. O'Shea, "A fast algorithm for estimating the parameters of a quadatic FM signal," *IEEE Tran. Signal Processing*, Vol. 52, No. 2, pp. 385-393, Feb. 2004.

[17] M. Farquharson, P. O'Shea, G. Ledwich, "A computationally efficient technique for estimating the parameters of polynomial phase signals from

noisy observations," *IEEE Tran. Signal Processing*, Vol. 53, No. 8, pp. 3337-3342, Aug. 2005.

[18] M. Farquharson and P. O'Shea, "Extending the performance of the cubic phase function algorithm," *IEEE Tran. Signal Processing*, Vol. 55, No. 10, pp. 4767-4774, Oct. 2007.

[19] P. Wang, I. Djurović and J. Yang, "Modifications of the cubic phase function," *Chinese Journal of Electronics*, Vol. 17, No. 1, pp. 189-194, Jan. 2008.

[20] P. Wang, I. Djurović and J. Yang, "Generalized high-order phase function for parameter estimation of polynomial phase signal," *IEEE Trans. Signal Processing*, Vol. 56, No. 7, pp. 3023-3028, July 2008.

[21] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley.

[22] A. B. Gershman, M. Pesavento and M. G. Amin, "Estimating parameters of multiple wideband polynomial-phase sources in sensor arrays," *IEEE Trans. Signal Processing*, Vol. 49, No.12, pp. 2924-2934, Dec. 2001.

[23] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, Vol. 13, No. 6, pp. 22-37, Nov. 1996.

[24] S.-C. Sekhar, T.V. Creenivas, "Signal to noise ratio estimation using higher order moments", *Signal Processing*, Vol. 86, pp 716-732, 2006.