Chromatic Derivatives and Approximations in Practice (III): Continuous Time MUSIC Algorithm for Adaptive Frequency Estimation in Colored Noise

Aleksandar Ignjatović, Chamith Wijenayake and Gabriele Keller

***Abstract*—We present a new frequency estimation algorithm, which we call *Continuous Time MUSIC algorithm (CT-MUSIC)*. As we demonstrate by our extensive tests, CT-MUSIC significantly outperforms the original MUSIC algorithm for frequency estimation. As its name suggests, our algorithm can be seen as a continuous time version of the MUSIC algorithm. In addition to being significantly more noise robust and providing much better super-resolution, our algorithm, in the course of its operation, can "learn" the power spectral density of the noise present and can dynamically adapt to it by a background process, significantly further enhancing its noise robustness. Moreover, our algorithm is equally applicable to both uniformly and non-uniformly sampled signals, and if its front end is implemented as an analog filterbank, such a filterbank can not only be sampled at non-uniformly distributed sampling instants, but also the time stamps of these instants need not be kept (for as long as all the filters of the filterbank are sampled simultaneously at each sampling instant). The algorithm presented further demonstrates how our novel method of processing continuous time signals based on chromatic approximations, described in detail in the first part of this paper, can be applied in practice.**

I. Introduction

Chromatic derivatives are special, numerically robust differential operators which preserve spectral features of a signal; the associated chromatic approximations accurately capture local features of a signal. These notions allow extremely accurate and efficient representation and processing of continuous time signals, by operating on continuous time slices of such signals, rather than on their discrete time samples. Mathematical foundations of chromatic derivatives and expansions were presented in considerable detail in [1]–[3]. The first part of this paper, *Chromatic Derivatives and Approximations in Practice - Part I: A General Framework*, [4], contains a brief review of the basic features of these notions. It also develops a general framework for applications of these notions, consisting of formulas and theorems which were used in the second part of this paper, *Chromatic Derivatives and Approximations in Practice - Part II: Non-Uniform Sampling, Zero-Crossings Reconstruction and Denoising* [5], in which we presented four case studies of practical applications of these notions in design of novel signal processing algorithms.

Aleksandar Ignjatović and Gabriele Keller are with the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia; E-mail: {ignjat, g.keller}@cse.unsw.edu.au; Chamith Wijenayake is with the School of Electrical Engineering, University of New South Wales, Sydney, Australia; E-mail: c.wijenayake@unsw.edu.au.

In this paper we present yet another, more sophisticated application of chromatic derivatives and approximations – a design of a novel frequency estimation algorithm, which we call the Continuous Time MUSIC algorithm (CT-MUSIC). As we demonstrate by our extensive tests, such an algorithm significantly outperforms the original MUSIC algorithm for frequency estimation [6, Ch. 4.5], [7] in terms of both super-resolution and noise robustness. As its name suggests, the CT-MUSIC algorithm can be seen as a continuous time version of the standard MUSIC algorithm. In particular, while the standard MUSIC algorithm operates over discrete samples of a continuous time signal, the proposed CT-MUSIC algorithm is based on chromatic derivatives and associated chromatic approximations of a continuous time signal. *The main merit of chromatic derivatives and approximations is that they allow full use of the standard, continuous time calculus notions, such as integral as well as differential operators (which can be of very high orders,* > 100*).* For the sake of conciseness, in this paper we do not repeat general definitions and statements about chromatic derivatives, expansions and approximations, referring the reader to the first part of this paper [4].

II. Motivation for the Approach

The roots of the MUSIC algorithm for frequency estimation of real sinusoids or complex exponentials lie in the work of Prony [8], [9] and Pisarenko [6], [10], [11], which we present very briefly and in a very simplified manner, suitable for explaining our algorithm and how it is related to the original MUSIC algorithm.

A. The methods of Prony, Pisarenko and the MUSIC algorithm

Let a signal $s(t)$ be a linear combination of n complex exponentials, i.e., of the form

$$s(t) = \sum_{k=1}^{n} A_k \mathrm{e}^{j(\omega_k t + \phi_k)} \tag{1}$$

where real numbers A_k, ω_k and ϕ_k are the amplitude, the frequency and the phase of the k^{th} component, respectively; we assume that $A_k > 0$ and $-\pi < \omega_k, \phi_k < \pi$ for $1 \leq k \leq n$.

Assume further that we have a sequence of samples $s(t+m)$ of such a signal, taken at consecutive instants a unit distance apart, starting with an instant t; note that, given our assumption on frequencies ω_k, such samples satisfy the Nyquist criterion.

Let us form a linear combination of $n+1$ such consecutive samples with coefficients $c_0, \ldots, c_n$ to be specified below. Using (1), after some simplification we obtain

$$\sum_{m=0}^{n} c_m s(t+m) = \sum_{k=1}^{n} \left(\sum_{m=0}^{n} c_m \left(\mathrm{e}^{j\,\omega_k} \right)^m \right) A_k \mathrm{e}^{j(\omega_k t + \phi_k)}. \tag{2}$$

Consider now a polynomial $P(z)$ with the leading coefficient 1, given by the product $P(z) = (z - \mathrm{e}^{j\,\omega_1}) \ldots (z - \mathrm{e}^{j\,\omega_n})$; if we choose $c_0, \ldots, c_n$ to be the coefficients of this polynomial, such that

$$\sum_{m=0}^{n} c_m z^m = \prod_{k=1}^{n} (z - \mathrm{e}^{j\,\omega_k}),$$

then, since the right hand side of (2) is of the form $\sum_{k=1}^{n} P(\mathrm{e}^{j\,\omega_k}) A_k \mathrm{e}^{j(\omega_k t + \phi_k)}$ and since $\mathrm{e}^{j\,\omega_1}, \ldots, \mathrm{e}^{j\,\omega_n}$ are the roots of $P(z)$, the righthand side of (2) will be equal to zero for all real t, and vice versa: since complex exponentials with distinct frequencies are linearly independent functions, if the right side of (2) is equal to zero for all real t, then $\mathrm{e}^{j\,\omega_k}$ must be the roots of the polynomial $P(z)$. Consequently, in order to find the frequencies ω_k, it is enough to find coefficients $c_0, \ldots, c_n$ such that for all t,

$$\sum_{m=0}^{n} c_m s(t+m) = 0, \tag{3}$$

and then find the roots of the associated algebraic equation $\sum_{m=0}^{n} c_m z^m = 0$; such roots lie on the unit circle and their arguments are the frequencies sought.

To find a non-zero vector $\mathbf{c} = (c_0, \ldots, c_n)^\top$ such that (3) holds for all t, we instantiate (3) from $t = 1$ to $t = n$, and if $2n$ samples $s(1), s(2), \ldots, s(2n)$ of the signal $s(t)$ are available, we obtain a system of linear equations in unknown coefficients $c_0, \ldots, c_n$ of the form

$$\begin{pmatrix} s(1) & s(2) & \ldots & s(n+1) \\ s(2) & s(3) & \ldots & s(n+2) \\ \ldots & \ldots & \ldots & \ldots \\ s(n-1) & s(n) & \ldots & s(2n-1) \\ s(n) & s(n+1) & \ldots & s(2n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \mathbf{0} \tag{4}$$

which, in absence of any noise, we can solve exactly. This is the *Prony method*.

In the presence of noise (3) will not hold exactly; to deal with this problem, we make the system of equations (4) overdetermined. Thus, we will assume that we have $N > 2n$ equidistant samples of the signal $s(t)$. Let us again first consider the noise-free case, i.e., assume that $s(t)$ is as in (1) and let us form the following Hankel matrix $\mathbf{M}_p^s$ of size $(N-n) \times (n+1)$:

$$\mathbf{M}_p^s = \begin{pmatrix} s(1) & s(2) & \ldots & s(n+1) \\ s(2) & s(3) & \ldots & s(n+2) \\ s(3) & s(4) & \ldots & s(n+3) \\ \ldots & \ldots & \ldots & \ldots \\ s(N-n) & s(N-n+1) & \ldots & s(N) \end{pmatrix} \tag{5}$$

If no noise were present, equation (3) would imply that the rank of this matrix is equal to n. Consequently, if we consider the singular value decomposition of this matrix, $\mathbf{M}_p^s = \mathbf{U \Sigma V}^*$, the diagonal of Σ would consist of $n+1$ singular values of $\mathbf{M}_p^s$, out of which there would be n nonzero singular values, while the last, the smallest singular value would be equal to 0.[1] Matrices $\mathbf{U}$ and $\mathbf{V}$ are both unitary, i.e., their columns represent a set of orthonormal vectors, called the left (the right) singular vectors, respectively. Let $\mathbf{v}_{n+1}$ be the rightmost singular vector which corresponds to the zero singular value of $\mathbf{M}_p^s$. Then, since $\mathbf{V}$ is unitary, $\mathbf{V}^* \mathbf{v}_{n+1} = (0, 0, \ldots, 0, 1)^\top$ and, since the last entry on the diagonal of $\mathbf{\Sigma}$ is zero, this is easily seen to imply that $\mathbf{M}_p^s \mathbf{v}_{n+1} = \mathbf{U \Sigma V}^* \mathbf{v}_{n+1} = \mathbf{0}$. Thus, the components of the rightmost singular vector $\mathbf{v}_{n+1}$ corresponding to the singular value zero are the required coefficients of a linear combination of the columns of the matrix $\mathbf{M}_p^s$ which is equal to the zero vector.

Assume now that we have $N > 2n$ equidistant samples $f(1), \ldots, f(N)$ of a signal $f(t) = s(t) + \nu(t)$ which is a sum of a signal $s(t)$ as in (1) and noise $\nu(t)$. We can still form a Hankel matrix $\mathbf{M}_p^f$ obtained from matrix $\mathbf{M}_p^s$ by replacing samples of $s(t)$ by the corresponding samples of $f(t)$.

Due to the presence of noise, $\mathbf{M}_p^f$ will generally have a full rank of $n+1$, and thus all singular values of $\mathbf{M}_p^f$ will be non-zero. We now take the rightmost singular vector, which corresponds to the smallest singular value, as an approximation of the rightmost singular vector if no noise were present, and its components $\mathbf{v}_{n+1}(m+1)$ as approximations of the values of c_m for which (3) holds.

As is well known, the right singular vectors of matrix $\mathbf{M}_p^f$ are the eigenvectors of the product matrix $\mathbf{A}_p^f = (\mathbf{M}_p^f)^* \mathbf{M}_p^f$. Note that, in our particular case, $1/N\, \mathbf{A}_p^f$ is just the auto-covariance matrix of the samples of the noisy signal. Since matrix $\mathbf{A}_p^f$ is of size $(n+1) \times (n+1)$ and since N is usually much larger than n, $\mathbf{A}_p^f$ is of much smaller size than $\mathbf{M}_p^f$. Consequently, finding the eigendecomposition of $\mathbf{A}_p^f$ is a computationally lighter task than finding the singular value decomposition of $\mathbf{M}_p^f$.[2] In this way we obtain the *Pisarenko frequency estimation algorithm*: compute the matrix $\mathbf{A}_p^f = (\mathbf{M}_p^f)^* \mathbf{M}_p^f$ and obtain its eigenvalue decomposition $\mathbf{A}_p^f = \mathbf{Q \Lambda Q}^*$. Take the rightmost eigenvector $\mathbf{q}_{n+1}$ which corresponds to the smallest eigenvalue of $\mathbf{A}_p^f$ and solve the associated algebraic equation $\sum_{m=0}^{n} \mathbf{q}_{n+1}(m+1) z^m = 0$; the arguments of its n roots are taken as estimates of the frequencies ω_k of the complex exponentials which are the n components of the signal $s(t)$.

It is well known that the Pisarenko method is not very noise robust and that it often has quite a poor performance. The MUSIC algorithm is a generalization of the Pisarenko method which significantly improves its noise robustness; it is essentially an averaging procedure of multiple estimates of the frequencies obtained by the Pisarenko method.

To obtain the MUSIC algorithm, let us again first consider the noise-free case and let $\mathbf{M}_m^s$ be the Hankel matrix of shifted

[1] Singular values on the diagonal of $\mathbf{\Sigma}$ are always non-negative reals and are usually ordered in a descending order; thus, the smallest singular value is the rightmost one on the diagonal of $\mathbf{\Sigma}$.

[2] However, this benefit is often offset by the cost of the computation of the matrix product $\mathbf{A}_p^f = (\mathbf{M}_p^f)^* \mathbf{M}_p^f$.

consecutive samples with a possibly larger number of columns $K \geq n+1$, thus of size $(N-K+1)\times K$,

$$\mathbf{M}^s_m = \begin{pmatrix} s(1) & s2) & \dots & s(K) \\ s(2) & s(3) & \dots & s(K+1) \\ s(3) & s(4) & \dots & s(K+2) \\ \dots & \dots & \dots & \dots \\ s(N\text{–}K\!+1) & s(N\text{–}K\!+\!2) & \dots & f(N) \end{pmatrix} \quad (6)$$

and again consider its singular value decomposition, $\mathbf{M}^s_m = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$. In absence of noise (3) again implies that the rank of this matrix would be equal to n and thus only n singular values on the diagonal of $\boldsymbol{\Sigma}$ would be non-zero. Just as in the case of the Pisarenko method, this would imply that for $K-n$ right singular vectors $\mathbf{v}_i$, $(n < i \leq K)$, which correspond to $K-n$ zero singular values of $\mathbf{M}^s_m$, we would have $\mathbf{M}^s_m\mathbf{v}_i = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*\mathbf{v}_i = 0$. Thus, each of the right singular vectors $\mathbf{v}_i$, $n < i \leq K$, produces an equation of the form

$$\sum_{m=0}^{K-1} \mathbf{v}_i(m+1) f(t+m) = 0$$

that holds for samples $f(t),\dots,f(t+K-1)$ for all integers $1 \leq t \leq N-K+1$; consequently, in the absence of any noise, for every $n < i \leq K$ and all $1 \leq t \leq N-K+1$, (2) would imply

$$\sum_{m=0}^{K-1} \mathbf{v}_i(m+1) f(t+m) = \sum_{k=1}^{n} A_k \mathrm{e}^{j\,\phi_k} \left(\sum_{m=0}^{K-1} \mathbf{v}_i(m+1) \left(\mathrm{e}^{j\,\omega_k}\right)^m \right) \mathrm{e}^{j\,\omega_k t} = 0. \quad (7)$$

If $N-K+1 \geq n$, since the Vandermonde matrix $\mathbf{V} = ((e^{j\,\omega_k})^t \;:\; 1 \leq k,t \leq n)$ is always nonsingular if all ω_k are distinct, equation (7) would imply that $\mathrm{e}^{j\,\omega_k}$ must be among the $K-1$ many roots of each of the polynomials $P_i(z)$,

$$P_i(z) = \sum_{m=0}^{K-1} \mathbf{v}_i(m+1) z^m, \quad n < i \leq K. \quad (8)$$

Thus, polynomials $P_i(z)$ for all $n < i \leq K$ share the same n roots $\mathrm{e}^{j\,\omega_1},\dots,\mathrm{e}^{j\,\omega_n}$ which belong to the unit circle, and each of the polynomials $P_i(z)$ has additional $K-1-n$ roots generally not belonging to the unit circle and which are different and specific to each polynomial $P_i(z)$. Thus, if we consider the real valued function $F(z)$ given by[3]

$$F(z) = \sum_{i=n+1}^{K} |P_i(z)|^2 = \sum_{i=n+1}^{K} P_i(z)\overline{P_i(z)}, \quad (9)$$

then this function will have n of its zeros lying on the unit circle, namely $\mathrm{e}^{j\,\omega_k}$ for $1 \leq k \leq n$ and the arguments of these zeros are the frequencies sought.

Clearly, the above no longer holds in the presence of noise. If $f(t) = s(t) + \nu(t)$ where $s(t)$ is of the form given by the right hand side of (1) and $\nu(t)$ is noise, then we can form matrix $\mathbf{M}^f_m$ of the same form as matrix $\mathbf{M}^s_m$ but with the samples the noisy signal $f(t)$ in place of the corresponding samples of the noise-free signal $s(t)$. However, such a matrix $\mathbf{M}^f_m$ will generally be of full rank, but we can take its $K-n$ right singular vectors $\mathbf{v}_i$ which correspond to the smallest $K-n$ singular values of $\mathbf{M}^f_m$ as an approximation of such singular vectors which would correspond to the noise-free matrix $\mathbf{M}^s_m$.

We can again form the corresponding polynomials $P_i(z)$ given by (8) and function $F(z)$ given by (9). Note that $F(z)$ is not a polynomial, due to the presence of the modulus (or the complex conjugation) function. Due to the effects of noise, no roots of polynomials $P_i(z)$ might belong to the unit circle, and also these polynomials might not share the exact same n roots. Thus, function $\Phi(\omega) = F(\mathrm{e}^{j\,\omega}) = \sum_{i=n+1}^{K} \left|P_i\left(\mathrm{e}^{j\,\omega}\right)\right|^2$ might not have any real zeros belonging to the interval $[-\pi,\pi]$. Since $\Phi(\omega) \geq 0$, the MUSIC algorithm thus instead searches for the n values of ω which lie in the interval $[-\pi,\pi]$, where $\Phi(\omega)$ attains n smallest local minima, or, equivalently but numerically more conveniently, for n largest local peaks of function $R(\omega)$,

$$R(\omega) = \frac{1}{\sum_{i=n+1}^{K} \left|P_i\left(\mathrm{e}^{j\,\omega}\right)\right|^2}. \quad (10)$$

To avoid such a numerical search, the root-MUSIC algorithm instead uses the fact that complex numbers z which are close to the unit circle satisfy $\overline{z} \approx z^{-1}$ and thus instead explicitly solves the following equation, reducible to an algebraic (i.e., polynomial) equation[4]

$$\sum_{i=n+1}^{K} P_i(z)\overline{P_i}(z^{-1}) = 0. \quad (11)$$

It is easy to see that if z_i is a root of (11), then so is $\overline{z_i^{-1}}$; thus the roots of (11) come in pairs of the form $\{\rho_i \mathrm{e}^{j\,\omega_i},\ \rho_i^{-1}\mathrm{e}^{j\,\omega_i}\}$. The root-MUSIC algorithm picks n pairs of such roots that lie closest to the unit circle; the arguments of these pairs are taken as the estimates of the frequencies ω_k, $1 \leq k \leq n$, of the n components of $s(t)$.

B. The heuristics behind our method

As we have seen, the above methods all rely on the crucial fact given by the following theorem, proven above.

Theorem II.1. *If a signal $s(t)$ is a linear combination of n complex exponentials as in* (1)*, then for every real t signal $s(t)$ satisfies a finite difference equation with $n+1$ terms of the form given by* (3)*:*

$$\sum_{m=0}^{n} c_m s(t+m) = 0.$$

Thus, in a sense, in the above algorithms we are forcing our signal model to "look like" a sum of n complex exponentials by forcing it, at the N many discrete sampling instants available, to approximately satisfy (3) which such a waveform ought to satisfy.

The main idea of our approach is to make our model more selective, by forcing it to look like a sum of n complex

[3]Here $\overline{z}$ denotes the complex conjugation.

[4]$\overline{P_i}(z)$ denotes the polynomial obtained from the polynomial $P_i(z)$ by taking the complex conjugates of the coefficients of $P_i(z)$.

exponentials as a *continuous waveform* not only at the discrete sampling instants, but also in a *continuous time interval* spanned by such sampling instants.

Thus, we will first use chromatic approximations to produce an initial approximation of the continuous waveform of the (noisy) signal within the interval spanned by the samples available, we will then reshape such an approximation, making it closer to being a sum of n complex exponentials and then rely on the following continuous time counterpart of Theorem II.1:

Theorem II.2. *If a signal $s(t)$ is a linear combination of n complex exponentials as in (1), then for every real t signal $s(t)$ satisfies a homogeneous linear differential equation with constant coefficients of order n, of the form*

$$\sum_{m=0}^{n} c_m s^{(m)}(t) = 0. \tag{12}$$

Let us start again by considering the noise-free case. Thus, assume that $s(t)$ is of the form (1), and let us form a linear combination of the first n derivatives of $s(t)$ with the coefficients c_k to be determined below; then a straightforward calculation shows that

$$\sum_{m=0}^{n} c_m s^{(m)}(t) = \sum_{k=0}^{n} \left(\sum_{m=0}^{n} c_m (j\omega_k)^m \right) A_k \mathrm{e}^{j(\omega_k\, t+\phi_k)}. \tag{13}$$

Note that (13) is, in a sense, a continuous time analog of (2). This time we consider a polynomial with real coefficients, $P(\omega) = \prod_{k=1}^{n}(\omega - \omega_k)$; let $P(\omega) = \sum_{m=0}^{n} a_m \omega^m$ be the coefficient representation of $P(\omega)$. If we choose $c_m = (-j)^m a_m$, we obtain

$$\sum_{m=0}^{n} c_m s^{(m)}(t) = \sum_{k=0}^{n} A_k \left(\sum_{m=0}^{n} a_m \omega_k^m \right) \mathrm{e}^{j(\omega_k\, t+\phi_k)} \tag{14}$$

$$= \sum_{k=0}^{n} A_k P(\omega) \mathrm{e}^{j(\omega_k\, t+\phi_k)} \tag{15}$$

Since ω_k are the roots of $P(\omega)$, the righthand side of (13) is equal to zero for all t; consequently, the corresponding linear combination $\sum_{m=0}^{n} c_m s^{(m)}(t)$ is also equal to zero for all t. As before, the converse is also true: if for some coefficients c_k the lefthand side of (13) is equal to zero for all real t, then ω_k must be the roots of the corresponding polynomial $P(\omega) = \sum_{k=0}^{n} a_k \omega^k$. This implies that, in order to find frequencies ω_k, it is enough to find coefficients c_k such that (12) holds for all t, and then solve the corresponding algebraic equation $\sum_{m=0}^{n} j^m c_m \omega^m = 0$; the roots of this equation are real and equal to the frequencies sought.

However, in order to make the above approach practically viable, we must replace the "standard" derivatives $s^{(k)}(t)$ with the chromatic derivatives which can be robustly approximated in presence of noise, even for very high orders $k > 100$.

Next we present the necessary theoretical details behind our Continuous Time MUSIC algorithm.

III. Theoretical Basis of CT-MUSIC Algorithm.

Assume again that a continuous time signal $f(t)$ is of the form $f(t) = s(t) + \nu(t)$, where $s(t)$ is as in (1) and $\nu(t)$ is a zero mean wide-sense stationary (WSS) π-band-limited noise; as usual, we will assume that the noise is circularly distributed, i.e., that the real and the imaginary components of $\nu(t)$ are independent and identically distributed random variables. While this is not absolutely necessary, (see footnote 11 in [5]), it allows us to conclude that the power spectral density (PSD) $S(\omega)$ of the noise is an even function, i.e., that $S(-\omega) = S(\omega)$ for all $\omega \in [-\pi, \pi]$, which considerably simplifies the matters. We also make the usual assumptions on the auto-covariance sequence $r(k) = E\left[\nu(t)\overline{\nu(t+k)}\right]$ of the noise $\nu(t)$, namely that $r(k)$ is square summable, $\sum_{k=0}^{\infty} r(k)^2 < \infty$, as well as that it decays sufficiently rapidly, so that $\lim_{N\to\infty} {}^1\!/\!{}_N \sum_{k=0}^{N} k\, r(k) = 0$. These assumptions imply that

$$S(\omega) = \sum_{k=-\infty}^{\infty} r(k) \mathrm{e}^{-j\,\omega\, k} \tag{16}$$

is a non-negative, even, square integrable function on the interval $[-\pi, \pi]$; see [6]. If we normalize $S(\omega)$ so that $\int_{-\pi}^{\pi} S(\omega)\mathrm{d}\omega = 1$, then $S(\omega)$ can be used as a weight function for a family of symmetric polynomials $p_n^\nu(\omega)$, orthonormal on $[-\pi, \pi]$,

$$\int_{-\pi}^{\pi} p_m^\nu(\omega) p_n^\nu(\omega) S(\omega)\mathrm{d}\omega = \delta(m-n). \tag{17}$$

If the noise is white then $S(\omega) = 1/(2\pi)$ and the resulting polynomials $p_n^L(\omega)$ are of the form

$$p_n^L(\omega) = \sqrt{2n+1} P_n^L(\omega/\pi) \tag{18}$$

where $P_n^L(x)$ are the usual Legendre polynomials; see [4].

Let $\mathcal{K}_t^n$ be the chromatic derivatives corresponding to $p_n^\nu(\omega)$; thus, $\mathcal{K}_t^n = (-j)^n p_n^\nu(j\, d/dt)$ where $p_n^\nu(j\, d/dt)$ is obtained by replacing the powers ω^k with $j^k d^k/dt^k$. Operators $\mathcal{K}^n$ are defined in such a way in order to satisfy[5]

$$\mathcal{K}^n[\mathrm{e}^{j\,\omega t}] = j^n p_n^\nu(\omega)\mathrm{e}^{j\,\omega t}; \tag{19}$$

as explained in [4], their values can be computed with very high accuracy (from a sufficient number of discrete samples of the signal) even for high orders $n > 100$.

*To perform a theoretical analysis, we will first **assume** that the values of chromatic derivatives of the noisy signal are available (for example, as outputs of an analog or digital filterbank); we will address this issue later.*

Starting again with the noise-free case, for a signal $s(t)$ of the form (1), equation (19) implies the following continuous time counterpart of (2):

$$\sum_{m=0}^{n} c_m\, \mathcal{K}^m[s](t) = \sum_{k=0}^{n} A_k \left(\sum_{m=0}^{n} c_m j^m p_m^\nu(\omega_k) \right) \mathrm{e}^{j\,\omega_k\, t}. \tag{20}$$

[5]Differentiation will be always performed with respect to the time variable t; thus, for simplicity of notation we drop the subscript t in $\mathcal{K}_t^n$.

Let $a_0, \ldots, a_n$ be the coefficients of the real polynomial $P(\omega) = \prod_{k=1}^{n}(\omega - \omega_k)$ in the basis consisting of orthonormal polynomials $p_n^{\nu}(\omega)$, i.e., such that $P(\omega) = \prod_{k=1}^{n}(\omega - \omega_k) = \sum_{m=0}^{n} a_m p_m^{\nu}(\omega)$, and let $c_m = (-j)^m a_m$; then (20) implies

$$\sum_{m=0}^{n} c_m \, \mathcal{K}^m[s](t) = \sum_{k=0}^{n} A_k P(\omega_k) \mathrm{e}^{j\,\omega_k\,t}.$$

Since $P(\omega_k) = 0$ for $1 \leq k \leq n$, the right hand side of (20) is zero for all t, and thus so is the left hand side:

$$\sum_{m=0}^{n} c_m \, \mathcal{K}^m[s](t) = 0. \tag{21}$$

Vice versa, as before, if (21) holds for $t = 1, \ldots, n$ then, since the matrix $\left((\mathrm{e}^{j\,\omega_k})^t : 1 \leq t \leq n;\ 1 \leq k \leq n\right)$ is non-singular, the polynomial $\sum_{m=0}^{n} c_m j^m p_m^{\nu}(\omega)$ must be zero for $\omega = \omega_1, \ldots, \omega_k$. Thus, to find frequencies $\omega_1, \ldots, \omega_n$, it is enough to find the coefficients $c_0, c_1, \ldots, c_n$ such that (21) holds for $t = 1, \ldots, n$ and then solve the associate algebraic equation

$$\sum_{m=0}^{n} c_m j^m p_m^{\nu}(\omega) = 0. \tag{22}$$

All the roots of this equation are real and equal to the frequencies sought. Coefficients $c_0, \ldots, c_n$ can be obtained by substituting $t = 1, \ldots, n$ in (21) thus obtaining the following system of equations:

$$\begin{pmatrix} \mathcal{K}^0[s](1) & \mathcal{K}^1[s](1) & \ldots & \mathcal{K}^n[s](1) \\ \mathcal{K}^0[s](2) & \mathcal{K}^1[s](2) & \ldots & \mathcal{K}^n[s](2) \\ \ldots & \ldots & \ldots & \ldots \\ \mathcal{K}^0[s](n) & \mathcal{K}^1[s](n) & \ldots & \mathcal{K}^n[s](n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \mathbf{0}. \tag{23}$$

The system of equations (23) can be seen as a continuous time analogue of (4); due to presence of genuine, continuous time differential operators in (23), as opposed to discrete samples present in (4), the method just described could be seen as a *continuous time version of the Prony method*.

We now form the following matrix $\boldsymbol{\mathcal{M}}_p^s$ of size $N \times (n+1)$, which can be seen as a continuous time analogue of matrix M_p^s given by (5):

$$\boldsymbol{\mathcal{M}}_p^s = \begin{pmatrix} \mathcal{K}^0[s](1) & \mathcal{K}^1[s](1) & \ldots & \mathcal{K}^n[s](1) \\ \mathcal{K}^0[s](2) & \mathcal{K}^1[s](2) & \ldots & \mathcal{K}^n[s](2) \\ \ldots & \ldots & \ldots & \ldots \\ \mathcal{K}^0[s](N) & \mathcal{K}^1[s](N) & \ldots & \mathcal{K}^n[s](N) \end{pmatrix} \tag{24}$$

Equation (21) implies that the rank of this matrix is equal to n. If we consider its singular value decomposition $\boldsymbol{\mathcal{M}}_p^s = \mathbf{U\Sigma V}^*$, we would have, just as in the case of the Pisarenko method, that the last singular value on the diagonal of $\boldsymbol{\Sigma}$ is equal to 0. This in turn implies that, if $\mathbf{v}_{n+1}$ is the right singular vector which corresponds to the singular value 0, then $\boldsymbol{\mathcal{M}}_p^s \mathbf{v}_{n+1} = \mathbf{U\Sigma V}^* \mathbf{v}_{n+1} = 0$, and we again obtain that the values of the constants in (21) can be taken to be the components of the rightmost singular vector of $\boldsymbol{\mathcal{M}}_p^s$. Equation (22) has all roots real and equal to the frequencies sought.

In the presence of noise the argument presented above no longer holds, but we can form a matrix $\boldsymbol{\mathcal{M}}_p^f$ analogous to matrix $\boldsymbol{\mathcal{M}}_p^s$, with the noisy signal $f(t) = s(t) + \nu(t)$ in place of signal $s(t)$. Since $f(t)$ is now a stochastic process, its derivative is taken to be the output of the corresponding LTI differentiation filter with input $f(t)$; consequently, expression $\mathcal{K}^m[f](t)$ stands for the output of the LTI filter $\mathcal{K}^m$ for input $f(t)$. Since $\nu(t)$ is also assumed to be π-band-limited, such a filter can be assumed to have a transfer function equal to $j^n p_n^{\nu}(\omega)$ for $\omega \in [-\pi, \pi]$ and zero outside such an interval. Since the output of an LTI filter for any WSS input is also WSS, $\mathcal{K}^n[\nu](t)$ is also a WSS process for every n.

Due to noise, matrix $\boldsymbol{\mathcal{M}}_p^f$ has a full rank of $n+1$; however, we can take its rightmost singular vector which corresponds to the smallest singular value of $\boldsymbol{\mathcal{M}}_p^f$ as an approximation of what such a vector would be in the absence of noise and solve the equation obtained from (22):

$$\sum_{m=0}^{n} \mathbf{v}_{n+1}(m+1) j^m p_m^{\nu}(\omega) = 0. \tag{25}$$

The roots of this equation are either real or have small imaginary parts, and we take the real parts of these roots as the estimates of the frequencies sought.

As before, recall that the right singular vectors of $\boldsymbol{\mathcal{M}}_p^f$ are also the right eigenvectors of the auto-covariance matrix of the chromatic derivatives, i.e., of matrix $\boldsymbol{\mathcal{A}}_p^f = {}^1\!/\!{}_N (\boldsymbol{\mathcal{M}}_p^f)^* \boldsymbol{\mathcal{M}}_p^f$. In this way we obtain a *continuous time version of the Pisarenko method*: we compute the matrix $\boldsymbol{\mathcal{A}}_p^f = {}^1\!/\!{}_N (\boldsymbol{\mathcal{M}}_p^f)^* \boldsymbol{\mathcal{M}}_p^f$, find its eigen-decomposition $\boldsymbol{\mathcal{A}}_p^f = \mathbf{Q\Lambda Q}^*$ and take its rightmost eigenvector $\mathbf{q}_{n+1}$ which corresponds to the smallest eigenvalue λ_{n+1}; the components of this vector are the coefficients c_m such that (21) holds in an approximate sense, and consequently we solve the corresponding algebraic equation

$$\sum_{m=0}^{n} \mathbf{q}_{n+1}(m+1) j^m p_m^{\nu}(\omega) = 0 \tag{26}$$

and take the real parts of its roots as the estimates of the frequencies sought.

To obtain a continuous time counterpart of the MUSIC algorithm we consider matrices $\boldsymbol{\mathcal{M}}_m^f$ and $\boldsymbol{\mathcal{M}}_m^s$ with a larger number of columns, of size $N \times (K+1)$, where $K > n$, defined as

$$\boldsymbol{\mathcal{M}}_m^f = \begin{pmatrix} \mathcal{K}^0[f](1) & \mathcal{K}^1[f](1) & \ldots & \mathcal{K}^K[f](1) \\ \mathcal{K}^0[f](2) & \mathcal{K}^1[f](2) & \ldots & \mathcal{K}^K[f](2) \\ \ldots & \ldots & \ldots & \ldots \\ \mathcal{K}^0[f](N) & \mathcal{K}^1[f](N) & \ldots & \mathcal{K}^K[f](N) \end{pmatrix} \tag{27}$$

and similarly for matrix $\boldsymbol{\mathcal{M}}_m^s$ with $s(t)$ replacing $f(t)$. Just as in the case of the MUSIC algorithm, in the absence of any noise, matrix $\boldsymbol{\mathcal{A}}_m^s = {}^1\!/\!{}_N (\boldsymbol{\mathcal{M}}_m^s)^* \boldsymbol{\mathcal{M}}_m^s$ has a rank equal to n, and consequently only n of its eigenvalues are non-zero. Thus, in this case we would have $\boldsymbol{\mathcal{A}}_m^s \mathbf{v}_k = \mathbf{0}$ for $K+1-n$ eigenvectors $\mathbf{v}_k$, $n < k \leq K+1$ which correspond to the zero eigenvalues.

In the presence of noise, matrices $\boldsymbol{\mathcal{M}}_m^f$ and $\boldsymbol{\mathcal{A}}_m^f$ have full ranks; however, we can choose $K+1-n$ rightmost eigenvectors which correspond to the $K+1-n$ smallest eigenvalues of $\boldsymbol{\mathcal{A}}_m^f$ as approximations of such eigenvectors for the noise-free matrix $\boldsymbol{\mathcal{A}}_m^s$. We can now perform a similar averaging procedure as is done in the course of the MUSIC algorithm. Thus, we consider the corresponding algebraic equations of the kind given by (22):

$$\sum_{m=0}^{K} \mathbf{v}_k(m+1) j^m p_m^{\nu}(\omega) = 0, \quad n < k \leq K, \tag{28}$$

which, due to the effects of the noise, no longer have to have n common real roots. For that reason we consider a real valued function

$$\mathcal{F}(\omega) = \sum_{k=n+1}^{K+1} \left| \sum_{m=0}^{K} \mathbf{v}_k(m+1) j^m p_m^{\nu}(\omega) \right|^2 = \tag{29}$$

$$\sum_{k=n+1}^{K+1} \sum_{m=0}^{K} \sum_{r=0}^{K} \mathbf{v}_k(m+1) j^m p_m^{\nu}(\omega) \overline{\mathbf{v}_k(r+1)} (-j)^r p_r^{\nu}(\omega)$$

and choose for the frequency estimates ω_k the n values of ω that correspond to n smallest local minima of $\mathcal{F}(\omega)$. Note that, unlike function $\Phi(\omega)$ of the standard MUSIC algorithm, function $\mathcal{F}(\omega)$ is a polynomial of a real variable, so its extrema can be found by explicitly solving the algebraic equation $\mathcal{F}'(\omega) = 0$, then choosing such roots ω_i of this equation for which $\mathcal{F}''(\omega_i)$ is positive and finally among these roots choosing n of them for which function $\mathcal{F}(\omega_i)$ achieves n smallest values. This is the foundation of our algorithm which we call *the Continuous Time MUSIC algorithm* or CT-MUSIC for short.

Note that in the above considerations we have not used the fact that the noisy signal $f(t)$ is uniformly sampled; thus, the above considerations hold true if the values of $\mathcal{K}^m[s](t)$ at integers $t = 1, \ldots, N$ are replaced by such values at any N instants, regardless of their spacing, uniform or non uniform; moreover, the exact timing of sampling instants do not even have to be recorded at all. In particular, if the front end is implemented as an analog filterbank, our method is entirely immune to clock jitter of A/D converters sampling such a filterbank.

IV. "Matched Filter" Case for Adaptive CT-MUSIC

We also note that so far we have not used our assumption that the operators $\mathcal{K}^n$ were chosen to correspond to the polynomials orthonormal with respect to the PSD of the noise. In fact, the above method is applicable if we take the polynomials which are orthonormal with respect to the PSD of the white noise, i.e., for $S(\omega) = 1/(2\pi)$, even if the actual noise present is not white. Such a choice results in the *non-adaptive mode of operation of the CT-MUSIC algorithm* which we also present and test. However, as our tests clearly demonstrate, CT-MUSIC performs much better if the operators $\mathcal{K}^n$ are "matched" to the PSD of the noise, i.e., if they correspond to polynomials $p_n^{\nu}(\omega)$ orthonormal with respect to the (normalized) PSD $S(\omega)$ of the noise $\nu(t)$, in which case we denote them by $\mathcal{K}_\nu^n$; see Section VII in [4] for details on matched filter chromatic derivatives. This leads to two more modes of operation of the CT-MUSIC algorithm, the *adaptive mode* which uses a PSD of noise obtained by an estimation of noise prior to frequency estimation and the *adaptive-learning mode* which obtains such a PSD estimate dynamically by a background process during the course of frequency estimation and tracking.

To explain the adaptive mode of operation of the CT-MUSIC algorithm we generalize Theorem 5.1 from the second part of this paper [5] to complex zero mean circular WSS noise; the proof remains essentially the same.

Theorem IV.1. *Let $\nu(t)$ be a zero mean circular WSS noise with RMS value equal to ρ and let a non-negative even function $S(\omega)$ be the PSD of $\nu(t)$. We normalize $S(\omega)$ such that $\int_{-\pi}^{\pi} S(\omega)\mathrm{d}\omega = 1$. Let $p_n^{\nu}(\omega)$ be polynomials orthonormal with respect to such normalized $S(\omega)$, i.e., such that*

$$\int_{-\pi}^{\pi} p_n^{\nu}(\omega)\, p_m^{\nu}(\omega) S(\omega)\mathrm{d}\omega = \delta(m-n).$$

Let also operators $\mathcal{K}_\nu^n = (-j)^n p_n^{\nu}(j\, d/dt)$ correspond to such polynomials $p_n^{\nu}(\omega)$. Then

$$E\left[\mathcal{K}_\nu^k[\nu(t)]\, \overline{\mathcal{K}_\nu^m[\nu(t)]}\right] = \rho^2 \delta(k-m). \tag{30}$$

Thus, in such a "matched filter" case, chromatic derivatives of the noise of all orders have the same mean square value, equal to the mean square value of the noise and are pairwise uncorrelated.

We now prove the following theorem.

Theorem IV.2. *Let $\nu(t)$ and $\mathcal{K}_\nu^n$ be as in the previous theorem. Then for all n the chromatic derivatives $\mathcal{K}_\nu^n[\nu](t)$ are zero mean.*

Proof: Since $\mathcal{K}_\nu^n$ is a linear operator with transfer function $j^n p_n^{\nu}(\omega)$, the PSD $S_n(\omega)$ of $\mathcal{K}_\nu^n[\nu](t)$ is given by

$$S_n(\omega) = |j^n p_n^{\nu}(\omega)|^2 S(\omega) = p_n^{\nu}(\omega)^2 S(\omega).$$

Thus,

$$E\left[\mathcal{K}_\nu^n[\nu(t)]\right] = \int_{-\pi}^{\pi} \omega\, p_n^{\nu}(\omega)^2\, S(\omega)\mathrm{d}\omega. \tag{31}$$

Since $p_n^{\nu}(\omega)$ are even functions for even n and odd functions for odd n, polynomials $p_n^{\nu}(\omega)^2$ are even functions for all n; thus, since $S(\omega)$ is also even, $\omega\, p_n^{\nu}(\omega)^2\, S(\omega)$ is an odd function. Consequently, the right side of (31) is equal to zero. ■

Theorem IV.1 and Theorem IV.2 have the following important consequence.

Theorem IV.3. *Let $\nu(t)$ be noise with properties as in Theorem IV.1; let also $s(t)$ be any band-limited signal, t any instant in time and k and m any two natural numbers. Then the noisy signal $f(t) = s(t) + \nu(t)$ satisfies*

$$E\left[\mathcal{K}_\nu^k[f(t)]\, \overline{\mathcal{K}_\nu^m[f(t)]}\right] = \begin{cases} \mathcal{K}_\nu^k[s(t)]\, \overline{\mathcal{K}_\nu^m[s(t)]} & \text{if } k \neq m \\ |\,\mathcal{K}_\nu^k[s(t)]|^2 + \rho^2 & \text{if } k = m. \end{cases} \tag{32}$$

Proof: Since

$$E\left[\mathcal{K}_\nu^k[s(t)+\nu(t)]\, \overline{\mathcal{K}_\nu^m[s(t)+\nu(t)]}\right] =$$

$$\mathcal{K}_\nu^k[s(t)]\, \overline{\mathcal{K}_\nu^m[s(t)]} + \overline{\mathcal{K}_\nu^m[s(t)]} E\left[\mathcal{K}_\nu^k[\nu(t)]\right] +$$

$$\mathcal{K}_\nu^k[s(t)] E\left[\overline{\mathcal{K}_\nu^m[\nu(t)]}\right] + E\left[\mathcal{K}_\nu^k[\nu(t)]\, \overline{\mathcal{K}_\nu^m[\nu(t)]}\right],$$

the claim follows directly from Theorem IV.1 and Theorem IV.2. ■

Let $\mathbf{Cov}_K^f(t)$ be the covariance matrix of the $K+1$ chromatic derivatives of orders $0,\dots,K$ of the noisy signal at an instant t:

$$\mathbf{Cov}_K^f(t)=\left(E\left[\mathcal{K}_\nu^k[f](t)\overline{\mathcal{K}_\nu^m[f](t)}\right]:0\le k,m\le K\right),$$

and let $\mathbf{I}_{K+1}$ be the identity matrix of size $(K+1)\times(K+1)$; then, by Theorem IV.3,

$$\mathbf{Cov}_K^f(t)=\left(\mathcal{K}_\nu^k[s](t)\overline{\mathcal{K}_\nu^m[s](t)}\;:\;0\le k,m\le K\right)+\rho^2\,\mathbf{I}_{K+1}. \tag{33}$$

Since the noise $\nu(t)$ is assumed to be WSS, by taking averages of realizations at instants $1,\dots,N$ we obtain that

$$1/N(\boldsymbol{\mathcal{M}}_m^f)^*\boldsymbol{\mathcal{M}}_m^f\approx 1/N(\boldsymbol{\mathcal{M}}_m^s)^*\boldsymbol{\mathcal{M}}_m^s+\rho^2\,\mathbf{I}_{K+1} \tag{34}$$

Let $\boldsymbol{\mathcal{A}}_m^f=1/N(\boldsymbol{\mathcal{M}}_m^f)^*\boldsymbol{\mathcal{M}}_m^f$ and $\boldsymbol{\mathcal{A}}_m^s=1/N(\boldsymbol{\mathcal{M}}_m^s)^*\boldsymbol{\mathcal{M}}_m^s$; let also $\boldsymbol{\mathcal{A}}_m^s=\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^*$ be the eigen-decomposition of $\boldsymbol{\mathcal{A}}_m^s$; since for identity matrices equality $\mathbf{I}_{K+1}=\mathbf{Q}\mathbf{I}_{K+1}\mathbf{Q}^*$ holds for every unitary matrix $\mathbf{Q}$, we get from (34) that

$$\boldsymbol{\mathcal{A}}_m^f\approx\mathbf{Q}\left(\boldsymbol{\Sigma}+\rho^2\mathbf{I}_{K+1}\right)\mathbf{Q}^*. \tag{35}$$

Thus, if operators $\mathcal{K}_\nu^m$ correspond to polynomials $p_m^\nu(\omega)$ orthonormal with respect to the PSD $S(\omega)$ of the noise $\nu(t)$ and if N is reasonably large, then the eigenvectors of matrix $\boldsymbol{\mathcal{A}}_m^f$ are good approximations of the eigenvectors of matrix $\boldsymbol{\mathcal{A}}_m^s$ and the eigenvalues of $\boldsymbol{\mathcal{A}}_m^f$ differ from the eigenvalues of matrix $\boldsymbol{\mathcal{A}}_m^s$ approximately by the mean square ρ^2 of the noise $\nu(t)$. Consequently, informally speaking, if operators $\mathcal{K}_\nu^m$ are "matched" to the PSD of the noise, then the noise will minimally perturb the eigenvectors of the auto-covariance matrix of the chromatic derivatives of the noise-free signal. Since $\boldsymbol{\mathcal{A}}_m^s$ and $\boldsymbol{\mathcal{A}}_m^f$ are both of size $(K+1)\times(K+1)$ but the rank of $\boldsymbol{\mathcal{A}}_m^s$ is only n, the $K-n$ smallest eigenvalues of $\boldsymbol{\mathcal{A}}_m^s$ are zero and the eigenvalue λ_{n+1}^f of $\boldsymbol{\mathcal{A}}_m^f$ is a good approximation of ρ^2, the mean square of the noise. Since the eigenvectors of $\boldsymbol{\mathcal{A}}_m^f$ are also the right singular values of matrix $\boldsymbol{\mathcal{M}}_m^f$ and the square roots of the eigenvalues of $\boldsymbol{\mathcal{A}}_m^f$ are the singular values of $\boldsymbol{\mathcal{M}}_m^f$, we obtain the following (somewhat informal) corollary.

Corollary IV.4. *Let operators $\mathcal{K}_\nu^m$ correspond to polynomials $p_m^\nu(\omega)$ orthonormal with respect to the PSD of the noise and let $\boldsymbol{\mathcal{M}}_m^f=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$ be the SVD of $\boldsymbol{\mathcal{M}}_m^f$. Let also σ_k for $k=1\dots N$ be the singular values of $\boldsymbol{\mathcal{M}}_m^f$, forming the diagonal of $\boldsymbol{\Sigma}$. Define a diagonal matrix $\widehat{\boldsymbol{\Sigma}}$ such that $\widehat{\boldsymbol{\Sigma}}(p,p)=\left(\sigma_p^2-\sigma_{n+1}^2\right)^{1/2}$ for all $1\le p\le n$ and $\widehat{\boldsymbol{\Sigma}}(p,p)=0$ for $p>n$. Then matrix $\widehat{\boldsymbol{\mathcal{M}}}_m^f=\mathbf{U}\widehat{\boldsymbol{\Sigma}}\mathbf{V}^*$ is a good approximation of the noise-free matrix $\boldsymbol{\mathcal{M}}_m^s$.*

We briefly provide a further insight into the significance of using operators $\mathcal{K}_\nu^m$ matched to the PSD of the noise. Let $\mathcal{D}_{\mathbf{X}}[f]$ be an arbitrary linear differential operator with constant coefficients, of the form $\mathcal{D}_{\mathbf{X}}[f]=\sum_{m=1}^K X_m\,\mathcal{K}_\nu^m[f]$, where X_m are complex coefficients satisfying $\sum_{m=1}^K|X_m|^2=1$ and operators $\mathcal{K}_\nu^m$ are matched to the PSD of the noise $\nu(t)$. Then, applying such an operator to the noisy signal $f(t)=s(t)+\nu(t)$, it is easy to verify that Theorem IV.3 implies that

$$E\left[|\mathcal{D}_{\mathbf{X}}[f](t)|^2\right]-|\mathcal{D}_{\mathbf{X}}[s](t)|^2=\rho^2\sum_{m=1}^K|X_m|^2=\rho^2.$$

Thus, remarkably, if the chromatic derivatives $\mathcal{K}_\nu^m$ are chosen to correspond to the PSD of the noise, then the impact of noise on the value of $|\mathcal{D}_{\mathbf{X}}[f](t)|$ for any operator $\mathcal{D}_{\mathbf{X}}=\sum_{i=1}^K X_i\,\mathcal{K}_\nu^i$ with X_i satisfying $\sum_{i=1}^K|X_i|^2=1$ *does not depend* on the values of the parameters X_i, and is always equal to the mean square value of the noise. Consequently, to find an operator $\mathcal{D}_{\mathbf{X}}$ of the above form which annihilates the signal $s(t)$ and thus produces a differential equation of the form (21), it is enough to find $\mathbf{X}^0=(X_1^0,\dots,X_K^0)$ which minimizes $E\left[|\mathcal{D}_{\mathbf{X}}[f](t)|^2\right]$; such a minimum is approximately equal to the mean square of the noise. Since the noise is assumed to be WSS, this problem is well approximated by minimizing $S(\mathbf{X})=1/N\sum_{i=1}^N|\mathcal{D}_{\mathbf{X}}[f](i)|^2$.

To find the minimum of $S(\mathbf{X})$ subject to the constraint $\sum_{i=0}^K|X_i|^2=1$, in order to avoid the problem that the modulus, or, alternatively, the complex conjugation, are not differentiable functions, all quantities and variables have to be represented via their real and imaginary parts, thus reducing a complex optimization problem to a real one which can be solved by Lagrangian multipliers. Remarkably, we obtain that the solutions are given by the eigenvectors which correspond to the $K-n$ smallest eigenvalues of the very same matrix $\boldsymbol{\mathcal{A}}_m^f$; see [12] where this is demonstrated in detail for the case of real sinusoids and $K=n+1$.

V. CT-MUSIC Algorithm

We are now ready to apply the above general considerations to obtain a practical algorithm. In general, there are two possibilities:

Case 1: The values of the chromatic derivatives $\mathcal{K}_\nu^m[f](i)$, $(1\le i\le N)$, are obtained from a front end filterbank, either analog or digital. In such a case the algorithm would be a straight forward implementation of what we have presented in the previous section, and in some circumstances this might be the best approach. However, the performance of such an algorithm would be hard to compare with the performance of the standard MUSIC algorithm. In the case of a front end analog filterbank, one would have to implement such a filterbank, for example, using a cascade of all pass analog filters of the kind presented in [13]. On the other hand, if the front end filterbank is digital, then a proper comparison would have to take into account the number of samples necessary for the support of the digital filters, rather than just the number of samples of the output of such a filterbank which are actually used in the frequency estimation algorithm. For these reasons we relegate such a case for a future work and present here an alternative approach which can be directly compared with the standard MUSIC algorithm.

Case 2: In this article, the input for our algorithm will be just N samples $f(1),\dots,f(N)$ of the noisy signal $f(t)=s(t)+\nu(t)$, which are also used as input of the standard MUSIC algorithm. We are especially interested in the performance of our algorithm when the number of samples N is relatively small, between 16 and 128 (or so) samples, which are the cases most relevant for a very important application of frequency estimation algorithms, namely the Direction of Arrival (DOA) estimation [7].[6] As we have mentioned, we will first assume that the PSD of the noise is known. We will later explain how an estimate of such a PSD is obtained (or it is assumed that the noise is white). It is important to note that, unlike in Case 1, with such a relatively small number of samples, accurate approximations of the values of the chromatic derivatives of the noisy signal are possible only for chromatic derivatives of low orders and only in the central part of the time interval spanned by samples $f(1),\ldots,f(N)$. Thus, in this case our algorithm is not just a direct implementation of the general theory, as it is in Case 1.

In the remaining part of this paper, we let $\mathbf{f} = (f(1),\ldots,f(N))^\top$ where $f(t)$ is a sum of a signal $s(t)$ of the form (1) where parameters A_k, ω_k and ϕ_k are unknown, but the number n of the complex exponentials is known (which is also a requirement for the standard MUSIC algorithm) and $\nu(t)$ is a WSS, circularly distributed zero mean π-band-limited noise. As we have mentioned, at the moment we also assume that the PSD $S(\omega)$ of the noise is known. Ideally, our aim should be to find a waveform $\widehat{\alpha}(t)$ which satisfies the following two properties:

1) $\widehat{\alpha}(t)$ is a sum of n complex exponentials;
2) The differences $\widehat{\alpha}(1)-f(1),\ldots,\widehat{\alpha}(N)-f(N)$ have the highest possible likelihood to have come from noise with PSD $S(\omega)$.

To make such a problem computationally tractable, our aim will be to find a waveform $\widehat{\alpha}(t)$ which satisfies both conditions only approximately, but with a sufficient degree of fidelity. We describe four modes of operation of our algorithm providing increasing levels of such fidelity which results in increasing noise robustness:

1) non adaptive mode;
2) adaptive mode;
3) single adaptive-learning mode, and
4) double adaptive-learning mode.

Our CT-MUSIC algorithm consists of the following steps, concisely represented on Figure 2.

Step 1 (initial signal capture): We first obtain an initial waveform $\alpha(t)$ which approximates the noisy signal $f(t)$ and which we will subsequently refine, by solving the following least squares fit: Find $\mathbf{X}=(X_0,\ldots,X_M)^\top$ which minimizes the expression

$$D(\mathbf{X})=\sum_{i=1}^{N}\left|\sum_{m=0}^{M}X_m B_m(i-N/2)-f(i)\right|^2+\mu\sum_{m=0}^{M}|X_m|^2 \tag{36}$$

Here $B_m(t)$ are the (real valued) chromatic expansion functions which correspond to the (normalized and rescaled) Legendre polynomials $p_n^{L}(\omega)=\sqrt{2n+1}P_n^{L}(\omega/\pi)$, where $P_n^{L}(\omega)$ are the standard Legendre polynomials; see Section II(B) in [4]. Thus, $B_m(t)=\sqrt{2n+1}j_n(\pi t)$, where $j_n(x)$ are the spherical Bessel functions of the first kind. As explained in Section II in [5], the choice of the regularization constant μ is not critical at all, with the same results achieved for $10^{-12}\leq\mu\leq 10^{-8}$; in our experiments μ was set to 10^{-10}. The degree M of the chromatic approximation is also not very critical for as long as it is sufficiently large for the given number of samples N a unit distance apart; as explained in [4], we can take $M=14+\lfloor 1.67N\rfloor$. Let $\mathbf{B}_0=(B_m(i-N/2)\ :\ 1\leq i\leq N;\ 0\leq m\leq M)$; by splitting (36) into the real and the imaginary part, it is easy to see that such a complex LSF reduces to a real one with a solution given by $\mathbf{X}^0=\mathbf{M}\cdot\mathbf{f}$ where $\mathbf{f}=(f(1),\ldots,f(N))^\top$ and

$$\mathbf{M}=\left(\mathbf{B}_0^\top\cdot\mathbf{B}_0+\mu\,\mathbf{I}_{M+1}\right)^{-1}\mathbf{B}_0^\top. \tag{37}$$

We now let $\alpha(t)=\sum_{m=0}^{M}X_m^0 B_m(t-N/2)$. As we have shown in [5], if the signal is even modestly oversampled, for example by a factor of $1.1-1.2$, then $\alpha(t)$ is already an excellent approximation of the noisy signal $f(t)=s(t)+\nu(t)$ in terms of the maximal value of the difference $|\alpha(t)-f(t)|$ over the *continuous time interval* $[1,N]$, even if such sampling is non-uniform. However, this *does not* imply that the same is true for the RMS values of differences $|\mathcal{K}_\nu^m[\alpha_0](t)-\mathcal{K}_\nu^m[f](t)|$, especially not for chromatic derivatives of higher orders whose errors can be quite large. In fact, from the "perspective" of derivatives, the waveform $\alpha(t)$ over the continuous time interval spanned by the samples might not quite look like a sum of n complex exponentials with additive noise with an RMS value ρ, because the RMS value of $|\mathcal{K}_\nu^m[\alpha_0](t)-\mathcal{K}_\nu^m[f](t)|$ might be significantly larger than ρ. Thus, in the next step of our algorithm we reshape the waveform $\alpha(t)$, by making it more compliant with being a sum on n complex exponentials from the perspective of derivatives.

Step 2 (model shaping): We now obtain a matrix $\boldsymbol{\mathcal{M}}_m^\alpha=(\mathcal{K}_\nu^k[\alpha](i),\ 0\leq k\leq N-1;\ 1\leq i\leq N)$ consisting of values of $\mathcal{K}_\nu^k[\alpha](i)$ for orders $k=0\ldots N-1$ at N sampling instants $i=1\ldots N$. Using chromatic derivatives of orders up to $N-1$ provides the best super-resolution performance while the algorithm remains numerically robust and stable. Since $\mathcal{K}_\nu^k[\alpha](i)=\sum_{m=0}^{M}X_m^0\,\mathcal{K}_\nu^k[B_m](i-N/2)$, obtaining the matrix $\boldsymbol{\mathcal{M}}_m^\alpha$ reduces to finding N many products of the form $(\mathcal{K}_\nu^k[\alpha](i),\ 1\leq i\leq N)^\top=\mathbf{B}_k^\nu\,\mathbf{M}\,\mathbf{f}$ for $k=0\ldots N-1$, where

$$\mathbf{B}_k^\nu=\left(\mathcal{K}_\nu^k[B_m](i-N/2)\ :\ 1\leq i\leq N;\ 0\leq m\leq M\right) \tag{38}$$

and $\mathbf{M}$ is given by (37). In the non-adaptive mode of operation of the CT-MUSIC algorithm we assume that the noise is white. Thus, matrices $\mathbf{B}_k^\nu$ are obtained by recursion, using Theorem 6.2 of Section VI(B) in [4], with both γ_n^p and $\gamma_n^{p^*}$ corresponding to the Legendre polynomials, i.e., with

$$\gamma_n^p=\gamma_n^{p^*}=\gamma_n^{L}=\pi(n+1)/\sqrt{4(n+1)^2-1}. \tag{39}$$

We will explain later how such matrices $\mathbf{B}_k^\nu$ are obtained for the adaptive and adaptive-learning modes of the CT-MUSIC algorithm.

[6]We explain how to deal with a larger number of samples in Section VI.

For increased efficiency of computations, we concatenate the columns of all matrices $\mathbf{B}_k^\nu$ thus obtaining a single matrix $\widetilde{\mathbf{B}}^\nu$; we then obtain matrix $\mathbf{G}_\nu = \widetilde{\mathbf{B}}^\nu \cdot \mathbf{M}$ thus reducing the computation of $\boldsymbol{\mathcal{M}}_m^\alpha$ to a product of a single matrix $\mathbf{G}_\nu$ with vector $\mathbf{f} = (f(1), \ldots, f(N))^\top$ and subsequent reshaping of the thus obtained product vector $\mathbf{G}_\nu \cdot \mathbf{f}$ into the matrix $\boldsymbol{\mathcal{M}}_m^\alpha$. Note that matrix $\mathbf{G}_\nu$ does not depend on the input but only on the input size and the PSD of the noise ν; thus, it is precomputed and stored during the noise adaptation background process, as we will explain later.

In order to reshape the initial approximation $\alpha(t)$ into an approximation which is more compliant with being a sum of n complex exponentials we apply Corollary IV.4. Thus, we obtain the SVD of $\boldsymbol{\mathcal{M}}_m^\alpha$, $\boldsymbol{\mathcal{M}}_m^\alpha = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$. Let $\widehat{\boldsymbol{\Sigma}}$ be obtained by replacing the singular values σ_p on the diagonal of $\boldsymbol{\Sigma}$ with $\widehat{\sigma}_p = \left(\sigma_p^2 - \sigma_{n+1}^2\right)^{1/2}$ for $1 \leq p \leq n$ and by setting $\widehat{\sigma}_p = 0$ for $p > n$; we now set $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha = \mathbf{U}\widehat{\boldsymbol{\Sigma}}\mathbf{V}^*$. Clearly, such a matrix has a rank equal to n, but it is no longer a matrix of chromatic derivatives of a band-limited signal. Thus, we now look for a waveform $\widehat{\alpha}(t)$ such that the matrix of its chromatic derivatives $\left(\mathcal{K}_\nu^k[\widehat{\alpha}](i) \ : \ 0 \leq k \leq N-1; \ 1 \leq i \leq N\right)$ is the closest in the sense of the Frobenius norm to matrix $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha$. We now use a LSF to look for $\mathbf{Y} = (Y_0, \ldots, Y_n)^\top$ which minimizes

$$\sum_{i=1}^{N}\sum_{k=0}^{N-1}\left|\sum_{m=0}^{M} Y_m\, \mathcal{K}_\nu^k[B_m](i) - \widehat{\boldsymbol{\mathcal{M}}}_m^\alpha(i,k)\right|^2 + \mu\sum_{m=0}^{M}|Y_m|^2.$$

Let $\mathbf{Z} = \sum_{k=0}^{N-1}(\mathbf{B}_k^\nu)^\top \mathbf{B}_k^\nu$; then, again splitting the real and the imaginary parts, a standard straightforward computation shows that this LSF has a solution of the form

$$\mathbf{Y}^0 = (\mathbf{Z} + \mu\,\mathbf{I}_{M+1})^{-1}\sum_{k=0}^{N-1}\mathbf{B}_k^\nu \cdot \widehat{\boldsymbol{\mathcal{M}}}_m^\alpha(:, k+1) \tag{40}$$

where $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha(:, k+1)$ is a vector which is column $k+1$ of $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha$. Having found $\mathbf{Y}^0$ we obtain our reshaped signal approximation $\widehat{\alpha}(t) = \sum_{m=0}^{M} Y_m^0 B_m(t)$ which, in light of Corollary IV.4, has the property that it approximates a sum of n complex exponentials from the perspective of higher order derivatives.

Approximation $\widehat{\alpha}(t)$ still contains some residual noise $\widehat{\nu}(t) = \widehat{\alpha}(t) - s(t)$, but the PSD of $\widehat{\nu}$ is different from the PSD of ν. Thus, we now have two options.

Option 1: In the non-adaptive, adaptive and single adaptive-learning modes of operation of our CT-MUSIC algorithm we treat such residual noise $\widehat{\nu}(t)$ as if it were white noise and we define a matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ as

$$\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}} = \left(\mathcal{K}_{L}^k[\widehat{\alpha}](i) \ : \ 0 \leq k \leq N-1; \ 1 \leq i \leq N\right),$$

where $\mathcal{K}_L^k$ are the chromatic derivatives associated with the Legendre polynomials.

Option 2: In the double adaptive-learning mode of our CT-MUSIC algorithm, which is the most effective for estimation and tracking of frequencies in a very noisy and changing environment, we dynamically, during the operation of our algorithm, estimate the PSD $\widehat{S}(\omega)$ of the residual noise $\widehat{\nu}$, as we will explain later. In this option matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ is defined by

$$\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}} = \left(\mathcal{K}_{\widehat{\nu}}^k[\widehat{\alpha}](i) \ : \ 0 \leq k \leq N-1; \ 1 \leq i \leq N\right), \tag{41}$$

where $\mathcal{K}_{\widehat{\nu}}^k$ are the chromatic derivatives matched to such estimated PSD $\widehat{S}(\omega)$ of the residual noise $\widehat{\nu}$ (which is, as we will describe later, obtained by a background noise adaptation process).

Let

$$\mathbf{B}_k^{\widehat{\nu}} = \left(\mathcal{K}_{\widehat{\nu}}^k[B_m](i) \ : \ 1 \leq i \leq N; \ 0 \leq m \leq M\right); \tag{42}$$

then row $k+1$, $0 \leq k \leq N-1$, of matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ given by (41) can be obtained using (40) as

$$\begin{aligned}\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}(k+1, :)^\top &= \mathbf{B}_k^{\widehat{\nu}} \cdot \mathbf{Y}^0 \\ &= \mathbf{B}_k^{\widehat{\nu}} \cdot (\mathbf{Z} + \mu\,\mathbf{I}_{M+1})^{-1} \cdot \sum_{k=0}^{N-1}\mathbf{B}_k^\nu \cdot \widehat{\boldsymbol{\mathcal{M}}}_m^\alpha(:, k+1).\end{aligned}$$

To improve the computational efficiency, the above step is performed in the following manner. All columns of matrix $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha$ are concatenated to obtain a single vector $\mathbf{h}$; the rows of matrices $\mathbf{B}_k^\nu$ for all $0 \leq k \leq N-1$ are also concatenated to obtain a single matrix $\mathbf{H}$; in this way $\sum_{k=0}^{N-1}\mathbf{B}_k^\nu \cdot \widehat{\boldsymbol{\mathcal{M}}}_m^\alpha(:, k+1) = \mathbf{H} \cdot \mathbf{h}$. Let us set $\widehat{\mathbf{h}} = \mathbf{H} \cdot \mathbf{h}$. Similarly, the columns of all matrices $\mathbf{B}_k^{\widehat{\nu}}$ are concatenated to obtain a single matrix $\mathbf{J}$; we then obtain a matrix $\mathbf{K} = \mathbf{J} \cdot (\mathbf{Z} + \mu\,\mathbf{I}_{M+1})^{-1}$. Note that matrices $\mathbf{H}$ and $\mathbf{K}$ do not depend on the input; they are pre-calculated and stored during the noise adaptation process. The product $\mathbf{K} \cdot \widehat{\mathbf{h}}$ is a vector which consists of concatenated columns of matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$;[7] thus, matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ is obtained by reshaping such a vector. Consequently, matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ is obtained from matrix $\widehat{\boldsymbol{\mathcal{M}}}_m^\alpha$ using only two matrix-vector multiplications and some matrix reshaping. Matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ is now used in the frequency estimation step, in lieu of matrix $\boldsymbol{\mathcal{M}}_m^f$ given by (27).

Step 3 (frequency estimation): We now obtain the SVD $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}} = \mathbf{PDR}^*$ of matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ and take $N-n$ left singular vectors $\mathbf{v}_k$ which are the last $N-n$ columns of $\mathbf{R}$ which correspond to $N-n$ smallest singular values in $\mathbf{D}$ as approximations of the $N-n$ left singular vectors of the noise-free matrix $\boldsymbol{\mathcal{M}}_m^s$. We form the polynomial $Q(\omega)$ given by (29) but this time with polynomials $p_n^{\widehat{\nu}}(\omega)$ which are orthonormal with respect to the PSD of the residual noise $\widehat{\nu}$; these polynomials are obtained by the background adaptation process as we will explain later. Thus, we consider

$$\mathcal{F}_{\widehat{\nu}}(\omega) = \sum_{k=n+1}^{N}\sum_{m=0}^{N-1}\sum_{r=0}^{N-1} j^{m+3k}\mathbf{v}_k(m+1)\overline{\mathbf{v}_k(r+1)}p_m^{\widehat{\nu}}(\omega)p_r^{\widehat{\nu}}(\omega) \tag{43}$$

and look for its local minima by finding the real roots of the equation $\mathcal{F}_{\widehat{\nu}}'(\omega) = 0$ and selecting those roots where $\mathcal{F}_{\widehat{\nu}}''(\omega)$ is positive. We then evaluate $\mathcal{F}_{\widehat{\nu}}(\omega)$ at these roots and select

[7] Note that we do not replace matrices $\mathbf{K}$ and $\mathbf{H}$ with their product $\mathbf{K} \cdot \mathbf{H}$ because such a product matrix would be of a very large size; it is computationally much more efficient to first compute the product $\mathbf{H} \cdot \mathbf{h}$ and then multiply matrix $\mathbf{K}$ by the resulting vector.

n roots where $\mathcal{F}_{\widehat{\nu}}(\omega)$ attains its smallest n values. These roots are taken as estimates of the frequencies sought.

As we have already mentioned, to achieve optimal super-resolution while retaining numerical robustness, matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}} = \left(\mathcal{K}_{\widehat{\nu}}^k[\widehat{\alpha}](i)\ :\ 0 \leq k \leq N-1;\ 1 \leq i \leq N\right)$ contains chromatic derivatives of orders up to $N-1$. Thus, polynomial $\mathcal{F}_{\widehat{\nu}}(\omega)$ is of degree $2N-2$. If the input consists of 128 samples of the noisy signal, this results in a polynomial $\mathcal{F}_{\widehat{\nu}}(\omega)$ of degree 254 with some of its coefficients of size of the order of 10^{94}; for this reason we can neither represent polynomial $\mathcal{F}_{\widehat{\nu}}(\omega)$ in the coefficient form nor can we use the *Matlab* '`roots`' function to find roots of the derivative of such a polynomial. Instead, the adaptation procedure produces polynomials $p_m^{\widehat{\nu}}(\omega)$ represented as linear combinations of the normalized and rescaled Legendre polynomials $p_n^L(\omega) = \sqrt{2n+1}P_n^L(\omega/\pi)$ which can be efficiently evaluated with high accuracy using the recursion formula

$$p_{n+1}^L(\omega) = \frac{\omega}{\gamma_n^L}\, p_n^L(\omega) - \frac{\gamma_{n-1}^L}{\gamma_n^L} p_{n-1}^L(\omega) \tag{44}$$

where γ_n^L are given by (39). When polynomials $p_n^{\widehat{\nu}}(\omega)$ are expressed in such a base as a linear combination $p_n^{\widehat{\nu}}(\omega) = \sum_{k=0}^{n} \beta_k^n p_k^L(\omega)$ the absolute values of the coefficients β_k^n are generally smaller than 10. Thus, the products of the form $p_m^{\widehat{\nu}}(\omega)\, p_r^{\widehat{\nu}}(\omega)$ in (43) can be represented as

$$p_m^{\widehat{\nu}}(\omega)\, p_r^{\widehat{\nu}}(\omega) = \sum_{k=0}^{m}\sum_{l=0}^{r} \beta_k^m \beta_l^r\, p_k^L(\omega)\, p_l^L(\omega).$$

Using Theorem 7.1 of Section VII(A) in [4], products of the form $p_k^L(\omega)\, p_l^L(\omega)$ can be linearized and represented in the form $p_k^{\widehat{\nu}}(\omega)\, p_l^{\widehat{\nu}}(\omega) = \sum_{s=|k-l|}^{k+l} \widehat{\beta}_s p_s^L(\omega)$; consequently, polynomial $\mathcal{F}_{\widehat{\nu}}(\omega)$ can be represented in the form

$$\mathcal{F}_{\widehat{\nu}}(\omega) = \sum_{i=0}^{2N-2} \eta_i p_i^L(\omega) \tag{45}$$

where η_i are real coefficients. This implies that the derivative of $\mathcal{F}_{\widehat{\nu}}(\omega)$ is of the form

$$(\mathcal{F}_{\widehat{\nu}}(\omega))' = \sum_{i=0}^{2N-2} \eta_i (p_i^L(\omega))'.$$

We now use the following fact which follows from the properties of the Legendre polynomials $P_n^L(x)$, see [14], Section 12.2:

$$(p_n^L(\omega))' = \frac{n\pi}{\omega^2-\pi^2}\left(\frac{(n+1)\, p_{n+1}^L(\omega)}{\sqrt{4(n+1)^2-1}} - \frac{(3n+1)\, p_{n-1}^L(\omega)}{\sqrt{4n^2-1}}\right).$$

This allows us to represent $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ in the following form

$$(\mathcal{F}_{\widehat{\nu}}(\omega))' = \frac{\pi}{\omega^2-\pi^2} \sum_{i=0}^{2N-1} \widehat{\eta}_i\, p_i^L(\omega). \tag{46}$$

In this way both $\mathcal{F}_{\widehat{\nu}}(\omega)$ and $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ can be evaluated with high precision by evaluating polynomials $p_i^L(\omega)$ using recursion (44). To find real zeros of $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ we now evaluate $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ on a grid of 4000 equally spaced points within $[-\pi, \pi]$ and look for the consecutive points where $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ changes sign from negative to positive; we then apply the Newton - Raphson method to find zeros of $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ lying between these consecutive points. In this way we obtain the minima of $\mathcal{F}_{\widehat{\nu}}(\omega)$. Finally, we evaluate $\mathcal{F}_{\widehat{\nu}}(\omega)$ at these local minima and pick n smallest such local minima which are taken as the estimates of the frequencies of the components. Figure V shows a typical plot of $\mathcal{F}_{\widehat{\nu}}(\omega)$ for a signal $s(t)$ consisting of three complex exponentials of frequencies represented by the three gridlines and with 10dB white noise.

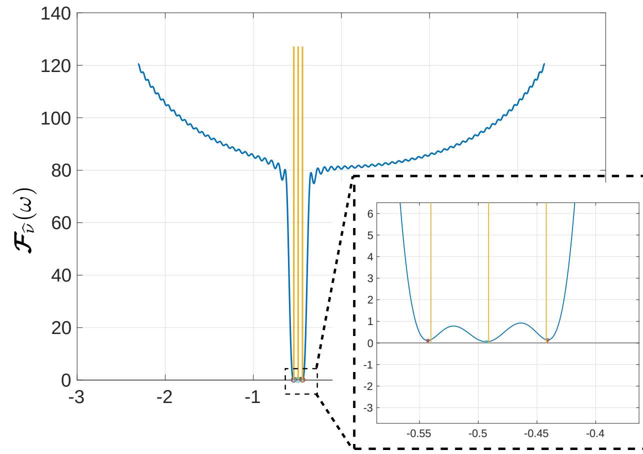

Fig. 1. The smallest local minima of the function $\mathcal{F}_{\widehat{\nu}}(\omega)$ resulting in the frequency estimates.

The background noise adaptation process: In order to enable the noise adaptation background process for adaptive-learning modes of the CT-MUSIC algorithm, we have to provide estimates of the PSD of the overall noise ν and the PSD of the residual noise $\widehat{\nu}$, left after Step 2 (model shaping). Let δ_p be the singular values of matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ as obtained at the beginning of Step 2, forming the diagonal of matrix $\mathbf{D}$. Since the chromatic derivatives $\mathcal{K}_{\widehat{\nu}}^k$ are chosen matched to the PSD of the residual noise $\widehat{\nu}$, relying again on Corollary IV.4, if we replace δ_p in $\mathbf{D}$ with $\widehat{\delta}_p = (\delta_p - \delta_{n+1})^{1/2}$ for $1 \leq p \leq n$ and if we set $\widehat{\delta}_p = 0$ for $p > n$ thus obtaining a diagonal matrix $\widehat{\mathbf{D}}$ with such $\widehat{\delta}_p$ on the diagonal, then vector $\mathbf{s} = \mathbf{P}\,\widehat{\mathbf{D}}\,\mathbf{R}(1,:)^*$ is a good approximation of the vector of samples $(s(1), \ldots s(N))^\top$ of the noise-free signal $s(t)$. Consequently, we can take $\boldsymbol{\nu}$ given by $\boldsymbol{\nu}(i) = \mathbf{f}(i) - \mathbf{s}(i)$ as an approximate realization of the overall noise ν and vector $\widehat{\boldsymbol{\nu}}$ given by $\widehat{\boldsymbol{\nu}}(i) = \boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}(i,1) - \mathbf{s}(i)$ as an approximation of a realization of the residual noise $\widehat{\nu}$, left after Step 2.

Vectors $\boldsymbol{\nu}$ resulting from several consecutive runs are concatenated; the same is done with vectors $\widehat{\boldsymbol{\nu}}$. In our experiments this is done over 16 consecutive runs, because the adaptation process in our implementation took slightly less time than 16 estimation runs; thus, if adaptation is done on a separate processor, the PSD estimates of the noise ν and the residual noise $\widehat{\nu}$ can be updated in cycles each consisting of 16 runs of estimation/tracking of frequencies of the components. Such concatenated noise realizations of ν and $\widehat{\nu}$ are split into several segments of equal size; in our experiments these segments (which can even be partially overlapping, to increase the number of segments) consisted of 128 sampling instants; we then find the absolute values of the discrete Fourier transforms (DFT) of such segments which are then averaged to obtain a rough PSD of the noise ν and the residual noise $\widehat{\nu}$ during a cycle t consisting of 16 runs of frequency estimation and tracking. Such averaged estimations of the PSD of noise ν and the residual noise $\widehat{\nu}$ are flipped and averaged with the original estimations, to obtain the PSD estimations $\boldsymbol{\pi}_t$ and $\widehat{\boldsymbol{\pi}}_t$ of the noise and the residual noise which are symmetric with respect to the positive and negative frequencies. Estimates $\boldsymbol{\pi}_t$ and $\widehat{\boldsymbol{\pi}}_t$ are then used to update the existing overall PSD estimates $\boldsymbol{\rho}_t$ and $\widehat{\boldsymbol{\rho}}_t$ of the noise ν and the residual noise $\widehat{\nu}$, respectively, via a smoothing process given by $\boldsymbol{\rho}_{t+1} = (1-c)\boldsymbol{\rho}_t + c\,\boldsymbol{\pi}_t$ and $\widehat{\boldsymbol{\rho}}_{t+1} = (1-c)\widehat{\boldsymbol{\rho}}_t + c\,\widehat{\boldsymbol{\pi}}_t$, respectively. The initial estimates $\boldsymbol{\rho}_0$ and $\widehat{\boldsymbol{\rho}}_0$ are taken to be the PSD of the white noise, i.e., constant vectors $(1,\dots,1)^\top$. The choice of the update constant c, $0 < c < 1$, determines the tradeoff between how quickly the estimation process can adapt to changes in the PSD of the noise in the environment (larger values of c) versus how accurately the PSD noise is estimated in such a noise learning procedure (smaller values of c). In the experiments we report in this paper we used $c = 0.3$.

Continuing with our description of the noise adaptation procedure, we now use the procedure described in Section VII(B) of [5] to obtain the recursion coefficients γ_k^ν for the polynomials $p_k^\nu(\omega)$ which are orthonormal with respect to the weight obtained by normalization of the new PSD estimates $\boldsymbol{\rho}_{t+1}$ of noise ν, as well as the recursion coefficients $\gamma_k^{\widehat{\nu}}$ and the polynomials $p_k^{\widehat{\nu}}(\omega)$ which are orthonormal with respect to the weight obtained by normalizing the PSD $\widehat{\boldsymbol{\rho}}_{t+1}$ of the residual noise $\widehat{\nu}$. As we have mentioned, such a procedure represents polynomials $p_k^{\widehat{\nu}}(\omega)$ as a linear combinations of the (normalized and rescaled) Legendre polynomials $p_n^L(\omega)$.

Recursion coefficients γ_k^ν and $\gamma_k^{\widehat{\nu}}$ are then used to obtain matrices $\mathbf{B}_k^\nu$ and $\mathbf{B}_k^{\widehat{\nu}}$ using the recurrence given by equation (37) of Theorem 6.2 in Section VI(B) in [4], with γ_n^p in [4] equal to our γ_n^L given by (39) and $\gamma_n^{p^*}$ equal to our γ_k^ν and $\gamma_k^{\widehat{\nu}}$, respectively.

Figure 2 gives a concise graphical representation of the signal capture/model shaping, the estimation as well as the adaptation procedures. As we have mentioned, there are four modes of operation of our algorithm.

In the non-adaptive mode, we assume that both the overall noise ν and the residual noise $\widehat{\nu}$ are (approximately) white; thus, in this mode the recursion coefficients γ_n^ν and $\gamma_n^{\widehat{\nu}}$ are both set to (precomputed) recursion coefficients given by (39) which correspond to the (normalized and rescaled) Legendre polynomials, i.e., $\gamma_n^\nu = \gamma_n^{\widehat{\nu}} = \gamma_n^L$ and $p_n^{\widehat{\nu}}(\omega) = p_n^L(\omega)$. In this mode switches SW1 and SW3 on Fig. 2 are both in position 1; since no PSD estimates of noise are computed, switch SW2 is inconsequential.

In the adaptive mode we assume that we have a PSD estimate of the noise present in the environment, obtained by "listening" to the environment noise prior to frequency estimation, i.e., by obtaining a sufficiently long sequence of samples of the environment noise, without the signal $s(t)$ present. We then slice such sequence into sections with 128 samples each, take the absolute value of the DFT of these sections and average them to obtain an estimate $\boldsymbol{\rho}$ of the PSD of the environment noise $\nu(t)$. Such an estimate is normalized and then used to produce the recursion coefficients γ_n^ν of polynomials orthonormal with such normalized PSD estimate as the weight function, using the procedure described in detail in Section VII(B) in [4]. In this mode the residual noise $\widehat{\nu}$ is assumed to be white; consequently, $\gamma_n^{\widehat{\nu}} = \gamma_n^L$. Thus, switch SW1 is in position 2 and switches SW2 and SW3 are both in position 1.

CT-MUSIC algorithm has two adaptive learning modes: *the single adaptive learning mode* and *the double adaptive learning mode*. In both of these modes we do not assume any prior knowledge of the PSD of the noise ν but instead an estimate $\boldsymbol{\rho}$ of the PSD of noise $\nu(t)$ is obtained dynamically, during frequency estimation and tracking and is then used to obtain the corresponding recursion coefficients γ_n^ν; thus, in both of these two modes both switch SW1 and switch SW2 are in position 2. In the single adaptive learning mode the PSD of the residual noise is assumed to be white; thus, in this mode switch SW3 is in position 1 and consequently $\gamma_n^{\widehat{\nu}} = \gamma_n^L$ and $p_n^{\widehat{\nu}}(\omega) = p_n^L(\omega)$. In the double adaptive mode an estimate $\widehat{\boldsymbol{\rho}}$ of the PSD of the residual noise is also obtained dynamically during frequency estimation and tracking in the manner explained and is then used to obtain the corresponding recursion coefficients $\gamma_n^{\widehat{\nu}}$ and the corresponding polynomials $p_n^{\widehat{\nu}}(\omega)$ orthonormal with respect to such PSD estimate $\widehat{\rho}$ of the PSD of the residual noise; thus, in double adaptive learning mode switch SW3 is in position 2.

The two graphs in the left bottom corner of Figure 2 show a typical evolution of estimates $\boldsymbol{\rho}$ of the PSD of noise $\nu(t)$ (top graph) and the evolution of the PSD estimate $\widehat{\boldsymbol{\rho}}$ of the residual noise $\widehat{\nu}(t)$ in double adaptive learning mode over 19 update cycles, each cycle with 16 runs of the frequency estimation/tracking. For each run of frequency estimation in each of the update cycles the frequencies of the components were chosen randomly. On the top plot the dashed red line shows the frequency response of the filter used to shape noise $\nu(t)$; the blue lines show the initial estimates of PSD which correspond to the white noise. From the two graphs it is clear that the estimate of the PSD of the overall noise of the environment converges towards the frequency response of the noise shaping filter fast and shows essentially no dependency on the randomly chosen frequencies of the components of the signal. The convergence of the PSD of the residual noise is slow for lower frequencies and appears to be dependent on the frequencies of the components. This is reflected in the performance of the algorithm; the doubly adaptive learning mode outperforms the single adaptive learning mode only when the noise is "extremely non-white", i.e., when its PSD exhibits very large variations across frequencies and when the SNR is below 5 dB.

VI. Performance Evaluation

We have performed extensive numerical simulations to verify the performance of all the modes of operation of our CT-MUSIC algorithm by considering numerically generated signals of size $16 \leq N \leq 256$ samples containing up to

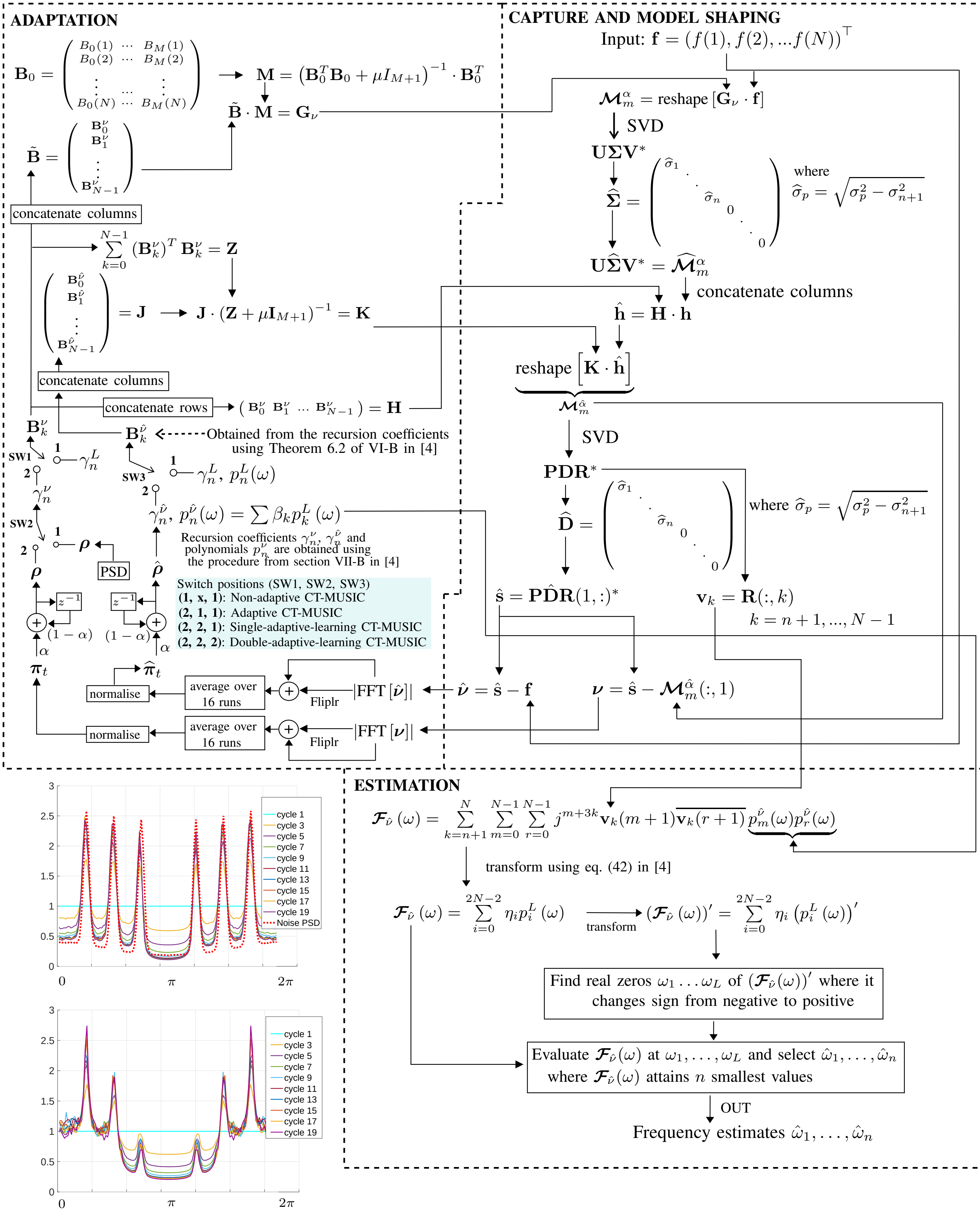

Fig. 2. Graphical representation of the CT-MUSIC algorithm describing its main components: initial signal capture and model shaping, frequency estimation and background noise adaptation process, along with the four modes of operation indicated by the positions of switches (SW1, SW2, SW3). Bottom left shows a typical evolution of the PSD estimates of the noise ν and of the residual noise $\widehat{\nu}$ obtained in the background noise adaptation process for the doubly adaptive-learning mode of operation during 19 cycles of the PSD updates.

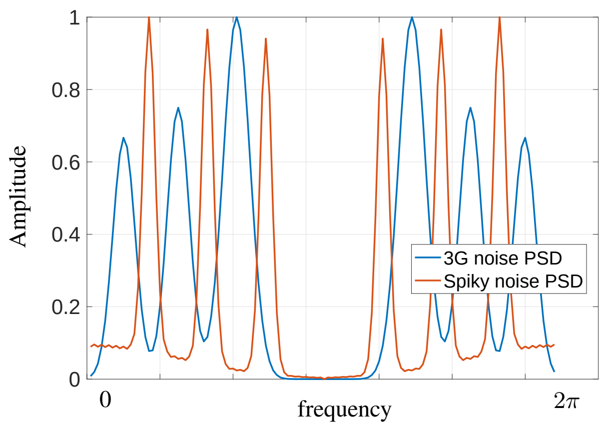

Fig. 3. Power spectral densities of the colored noise profiles "3G" and "spiky" used for simulations.

16 complex exponentials embedded in noise. Super-resolution was tested by choosing the frequencies of the n components to be within k bins of the DFT of the signal for several $k < n$. We have considered white noise as well as two colored noise PSD profiles, denoted "3G" and "spiky", whose PSDs are show in Fig. 3. To ensure that the input signal is slightly oversampled, we limited the frequencies of all components to at most 90% of the full bandwidth, which is consistent with presence of a standard anti-aliasing filter. The performance is expressed in terms of percentage of successful estimation of frequencies of k out of n frequencies, $(0 \le k \le n)$. The frequency of a component was deemed to have been successfully estimated if the error of estimation was less than half of a DFT bin size.[8]

Figs. 4-6 show the estimation performance of non-adaptive and adaptive modes of the CT-MUSIC algorithm in comparison with the *Matlab* implementation of the standard root-MUSIC algorithm, for three types of PSD of the noise: the white noise as well as two types of colored noise, denoted as"3G" and "spiky" with PSD shown on Fig. 3. To ensure a maximally fair comparison, we profiled the Matlab root-MUSIC implementation to determine the size of the auto-correlation matrix to be used in our tests that provides the best super resolution performance; such optimum degree for root-MUSIC was found to be approximately equal to $\lfloor N/2 \rfloor$ for sample sizes less than about 128 and slightly higher for larger sample sizes. Figs. 4-6 show the results of our tests with N=128 samples, with SNR $\in \{-5, 0, 5, 10\}$ dB for a number of demonstrative cases where n components were contained within p DFT bins for $n \in \{2, 3, 5, 7\}$ and for $p \le n-1$ thus testing super-resolution. These figures demonstrate decisively superior performance of CT-MUSIC in comparison to root-MUSIC in terms of both super-resolution and noise robustness. As expected, CT-MUSIC performs particularly well when there is high level of non-white noise; see Fig. 5-6. At high SNR ($\ge$ 10dB), noise adaptation becomes unnecessary. For this reason, after obtaining the SVD of matrix $\boldsymbol{\mathcal{M}}_m^{\alpha}$ we can roughly estimate the SNR as the ratio of the sum of squares of the first n singular values corresponding to the signal and sum of squares of the remaining singular values corresponding to noise, and while such an estimate is above a threshold, the algorithm operates in the non-adaptive mode.

[8]Our algorithm has a similar advantage over MUSIC for both more and less stringent error thresholds.

TABLE I
PERFORMANCE OF CT-MUSIC AND THE ROOT-MUSIC ALGORITHM FOR SAMPLE SIZES $N \in \{16, 32, 64, 256\}$.

	No. of components	Percentage of successful detection (%)		
		CT−MUSIC−NA	CT−MUSIC−A	Root MUSIC
3 in 2 bins	0 out of 3	0	0	0.32
$N = 16$	1 out of 3	0.52	0.52	4.05
white noise	2 out of 3	42.21	42.18	87.10
5 dB SNR	All 3	57.27	57.30	8.53
3 in 2 bins	0 out of 3	0	0	0.33
$N = 16$	1 out of 3	0.90	0.76	5.05
spiky noise	2 out of 3	57.69	49.93	89.29
5 dB SNR	All 3	41.41	49.31	5.33
3 in 2 bins	0 out of 3	0	0	0
$N = 32$	1 out of 3	0.03	0.03	0.97
white noise	2 out of 3	11.44	11.48	73.07
5 dB SNR	All 3	88.53	88.49	25.96
3 in 2 bins	0 out of 3	0	0	0
$N = 32$	1 out of 3	0.29	0.31	4.24
spiky noise	2 out of 3	47.80	37.34	88.46
5 dB SNR	All 3	51.91	62.35	7.30
4 in 3 bins	0−1 out of 4	0	0	1.10
$N = 64$	2 out of 4	1.25	1.25	14.3
white noise	3 out of 4	26.39	26.36	78.51
5 dB SNR	All 4	72.36	72.39	6.09
4 in 3 bins	0−1 out of 4	0	0	2.07
$N = 64$	2 out of 4	3.97	3.43	29.93
spiky noise	3 out of 4	74.12	65.68	67.72
5 dB SNR	All 4	21.91	30.88	0.28
5 in 4 bins	0−2 out of 5	0.15	0.15	0.40
$N = 256$	3 out of 5	2.85	2.87	13.10
white noise	4 out of 5	36.29	36.29	86.50
5 dB SNR	All 5	60.71	60.69	0
5 in 4 bins	0−2 out of 5	0.15	0.17	3.15
$N = 256$	3 out of 5	3.60	3.62	47.37
spiky noise	4 out of 5	69.71	59.67	49.48
5 dB SNR	All 5	26.54	36.54	0

TABLE II
COMPLEXITY OF THE CT-MUSIC AND CONVENTIONAL ROOT-MUSIC ALGORITHMS EXPRESSED IN TERMS OF COMPUTATION TIME.

Algorithm	Computation time in milisec. for different sample sizes				
	16	32	64	128	256
CT-MUSIC	0.88409	1.6648	4.7751	20.913	127.22
MUSIC	2.1582	3.1534	5.9839	27.623	159.65

Similar experimental results are shown in Table I for sample sizes $N \in \{16, 32, 64, 256\}$ with white and spiky noise, 5dB SNR for a number of demonstrative cases for super-resolution. All the results in Figs. 4-6 and Table I are obtained by averaging over 10,000 runs with random selection of frequencies with the given separation.

We have also compared the complexity of the proposed CT-MUSIC with standard root-MUSIC (available in Matlab) by considering the computation time measured using Matlab implementations running on a Windows 10 host computer

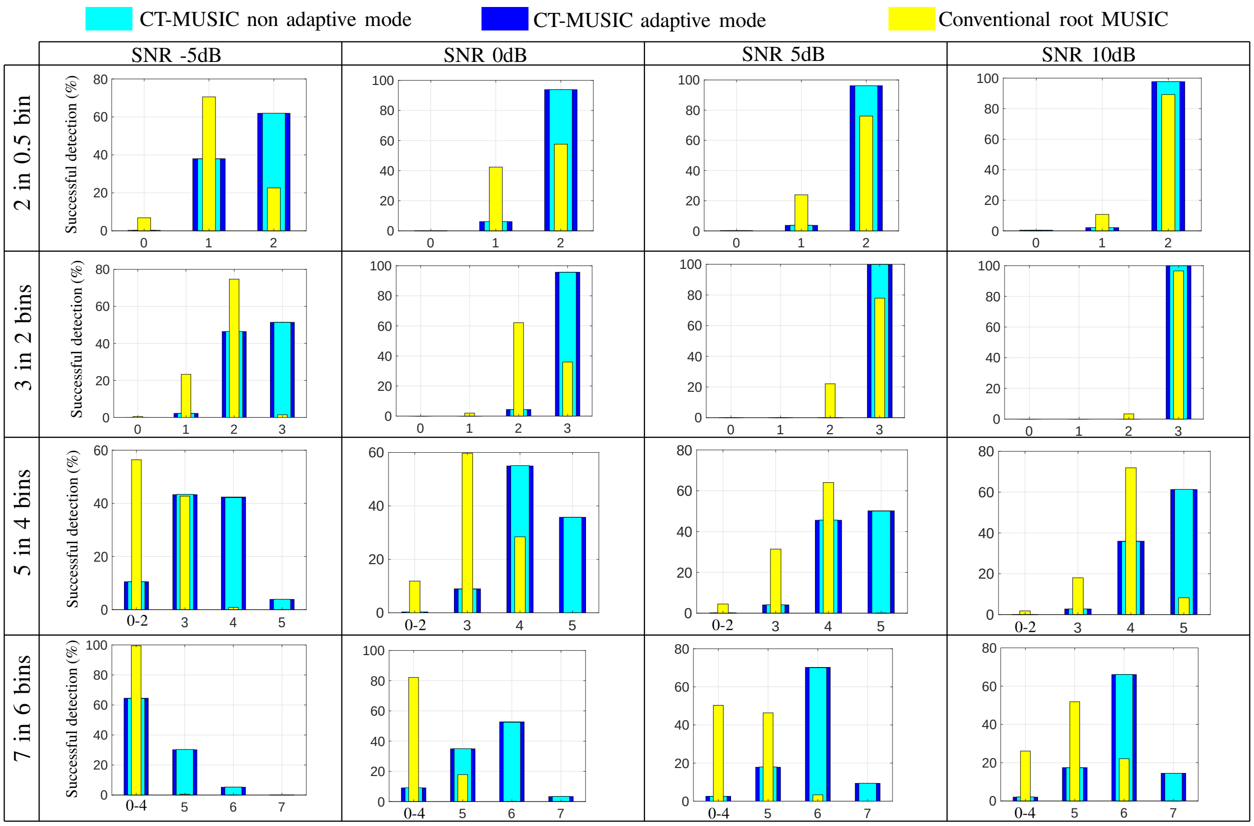

Fig. 4. Frequency estimation performance of adaptive and non-adaptive modes of the CT-MUSIC algorithm in comparison with conventional root-MUSIC algorithm. Noise color is "white".

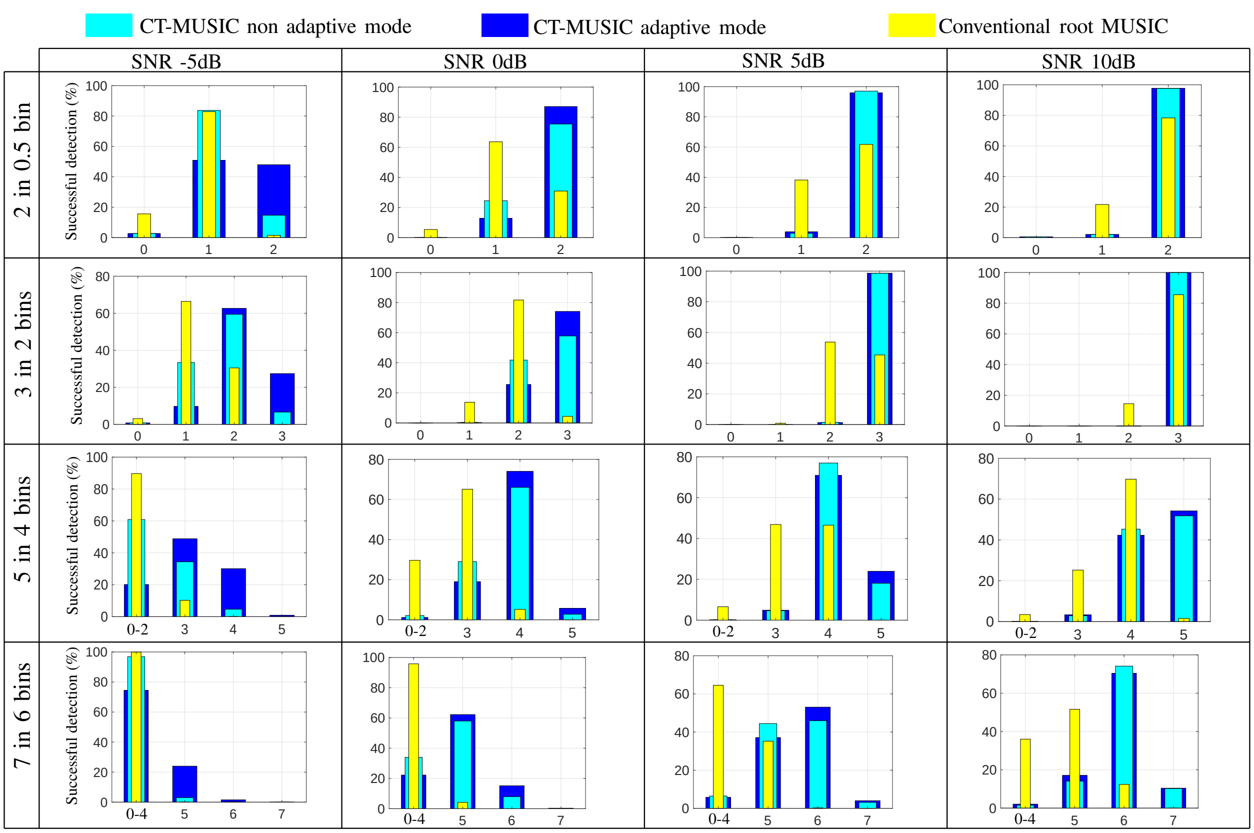

Fig. 5. Frequency estimation performance of adaptive and non-adaptive modes of the CT-MUSIC algorithm in comparison with conventional root-MUSIC algorithm. Noise color is "3G".

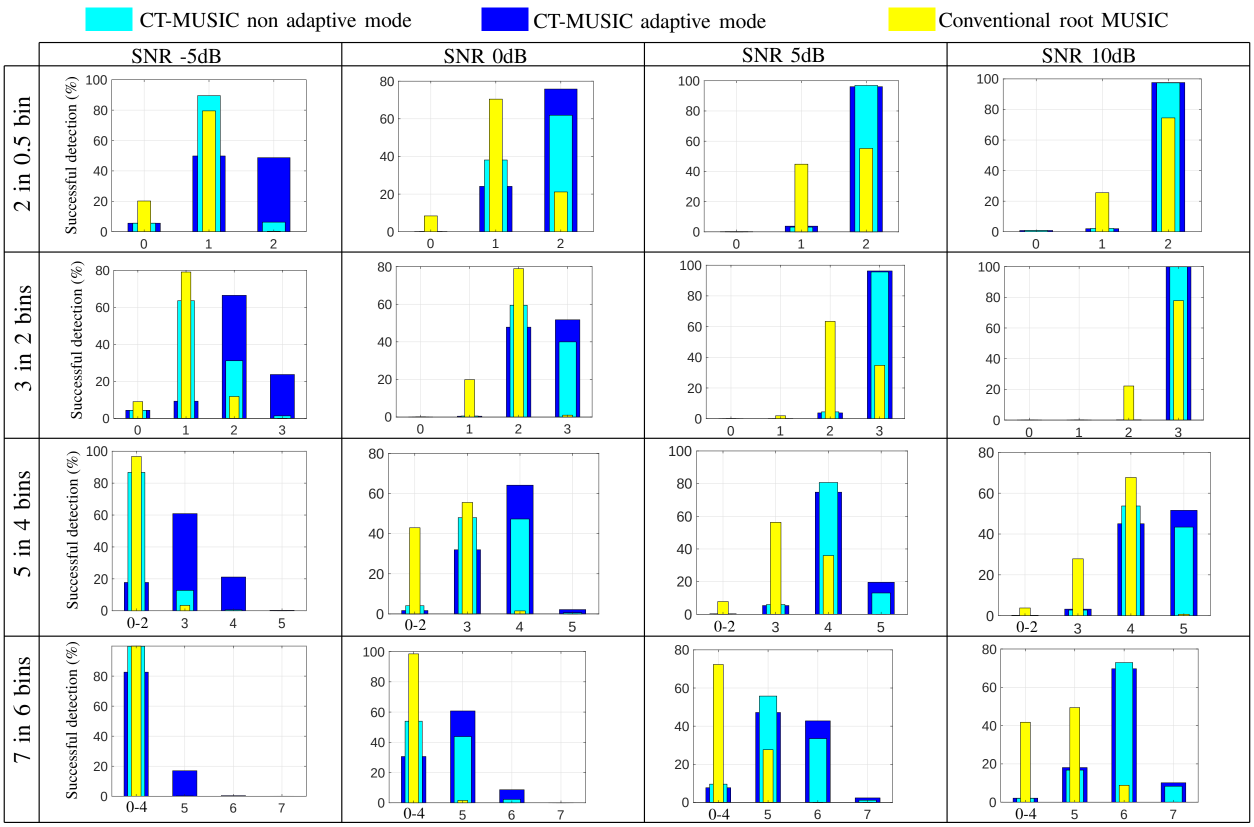

Fig. 6. Frequency estimation performance of adaptive and non-adaptive modes of the CT-MUSIC algorithm in comparison with conventional root-MUSIC algorithm. Noise color is "spiky".

with a Core-i7 4-core processor with 16 GB of DDR3 memory, running at 3.4 GHz. As can be seen from Table II our CT-MUSIC algorithm runs faster than the original root-MUSIC algorithm for sample sizes 16-256 and similar results are obtained up to about 512 samples. For larger number of samples our least square fits become the bottlenecks. This problem is solved by performing the least squares fits piecewise over slightly overlapping windows of 64-128 samples; we have achieved the best results with overlaps of 16 points. These least square fits can be done in parallel. In fact, since most of the operations of our algorithm are easily parallelisable, such as the matrix-vector multiplications or evaluations of $(\mathcal{F}_{\widehat{\nu}}(\omega))'$ on a grid of points, if executed on an appropriate platform, our algorithm can be made even faster by exploiting such parallelism.[9]

Figure 7 shows performance of the single adaptive learning mode and doubly adaptive learning mode of our CT-MUSIC algorithm with respect to the non-adaptive mode demonstrating the significance of dynamic adaptation and learning ability of CT-MUSIC in a changing environment. This experiment used 3 complex exponentials one DFT bin apart and with the lowest frequency chosen at random. The performance is expressed in terms of the percentage of runs during which the all the 3 frequencies were successfully estimated with an error tolerance of one half of a DFT bin. The SNR was -5dB with

[9]Timing reported in Table II refers only to completely sequential execution of CT-MUSIC algorithm.

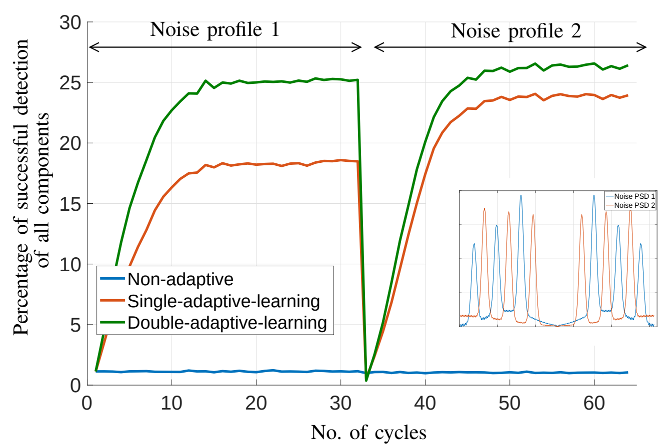

Fig. 7. Performance of the single adaptive learning mode and the doubly adaptive learning mode of the CT-MUSIC algorithm compared to the non-adaptive mode, demonstrating the additional gain obtained from dynamic adaptation to background noise in changing conditions. Experiment is for 128 samples, 3 components with a single DFT bin separation, -5dB SNR with two "spiky" noise profiles shown, abruptly switched at $t = 32$.

two PSD profiles; one used to produce noise during the first 32 update cycles of 16 runs and then abruptly changing to the second profile used for the remaining 32 cycles; these two PSD profiles are also shown on the same figure.

VII. Conclusion

In this paper we have presented a novel algorithm for frequency estimation and tracking in presence of high levels of colored noise, which we call the Continuous Time MUSIC algorithm. There are two reasons why our CT-MUSIC algorithm significantly outperforms the original root-MUSIC algorithm in terms of super-resolution and noise robustness.

Firstly, regardless of whether noise adaptation is used, matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ consisting of the values of chromatic derivatives of the model at the sampling instants and which is used in the estimation procedure is close (in the Frobenius norm) to a matrix of rank equal to the number n of complex exponentials in the input signal. Thus, all higher order chromatic derivatives are "forced " into (approximately) satisfying differential equations which a sum of n complex exponentials ought to satisfy. Thus, higher order derivatives of the model approximate the corresponding higher order derivatives of a linear combination of n complex exponentials. This, in virtue of the Taylor formula, forces our model *as a continuous time waveform* to approximate such a linear combination *everywhere between the sampling instants* rather than only at the sampling instants. The discrete time model cannot achieve this because the Shannon formula with relatively few sampling points provides rather poor approximation of the continuous time waveform between the sampling instants; see [15] for a detailed analysis of this fact.

Secondly, in the adaptive modes of our CT-MUSIC algorithm, as we have shown in our theoretical analysis in Section IV, using chromatic derivatives "matched" to the PSD of the noise ensures that the singular vectors of matrix $\boldsymbol{\mathcal{M}}_m^{\widehat{\alpha}}$ are a minimal noise induced perturbations of the corresponding singular vectors of the noise-free matrix $\boldsymbol{\mathcal{M}}_m^{s}$.

Both of these advantageous factors are due to the ability of our methods to operate effectively on the continuous time waveforms rather than on discrete samples of the signal. We have not compared CT-MUSIC with other frequency estimation methods because, in all likelihood, these methods also have their own continuous time counterparts to which they should be compared. We hope that demonstrated advantages of CT-MUSIC algorithm over (ROOT-)MUSIC algorithm will motivate DSP researchers to apply chromatic derivatives and approximations not only to other frequency estimation methods, but also to other problems which can be tackled more effectively in the continuous time domain than in the discrete time domain.

Extensively commented Matlab implementation of the CT-MUSIC algorithm which was used to produce the results reported in this paper can be downloaded at www.cse.unsw.edu.au/~ignjat/diff/ctmusic.zip. The authors will be happy to provide every additional help or information regarding this implementation or material presented in this paper.

REFERENCES

[1] A. Ignjatovic, "Chromatic derivatives and local approximations," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2998–3007, 2009.

[2] ——, "Chromatic derivatives, chromatic expansions and associated spaces," *East Journal of Approximations*, vol. 15, pp. 263–302, 2009.

[3] ——, "Local approximations based on orthogonal differential operators," *J. Fourier Anal. and Appl.*, vol. 13, no. 3, pp. 309 – 330, 2007.

[4] A. Ignjatovic, C. Wijenayake, and G. Keller, "Chromatic derivatives and approximations in practice - part i: A general framework," vol. 66, no. 6, pp. 1498–1512, 2018.

[5] ——, "Chromatic derivatives and approximations in practice - part ii: Nonuniform sampling, zero-crossings reconstruction and denoising," vol. 66, no. 6, pp. 1513–1525, 2018.

[6] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Prentice Hall; 1 edition, ISBN 0-13-258419-0, 1997.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[8] R. de Prony, "Essai experimentale et analytique," *J. Ecole Polytechnique (Paris)*, pp. 24–76, 1795.

[9] R. Kumaresan, D. W. Tufts, and L. L. Scharf, "A Prony method for noisy data: Choosing the signal components and selecting the order in exponential signal models," *Proceedings of the IEEE*, vol. 72, no. 2, pp. 230 – 233, 1984.

[10] V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophysical Journal International*, vol. 33, pp. 347–366, 1973.

[11] L. Marple, "Spectral line analysis by pisarenko and prony methods," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1979, pp. 159–161.

[12] A. Ignjatovic, "Frequency estimation using time domain methods based on robust differential operators," in *Proc. IEEE 10th International Conference on Signal Processing (ICSP)*, 2010.

[13] C. Wijenayake, Y. Xu, A. Madanayake, L. Belostotski, and L. T. Bruton, "RF analog beamforming fan filters using CMOS all-pass time delay approximations," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1061–1073, May 2012.

[14] L. Råde and B. Westergreen, *Mathematics Handbook for Science and Engineering*. Birkhäuser, 1995.

[15] G. Han, Y. Li, and M. Jiang, "How slow is shannon's reconstruction for bandlimited signals?" *Signal Processing*, vol. 111, pp. 26–30, 2015.