Subspace-based
Fundamental Frequency
Estimation
- Efficient Implementation of the MUSIC Estimator -
Title:
Subspace-based Fundamental Frequency Estimation

Master Programme:
Wireless Communication

Project period:
E7, fall semester 2007

Project group:
Group 710

Participants:
Jesper Rindom Jensen
Jesper Kjær Nielsen

Supervisor:
Søren Holdt Jensen
Mads Græsbøll Christensen

Copies: 6
Page numbers: 63

Date of completion: 20. December 2007

The content of this report is freely available, but publication (with reference source) may only be pursued due to agreement with the respective authors.
Preface

This report is written by project group 07gr710 at the department of Electronic Systems on Aalborg University during the 7th semester in the period spanning from September 1st, 2007 to December 20th, 2007. The project theme of the semester was "Scientific Communication" for which reason an article was written and a poster was created. The article is supported by worksheets. Both the article, the worksheets and the poster are enclosed in this report.

The reader should pay attention to the following on perusal of this report:

- The report is divided into three major parts:
  - The article.
  - The worksheets.
  - The poster.

- Figures, tables and equations are numerated consecutively according to the chapter number. Hence, the first figure in chapter one is named figure 1.1, the second figure figure 1.2 and so on.

- The article and each worksheet have their own bibliography.

Aalborg University May 30th, 2007

Jesper Rindom Jensen
<jesperrj@es.aau.dk>

Jesper Kjær Nielsen
<jkjaer@es.aau.dk>
Article

- Fast Implementation of Harmonic MUSIC for a Known and an Unknown Model Order -
FAST IMPLEMENTATION OF HARMONIC MUSIC FOR A KNOWN AND AN UNKNOWN MODEL ORDER

Jesper Rindom Jensen and Jesper Kjær Nielsen

Aalborg University, Denmark

ABSTRACT

In this paper we present a fast implementation of the subspace-based MULTiple SIgnal Classification (MUSIC) estimation criterion for harmonic signals with a known and an unknown model order, respectively. The proposed implementation improves the MUSIC criterion in two ways: First, we introduce a novel and fast implementation for evaluating the cost function of the MUSIC estimator involving only one FFT for known model order, and we extend it to the case for unknown model order. Second, we introduce an algorithm which use a low complexity subspace tracker and a gradient based minimization method depending on the minimum value of the MUSIC cost function. The performance gain obtained by the proposed algorithms is significant and enables real-time implementation for a known model order and faster computation in some cases for an unknown model order.

Index Terms— Frequency estimation, MULTiple SIgnal Classification (MUSIC), Harmonic MUSIC (HMUSIC), adaptive subspace tracking, harmonic signals in white noise.

1. INTRODUCTION

In this paper we present a fast implementation of the MULTiple SIgnal Classification (MUSIC) estimation criterion for fundamental frequency estimation with a known or an unknown model order. Fundamental frequency estimation of a periodic signal in white Gaussian noise is a classical signal processing problem and is encountered in practice in many speech and audio applications. This comprises among others speech and audio coding [1], automatic music transcription [2] and musical genre classification [3].

The fundamental frequency estimation problem can be summarized in the following way: We consider a periodic signal with a known or an unknown model order \( L \) and fundamental frequency \( \omega_0 \) corrupted by white Gaussian noise \( e[n] \). Using complex representation of signals for mathematical convenience this can be stated as

\[
x[n] = \sum_{k=1}^{L} A_k e^{j\omega_0 n} + e[n]
\]

where \( A_k = |A_k|e^{j\phi_k} \) is the complex amplitude of the \( k \)th harmonic. There are several unknown parameters in this signal model, but in this paper we will only treat the estimation of two of these: The fundamental frequency \( \omega_0 \) and the model order \( L \).

The MUSIC estimation criterion is a well-known subspace-based method that can be used for estimating the frequency of individual sinusoids corrupted with white Gaussian noise [4]. In [5] and [6] the MUSIC estimation criterion has been used to estimate the fundamental frequency and the model order of a periodic signal as in (1), and the resulting estimator was termed Harmonic MUSIC (HMUSIC). This estimation criterion will be the starting point of this paper, and we will refer to it as MUSIC.

The MUSIC algorithm for both a known and an unknown model order suffers from very high computational complexity. This is mainly due to two integral steps of the MUSIC algorithm:

1. The forming of an estimate of the auto-correlation matrix and the partitioning of its eigenspace into a signal and a noise subspace spanned by the most and the least dominant eigenvectors, respectively.
2. The minimization of a non-convex cost function which depends on the eigenvectors spanning the noise subspace.

The first issue was originally resolved by using an eigenvalue decomposition (EVD) to form the signal and noise subspaces, and it has a dominating cost in the order of \( O(M^3) \). Throughout the years comprehensive research has resulted in several subspace tracking algorithms with a lower computational complexity. An overview over some of these algorithms is given in [7] and [8]. If \( M \) is the length of a sample vector from the signal in (1) and \( L \) is the rank of the signal or the noise subspace, the subspace tracking algorithms can be classified into three groups according to their computational complexity as [8]

- High complexity algorithms with a dominating cost of \( O(M^2L) \) or \( O(M^4) \) operations.
- Medium complexity algorithms with a dominating cost of \( O(ML^2) \) operations.
- Low complexity (fast) algorithms with a dominating cost of \( O(ML) \) operations.

The low complexity subspace trackers introduce approximations in order to achieve the \( O(ML) \) cost. For example the Projection Approximation Subspace Tracking (PAST) algorithm in [9] is a famous example of the projection approximation. Other examples of low complexity subspace trackers are NIC [10], SWASVD3 [11], FAPI [12] and FDPM in [8]. The various approximations influences the performance of the subspace trackers. The PAST, NIC and FDPM algorithms, for example, do not guarantee an orthonormal subspace at each time step. Additionally, the mentioned low complexity subspace trackers do only track an arbitrary orthonormal basis of the subspace and not the eigenvectors for which reason they cannot be used directly with the MUSIC estimator for an unknown model order.

The only low complexity subspace tracker, that tracks the eigenvectors, is the PAST algorithm with deflation (PASTd) [13], but it works in a fundamentally different way than the other subspace trackers since it estimates the eigenvectors sequentially. Therefore, the low complexity subspace trackers cannot be used directly with MUSIC for an unknown model order for joint model order and fundamental frequency estimation.

Efficient implementation of the minimization of the non-convex cost function in MUSIC has not been treated in many publications.
In [14] root MUSIC was proposed where the MUSIC estimate is found by examining the roots of a polynomial, and in [6] an efficient implementation is proposed using an FFT for each column vector in the noise subspace. The formulation of the latter approach, however, requires the eigenvectors spanning the noise subspace to be calculated explicitly which is more demanding from a computational point of view. The reason for this is that the fast subspace tracking algorithms track an arbitrary orthonormal basis of the signal subspace. Therefore, the requirement for an efficient implementation of the non-convex cost function in MUSIC as proposed in [6] entails, in many situations, a less efficient subspace tracking algorithm.

In this paper we present several performance improvements of the existing algorithms for solving the problem in (1). For known model order we introduce a novel and fast implementation of the minimization of a non-convex cost function which involves just one FFT regardless of the model order and without direct access to the eigenvectors spanning the signal or noise subspace, but only an arbitrary orthonormal basis of the subspace. This enables the use of a fast signal subspace tracker and MUSIC coherently. The algorithm is also extended for the case of an unknown model order. Additionally, we improve the performance by exploiting the local convexity of the MUSIC cost function to refine our frequency estimate and lower the computational complexity even further. Last, the proposed algorithms are evaluated using synthetic and real world signals.

The paper is organized as follows. In section 2 we briefly brush equations and algorithms for it. The algorithms, however, can easily be modified for the matrix \( G \) which depends on the eigenvectors in \( S \) and \( L \) while the noise subspace matrix \( G \) only depends on \( L \). Additionally, the set of model orders \( L \) depends on \( \omega_0 \) since the harmonics are bounded by the Nyquist frequency.

2. THE MUSIC ESTIMATION CRITERION

In this section, we present the basics of the MUSIC estimation criterion for known and unknown model order. Consider the assumed model of the data in (1) and denote a sample of \( M \) consecutive samples drawn from the signal as the vector \( x[n] \) given by

\[
\begin{bmatrix}
x[n] \\
x[n+1] \\
\vdots \\
x[n+M-1]
\end{bmatrix}^T
\]  

(2)

where \((\cdot)^H\) denotes the transpose. If the phases are uniformly distributed on the interval \([-\pi, \pi]\), the auto-correlation function of the data vector \( x[n] \) is [15]

\[
R_{xx} = E \left\{ x[n] x^H[n] \right\} = APA^H + \sigma^2 I
\]  

(3)

where \( E \{ \cdot \} \) denotes the statistical expectation and \((\cdot)^H\) denotes the conjugate transpose. The scalar \( \sigma^2 \) is the variance of the complex white Gaussian noise assumed uncorrelated with the signal, \( I \) is the identity matrix and \( P \) is a diagonal matrix given by

\[
P = \text{diag}(\{P_1 \ P_2 \ \cdots \ P_L\})
\]  

(4)

where \( P_k \) is the power of the \( k \)-th complex exponential. Further, the matrix \( A \) is given by

\[
A = \begin{bmatrix} a_1 \ a_2 \ \cdots \ a_L \end{bmatrix}
\]  

(5)

where the vector \( a_i \) is given by

\[
a_i = [1 \ \ e^{j\omega_0} \ \ \cdots \ \ e^{j(M-1)\omega_0}]^T.
\]  

(6)

The EVD of (3) is [8]

\[
R_{xx} = [S \ G] \begin{bmatrix} D_S + \sigma^2 I \ & \ 0 \\ 0 & \sigma^2 I_G \end{bmatrix} [S^H \ G^H]
\]  

(7)

where \( S = [s_1 \ s_2 \ \cdots \ s_L] \) contains the \( L \) dominant orthonormal eigenvectors that span the signal subspace and \( G = [g_1 \ g_2 \ \cdots \ g_{M-L}] \) contains the \( M-L \) least dominant orthonormal eigenvectors that span the noise subspace. Together, \( U = [S \ G] \) forms the unitary eigenspace of \( R_{xx} \). The matrix \( D_S \) is diagonal and contains the \( L \) largest eigenvalues of \( AP A^H \) in decreasing order. The noise subspace is orthogonal on the signal subspace which is also spanned by the columns of \( A \). Therefore, the noise subspace is orthogonal on \( A \) which yields

\[
A^H G = 0.
\]  

(8)

This leads to the MUSIC estimator for a known model order given by

\[
\hat{\omega}_0 = \arg \min_{\omega_0 \in \Omega} \| A^H G \|_F^2
\]  

(9)

where \( \| \cdot \|_F \) denotes the Frobenius norm and \( \Omega \) is a set of candidate fundamental frequencies with cardinality \(|\Omega|\). The cost function of the MUSIC estimator is

\[
J = \| A^H G \|_F^2 = \text{Tr} \left\{ A^H GG^H A \right\}
\]  

(10)

where \( \text{Tr} \{ \cdot \} \) denotes the trace. Since the value of the estimator in (9) varies with the model order \( L \), the MUSIC estimator must be normalized to enable joint estimation of fundamental frequency and model order. This yields the MUSIC estimator for an unknown model order [6]

\[
(\hat{\omega}_0, \hat{L}) = \arg \min_{\omega_0 \in \Omega, \ L \in \mathcal{L}} \| A^H G \|_F^2
\]  

(11)

where \( \mathcal{L} \) is a set of candidate model orders with cardinality \(|\mathcal{L}|\). The matrix \( A \) depends on both \( \omega_0 \) and \( L \), while the noise subspace matrix \( G \) only depends on \( L \). Additionally, the set of model orders \( \mathcal{L} \) depends on \( \omega_0 \) since the harmonics are bounded by the Nyquist frequency.

3. EFFICIENT IMPLEMENTATION FOR AN KNOWN AND AN UNKNOWN MODEL ORDER

The estimators in (9) and (11) use the common cost function which in (10) is seen to depend on the eigenvectors in \( G \) spanning the noise subspace. The cost function could, however, be rewritten using \( I = SS^H + GG^H \) into

\[
J = ML - \| A^H S \|_F^2 = ML - \text{Tr} \left\{ A^H SS^H A \right\}
\]  

(12)

which depends on the eigenvectors in \( S \) spanning the signal subspace. Thus for fastest evaluation of \( J \), the formulation involving the subspace with the lowest rank, the minor subspace, should be used. In this section, we assume that the matrix \( S \) spanning the signal subspace has the lowest rank for which reason we develop the equations and algorithms for it. The algorithms, however, can easily be modified for the matrix \( G \) spanning the noise subspace.

Subspace trackers can be classified into two groups: Those that track the eigenvectors in \( S \) and those that track an arbitrary orthonormal basis \( W \) of the subspace spanned by \( S \). The relation between these two matrices is

\[
S = WQ
\]  

(13)

where \( Q \) is an arbitrary unitary matrix. The fast subspace trackers find \( W \) from which \( S \) in general cannot be recovered. The projection matrix of \( S \), however, can be found since

\[
SS^H = WQ(WQ)^H = WW^H.
\]  

(14)
Thus, $WW^H$ can be used in (12) instead of $SS^H$.

For an unknown model order fast subspace trackers cannot be used. This is because projection matrices of subsets of $S$ cannot uniquely be recovered from subsets of $W$. To demonstrate this, partition $S = \{S_1, S_2\}$ and $W = \{W_1, W_2\}$ into two subsets whose dimensions are pairwise equal. Inserting this into (14) readily yields

$$S_1S_1^H + S_2S_2^H = W_1W_1^H + W_2W_2^H$$

(15)

from which $S_1S_1^H$ cannot be recovered since only $W$ is known. That is, the subspace for each candidate model order in $L$ should be tracked separately if fast subspace trackers should be used.

In this section we will first propose a novel and efficient implementation of the MUSIC estimator for a known model order. The resulting algorithm finds the estimate of the MUSIC estimator in (9) from an arbitrary basis $W$ of the signal subspace which enables the use of fast subspace trackers. Second, the algorithm is extended to the MUSIC estimator for an unknown model order. The resulting algorithm still suffers, as the algorithm in [6], from the requirement of $S$ to be known, but it has in many situations a lower computational complexity.

### 3.1. The MUSIC Estimator for a Known Model Order

First we consider the case where the model order and an arbitrary orthonormal basis $W$ of the signal subspace are known. Define the matrix $M = I - WW^H$ and note that it is a Hermitian matrix. Evaluating the trace of the cost function in (10) yields

$$J = \sum_{i=1}^{L} a_i^H Ma_i = \sum_{i=1}^{L} J_i$$

(16)

where $J_i$ denotes the $i$th partial cost function. The partial cost function $J_i$ can be rewritten into

$$J_i = \begin{bmatrix}
1 & e^{-j\omega_1} & \cdots & e^{-j(M-1)\omega_1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & e^{j\omega_1} & \cdots & e^{j(M-1)\omega_1} \\
\end{bmatrix}^T \begin{bmatrix}
m_{11} & \cdots & m_{1M} \\
m_{M1} & \cdots & m_{MM} \\
m_{1M} & \cdots & m_{M-M+1} \\
\vdots & \ddots & \vdots \\
m_{M1} & \cdots & m_{M-M+1} \\
\end{bmatrix} \begin{bmatrix}
1 & e^{j\omega_1} & \cdots & e^{j(M-1)\omega_1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & e^{-j\omega_1} & \cdots & e^{-j(M-1)\omega_1} \\
\end{bmatrix}
$$

$$= \sum_{r=-M+1}^{M-1} c[r]e^{jr\omega_1}$$

(17)

where $c[r]$ is the sum of the elements on the $r$th diagonal of $M$. Exploiting the Hermitian property of $M$, (17) can be rewritten as

$$J_i = c[0] + \sum_{r=1}^{M-1} c[r]e^{jr\omega_1} + \sum_{r=-M+1}^{-1} c[r]e^{-jr\omega_1}$$

$$= c[0] + \sum_{r=1}^{M-1} c^*[r]e^{-jr\omega_1} + \sum_{r=-M+1}^{-1} c^*[r]e^{jr\omega_1}$$

$$= 2\text{Re}\sum_{r=0}^{N-1} g[r]e^{-jr\omega_1} = 2\text{Re}\left[G(e^{j\omega_1})\right]$$

(18)

where $(\cdot)^*$ denotes the complex conjugate and $g[r]$ is given by

$$g[r] = \begin{cases}
  c[0]/2 & \text{for } r = 0 \\
  c^*[r] & \text{for } r = 1, 2, \ldots, M-1 \\
  0 & \text{for } r = M, M+1, \ldots, N-1
\end{cases}$$

(19)

If the discrete-time Fourier transform in (18) is sampled with $\omega = 2\pi k/N$, the discrete Fourier transform $J_i[k]$ is obtained. This is desirable from a computational complexity point of view since it can be computed by an FFT of $g[r]$ with $k$ as the frequency index. The dominant cost for evaluating (18) is thus the order of $O(N \log_2 N)$ where $N$ is the FFT-length.

The cost function in (16) can be computed from (18). First, we simply evaluate (18) for $i = 1$ and obtain $J_1[k]$ from which we create the partial cost function vector $J_1$ containing the values corresponding to the desired subset $\Omega$ of fundamental frequencies. Next, for $i = 2, 3, \ldots, L$ we extract every $i$th sample from $J_1[k]$ for $k \mapsto \omega \in \Omega$, and we obtain $J_2, J_3, \ldots, J_L$, all of the same length as $J_1$. Note, that since we extract the samples from $J_1[k]$, we need to restrict the maximum frequency in $\Omega$ to be less than $\pi/L$.

The discrete version of the cost function in (16) defined on the subset $\Omega$ of fundamental frequencies is thus the sum of the partial cost.
function vectors with the resolution determined by the FFT-length, i.e.

\[ J = \sum_{i=1}^{L} J_i. \]  

(20)

Using this approach, the dominant cost of computing the cost function of MUSIC in (16) from the sequence \( g[r] \) is still in the order of \( O(N \log_2 N) \) computations.

In (18) the discrete fourier transform is computed from the sequence \( g[n] \) defined in (19). These coefficients are computed by, first, computing the matrix \( M \) from an arbitrary basis \( W \) of the signal subspace and, second, summing the diagonals of \( M \). This has a cost in the order of \( O(M^2) \) and \( O(M^2) \), respectively and can be improved so that the \( O(M^2) \) part is saved. Further, the resulting algorithm has some nice properties with respect to the implementation proposed in [6]. Another obvious advantage of our implementation is that, instead of spanning the noise subspace.

As seen from the table, the dominant cost is in the order of \( O(M^2 L) + O(N \log_2 N) \). This cost should be compared to the dominant cost \( O((M-L)N \log_2 N) + O((M-L)N|\Omega|) \) of the fast implementation proposed in [6]. Another obvious advantage of our algorithm is that it computes the MUSIC estimator from an arbitrary orthonormal basis of the signal subspace and not the eigenvectors spanning the noise subspace.

### 3.2. The MUSIC estimator for an Unknown Model Order

We now extend our fast implementation of the MUSIC estimator for a known model order to the case with an unknown model order in (11). The extension is straightforward, but it comes with one limitation. As already discussed the limitation is that, instead of just an arbitrary basis of the signal subspace, the eigenvectors of the auto-correlation matrix in (7) spanning the signal subspace must be known.

| Input | 
| - Eigenvectors spanning the signal subspace \( S \in \mathbb{C}^{M \times L_{\max}} \)  
| - FFT-length \( N \)  
| - Subset of candidate frequencies \( \Omega = \{\omega_{\min}, \cdots, \omega_{\max}\} \) with cardinality \(|\Omega|\)  
| - Subset of candidate model orders \( \mathcal{L} = \{L_{\min}, \cdots, L_{\max}\} \) with cardinality \(|\mathcal{L}|\) |

| for \( l = L_{\min}, \cdots, L_{\max} \) |
| Step 1: Find \( g_l[n] \)  
| if \( l = L_{\min} \)  
| \[ c' = Me_1 - \sum_{k=1}^{M} [S]\{k:h+k-M-1,l\}|S|_{k,l}^H \]  
| \( \mathcal{O}(M^2) \)  
| else  
| \[ c_l' = c_{l-1}' - \sum_{k=1}^{M} [S]\{k:h+k-M-1,l\}|S|_{k,l}^H \]  
| \( \mathcal{O}(M^2) \)  
| end |

| g_l[r] = \begin{cases}  
| c_l[r]/2 & \text{for } r = 0 \\
| 0 & \text{for } 1 \leq r < M \\
| c_0'[r] & \text{for } M \leq r < N \end{cases} \]  
| \( \mathcal{O}(1) \)  
| Step 2: Compute \( J_{l,1}[k] \) from an FFT of \( g_l[r] \)  
| \[ J_{l,1}[k] = 2 \text{Re} \sum_{r=0}^{N-1} g_l[r]e^{-j2\pi rk/N} \]  
| \( \mathcal{O}(N \log_2 N) \)  
| Step 3: Downsample \( J_{l,1}[k] \)  
| \[ J_l[k] = \sum_{i=1}^{l} J_{l,i}[k] \forall k \mapsto \omega \in \Omega_l \]  
| \( \mathcal{O}(|\Omega_l|) \) |

### Table 2: Fast evaluation of the MUSIC estimator for unknown model order.

The MUSIC estimator for unknown model order is obtained by extending the MUSIC estimator for known model order to be evaluated for each candidate model order in the subset \( \mathcal{L} \) and then seeking the joint minimum among the candidate fundamental frequencies and model orders. Thus for all \( L \in \mathcal{L} \), the MUSIC estimator for unknown model order is evaluated using the proposed implementation in subsection 3.1 for a subset of candidate fundamental frequencies \( \Omega_L \) dependent on the model order where the set \( \Omega_L \) is a subset of \( \Omega \) such that \( \Omega_L = \{\omega|\omega \in \Omega, \omega < \pi/L\} \). This dependence is required since the largest harmonic in the observed signal in (1) is bounded by the Nyquist frequency.

The proposed implementation structure entails recalibration of the sequence \( g_l[n] \) for all \( L \in \mathcal{L} \). This can be done recursively if equation (21) is rewritten into

\[ c' = Me_1 - \sum_{j=1}^{M} \sum_{k=1}^{M} [W]_{k:h+k-M-1,l}|W|_{k,l}^H \]  

(22)

from which the recursive form for \( l = 2, \cdots, L_{\max} \) is obtained as

\[ c_l' = c_{l-1}' - \sum_{k=1}^{M} [W]_{k:h+k-M-1,l}|W|_{k,l}^H \]  

(23)
which is determined by the threshold estimate can be entirely omitted if the gradient is sufficiently small this approach is summarized in table 3. Note, that the update of the section is used. The idea is sketched in figure 1. The algorithm for which enables the use of fast subspace trackers. If the cost function of the MUSIC estimator for both a known and an unknown model order, the model order can be assumed constant and the frequency estimation abilities of the proposed estimators in relation to the fast implementation in [6] was made for MUSIC for a known model order. In the case where different values of |L| was small and largest when |L| was large. Furthermore it can be seen that η peaked in the interval from about M = 40 to M = 100. It should be emphasized that this comparison only were made for MUSIC for a known model order. In the case where the model order is unknown the proposed implementation would of- ten be faster than the implementation in [6], but it depends on the chosen estimation setup.

5. SIMULATION RESULTS

In this section, the performance gain of our fast implementation of the MUSIC estimators in relation to the fast implementation in [6] is evaluated, and the frequency estimation abilities of the MUSIC estimators with known and unknown model order are simulated.

5.1. Comparison of Computational Complexities of the MUSIC Estimator for a Known Model Order

The computational complexities in terms of the dominant cost of the proposed fast implementation and the fast implementation in [6] was found in section 3. To compare these costs we need the total cost and not only the dominant cost. These can be found to

\[ f_1(M, L, N, |Ω|) = M^2(2L - 1) + 4N \log_2(N) - 6N + 10 + (L - 1)|Ω| + M \]
\[ f_2(M, L, N, |Ω|) = (M - L)(4N \log_2(N) - 6N + 8) + MN - LN + |Ω|(L - 1)(M - L - 1) \]

where \( f_1 \) and \( f_2 \) are the computational cost of the proposed implementation and the implementation in [6], respectively. The value \( 4N \log_2(N) - 6N + 8 \) denotes the computational complexity of the split radix-2 FFT algorithm [16]. The two costs were calculated for different values of \( M, L \) and \( N \) with \(|Ω|=100\) and the results are shown in Fig. 2. For the chosen values it can be concluded that the performance gain measured by the ratio \( \eta \) between the computational cost of the algorithm was smallest when \( L \) was large and \( N \) was small and largest when \( L \) was small and \( N \) was large. Furthermore it can be seen that \( \eta \) peaked in the interval from about \( M = 40 \) to \( M = 100 \).

5.2. Frequency Estimation using MUSIC Estimator for a Known Model Order

Next, the frequency estimation abilities of the proposed estimators were simulated. As mentioned in section 3 the signal subspace can

![Fig. 1: The different stages of the MUSIC algorithm.](image-url)
be tracked with a fast subspace tracker when the model order $L$ is known. Therefore, we evaluated the fast implementation of the MUSIC estimator for a known model order by comparing implementations where the signal subspace was tracked by the PAST and the FDPM subspace trackers, respectively. The MUSIC estimator was tested on a synthetic signal since the model order had to be known. For this simulation the experimental setup was as follows. The synthetic signal was generated as a harmonic signal with model order of three, a fundamental frequency changing in steps and sampled with a sampling frequency $f_s = 6000$ Hz. The set of candidate frequencies $\Omega$ was chosen to be in the interval from 60 Hz to 400 Hz with a resolution of 10 Hz. This resulted in an FFT-length of 3000. The PAST and FDPM algorithms were initialized to track a signal subspace of size $M \times L$ where $M = 82$ and $L = 3$. The forgetting factor of the PAST algorithm was set to 0.975 and the step size $\mu$ of the FDPM algorithm was set to $1/||x[n]||^2$. The cost function threshold was set to $\alpha = 10$ and the gradient threshold to $\beta = 0.5$.

In Fig. 3 the estimated frequencies from the PAST and FDPM based MUSIC estimators are shown versus the true frequency of the synthetic signal with an SNR of 40 dB. It can be seen from the figure that the frequency estimate based on both of the trackers led to the true frequency estimate except in the frequency transitions. This has two explanations. First, the assumed model in (1) did not fit in the transitions since it is only valid for stationary signals. Second, it takes some time before the trackers had converged to the signal subspace after a transition. Furthermore, it can be seen that the FDPM algorithm tracked the signal subspace faster than the PAST algorithm which was expected according to [8]. Besides the frequency estimates, the mode of the two estimators are shown in Fig. 3. The three modes denote whether, for mode 0, the full MUSIC method from table 1 was used, for mode 1, the estimated frequency was updated using a gradient method or, for mode 2, the frequency estimate was not updated at all. For the case where the SNR was 40 dB, it can be seen that mode 0 was rarely entered and that mode 1 and 2 were entered often. This is desirable from a computational point of view.

The values of the cost functions at each time step are shown in Fig. 4. Here it is seen that the cost function peaked when there was a transition in the fundamental frequency and thereby it exceeded the threshold $\alpha$. Comparing Fig. 4 with Fig. 3 shows that the MUSIC estimators used the full MUSIC method when $\alpha$ was exceeded before returning to just updating the estimated frequency with the gradient method for $J < \alpha$. It should be stressed that the performance of the MUSIC estimator was sensitive to changes in $\alpha$ since the gradient method converged to the wrong frequency if $\alpha$ was chosen too large.

The simulations were run again with the same experimental setup except for an SNR of 10 dB. The result is shown on Fig. 5. The results depicted by the figure show that the MUSIC estimators retained their ability to estimate the fundamental frequency with an SNR of 10 dB. The bottom figure shows that the estimators tended to be more often in mode 0 and less in mode 2 as compared to the experiment with an SNR of 40 dB.

Fig. 2: The performance gain obtained by calculating the cost function of the MUSIC estimator with known model order in (9) expressed as the ratio $\eta$ between our proposed algorithm and the algorithm in [6] with a cardinality $|\Omega| = 100$.

Fig. 3: True frequency and frequency estimates using PAST and FDPM based MUSIC estimators for a known model order (top) and estimation modes (bottom) with an SNR of 40 dB. Mode 0 indicates use of the full MUSIC method, mode 1 indicates updating the estimated frequency using a gradient method and mode 2 indicates no update of the estimate.

Fig. 4: Values of the cost functions of the PAST and FDPM based MUSIC estimators with an SNR of 40 dB.
Fig. 5: True frequency and frequency estimates using PAST and FDPM based MUSIC estimators for a known model order (top) and estimation modes (bottom) with an SNR of 10 dB. Mode 0 indicates use of the full MUSIC method, mode 1 indicates updating the estimated frequency using a gradient method and mode 2 indicates no update of the estimate.

Fig. 6: Spectrogram of a violin signal (top), frequency estimates using MUSIC estimators for an unknown model order with fast subspace tracking (middle) and estimation modes (bottom) with an SNR of 40 dB. Mode 0 indicates use of the full MUSIC method, mode 1 indicates subspace tracking and updating the estimated frequency using a gradient method and mode 2 indicates subspace tracking and no update of the estimate.

5.3. Frequency Estimation using MUSIC Estimator for an Unknown Model Order

In a similar manner, the frequency estimation abilities of the MUSIC estimator for an unknown model order were evaluated through simulations. A violin signal with an SNR of 40 dB and 10 dB, respectively, and downsampled to a sampling frequency of \( f_s = 11025 \) Hz were used for these simulations. The set of candidate frequencies \( \Omega \) was chosen to be in the interval from 60 Hz to 1000 Hz with a resolution of 3.125 Hz. This resulted in an FFT-length of 3528.

Only the FDPM subspace tracker was used for tracking the signal subspace and, as in the previous simulations of the MUSIC estimators, it was initialized to track a signal subspace of size \( M \times L_{\text{max}} \) where \( M = 110 \) and \( L_{\text{max}} \) was the maximum candidate model order in \( L \in \{5, \cdots, 13\} \). The step size \( \mu \) of the FDPM algorithm was \( 1/||\hat{\pi}[n]||^2 \). The cost function threshold was \( \alpha = 1.3 \cdot 10^{-3} \) and the gradient threshold was \( \beta = 0.5 \). The results are shown in Fig. 6 and Fig. 7, respectively.

The SNR in the first simulation illustrated in Fig. 6 was 40 dB. As shown in the figure, the MUSIC estimator for an unknown model order showed good frequency estimation abilities. Only in the frequency transitions we obtained spurious frequency estimates since the assumed signal model in (1) did not fit under these circumstances. Besides this, it can be seen that the full MUSIC method only was used around the transitions. Meanwhile a gradient update or no update were used which, as mentioned previously, are the cheapest computational tasks. As for the simulations of the MUSIC estimator for a known model order, it should be stressed that the performance of the MUSIC estimator for an unknown model order was sensitive to changes in \( \alpha \), and should be chosen with caution to avoid spurious frequency estimates.

When the simulation was run with an SNR of 10 dB a more incorrect frequency estimation was obtained as shown in Fig. 7. This was especially the case in the time interval from 1 to 3 seconds. The reason for this should be found by looking at the spectrogram in the
upper part of Fig. 7. As can be seen the second harmonic of the signal was dominant compared to the first harmonic of the signal within this time slot. Therefore, when noise was applied, the second harmonic was falsely identified as the fundamental frequency. Besides this, the MUSIC estimator for an unknown model seemed to retain good frequency estimation abilities at an SNR of 10 dB. Like in the simulation of the MUSIC estimator for a known model order, an increase in the noise power entailed that mode 0 was entered more often.

6. CONCLUSION

In this paper, we have presented fast implementations of the MUSIC estimator for a known and an unknown model order. These implementations made it possible to calculate the whole frequency spectrum by use of only one FFT for the MUSIC estimator with known model order and by use of \( L \) FFTs for the MUSIC estimator for an unknown model order. For a known model order we also enabled the possibility of using an arbitrary basis of the signal subspace. Furthermore, we showed how the frequency estimate can be refined by use of a gradient method dependent of the value of the cost function of both estimation criteria and for the MUSIC estimator for an unknown model order we showed how the EVD can be replaced by a fast subspace tracker likewise dependent on the cost function. The application of the two proposed methods was shown through simulations. The proposed implementation of the MUSIC estimators were evaluated on a synthetic signal for a known model order and on a violin signal for an unknown model order, respectively. These simulations showed that the proposed methods had the desired abilities with respect to correct frequency estimation and performance gain.

7. REFERENCES

Worksheets
- Subspace-based Fundamental Frequency Estimation -
# Table of Contents

1 Spectral estimation using the MUSIC method .................................................. 1
   1.1 Overview of the estimation problem ....................................................... 1
   1.2 Description of the subspace-based method ........................................... 2
   1.3 The MUSIC method ............................................................................. 9
   Bibliography .............................................................................................. 11

2 Joint spectral and order estimation using the HMUSIC method ......................... 12
   2.1 The HMUSIC method ........................................................................... 12
   Bibliography .............................................................................................. 14

3 Adaptive Subspace tracking with the PAST and PASTd algorithms ..................... 15
   3.1 Relationship between PAST and MUSIC .............................................. 15
   3.2 Review of Weighted Recursive Least Squares ....................................... 16
   3.3 The PAST Algorithm .......................................................................... 21
   3.4 The PASTd algorithm .......................................................................... 22
   3.5 Examples of the MUSIC method using the PAST and PASTd algorithms 23
   Bibliography .............................................................................................. 26

4 Optimized Fast Implementation of the MUSIC Algorithm ................................. 28
   4.1 Fast Subspace Tracking for a Known and an Unknown Model order ........ 29
   4.2 Fast Implementation of the Calculation of the MUSIC Cost Functions ... 29
   4.3 Fast and Refined Frequency Estimation Using a Gradient Method .......... 34
   4.4 Examples Frequency Estimation using the MUSIC Estimator ............... 36
   Bibliography .............................................................................................. 37
Spectral estimation using the MUSIC method

Spectral estimation is an integral part of many signal processing applications. The classical (nonparametric) approach to spectral estimation is to use the Fourier transform. The primary advantage of this approach is that the method works for all stationary signals [Therrien, 1992, p. 586]. That is, no assumptions on the underlying signal (except for stationarity) are made. The frequency resolution of the spectrum is the major disadvantage of the Fourier transform approach and requires long data sets in order to accurately display closely spaced sinusoids within the signal. For this reason, other methods to estimate the spectrum of a signal have been developed. These are in general named parametric spectral estimation methods and are characterized by a build-in assumption on the model of the signal of interest. The spectral estimation problem is thereby reduced to an estimation of the parameters of the assumed model. The MUltiple SIgnal Classification (MUSIC) method is one of the many parametric methods and it will be the main topic of this worksheet.

1.1 Overview of the estimation problem

The spectral estimation problem can be summed up to [Stoica and Moses, 1997, p. 1]:

From a finite record of a stationary data sequence, estimate how the total power is distributed over frequency.

As described in the introduction, there exist several approaches to the spectral estimation problem, and they are generally partitioned into two overall groups: nonparametric and parametric approaches. The MUSIC method belongs to a class of approaches that are designated subspace-based parametric approaches where the term subspace-based refers to the partitioning of the data into a signal and a noise subspace [Therrien, 1992, p. 615]. This will be described in greater detail later. The term parametric means that
the observed data is assumed to fit into a certain model whose parameters are estimated from the observed data. For the MUSIC method, the model of the data is complex exponentials in white noise [Therrien, 1992, p. 615].

The primary advantage of the parametric approaches is that the estimation is very precise, if the the observed data fits the assumed model. If the condition is not fulfilled, the parametric approach may perform worse in terms of precision as compared to the classical nonparametric approaches [Stoica and Moses, 1997, p. 2]. For the rest of this worksheet, however, it is assumed that the observed data fits or at least approximates the assumed model very well for which reason the outlined issue not will be considered any further.

1.2 Description of the subspace-based method

In the subspace-based method the model of the observed data is [Therrien, 1992, p. 619]

\[ x[n] = \sum_{k=1}^{L} A_k e^{j\omega_k n} + e[n], \]

where \( x[n] \) is the assumed model of the observed sampled data, \( A_k = |A_k|e^{j\phi_k} \) is the complex amplitude of the \( k \)th complex exponential, \( \omega_k \) is the normalized angular frequency of the \( k \)th complex exponential and \( e[n] \) is complex white noise uncorrelated with the signal. The magnitude \( |A_k| \) and the phase \( \phi_k \) are real random variables with \( \phi_k \) uniformly distributed on the interval \([−\pi, \pi]\). The second-order moment or power of \( |A_k| \) is denoted by \( P_{A_k} \), and the variance or power of \( e[n] \) is denoted by \( \sigma_e^2 \).

In equation 1.1 a complex representation of the signal \( x[n] \) is used. This is convenient from a mathematical point of view and a generalization of the real-valued case. A real-valued signal can be transformed into an analytic (complex) signal with the Hilbert transform.

The power spectral density (PSD) of \( x \) can be found to [Stoica and Moses, 1997, p. 141]

\[ S_{xx}(\omega) = 2\pi \sum_{k=1}^{L} P_{A_k} \delta(\omega - \omega_k) + \sigma_e^2, \]

where \( \delta(\omega - \omega_k) \) is the Dirac delta function. A spectrum for \( L = 3 \) is sketched in figure 1.1. Thus, the estimation problem is to find the unknown parameters of the spectrum which are \( \omega_k \), \( P_{A_k} \) and \( \sigma_e^2 \).

In the following subsections the parameter estimation with the subspace-based method will be treated in three forms with increasing complexity.
1.2. Description of the subspace-based method

1. The subspace-based method is described with only one complex exponential. That is \( L = 1 \).

2. The subspace-based method is described with only two complex exponential. That is \( L = 2 \).

3. The subspace-based method is described with \( L \) complex exponential. The first two forms are, of course, special cases of the third form, but they serve as a good introduction to the idea behind the subspace-based method for which reason they are considered.

1.2.1 The subspace-based method for one complex exponential

Consider the assumed model of the data in equation 1.1 and let \( L = 1 \). Take \( M \) consecutive samples and denote them

\[
\mathbf{x} = \begin{bmatrix} x[0] & x[1] & \cdots & x[M-1] \end{bmatrix}^T, \tag{1.3}
\]

\[
\mathbf{a}_1 = \begin{bmatrix} 1 & e^{j\omega_1} & \cdots & e^{j(M-1)\omega_1} \end{bmatrix}^T \tag{1.4}
\]

and

\[
\mathbf{e} = \begin{bmatrix} e[0] & e[1] & \cdots & e[M-1] \end{bmatrix}^T, \tag{1.5}
\]

where (·)\(^T\) denotes the transpose. Equation 1.1 can now be rewritten into vector notation as

\[
\mathbf{x} = \mathbf{A}_1 \mathbf{a}_1 + \mathbf{e}. \tag{1.6}
\]

The next step is to compute the autocorrelation matrix of the vector \( \mathbf{x} \). That is

\[
\mathbf{R}_{\mathbf{x}\mathbf{x}} = E \{ \mathbf{x}\mathbf{x}^H \} \tag{1.7}
\]

where \( E \{ \cdot \} \) denotes the statistical expectation and (·)\(^H\) denotes the Hermitian or the conjugate transpose. Inserting \( \mathbf{x} \) gives

\[
\mathbf{R}_{\mathbf{x}\mathbf{x}} = E \{ (\mathbf{A}_1 \mathbf{a}_1 + \mathbf{e})(\mathbf{A}_1 \mathbf{a}_1 + \mathbf{e})^H \} = E \{ (\mathbf{A}_1 \mathbf{a}_1 + \mathbf{e})(\mathbf{A}_1^H \mathbf{a}_1^H + \mathbf{e}^H) \} \tag{1.8}
\]

\[
= E \{ |\mathbf{A}_1|^2 \mathbf{a}_1 \mathbf{a}_1^H \} + E \{ \mathbf{e}\mathbf{e}^H \} + E \{ \mathbf{A}_1 \mathbf{a}_1 \mathbf{e}^H \} + E \{ \mathbf{A}_1^H \mathbf{e}\mathbf{a}_1^H \}. \tag{1.9}
\]

The autocorrelation matrix now consists of four terms:
1 Spectral estimation using the MUSIC method

1. In the first term $a_1$ is a non-random vector so $a_1a_1^H$ can be moved outside the expectation operator. Left inside the expectation operator is $|A_1|^2$ which is the second-order moment of the random variable $|A_1|$ and previously defined as $P_{A_1}$.

2. The second term is the autocorrelation matrix of the complex white noise. Since the autocorrelation matrix of white noise is zero except for the main diagonal where the value is the variance $\sigma_e^2$, the second term can be written as $\sigma_e^2I$ where $I$ denotes the identity matrix.

3. The third term is equal to 0 since the signal and the noise vector are assumed uncorrelated in the model for the observed data.

4. The fourth term is equal to 0 for the same reason as the third term.

Thus, the final expression for the autocorrelation matrix is

$$R_{xx} = P_{A_1}a_1a_1^H + \sigma_e^2I.$$ (1.10)

If the autocorrelation matrix is written out

$$R_{xx} = \begin{bmatrix}
P_{A_1} + \sigma_e^2 & P_{A_1}e^{-j\omega_1} & P_{A_1}e^{-j2\omega_1} & \ldots & P_{A_1}e^{-j(M-1)\omega_1} \\
P_{A_1}e^{j\omega_1} & P_{A_1} + \sigma_e^2 & P_{A_1}e^{-j\omega_1} & \ldots & P_{A_1}e^{-j(M-2)\omega_1} \\
P_{A_1}e^{j2\omega_1} & P_{A_1}e^{j\omega_1} & P_{A_1} + \sigma_e^2 & \ldots & P_{A_1}e^{-j(M-3)\omega_1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P_{A_1}e^{j(M-1)\omega_1} & P_{A_1}e^{j(M-2)\omega_1} & P_{A_1}e^{j(M-3)\omega_1} & \ldots & P_{A_1} + \sigma_e^2
\end{bmatrix}$$ (1.11)

it is seen that the autocorrelation matrix is Hermitian, Toeplitz and additionally, it can be shown that it is positive semi-definite [Stoica and Moses, 1997, p. 6]. Therefore, the autocorrelation matrix has the following properties which are essential for the subspace-based method:

- It is possible to perform an eigenvalue decomposition [Lay, 2003, p. 452].
- The eigenvalues are real and non-negative [Lay, 2003, p. 452 and 461].
- The eigenvectors are orthogonal [Lay, 2003, p. 452].

The Toeplitz property shows that the autocorrelation function only depends on the time-shift. This is required for the observed signal in order to be stationary.

As the previous discussion indicates, the next step of the subspace-based method is to perform an eigenvalue decomposition of the autocorrelation matrix. It turns out that $a_1$ is an eigenvector of the autocorrelation matrix since

$$R_{xx}a_1 = (P_{A_1}a_1a_1^H + \sigma_e^2I)a_1 = P_{A_1}a_1a_1^Ha_1 + \sigma_e^2a_1$$

$$= P_{A_1}a_1M + \sigma_e^2a_1 = (P_{A_1}M + \sigma_e^2)a_1$$ (1.12)
where the second last equality follows since

\[ a_1^H a_1 = \begin{bmatrix} 1 & e^{-j\omega_1} & \cdots & e^{-j(M-1)\omega_1} \end{bmatrix} \begin{bmatrix} 1 \\ e^{j\omega_1} \\ \vdots \\ e^{j(M-1)\omega_1} \end{bmatrix} = 1 + 1 + \cdots + 1 = M. \]  

(1.13)

It is readily seen that equation (1.12) has the structure \(Av = \lambda v\), so \(a_1\) is indeed an eigenvector with the corresponding eigenvalue given by

\[ \lambda_1 = P_{A_1}M + \sigma_e^2. \]  

(1.14)

Since all eigenvectors are orthogonal, they satisfy \(v_i^H v_j = 0\) for \(i \neq j\). Thus, for \(v_i \neq v_1 = a_1, i = 2, 3, \ldots, M\)

\[ R_{xx}v_i = (P_{A_1}a_1 a_1^H + \sigma_e^2 I)v_i = P_{A_1}a_1 a_1^H v_1 + \sigma_e^2 v_i = \sigma_e^2 v_i. \]  

(1.15)

That is, the noise variance \(\sigma_e^2\) is an eigenvalue with multiplicity \(M - 1\).

To summarize the previous discussion it is possible to find the three unknown parameters \(\omega_k, P_{A_k}\) and \(\sigma_e^2\) that define the power density spectrum for one complex exponential in complex white noise by use of the following steps:

1. Form the autocorrelation matrix from \(M\) consecutive samples of the observed signal.
2. Compute the eigenvalues and eigenvectors of the autocorrelation matrix and order them by the eigenvalues in decreasing order.
3. Identify the variance or power \(\sigma_e^2\) of the complex white noise as the \(M - 1\) smallest eigenvalues that all have the same value, namely \(\sigma_e^2\).
4. Identify the largest eigenvalue \(\lambda_1\) and compute the power \(P_{A_1}\) of the complex exponential from the relation \(\lambda_1 = P_{A_1}M + \sigma_e^2\).
5. Find the frequency \(\omega_1\) of the complex exponential from the eigenvector \(v_1\) that correspond to the largest eigenvalue \(\lambda_1\). That is, solve one of the \(M\) equations given by \(v_1 = a_1\). In theory, they should all have the same solution, namely \(\omega_1\).

1.2.2 The subspace-based method for two complex exponentials

Before the general subspace-based approach for \(L\) complex exponentials is presented, the case for \(L = 2\) is presented. Of course, it has a lot in common with the case for \(L = 1\) that was presented in previous subsection, so the emphasis of this subsection will be on the differences.
Consider again the assumed model of the data in equation 1.1 and let $L = 2$. Equation 1.1 can now be rewritten into vector notation as

$$x = A_1 a_1 + A_2 a_2 + e$$

(1.16)

where $A_i$, $a_i$ and $e$ are defined in the same way as for the case where $L = 1$. Again the autocorrelation matrix of $x$ is computed

$$R_{xx} = E \{xx^H\} = E \{(A_1 a_1 + A_2 a_2 + e)(A_1 a_1 + A_2 a_2 + e)^H\}$$

(1.17)

$$= E \{(A_1 a_1 + A_2 a_2 + e)(A_1^H a_1^H + A_2^H a_2^H + e^H)\}$$

(1.18)

$$= E \{|A_1|^2 a_1 a_1^H\} + E \{|A_2|^2 a_2 a_2^H\} + E \{ee^H\} + E \{(A_1 a_1 + A_2 a_2)e^H\}$$

$$+ E \{e(A_1^H a_1^H + A_2^H a_2^H)\} + E \{A_1 A_2^H a_1 a_2^H\} + E \{A_2 A_1^H a_2 a_1^H\} \right) .$$

(1.19)

Whereas the first five terms are equal in structure to the four terms in equation 1.9 and therefore are reduced in the same way, the last two terms are new and represent the cross-correlation of the two complex exponentials. For $\phi_1 \neq \phi_2$ they are uncorrelated [Stoica and Moses, 1997, p. 140] for which reason the autocorrelation matrix reduces to

$$R_{xx} = P_{A_1} a_1 a_1^H + P_{A_2} a_2 a_2^H + \sigma_e^2 I \right) .$$

(1.20)

Now, since the sum of two Hermitian and positive semidefinite matrices is also Hermitian and positive semidefinite, the autocorrelation matrix for $L = 2$ has the same properties with respect to the eigenvalue decomposition as for the autocorrelation matrix with $L = 1$. That is, if $a_1$ and $a_2$ are linearly independent (as they are for $\omega_1 \neq \omega_2$), the autocorrelation matrix has exactly $M - 2$ eigenvectors which are orthogonal to $a_1$ and $a_2$, for which reason they correspond to the same eigenvalue, namely $\sigma_e^2$. The subspace spanned by these eigenvectors are often referred to as the noise subspace. The remaining two eigenvectors $v_1$ and $v_2$ which span the signal subspace, are in the subspace spanned by $a_1$ and $a_2$, but they are not in general equal to $a_1$ and $a_2$, and $a_1$ and $a_2$ are not in general orthogonal. Thus, the eigenvalues corresponding to the two eigenvectors $v_1$ and $v_2$ cannot be expressed as easily as for the case where $L = 1$.

Instead another approach is used. Note, that since the vectors in the signal subspace are orthogonal to the vectors in the noise subspace, the following equations are all valid

$$a_1^H v_3 = 0 , \quad a_2^H v_3 = 0 , \quad a_1^H v_4 = 0 , \quad a_2^H v_4 = 0 , \quad \cdots , \quad a_1^H v_M = 0 , \quad a_2^H v_M = 0 \right) .$$

(1.21)

which can be written in the more compact form as

$$\begin{bmatrix} a_1^H \\ a_2^H \end{bmatrix} \begin{bmatrix} v_3 & v_4 & \cdots & v_M \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} = 0 \right) .$$

(1.22)

Solving the equations yield the desired frequencies since, in each row, all the equations in theory have the same solution, namely $\omega_1$ and $\omega_2$, respectively.
To summarize the previous discussion it is possible to find the three unknown parameters $\omega_k$, $P_{A_k}$ and $\sigma_e^2$ that define the power density spectrum for two complex exponentials in complex white noise by use of the following steps:

1. Form the autocorrelation matrix from $M$ consecutive samples of the observed signal.

2. Compute the eigenvalues and eigenvectors of the autocorrelation matrix and order them by the eigenvalues in decreasing order.

3. Identify the variance or power $\sigma_e^2$ of the complex white noise as the $M-2$ smallest eigenvalues that all have the same value, namely $\sigma_e^2$.

4. Find the frequencies $\omega_1$ and $\omega_2$ of the complex exponentials by solving the equations in equation (1.22).

5. Find the power of the two complex exponentials by solving equation (1.20) for $P_{A_1}$ and $P_{A_2}$.

1.2.3 The subspace-based method for $L$ complex exponentials

From the preceding discussion a generalization of the subspace-based method can be derived. Consider again the assumed model of the data in equation (1.1) for $L$ complex exponentials. Equation (1.1) can now be rewritten into vector notation as

$$x = A_1a_1 + A_2a_2 + \cdots + A_La_L + e = \sum_{k=1}^{L} A_k a_k + e$$

$$= \begin{bmatrix} a_1 & a_2 & \cdots & a_L \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_L \end{bmatrix} + e = A \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_L \end{bmatrix} + e.$$  \hspace{1cm} (1.23)

The autocorrelation matrix of $x$ is computed by

$$R_{xx} = E \{xx^H\} = P_1a_1a_1^H + P_2a_2a_2^H + \cdots + P_La_La_L^H + \sigma_e^2I$$

$$= A \begin{bmatrix} P_{A_1} & 0 & \cdots & 0 \\ 0 & P_{A_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{A_L} \end{bmatrix} A^H + \sigma_e^2I = APA^H + \sigma_e^2I \hspace{1cm} (1.24)$$

by using the same arguments as for $L = 2$. Again, it is noted that the sum of Hermitian and positive semidefinite matrices is also Hermitian and positive semidefinite for which reason an eigenvalue decomposition of the autocorrelation matrix is possible and leads
to $M$ orthogonal eigenvectors. If the eigenvectors are sorted with respect to their corresponding eigenvalues with the largest eigenvalue first and the smallest last, the $L$ first eigenvectors span the signal subspace and are denoted

$$ S = \begin{bmatrix} s_1 & s_2 & \cdots & s_L \end{bmatrix} \quad (1.25) $$

and the $M - L$ last eigenvectors span the noise subspace and are denoted

$$ G = \begin{bmatrix} g_1 & g_2 & \cdots & g_{M-L} \end{bmatrix} . \quad (1.26) $$

Figure 1.2: Sketch of the magnitude of the sorted eigenvalues.

Figure 1.2 shows a sketch of the sorted eigenvalues. The $M - L$ smallest eigenvalues which corresponds to the eigenvalues that form the noise subspace, are all equal to the noise variance. The $L$ largest eigenvalues which corresponds to the eigenvectors in the signal subspace, are all larger than the noise variance. Thus in theory, the sorted eigenvalues reveal the number of complex exponentials $L$ in the signal.

As for the case with $L = 2$ the column vectors of the matrix $A$ span the signal subspace and is therefore orthogonal to the column vectors of the noise subspace $G$ (but not in general equal to the column vectors of the signal subspace $S$). Thus, the equation

$$ A^H G = \begin{bmatrix} a_1^H & a_2^H & \cdots & a_L^H \end{bmatrix} \begin{bmatrix} g_1 & g_2 & \cdots & g_{M-L} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = 0 . \quad (1.27) $$

can be used to compute the frequencies of the complex exponentials. Solving the equations yield the desired frequencies since, in each row, all the equations in theory have the same solution, namely $\omega_1$, $\omega_2$, $\cdots$, $\omega_{L-1}$ and $\omega_L$, respectively.

To summarize the previous discussion it is possible to find the three unknown parameters $\omega_k$, $P_{Ak}$ and $\sigma^2_e$ that define the power density spectrum for $L$ complex exponentials in complex white noise by use of the following steps:

1. Form the autocorrelation matrix from $M$ consecutive samples of the observed signal.
2. Compute the eigenvalues and eigenvectors of the autocorrelation matrix and order them by the eigenvalues in decreasing order.

3. Identify the variance or power $\sigma^2$ of the complex white noise as the $M-L$ smallest eigenvalues that all have the same value, namely $\sigma^2_e$.

4. Find the frequencies $\omega_1$ and $\omega_2$, $\omega_{L-1}$ and $\omega_L$ of the complex exponentials by solving the equations in equation 1.27.

5. Find the power of the $L$ complex exponentials by solving the equation $R_{xx} = APA^H + \sigma^2_e I$ for $P$.

1.3 The MUSIC method

The observed data does rarely fit the assumed model. For this reason the theoretical approach must be modified. Historically, several modifications have been proposed and one of them is the MUSIC method. This section will describe how the MUSIC method is used to estimate the frequencies of the power density spectrum. The estimation of the complex noise power and the power of the individual complex exponentials is not considered any further.

1.3.1 Estimation of the autocorrelation matrix

As previously described (see section 1.2.1) the subspace-based method is based on the eigenvalue decomposition of the $M \times M$ autocorrelation matrix which until now has been assumed known. In practice the autocorrelation matrix is unknown and is estimated from the observed data. Thus, the first step of the MUSIC method is to find an estimate of the autocorrelation matrix from the observed data.

Several approaches in order to estimate the autocorrelation matrix exist, but only one will be considered here: The standard unbiased estimate. The unbiased estimate is the most intuitive one and works like this: Take $N$ consecutive samples $\{x(1), x(2), \cdots, x(N)\}$ of the observed data and form $N - M + 1$ autocorrelation matrices of size $M \times M$ where each correlation matrix corresponds to a different extraction of $M$ consecutive samples $x$ of the $N$ consecutive samples. Finally, compute the average autocorrelation matrix by averaging each equivalent element of the $N - M + 1$ autocorrelation matrices. Stated in a mathematical way, the unbiased estimate of the autocorrelation matrix is [Therrien, 1992, p. 309]

$$\hat{R}_{xx} = \frac{1}{N - M + 1} \sum_{k=M}^{N} x[k]x[k]^H$$  \hspace{1cm} (1.28)

where $x[k] = [x[k]\ x[k-1]\ \cdots\ x[k-(M-1)]]^T$. Now, since

$$E\{\hat{R}_{xx}\} = \frac{1}{N - M + 1} \sum_{k=M}^{N} E\{x[k]x[k]^H\} = \frac{1}{N - M + 1} \sum_{k=M}^{N} R_{xx} = R_{xx}$$  \hspace{1cm} (1.29)
the estimate is clearly unbiased. It can also be shown that the variance of the estimate $\text{Var}(\hat{R}_{xx})$ approaches 0 for $N - M \to \infty$ which implies consistency of the estimate [Stark and Woods, 2002, p. 289].

Besides the estimation method the value of $N$ and $M$ are also of importance for the characteristics of the estimate. As implied by the consistency property the precision of the estimate increases with increasing $N - M$ for which reason $N$ should be chosen large and $M$ small. However, it can be shown that the accuracy of the MUSIC frequency estimates increases with increasing $M$ [Stoicha and Moses, 1997, p. 158] and a large $N$ entails poor time resolution of successive frequency estimates. Finally, large values of $N$ and especially $M$ leads to increased computational complexity. Thus, the optimum values of $N$ and $M$ is application dependent and should be chosen accordingly.

### 1.3.2 Estimation of the frequencies

Since the observed data rarely fits the assumed model, it is most likely that there do not exist $L$ frequencies which satisfy the equations in equation 1.27. For this reason the MUSIC method does not try to solve the equations but instead it minimizes the norm of the row equations. That is, for each row of equation 1.27 the method seeks the $\hat{\omega}_k$ that minimizes

$$J = \|a_k^H G\|_2 = a_k^H G G^H a_k$$  \hspace{1cm} (1.30)

where $\|\cdot\|_2$ denotes the Euclidean norm. If the observed data fits the assumed model, the minimum value of $J$ is equal to zero and additionally, it can be shown, that only the correct estimate of $\omega_k$ minimizes $J$ [Stoicha and Moses, 1997, p. 157].

The MUSIC method offers two different approaches to the minimization of $J$ [Stoicha and Moses, 1997, p. 158]:

1. **Spectral MUSIC**

   Substitute $a_k$ for $a = [1 \ e^{j\omega} \ \cdots \ e^{j(M-1)\omega}]^T$ in equation 1.30 and define the MUSIC pseudospectrum as

   $$\hat{P}_{MU}(e^{j\omega}) = \frac{1}{a_k^H G G^H a_k}, \quad \text{for } \omega \in [-\pi, \pi].$$  \hspace{1cm} (1.31)

   For each $\omega = \omega_k$ the function $\hat{P}_{MU}(e^{j\omega})$, like the power spectral density function, exhibits sharp peaks (theoretically infinite) and thus, it can be used to determine the frequencies that minimizes $J$. The term pseudospectrum refers to the fact that the plot can be used to locate the desired frequencies, but contains no information of the signal or noise power in contrast to the true power spectral density function.

2. **Root MUSIC**

   Substitute $a_k$ for $z = [1 \ z^1 \ \cdots \ z^{M-1}]^T$ in equation 1.30 and find the roots of the equation

   $$z^H G G^H z = 0.$$  \hspace{1cm} (1.32)
Identify the angle of roots that lie closest to the unit circle in the $z$-plane as the desired frequencies.

Bibliography


Joint spectral and order estimation using the HMUSIC method

There are several methods available for parametric spectral estimation and one of these is the MUSIC (MUltiple SIgnal Characterization) method. The MUSIC method is a subspace-based estimation method which estimate is found as the frequencies minimizing  

\[ J = \| A^H G \|_F^2 , \]  

with \( A \) and \( G \) spanning the signal and noise subspace, respectively, and \( \| \cdot \|_F \) denoting the Frobenius norm. For a definition of these subspaces see worksheet 1. It should be noticed, that in this worksheet, it is assumed that the signal to be analyzed has perfect harmonicity (i.e. equally spaced harmonics), for which reason the signal model \( A \) can be written as  

\[ A = \begin{bmatrix} a(\omega_0) & \cdots & a(\omega_0L) \end{bmatrix} . \]  

One problem with the MUSIC criterion is that it can only be used for estimating the fundamental frequency of harmonic signals with known model order \( L \) (i.e. known number of harmonics). Therefore, it is desired to extend the MUSIC estimation criterion from equation 2.1 for jointly estimating both the fundamental frequency \( \omega_0 \) and the model order \( L \). The topic of this worksheet will therefore be the extension of the MUSIC estimation criterion, and it is termed the HMUSIC (Harmonic MUSIC) estimation criterion.

2.1 The HMUSIC method

The MUSIC estimation criterion from equation 2.1 is dependent on both the model order \( L \) and the size \( M \) of the estimated correlation matrix. Therefore the first step, when
2.1. The HMUSIC method

deriving a new criterion for jointly estimation of $\omega_0$ and $L$, is to scale the cost function given by the MUSIC estimation criterion. To achieve this the Cauchy-Schwarz inequality is used which leads to \cite{Lay2003, p. 432}

$$\|A^H G\|_F \leq \|A^H\|_F \|G\|_F .$$

(2.3)

The signal model $A$ is given by \cite{StoicaMoses2005, p. 156}

$$A = \begin{bmatrix} a(\omega_0) & \cdots & a(\omega_0L) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ e^{i\omega_0(M-1)} & \cdots & e^{i\omega_0L(M-1)} \end{bmatrix} ,$$

(2.4)

and the Frobenius norm of a matrix $X_{m \times n}$ is defined as \cite{StoicaMoses2005, p. 355}

$$\|X\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |x_{ij}|^2} .$$

(2.5)

Since each element in $A$ have a length equal to one it is readily seen that the Frobenius norm of $A^H$ is

$$\|A^H\|_F = \sqrt{ML} .$$

(2.6)

The next task is to determine the Frobenius norm of the noise subspace $G$. The noise subspace consists of the orthonormal eigenvectors corresponding to the $M - L$ least significant eigenvalues of the estimated autocorrelation matrix (see worksheet 1)

$$G = \begin{bmatrix} g_1 & \cdots & g_{M-L} \end{bmatrix} .$$

(2.7)

Because the columns of $G$ are normal vectors (i.e. they have a length of one), the Frobenius norm of $G$ is

$$\|G\|_F = \sqrt{g_1^H g_1 + \cdots + g_{M-L}^H g_{M-L}} = \sqrt{\|g_1\|^2 + \cdots + \|g_{M-L}\|^2} = \sqrt{M-L} ,$$

(2.8)

By inserting the results of equation 2.6 and 2.8 into equation 2.3 it is obtained that

$$\frac{\|A^H G\|_F}{\sqrt{LM(M-L)}} \leq 1 .$$

(2.9)
Next equation 2.9 is rewritten into a new estimation criterion which finds the fundamental frequency as the frequency that maximizes

$$\hat{\omega}_0 = \arg \max_{\omega_0 \in \Omega} \frac{LM(M - L)}{||A^H G||_F^2}.$$  \hspace{1cm} (2.10)

where $\Omega$ is the set of candidate fundamental frequencies to maximize over. This new criterion is, compared to the MUSIC estimation criterion, scaled by the factor $ML(M - L)$ found from the Cauchy-Schwarz inequality in equation 2.3, and has thereby a noise floor which is independent of $M$ and $L$ (i.e. the size of the estimated covariance matrix and the model order) [Christensen, Jakobsson, and Jensen, 2007, p. 1636]. It should now be possible to reformulate the estimation criterion so it jointly estimates the model order and the fundamental frequency.

It is known, that the orthogonality described by equation 1.27 worksheet 1 only holds when the model order $L$ is chosen so that the range of $A$ is equal to the range of the signal subspace $S$. That is, equation 2.10 will only be maximum when both the true fundamental frequency and model order are inserted. The estimation criterion is therefore reformulated into

$$(\hat{\omega}_0, \hat{L}) = \arg \max_{\omega_0 \in \Omega} \max_{L \in \mathbb{L}} \frac{LM(M - L)}{||A^H G||_F^2}.$$ \hspace{1cm} (2.11)

As mentioned in the beginning of the worksheet this estimation criterion is termed the HMUSIC estimation criterion.

**Bibliography**


Adaptive Subspace tracking with the PAST and PASTd algorithms

This worksheet introduces the PAST- and PASTd-algorithms that can be used for recursive and adaptive subspace tracking. Subspace tracking is used to track a basis of the eigenspace of the autocorrelation matrix by means of an adaptive algorithm. This is of great interest with relation to the MUSIC method considered in worksheet 1 since an eigenvalue decomposition is an essential part of it. This relation will be the topic of the first section of this worksheet. Next, the weighted recursive least squares adaptive algorithm is introduced since it constitutes the foundation of the PAST-algorithm. The last section of this worksheet describes the PASTd-algorithm which is derived from the PAST-algorithm.

### 3.1 Relationship between PAST and MUSIC

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fetch input data vector</td>
</tr>
<tr>
<td>2</td>
<td>Form estimate of autocorrelation matrix</td>
</tr>
<tr>
<td>3</td>
<td>Perform EVD of autocorrelation matrix</td>
</tr>
<tr>
<td>4</td>
<td>Form noise subspace</td>
</tr>
<tr>
<td>5</td>
<td>Find frequency estimate from either spectral or root MUSIC</td>
</tr>
</tbody>
</table>

**Figure 3.1:** The different stages of the MUSIC algorithm.

The MUSIC method was considered in worksheet 1 and the algorithm is depicted in figure 3.1. In the figure, the MUSIC method is divided into five steps. In the first step \( N \) samples from the observed signal are collected and used to compute an estimate of the \( M \times M \) autocorrelation matrix in step two. The third step is to perform an eigenvalue decomposition (EVD) of the autocorrelation matrix. From this, the noise subspace is
found in step four as the $M - L$ least dominant eigenvectors where $L$ is the number of complex exponentials within the observed signal. The final step is to estimate the frequencies of the complex exponentials assumed to constitute the observed signal together with additive white complex Gaussian noise.

The computational complexity of the algorithm in figure 3.1 is high. Especially the EVD is very time consuming and is not suited for real-time implementation [Yang, 1995, p. 1]. For this reason several alternative approaches have been published which seek to form the signal subspace (or equivalently the noise subspace) directly from the observed data without computing neither the estimated autocorrelation matrix nor the EVD. One of these algorithms is the projection approximation subspace tracking (PAST) algorithm which was proposed by Bin Yang in 1995 [Yang, 1995]. The PAST algorithm is based upon the weighted recursive least squares (RLS) which is a well-known adaptive signal processing algorithm. Therefore, before describing the PAST algorithm in greater detail, an review of the weighted RLS is given.

### 3.2 Review of Weighted Recursive Least Squares

Before describing the weighted recursive least squares (RLS), a short introduction to the optimal Wiener filtering problem is given.

Figure 3.2 shows the block diagram of the optimum Wiener filter problem. The different signals in the figure are

- $x[n]$: The observed signal.
- $w$: The impulse response of the linear filter. For an FIR filter, the impulse response equals the filter coefficients. In this worksheet, only FIR filters of length $M$ are considered.
- $d[n]$: Desired response.
These signals depend on each other in some known or unknown statistical way.

The task of the optimum Wiener filter can be formulated in the following way:

*Determine the impulse response $w$ such that the cost-function $J(w) = f(e[n])$ is minimized.*

The cost-function depends on the estimation error $e[n]$. Two commonly used cost-functions are the least mean squares error [Haykin, 2002, p. 97]

$$J(w) = E\{e[n]e^*[n]\} = E\{|e[n]|^2\}$$

(3.1)

where $E\{\cdot\}$ denotes the statistical expectation, and the least squares error [Haykin, 2002, p. 387]

$$J(w) = \sum_{i=i_1}^{i_2} e[i]e^*[i] = \sum_{i=i_1}^{i_2} |e[i]|^2$$

(3.2)

where the index limits $i_1$ and $i_2$ define the region over which the minimization occurs. These cost-functions are attractable since they are quadratic and thus have exactly one minimum which can be found from iterative methods.

**Wiener-Hopf equations**

If the cost-function is chosen to be the least mean squares error, and the observed signal $x[n]$ and the desired signal $d[n]$ are wide sense stationary (WSS), the filter coefficients $w$ can be found as the solution to the Wiener-Hopf equations given by [Haykin, 2002, p. 104]

$$R_{xx}w = r_{xd}$$

(3.3)

where $R_{xx}$ is the autocorrelation matrix of the observed signal

$$R_{xx} = E\{x[n]x^H[n]\}, \quad x[n] = \{x[n], x[n-1], \cdots, x[n-(M-1)]\}^T,$$

(3.4)

$r_{xd}$ is the cross-correlation vector between the desired signal $d[n]$ and the observed signal $x[n]$

$$r_{xd} = E\{x[n]d^*[n]\},$$

(3.5)

and the Wiener filter coefficients $w$ is given by

$$w = \{w[0], w[1], \cdots, w[M-1]\}^T.$$

(3.6)

Note that since the observed signal $x[n]$ and the desired signal $d[n]$ are WSS, the autocorrelation matrix $R_{xx}$ and the cross-correlation vector $r_{xd}$ do not change with time and thus, the Wiener filter coefficients $w$ do not change with time.
Often, the statistics of the observed and desired signals are unknown and change with time. Therefore, the statistics must be estimated from the observed and desired signals and new filter coefficients must be calculated for every new observation. This is known as adaptive filtering, and two well-known adaptive methods are the least mean squares (LMS) and the least squares (LS) algorithms that minimize the least mean squares error cost function and the least squares error cost function, respectively. One of the LS-algorithms is the weighted recursive least squares (weighted RLS) which will be considered next.

### 3.2.1 Weighted Recursive Least Squares

The weighted LS cost function is a slightly modified version of the LS cost function given in equation 3.2. A weighting factor \( \beta[n, i] \) is introduced into the equation as [Haykin, 2002, p. 436]

\[
J(w[n]) = \sum_{i=0}^{n} \beta[n, i] e[i] e^*[i], \quad 0 < \beta[n, i] \leq 1 . \tag{3.7}
\]

The weighting factor is often referred to as the forgetting factor since it suppresses the influence of old data and hence enables the algorithm to track statistical variations. A commonly used forgetting factor is the exponentially weighting factor given by [Haykin, 2002, p. 437]

\[
\beta[n, i] = \lambda^{n-i} . \tag{3.8}
\]

Since the estimation error from figure 3.2 is given by \( e[n] = d[n] - w^H[n]x[n] \), the sequence of filter coefficients \( w[n] \), that minimizes equation 3.7, can be found by differentiating the cost function with respect to \( w^* \)

\[
0 = \frac{\partial}{\partial w^*} J(w[n]) = \frac{\partial}{\partial w^*} \sum_{i=0}^{n} \beta[n, i] e[i] e^*[i] \tag{3.9}
\]

\[
= \frac{\partial}{\partial w^*} \sum_{i=0}^{n} \beta[n, i](d[i] - w^H[n]x[i])(d[i] - w^H[n]x[i])^* \tag{3.10}
\]

\[
= - \frac{\partial}{\partial w^*} \sum_{i=0}^{n} \beta[n, i] w^H[n]x[i](d^*[i] - x^H[i]w[n]) \tag{3.11}
\]

\[
= \sum_{i=0}^{n} \beta[n, i] x[i](d^*[i] - x^H[i]w[n]) . \tag{3.12}
\]

This can be rewritten into

\[
\sum_{i=0}^{n} \beta[n, i] x[i]d^*[i] = \sum_{i=0}^{n} \beta[n, i] x[i]x^H[i]w[n] = \left[ \sum_{i=0}^{n} \beta[n, i] x[i]x^H[i] \right] w[n] . \tag{3.13}
\]

Writing these equations in the same form as the Wiener-Hopf equations in equation 3.3 yields

\[
\hat{R}_{xx}[n]w[n] = \hat{r}_{xd}[n] . \tag{3.14}
\]
which shows that
\[ \hat{R}_{xx}[n] = \sum_{i=0}^{n} \beta[n, i] x[i] x^H[i] \] (3.15)
and
\[ \hat{r}_{xd}[n] = \sum_{i=0}^{n} \beta[n, i] x[i] d^*[i] \] (3.16)
are build-in estimates of the autocorrelation matrix and the cross-correlation vector of the weighted least squares method.

Comparing the Wiener-Hopf equations in equation 3.3 and the weighting LS counterpart in equation 3.14 reveals two main differences. First of all, no prior statistical knowledge of the observed or desired signals are required in the LS approach. Secondly, the LS approach is able to track statistical variations since the estimates of the autocorrelation matrix and the cross-correlation vector are time-dependent.

The LS solution to the optimal Wiener filtering problem in equation 3.14 is not suitable for direct implementation. The reason for this is the computation complexity and memory consumption which are high due to the inversion of the estimated autocorrelation matrix and the requirement to store all old input samples. Therefore a recursive formulation of the LS solution has been developed which is known as the recursive least squares (RLS) algorithm.

**Recursive Implementation of the Exponentially Weighted LS**

In the recursive formulation the new filter coefficients \( w[n] \) is computed by adding a correction \( \Delta w[n] \) to the previous filter coefficients \( w[n-1] \) which is written as
\[ w[n] = w[n-1] + \Delta w[n] . \] (3.17)
3 Adaptive Subspace tracking with the PAST and PASTd algorithms

This is depicted in figure 3.3. Since $R_{xx}$ is non-singular, the filter coefficients $w[n]$ at time $n$ can be calculated from equation 3.14 by

$$w[n] = \hat{R}_{xx}^{-1}n\hat{r}_{xd}[n].$$

(3.18)

Thus, the filter coefficients can be computed recursively if the cross-correlation vector and the inverse of the autocorrelation matrix can be computed recursively. A recursive formulation of the exponentially weighted cross-correlation vector can be readily expressed from equation 3.8 and equation 3.16 as

$$\hat{r}_{xd}[n] = \lambda^{n-n}x[n]d^*[n] + \lambda \sum_{i=0}^{n-1} \lambda^{n-1-i}x[i]d^*[i] = x[n]d^*[n] + \lambda \hat{r}_{xd}[n-1].$$

(3.19)

Similarly, the exponentially weighted autocorrelation matrix can be formulated in a recursive way from equation 3.15 as

$$\hat{R}_{xx}[n] = \lambda^{n-n}x[n]x^H[n] + \lambda \sum_{i=0}^{n-1} \lambda^{n-1-i}x[i]x^H[i] = x[n]x^H[n] + \lambda \hat{R}_{xx}[n-1]$$

(3.20)

Now, in order to find the inverse of the autocorrelation matrix the matrix inversion lemma (also known as Woodbury’s Identity) is used which yields [Hayes, 1996, p. 543]

$$\hat{R}_{xx}^{-1}[n] = P[n] = \lambda^{-1}P[n-1] - \frac{\lambda^{-2}P[n-1]x[n]x^H[n]P[n-1]}{1 + \lambda^{-1}x^H[n]P[n-1]x[n]}$$

(3.21)

where $\hat{R}_{xx}^{-1}[n] = P[n]$ is introduced in order to simplify the notation.

Inserting equation 3.21 and equation 3.19 into equation 3.18 yields the recursive formulation [Hayes, 1996, p. 544] of the exponential weighted LS as

$$w[n] = w[n-1] + g[n]e_p^*[n]$$

(3.22)

where the gain vector $g[n]$ is

$$g[n] = \frac{\lambda^{-1}P[n-1]x[n]}{1 + \lambda^{-1}x^H[n]P[n-1]x[n]}$$

(3.23)

and the a priori estimation error $e_p[n]$ is

$$e_p[n] = d[n] - w^H[n-1]x[n].$$

(3.24)

The a priori estimation error is the error that would occur if the filter coefficients were not updated. After the update the error is $e[n] = d[n] - w^H[n]x[n]$.

The exponentially weighted recursive least squares algorithm is shown in algorithm □
3.3 The PAST Algorithm

Algorithm 1: The exponentially weighted recursive least squares

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Exponential weighting factor $\lambda$.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Filter order $M$.</td>
</tr>
<tr>
<td>Initialize</td>
<td>The filter coefficients $w[0]$.</td>
</tr>
<tr>
<td></td>
<td>The inverted correlation matrix $P[0]$.</td>
</tr>
</tbody>
</table>

for $n = 1, 2, 3, \cdots$ do

$h[n] = P[n-1]x[n]$;

$g[n] = \frac{h[n]}{\lambda + x_H[n]h[n]}$;

e_p[n] = d[n] - w_H[n-1]x[n];

$w[n] = w[n-1] + g[n]e_p^*[n]$;

$P[n] = \lambda^{-1}(P[n-1] - g[n]h_H[n])$
end

3.3 The PAST Algorithm

Equipped with the exponentially weighted RLS the projection approximation subspace tracking (PAST) algorithm can now be presented.

First, define the cost-function

$$J(W[n]) = \sum_{i=1}^{n} \lambda^{n-i}\|x[i] - W[n]W_H[n]x[i]\|^2$$ (3.25)

where $W[n]$ is an $M \times L$ matrix and $x[i]$ is an $M \times 1$ vector. It can be shown that the column vectors of $W[n]$ are orthonormal and span the signal subspace of the autocorrelation matrix of the signal $x[n]$ if $J(W[n])$ attains the global minimum [Yang, 1995, p. 3]. Additionally, it can be shown that the cost-function has no other local minima, only saddle points, and thus, global convergence can be guaranteed via iterative methods with some nursing [Yang, 1995, p. 3]. The nursing must ensure that the iterative algorithms do not terminate on the saddle points.

To summarize the discussion above, the cost-function in equation (3.25) can be used in a unconstrained minimization problem to compute a basis of the signal and noise subspaces in the MUSIC method instead of computing the EVD. That is, the second and third step of figure 3.1 can be replaced by the unconstrained minimization problem which is more efficient from a computational complexity point of view.

One of the iterative methods, that has been proposed in order to minimize the cost function in equation (3.25) is the projection approximation subspace tracking (PAST) algorithm. In the PAST algorithm the cost-function in equation (3.25) is rewritten into the weighted LS cost-function in equation (3.7) by making the approximation [Yang, 1995].
It should be noted that the approximation is only valid for stationary or nearly stationary signals.

If the approximation is denoted $y[i] = W^H[i-1]x[i]$ the cost-function in equation 3.25 can be rewritten as

$$J(W[n]) \approx J_{pa}(W[n]) = \sum_{i=1}^{n} \lambda^{n-i}||x[i] - W[n]y[i]||^2$$

(3.27)

which has the same form as the weighted LS cost-function in equation 3.7. Therefore, the exponentially weighted RLS can be applied to minimize $J_{pa}(W[n])$. Algorithm 2 summarizes the PAST algorithm.

---

**Algorithm 2: Projection Approximation Subspace Tracking (PAST)**

**Inputs**
- Exponential weighting factor $\lambda$.
- Filter order $M$.

**Initialize**
- The signal subspace $W[0]$.
- The inverted correlation matrix $P[0]$.

**for** $n = 1, 2, 3, \ldots$ **do**

- $h[n] = P[n-1]y[n]$;
- $g[n] = \lambda + y^H[n]h[n]$;
- $P[n] = \lambda^{-1}(P[n-1] - g[n]h^H[n])$;

**end**

The initial values of $W[0]$ and $P[0]$ can be set to the $L$ leading unit vectors and the $L \times L$ identity matrix, respectively [Yang, 1995, p. 5].

### 3.4 The PASTd algorithm

The PAST algorithm can be modified so it computes estimates of the eigenvectors sequentially. The modified PAST algorithm is known as the PASTd algorithm where the 'd' refers to the use of the deflation technique within the algorithm. The algorithm consists of two basic steps [Yang, 1995, p. 5]

1. Find the estimate of the most dominant eigenvector by using the PAST algorithm for the one vector case.
3.5. Examples of the MUSIC method using the PAST and PASTd algorithms

2. Remove the contribution of the most dominant eigenvector from the observed signal $x[n]$ by using the deflation technique.

These two steps are repeated until the $L$ most dominant eigenvectors, that span the signal subspace, are found.

The deflation technique, which is used to remove the contribution of the most dominant eigenvector, is based on the Karhunen-Loève expansion of the observed signal, which is [Haykin, 2002, p. 822]

$$x[n] = \sum_{i=1}^{M} (w_i^H[n]x[n])w_i[n] \quad (3.28)$$

where $w_1, w_2, \ldots, w_M$ are the eigenvectors associated with the $M$ eigenvalues of the autocorrelation matrix $R_{xx}$ of $x[n]$. If an eigenvector $w_1$ is known, its contribution to $x_1[n] = x[n]$ can thus be removed by applying

$$x_2[n] = x_1[n] - (w_1^H[n]x_1[n])w_1[n]. \quad (3.29)$$

Algorithm 3 summarizes the PASTd algorithm. The algorithm is basically the one vector PAST algorithm with some minor modifications. The inner for-loop of the PASTd algorithm contains the one vector PAST algorithm (the first four lines) and the deflation step (the last line). The scalar $d_i[n]$ is not used in the PAST algorithm, but it is introduced in the PASTd algorithm for two reasons. First of all, it reduces the computational complexity of the algorithm since neither $h[n]$ nor $g[n]$ from the PAST algorithm have to be computed. Secondly, it can be show that $d_i[n]$ is an estimate of the $i$th eigenvalue associated with the estimate of the $i$th eigenvector $w_i[n]$ [Yang, 1995, p. 5].

In order to lower the computational complexity, the deflation step in the last line is only an approximation to the outlined deflation step in equation 3.29. The approximation is

$$(w_i^H[n]x_i[n])w_i[n] \approx g_i[n]w_i[n] = (w_i^H[n-1]x_i[n])w_i[n] \quad (3.30)$$

and is justified by the fact that $w_i^H[n] \approx w_i^H[n-1]$ when the algorithm is in a neighborhood of the solution.

3.5 Examples of the MUSIC method using the PAST and PASTd algorithms

The last section of this worksheet gives some examples of the applicability of the PAST and PASTd algorithms. Based on the previous discussion, figure 3.4 shows how the original MUSIC method from figure 3.1, which uses an EVD, can be modified to use either
Algorithm 3: Projection Approximation Subspace Tracking with Deflation (PASTd)

**Inputs**
- Exponential weighting factor $\lambda$.
- Filter order $M$.

**Initialize**: The signal subspace $W[0]$.
- The $L$ most dominant eigenvalues $d[0]$.

for $n = 1, 2, 3, \cdots$ do

1. $x_1[n] = x[n]$;
2. for $i = 1, 2, 3, \cdots, L$ do
   3. $y_i[n] = w_i^H[n-1]x_i[n]$;
   4. $d_i[n] = \lambda d_i[n-1] + |y_i[n]|^2$;
   5. $e_i[n] = x_i[n] - w_i[n-1]y_i[n]$;
   6. $w_i[n] = w_i[n-1] + e_i[n]y_i^*[n]$;
   7. $x_{i+1}[n] = x_i[n] - y_i[n]w_i[n]$;
3. end
4. end

Table 3.1: The different parameters used in the two examples. The signal to noise ratio is the only parameter that is different in the two examples.
In this section, the frequency estimates obtained by using the PAST and PASTd algorithms in the MUSIC method are compared for two different signal to noise ratios (SNR) with the parameters given in table 3.1. The observed signal is defined to contain a fundamental frequency of \( f(t) \) and its first three harmonics are corrupted by additive white complex Gaussian noise, i.e.

\[
x[n] = e^{j2\pi f[n]n} + 0.8e^{j4\pi f[n]n} + 0.6e^{j6\pi f[n]n} + 0.4e^{j8\pi f[n]n} + e[n] \quad (3.31)
\]

where \( f[n] \) is a sampled version of \( f(t) \) from table 3.1 and \( e[n] \) is the noise. The amplitude of the noise is set according to the desired SNR. That is 0 for an SNR of \( \infty \) dB and 1.47 for an SNR of 0 dB.

Figure 3.5 and figure 3.6 shows for different curves for a SNR of \( \infty \) dB and 0 dB, respectively. The four curves show

- the orthogonality error measured by the quantity \( \| I - WHW \|_F^2 \) where \( F \) denoted the Frobenius norm,
- the value of the cost function given by \( \| x - Wy \|_2^2 \) (see equation 3.27) for the PAST and PASTd algorithms,
- the minimum of the music pseudo-spectrum given by \( \sum_{i=1}^{n} \lambda^{n-i} \| A^H G \|_F^2 \) which in theory should be equal to zero for a correct estimate,
• the frequency estimate of the PAST and PASTd algorithm compared to the true frequency of the observed signal.

In the noiseless case (SNR of $\infty$ dB) shown in figure 3.5 a general disadvantage of the PASTd algorithm as compared to the PAST algorithm is illustrated. The estimated signal subspace of the PAST algorithm is closer to being orthogonal than the signal subspace of the PASTd algorithm. This is also true for the case of noise (see figure 3.6) and is generally a problem [Yang, 1995, p. 5].

Despite that the signal subspace obtained by the PASTd algorithm is farther away from orthogonality than the signal subspace of the PAST algorithm, the tracking characteristics (at least in these examples) seems to be better for the PASTd algorithm. Both for the noiseless and the noisy case, the PASTd algorithm tracks the sudden frequency jump faster than the PAST algorithm. However, the PASTd algorithm also suffers from fluctuations in the frequency estimate around the true frequency value.

Bibliography

Figure 3.6: Different curves for the PAST and PASTd estimation for a SNR of 0 dB.


Optimized Fast Implementation of the MUSIC Algorithm

One of the first steps in the MUSIC algorithm is to find the eigenvectors spanning the noise subspace, or an arbitrary orthonormal basis of it, which is a very time consuming part of the algorithm. Instead of using the classic approach for this involving an EVD, it can be solved by using a subspace tracking algorithm. A few examples of such are the PAST and the FDPM algorithms. The subspace trackers can be grouped into high, medium and low complexity algorithms. Fast subspace trackers (low complexity) find an arbitrary basis of the signal subspace instead of the noise subspace. Therefore, the original MUSIC algorithm has to be reformulated. Besides that, using fast subspace trackers would not in general be usable when the model order is unknown.

Another time consuming part of the MUSIC algorithm is the evaluation of the cost function. In this worksheet, a fast implementation will be presented which takes into account that many fast subspace trackers find a basis of the signal subspace instead of the noise subspace. First, a fast implementation will be presented for MUSIC for a known model order which is extended to the case with an unknown model order. In most cases the proposed implementations will be much faster than the efficient implementation described in [Christensen et al., 2007] which is also shown.

To make the evaluation of the cost function even faster it is possible to exploit the local convexity of the cost function. In this way a gradient method can be introduced when the previous frequency estimate is ensured to be close to true frequency estimate. One way of doing this is by looking at the value of the cost function. In the subsequent worksheet the three mentioned issues are considered.
4.1 Fast Subspace Tracking for a Known and an Unknown Model order

If the model order is known, it is straightforward to use a fast subspace tracker instead of the computationally inefficient EVD since only an arbitrary orthonormal basis of the signal subspace is required for finding the frequency estimate. On the other hand, when the model order is unknown the fast subspace trackers cannot be directly applied. The reason for this is, that the fast subspace trackers only track an arbitrary orthonormal basis $W$ of the signal subspace and not the actual eigenvectors spanning the signal subspace $S = WQ$ where $Q$ is an arbitrary unitary matrix. Since

$$SS^H = WQQ^H W^H = WW^H$$ (4.1)

this is not a problem when the model order is known because the projection matrix is uniquely determined. For unknown model order, however, it is a problem, since projection matrices of subsets of $S$ cannot be recovered from subsets of $W$ where

$$S = [S_1 \ S_2], \ W = [W_1 \ W_2].$$ (4.2)

To show this, the expressions from equation (4.2) is inserted into equation (4.1) which leads to

$$S_1 S_1^H + S_2 S_2^H = W_1 W_1^H + W_2 W_2^H$$ (4.3)

from which $S_1 S_1^H$ cannot be uniquely determined since only $W$ is known. To get around this issue a possible solution is now mentioned. As stated in the preceding section it is possible to use the fast subspace trackers when the model order is known. Therefore, if the model order somehow could be estimated it may still be possible to use the fast subspace trackers. To do this, an EVD can be used and the frequency estimate can be found using the MUSIC method for an unknown model order until the value of the cost function gets sufficiently small. When this is the case, it can be assumed that the frequency and model order estimates are correct and as long as the value of the cost function stays sufficiently small (under a threshold $\alpha$) it can be assumed that the true frequency and model order do not change. Then, it is possible to use the MUSIC method for a known model order, and thereby use a fast subspace tracker. In figure 4.1 the described solution is depicted by means of a flowchart diagram.

4.2 Fast Implementation of the Calculation of the MUSIC Cost Functions

The cost function of the MUSIC estimation criterion is

$$J(\omega) = ||A^H G||^2_F = \text{tr}\{A^H G G^H A\}$$ (4.4)
4 Optimized Fast Implementation of the MUSIC Algorithm

Figure 4.1: Flowchart diagram of the MUSIC estimator implemented with fast subspace tracking.

where $A$ and $G$ are $M \times L$ matrices, and $A$ is given by

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{j\omega} & e^{j2\omega} & \cdots & e^{jL\omega} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(M-1)\omega} & e^{j(M-1)2\omega} & \cdots & e^{j(M-1)L\omega} \end{bmatrix} . \quad (4.5)$$

and $G$ spans the noise subspace.

The MUSIC cost function is non-convex which means that it has several minimums. Thus, iterative methods cannot in general be applied to find the global minimum. Instead another approach is often used which in worksheet 1 is referred to as spectral MUSIC [Stoica and Moses, 2005, p. 158]. It involves two steps: First, $J(\omega)$ must be evaluated for a set of candidate frequencies denoted by $\Omega$ and second, the frequency, at which $J(\omega)$ attains its minimum among the candidate frequencies, must be found. The first step is the most demanding one from a computational point view, and in this worksheet a fast implementation of it is provided.

4.2.1 Fast implementation for known model order

In [Christensen et al., 2007] a fast implementation of equation (4.4) is proposed using a series of FFTs. The approach requires the noise subspace and not the signal subspace to be known which is undesirable from computational complexity point of view. This is due to the fact that a basis of the noise subspace typically has a greater rank than a basis of the signal subspace and that most fast subspace trackers, like PAST, track the signal subspace [Doukopoulos and Moustakides, 2007]. It is therefore desirable to overcome this limitation.

Therefore, assume that the projection matrix of the of the noise subspace is known and define it as

$$M = GG^H = I - WW^H \quad (4.6)$$
4.2. Fast Implementation of the Calculation of the MUSIC Cost Functions

where $W$ is an arbitrary orthonormal basis of the signal subspace. Now, it can readily be shown that

$$J(\omega) = \text{tr}\{A^H M A\} = \sum_{i=1}^{L} a_i^H M a_i = \sum_{i=1}^{L} J_i(\omega)$$  \hspace{1cm} (4.7)$$

where $a_i$ is the $i$th column vector of the matrix $A$ and $L$ is the model order.

The partial cost function $J_i(\omega) = a_i^H M a_i$ can be written out as

$$J_i(\omega) = \left[ 1 \quad e^{-j\omega i} \quad e^{-j2\omega i} \quad \cdots \quad e^{-j(M-1)\omega i} \right] \left[ \begin{array}{cccc} m_{11} & m_{12} & \cdots & m_{1M} \\ m_{21} & m_{22} & \cdots & m_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ m_{M1} & m_{M2} & \cdots & m_{MM} \end{array} \right] \left[ \begin{array}{c} 1 \\ e^{j\omega i} \\ e^{j2\omega i} \\ \cdots \\ e^{j(M-1)\omega i} \end{array} \right]$$

$$= m_{1M} e^{j(M-1)\omega i} + (m_{1(M-1)} + m_{2M}) e^{j(M-2)\omega i} + \cdots$$
$$+ (m_{12} + m_{23} + \cdots + m_{(M-1)M}) e^{j(\omega i)} + (m_{11} + m_{22} + \cdots + m_{MM})$$
$$+ (m_{21} + m_{32} + \cdots + m_{M(M-1)}) e^{-j\omega i} + \cdots$$
$$+ (m_{(M-1)1} + m_{M2}) e^{-j(M-2)\omega i} + m_{MM} e^{-j(M-1)\omega i}$$

$$= \sum_{n=-(M-1)}^{M-1} c[n] e^{jn\omega i}. \hspace{1cm} (4.8)$$

where $c[n]$ is the sum of the elements in the $n$th diagonal of the matrix $M$.

Now, since $c[-n] = c^*[n]$ by the Hermitian property of the matrix $M$, the last equation
can be rewritten into

\[
J_i(\omega) = \sum_{n=-(M-1)}^{M-1} c[n]e^{jn\omega i} \quad (4.14)
\]

\[
= c[0] + \sum_{n=1}^{M-1} c[n]e^{jn\omega i} + \sum_{n=-(M-1)}^{-1} c[n]e^{jn\omega i} \quad (4.15)
\]

\[
= c[0] + \left[ \sum_{n=1}^{M-1} c[n]e^{jn\omega i} \right]^* + \sum_{n=1}^{M-1} c[-n]e^{-jn\omega i} \quad (4.16)
\]

\[
= c[0] + \left[ \sum_{n=1}^{M-1} c[n]e^{-jn\omega i} \right]^* + \sum_{n=1}^{M-1} c[n]e^{-jn\omega i} \quad (4.17)
\]

\[
= c[0] + 2\text{Re} \left[ \sum_{n=1}^{M-1} c[n]e^{-jn\omega i} \right] \quad (4.18)
\]

\[
= 2\text{Re} \left[ \sum_{n=0}^{N-1} g[n]e^{-jn\omega i} \right] = 2\text{Re} \left[ G(e^{j\omega i}) \right] \quad (4.19)
\]

where \( g[n] \) is given by

\[
g[n] = \begin{cases} 
  c[0]/2 & \text{for } n = 0 \\
  c^*[n] & \text{for } n = 1, 2, \cdots, M - 1 \\
  0 & \text{for } n = M, M + 1, \cdots, N - 1 
\end{cases} \quad (4.20)
\]

Equation 4.19 is for \( i = 1 \) recognised as two times the real part of the discrete-time Fourier transform of the sequence \( g[n] \). If it is sampled by \( \omega = 2\pi k/N \) for an integer \( 0 \leq k < N \), the discrete Fourier transform is obtained and equation 4.19 for \( i = 1 \) can be rewritten as

\[
J_1[k] = 2\text{Re} \left[ \sum_{n=0}^{N-1} g[n]e^{-jn2\pi k/N} \right] \quad (4.21)
\]

which can be evaluated in a fast way using an \( N \)-point FFT. Adding to this, the sequence \( J_1[k] \) can be used directly to compute \( J_i[k] \) for \( i = 2, \cdots, L \) if \( ki < N/2 \). Thus, if the maximum frequency in the subset of candidate frequencies \( \Omega \) is bounded by \( \pi/L \), the cost function can be evaluated by the sum

\[
J[k] = \sum_{i=1}^{L} J_i[k] \quad \forall k \mapsto \omega \in \Omega . \quad (4.22)
\]

The dominant cost of evaluation \( J[k] \) from 4.19 is thus in in the order of \( O(N \log_2 N) \).

Equation 4.19 involves the sequence \( g[n] \) that depends on the projection matrix \( M \) of the noise subspace which is computed from a basis of the signal subspace in equation
The cost of this is in the order of $O(M^2)$ and $O(M^2L)$, respectively. This can be improved by computing the elements of $g[n]$ directly from $W$. Recall, that $c^*[r]$, $r = 0, 1, \cdots, M - 1$ is computed from the lower triangular part of $M$. Therefore, we reformulate equation 4.19 to yield $c^*[r]$ directly by

$$
c' = 
\begin{bmatrix}
c[0] \\
c[1] \\
\vdots \\
c[M-1]
\end{bmatrix} = 
\begin{bmatrix}
m_{11} + m_{22} + m_{33} + \cdots + m_{MM} \\
m_{21} + m_{32} + \cdots + m_{M(M-1)} \\
\vdots \\
m_{M1}
\end{bmatrix}
$$

$$
= Me_1 - \sum_{k=1}^{M} [W_a]_{k:k+M-1,:}[W]^H_{k,:}
$$

where $e_1 = [1 \ 0 \ \cdots \ 0]^T$ is the unit vector and $W_a = [W \ 0]^T$ is an augmented matrix obtained by combining $W$ and the $M - 1 \times L$ zero matrix. The notation $[\cdot]_{a:b,c:d}$ indicates that a submatrix is created from the rows running from $a$ to $b$ and the columns running from $c$ to $d$ of the original matrix. If all the rows or columns are used from the original matrix, we simply write ':'. From this equation $g[n]$ is readily found. This reduces the computational complexity to $O(M^2L)$.

The entire algorithm is summarized in table 4.1. The total dominant cost is in the order of $O(M^2L) + O(N \log_2 N)$.

### 4.2.2 Fast implementation for an unknown model order

For unknown model order, the algorithm in table 4.1 can easily be extended. Recall from the HMUSIC estimator in worksheet 2 that the cost function in 4.4 must be evaluated for each candidate model order. Thus, the extension basically reduces to computing $J[k]$ in table 4.1 for each candidate model order and then comparing the normalized cost functions to find the minimum.

The algorithm can be improved if the computation of the sequence $g[n]$ is formulated in a recursive way. The recursive form is obtained from equation 4.23 which can be rewritten as

$$
c' = Me_1 - \sum_{l=1}^{L} \sum_{k=1}^{M} [S_a]_{k:k+M-1,l}[S]^H_{k,l}
$$

from which the recursive form for $l = 2, \cdots, L_{\text{max}}$ is obtained as

$$
c'_l = c'_{l-1} - \sum_{k=1}^{M} [S_a]_{k:k+M-1,l}[S]^H_{k,l}
$$

where $c'_1$ is found from (4.24) for $L = 1$.

Table 4.2 summarizes the algorithm for fast evaluation of the MUSIC estimator for unknown model order. The total dominant cost is in the order of $O(M^2L_{\text{max}}) + O(|L|N \log_2 N)$. 

---

4.2 Fast Implementation of the Calculation of the MUSIC Cost Functions
4 Optimized Fast Implementation of the MUSIC Algorithm

**Input:**
- An arbitrary orthonormal basis of the signal subspace \( W \in \mathbb{C}^{M \times L} \)
- FFT-length \( N \)
- Subset of candidate frequencies \( \Omega = \{ \omega_{\text{min}}, \ldots, \omega_{\text{max}} \} \) with cardinality \( |\Omega| \)

**Step 1:** Find \( g[n] \)
\[
c'[n] = Me_1 - \sum_{k=1}^{M} W_{a,k,k+M-1,:,:} W^{H}_{k,:} \\
g[n] = \begin{cases} 
  c[0]/2 & \text{for } r = 0 \\
  c^*[r] & \text{for } 1 \leq r < M \\
  0 & \text{for } M \leq r < N
\end{cases}
\]
\( \text{O}(M^2L) \)

**Step 2:** Compute \( J_1[k] \) from an FFT of \( g[r] \)
\[
J_1[k] = 2\text{Re} \left( \sum_{r=0}^{N-1} g[r] e^{-j2\pi rk/N} \right) \\
\text{O}(N \log_2 N)
\]

**Step 3:** Downsample \( J_1[k] \)
\[
J[k] = \sum_{i=1}^{L} J_1[k_i] \quad \forall k \mapsto \omega \in \Omega \\
\text{O}(L|\Omega|)
\]

<table>
<thead>
<tr>
<th>Table 4.1: Fast evaluation of the MUSIC estimator for a known model order.</th>
</tr>
</thead>
</table>

### 4.3 Fast and Refined Frequency Estimation Using a Gradient Method

To make the calculation of the frequency estimate even faster and more accurate, a gradient method can be introduced. Before describing how a gradient method is implemented into the MUSIC estimators a short introduction to the gradient-descent method is given.

#### 4.3.1 The Gradient-Descent Method with Backtracking Line Search

The gradient descent method is used for unconstrained minimization. As an example, an unconstrained minimization problem could be [Boyd and Vandenberghe, 2004, p. 457]

\[
\min f(x) 
\]

where \( f \) is a twice differentiable and a convex function. It is assumed that the problem is solvable, hence there exists an optimal point \( x^* \). Since it is known that \( f \) is convex and twice differentiable, a condition for the point \( x^* \) to be optimal is

\[
\nabla f(x^*) = 0. 
\]
Thus the unconstrained minimization problem can be solved by searching for a solution to equation [4.27].

When using the gradient descent method, the negative gradient \( \Delta x = -\nabla f(x) \) is used as the search direction. Knowing the search direction, the next step is to update the previous estimated value for \( x^* \) [Boyd and Vandenberghe, 2004, p. 463]

\[
x_{k+1} = x_k + t \Delta x_k
\]

(4.28)

where \( t \) is the step size which is to be suitably chosen. This can be accomplished through backtracking line search. Here, the idea is to check if the value of \( f \) decreases with the updated \( x \). If not, the step size is reduced and a new update of \( x \) is found, else, use the already updated value of \( x \). This procedure is continued until convergence is ensured.

The gradient descent algorithm is summarized in table 4.3.

### 4.3.2 Implementation of the Gradient Descent Method into MUSIC

To use the gradient descent method in MUSIC the update equation for the estimated frequency is formulated as [Christensen et al., 2007, p. 1637]

\[
\hat{\omega}_{0,k+1} = \hat{\omega}_{0,k} + t \Delta \hat{\omega}_{0,k}
\]

(4.29)

where \( \Delta \hat{\omega}_{0,k} = -\nabla J(\hat{\omega}_{0,k}) \). As seen from equation [4.29], the gradient of \( J \) with respect to the frequency has to be found. Using step 2 and step 3 from table 4.2 it can be derived as

\[
\nabla J_k = \frac{\partial}{\partial \omega} \sum_{i=1}^{L_k-1} 2 \text{Re} \left[ \sum_{r=0}^{M-1} g[r] e^{-jr\hat{\omega}_{0,k-1}} \right]
\]

(4.30)

\[
= - \sum_{i=1}^{L_k-1} i 2 \text{Re} \left[ \sum_{r=0}^{M-1} j r g[r] \left( \cos(ir\hat{\omega}_{0,k-1}) - j \sin(ir\hat{\omega}_{0,k-1}) \right) \right]
\]

(4.31)

\[
= 2 \sum_{i=1}^{L_k-1} i \sum_{r=0}^{M-1} r \left[ \text{Im}[g[r]] \text{Re}[g[r]] \right] \left[ \frac{\cos(ir\hat{\omega}_{0,k-1})}{-\sin(ir\hat{\omega}_{0,k-1})} \right].
\]

(4.32)

Besides this it is also required to calculate new values of the cost function at one frequency when the frequency is updated through the backtracking step. In a similar manner this can be found to

\[
J_k = 2 \sum_{i=1}^{L_k-1} \sum_{r=0}^{M-1} \left[ \text{Re}[g[r]] \text{Im}[g[r]] \right] \frac{\cos(ir\hat{\omega}_{0,k})}{\sin(ir\hat{\omega}_{0,k})}
\]

(4.33)

To make the gradient method faster it is possible to look at the absolute value of the gradient of \( J \) and when it gets smaller than a value \( \beta \), the backtracking step is skipped.
and $t = 0$. That is, the maximum error in the frequency estimate after using the gradient method will always be smaller than some value implicitly given by $\beta$. The gradient method applied on the MUSIC estimator is summed up in table 4.4.

The cost function $J$ is a non-convex function. Therefore, if the starting point of the gradient-descent algorithm is too far away from the optimal solution, it would return a false estimate of the fundamental frequency. To avoid this the MUSIC method should be initialized by estimating the frequency over all feasible frequencies and then, when the estimate is assumed to be close to the true fundamental frequency, the gradient-descent method can be used. A measure for the reliability of the frequency estimate could be the cost function $J$ which is a measure of how close the estimated frequency is to the true frequency. The flowchart diagram of the MUSIC method with the gradient-descent method is shown in figure 4.2.

![Figure 4.2: MUSIC estimator implemented with the gradient-descent method.](image)

4.4 Examples Frequency Estimation using the MUSIC Estimator

The proposed solutions were simulated. First, the MUSIC estimator for a known model order depicted in figure 4.2 was simulated. It was simulated on a synthetic signal with a model order of three and consisting of two equally sized time segments where the fundamental frequencies are 235 Hz and 285 Hz respectively. When the previous value of the cost function exceeded $\alpha = 20$ the full pseudo spectrum was calculated in steps of 10 Hz and the frequency was estimated as the argument of the minimum. Otherwise, the previous estimated frequency was updated using the gradient-descent method. As can be seen from the resulting frequency estimate in figure 4.3 the estimation proceeds as expected. When the value of the cost function was below $\alpha$ the frequency estimate is refined using the gradient-descent method and converged against the true frequency of
the synthetic signal.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure43.png}
\caption{Estimation of the fundamental frequency using MUSIC for a known model order with PAST and FDPM fast subspace trackers and the gradient-descent method for an SNR of 40 dB.}
\end{figure}

Next, the MUSIC estimator for an unknown model order with subspace tracking and the fast implementation of calculation of the cost function depicted in figure 4.1 was simulated. The results are shown in figure 4.4. The simulation was made as an frequency estimation using the proposed solution on a violin signal downsamplend to a sampling frequency of 11025 Hz with an SNR of 40 dB. When the previous value of the HMUSIC cost function exceeded $\alpha = 1 \cdot 10^{-3}$ the full spectrum was calculated on basis of an EVD. In the case where it was below $\alpha$ a fast subspace tracker was used and the frequency updated by the gradient-descent method. As can be seen from the figure the proposed solution is applicable since it is able to estimate the frequency correctly. Only in the frequency transitions the estimate deviates from the true frequency, but this is due to incorrect assumptions about the signal model.

**Bibliography**


Figure 4.4: Spectrogram of violin signal (top) and estimation of its fundamental frequency using MUSIC for an unknown model order with fast subspace tracking, fast implementation of calculation of the cost function and the gradient-descent method for an SNR of 40 dB (bottom).

X.G. Doukopoulos and G.V. Moustakides. Fast and stable subspace tracking. 2007. Accepted for publication in the IEEE Transactions on Signal Processing.

**Input:**
- Eigenvectors spanning the signal subspace $S \in \mathbb{C}^{M \times L_{\text{max}}}$
- FFT-length $N$
- Subset of candidate frequencies $\Omega = \{\omega_{\text{min}}, \cdots, \omega_{\text{max}}\}$ with cardinality $|\Omega|$
- Subset of candidate model orders $L = \{L_{\text{min}}, \cdots, L_{\text{max}}\}$ with cardinality $|L|$

<table>
<thead>
<tr>
<th>for $l = L_{\text{min}}, \cdots, L_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1:</strong> Find $g_l[n]$</td>
</tr>
<tr>
<td>if $l = L_{\text{min}}$</td>
</tr>
<tr>
<td>$c' = M e_1 - \sum_{k=1}^{M} [S_a]<em>{k:k+M-1,1:l} S</em>{k,1:l}^H$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$c'<em>l = c'</em>{l-1} - \sum_{k=1}^{M} [S_a]<em>{k:k+M-1,1:l} S</em>{k,1:l}^H$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

$g_l[r] = \begin{cases} c_l[0]/2 & \text{for } r = 0 \\ c'_l[r] & \text{for } 1 \leq r < M \\ 0 & \text{for } M \leq r < N \end{cases} \quad O(1)$

**Step 2:** Compute $J_{1,l}[k]$ from an FFT of $g_l[r]$

$J_{1,l}[k] = 2 \text{Re} \left[ \sum_{r=0}^{N-1} g_l[r] e^{-j2\pi rk/N} \right] \quad O(N \log_2 N)$

**Step 3:** Downsample $J_{1,l}[k]$

$J_l[k] = \sum_{i=1}^{l} J_{1,l}[ki] \quad \forall k \mapsto \omega \in \Omega_l \quad O(l|\Omega_l|)$

| Table 4.2: Fast evaluation of the MUSIC estimator for an unknown model order. |
Table 4.3: Gradient-descent method with backtracking line search.

Table 4.4: The cost function dependent gradient method.

Bibliography

Input:
- Starting point $x_1$

for $k = 1, 2, 3, \cdots$ do
  $\Delta x_k = -\nabla f(x_k)$
  $t = 1$
  while $f(x_k) + t\Delta x_k > f(x_k) + \alpha t\nabla f(x_k)\Delta x_k$ do
    $t = \mu t$
  end
  $x_{k+1} = x_k + t\Delta x_k$
end

Input:
- Estimated values from last iteration, i.e. $\hat{\omega}_{0,k-1}, \hat{L}_{k-1}$

Step 1: Find $g[r]

Step 2: Compute gradient and find descent direction

$\nabla J_k = 2 \hat{L}_{k-1} \sum_{i=1}^{M-1} \sum_{r=1}^{M-1} \begin{bmatrix} \text{Im}[g[r]] \\ \text{Re}[g[r]] \end{bmatrix}^T \begin{bmatrix} \cos(i r \hat{\omega}_{0,k-1}) \\ -\sin(i r \hat{\omega}_{0,k-1}) \end{bmatrix}$

$\Delta \omega_k = f(\nabla J_k)$

Step 3: Find step-size $t$

if $|\nabla J_k| < \beta$
  $t = 0$
else
  Find $t$ from line search
end

Step 4: Update estimates

$\hat{\omega}_{0,k} = \hat{\omega}_{0,k-1} + t\Delta \omega_k$

$J_k = 2 \hat{L}_{k-1} \sum_{i=1}^{M-1} \sum_{r=0}^{M-1} \begin{bmatrix} \text{Re}[g[r]] \\ \text{Im}[g[r]] \end{bmatrix}^T \begin{bmatrix} \cos(i r \hat{\omega}_{0,k}) \\ \sin(i r \hat{\omega}_{0,k}) \end{bmatrix}$
Poster
- Subspace-based Fundamental Frequency Estimation -