Fast Algorithms for Fundamental Frequency Estimation in Autoregressive Noise[⋆]

Barry Gerard Quinn[a], Jesper Kjær Nielsen[b,*], Mads Græsbøll Christensen[b]

[a]*Department of Mathematics and Statistics, Faculty of Science and Engineering, Macquarie University, NSW, 2109 Australia*
[b]*Audio Analysis Lab, CREATE, Aalborg University, 9000 Aalborg, Denmark*

Abstract

Many signals can accurately be modelled as a periodic function in coloured noise. An important parameter of the periodic function is the fundamental frequency. Often, fundamental frequency estimators are either ad hoc or have been derived under a white Gaussian noise (WGN) assumption. In this paper, we first derive the joint maximum likelihood (ML) estimator of the fundamental frequency estimator in autoregressive noise. Since a naïve implementation of this ML estimator has a very high computational complexity, we derive three fast algorithms that produce either exact or asymptotically equivalent estimators for all candidate sinusoidal and AR-orders. Through experiments, we show that the fast algorithms are at least two orders of magnitude faster than the naïve implementation and that the two fast approximate algorithm are faster and have a worse time-frequency resolution than the fast exact algorithm. Moreover, we show that jointly estimating the fundamental frequency and AR-parameters using our fast, exact algorithm is both faster and more accurate than computing the estimates iteratively. Finally, we apply the estimator to real data to show examples of how modelling the noise to be coloured significantly reduces the number of outliers produced by the fundamental frequency estimator compared to modelling the noise as WGN.

Keywords: Harmonic regression, coloured noise estimation, fundamental frequency estimation; pitch estimation.

[⋆] MATLAB™ implementations of the presented fast estimators as well as the code for generating all presented results can be found at `https://github.com/jkjaer/fastF0ArMl`.

[*] Corresponding author

Email addresses: `barry.quinn@mq.edu.au` (Barry Gerard Quinn), `jkn@create.aau.dk`

1. Introduction

The mathematical modelling of periodic functions has been topical since the 1807 results of Fourier [1], but had interested natural philosophers for two millennia before that, mainly because of observations of the motion of the planets. Although Fourier derived his famous results in order to solve the heat equation, his methods may be used to solve many differential equations, and are thus applicable to many other problems in acoustics and electrical engineering, in particular. A periodic function is a function which repeats its function values in regular periods. The inverse of the shortest of these periods is the fundamental frequency, and it contains important information about the periodic function. In speech processing [2], for example, a short segment of voiced speech is often modelled as a periodic function, and the fundamental frequency[1] can be used in applications such as enhancement, compression, classification, and diagnostics. Another example is order analysis [4, 5] where the fundamental frequency is used for resampling vibrational and/or acoustical data, typically generated by a rotating machine, uniformly in rotation angle instead of in time. A third example is ECG-signal analysis [6] in which the ECG signal can be accurately modelled as a periodic function [7] whose fundamental frequency is not only important for estimating the heart rate, but also for the investigation of heart rate variability [8].

Since periodic functions are encountered in many applications, there are many papers concerned with fundamental frequency estimation when the available data have been obtained at equidistant time points, with some of the most cited ones being PRAAT [9], RAPT [10], YIN [11], and recently Kaldi [12]. These four estimators (and many others) can all be derived from the comb filtering principle in which the fundamental frequency is estimated by designing a feedforward comb filter that filters out the maximum amount of energy from its input signal. This is equivalent to maximising a normalized autocorrelation function and can, therefore, be implemented using computationally efficient algorithms. The comb filtering principle is not based on a signal model and is, there-

(Jesper Kjær Nielsen), mgc@create.aau.dk (Mads Græsbøll Christensen)

[1] Note that in speech and audio processing, pitch and fundamental frequency estimation are often used synonymously, despite the pitch is referring to a perceptual phenomenon whereas the fundamental frequency is a physical quantity [3].

fore, difficult to improve in a systematic fashion. This has led to many published comb filtering-based estimators based on various heuristics. In particular, problems with the so-called subharmonic (or octave) errors have been a major issue which stems from the fact that an integer multiple of the fundamental period is also a period of the periodic function. Consequently, designing a comb filter to minimise the output energy often results in the fundamental frequency being erroneously estimated as an integer fraction of the true fundamental frequency if the periodic function is contaminated in noise. Since the comb filter seeks to minimise the total output energy, not only that pertaining to the periodic function, we can, therefore, think of the subharmonic error problem as an over-fitting problem.

As an alternative to the nonparametric approaches described above, some work has been done on model-based fundamental frequency estimation (see [13–16] and the references therein for some examples). The main idea is to model the periodic function as a finite Fourier series and the noise using a statistical model, and then estimate the model parameters, including the fundamental frequency, from the observed data. The main advantage of this approach is that the model assumptions are explicit and can be improved if they are too crude for a given application. Moreover, the model-based estimators typically outperform the nonparametric ones, provided that the model describes the data sufficiently well. The model-based estimators are also more robust to subharmonic errors since they can be combined with model order estimation to avoid (over-)fitting sinusoidal components to the noise. The main reasons for not using the model-based estimators have been the computational complexity of the developed algorithms and the fact that white Gaussian noise (WGN) is normally assumed, mostly due to mathematical tractability. Although a computationally efficient algorithm for the nonlinear least-squares fundamental frequency estimator has recently been developed in [17], its statistical and computational efficiency rely on the WGN assumption being satisfied. In practice, however, the WGN assumption is often inaccurate, and pre-processing techniques such as pre-whitening have only received limited attention (see [16, 18, 19] for some exceptions) since the noise colour is seldom known in advance. For general signal enhancement, however, noise statistics estimation is a big topic due to its time-varying, complex nature and some of these techniques have recently been compared and benchmarked for fundamental frequency estimation [20, 21]. In [21], it was found that simply iterating between a noise statistics estimator and a fundamental frequency estimator improved the performance on real-world data significantly compared to making a white noise assumption. Unfortunately, however, an iterative approach to estimating the sinusoidal and noise parameters is much more prone than a joint approach to converging to a local maximiser of the underlying non-convex optimisation problem. For fundamental frequency estimation, an iterative approach, therefore, produces more outliers, which we also demonstrate via simulations.

Only a handful of model-based fundamental frequency estimators have been derived using a coloured noise model assumption, and we here give an overview over them. In [13], it was shown that the nonlinear regression estimator assuming coloured noise could be approximated by the maximiser of a weighted sum of the periodogram evaluated at the fundamental frequency and its harmonics, with weights inversely proportional to the noise spectral density at the relevant frequencies. These weights had to be estimated, possibly by median smoothing. For the complex-valued case, [22, 23] modelled the coloured noise as being autoregressive and obtained approximate maximum likelihood estimators of the parameters, estimating the autoregressive and sinusoidal orders using MDL. In [24], the real-valued case was considered in a full Bayesian framework in which the autoregressive order was assumed known. Finally, in [16], coloured noise was modelled using a covariance matrix and a recursive scheme for updating it was proposed. Unfortunately, however, this updating has to be done for every candidate fundamental frequency which leads to a large computational complexity and memory requirements.

For completeness, we also briefly discuss the related problem of estimating the frequencies of unrelated sinusoidal components in coloured noise. In [25], it was shown that local maximizers of the periodogram, i.e., regression estimators of the frequencies, were statistically efficient in the Gaussian noise case. This was also demonstrated in [26, 27], including for the complex-valued case. In [28], it was shown how to estimate autoregressive parameters and the parameters of a single sinusoid in parallel and recursively, and this was generalised to the complex-valued case and several unrelated frequencies in [29]. For estimating the number of sinusoidal and autoregressive components in the data, [30] devised a BIC-like criterion to estimate the number of frequencies for the unrelated frequencies in the case of white noise, and [31] extended this to the coloured noise case. Important here was the idea that although the noise was neither assumed to be Gaussian nor autoregressive, the BIC procedure was computed as though the noise was autoregressive, with order bounded above by some function of the sample size. In [26], the orders were estimated using a $\log\log$ criterion which will not work here for reasons given in [30, 31].

In this paper, we first derive a joint maximum likelihood (ML) estimator of the sinusoidal and autoregressive parameters for a periodic function contaminated by autoregressive noise, including a BIC-type procedure for estimating the orders. The motivation for deriving a joint ML estimator is that it is statistically efficient asymptotically, provided that a regularity condition is fulfilled [32, Ch. 7]. Thus, the presented estimator is the optimal unbiased estimator and attains the Cramér-Rao lower bound (CRLB) if enough data are available. Unfortunately, a naïve implementation of this ML estimator has a very high computational complexity. The main part of this paper is, therefore, concerned with deriving several fast algorithms that will either exactly or approximately compute the ML estimate in a computationally efficient manner. For increased generality, we assume in what follows that the frequencies are unrelated, but the main application is for the case of harmonically related frequencies, i.e., where frequencies are integer multiples of a fundamental frequency. It should be stressed that similar, and indeed, simpler algorithms may be obtained for the complex-valued case, but also that this case leads to other problems such as a worse time-frequency resolution. We refer the interested reader to [19] for a more thorough discussion on the real- and complex-valued models for fundamental frequency estimation.

2. The autoregressive sinusoidal model

In what follows, we assume that the time between samples is 1 unit, and that frequencies are measured in radians per unit time. When time is measured in seconds, or some other unit, and frequency in Hz, it is trivial to translate the results. We assume the data to form one realisation of $\{x_t; t = 0, 1, \ldots, T-1\}$, where, for $t = 0, 1, \ldots, T-1$,

$$x_t = \sum_{j=1}^{q} \{\delta_j \cos(\omega_j(t + t_0)) + \xi_j \sin(\omega_j(t + t_0))\} + \varepsilon_t, \tag{1}$$

$$\varepsilon_t = \sum_{k=1}^{p} \beta_k \varepsilon_{t-k} + u_t. \tag{2}$$

$\{u_t\}$ are assumed to be uncorrelated, and to have enough structure that the parameter estimators have good asymptotic properties. Moreover, $\{u_t\}$ will be assumed to be stationary and ergodic, with common mean 0 and common variance σ_u^2, and the polynomial $1 - \sum_{j=1}^{p} \beta_j z^j$ to have all of its zeroes outside the unit circle. The u_t need not be independent but a martingale difference condition suffices. The start index t_0 can be any value, as it only affects the interpretation of phase which is a nuisance parameter in many applications. The start index is usually set to $t_0 = 0$, but we will see later that using symmetric time indices is computationally advantageous. The ω_j, δ_j, ξ_j, and β_j, will be assumed to be unknown (in the harmonically related case, of course, $\omega_j = j\omega$ where the fundamental frequency ω is unknown), as will the orders p and q. The key to the algorithms that follow will be that if the ω_j are known, the system of equations above can be forced into a linear system, so that linear regression and Toeplitz-like simplifications can be used. A constant or DC term μ has been deliberately omitted, since the data are usually mean-corrected, and this will have no asymptotic effect. Moreover, the DC term is known to be zero in some applications such as, e.g., in audio recordings where the physical interpretation of the DC term is the constant pressure difference between the two sides of the microphone membrane [19]. For exact least squares estimation, the term may be included. The resulting design matrix $M_{p,q}$ will then have an extra column of 1's.

For the harmonically related case, it is shown in [13] and [33] that the 'Whittle likelihood' estimator of the fundamental frequency, when the noise power spectral density (PSD)

$$\phi_\varepsilon(\omega) = (2\pi)^{-1} \sum_{k=-\infty}^{\infty} \mathbb{E}(\varepsilon_t \varepsilon_{t-k}) e^{-i\omega k}$$

is known, is

$$\widehat{\omega} = \underset{\omega}{\operatorname{argmax}} \sum_{j=1}^{q} \frac{I_X(j\omega)}{\phi_\varepsilon(j\omega)},$$

where I_X is the periodogram. Unfortunately, however, the noise PSD is seldom known in practice which makes the above estimator impractical. In [13], it was shown that $T^{3/2}\left(\widehat{\omega}-\omega\right)$ is asymptotically normally distributed with mean 0 and variance

$$\frac{48\pi}{\sum_{j=1}^{f}\frac{j^2\left(\delta_j^2+\xi_j^2\right)}{\phi_\varepsilon(j\omega)}}. \tag{3}$$

When $\{\varepsilon_t\}$ is autoregressive, this central limit theorem is true whether or not the autoregressive parameters are estimated, as the autoregressive parameter estimators are asymptotically independent of the estimators of the sinusoidal parameters. This result thus holds when the full Gaussian likelihood is maximised, and for the procedure outlined in this paper, whether of not $\{u_t\}$ is Gaussian.

3. The F$_0$-AR-ML Estimator

By solving (1) for ε_t and inserting this in (2), we obtain

$$x_t = u_t + \sum_{k=1}^{p}\beta_k x_{t-k} + \sum_{j=1}^{q}\left\{\alpha_j\cos\left(\omega_j(t+t_0)\right)+\gamma_j\sin\left(\omega_j(t+t_0)\right)\right\}, \tag{4}$$

for $t=0,1,\ldots,T-1$, where

$$\begin{bmatrix}\alpha_j\\ \gamma_j\end{bmatrix} = \begin{bmatrix}\operatorname{Re}\beta\left(\omega_j\right) & \operatorname{Im}\beta\left(\omega_j\right)\\ -\operatorname{Im}\beta\left(\omega_j\right) & \operatorname{Re}\beta\left(\omega_j\right)\end{bmatrix}\begin{bmatrix}\delta_j\\ \xi_j\end{bmatrix}, \tag{5}$$

$$\beta\left(\omega\right) = 1-\sum_{k=1}^{p}\beta_k e^{-ik\omega} \tag{6}$$

so that

$$\alpha_j - i\gamma_j = \left(\delta_j - i\xi_j\right)\beta\left(\omega_j\right). \tag{7}$$

Since the AR-parameters $\{\beta_k\}$ and the sinusoidal weights $\{\alpha_j\}$ and $\{\gamma_j\}$ are all linear parameters, we can for fixed $\{\omega_j\}$ write the signal model as a linear regression

$$X_P = M_{p,q}\theta_{p,q}+u_P. \tag{8}$$

To facilitate fast algorithms, we assume that unobserved data points are zero, i.e., we put all values of x_t for $t<0$ and $t>T-1$ equal to 0^2. Thus,

$$X = \begin{bmatrix}x_0 & \cdots & x_{T-1}\end{bmatrix}', \quad X_P = \begin{bmatrix}X' & \overbrace{0 \;\cdots\; 0}^{P\text{ zeros}}\end{bmatrix}'.$$

[2] Note that this assumption is often referred to as the autocorrelation method in the context of linear prediction [34].

where P is the maximum candidate AR-order. Although this assumption will not matter asymptotically, it will cause the estimator to have a variance slightly higher than the optimal value in (3) for a finite T. We quantify this loss through experiments in Sec. 5. The remaining quantities in (8) are defined as

$$\begin{aligned}
u_P &= \begin{bmatrix} u_0 & \cdots & u_{T_P-1} \end{bmatrix}', \\
M_{p,q} &= \begin{bmatrix} Z_p & E_q \end{bmatrix}, \\
Z_p &= \begin{bmatrix} x_{-1} & \cdots & x_{-p} \\ \vdots & & \vdots \\ x_{T_P-2} & \cdots & x_{T_P-1-p} \end{bmatrix} \\
E_q &= \begin{bmatrix} \widetilde{E}_1 & \cdots & \widetilde{E}_q \end{bmatrix}, \\
\widetilde{E}_j &= \begin{bmatrix} \cos\left(t_0\omega_j\right) & \sin\left(t_0\omega_j\right) \\ \cos\left((t_0+1)\omega_j\right) & \sin\left((t_0+1)\omega_j\right) \\ \vdots & \vdots \\ \cos\left((t_0+T_P-1)\,\omega_j\right) & \sin\left((t_0+T_P-1)\,\omega_j\right) \end{bmatrix}, \\
\theta'_{p,q} &= \begin{bmatrix} \widetilde{\beta}_1 & \cdots & \widetilde{\beta}_p & \alpha_1 & \gamma_1 & \cdots & \alpha_q & \gamma_q \end{bmatrix} = \begin{bmatrix} \beta'_p & d'_q \end{bmatrix}.
\end{aligned}$$

Note that only the last $2q$ columns of $M_{p,q}$, i.e., E_q, depend on $\{\omega_j\}$. To make sure the estimated noise variance is non-negative for a finite data length T, the sinusoidal matrix E_q must have the same number of rows as Z_p, i.e., $T_p = T + P$ rows. For fixed values of $\{\omega_j\}$, the least squares estimator of $\theta_{p,q}$ is

$$\widehat{\theta}_{p,q} = \left(T^{-1}M'_{p,q}M_{p,q}\right)^{-1}\left(T^{-1}M'_{p,q}X_P\right).$$

If the linear parameters in the nonlinear regression model in (8) are expressed in terms of this least squares estimator, we obtain the residual

$$\widehat{u}_{p,q} = X_P - M_{p,q}\widehat{\theta}_{p,q},$$

and the residual mean square

$$\widehat{\sigma}^2_{p,q} = T^{-1}\widehat{u}'_{p,q}\widehat{u}_{p,q} = T^{-1}\left(X'_P X_P - X'_P M_{p,q}\widehat{\theta}_{p,q}\right). \tag{9}$$

The estimators of $\{\omega_j\}$ in the unrelated frequencies case are the values minimising this with respect to $\omega_1, \ldots, \omega_q$. In the harmonically related frequencies case, $\omega_j = j\omega$ and the residual mean square is a function of ω alone, but depends on both p and q, and our aim is to estimate p, q, and ω efficiently. This is quite complicated because of the nonlinear dependence of $\widehat{\sigma}^2_{p,q}$ on the harmonic frequencies $\{\omega_j = j\omega\}$ and difficult to implement in a real-time signal processing system. For estimating the orders p and q, we use the BIC algorithm proposed in [31]. For unrelated frequencies, this suggests minimising with respect to $p \geq 0$ and $q \geq 0$

$$\phi_{p,q} = T \log \widehat{\sigma}^2_{p,q} + (p + 5q) \log T. \tag{10}$$

The number 5 appears because the asymptotic variances of the estimators of $\{\omega_j\}$ are of order T^{-3}, while those of the components of the d_q are of order T^{-1} (see also [31, 33] for an in-depth discussion on this). Thus, in the harmonically related case, we should minimise

$$\phi_{p,q} = T \log \widehat{\sigma}^2_{p,q} + \begin{cases} p \log T & q = 0 \\ (p + 2q + 3) \log T & q > 0 \end{cases}$$

or equivalently

$$\phi_{p,q} = T \log \widehat{\sigma}^2_{p,q} + (p + 2q) \log T, \tag{11}$$

since the fundamental frequency is then the only frequency that is being estimated.

4. Three Fast Algorithms

The main contribution in this paper is three fast algorithms for either exactly or approximately computing the residual mean square $\widehat{\sigma}^2_{p,q}$ over a grid of candidate fundamental frequencies $\omega \in (0, \pi/q)$ for all candidate model orders $p = 0, 1, \ldots, P$ and $q = 0, 1, \ldots, Q$. The algorithms will be referred to as the F$_0$-AR-ML-E, F$_0$-AR-ML-A1, and F$_0$-AR-ML-A2 algorithms, all being asymptotically equivalent and asymptotically efficient. For a finite data length T, however, only F$_0$-AR-ML-A1 and F$_0$-AR-ML-A2 will produce the same estimate (as we suggest by the naming). Moreover, these two algorithms only approximately solve the problem since they do not compute the residual mean square in (9) exactly. The F$_0$-AR-ML-E algorithm, on the other hand, will compute the residual mean square exactly, but will be computationally more complex. The two approximate algorithms, F$_0$-AR-ML-A1 and F$_0$-AR-ML-A2, exploit the fact that

$$\lim_{T \to \infty} T^{-1} E'_q E_q = \frac{1}{2} I_{2q} \tag{12}$$

to replace $T^{-1}M'_{p,q}M_{p,q}$ with a matrix that is asymptotically equivalent, but with a much simpler structure. This allows us to formulate fast algorithms which are computationally simpler. Note that the asymptotic result in (12) is frequently employed for deriving computationally efficient frequency estimators such as the harmonic summation method [35, 36] for fundamental frequency estimation in white noise. The main disadvantage of the approximation is that it might produce spurious frequency estimates for a small data length T, fundamental frequencies in the order of $1/T$, or high SNRs. This is a well-known problem, even for periodic functions in white noise, and the remedy is to employ exact estimators when the above conditions are not satisfied [19].

Before describing these three fast algorithms in detail, we first rewrite the residual mean square in (9) in two different ways, since these will be the starting points for the derivation of the fast algorithms. By inserting the expression for the least squares estimator of the linear parameters in (9), we obtain

$$\widehat{\sigma}^2_{p,q} = T^{-1}X'_P P^{\perp}_{M_{p,q}} X_P = T^{-1}\left(X'_P X_P - X'_P P_{M_{p,q}} X_P\right)$$

where $P_{M_{p,q}} = M_{p,q}(M'_{p,q}M_{p,q})^{-1}M'_{p,q}$ and $P^{\perp}_{M_{p,q}} = I_{T_P} - P_{M_{p,q}}$ are the orthogonal projector and the complementary orthogonal projector, respectively. The projection matrix $P_{M_{p,q}}$ can be written in two different ways as described in the following lemma.

Lemma 1. *The projection matrix $P_{M_{p,q}} = M_{p,q}(M'_{p,q}M_{p,q})^{-1}M'_{p,q}$ with $M_{p,q} = \begin{bmatrix} Z_p & E_q \end{bmatrix}$ can be written as*

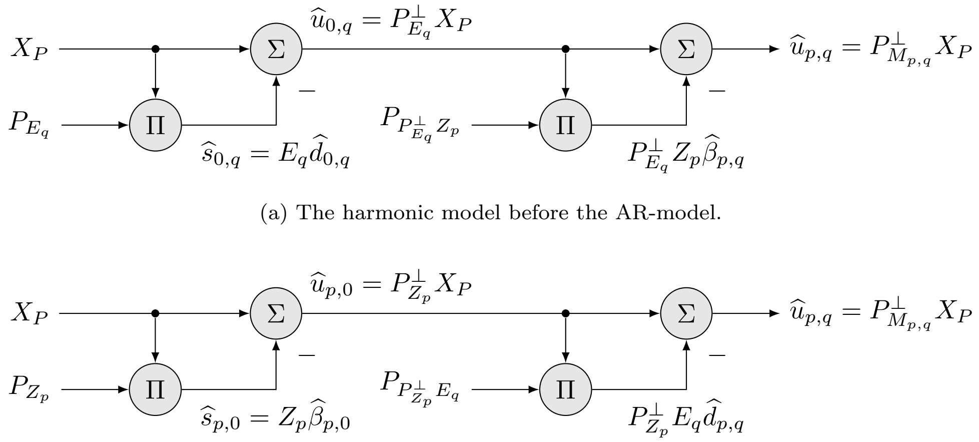
$$P_{M_{p,q}} = P_{E_q} + P^{\perp}_{E_q} Z_p \left(Z'_p P^{\perp}_{E_q} Z_p\right)^{-1} Z'_p P^{\perp}_{E_q} \tag{13}$$

$$= P_{Z_p} + P^{\perp}_{Z_p} E_q \left(E'_q P^{\perp}_{Z_p} E_q\right)^{-1} E'_q P^{\perp}_{Z_p} \tag{14}$$

where

$$P^{\perp}_{E_q} = I_{T_P} - P_{E_q} = I_{T_P} - E_q\left(E'_q E_q\right)^{-1} E'_q$$
$$P^{\perp}_{Z_p} = I_{T_P} - P_{Z_p} = I_{T_P} - Z_p\left(Z'_p Z_p\right)^{-1} Z'_p.$$

Proof. Follows from first forming the 2×2 block matrix $M'_{p,q}M_{p,q}$. The inverse of this matrix can then be expressed in two different ways as described in, e.g., [37]. The two expressions for $P_{M_{p,q}}$ then follow from left and right multiplying these inverses with $M_{p,q}$ and $M'_{p,q}$, respectively. □


X_P
$\widehat{u}_{0,q} = P^{\perp}_{E_q} X_P$
Σ
$\widehat{u}_{p,q} = P^{\perp}_{M_{p,q}} X_P$
$-$
P_{E_q}
Π
$\widehat{s}_{0,q} = E_q \widehat{d}_{0,q}$
$P_{P^{\perp}_{E_q} Z_p}$
$P^{\perp}_{E_q} Z_p \widehat{\beta}_{p,q}$
X_P
$\widehat{u}_{p,0} = P^{\perp}_{Z_p} X_P$
Σ
$\widehat{u}_{p,q} = P^{\perp}_{M_{p,q}} X_P$
$-$
P_{Z_p}
Π
$\widehat{s}_{p,0} = Z_p \widehat{\beta}_{p,0}$
$P_{P^{\perp}_{Z_p} E_q}$
$P^{\perp}_{Z_p} E_q \widehat{d}_{p,q}$

(a) The harmonic model before the AR-model.

(b) The AR-model before the harmonic model.

Figure 1: Block diagrams of two different ways of computing the residual $\widehat{u}_{p,q}$. Note that the multiplication nodes map the data onto the different subspaces.

Fig. 1 illustrates how Lemma 1 can be used for writing the residual $\widehat{u}_{p,q} = P^{\perp}_{M_{p,q}} X_P$ in two different ways. This also means that the residual mean square can be written in two different ways as

$$\widehat{\sigma}^2_{p,q} = \widehat{\sigma}^2_{0,q} - \rho'_{p,q} \Upsilon^{-1}_{p,q} \rho_{p,q} = \widehat{\sigma}^2_{0,q} - \rho'_{p,q} \widehat{\beta}_{p,q} \tag{15}$$

$$= \widehat{\sigma}^2_{p,0} - g'_{p,q} \Omega^{-1}_{p,q} g_{p,q} = \widehat{\sigma}^2_{p,0} - g'_{p,q} \widehat{d}_{p,q}, \tag{16}$$

where (15) and (16) follow from (13) and (14), respectively. In (15), we have defined

$$\widehat{\sigma}^2_{0,q} = T^{-1} X'_P P^{\perp}_{E_q} X_P \tag{17}$$

$$\rho_{p,q} = T^{-1} Z'_p P^{\perp}_{E_q} X_P \tag{18}$$

$$\Upsilon_{p,q} = T^{-1} Z'_p P^{\perp}_{E_q} Z_p. \tag{19}$$

Note that $\widehat{\beta}_{p,q} = \Upsilon^{-1}_{p,q} \rho_{p,q}$ and that $\rho_{p,q}$ can be interpreted as a covariance vector. In (16), we have defined

$$\widehat{\sigma}^2_{p,0} = T^{-1} X'_P P^{\perp}_{Z_p} X_P,$$

$$g_{p,q} = T^{-1} E'_q P^{\perp}_{Z_p} X_P, \tag{20}$$

$$\Omega_{p,q} = T^{-1} E'_q P^{\perp}_{Z_p} E_q. \tag{21}$$

Note that $\widehat{d}_{p,q} = \Omega_{p,q}^{-1} g_{p,q}$ and that $g_{p,q}$ can be interpreted as the DFT of the residual $\widehat{u}_{p,0}$ at the frequencies $\{\omega_j\}$.

4.1. The F_0-AR-ML-E Algorithm

As described above, the F$_0$-AR-ML-E computes the residual mean square exactly. The algorithm is based on the expression in (15). From the expressions in (17) – (19), we see that computing

$$\Sigma_{p,q} = T^{-1} \begin{bmatrix} X_P' \\ Z_p' \end{bmatrix} P_{E_q}^{\perp} \begin{bmatrix} X_P & Z_p \end{bmatrix} \tag{22}$$

efficiently for all values of p and q is central in a fast algorithm since the values of $\widehat{\sigma}_{0,q}^2$, $\rho_{p,q}$, and $\Upsilon_{p,q}$ can all be extracted directly from $\Sigma_{p,q}$ as

$$\Sigma_{p,q} = \begin{bmatrix} \widehat{\sigma}_{0,q}^2 & \rho_{p,q}' \\ \rho_{p,q} & \Upsilon_{p,q} \end{bmatrix}. \tag{23}$$

Writing out the expression for $\Sigma_{p,q}$ in (22) gives

$$\Sigma_{p,q} = T^{-1} \begin{bmatrix} X_P' \\ Z_p' \end{bmatrix} \begin{bmatrix} X_P & Z_p \end{bmatrix} - T^{-1} \begin{bmatrix} X_P' \\ Z_p' \end{bmatrix} E_q \left(E_q' E_q\right)^{-1} E_q' \begin{bmatrix} X_P & Z_p \end{bmatrix}.$$

The first term is a symmetric Toeplitz matrix which means we only have to compute the first column. The first element of this column is $\widehat{\sigma}_{0,0}^2 = T^{-1} X_P' X_P$ whereas the last p elements will be $\rho_{p,0}$, defined in (18). The second term in the expression for $\Sigma_{p,q}$ is more challenging since this involves a matrix inversion which depends on $\{\omega_j\}$ and must, therefore, be evaluated for all candidate frequencies. As shown in the following theorem, the matrix of residual mean squares $\Sigma_{p,q}$ can be computed recursively in q using a recursive algorithm.

Theorem 2. *The matrix $\Sigma_{p,q+1}$ in* (22) *for an AR-order p can be computed recursively for $q = 0, 1, \ldots, Q-1$ from*

$$\lambda_{p,q+1} = \begin{bmatrix} X' \\ Z_p' \end{bmatrix} E_{q+1} \Gamma_{q+1}$$

$$\Sigma_{p,q+1} = \Sigma_{p,q} - T^{-1} \lambda_{p,q+1} \lambda_{p,q+1}'. \tag{24}$$

The $2q \times 2$ *matrix* Γ_q *is defined as*

$$\Gamma_q \triangleq -\zeta_q \tilde{\zeta}_q^{-1/2} \tag{25}$$

where $\tilde{\zeta}_q$ *is the last* 2×2 *block of* ζ_q *given by*

$$\zeta_q = \left(E_q' E_q\right)^{-1} \begin{bmatrix} 0 \\ I_2 \end{bmatrix}. \tag{26}$$

Note that since $\Sigma_{p,q}$ *is a symmetric matrix, only its unique elements have to be computed in the recursive computation of it. Note also that* $\Sigma_{p,q}$ *for* $p = 1, 2, \dots P-1$ *can be extracted from* $\Sigma_{P,q}$ *as the upper-left* $p \times p$ *submatrix.*

Proof. Since

$$\begin{aligned}\left(E_{q+1}' E_{q+1}\right)^{-1} &= \begin{bmatrix} E_q' E_q & E_q' \tilde{E}_{q+1} \\ \tilde{E}_{q+1}' E_q & \tilde{E}_{q+1}' \tilde{E}_{q+1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \left(E_q' E_q\right)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} E_q^{+} \tilde{E}_{q+1} \\ -I_2 \end{bmatrix} \left(\tilde{E}_{q+1}' P_{E_q}^{\perp} \tilde{E}_{q+1}\right)^{-1} \begin{bmatrix} \tilde{E}_{q+1}' (E_q^{+})' & -I_2 \end{bmatrix}\end{aligned}$$

where $E_q^{+} = \left(E_q' E_q\right)^{-1} E_q'$ is the pseudo-inverse, it follows that

$$\left(E_{q+1}' E_{q+1}\right)^{-1} \begin{bmatrix} 0 \\ I_2 \end{bmatrix} = \begin{bmatrix} -E_q^{+} \tilde{E}_{q+1} \\ I_2 \end{bmatrix} \left(\tilde{E}_{q+1}' P_{E_q}^{\perp} \tilde{E}_{q+1}\right)^{-1} \triangleq \zeta_{q+1}.$$

From the definition of Γ_q in (25), we, therefore, get

$$\left(E_{q+1}' E_{q+1}\right)^{-1} = \begin{bmatrix} \left(E_q' E_q\right)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \Gamma_{q+1} \Gamma_{q+1}'.$$

If we now let $Y_p = \begin{bmatrix} X_P & Z_p \end{bmatrix}$, we obtain

$$Y_p' P_{E_{q+1}} Y_p = \begin{bmatrix} Y_p' E_q & Y_p' \tilde{E}_{q+1} \end{bmatrix} \left(E_{q+1}' E_{q+1}\right)^{-1} \begin{bmatrix} E_q' Y_p \\ \tilde{E}_{q+1}' Y_p \end{bmatrix} = Y_p' P_{E_q} Y_p + \lambda_{p,q+1} \lambda_{p,q+1}'.$$

Inserting this into (22) then gives the final recursion. □

The recursive algorithm in Theorem 2 includes a data independent step in which ζ_q is computed in (26). In real-time applications where X is just one segment of a much longer signal, $\{\Gamma_q\}$ in (25) should be computed offline and stored in memory prior to running the recursion in Theorem 2 for every data vector X_P. Note that in the harmonically related case, (26) can also be solved efficiently and recursively by exploiting the block Toeplitz-plus-Hankel structure in a permuted version of the matrix $E_q' E_q$. For more details on this, we refer the interested reader to [17] where an algorithm was proposed for computing $\widehat{\sigma}^2_{0,q}$ in a computationally efficient manner over a grid of fundamental frequencies $\omega \in (0, \pi/q)$ for the candidate model orders $q = 0, 1, \ldots, Q$.

The $(p+1) \times 2$ matrix $\lambda_{p,q}$ in Theorem 2 can be computed more efficiently by using symmetric time indices and by exploiting the structure in E_q and $\begin{bmatrix} X_P & Z_p \end{bmatrix}$. We first focus on the data independent computation of ζ_q in (26) and show that half of its elements are zero when symmetric time indices are used.

Lemma 3. *For the start index $t_0 = -(T_P - 1)/2$, we obtain*

$$\widetilde{\Psi}_{j,k} = \widetilde{E}_j' \widetilde{E}_k = \begin{bmatrix} \psi(\omega_j - \omega_k) + \psi(\omega_j + \omega_k) & 0 \\ 0 & \psi(\omega_j - \omega_k) - \psi(\omega_j + \omega_k) \end{bmatrix}$$

where, for $\eta \in (-2\pi, 2\pi)$,

$$\psi(\eta) = \begin{cases} T_P/2 & \eta = 0 \\ \frac{1}{2}\frac{\sin(\eta T_P/2)}{\sin(\eta/2)} & \eta \neq 0 \end{cases} .$$

Therefore, the matrix $E_q' E_q$ can easily by permuted into a 2×2 block diagonal matrix from which is follows that the solution ζ_q to (26) *consists of q 2×2 diagonal matrices. In the harmonically related case, i.e., $\omega_j \pm \omega_k = \omega(j \pm k)$, the matrix $E_q' E_q$ can be permuted into a block Toeplitz-plus-Hankel matrix and* (26) *can be solved efficiently as shown in [17].*

Proof. Let $\eta \in (-2\pi, 0) \cup (0, 2\pi)$ unless otherwise stated. From the geometric series, it follows that

$$\sum_{t=0}^{T_P - 1} e^{\pm i\eta(t_0 + t)} = e^{\pm i\eta\left(t_0 + \frac{T_P - 1}{2}\right)} \frac{\sin(\eta T_P/2)}{\sin(\eta/2)}.$$

For $\eta = 0$, the sum equals T_P. By combining the above result with Euler's formula, we obtain for $t_0 = -(T_P - 1)/2$ that

$$\sum_{t=0}^{T_P-1} \cos(\eta(t-(T_P-1)/2)) = \frac{\sin(\eta T_P/2)}{\sin(\eta/2)}$$
$$\sum_{t=0}^{T_P-1} \sin(\eta(t-(T_P-1)/2)) = 0$$

The two sums equal T_P and 0, respectively, for $\eta = 0$. The final result then follows by rewriting each element of the 2×2 matrix $\widetilde{E}_j'\widetilde{E}_k$ using the product-to-sum identities

$$\cos(\theta_1)\cos(\theta_2) = [\cos(\theta_1 - \theta_2) + \cos(\theta_1 + \theta_2)]/2 \tag{27}$$
$$\sin(\theta_1)\sin(\theta_2) = [\cos(\theta_1 - \theta_2) - \cos(\theta_1 + \theta_2)]/2 \tag{28}$$
$$\sin(\theta_1)\cos(\theta_2) = [\sin(\theta_1 - \theta_2) + \sin(\theta_1 + \theta_2)]/2. \tag{29}$$

□

To compute $\lambda_{p,q}$ efficiently, we not only need to compute Γ_q efficiently, but also $E_q'\begin{bmatrix} X_P & Z_p \end{bmatrix}$. Let

$$f_q = \begin{bmatrix} \widetilde{f}_1' & \cdots & \widetilde{f}_q' \end{bmatrix}' = T^{-1}E_q'X_P, \tag{30}$$

and let $\widetilde{f}_j$ be the j'th 2×1 vector of f_q As we show in the next lemma, $E_q'Z_p$ can be obtained by rotating the $\widetilde{f}_j$.

Lemma 4. *The $p \times 2q$ matrix $T_P^{-1}Z_p'E_q$ can be written as*

$$T^{-1}Z_p'E_q = \Xi_{p,q}G_q$$

where

$$\Xi_{p,q} = \begin{bmatrix} \widetilde{\Xi}_{1,q} & \cdots & \widetilde{\Xi}_{p,q} \end{bmatrix}' \tag{31}$$
$$\widetilde{\Xi}_{k,q} = \begin{bmatrix} \cos(k\omega_1) & \sin(k\omega_1) & \cdots & \cos(k\omega_q) & \sin(k\omega_q) \end{bmatrix}'$$
$$G_q = \begin{bmatrix} \widetilde{G}_1 & & 0 \\ & \ddots & \\ 0 & & \widetilde{G}_q \end{bmatrix}, \ \widetilde{G}_j = \begin{bmatrix} \left[\widetilde{f}_j\right]_1 & \left[\widetilde{f}_j\right]_2 \\ -\left[\widetilde{f}_j\right]_2 & \left[\widetilde{f}_j\right]_1 \end{bmatrix}. \tag{32}$$

The notation $[\cdot]_k$ means the kth entry of a vector. Note that

$$\widetilde{\Xi}_{k,q} = W_q^k \iota_q$$
$$G_q^{-\prime} f_q = \iota_q$$

where

$$W_q = W_q^{-\prime} = \begin{bmatrix} \widetilde{W}_1 & & 0 \\ & \ddots & \\ 0 & & \widetilde{W}_q \end{bmatrix} \tag{33}$$

$$\widetilde{W}_j = \begin{bmatrix} \cos(\omega_j) & -\sin(\omega_j) \\ \sin(\omega_j) & \cos(\omega_j) \end{bmatrix} \tag{34}$$

$$\iota_q = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}^{\prime}. \tag{35}$$

Moreover, the $\{\widetilde{G}_j\}$ *can all be written in terms of* $\widetilde{f}_1$ *in the harmonically related case, i.e.,* $\omega_j = j\omega$*, as shown later in Lemma 5.*

Proof. Let L be a cyclic permutation matrix given by

$$L = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix}$$

so that the k'th column vector $\widetilde{z}_k$ of Z_p can be expressed in terms of X_P as

$$\widetilde{z}_k = L^k X_P.$$

Since

$$\widetilde{E}_j = \begin{bmatrix} \cos(\omega_j t_0) & \cdots & \cos(\omega_j (t_0 + T_P - 1)) \\ \sin(\omega_j t_0) & \cdots & \sin(\omega_j (t_0 + T_P - 1)) \end{bmatrix}^{\prime},$$

it follows from the product-to-sum identities in (27)—(29) that

$$\widetilde{E}_j^{\prime} L = \widetilde{W}_j \widetilde{E}_j^{\prime},$$

or, more generally, that

$$E_q^{\prime} L^k = W_q^k E_q^{\prime},$$

where $\widetilde{W}_j$ and W_q are defined in (34) and (33), respectively. From the product-to-sum identities, it also follows that

$$\widetilde{W}_j^k = \begin{bmatrix} \cos(k\omega_j) & -\sin(k\omega_j) \\ \sin(k\omega_j) & \cos(k\omega_j) \end{bmatrix}.$$

This leads to

$$\begin{aligned}T^{-1}E_q'Z_p &= T^{-1}E_q'\begin{bmatrix}LX_P & \cdots & L^pX_P\end{bmatrix} = \begin{bmatrix}W_qf_q & \cdots & W_q^pf_q\end{bmatrix}\\ &= G_q'\begin{bmatrix}W_q\iota_q & \cdots & W_q^p\iota_q\end{bmatrix} = G_q'\Xi_{p,q}'\end{aligned}$$

where f_q, $\Xi_{p,q}$, G_q, and ι_q are defined in (30), (31), (32), and (35), respectively. The final result then follows by transposing both sides. □

Lemma 3 and Lemma 4 describe how $\lambda_{p,q}$ in Theorem 2 can be computed efficiently given f_q. Although the vector f_q depends on $\{\omega_j\}$, we have so far not made this dependency explicit to simplify the notation. However, f_q has to be evaluated for all candidate frequencies and the next lemma shows how this can be done efficiently for the harmonically related case on a uniform grid of candidate frequencies using an FFT algorithm. For the unrelated frequencies case, we in principle have to form f_q from all unique combinations of candidate frequencies which is clearly impractical for even moderate values of q's.

Lemma 5. *For the frequencies $\omega^{(f)} = 2\pi f/F$ where $F > T_P$ and $f = 1, \ldots, \lfloor F/(2q) \rfloor$, the vector $\widetilde{f}_j^{(f)}$ can be computed as*

$$\widetilde{f}_j^{(f)} = \widetilde{f}_1^{(jf)}$$

in the harmonically related frequencies case where

$$\widetilde{f}_1^{(f)} = T^{-1}\begin{bmatrix}\operatorname{Re}\left(e^{-i\omega^{(f)}t_0}\widehat{X}^{(f)}\right) & -\operatorname{Im}\left(e^{-i\omega^{(f)}t_0}\widehat{X}^{(f)}\right)\end{bmatrix}'$$

with $\widehat{X}^{(f)}$ being the f'th DFT bin of the F-point DFT of X.

Proof. The vector $\widetilde{f}_j^{(f)} = T^{-1}\left(\widetilde{E}_j^{(f)}\right)' X_P$ is defined as

$$\widetilde{f}_j^{(f)} = \begin{bmatrix} T^{-1}\operatorname{Re}\left(\sum_{t=0}^{T-1} x_t e^{-ij\omega^{(f)}(t+t_0)}\right) \\ -T^{-1}\operatorname{Im}\left(\sum_{t=0}^{T-1} x_t e^{-ij\omega^{(f)}(t+t_0)}\right)\end{bmatrix}.$$

Since

$$j\omega^{(f)} = 2\pi jf/F = \omega^{(jf)}$$

in the harmonically related frequencies case, the term inside the real and imaginary operators can be written as

$$\sum_{t=0}^{T-1} x_t e^{-ij\omega^{(f)}(t+t_0)} = e^{-i\omega^{(jf)}t_0}\widehat{X}^{(fj)}$$

and $\widehat{X}^{(f)}$ can be recognised as the DFT of X. □

Theorem 2, Lemma 3, and Lemma 4 allow us to compute the unique elements in the symmetric, positive definite matrix $\Sigma_{p,q}$ efficiently for all candidate model orders p and q. For the case of harmonically related frequencies, which is what we focus on in this paper, Lemma 5 results in a further computational reduction since it shows how the vector f_q can be formed for all candidate frequencies by selecting elements from the DFT of the data vector. The values for $\widehat{\sigma}^2_{0,q}$, $\rho_{p,q}$, and $\Upsilon_{p,q}$ can all be extracted directly from $\Sigma_{p,q}$ as described by (23). Thus, the only term left computing in the residual mean square $\widehat{\sigma}^2_{p,q}$ in (15) is the error term $\rho'_{p,q}\Upsilon^{-1}_{p,q}\rho_{p,q}$ which is what we focus on now.

We first rewrite $\rho'_{p,q}\Upsilon^{-1}_{p,q}\rho_{p,q}$ in terms of the lower triangular Cholesky factor $C_{p,q}$ of $\Upsilon_{p,q}$ as

$$\rho'_{p,q}\Upsilon^{-1}_{p,q}\rho_{p,q} = \rho'_{p,q}\left(C_{p,q}C'_{p,q}\right)^{-1}\rho_{p,q} = \eta'_{p,q}\eta_{p,q}$$

where $\eta_{p,q} = C^{-1}_{p,q}\rho_{p,q}$. For $p = 1, \dots, P$, $\eta_{p,q}$ and, consequently, $\rho'_{p,q}\Upsilon^{-1}_{p,q}\rho_{p,q}$ can, therefore, be computed efficiently and recursively using forward substitution as described in, e.g., [38, Sec. 3.1.1]. Computing the Cholesky factor $C_{p,q}$ directly from $\Upsilon_{p,q}$ requires $\mathcal{O}(p^3)$ operations. Due to the rank-2 update of $\Sigma_{p,q}$ and, consequently, $\Upsilon_{p,q}$ in (24), however, we can instead reqursively update $C_{p,q}$ in q using two rank-1 Cholesky downdatings, requiring $\mathcal{O}(p^2)$ operations each. The rank-1 downdating procedure is described in the next lemma.

Lemma 6. *Let the Cholesky factor $C_{p,q}$ of the positive definite matrix $\Upsilon_{p,q}$, the vector $v_{p,q+1}$, and the positive definite matrix $\Upsilon_{p,q+1}$ all be known and related by*

$$\Upsilon_{p,q+1} = \Upsilon_{p,q} - v_{p,q+1}v'_{p,q+1} = C_{p,q}C'_{p,q} - v_{p,q+1}v'_{p,q+1}.$$

The columns of the Cholesky factor $C_{p,q+1}$ of $\Upsilon_{p,q+1}$ can then be computed for $k = 1, \dots, p$

as

$$\begin{aligned}\phi_k &= \frac{[v_{p,q+1}]_k}{[C_{p,q}]_{k,k}}\\ [C_{p,q+1}]_{1:p,k} &= \frac{[C_{p,q}]_{1:p,k} - \phi_k v_{p,q+1}}{\sqrt{1-\phi_k^2}}\\ v_{p,q+1} &= \sqrt{1-\phi_k^2} v_{p,q+1} - \phi_k \left[C_{p,q+1}\right]_{1:p,k}\end{aligned}$$

where $[\cdot]_{k:p,j}$ *selects the matrix elements in row* k *trough* p *in column* j*. Note that the* k*'th iteration introduces a zero in the* k*'th position of* $v_{p,q+1}$ *and the first* $k-1$ *positions of* $[C_{p,q+1}]_{1:p,k}$*. This can be exploited to make the above recursion more efficient by only updating the non-zero elements in each iteration. Also note that the Cholesky factor of an AR-order* p *can be extracted from the Cholesky factor of the largest AR-order* P *as the upper left* $p \times p$ *submatrix.*

Proof. See [38, Sec. 6.5.4]. The so-called mixed downdating implementation of the hyperbolic rotations used in the lemma is due to [39, App. B.3]. □

To use Lemma 6 to compute $C_{p,q}$ for $q > 0$, we need the Cholesky factor $C_{p,0}$ of $\Upsilon_{p,0}$. Since $\Upsilon_{p,0}$ is a Toeplitz matrix, its Cholesky factorisation can be computed effiently using a generalized Schur algorithm. Moreover, this algorithm also computes the residual mean square $\widehat{\sigma}^2_{p,0}$ which for $q = 0$ equals the prediction error variance in a linear prediction problem. The following theorem describes the procedure.

Theorem 7. *Initialise the vectors* t_0 *and* τ_0 *as*

$$\begin{aligned}t_0 &= \begin{bmatrix}\widehat{\sigma}^2_{0,0} & \rho'_{p,0}\end{bmatrix}'\\ \tau_0 &= \begin{bmatrix}0 & \rho'_{p,0}\end{bmatrix}'\end{aligned}$$

where t_0 *is the first column of the symmetric, positive definite, and Toeplitz matrix* $\Sigma_{p,0}$*. The Cholesky factor* $C_{p,0}$ *of* $\Upsilon_{p,0}$*, which is the lower-right submatrix of* $\Sigma_{p,0}$ *as described by* (23)*, and the residual mean squares* $\{\widehat{\sigma}^2_{k,0}\}_{k=1}^p$ *can then be computed recursively for*

$k = 1, \ldots, p$ as

$$\begin{aligned}
\left[C_{p,0}\right]_{1:p,k} &= \widehat{\sigma}_{0,0}^{-1} \left[t_{k-1}\right]_{1:p} \\
\nu_k &= \frac{\left[\tau_{k-1}\right]_{k+1}}{\left[t_k\right]_k} \\
\widehat{\sigma}_{k,0}^2 &= \widehat{\sigma}_{k-1,0}^2(1-\nu_k^2) \\
t_k &= \frac{\begin{bmatrix} 0 & \left[t_{k-1}\right]_{1:p}' \end{bmatrix}' - \nu_k \tau_{k-1}}{\sqrt{1-\nu_k^2}} \\
\tau_k &= \sqrt{1-\nu_k^2}\tau_{k-1} - \nu_k t_k
\end{aligned}$$

where $[\cdot]_{k:p,j}$ selects the matrix elements in row k trough p in column j. Note that the k'th iteration introduces a zero in the k'th position of t_k, in the $k+1$'th position of τ_k, and in the first $k-1$ positions of $\left[C_{p,q+1}\right]_{1:p,k}$. This can be exploited to make the above recursion more efficient by only updating the non-zero elements in each iteration. Also note that the Cholesky factor of an AR-order p can be extracted from the Cholesky factor of the largest AR-order P as the upper left $p \times p$ submatrix.

Proof. The generalized Schur algorithm is derived in [40, Sec. 1.6.4]. The simpler version of the generalized Schur algorithm for the case of symmetric, positive definite matrices with a displacement rank of 2 w.r.t. to a lower triangular displacement operator is given in [41, Sec. 2.4]. The final result is then obtained from this simplified algorithm by using the Toeplitz displacement operator and the mixed downdating implementation of the hyperbolic rotations [39, App. B.3]. □

The results above show how the F$_0$-AR-ML-E algorithm can evaluate the residual mean square $\widehat{\sigma}_{p,q}^2$ for all orders p and q. In the harmonically related case, pseudo-code for the F$_0$-AR-ML-E algorithm is outlined in algorithm 1. The total complexity of the algorithm is in the order of $\mathcal{O}(F \log_2 F) + \mathcal{O}(FP(P+Q))$, assuming that $P \ll \log_2 F$, where F is typically selected around $5TQ$ [17, 42].

4.2. The F$_0$-AR-ML-A1 Algorithm

The F$_0$-AR-ML-A1 algorithm is based on the asymptotic result in (12) to make the approximation

$$E_q' E_q \approx T_P I_{2q}/2.$$

Algorithm 1 Pseudo code for the F$_0$-AR-ML-E algorithm with $\omega^{(f)} = 2\pi f/F$ and $\varpi = \begin{bmatrix}\omega^{(0)} & \cdots & \omega^{(F-1)}\end{bmatrix}'$. The functions `gsa` and `cdd` refer to the generalized Schur algorithm and the Cholesky downdating in Theorem 7 and Lemma 6, respectively. Note that $\lambda_{P,q} = \begin{bmatrix}\widetilde{\lambda}'_{0,q} & \widetilde{\lambda}'_{1,q} & \cdots & \widetilde{\lambda}'_{P,q}\end{bmatrix}'$ and $\rho_{P,q} = \begin{bmatrix}\widetilde{\rho}_{1,q} & \cdots & \widetilde{\rho}_{P,q}\end{bmatrix}'$.

```
$\Gamma_q^{(f)}$ has been pre-computed for all candidate fundamental frequencies and orders (see Theorem 2).
$\widehat{\sigma}^2_{0,0} = T^{-1}X'_P X_P$                                          ▷ $\mathcal{O}(T)$
$\rho_{P,0} = T^{-1}Z'_P X_P$                                                    ▷ $\mathcal{O}(TP)$
$\{\widehat{\sigma}^2_{p,0}\}_{p=1}^{P}, C_{P,0} = \text{gsa}(\widehat{\sigma}^2_{0,0}, \rho_{P,0})$            ▷ $\mathcal{O}(P^2)$
$\widehat{X} = \text{fft}(X, F)$                                                  ▷ $\mathcal{O}(F\log_2 F)$
$\widehat{Y} = \text{diag}(\widehat{X})e^{-i\varpi\left(\begin{bmatrix}0 & \cdots & P\end{bmatrix}+t_0\right)}$                          ▷ $\mathcal{O}(FP)$
for $f = 1, 2, \ldots, \lfloor F/2 \rfloor$ do
    for $q = 1, 2, \ldots, Q$ do
        if $f < \lceil F/(2q) \rceil$ then                                   ▷ Make $\omega^{(f)} \in (0, \pi/q)$
            $\lambda_{P,q} = \sum_{j=1}^{q}\left[(\Gamma_q)_{1,2j-1}\,\text{Re}(\widehat{Y}^{(jf)})' \quad -(\Gamma_q)_{2,2j}\,\text{Im}(\widehat{Y}^{(jf)})'\right]$   ▷ $\mathcal{O}(qP)$
            $C_{P,q} = \text{cdd}\left(C_{P,q-1}, (\lambda_{P,q})_{2:P+1,1}\right)$                 ▷ $\mathcal{O}(P^2)$
            $C_{P,q} = \text{cdd}\left(C_{P,q}, (\lambda_{P,q})_{2:P+1,2}\right)$                   ▷ $\mathcal{O}(P^2)$
            $\widehat{\sigma}^2_{0,q} = \widehat{\sigma}^2_{0,q-1} - \widetilde{\lambda}_{0,q}\widetilde{\lambda}'_{0,q}$                            ▷ $\mathcal{O}(1)$
            for $p = 1, \ldots, P$ do                              ▷ Perform forward substitution
                $\widetilde{\rho}_{p,q} = \widetilde{\rho}_{p,q-1} - \widetilde{\lambda}_{p,q}\widetilde{\lambda}'_{0,q}$                           ▷ $\mathcal{O}(1)$
                $\widetilde{\eta}_{p,q} = \frac{\widetilde{\rho}_{p,q} - (C_{P,q})_{p,1:p-1}\eta_{p-1,q}}{(C_{P,q})_{p,p}}$                       ▷ $\mathcal{O}(p)$
                $\widehat{\sigma}^2_{p,q} = \widehat{\sigma}^2_{p-1,q} - \widetilde{\eta}^2_{p,q}$                              ▷ $\mathcal{O}(1)$
                $\eta_{p,q} = \begin{bmatrix}\eta'_{p-1,q} & \widetilde{\eta}_{p,q}\end{bmatrix}'$                          ▷ $\mathcal{O}(1)$
            end for
        end if
    end for
end for
```

Making this approximation leads to a lower computational complexity for two reasons. First, the matrix $\Sigma_{p,q}$ becomes a Toeplitz matrix which means that we can compute the error term $\rho'_{p,q}\Upsilon^{-1}_{p,q}\rho_{p,q}$ efficiently using the the Levinson-Durbin algorithm described in the next theorem.

Theorem 8. *The residual mean square* $\widehat{\sigma}^2_{p,0}$ *can be computed recursively for* $p = 0, \ldots, P-1$ *using the Levinson-Durbin algorithm*

$$\begin{aligned}
\nu_{p+1} &= \widehat{\sigma}^{-2}_{p,0}\begin{bmatrix}\widehat{\beta}'_{p,0}J_p & -1\end{bmatrix}\rho_{p+1,0} \\
\widehat{\beta}_{p+1,0} &= \begin{bmatrix}\widehat{\beta}_{p,0} \\ 0\end{bmatrix} + \nu_{p+1}\begin{bmatrix}J_p\widehat{\beta}_{p,0} \\ -1\end{bmatrix} \\
\widehat{\sigma}^2_{p+1,0} &= \widehat{\sigma}^2_{p,0}(1-\nu^2_{p+1})
\end{aligned}$$

where J_p *is the* p*-dimensional exchange matrix, i.e.,*

$$J_p = \begin{bmatrix}0 & & 1 \\ & \iddots & \\ 1 & & 0\end{bmatrix},$$

and $\widehat{\beta}_{0,0}$ *is an empty vector.*

Proof. See [38, Sec. 4.7.3]. □

The second reason is the fact that the solution to the data independent step in (26) becomes $\zeta_q = \begin{bmatrix}0' & I_2\end{bmatrix}'$. Consequently, $\lambda_{p,q}$ in Theorem 2 will simplify to

$$\lambda_{p,q} = \begin{bmatrix}X'_P \\ Z'_p\end{bmatrix}\widetilde{E}_q.$$

In total, the computational complexity drops to $\mathcal{O}(F\log_2 F) + \mathcal{O}(FP^2)$ which is essentially the cost of computing a single FFT and that of running the Levinson-Durbin algorithm for all candidate frequencies. The pseudo-code for the F$_0$-AR-ML-A1 algorithm is given in algorithm 2. Whereas the approximation in (12) has a positive impact on the computational complexity, it has a negative impact on the estimation accuracy in some scenarios. We investigate this through experiments in Sec. 5.

4.3. The F$_0$-AR-ML-A2 Algorithm

While the two fast algorithms described so far are all based on the expression of the residual mean square in (15), it is also possible to derive a fast algorithm, named F$_0$-AR-ML-A2, based on the expression in (16), provided that the asymptotic approximation

Algorithm 2 Pseudo code for the F$_0$-AR-ML-A1 algorithm with $\omega^{(f)} = 2\pi f/F$ and $\varpi = \begin{bmatrix}\omega^{(0)} & \cdots & \omega^{(F-1)}\end{bmatrix}'$. The function `lda` refers to the Levinson-Durbin algorithm in Theorem 8. Note that the loop over the frequency bin index f can be vectorised in many programming languages.

```
$\widehat{\sigma}^2_{0,0} = T^{-1} X_P' X_P$                                          ▷ $\mathcal{O}(T)$
$\rho_{P,0} = T^{-1} Z_P' X_P$                                                        ▷ $\mathcal{O}(TP)$
$\{\widehat{\sigma}^2_{p,0}\}_{p=1}^{P} = \text{lda}(\widehat{\sigma}^2_{0,0}, \rho_{P,0})$                  ▷ $\mathcal{O}(P^2)$
$\widehat{\phi} = 2\,|\text{fft}(X, F)|^2 / (T T_P)$                                  ▷ $\mathcal{O}(F \log_2 F)$
$\widehat{\Phi} = \cos\left(\begin{bmatrix}1 & \cdots & P\end{bmatrix}' \varpi'\right) \text{diag}(\widehat{\phi})$            ▷ $\mathcal{O}(FP)$
for $f = 1, 2, \ldots, \lfloor F/2 \rfloor$ do
    for $q = 1, 2, \ldots, Q$ do
        if $f < \lceil F/(2q) \rceil$ then                                  ▷ Make $\omega^{(f)} \in (0, \pi/q)$
            $\widehat{\sigma}^2_{0,q} = \widehat{\sigma}^2_{0,q-1} - \widehat{\phi}^{(qf)}$                        ▷ $\mathcal{O}(1)$
            $\rho_{P,q} = \rho_{P,q-1} - \widehat{\Phi}^{(qf)}$                            ▷ $\mathcal{O}(P)$
            $\{\widehat{\sigma}^2_{p,q}\}_{p=1}^{P} = \text{lda}(\widehat{\sigma}^2_{0,q}, \rho_{P,q})$          ▷ $\mathcal{O}(P^2)$
        end if
    end for
end for
```

in (12) is made. As the naming suggests, the F$_0$-AR-ML-A2 algorithm will produce the exact same estimates as the F$_0$-AR-ML-A1 algorithm, but will have a complexity in the order of $\mathcal{O}(F \log_2 F) + \mathcal{O}(FPQ)$ whose last terms is an order Q/P that of the F$_0$-AR-ML-A1 algorithm. Thus, the complexities suggest that one should prefer F$_0$-AR-ML-A2 over F$_0$-AR-ML-A1 if P is larger than Q and vice versa. Unfortunately, however, our simulations suggest that the complexity constants omitted in the big-$\mathcal{O}$ notation are so large for the F$_0$-AR-ML-A2 algorithm that the computation time always seems to be larger than that of the F$_0$-AR-ML-A1 algorithm for typical values of P and Q. Therefore, we here only give the pseudo-code for the algorithm in algorithm 3 and refer the interested reader to the appendix for a derivation of the algorithm. For easy comparison, we have listed the order of complexities of the three fast algorithms in Table 1.

$\mathrm{F_0}$-AR-ML-E	$\mathrm{F_0}$-AR-ML-A1	$\mathrm{F_0}$-AR-ML-A2
$\mathcal{O}(F \log_2 F) + \mathcal{O}(FP(P+Q))$	$\mathcal{O}(F \log_2 F) + \mathcal{O}(FP^2)$	$\mathcal{O}(F \log_2 F) + \mathcal{O}(FPQ)$

Table 1: Order of complexities of the three fast algorithms.

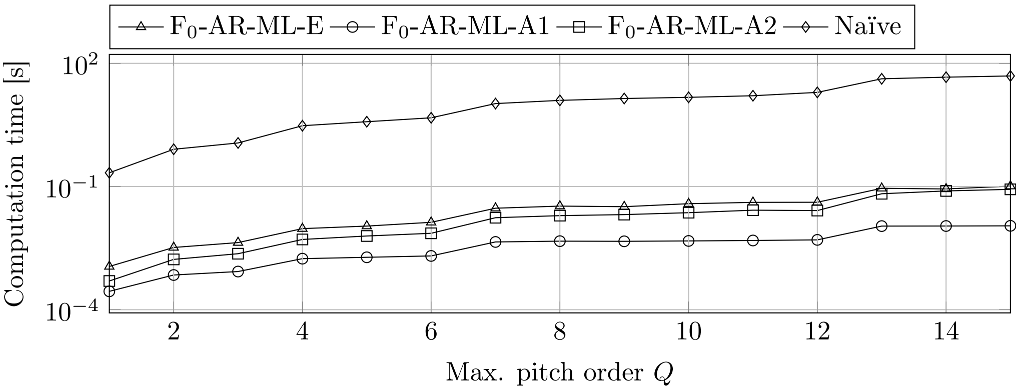


Figure 2: The computation time in seconds as a function of the maximum harmonic order Q. The data length was $T = 512$ and the maximum AR-order was $P = 3$. Note that the non-smooth increases in computation time in Fig. 2 is caused by the fact that the number of frequencies F given in (36) depends nonlinearly on Q.

5. Experimental Results

In this section, we present a number of experimental results. As the main contribution is fast algorithms, we first measure the computation time of the different algorithms in different settings and compare these to the computation time of a naïve implementation of the estimator. Secondly, we assess how the approximations made in the two approximate estimators affect the estimation accuracy. Thirdly, we show in terms of both computation time and estimation accuracy that it is advantageous to jointly estimate the sinusoidal and autoregressive parameters instead of just employing a simpler iterative estimator. Finally, we give two application examples on real data to demonstrate the benefit of modelling the noise using an AR-process instead of just making the traditional WGN assumption. All simulations have been run in MATLAB™ R2020a on a 64 bit Ubuntu Linux 16.04.6 computer with Linux kernel 4.15.0.101. The code for the estimators and

Algorithm 3 Pseudo code for the F_0-AR-ML-A2 algorithm with $\omega^{(f)} = 2\pi f/F$ and $\varpi = \begin{bmatrix}\omega^{(0)} & \cdots & \omega^{(F-1)}\end{bmatrix}'$. The function `lda` refers to the Levinson-Durbin algorithm in Theorem 8. Note that the loop over the frequency bin index f can be vectorised in many programming languages.

```
$\widehat{\sigma}^2_{0,0} = T^{-1}X'_P X_P$                                        ▷ $\mathcal{O}(T)$
$\rho_{P,0} = T^{-1}Z'_P X_P$                                          ▷ $\mathcal{O}(TP)$
$\{\widehat{\sigma}^2_{p,0}\}_{p=1}^P, \{\nu_p\}_{p=1}^P = \mathrm{lda}(\widehat{\sigma}^2_{0,0}, \rho_{P,0})$          ▷ $\mathcal{O}(P^2)$
$\widehat{X} = T^{-1}\,\mathrm{diag}(\mathrm{fft}(X, F))e^{-i\varpi t_0}$                      ▷ $\mathcal{O}(F\log_2 F)$
for $f = 1, 2, \dots, \lfloor F/2 \rfloor$ do
    for $q = 1, 2, \dots, Q$ do
        if $f < \lceil F/(2q) \rceil$ then                        ▷ Make $\omega^{(f)} \in (0, \pi/q)$
            $\widetilde{g}_{0,q} = \widehat{X}^{(fq)}$                              ▷ $\mathcal{O}(1)$
            $\widetilde{h}_{0,q} = \widetilde{W}_q\widetilde{g}_{0,q}$                            ▷ $\mathcal{O}(1)$
            $\widehat{\sigma}^2_{0,q} = \widehat{\sigma}^2_{0,q-1} - 2|\widetilde{g}_{0,q}|^2 T/T_P$              ▷ $\mathcal{O}(1)$
            $\widehat{d}_{0,q} = 2g_{0,q}T/T_P$                            ▷ $\mathcal{O}(1)$
            $\mu_{0,q} = 2h_{0,q}T/T_P$                            ▷ $\mathcal{O}(1)$
            for $p = 1, 2, \dots, P$ do
                $\kappa_{p,q} = \frac{\nu_p\widehat{\sigma}^2_{p-1,0} - h'_{p-1,q}\widehat{d}_{p-1,q}}{\widehat{\sigma}^2_{p-1,q}}$            ▷ $\mathcal{O}(q)$
                $\widehat{\sigma}^2_{p,q} = \widehat{\sigma}^2_{p-1,q}(1 - \kappa^2_{p,q})$                  ▷ $\mathcal{O}(1)$
                if $p < P$ then
                    $\widetilde{g}_{p,q} = \widetilde{g}_{p-1,q} - \nu_p\widetilde{h}_{p-1,q}$            ▷ $\mathcal{O}(1)$
                    $\widetilde{h}_{p,q} = \widetilde{W}_q(\widetilde{h}_{p-1,q} - \nu_p\widetilde{g}_{p-1,q})$       ▷ $\mathcal{O}(1)$
                    $\widehat{d}_{p,q} = \widehat{d}_{p-1,q} - \kappa_{p,q}\mu_{p-1,q}$          ▷ $\mathcal{O}(q)$
                    $\mu_{p,q} =$
                        $W_q(\mu_{p-1,q} - \kappa_{p,q}\widehat{d}_{p-1,q})$        ▷ $\mathcal{O}(q)$
                end if
            end for
        end if
    end for
end for
```

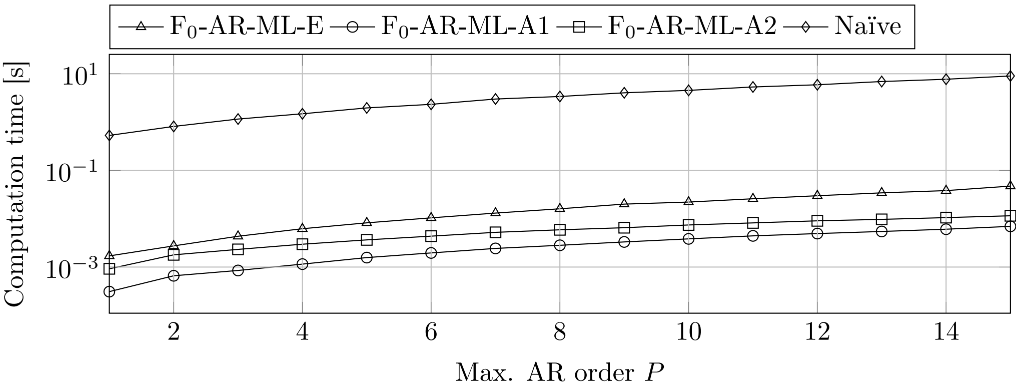


Figure 3: The computation time in seconds as a function of the maximum AR order P. The data length was $T = 512$ and the harmonic order was $Q = 3$.

for generating all results can be found at `https://github.com/jkjaer/fastF0ArMl`. In all experiments, we used

$$F = 2^{\lceil \log_2(5QT) \rceil} \tag{36}$$

frequency grid points where $\lceil \cdot \rceil$ denotes the ceiling operation.

5.1. Computation time

We estimated the computation time as a function of the maximum harmonic order Q, the maximum AR-order P, and the data length T. In all cases, the time for evaluating the residual mean square $\widehat{\sigma}^2_{p,q}$ for all candidate model orders were estimated by taking the minimum value of 50 Monte Carlo runs, each measuring the computation time using the `timeit()` function in MATLAB™. The estimated computation time should be seen as an upper bound on the computation time since it can only be overestimated on a multitasking operating system. Fig. 2, Fig. 3, and Fig. 4 all show the same trend; the F_0-AR-ML-A1 was the fastest algorithm, the F_0-AR-ML-A2 was the second fastest algorithm, and the F_0-AR-ML-E was the third fastest algorithm. For high AR-orders, Fig. 3 shows that the F_0-AR-ML-A2 algorithm approached, but never outperformed, the computation time of the F_0-AR-ML-A1 algorithm, despite being an order of complexity lower in terms of the AR-order P. The fast algorithms all significantly outperformed

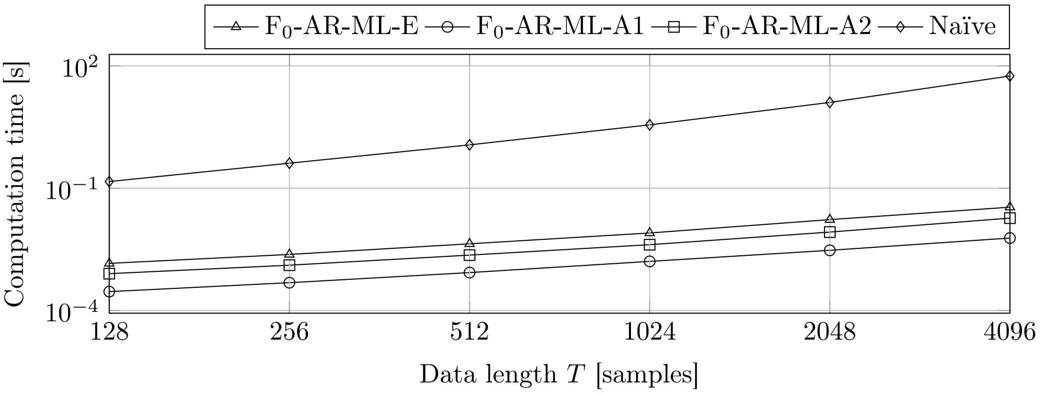


Figure 4: The computation time in seconds as a function of the data length T. The AR-order was $P = 3$ and the harmonic order was $Q = 3$.

the naïve algorithm by at least a factor of 100. The naïve implementation is simply a straight-forward implementation of the expression in (9).

5.2. Estimation accuracy

The estimation accuracy was evaluated as a function of a pre-whitened signal-to-noise ratio (SNR) and the time-frequency resolution using 10,000 Monte Carlo runs for each configuration. We here use the SNR of the pre-whitened data instead of the global SNR since the local SNR at the frequencies of the harmonic components is what influences the performance of the estimator (see (3)). In each Monte Carlo run, we generated a periodic function with random fundamental frequency from $2\pi(f_{0,\mathrm{MIN}}, 0.4)$, a harmonic-order of $q = 6$, random phases, and exponentially decreasing amplitudes. The generated AR-noise was of order $p = 3$ where the AR-coefficients were generated by selecting one real root and a complex conjugate root pair from a disc in the complex plane with a minimum radius of 0.5 and a maximum radius of 0.9. The excitation variance was computed by 1) pre-whitening the generated periodic function with an FIR-filter having the generated AR-parameters as its coefficients, 2) adding white Gaussian noise with a variance equal to the excitation variance resulting in the desired SNR of the pre-whitened periodic function, and 3) post-filtering the noisy pre-whitened periodic function with an all-pole filter having the generated AR-parameters as its coefficients.

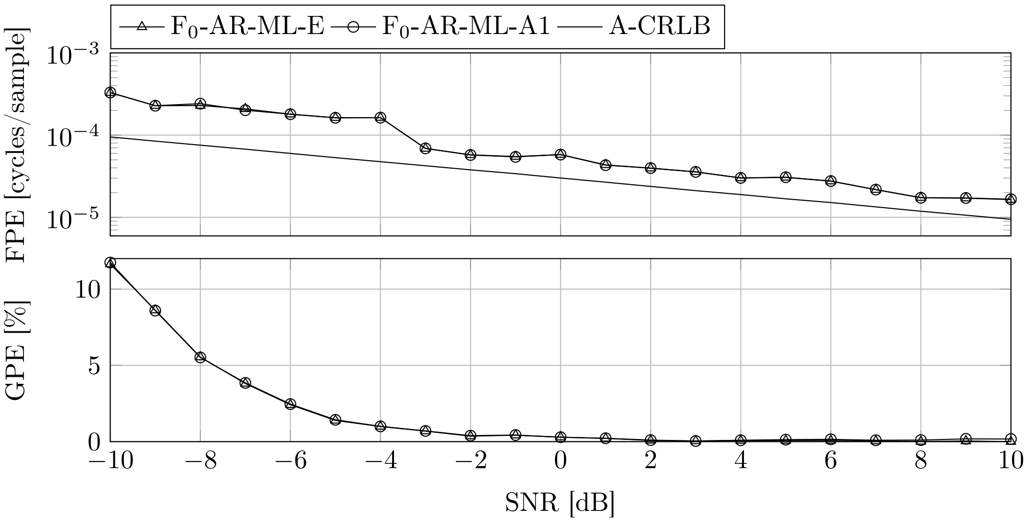


Figure 5: The FPE and GPE as a function of the SNR. The maximum orders were set to $Q = 6$ and $P = 3$, and the data length was $T = 512$.

For each SNR and value for $f_{0,\text{MIN}}$, the fundamental frequency estimates from 10,000 Monte Carlo runs were divided into fine pitch errors (FPE) and gross pitch errors (GPE). These metrics are often used in speech processing (see [2, Ch. 10] for an in-depth discussion) which also explains why the term pitch is used instead of fundamental frequency. The FPE is the root mean squared error of all fundamental frequency estimates within 20 % of the true value[3]. The GPE is then the proportion of estimates that falls outside this range. The main reason for dividing the estimation errors into these two metrics is that the subharmonic errors discussed in the introduction appear as systematic outliers in the computed estimates. The GPE metric is used to quantify how often these outliers occur whereas the FPE metric measures the estimation accuracy of an outlier-free estimator.

Fig. 5 shows the computed results for two of the fast algorithms together with the asymptotic Cramér-Rao lower bound (CRLB) given by (3), averaged over the 10,000 Monte Carlo runs for each SNR. We did not include the F$_0$-AR-ML-A2 estimator in the experiment since it produces the same estimates as the F$_0$-AR-ML-A1 estimator.

[3] Note that this threshold seems to vary between references. Sometimes a threshold of 10 % or an absolute threshold is used instead.

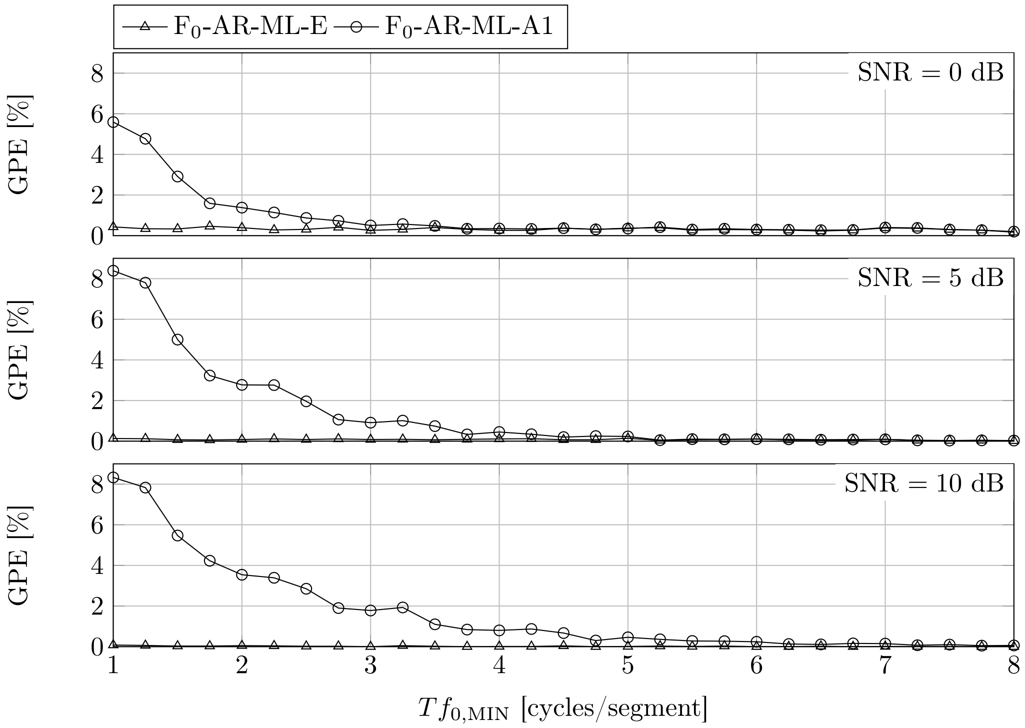


Figure 6: The GPE as a function of the lower bound on the fundamental frequency for three different pre-whitened SNRs. The maximum orders were set to $Q = 6$ and $P = 3$.

For this experiment, the value for $f_{0,\mathrm{MIN}}$ was set to $6/T$ which means that at least six cycles of the periodic function is in a segment. From the figure, we see that the exact and approximate estimators had the same performance, except for SNRs close to 10 dB where the GPE of the approximate F$_0$-AR-ML-A1 estimator increased slightly. This is caused by the approximation made in the estimator as shown in the next experiment. Both methods had a small gap to the asymptotic CRLB which we believe is due to employing the autocorrelation method for windowing the data. This suspicion is confirmed by the fact that both estimators attain the asymptotic CRLB in the WGN case.

To quantify how the asymptotic approximation based on (12) affects the performance of the F$_0$-AR-ML-A1 and F$_0$-AR-ML-A2 estimators, we evaluated the estimation accuracy as a function of the minimum value of the lower bound on the fundamental frequency at different SNRs. This is essentially the same as benchmarking the time-frequency res-

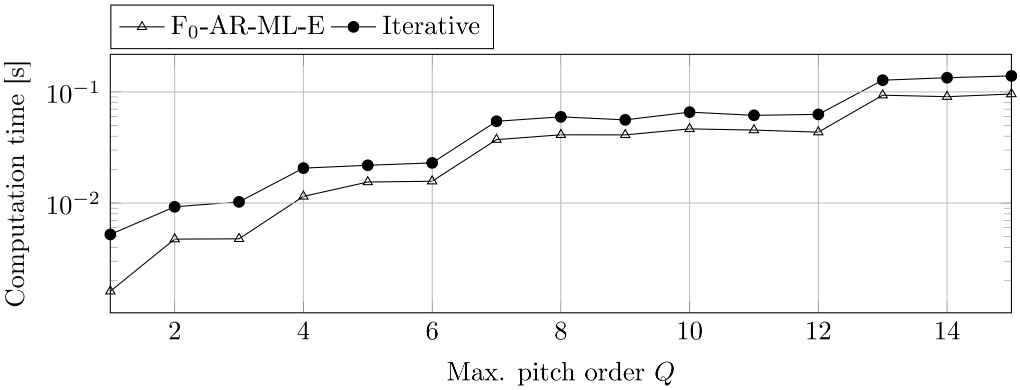


Figure 7: The computation time in seconds as a function of the maximum harmonic order Q. The data length was $T = 512$ and the AR-order was $P = 3$.

olution of the estimators. Fig. 6 shows the computed GPEs for pre-whitened SNRs of 0 dB, 5 dB, and 10 dB. The figure shows that the time-frequency resolution of the F$_0$-AR-ML-A1 estimator decreased with an increasing SNR. This is explained by the fact that the residual mean square $\widehat{\sigma}^2_{p,q}$ becomes increasingly sensitive to perturbations in the projection matrices as the SNR increases. Consequently, the F$_0$-AR-ML-A1 and F$_0$-AR-ML-A2 estimators should only be used in low SNR conditions and when the segment length is sufficiently long relative to what the minimum expected fundamental frequency is. We also remark that the threshold for when the approximate estimators start producing outliers depends on other factors than the SNR, including the number of sinusoidal and AR components and how close the magnitude of the poles are to one.

5.3. Joint versus iterative estimation

To compare the proposed joint estimation of the sinusoidal and AR-parameters with an iterative approach, we computed both the computation time as a function of the harmonic order and the estimation accuracy as a function of the SNR, using the same setup as in the previous two experiments, except for that $f_{0,\text{MIN}}$ was set to $2/T$. The iterative approach consists of a traditional linear predictor and the F$_0$-AR-ML-E estimator with an order of $P = 0$ (i.e., assuming white noise). In each iteration, the linear predictor first computes an estimate of the AR-parameters, including the order, from the data vector

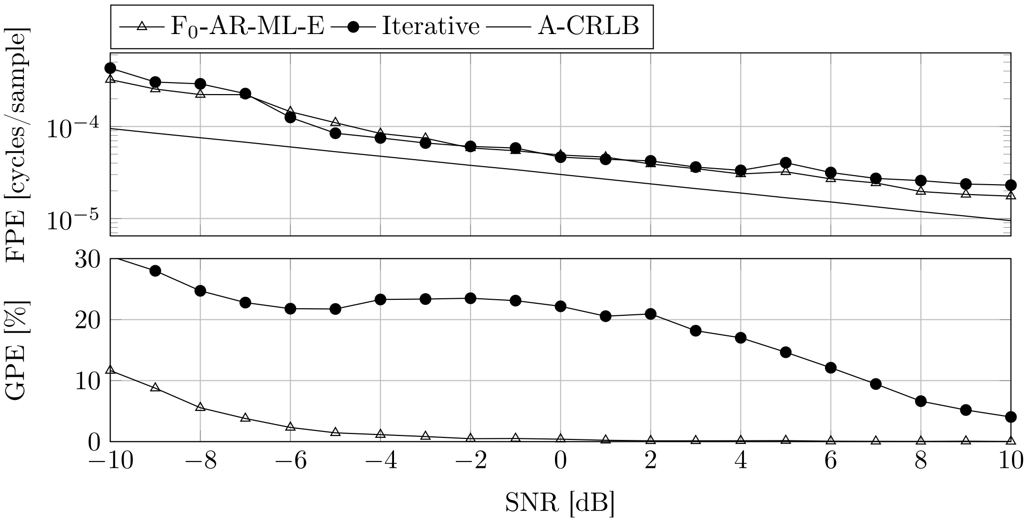


Figure 8: The FPE and GPE as a function of the SNR. The maximum orders were set to $Q = 6$ and $P = 3$, and the data length was $T = 512$.

subtracted by an estimate of the periodic function. The prediction of the AR-signal is then subtracted from the data vector which is then used as input to the fundamental frequency estimator that will estimate all harmonic parameters, including the order, and reconstruct the periodic function. In the experiment, we repeated this procedure 10 times. Fig. 7 and Fig. 8 show that, compared to the F_0-AR-ML-E algorithm, the iterative approach was both slower and produced many outliers compared to the joint approach. The FPE is more or less the same for the two approaches. The estimation accuracy of the iterative approach can be improved by running more iterations, but this will also make the algorithm slower.

5.4. Application example I: order tracking analysis

Order tracking analysis [4, 5] is concerned with finding structural resonances excited by a rotating machine. During testing, the machine is often accelerated (run-up) and/or decelerated (coast-down) while the vibrational or acoustical response is recorded. To avoid time-frequency smearing, the recorded response is resampled uniformly in rotation angle instead of in time. The resampling requires that the rotation speed of the engine is

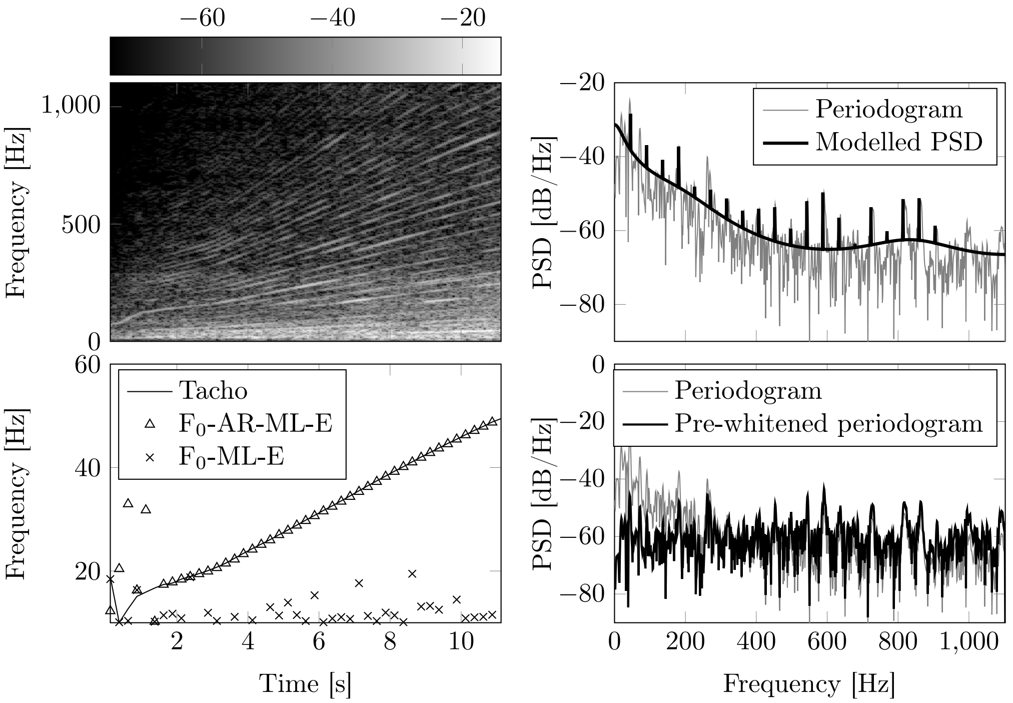


Figure 9: A recording of an engine run-up (upper-left panel) and the estimates of the engine rotation speed (lower left panel). The upper right panel shows the periodogram and the estimated PSD for one segment around 10 seconds. The lower right panel shows the periodogram and the periodogram obtained after pre-whitening the data with the estimated AR-parameters. The F0-ML-E algorithm is the F_0-AR-ML-E algorithm with an AR-order of $P = 0$.

estimated accurately, and this is often referred to as tracking. The tracking is typically performed by mounting a tachometer on the engine axle from which the rotation speed can easily be extracted. However, mounting a tachometer might be difficult and costly for which reason it is desirable to estimate the rotation speed directly from the recorded vibrational or acoustical response(s). This is often referred to as autotracking and what we here focus on.

The upper left panel of Fig. 9 shows a spectrogram of a car engine run-up. The harmonic structure can clearly be seen from the upper part of the spectrogram, but the fundamental components are completely hidden in wind and tyre noise also present in the recording. The lower left panel of Fig. 9 shows the fundamental frequency estimated from the tachometer signal as well as by the F_0-AR-ML-E and the F0-ML-E algorithms. The latter is identical to the first, with the exception that the maximum AR-order is set to $P = 0$, i.e., white noise is assumed. Both algorithms were set-up with a minimum harmonic order of 20 and a maximum harmonic order of $Q = 25$. Moreover, the fundamental frequency was constrained to be in the interval $(10 \text{ Hz}, 100 \text{ Hz})$ and the maximum AR-order of the F_0-AR-ML-E algorithm was set to $P = 5$. The data were resampled to a sampling frequency of 2205 Hz and the data length was set to 250 ms. The lower left panel of Fig. 9 shows that F_0-AR-ML-E algorithm worked very well as an autotracker, and that the F0-ML-E algorithm completely broke down due to the violation of the white noise assumption. The right panels in Fig. 9 show an example of the periodogram at around the 10 second mark overlayed with the modelled PSD (top) and the pre-whitened PSD (bottom). The pre-whitening should be understood in the context of Fig. 1b where $\widehat{u}_{p,0}$ is the pre-whitened signal we compute the periodogram of in the lower right panel of Fig. 9.

5.5. Application example II: speech in wind noise

Speech recordings are often contaminated by some background noise, and we here focus on the case where the background noise is wind. In speech enhancement [43], the goal is to remove as much of the background noise as possible while altering the speech as little as possible. Typically, the enhancement is performed by designing a soft time-frequency mask (often a Wiener mask) that applies a small weight (close to zero) to the noisy parts of the data spectrogram and a large weight (close to one) to the clean parts of the data spectrogram. To calculate the mask, estimates of the speech and/or the noise spectra are required and these can, for voiced speech segments, be estimated with the joint fundamental frequency and AR coefficient estimator.

The upper left panel of Fig. 10 shows a spectrogram of a speech signal in wind noise at an SNR of 5 dB. The speech signal is taken from the TSP speech database [44] and the wind noise is taken from a wind noise database [45]. We again compare the F_0-AR-ML-E and F0-ML-E algorithms. As we do not have access to ground truth values of the fundamental frequency, we also ran the F_0-AR-ML-E on the clean speech which resulted in the solid line shown in the lower left panel of Fig. 10. This panel also shows the estimates produced by the F_0-AR-ML-E and F0-ML-E algorithms. We again see that modelling the wind noise using a low-order AR-process increased the robustness of

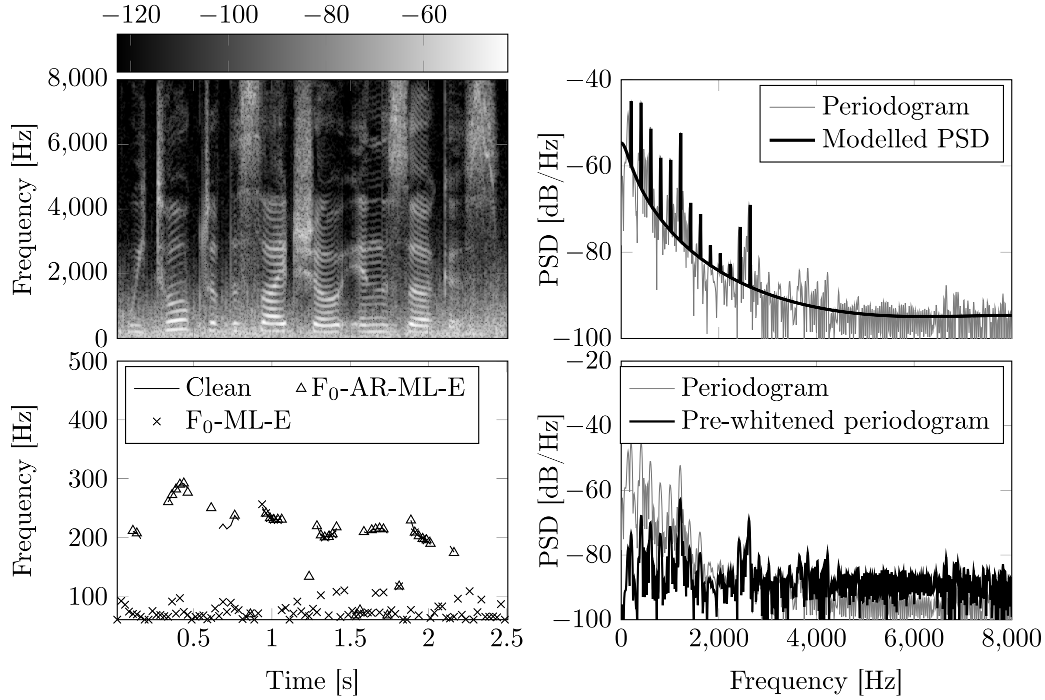


Figure 10: A recording of a speech signal in wind noise at an SNR of 5 dB (upper-left panel) and the estimates of the fundamental frequency (lower left panel). The upper right panel shows the periodogram and the estimated PSD for one segment around 1.4 second. The lower right panel shows the periodogram and the periodogram obtained after pre-whitening the data with the estimated AR-parameters. The F0-ML-E algorithm is the F_0-AR-ML-E algorithm with an AR-order of $P = 0$.

the fundamental frequency estimator significantly. As in the previous experiment, the F_0-AR-ML-E and F0-ML-E algorithms were set-up in the same way, with the exception that we used $P = 0$ for the F0-ML-E algorithm. The maximum harmonic order was set to $Q = 15$, and all lower orders were considered, except for $q = 1, 2$ which are typically not observed for speech signals. The fundamental frequency was constrained to be in the interval $(60\text{ Hz}, 400\text{ Hz})$, the sampling frequency was 16 kHz, the data length was set to 25 ms, and the maximum AR-order was $P = 10$. The right-most panels in Fig. 10 shows the periodiogram, the modelled PSD, and the pre-whitened periodogram around the 1 second mark.

6. Conclusion

In this paper, we have derived a joint maximum likelihood estimator and three fast algorithms for joint fundamental frequency and AR-parameter estimation under model uncertainty. One of these algorithms evaluates the residual mean square exactly, but also has the highest computational complexity. The other two fast algorithms are faster, but only compute approximations to the residual mean square. All three fast algorithms are asymptotically equivalent. Through experiments and for finite data lengths, we showed that the two approximate algorithm are accurate provided the data length is sufficiently long relative to the smallest expected fundamental frequency. If this condition is violated, the approximate algorithms produced more outliers. The exact algorithm, on the other hand, worked well, even for short data length, but was also approximately a factor of ten slower than the fastest approximate algorithm called F$_0$-AR-ML-A1. We also showed that performing the sinusoidal and autoregressive parameter estimation jointly with the exact fast algorithm called F$_0$-AR-ML-E algorithm is both faster and more accurate than performing the estimation iteratively. Finally, we applied the exact algorithm to real-world data to show examples of the benefit of doing fundamental frequency estimation using an estimator allowing for autoregressive noise instead of making the convinient and simpler WGN assumption. These examples clearly showed that the number of ourliers is dramatically reduced when the estimator allows for autoregressive noise.

Appendix A. Derivation of the F$_0$-AR-ML-A2 Algorithm

As alluded to in Sec. 4, the F$_0$-AR-ML-A2 algorithm is based on the second expression for the residual error variance $\widehat{\sigma}^2_{p,q}$ given in (16). Since the first term in the expression, i.e., $\widehat{\sigma}^2_{p,0}$, can be computed using the Levinson-Durbin algorithm described in Theorem 8, we will in this section focus on how the second term in the expression, i.e., $g'_{p,q}\Omega^{-1}_{p,q}g_{p,q}$, can be computed efficiently. We first focus on a recursive update of $\Omega_{p,q}$.

Lemma 9. *The $2q \times 2q$ matrix $\Omega_{p+1,q}$ can be recursively updated using*

$$\Omega_{p+1,q} = \Omega_{p,q} - \widehat{\sigma}^{-2}_{p,0} h_{p,q} h'_{p,q},$$

where

$$h_{p,q} = T^{-1}E_q'(\widetilde{z}_{p+1} - Z_p J_p \widehat{\beta}_{p,0}) = T^{-1}E_q' P_{Z_p}^{\perp} \widetilde{z}_{p+1},$$

and $\widetilde{z}_{p+1}$ *is the last column of* Z_{p+1}*.*

Proof. First note that

$$(Z_{p+1}'Z_{p+1})^{-1} = \begin{bmatrix} (Z_p'Z_p)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\widetilde{z}_{p+1}' P_{Z_p}^{\perp} \widetilde{z}_{p+1}} \begin{bmatrix} Z_p^{+}\widetilde{z}_{p+1} \\ -1 \end{bmatrix} \begin{bmatrix} \widetilde{z}_{p+1}' \left(Z_p^{+}\right)' & -1 \end{bmatrix}.$$

This leads to

$$P_{Z_{p+1}} = \begin{bmatrix} Z_p & \widetilde{z}_{p+1} \end{bmatrix} (Z_{p+1}'Z_{p+1})^{-1} \begin{bmatrix} Z_p' \\ \widetilde{z}_{p+1}' \end{bmatrix} = P_{Z_p} + \frac{P_{Z_p}^{\perp} \widetilde{z}_{p+1} \widetilde{z}_{p+1}' P_{Z_p}^{\perp}}{\widetilde{z}_{p+1}' P_{Z_p}^{\perp} \widetilde{z}_{p+1}}.$$

Since $Z_p'Z_p$ is a Toeplitz matrix, we have from the Levinson-Durbin algorithm that

$$\begin{aligned} J_p \widehat{\beta}_{p,0} &= (Z_p'Z_p)^{-1} Z_p' \widetilde{z}_{p+1} \\ \widehat{\sigma}_{p,0}^2 &= \widetilde{z}_{p+1}' P_{Z_p}^{\perp} \widetilde{z}_{p+1}. \end{aligned}$$

The final result then follows by inserting the above expressions in the definition of $\Omega_{p+1,q}$ in (21), i.e.,

$$\Omega_{p+1,q} = T^{-1}E_q'E_q - T^{-1}E_q' P_{Z_{p+1}} E_q.$$

□

Next, we find a recursive expression for $g_{p,q}$ in (20).

Lemma 10. *The* $2q$ *dimensional vector* $g_{p+1,q}$ *can be recursively updated as*

$$g_{p+1,q} = g_{p,q} - \nu_{p+1} h_{p,q}$$

where $g_{0,q} = T^{-1}E_q'X_P$*, and* ν_{p+1} *and* $h_{p,q}$ *are defined in Theorem 8 and Lemma 9, respectively.*

Proof. From the definition of $g_{p,q}$ in (20), it follows that

$$\begin{aligned} g_{p+1,q} &= T^{-1}E_q'X_P - T^{-1}E_q' Z_{p+1} (Z_{p+1}'Z_{p+1})^{-1} Z_{p+1}' X_P \\ &= T^{-1}E_q'X_P - T^{-1}E_q' \begin{bmatrix} Z_p & \widetilde{z}_{p+1} \end{bmatrix} \widehat{\beta}_{p+1,0}. \end{aligned}$$

The result then follows by inserting the recursive expression for $\widehat{\beta}_{p+1,0}$ from Theorem 8 and by using the definition of $h_{p,q}$ from Lemma 9. □

We now have recursive expressions for both $\Omega_{p,q}$ and $g_{p,q}$. The next result shows how these can be used in a recursive expression for $\widehat{d}_{p,q} = \Omega_{p,q}^{-1} g_{p,q}$.

Lemma 11. *The linear parameters $\widehat{d}_{p,q} = \Omega_{p,q}^{-1} g_{p,q}$ can be recursively computed for $p = 0, \ldots, P-1$ as*

$$\widehat{d}_{p+1,q} = \widehat{d}_{p,q} - \kappa_{p+1,q} \Omega_{p,q}^{-1} h_{p,q},$$

where

$$\kappa_{p+1,q} = \frac{\nu_{p+1} \widehat{\sigma}_{p,0}^2 - h'_{p,q} \widehat{d}_{p,q}}{\widehat{\sigma}_{p,0}^2 - h'_{p,q} \Omega_{p,q}^{-1} h_{p,q}}. \tag{A.1}$$

Proof. From the recursive expression for $\Omega_{p+1,q}$ from Lemma 9 and the matrix inversion lemma, we obtain for $h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} \neq \widehat{\sigma}_{p,0}^2$ that

$$\Omega_{p+1,q}^{-1} = \left(\Omega_{p,q} - \widehat{\sigma}_{p,0}^{-2} h_{p,q} h'_{p,q} \right)^{-1} = \Omega_{p,q}^{-1} + \frac{\Omega_{p,q}^{-1} h_{p,q} h'_{p,q} \Omega_{p,q}^{-1}}{\widehat{\sigma}_{p,0}^2 - h'_{p,q} \Omega_{p,q}^{-1} h_{p,q}}.$$

By right multiplying this by the recursive expression for $g_{p+1,q}$ from Lemma 10, we obtain the final result. □

Based on the above results, a recursive expression for the term $g'_{p,q} \Omega_{p,q}^{-1} g_{p,q}$ in (16) can now be computed.

Corollary 12. *The mean square error of the estimated sinusoidal signal projected onto the null space of Z_p can be computed recursively for $p = 0, 1, \ldots, P-1$ as*

$$g'_{p+1,q} \widehat{d}_{p+1,q} = g'_{p,q} \widehat{d}_{p,q} + \kappa_{p+1,q}^2 \left(\widehat{\sigma}_{p,0}^2 - h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} \right) - \nu_{p+1}^2 \widehat{\sigma}_{p,0}^2.$$

Proof. By inserting the recursive expressions for $g_{p+1,q}$ and $\widehat{d}_{p+1,q}$ from Lemma 10 and 11, respectively, we obtain

$$\begin{aligned} g'_{p+1,q} \widehat{d}_{p+1,q} = {} & (g_{p,q} - \nu_{p+1} h_{p,q})' \, \widehat{d}_{p,q} - \kappa_{p+1,q} \left(g_{p,q} - \nu_{p+1} h_{p,q} \right)' \Omega_{p,q}^{-1} h_{p,q} \\ = {} & g'_{p,q} \widehat{d}_{p,q} - \nu_{p+1} h'_{p,q} \widehat{d}_{p,q} \\ & - \kappa_{p+1,q} \left\{ \widehat{d}'_{p,q} h_{p,q} - \nu_{p+1} \left(h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} - \widehat{\sigma}_{p,0}^2 + \widehat{\sigma}_{p,0}^2 \right) \right\} \\ = {} & g'_{p,q} \widehat{d}_{p,q} - \nu_{p+1} h'_{p,q} \widehat{d}_{p,q} - \kappa_{p+1,q} \left(\widehat{d}'_{p,q} h_{p,q} - \nu_{p+1} \widehat{\sigma}_{p,0}^2 \right) \\ & - \nu_{p+1} \left(\nu_{p+1} \widehat{\sigma}_{p,0}^2 - h'_{p,q} \widehat{d}_{p,q} \right) \\ = {} & g'_{p,q} \widehat{d}_{p,q} + \kappa_{p+1,q}^2 \left(\widehat{\sigma}_{p,0}^2 - h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} \right) - \nu_{p+1}^2 \widehat{\sigma}_{p,0}^2. \end{aligned}$$

□

From the result in Corollary 12 and the definition of $\kappa_{p+1,q}$ in (A.1), we see that we need recursions for $h_{p,q}$ and $h'_{p,q}\Omega^{-1}_{p,q}h_{p,q}$ as well. The next lemma will be important in that connection.

Lemma 13. *For a non-negative integer $j \leq p$, we have*

$$Q_{p,q}E'_q\widetilde{z}_j = E'_q\widetilde{z}_{p-j}$$

where $\widetilde{z}_0 = X_P$ and $Q_{p,q}$ is an involutory (i.e., symmetric and orthogonal) matrix given by

$$\begin{aligned} Q_{p,q} &= G'_qW^p_qS_qG^{-\prime}_q = W^p_qS_qG_qG^{-\prime}_q = W_qQ_{p-1,q} \qquad \text{(A.2)} \\ S_q &= I_q \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

The matrices G_q and W_q are both defined in Lemma 4.

Proof. From Lemma 4, it follows that

$$\begin{aligned} E'_q\widetilde{z}_{p-j} &= TG'_q\widetilde{\Xi}_{p-j,q} = TG'_qW^{p-j}_q\iota_q = TG'_qW^p_qW^{-j}_q\iota_q = TG'_qW^p_qS_qW^j_q\iota_q \\ &= TG'_qW^p_qS_q\widetilde{\Xi}_{j,q} = G'_qW^p_qS_qG^{-\prime}_qE'_q\widetilde{z}_j = Q_{p,q}E'_q\widetilde{z}_j. \end{aligned}$$

□

Lemma 13 can now be used to derive a simple recursion for $h_{p,q}$.

Lemma 14. *The vector $h_{p,q}$ can be updated recursively for $p = 0, 1, \ldots, P-1$ using*

$$h_{p+1,q} = W_q\left(h_{p,q} - \nu_{p+1}g_{p,q}\right)$$

with initial conditions $h_{0,q} = Q_{0,q}g_{0,q}$ and $g_{0,q} = T^{-1}E'_qX_P$.

Proof. From Lemma 13, we first obtain

$$Q_{p,q}E'_qZ_{p-1}J_{p-1} = E'_qZ_{p-1}.$$

Thus, since

$$\begin{aligned} g_{p,q} &= T^{-1}E'_qP^{\perp}_{Z_p}X_P \\ h_{p,q} &= T^{-1}E'_qP^{\perp}_{Z_p}\widetilde{z}_{p+1}, \end{aligned}$$

it follows that

$$h_{p,q} = Q_{p,q} g_{p,q}. \tag{A.3}$$

Consequently, we have from (A.2) that

$$h_{p+1,q} = Q_{p+1,q} g_{p+1,q} = W_q Q_{p,q} g_{p+1,q}.$$

Inserting the recursive expression for $g_{p+1,q}$ from Lemma 10 then gives

$$h_{p+1,q} = W_q \left(h_{p,q} - \nu_{p+1} Q_{p,q} h_{p,q} \right).$$

The final result then follows since $Q_{p,q} h_{p,q} = Q_{p,q}^{-1} h_{p,q} = g_{p,q}$. □

So far, we have not made use of the asymptotic result in (12). For the recursion $h'_{p,q} \Omega_{p,q}^{-1} h_{p,q}$, however, we will use it to make the approximation

$$E'_q E_q \approx T_P I_{2q}/2$$

to obtain the following important result.

Lemma 15. *When the result in* (12) *is used to approximate* $\Omega_{0,q} = T_P^{-1} E'_q E_q$ *by its asymptotic value* $I_{2q}/2$*, we obtain*

$$h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} = g'_{p,q} \widehat{d}_{p,g}.$$

Proof. From (A.3), we have

$$h'_{p,q} \Omega_{p,q}^{-1} h_{p,q} = g'_{p,q} Q'_{p,q} \Omega_{p,q}^{-1} Q_{p,q} g_{p,q}.$$

Since, from (12),

$$\Omega_{p,q} = I_{2q} T_P/(2T) - T^{-1} E'_q P_{Z_p} E_q,$$

and since $Q_{p,q}$ is symmetric and orthogonal, we get

$$\begin{aligned} Q'_{p,q} \Omega_{p,q}^{-1} Q_{p,q} &= \left(Q'_{p,q} \Omega_{p,q} Q_{p,q} \right)^{-1} = \left(Q'_{p,q} Q_{p,q}/2 - T^{-1} Q'_{p,q} E'_q P_{Z_p} E_q Q_{p,q} \right)^{-1} \\ &= \left(I_{2q} T_P/(2T) - T^{-1} E'_q P_{Z_p} E_q \right)^{-1} = \Omega_{p,q}^{-1} \end{aligned}$$

where the second last equality follows from $Q'_{p,q} Q_{p,q} = I_{2q}$, from Lemma 13, and since $(Z'_p Z_p)^{-1}$ is centrosymmetric so that $(Z'_p Z_p)^{-1} = J_p (Z'_p Z_p)^{-1} J_p$. Thus,

$$g'_{p,q} Q'_{p,q} \Omega_{p,q}^{-1} Q_{p,q} g_{p,q} = g'_{p,q} \Omega_{p,q}^{-1} g_{p,q}$$

from which the result follows. □

The results in Corollary 12 and Lemma 15 can now be used to derive an efficient recursion for the residual mean square $\widehat{\sigma}^2_{p+1,q}$.

Theorem 16. *The residual mean square* $\widehat{\sigma}^2_{p,q}$ *can be updated recursively for* $p = 0, \ldots, P-1$ *using*

$$\widehat{\sigma}^2_{p+1,q} = \widehat{\sigma}^2_{p,q}\left(1 - \kappa^2_{p+1,q}\right)$$

with initial condition

$$\widehat{\sigma}^2_{0,q} = T^{-1}X'_P X_P - 2g'_{0,q}g_{0,q}T/T_p = \widehat{\sigma}^2_{0,q-1} - 2\widetilde{g}'_{0,q}\widetilde{g}_{0,q}T/T_p$$

where $\widetilde{g}_{0,q}$ *contains the last two elements of* $g_{0,q}$.

Proof. Inserting the result from Corollary 12 and Theorem 8 in (16) gives

$$\begin{aligned}\widehat{\sigma}^2_{p+1,q} &= \widehat{\sigma}^2_{p+1,0} - g'_{p+1,q}\widehat{d}_{p+1,q} \\ &= \widehat{\sigma}^2_{p,0}\left(1 - \nu^2_{p+1}\right) - g'_{p,q}\widehat{d}_{p,q} - \kappa^2_{p+1,q}\left(\widehat{\sigma}^2_{p,0} - h'_{p,q}\Omega^{-1}_{p,q}h_{p,q}\right) + \nu^2_{p+1}\widehat{\sigma}^2_{p,0} \\ &= \widehat{\sigma}^2_{p,0} - g'_{p,q}\widehat{d}_{p,q} - \kappa^2_{p+1,q}\left(\widehat{\sigma}^2_{p,0} - h'_{p,q}\Omega^{-1}_{p,q}h_{p,q}\right) = \widehat{\sigma}^2_{p,q}\left(1 - \kappa^2_{p+1,q}\right)\end{aligned}$$

where the last equality follows from Lemma 15. □

This concludes the derivation of the F$_0$-AR-ML-A2 algorithm whose pseudo-code can be found in algorithm 3.

References

[1] A. Zygmund, R. Fefferman, Trigonometric Series, 3rd Edition, Cambridge University Press, 2003.

[2] J. Benesty, M. M. Sondhi, Y. Huang (Eds.), Springer handbook of speech processing, Berlin Heidelberg, Germany: Springer-Verlag, 2007.

[3] C. J. Plack, A. J. Oxenham, R. R. Fay, Pitch: neural coding and perception, New York, NY, USA: Springer Science & Business Media, 2005.

[4] S. Gade, H. Herlufsen, H. Konstantin-Hansen, N. J. Wismer, Order tracking analysis, Technical review no. 2, Brüel & Kjær A/S (1995).

[5] A. Brandt, Noise and vibration analysis: signal analysis and experimental procedures, Chichester, West Sussex, UK: John Wiley & Sons, 2011.

[6] A. Gacek, W. Pedrycz (Eds.), ECG signal processing, classification and interpretation: a comprehensive framework of computational intelligence, London, UK: Springer Science & Business Media, 2011.

[7] V. K. Murthy, L. J. Haywood, J. Richardson, R. Kalaba, S. Salzberg, G. Harvey, D. Vereeke, Analysis of power spectral densities of electrocardiograms, Mathematical Biosciences 12 (1–2) (1971) 41–51.

[8] A. K. Barros, N. Ohnishi, Heart instantaneous frequency (hif): an alternative approach to extract heart rate variability, IEEE Trans. Biomed. Eng. 48 (8) (2001) 850–855.

[9] P. Boersma, Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound, in: Proc. Inst. Phonetic Sci., Vol. 17, 1993, pp. 97–110.

[10] D. Talkin, A robust algorithm for pitch tracking (RAPT), in: W. B. Kleijn, K. K. Paliwal (Eds.), Speech Coding and Synthesis, Elsevier, 1995, Ch. 14, pp. 495–518.

[11] A. de Cheveigné, H. Kawahara, YIN, a fundamental frequency estimator for speech and music, J. Acoust. Soc. Am. 111 (4) (2002) 1917–1930.

[12] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, S. Khudanpur, A pitch extraction algorithm tuned for automatic speech recognition, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., IEEE, 2014, pp. 2494–2498.

[13] B. G. Quinn, P. J. Thomson, Estimating the frequency of a periodic function, Biometrika 78 (1) (1991) 65–74.

[14] J. Tabrikian, S. Dubnov, Y. Dickalov, Maximum a-posteriori probability pitch tracking in noisy environments using harmonic model, IEEE Trans. Speech Audio Process. 12 (1) (2004) 76–87.

[15] Y. Stylianou, Modeling speech based on harmonic plus noise models, in: G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications, 2005, pp. 244–260.

[16] M. G. Christensen, A. Jakobsson, Multi-Pitch Estimation, San Rafael, CA, USA: Morgan & Claypool, 2009.

[17] J. K. Nielsen, T. L. Jensen, J. R. Jensen, M. G. Christensen, S. H. Jensen, Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient, Signal Processing 135 (2017) 188–197.

[18] S. M. Nørholm, J. R. Jensen, M. G. Christensen, Instantaneous pitch estimation with optimal segmentation for non-stationary voiced speech, IEEE/ACM Trans. Audio, Speech, Language Process. 24 (12) (2016) 2354–2367.

[19] M. G. Christensen, Accurate estimation of low fundamental frequencies from real-valued measurements, IEEE/ACM Trans. Audio, Speech, Language Process. 21 (10) (2013) 2042–2056.

[20] A. E. Jaramillo, J. K. Nielsen, M. G. Christensen, A study on how pre-whitening influences fundamental frequency estimation, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2019.

[21] A. E. Jaramillo, A. Jakobsson, J. K. Nielsen, M. G. Christensen, Robust fundamental frequency estimation in coloured noise, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2020.

[22] S. Kay, V. Nagesha, Extraction of periodic signals in colored noise, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Vol. 5, 1992, pp. 281–284.

[23] S. M. Kay, V. Nagesha, Maximum likelihood estimation of signals in autoregressive noise, IEEE Trans. Signal Process. 42 (1) (1994) 88–101.

[24] S. J. Godsill, M. Davy, Bayesian harmonic models for musical pitch estimation and analysis, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Vol. 2, 2002, pp. 1769–1772.

[25] E. J. Hannan, The estimation of frequency, J. App. Prob 10 (1973) 510–519.

[26] J. Li, P. Stoica, Efficient mixed-spectrum estimation with applications to target feature extraction, IEEE Trans. Signal Process. 44 (2) (1996) 281–295.

[27] P. Stoica, A. Jakobsson, J. Li, Cisoid parameter estimation in the colored noise case: asymptotic cramér-rao bound, maximum likelihood, and nonlinear least-squares, IEEE Trans. Signal Process. 45 (8) (1997) 2048–2059.

[28] B. G. Quinn, Estimating a sinusoid in low snr coloured noise, in: Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004., 2004, pp. 301–306.

[29] B. G. Quinn, Efficient estimation of the parameters in a sum of complex sinusoids in complex autoregressive noise, in: Rec. Asilomar Conf. Signals, Systems, and Computers, 2007, pp. 636–640.

[30] B. G. Quinn, Estimating the number of terms in a sinusoidal regression, Journal of Time Series Analysis 10 (1) (1989) 71–75.

[31] L. Kavalieris, E. J. Hannan, Determining the number of terms in a trigonometric regression, Journal of Time Series Analysis 15 (6) (1994) 613–625.

[32] S. M. Kay, Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory, Englewood Cliffs, NJ, USA: Prentice Hall PTR, 1993.

[33] B. G. Quinn, E. J. Hannan, The Estimation and Tracking of Frequency, Cambridge University Press, 2001.

[34] J. Makhoul, Linear prediction: A tutorial review, P. IEEE 63 (4) (1975) 561–580.

[35] A. M. Noll, Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate, in: Proc. of the symposium on computer process. commun., Vol. 779, 1969.

[36] D. J. Hermes, Measurement of pitch by subharmonic summation, J. Acoust. Soc. Am. 83 (1) (1988) 257–264.

[37] T.-T. Lu, S.-H. Shiou, Inverses of 2× 2 block matrices, Comput. Math. Appl. 43 (1-2) (2002) 119–129.

[38] G. H. Golub, C. F. van Loan, Matrix Computations, 4th Edition, New Delhi, India: Hindustan Book Agency, 2015.

[39] T. Kailath, A. H. Sayed, Elementary transformations, in: T. Kailath, A. H. Sayed (Eds.), Fast Reliable Algorithms for Matrices with Structure, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999, pp. 309–319.

[40] T. Kailath, Displacement structere and array algorithms, in: T. Kailath, A. H. Sayed (Eds.), Fast Reliable Algorithms for Matrices with Structure, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999, pp. 1–56.

[41] S. Chandrasekaran, A. H. Sayed, Stabilized schur algorithms, in: T. Kailath, A. H. Sayed (Eds.), Fast Reliable Algorithms for Matrices with Structure, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999, pp. 57–83.

[42] J. K. Nielsen, T. L. Jensen, J. R. Jensen, M. G. Christensen, S. H. Jensen, Grid size selection for nonlinear least-squares optimisation in spectral estimation and array processing, in: Proc. European Signal Processing Conf., 2016.

[43] P. C. Loizou, Speech Enhancement: Theory and Practice, 2nd Edition, Boca Raton, FL, USA: CRC Press, 2013.

[44] P. Kabal, TSP speech database, Tech. rep., Dept. of Electrical & Computer Engineering, McGill University (2002).

[45] C. M. Nelke, P. Vary, Measurement, analysis and simulation of wind noise signals for mobile communication devices, in: Proc. Intl. Workshop Acoust. Echo Noise Control, 2014, pp. 327–331.