POPULATION MONTE CARLO METHODOLOGY A LA GIBBS SAMPLING

Petar M. Djurić, Bingxin Shen, and Mónica F. Bugallo

Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794, USA
E-mails: {djuric,bxshen,monica}@ece.sunysb.edu

ABSTRACT

Population Monte Carlo (PMC) algorithms iterate on sets of samples and weights to approximate a stationary target distribution. The target distribution is often the a posteriori distribution of a set of unknowns of interest given observed data and the employed model. The accuracy of the estimation depends on many factors including the number and "quality" of the generated samples. In this paper, we propose a PMC algorithm that can be used for high-dimensional models and that is built in the spirit of the Gibbs sampling method. We demonstrate the proposed approach on the classical problem of estimating the frequencies of multiple sinusoids. The simulation results show the accuracy of the estimates and their comparison with the results of an alternative approach.

Index Terms— Population Monte Carlo, Gibbs sampling, Rao-Blackwellization, high dimensional systems

1. INTRODUCTION

The population Monte Carlo (PMC) is a methodology for approximating joint distributions of unknowns. The approximation is with random measures that are represented by particles (samples) and weights [1]. The method is iterative, where at each iteration samples of the unknowns are generated from a known distribution. These particles are then assigned weights according to the importance sampling principle [2]. The particles and their weights from *all* the iterations are used for approximation.

The PMC has some resemblance to Markov chain Monte Carlo (MCMC) methods. However, the particles of PMC, unlike in MCMC methods, have different weights, and the PMC does not require burn-in periods.

The key principle for constructing the approximations with PMC is importance sampling, which is a technique for estimating properties of a particular distribution with samples generated from a different distribution. This principle is also employed in the well known particle filtering methods [3]. Recently, importance sampling has also been used in a series of papers with the objective of finding the maximum likelihood estimate of frequencies of multiple sinusoids [4], parameters of chirp signals [5], and directions of arrival [6].

As with every method based on importance sampling, the crucial factor for good performance is the choice of generating functions of the particles. In this paper, we propose that the generating functions be *alternating conditionals*, thereby mimicking the idea behind Gibbs sampling [7]. With this approach, it is expected, that one can generate particles in high dimensions more efficiently. We demonstrate the performance of the proposed approach on the problem of frequency estimation of 10 sinusoids from only 25 observations.

The paper is organized as follows. In the next section we provide a general formulation of the problem. Then, in Section 3, we describe the PMC method and briefly review some recent advances. In Section 4, we propose the Gibbs sampling-inspired PMC and the details of its implementation. We demonstrate the use and performance of the method on the problem of frequency estimation of multiple sinusoids in Section 5. We conclude the paper with Section 6.

2. PROBLEM FORMULATION

We observe a set of data $\boldsymbol{y}$ which are modeled according to

$$\boldsymbol{y} \quad = \quad h(\boldsymbol{\theta}, \boldsymbol{w}), \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{d_y \times 1}$ (or $\mathbb{C}^{d_y \times 1}$) is a vector of observations, $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta \times 1}$ is a vector of unknowns, $\boldsymbol{w} \in \mathbb{R}^{d_w \times 1}$ (or $\mathbb{C}^{d_w \times 1}$) is a noise vector with a known parametric distribution (typically $d_w = d_y$), and $h : \mathbb{R}^{d_\theta} \times \mathbb{R}^{d_w} \rightarrow \mathbb{R}^{d_y}$ (or $h : \mathbb{R}^{d_\theta} \times \mathbb{C}^{d_w} \rightarrow \mathbb{C}^{d_y}$) is a known function of the unknowns and the noise.

For the unknowns, we assume that we have the a priori distribution $\pi(\boldsymbol{\theta})$, and that given the noise probability distribution, we can write the conditional distribution $p(\boldsymbol{y}|\boldsymbol{\theta})$. Given the observation vector $\boldsymbol{y}, \pi(\boldsymbol{\theta})$, and $p(\boldsymbol{y}|\boldsymbol{\theta})$, we want to compute the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, which can be written as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \quad \propto \quad p(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \tag{2}$$

This work was supported by the NSF under Awards CCF-0953316 and CCF-1018323 and by the ONR under Award N00014-09-1-1154.

where $\propto$ symbolizes proportionality. We refer to $p(\boldsymbol{\theta}|\boldsymbol{y})$ as our target distribution. In some cases, we may not be interested in the complete posterior of $\boldsymbol{\theta}$, and instead, only in the posterior of some subset of $\boldsymbol{\theta}$.

3. A BRIEF REVIEW OF THE PMC METHOD

The PMC method was introduced in [1]. The origins of PMC can be traced back in the works of Von Neumann, Metropolis, Ulam and others [8]. In [1], PMC was applied to Bayesian modeling of a Gaussian mixture and ion channel models, where it was proposed that the generating distributions be split into classes of distributions with different parameters. As the generation of particles with iterations proceeds, the quality of the generated particles improves. This is a very important feature of PMC. For example, it has been shown that PMC can be used for variance reduction [9], where a mixture of generating functions can be iteratively optimized to achieve a minimum asymptotic variance for a function of interest.

In our previous work with PMC, we have addressed the problem of estimation of frequencies of multiple sinusoids [10]. There we have exploited the principle of Rao-Blackwellization to improve the efficiency of the method by marginalizing the unwanted parameters (all the parameters except the frequencies). In other words, we applied the PMC only on the nonlinear parameters of the model. Also, we used several PMC algorithms that operated in parallel, each of them producing samples and weights of a subset of the parameters.

4. PMC IN THE SPIRIT OF GIBBS SAMPLING

Gibbs sampling is an algorithm for generation of particles that represent samples from the joint probability distribution of two or more unknowns [7]. The particles have equal weights and they approximate the joint distribution or are used for computing integrals under the joint distribution. Gibbs sampling belongs to the larger class of MCMC methods and is often used for Bayesian inference [11].

In MCMC methods, sampling from a target distribution is achieved by constructing a Markov chain whose equilibrium distribution is the target distribution. In general, at iteration j, one proposes a sample (particle) from a proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}_{j-1})$, i.e., $\boldsymbol{\theta}_j \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}_{j-1})$, [11]. We either accept or reject the proposals, where rejection means that the particle of iteration j remains the same as that from iteration $j-1$. Gibbs sampling is a special type of MCMC sampling where each $\theta_{k,j}$ (where $\theta_{k,j}$ is the $k-$th element of $\boldsymbol{\theta}$ at iteration j) is sampled from the conditional distribution $p(\theta_k|\boldsymbol{\theta}_{-k,j-1})$, where $\boldsymbol{\theta}_{-k,j-1}$ is the vector of all the parameters in $\boldsymbol{\theta}$ except for θ_k and the remaining conditioning parameters are at their current values (i.e., we use the last drawn values for the conditioning parameters). In Gibbs sampling the drawn values are always accepted.

Irrespective of which MCMC approach we use, we have issues with convergence assessment, that is, we have to run the simulations long enough so that the distribution of the drawn particles gets close to the target distribution. This problem, however, can be put away if we introduce importance sampling. In other words, if a particle $\boldsymbol{\theta}^{(m)}$ is obtained from $q(\boldsymbol{\theta})$ and we want that $\boldsymbol{\theta}^{(m)}$ is used in the approximation of $p(\boldsymbol{\theta})$, then we need to assign the particle an importance weight given by

$$w^{(m)^*} = \frac{p(\boldsymbol{\theta}^{(m)})}{q(\boldsymbol{\theta}^{(m)})}. \qquad (3)$$

The weights and the particles form a random measure, $\chi = \{\boldsymbol{\theta}^{(m)}, w^{(m)}\}_{m=1}^{M}$, where the $w^{(m)}$s are normalized weights and M denotes the total number of samples. In PMC, we implement the generation of particles through iterations. For example, at iteration one, we get the random measure χ_1, at iteration two, the random measure χ_2 and so on. The objective is that, as we proceed with iterations, we improve the accuracy of the approximation. To that end, for obtaining better generating functions, one can use the approximations from the previous iterations. One way of exploiting the previous iteration is to employ resampling (another operation that is common in particle filtering) [3]. That is, we construct new generating functions by using particles from the previous iteration that are selected based on their weights.

Here we propose a general approach for constructing generating functions for the PMC method. We draw the particles of particular unknowns from a conditional distribution, where the conditioning is on the remaining unknowns. We basically mimic the Gibbs sampling idea, where as explained, we replicate the same steps except that our conditionals are not obtained from the target distribution. Note that we apply PMC because we *cannot* generate from the conditionals of the target distribution, and therefore we work with a different joint distribution, but one that allows for easy drawing of particles.

We now describe the specific steps of the proposed scheme. At iteration $j = 0$, we initialize the particle streams by drawing them from the prior $\pi(\boldsymbol{\theta})$. We draw M particles, and to each of them we assign the weights according to

$$w_0^{(m)^*} = p(\boldsymbol{y}|\boldsymbol{\theta}_0^{(m)}). \qquad (4)$$

We assume now that at iteration $j-1$, we have the particles and the weights $\chi_{j-1} = \{\boldsymbol{\theta}_{j-1}^{(m)}, w_{j-1}^{(m)}\}_{m=1}^{M}$. We also recall that $\boldsymbol{\theta}_{j-1}^{(m)} = [\theta_{1,j-1}^{(m)}, \theta_{2,j-1}^{(m)}, \cdots, \theta_{d_\theta,j-1}^{(m)}]^\top$. The particles at the $j-$th iteration are obtained as follows:

Step 1 Randomly choose the order of generation of the parameters θ_k. Let the order be $l_1, l_2, \cdots, l_{d_\theta}$.

Step 2 For $m = 1, 2, \cdots, M$, proceed as follows. Choose a particle for conditioning based on the normalized weights of the particles from the previous iterations, which amounts to sampling from a multinomial distribution defined by the normalized weights of the particles. Let the selected particle be with index λ_m. Then generate new particles according to

$$\theta_{l_1,j}^{(m)} \sim q_{l_1,j}\left(\theta_{l_1}|\theta_{l_2,j-1}^{(\lambda_m)}, \theta_{l_3,j-1}^{(\lambda_m)}, \cdots, \theta_{l_{d_\theta},j-1}^{(\lambda_m)}\right)$$

for $n = 2, 3, \cdots, d_\theta - 1$,

$$\begin{aligned}\theta_{l_n,j}^{(m)} \sim q_{l_n,j}\Big(\theta_{l_n}|\theta_{l_1,j}^{(m)}, \cdots \theta_{l_{n-1},j}^{(m)},\\ \theta_{l_{n+1},j-1}^{(\lambda_m)} \cdots, \theta_{l_{d_\theta},j-1}^{(\lambda_m)}\Big)\end{aligned}$$

and

$$\theta_{l_{d_\theta},j}^{(m)} \sim q_{l_{d_\theta},j}\left(\theta_{l_{d_\theta}}|\theta_{l_1,j}^{(m)}, \theta_{l_2,j}^{(m)}, \cdots, \theta_{l_{d_\theta-1},j}^{(m)}\right).$$

Step 3 Computation of the weights by

$$w_j^{(m)^*} = \frac{p\left(\boldsymbol{y}|\boldsymbol{\theta}_j^{(m)}\right) p\left(\boldsymbol{\theta}_j^{(m)}\right)}{\prod_{n=1}^{d_\theta} q_{l_n,j}(\theta_{l_n,j}^{(m)})}.$$

The computed weights are stored as they were computed by the last expression. The particles from all the iterations are normalized at the end for best possible approximation of the distribution of interest. However, the weights from Step 3 are also separately normalized before starting the next iteration, so that one can use the normalized weights. The method can stop at any iteration.

We note that if we cannot generate $\boldsymbol{\theta}_0^{(m)}$ from $\pi(\boldsymbol{\theta})$, we can use a convenient generating function $q(\boldsymbol{\theta})$, and therefore the initial weights of the particles are

$$w_0^{(m)^*} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta}_0^{(m)}) p\left(\boldsymbol{\theta}_0^{(m)}\right)}{q(\boldsymbol{\theta}_0^{(m)})}. \quad (5)$$

5. FREQUENCY ESTIMATION OF MULTIPLE SINUSOIDS

In this section we demonstrate the proposed method on the problem of frequency estimation of complex sinusoids in noise [4]. We model the data as

$$y_t = \sum_{k=1}^{K} a_k e^{i(2\pi f_k t + \phi_k)} + v_t, \quad t = 1, 2, ..., d_y \quad (6)$$

where $i = \sqrt{-1}$, $0 < f_1 < f_2 < ... < f_K < 1$, and $a_k > 0$ and ϕ_k are the amplitude and phase of the $k-$th sinusoid, respectively. The noise in the collected data, v_t, is a white complex Gaussian noise of the form

$$v_t \sim \mathcal{CN}(0, \sigma_v^2)$$

with real and imaginary components that are independent and come from $\mathcal{N}(0, \frac{\sigma_v^2}{2})$. The vector of unknowns is $\boldsymbol{\theta} = [a_1, \phi_1, f_1, ..., a_K, \phi_K, f_K, \sigma_v^2]^\top$, and therefore the space of unknowns has dimension $3K + 1$. The prior of the unknowns is proportional to a constant, i.e.

$$p(\boldsymbol{\theta}) \propto \text{const.} \quad (7)$$

over the support of $\boldsymbol{\theta}$.

We are primarily interested in the frequencies, so we work with the marginalized PMC (MPMC) method as described in [12]. Thus, the parameter space of interest is $\boldsymbol{\theta} = [f_1, f_2, ..., f_K]^\top$. The posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ can be obtained in a closed analytical form [12], but one cannot draw samples from it. Here, we apply the proposed scheme where each of the conditionals is a truncated Gaussian centered at the selected particle from the previous iteration. The conditioning parameters are used for deciding the truncation points of the Gaussian. For example, if the frequency $f_{k,j}$ needs to be generated, we use as a generating function the truncated Gaussian, which is centered at $f_{k,j-1}^{(m)}$ with cutoff points $f_{k-1,*}^{(m)}$ and $f_{k+1,*}^{(m)}$, where the $*$ stands for the most recent sample of f_{k-1} and f_{k+1}, respectively.[1] The importance of this choice is demonstrated in the experimental results shown below. All the frequency estimates are minimum mean square error estimates.

We tested the method by conducting simulations as follows. We simulated $d_y = 25$ observations with $K = 10$ sinusoids, whose frequencies were $f_1 = 0.2$, $f_2 = 0.3$, $f_3 = 0.32$, $f_4 = 0.5$, $f_5 = 0.52$, $f_6 = 0.7$, $f_7 = 0.75$, $f_8 = 0.8$, $f_9 = 0.82$, and $f_{10} = 0.9$, with amplitudes $a_k = 1$, for $k = 1, 2, ..., 10$, and phases $\phi_k = 0$, for $k = 1, ..., 4, 6, ..., 10$ and $\phi_5 = \pi/4$, respectively. The value of the noise power was defined by using the signal-to-noise ratio (SNR)

$$SNR = 10 \log_{10} \frac{a^2}{\sigma_v^2}$$

measured in dB.

For comparisons, we employed the MPMC method, which also uses the truncated Gaussians with the same centers but with truncating points obtained at the previous iteration only (so, they are not the most recent drawings). We also found the Cramér-Rao lower bounds (CRLBs) for the frequencies of interest. When implementing the algorithms, we used the Yule-Walker method for getting the initial estimates [13].

The performance of the algorithms was quantified in terms of the mean square error (MSE) given by

$$MSE(f_k) = \frac{1}{R} \sum_{r=1}^{R} (\widehat{f}_k(r) - f_k)^2, \quad (8)$$

[1] Note that when $k = 1$, the lower cutoff point is 0, and when $k = K$ the upper cutoff point is 1.

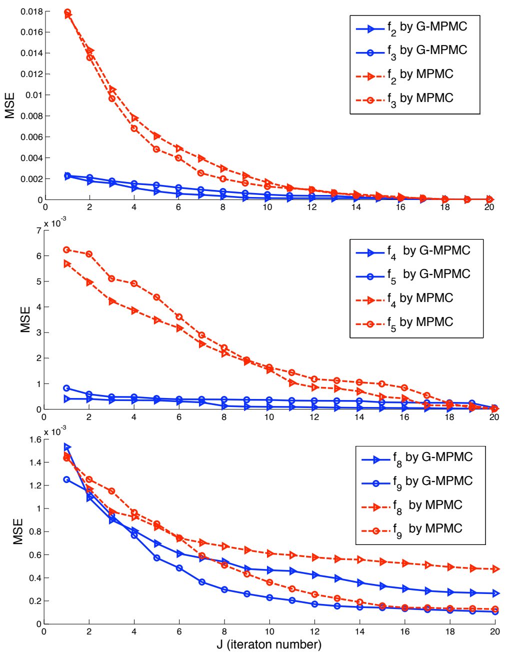

Fig. 1. MSE as a function of iteration.

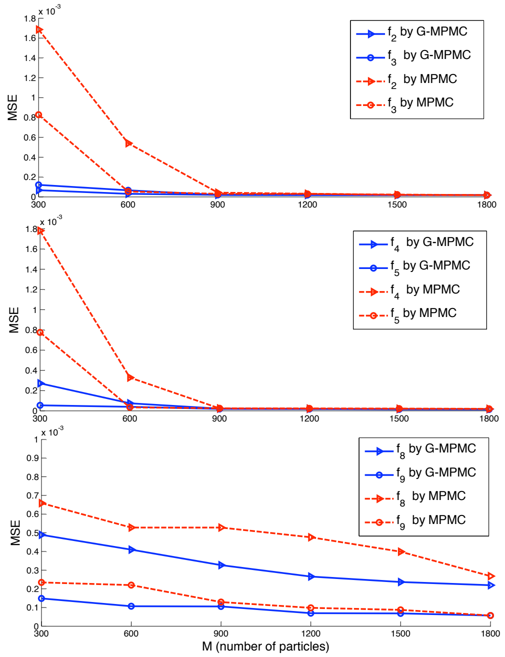

Fig. 2. MSE as a function of particle size.

where R represents the number of realizations, $\widehat{f}_k(r)$ denotes the estimate obtained in the r-th run, and f_k is the true value of the frequency.

Figure 1 shows the MSE of the two algorithms as a function of iterations (the maximum number was $J = 20$ iterations). The SNR was 5 dB, $M = 900$ particles, and $R = 500$ runs. At each run, the estimates at the $j-$th iteration were obtained from all the generated particles and associated weights up to that iteration. In the figure, the performance of the novel scheme is denoted by G-MPMC. From the graphs, it is clear that G-MPMC outperforms the MPMC. The G-MPMC estimates of the unknown frequencies converge much more quickly to the true values.

In Figure 2, we see the MSE for different sizes of particle populations (M was changed from 300 to 1800 particles), SNR $= 5$ dB, $J = 10$ iterations, and $R = 500$ runs. At each run, the estimates were obtained from all the particles after $J = 10$ iterations. The plots show that the G-MPMC can achieve the same accuracy with a smaller amount of particles than the MPMC algorithms, and therefore, it is less computationally expensive.

The MSE for various values of SNR is shown in Figure 3. All the points on the plot were averaged over $R = 500$ runs with a particle size of $M = 900$ and for $J = 10$ iterations. The proposed method again outperforms the MPMC considerably.

We point out that in all the simulations, we present the results of point estimates. The particles and their weights provide, however, much more information. They can readily be used to obtain other types of statistical inference.

Finally, Table 1 shows the number of poor estimates (defined as estimates of $\widehat{\mathbf{f}}$ where $|f_k - \widehat{f}_k| > 0.1$ is true for at least one k, and $k = 1, 2, \cdots, 10$) of the MUSIC (Multiple Signal Classification) algorithm, the Yule-Walker method, the MPMC method, and the proposed G-MPMC method. All the data were averaged over $R = 500$ runs with $SNR = 5$. Clearly, the G-MPMC had the best performance again.

6. CONCLUSION

The most critical issue in applying population Monte Carlo methods is the choice of generating functions of the particles. In this paper, we proposed that these functions are alternating conditionals, as in the case of Gibbs sampling. Thus, the overall proposal function is a product of conditionals, and where the sampling from each conditional is easy. It is expected that with alternating conditionals one can efficiently generate particles in high dimensions. The method was tested on the problem of frequency estimation of 10 sinusoids from 25

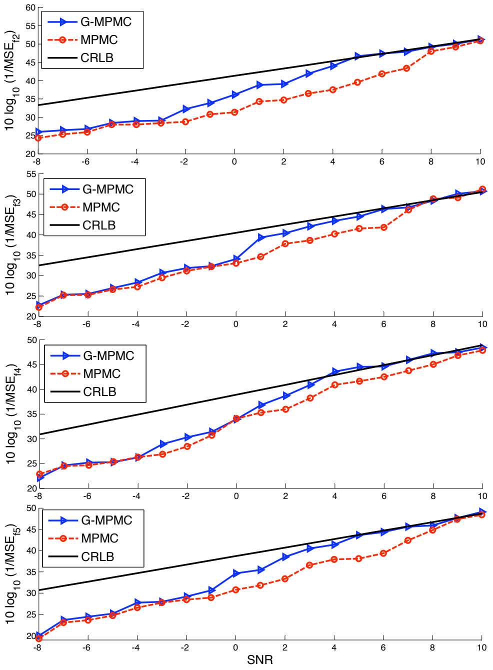

Fig. 3. MSE as a function of SNR.

observations only. The obtained results show very good performance of the method.

7. REFERENCES

[1] O. Cappé, A. Guillin, and C. P. Robert, "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, vol. 13, pp. 907–929, 2004.

[2] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creation a few imputations when fractions of missing information are modest: the SIR algorithm," *Journal of the American Statistical Association*, vol. 82, pp. 543–546, 1987.

[3] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[4] S. Kay and S. Saha, "Mean likelihood frequency estimation," *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 1937–1946, 2000.

	outliers	outlier rate
MUSIC	278	0.556
Yule-Walker	258	0.516
MPMC (M=900, J=10)	9	0.018
G-MPMC (M=900, J=10)	3	0.006
MPMC (M=900, J=20)	2	0.004
G-MPMC (M=900, J=20)	0	0

Table 1. Outliers of different methods.

[5] S. Saha and S. Kay, "Maximum likelihood parameter estimation of superimposed chirp signals using Monte Carlo importance sampling," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 224–230, 2002.

[6] H. Wang, S. Kay, and S. Saha, "An importance sampling maximum likelihood direction of arrival estimator," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5082–5092, 2008.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[8] Y. Iba, "Population Monte Carlo algorithms," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 16, no. 2, pp. 279–286, 2001.

[9] R. Douc, A. Guillin, J. M. Marin, and C. P. Robert, "Minimum variance importance sampling via population Monte Carlo," *ESAIM: Probability and Statistics*, vol. 11, pp. 420–448, 2007.

[10] B. Shen, M. F. Bugallo, and P. M. Djurić, "Multiple marginalized population Monte Carlo," in *the Proceedings of EUSIPCO*, Aalborg, Denmark, 2010.

[11] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman & Hall, New York, 1995.

[12] M. Bugallo, M. Hong, and P. M. Djurić, "Marginalized population Monte Carlo," in *the Proceedings of ICASSP*, Taipei, Taiwan, 2009.

[13] S. M. Kay, *Modern Spectral Estimation*, Prentice Hall, Englewood Cliffs, NJ, 1988.