Comparative Performance Analysis of Three Algorithms for Principal Component Analysis

Ronnie LANDQVIST, Abbas MOHAMMED

Dept. of Signal Processing, Blekinge Institute of Technology, 372 25 Ronneby, Sweden

ronnie.landqvist@gmail.com, abbas.mohammed@bth.se

Abstract. *Principal Component Analysis (PCA) is an important concept in statistical signal processing. In this paper, we evaluate an on-line algorithm for PCA, which we denote as the Exact Eigendecomposition (EE) algorithm. The algorithm is evaluated using Monte Carlo Simulations and compared with the PAST and RP algorithms. In addition, we investigate a normalization procedure of the eigenvectors for PAST and RP. The results show that EE has the best performance and that normalization improves the performance of PAST and RP algorithms, respectively.*

Keywords

Principal component analysis, signal processing, exact eigen-decomposition.

1. Introduction

Principal Component Analysis (PCA) is an important concept in statistical signal processing. PCA is used in various applications such as feature extraction (compression), signal estimation and detection. In blind signal separation, PCA may be used to prewhite (i.e. decorrelate) signals before Independent Component Analysis (ICA) is applied for finding the independent signal components. ICA algorithms are generally more efficient if the input data is white since the number of possible solutions to the problem decreases significantly.

There are two different approaches to PCA. If the whole data set is available, analytical methods can be used to calculate the Principal Components (PC). On the other hand, if PCA is to be used in real time applications, then the PCs might have to be estimated on-line for each new sample. Algorithms that operate on the whole date set are generally denoted off-line algorithms, while the latter type is denoted on-line algorithms.

Two on-line algorithms for PCA are the Projection Approximation Subspace Tracking (PAST) algorithm and the Rao-Principe (RP) algorithm. The first algorithm was proposed by Yang in [1]. Essentially, PAST finds the PCs by minimizing the linear PCA criterion [2], [3], [4] using a gradient-descent technique or any recursive least squares variant. In this paper, the well known Recursive Least Squares (RLS) algorithm [5] is used. The second algorithm, RP, was proposed by Rao and Principe in [6], [7]. RP estimates the PCs by using update equations derived from the Rayleigh quotient, without the use of any external parameter such as a step-size or forgetting factor. PAST, on the other hand, relies on a forgetting factor parameter to be tuned before it can be used.

This paper investigates the effect of incorporating normalization of the eigenvectors in the PAST and RP algorithms. In addition, an on-line algorithm for PCA, denoted as the Exact Eigendecomposition (EE) algorithm, is proposed. EE is based on direct estimation of the correlation matrix followed by calculation of the PCs using the EIG-command in Matlab. The performance of these algorithms will be assessed for different configurations using Monte Carlo computer simulations in Matlab.

The organization of the rest of the paper is as follows. In section 2, the different investigated algorithms for PCA are reviewed. The estimation of coefficients and normalization procedure of the eigenvectors for the algorithms is outlined in section 3. The simulation results and performance evaluations of the different algorithms are presented in section 4. Finally, section 5 concludes the paper.

2. Principal Component Analysis

On-line PCA can be viewed as a functional block where the input is a complex m-by-1 data vector $\mathbf{x}(n)$ at the n^{th} time instant. At each time instant the outputs are estimated eigenvectors and eigenvalues of the correlation matrix $\mathbf{R}$ of the data. These eigenvectors and eigenvalues are denoted PCs of the data. Note that in this paper it is assumed that the correlation matrix $\mathbf{R}$ of the data is time-invariant.

2.1 The PAST Algorithm

The PAST algorithm [1] minimizes an approximation of the linear PCA cost criterion,

$$J(\mathbf{w}(n)) = \sum_{i=1}^{n} \beta^{\mathrm{n-1}} \| \mathbf{x}(i) - \mathbf{w}(n)\, x'(i) \|^2 \qquad (1)$$

where $x'(n) = \mathbf{w}^H(n\text{-}1)\mathbf{x}(n)$, β is a scalar forgetting factor, $\mathbf{w}(n)$ is an m-by-1 coefficient vector and $\|.\|$ denotes the vector norm. This approximate version of the cost criterion is quadratic and has the same form as the cost criterion of the RLS algorithm, with exception of the error signal (inside the vector norm) that is a vector instead of a scalar. Thus, the minimization may be performed by incorporating the new signal $x'(n)$ in the RLS. The RLS will update the coefficient vector $\mathbf{w}(n)$ using $x'(n)$ as the input signal and $\mathbf{x}(n)$ as the desired signal. When the algorithm has converged, $\mathbf{w}(n)$ will contain the eigenvector corresponding to the largest eigenvalue of the correlation matrix $\mathbf{R}$, i.e. the largest PC. The eigenvalue can be found directly by the RLS via a reformulation of the update equation for the inverse correlation matrix. The deflation technique is used for sequential estimation of the remaining PCs.

2.2 The RP Algorithm

The RP [6], [7] algorithm is derived from the Rayleigh quotient and uses the following rule for extraction of the first PC:

$$\mathbf{w}(n) = (\mathbf{w}(n\text{-}1) + \mathbf{Rw}(n\text{-}1)\ ) \ /\ (1+\mathbf{w}^H(n\text{-}1)\mathbf{Rw}(n\text{-}1)). \quad (2)$$

In the implementation, terms $\mathbf{Rw}(n\text{-}1)$ and $\mathbf{w}^H(n\text{-}1)\mathbf{Rw}(n\text{-}1)$ are redefined as $\mathbf{P}(n\text{-}1)$ and $Q(n\text{-}1)$ and updated independently before calculating the new coefficient vector $\mathbf{w}(n)$. This is convenient since $Q(n\text{-}1)$ will be an estimate of the largest eigenvalue. The update rules for $\mathbf{P}(n\text{-}1)$ and $Q(n\text{-}1)$ may be rewritten so that the need for an explicit estimate of the correlation matrix $\mathbf{R}$ vanishes. No forgetting factor is incorporated in the estimation of $\mathbf{P}(n\text{-}1)$ and $Q(n\text{-}1)$, nevertheless this could be useful in a time-varying environment. The remaining PCs are estimated by using the same deflation technique as in the PAST algorithm.

2.3 The EE Algorithm

In this paper the EE algorithm is used and evaluated for PCA. In the EE algorithm, the PCs are found by directly computing the eigenvalues and eigenvectors of an estimate of the correlation matrix. The following update rule is proposed to be used for EE:

$$\mathbf{R}(n) + \beta\mathbf{R}(n\text{-}1) + \mathbf{x}(n)\mathbf{x}^H(n)\ . \quad (3)$$

Here, β is a scalar forgetting factor. The correlation matrix is estimated by $(1\text{-}\beta)\mathbf{R}(n)$ and is used for calculating the eigenvectors and eigenvalues (i.e. the PCs) by calling the EIG-function in Matlab. A key difference between EE and the other algorithms is that EE uses direct estimation of the correlation matrix as the basis for calculating the PCs, while PAST and RP directly estimates the PCs without making a detour via the correlation matrix.

3. Choice of Coefficients and Normalization

The PAST and RP algorithms estimate the different PCs in an iterative way, where the largest PC is estimated first followed by a deflation step. After the deflation, the second largest PC is estimated followed by a new deflation step and so on. This procedure continues until all desired PCs are estimated, then the sample instant n is advanced by one and the procedure is repeated for the next data vector $\mathbf{x}(n)$. The deflation step is essentially a simple algebraic rule that removes the contribution of the latest estimated PC from the data vector. The most straight forward approach is to use the most current estimate of the eigenvector in this rule, but one could also choose to use the previous estimate. Which choice that is the best is not clear and is therefore investigated by Monte Carlo simulations in this paper. Both the PAST and RP algorithm are simulated in two configurations, using the old or the new estimate.

Eigenvectors are by definition normalized so that $\mathbf{w}^H\mathbf{w}$ equals to 1. However, the formulation of the PAST and RP algorithms does not guarantee this condition. Depending on the application, the eigenvectors can be normalized after each iteration, every l iteration or when the coefficients have converged [7]. In this paper, it is proposed that both algorithms are adjusted so that the eigenvectors are normalized between iterations. Normalization is not required for EE, since the EIG-function assures that the eigenvectors are normalized at all time.

According to the previous discussion, there are now a total of four possible configurations of PAST and RP. First, the old or the new estimate may be used in the deflation rule. Second, normalization of the coefficient vectors is *on* or *off*. The different configurations are denoted as A, B, C or D: old weights and normalization off (A), old weights and normalization on (B), new weights and normalization off (C), new weights and normalization on (D). Two more configurations, denoted as E and F are also used. In these configurations, normalization is performed every 10th sample (E) and in the 10 last samples of a realization (F), respectively. In both E and F the new weights are used in the deflation step. The different configurations are summarized in Table 1.

The configuration letter (A, B, C, D, E, F) is added to the algorithm name in order to differentiate between them. Examples are PAST-A, PAST-B, …, RP-A, RP-B, etc. EE does not have these configurations and hence it is simply denoted as EE. The performance of PAST and RP is dependent of the configuration and is investigated in the next section.

	New weights	Old weights
Normalization off	C	A
Normalization every sample	D	B
Normalization every 10th sample	E	-
Normalization last 10 samples	F	

Tab. 1. The six possible configurations for PAST and RP, respectively. The configuration letter (A, B, C, D, E, F) is added to the algorithm name in order to differentiate between them. Examples are PAST-A, PAST-B, … , RP-A, RP-B, etc.

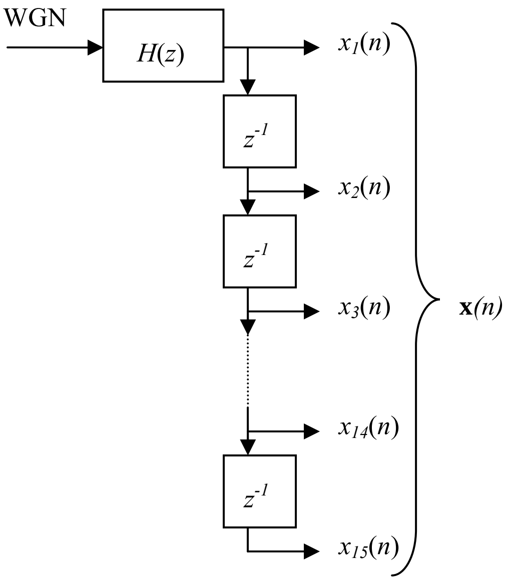

Fig. 1. The signal model used in the simulations.

4. Performance Evaluation

In this section we evaluate the performance of the PAST-x, RP-x and EE algorithms by means of computer simulations that employ a Monte Carlo approach. Here, x denotes configuration A, B, C, D, E or F, respectively. In the simulations, a random White Gaussian Noise (WGN) signal with variance 1 is filtered by a first-order AR filter H(z) = $1/(1-0.9495z^{-1})$. The resulting filtered signal is then fed to a 15-tap delay line resulting in input data vectors $\mathbf{x}(n)$ of size 15-by-1, see Fig. 1. The five largest true eigenvalues of the correlation matrix $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^H(n)\}$ are found to be 119, 17.8, 5.58, 2.65 and 1.55, respectively. The condition number or the eigenvalue spread of **R** is 451. The choice of the first-order AR filter was made so that it results in approximately the same eigenvalues used in [6].

The performance evaluation is set up so that each algorithm (PAST-x, RP-x and EE) is simulated for 10000 Monte Carlo runs (i.e. realizations) consisting of 2000 iterations (i.e. data samples). After the last iteration in each Monte Carlo run, the five largest estimated eigenvalues, λ_i, $i = 1, .., 5$, and the corresponding estimated eigenvectors, $\mathbf{w}_{i,\text{est}}$, $i = 1, .., 5$, are examined. An estimation error (the difference between the estimated eigenvalue and true eigenvalue, i.e. $\lambda_{i,est} - \lambda_i)$ is calculated and saved. Also, the absolute value of the Directional Cosines (DC), d_i, between the true and estimated eigenvectors are calculated and saved. A value of +1 indicates perfect alignment of the estimated and true eigenvectors, while 0 indicates that the estimated and true eigenvector are orthogonal. The forgetting factor β is equal to 1 for all simulations of PAST and EE.

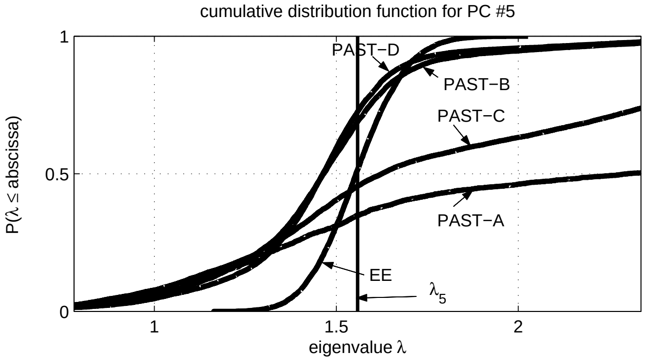

Fig. 2. Cumulative distribution functions for the 5th principal component for PAST and EE. The vertical line corresponds to the true eigenvalue.

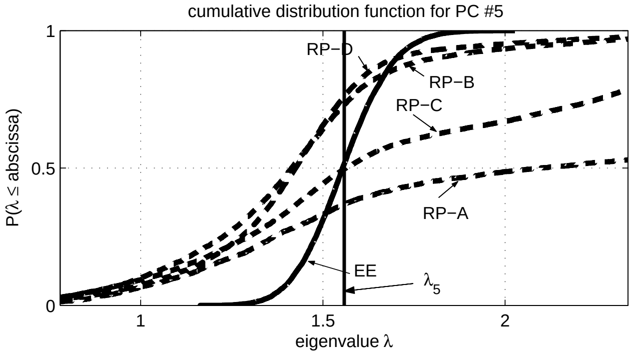

Fig. 3. Cumulative distribution functions for the 5th principal component for RP and EE. The vertical line corresponds to the true eigenvalue.

The saved eigenvalue estimates and directional cosines may be evaluated by plotting the Cumulative Distribution Function (CDF). The CDF for the eigenvalue estimates for the 5th principal component is plotted in Fig. 2 and 3 for configurations A-D for PAST and RP, respectively. The CDF for EE and the true eigenvalue λ_5 are also plotted in these figures. The figures show that PAST and RP have similar performance, that EE has the best performance and B and D modes perform better than A and C modes of the algorithms.

A compilation of all results are shown in Figures 4 to 23. The figures show performance results for all 13 algorithms for the five largest PCs. Figures 4-8 show the Normalized Magnitude Bias (NMB) for the eigenvalues, Figures 9 to 13 the Coefficient of Variation (CV) of the eigenvalue estimates, Figures 14-18 the Average Magnitude Directional Cosine (AMDC) for the eigenvectors and Figures 19-23 the Variance of the Magnitude Directional Cosines (VMDC). All these performance measures are defined in detail in Table 2.

Measure	**Definition**
Normalized Magnitude Bias (NMB)	$\mathrm{Mean}\{\lvert\lambda_{i,est}-\lambda_i\rvert / \lambda_i \}$
Coefficient of Variation (CV)	$\mathrm{Std}\{ \lambda_{i,est} \} / \mathrm{Mean}\{ \lambda_{i,est} \}$
Average Magnitude DC (AMDC)	$\mathrm{Mean}\{\lvert(\mathbf{w}_i^H\mathbf{w}_{i,est})/(\mathbf{w}_i^H\mathbf{w}_i\mathbf{w}_{i,est}^H\mathbf{w}_{i,est})\rvert\}$
Variance of Magnitude DC (VMDC)	$\mathrm{Var}\{\lvert(\mathbf{w}_i^H\mathbf{w}_{i,est})/(\mathbf{w}_i^H\mathbf{w}_i\mathbf{w}_{i,est}^H\mathbf{w}_{i,est})\rvert\}$

Tab. 2. Performance measures used in the evaluation.

By analyzing the figures, the following observations can be made regarding the eigenvalue estimation (Fig. 4 to 13):

- The B and D configurations of PAST and RP are the best since they have the lowest NMB and CV for most eigenvalues.
- The EE has the best performance of all algorithms since it results in the lowest NMB and CV for most eigenvalues.
- The PAST has slightly better performance than RP, see for example eigenvalues 4 and 5.

Similar observations can be made for the eigenvector estimation (Fig. 14 to 23):

- All algorithms perfectly estimate the first eigenvector since AMDC is equal to one and the VMDC is equal to zero for all algorithms.
- The EE perfectly estimates the first five eigenvectors since the AMDC is equal to one and the VMDC is equal to zero.
- The B and D configurations of PAST and RP are the best since AMDC is closest to one and the VMDC is the lowest for all eigenvectors.
- The PAST has slightly better performance than RP, see for example eigenvectors 4 and 5.

From these observations, the following can be concluded:

- The B and D configurations of PAST and RP have similarly good performance. The A, C, E and F configurations all have worse performance and A is the worst.
- When normalization is not used, the new weights must be used in the deflation step in order for the algorithm to be robust.
- When normalization is used, the choice of the old or the new weights in the deflation step is not crucial. Both configurations result in similar performance.
- Normalization is highly recommended since the algorithms have better performance than if normalization is not used.
- Finally, the EE algorithm has the best overall performance and may therefore be used as a benchmark for comparing different PCA algorithms.

5. Conclusions

In this paper we proposed the on-line algorithm, denoted as the Exact Eigendecomposition (EE), for PCA. In addition, we incorporated a normalization procedure of the eigenvectors in the PAST and RP algorithms, and investigated the effect for different configurations. The performance of the algorithms was compared using Monte Carlo simulations using several performance measures. Simulation results clearly demonstrate that the algorithms operate more reliably when normalization is adopted. It is also shown that the EE algorithm provides the best performance and can be used as a benchmark for comparing PCA algorithms.

References

[1] YANG, B. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 1995, vol. 32, p. 95-107.

[2] XU, L. Least mean square error reconstruction principle for self-organizing neural nets. *Neural Networks*, 1993, vol. 6, p. 627-648.

[3] PAJUNEN, P., KARHUNEN, J. Least-squares methods for blind source separation based on nonlinear PCA. *International Journal of Neural Systems*, 1997, vol. 8, p. 601-612.

[4] KARHUNEN, J., JOUTSENSALO, J. Representation and separation of signals using nonlinear PCA type learning. *Neural Networks*, 1994, vol. 7, p. 113-127.

[5] HAYKIN, S. *Adaptive Filter Theory*. Prentice Hall, 4th ed., 2002.

[6] RAO, Y. N., PRINCIPE, J. C. On-line principal component analysis based on a fixed-point approach. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002, vol. 1, p. I-981-I-984.

[7] RAO, Y. N., PRINCIPE, J. C. A fast, on-line algorithm for PCA and its convergence characteristics. *IEEE Signal Processing Society Workshop*, 2000, vol. 1, p. 299-307.

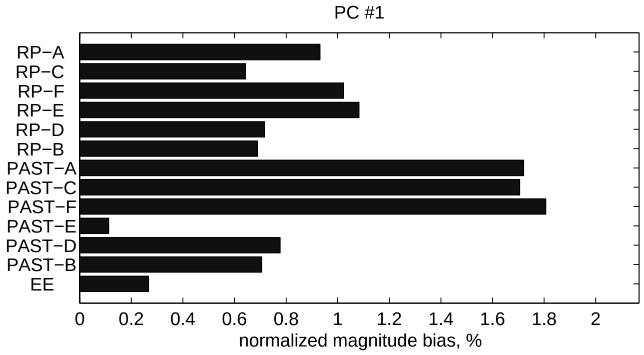

Fig. 4. Normalized magnitude bias for eigenvalue 1 for the different algorithms.

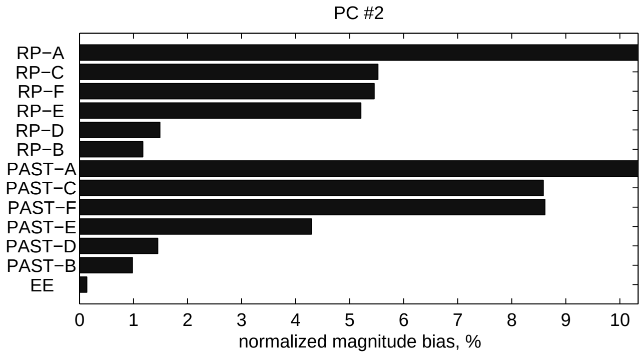

Fig. 5. Normalized magnitude bias for eigenvalue 2 for the different algorithms.

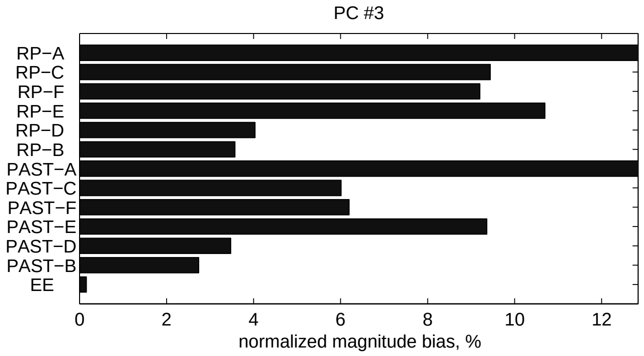

Fig. 6. Normalized magnitude bias for eigenvalue 3 for the different algorithms.

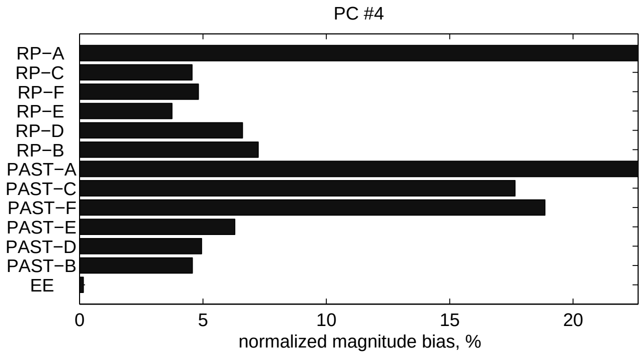

Fig. 7. Normalized magnitude bias for eigenvalue 4 for the different algorithms.

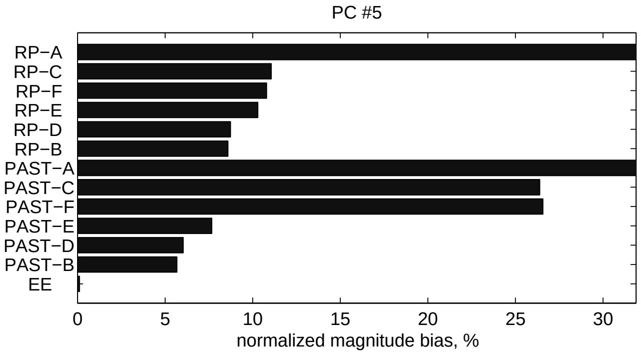

Fig. 8. Normalized magnitude bias for eigenvalue 5 for the different algorithms.

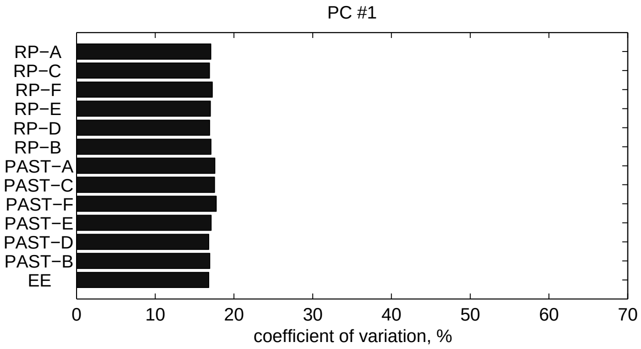

Fig. 9. Coefficient of variation for eigenvalue 1 for the different algorithms.

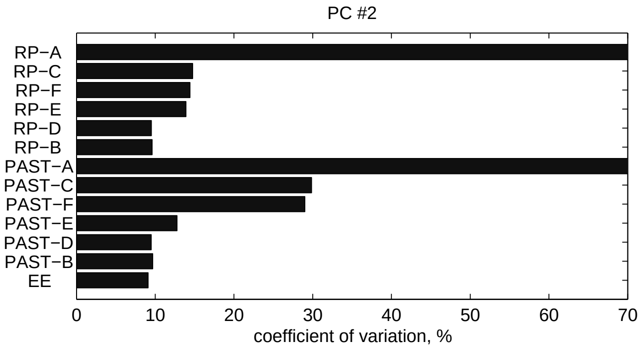

Fig. 10. Coefficient of variation for eigenvalue 2 for the different algorithms.

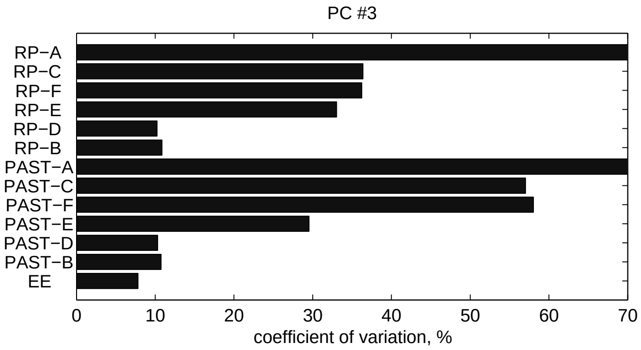

Fig. 11. Coefficient of variation for eigenvalue 3 for the different algorithms.

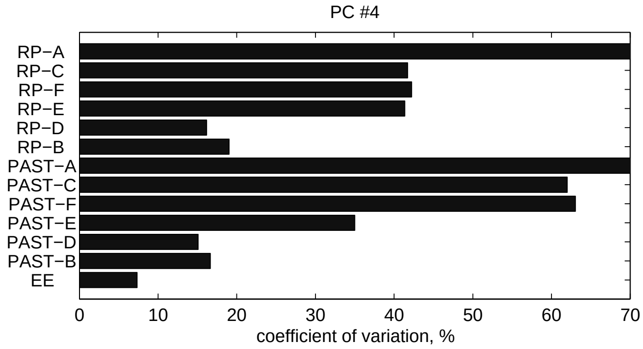

Fig. 12. Coefficient of variation for eigenvalue 4 for the different algorithms.

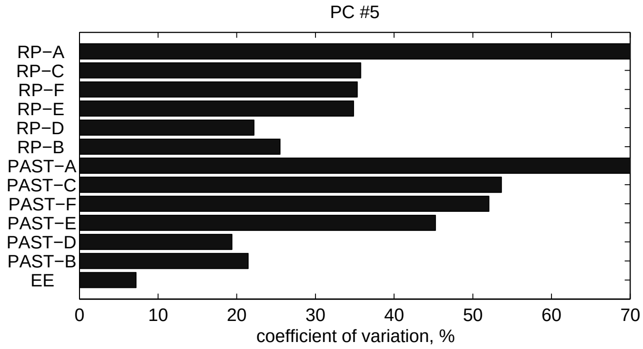

Fig. 13. Coefficient of variation for eigenvalue 5 for the different algorithms.

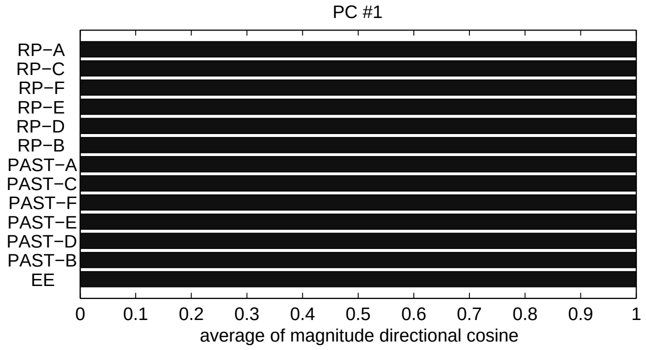

Fig. 14. Average magnitude directional cosine for eigenvector 1 for the different algorithms.

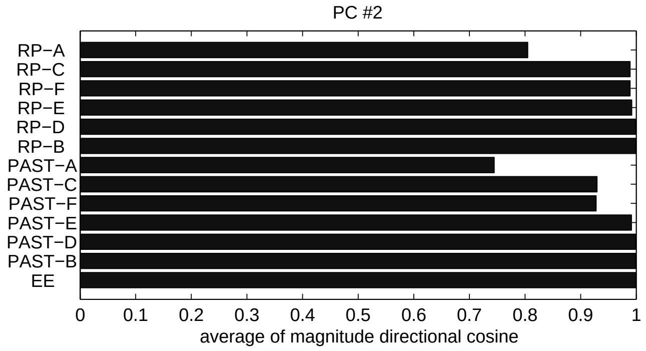

Fig. 15. Average magnitude directional cosine for eigenvector 2 for the different algorithms.

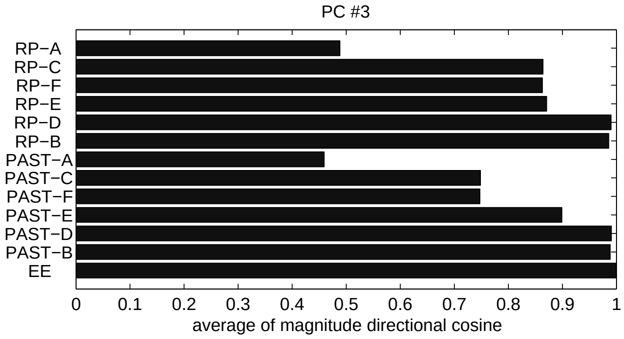

Fig. 16. Average magnitude directional cosine for eigenvector 3 for the different algorithms.

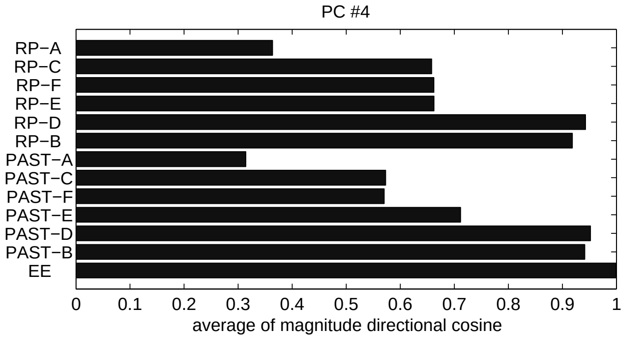

Fig. 17. Average magnitude directional cosine for eigenvector 4 for the different algorithms.

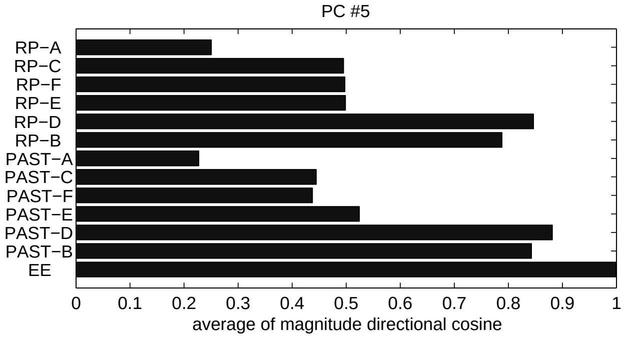

Fig. 18. Average magnitude directional cosine for eigenvector 5 for the different algorithms.

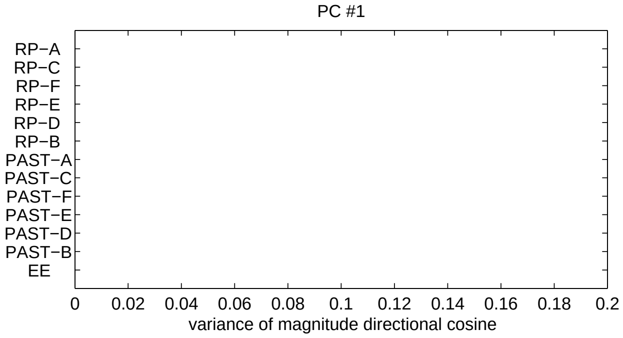

Fig. 19. Variance of magnitude directional cosine for eigenvector 1 for the different algorithms.

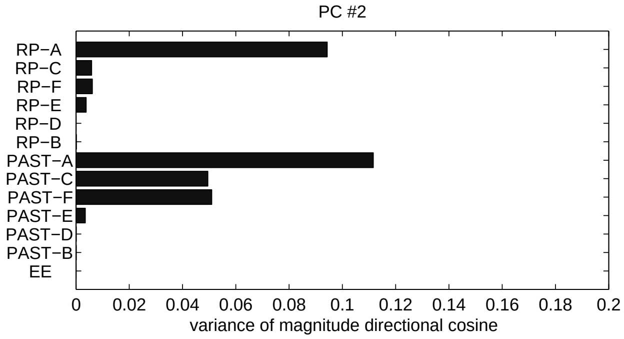

Fig. 20. Variance of magnitude directional cosine for eigenvector 2 for the different algorithms.

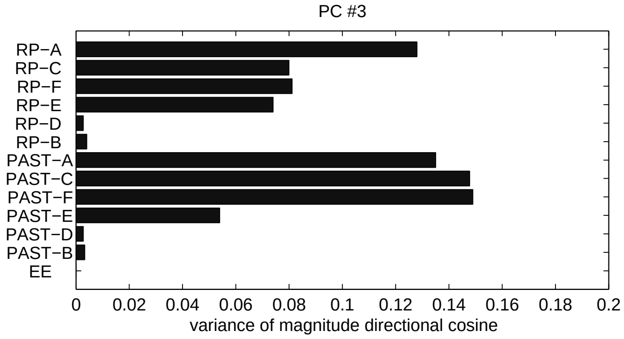

Fig. 21. Variance of magnitude directional cosine for eigenvector 3 for the different algorithms.

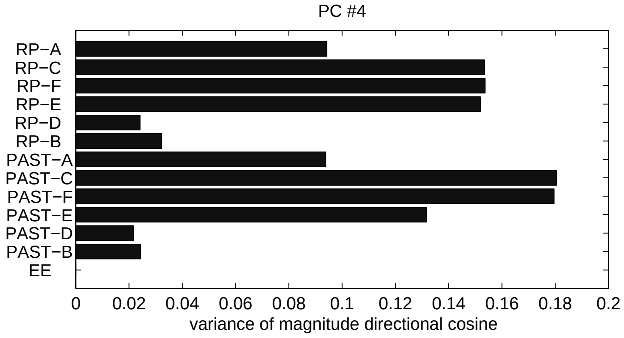

Fig. 22. Variance of magnitude directional cosine for eigenvector 4 for the different algorithms.

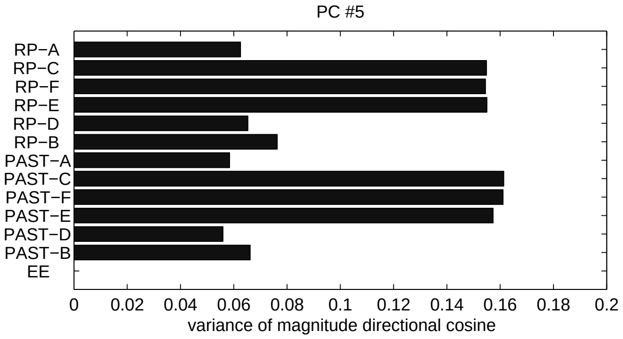

Fig. 23. Variance of magnitude directional cosine for eigenvector 5 for the different algorithms.

About Authors...

Ronnie LANDQVIST received his PhD degree from Blekinge University of Technology, Sweden, in 2005. He is currently employed by Ericsson AB where he is working in the system simulation group with development for GSM, 3G and 4G.

Abbas MOHAMMED received his PhD degree from the University of Liverpool, UK, in 1992. He is currently an Associate Professor in Radio Communications and Navigation at Blekinge Institute of Technology, Sweden. Recently, he received the Blekinge Research Foundation “Researcher of the Year Award” for 2006. He is a Fellow of IEE, life-member of the International Loran Association, member of IEEE and IEICE. In 2006, he received Fellowship of the UK’s Royal Institute of Navigation “in recognition of his significant contribution in navigation and in particular to advanced signal processing techniques that have enhanced the capability of Loran-C”. He is also a Board Member of the Radio Engineering Journal and the IEEE Signal Processing Swedish Chapter. He has published over 100 papers on telecommunications and navigation systems. He has also developed techniques for measuring skywave delays in Loran-C receivers. He was the Editor of a special issue “Advances in Signal Processing for Mobile Communication Systems” of Wiley’s International Journal of Adaptive Control and Signal Processing.