ADAPTIVE ESPRIT ALGORITHM BASED ON THE PAST SUBSPACE TRACKER

Roland Badeau, Gaël Richard and Bertrand David

Ecole Nationale Supérieure des Télécommunications
46 rue Barrault, 75634 Paris Cedex 13 France
roland.badeau, gael.richard, bertrand.david@enst.fr

ABSTRACT

The Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm is a subspace-based analysis method used in source localization or frequency estimation, originally designed in a block signal processing context. In other respects, the Projection Approximation Subspace Tracker (PAST) is a fast and robust subspace tracking method. This paper introduces a new frequency estimation and tracking algorithm, which relies on the PAST subspace tracker and a fast adaptive implementation of the ESPRIT algorithm.

1. INTRODUCTION

The Exponentially Damped Sinusoidal (EDS) model represents a signal as a sum of damped sinusoids [1]. Although the Fourier transform remains a prominent tool for frequency estimation, the ESPRIT algorithm [2] overcomes the resolution limit of the Fourier analysis and provides straight estimates of the model parameters. This method relies on the rotational invariance property of the signal subspace spanned by the damped sinusoids. Its main drawback is a high computational cost. Consequently, in an adaptive context, tracking the model parameters by recursively applying the ESPRIT algorithm would be very time consuming. Therefore, there is a clear need for fast implementations of ESPRIT that make the best of the recent subspace tracking techniques.

Adaptive implementations of ESPRIT were already proposed in the literature, based on specific subspace trackers. For example, the adaptive ESPRIT technique presented in [3] relies on an SVD updating algorithm [4], and that proposed in [5] is based on the rank-revealing URV decomposition [6]. The complexity of these methods is proportional to $O(n^2)$ operations for each time update, where n is the dimension of the data vectors. In [7], other adaptive ESPRIT algorithms were proposed for use with the Loraf [8] or the Bi-SVD subspace tracker [9]. In comparison with [3] and [5], the complexity of these algorithms is reduced to $O(nr^2)$ or $O(nr)$, where r is the dimension of the signal subspace (in practice $r << n$). Concurrently, the PAST algorithm is a very light and robust subspace tracker, which requires only $O(nr)$ operations at each time step. The PAST estimation consists in the iterative optimization of a specific cost function involving the estimated covariance matrix of the signal, in combination with a projection approximation hypothesis [10]. Relying on this very fast subspace tracker, the adaptive ESPRIT algorithm presented below reaches the linear complexity[1] $O(nr)$.

---

[1]In fact, the last step of this algorithm is the EVD of a $r \times r$ matrix, which involves $O(r^3)$ operations. Such a computation is also required in [5] and [7]. The latter proposes a fast eigenvalue tracking method.

This paper is organized as follows. Section 2 briefly describes the application of the ESPRIT method to the EDS model. Section 3 summarizes the PAST algorithm. Section 4 introduces our fast adaptive ESPRIT method, whose performance is compared to that of the reference algorithm [7] in section 5. Finally, section 6 summarizes the main conclusions of this paper.

2. ESTIMATION OF SIGNAL PARAMETERS VIA ROTATIONAL INVARIANCE TECHNIQUES

The noiseless EDS model defines the discrete signal $x(t)$ as

$$x(t) = \sum_{m=1}^{r} \alpha_m z_m^t$$

where r is the order of the model, $\alpha_m \in \mathbb{C}^*$ are the complex amplitudes, and $z_m \in \mathbb{C}^*$ denote the complex poles. The vector $\boldsymbol{x}(t) = [x(t - n + 1), \ldots, x(t)]^T$ (with $n \geq r$) belongs to the r-dimensional subspace spanned by the Vandermonde matrix

$$\boldsymbol{V} = \begin{bmatrix} 1 & \cdots & 1 \\ z_1 & \cdots & z_r \\ \vdots & \ddots & \vdots \\ z_1^{n-1} & \cdots & z_r^{n-1} \end{bmatrix}.$$

The poles can be calculated by exploiting the rotational invariance property of this subspace, referred to as the signal subspace. The classical way of obtaining a $n \times r$ matrix $\boldsymbol{U}$ which spans the signal subspace consists in computing the singular value decomposition (SVD) of a data matrix, or the eigenvalue decomposition (EVD) of a covariance matrix, which requires $O(n^3)$ operations.

Let $\boldsymbol{U}_\downarrow$ be the matrix extracted from $\boldsymbol{U}$ by deleting the last row. Similarly, let $\boldsymbol{U}_\uparrow$ be the matrix extracted from $\boldsymbol{U}$ by deleting the first row. Then consider the $r \times r$ matrix $\boldsymbol{\Phi} = (\boldsymbol{U}_\downarrow)^\dagger \boldsymbol{U}_\uparrow$ (where the symbol $\dagger$ denotes the Moore-Penrose pseudo-inverse). It can be shown that the eigenvalues of $\boldsymbol{\Phi}$ are exactly the complex poles of the signal. Therefore, the ESPRIT algorithm [2] consists in:

- computing $\boldsymbol{\Phi}$ (which requires $O(nr^2)$ operations),
- extracting the estimated poles $\hat{z}_m$ as the eigenvalues of $\boldsymbol{\Phi}$ (which can be achieved in $O(r^3)$ operations).

3. PROJECTION APPROXIMATION SUBSPACE TRACKING

The PAST algorithm for signal subspace tracking [10] requires only $O(nr)$ operations at each time step (instead of $O(n^3)$ for a full EVD).

Table 1. PAST algorithm

- - - - - - - - - - - - - - - - - - -

PAST initialization

$$\boldsymbol{U}(0) = \begin{bmatrix} \boldsymbol{I}_r \\ \boldsymbol{0}_{(n-r)\times r} \end{bmatrix}, \boldsymbol{Z}(0) = \boldsymbol{I}_r$$

- - - - - - - - - - - - - - - - - - -

for each time step do

PAST main section

input vector : $\boldsymbol{x}(t)$

$\boldsymbol{y}(t) = \boldsymbol{U}(t-1)^H \boldsymbol{x}(t)$

$\boldsymbol{h}(t) = \boldsymbol{Z}(t-1)\,\boldsymbol{y}(t)$

$\gamma(t) = \frac{1}{\beta + \boldsymbol{h}(t)^H \boldsymbol{y}(t)}$

$\boldsymbol{g}(t) = \gamma(t)\,\boldsymbol{h}(t)$

$\boldsymbol{Z}(t) = \frac{1}{\beta}\left(\boldsymbol{Z}(t-1) - \boldsymbol{g}(t)\,\boldsymbol{h}(t)^H\right)$

$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{U}(t-1)\,\boldsymbol{y}(t)$

$\boldsymbol{U}(t) = \boldsymbol{U}(t-1) + \boldsymbol{e}(t)\,\boldsymbol{g}(t)^H$

- - - - - - - - - - - - - - - - - - -

The dominant subspace estimation consists in minimizing the approximated scalar cost function

$$J'(\boldsymbol{U}(t)) = \sum_{i=1}^{t} \beta^{t-i} \|\boldsymbol{x}(i) - \boldsymbol{U}(t)\,\boldsymbol{y}(i)\|^2$$

where $\{\boldsymbol{x}(i)\}$ is a sequence of $n \times 1$ data vectors, β is the forgetting factor, and $\boldsymbol{y}(i) = \boldsymbol{U}(i-1)^H \boldsymbol{x}(i)$ (where the superscript ^{H} denotes the transpose conjugate of a matrix).

B. Yang showed that the solution $\boldsymbol{U}(t) \in \mathbb{C}^{n \times r}$ (with $r < n$) could be updated with few computations (see table 1)[2,3]. The update of $\boldsymbol{U}(t)$ is just a rank-1 modification:

$$\boldsymbol{U}(t) = \boldsymbol{U}(t-1) + \boldsymbol{e}(t)\,\boldsymbol{g}(t)^H \qquad (1)$$

where $\boldsymbol{e}(t)$ and $\boldsymbol{g}(t)$ are respectively n and r dimensional vectors.

4. ADAPTIVE ESPRIT ALGORITHM

The EDS model relies on the assumption of non varying signal parameters within the observation window. However, slow variations of these parameters can be taken into account in the analysis process. The main idea of this paper consists in tracking these time variations by applying the ESPRIT algorithm to the matrix $\boldsymbol{U}(t)$ estimated by the PAST algorithm (instead of a classical EVD or SVD). Thus, at each time step, the matrix $\boldsymbol{\Phi}(t)$ is given by the equation

$$\boldsymbol{\Phi}(t) = \boldsymbol{U}_\downarrow(t)^\dagger\,\boldsymbol{U}_\uparrow(t). \qquad (2)$$

The complexity of this semi-adaptive ESPRIT algorithm is $O(nr^2)$ because of the complete computation of the matrix $\boldsymbol{\Phi}(t)$. In fact, this time-consuming operation can be avoided by recursively updating $\boldsymbol{\Phi}(t)$. This fast update can be obtained after a few mathematical derivations. First, equation (1) yields

$$\boldsymbol{U}_\downarrow(t) = \boldsymbol{U}_\downarrow(t-1) + \boldsymbol{e}_\downarrow(t)\,\boldsymbol{g}(t)^H \qquad (3)$$

---

[2] $\boldsymbol{I}_r$ denotes the $r \times r$ identity matrix.

[3] Note that $\beta > 0$ and $\boldsymbol{h}(t)^H \boldsymbol{y}(t) \geq 0$, so that $\gamma(t)$ is always defined.

and

$$\boldsymbol{U}_\uparrow(t) = \boldsymbol{U}_\uparrow(t-1) + \boldsymbol{e}_\uparrow(t)\,\boldsymbol{g}(t)^H. \qquad (4)$$

Consider the $r \times r$ hermitian matrix

$$\boldsymbol{C}(t) = \boldsymbol{U}_\downarrow(t)^H \boldsymbol{U}_\downarrow(t). \qquad (5)$$

From now on, suppose that $\boldsymbol{U}_\downarrow(t)$ is always full rank (the rank deficiency case will be discussed later). Since in this case $\boldsymbol{C}(t)$ is non-singular, consider the $r \times r$ hermitian matrix

$$\boldsymbol{W}(t) = \boldsymbol{C}(t)^{-1} \qquad (6)$$

so that

$$\boldsymbol{U}_\downarrow(t)^\dagger = \boldsymbol{W}(t)\,\boldsymbol{U}_\downarrow(t)^H. \qquad (7)$$

Substituting equation (3) into equation (5) yields

$$\boldsymbol{C}(t) = \boldsymbol{C}(t-1) + \boldsymbol{F}(t)\,\boldsymbol{J}(t)\,\boldsymbol{F}(t)^H \qquad (8)$$

where $\boldsymbol{F}(t)$ is the $r \times 2$ matrix

$$\boldsymbol{F}(t) = \left[\ \boldsymbol{U}_\downarrow(t-1)^H \boldsymbol{e}_\downarrow(t)\ \middle|\ \boldsymbol{g}(t)\ \right] \qquad (9)$$

and $\boldsymbol{J}(t)$ is the 2×2 non-singular matrix

$$\boldsymbol{J}(t) = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & \|\boldsymbol{e}_\downarrow(t)\|^2 \end{array}\right]. \qquad (10)$$

Then the following matrix inversion lemma [11, pp. 18-19] will transform equation (8) into a recursion involving $\boldsymbol{W}(t)$.

Lemma 1 *Let $\boldsymbol{A}$ be a $r \times r$ non-singular complex matrix. Consider the $r \times r$ matrix $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{X}\,\boldsymbol{J}\,\boldsymbol{Y}$, where $\boldsymbol{X}$ is $r \times m$, $\boldsymbol{Y}$ is $m \times r$, and $\boldsymbol{J}$ is $m \times m$ and non-singular. Then $\boldsymbol{B}$ is non-singular if and only if $\boldsymbol{J}^{-1} + \boldsymbol{Y}\,\boldsymbol{A}^{-1}\,\boldsymbol{X}$ is non-singular, and in this case*

$$\boldsymbol{B}^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\,\boldsymbol{X}\left(\boldsymbol{J}^{-1} + \boldsymbol{Y}\,\boldsymbol{A}^{-1}\,\boldsymbol{X}\right)^{-1} \boldsymbol{Y}\,\boldsymbol{A}^{-1}.$$

Lemma 1 applied to equation (8) shows that $\boldsymbol{W}(t)$ satisfies

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) - \boldsymbol{S}(t)\,\boldsymbol{\Lambda}(t)\,\boldsymbol{S}(t)^H \qquad (11)$$

where $\boldsymbol{S}(t)$ is the $r \times 2$ matrix

$$\boldsymbol{S}(t) = \boldsymbol{W}(t-1)\,\boldsymbol{F}(t) \qquad (12)$$

and $\boldsymbol{\Lambda}(t)$ is the 2×2 matrix

$$\boldsymbol{\Lambda}(t) = \left(\boldsymbol{J}(t)^{-1} + \boldsymbol{S}(t)^H \boldsymbol{F}(t)\right)^{-1}. \qquad (13)$$

Note that lemma 1 also proves that $\boldsymbol{C}(t)$ is non-singular if and only if the 2×2 matrix $\boldsymbol{J}(t)^{-1} + \boldsymbol{S}(t)^H \boldsymbol{F}(t)$ is non-singular. Therefore, detecting the non inversibility of $\boldsymbol{J}(t)^{-1} + \boldsymbol{S}(t)^H \boldsymbol{F}(t)$ is a fast way of detecting the rank deficiency of $\boldsymbol{U}_\downarrow(t)$. In the following developments, the full-rank case will be presented first. After some calculations (see the appendix), the following lemma can be derived:

Lemma 2 *The matrix $\boldsymbol{U}_\downarrow(t)^\dagger$ satisfies the recursion*

$$\boldsymbol{U}_\downarrow(t)^\dagger = \boldsymbol{U}_\downarrow(t-1)^\dagger + \boldsymbol{R}(t)\,\boldsymbol{T}(t)^H \qquad (14)$$

where $\boldsymbol{R}(t)$ is the $r \times 2$ matrix

$$\boldsymbol{R}(t) = \boldsymbol{S}(t)\,\boldsymbol{\Lambda}(t) \qquad (15)$$

and $\boldsymbol{T}(t)$ is the $(n-1) \times 2$ matrix

$$\boldsymbol{T}(t) = \boldsymbol{e}_\downarrow(t)\left[\ 1\ \middle|\ 0\ \right] - \boldsymbol{U}_\downarrow(t-1)\,\boldsymbol{S}(t). \qquad (16)$$

Now, substituting equations (14) and (4) into equation (2) yields

$$\mathbf{\Phi}(t) = \mathbf{\Phi}(t-1) + \mathbf{P}(t)\,\mathbf{Q}(t)^H \qquad (17)$$

where $\mathbf{P}(t)$ is the $r \times 3$ matrix

$$\mathbf{P}(t) = \left[\ \mathbf{U}_\downarrow(t-1)^\dagger\, \mathbf{e}_\uparrow(t)\ \middle|\ \mathbf{R}(t)\ \right] \qquad (18)$$

and $\mathbf{Q}(t)$ is the $r \times 3$ matrix

$$\mathbf{Q}(t) = \left[\ \mathbf{g}(t)\ \middle|\ \mathbf{U}_\uparrow(t)^H \mathbf{T}(t)\ \right]. \qquad (19)$$

Note that equation (17) shows that the update of the matrix $\mathbf{\Phi}$ is a rank-3 modification. The whole processing requires only $O(nr)$ operations. Finally, the complete pseudo-code for the adaptive ESPRIT algorithm is given in table 2.

Table 2. Adaptive ESPRIT algorithm

PAST initialization (*cf.* table 1)

ESPRIT initialization

$\mathbf{W}(0) = \mathbf{I}_r,\ \mathbf{\Phi}(0) = \mathbf{U}_\downarrow(0)^\dagger \mathbf{U}_\uparrow(0)$

----

for each time step do

> **PAST main section** (*cf.* table 1)
>
> **ESPRIT main section**
>
> $\mathbf{F}(t) = \left[\ \mathbf{U}_\downarrow(t-1)^H \mathbf{e}_\downarrow(t)\ \middle|\ \mathbf{g}(t)\ \right]$ (9)
>
> $\mathbf{S}(t) = \mathbf{W}(t-1)\,\mathbf{F}(t)$ (12)
>
> $\mathbf{J}(t)^{-1} = \begin{bmatrix} -\|\mathbf{e}_\downarrow(t)\|^2 & 1 \\ 1 & 0 \end{bmatrix}$ from (10)
>
> $\mathbf{\Lambda}(t) = \left(\mathbf{J}(t)^{-1} + \mathbf{S}(t)^H \mathbf{F}(t)\right)^{-1}$ (13)
>
> $\mathbf{R}(t) = \mathbf{S}(t)\,\mathbf{\Lambda}(t)$ (15)
>
> $\mathbf{W}(t) = \mathbf{W}(t-1) - \mathbf{R}(t)\,\mathbf{S}(t)^H$ from (11)
>
> $\mathbf{T}(t) = \mathbf{e}_\downarrow(t)\left[\ 1\ \middle|\ 0\ \right] - \mathbf{U}_\downarrow(t-1)\,\mathbf{S}(t)$ (16)
>
> $\mathbf{P}(t) = \left[\ \mathbf{U}_\downarrow(t-1)^\dagger \mathbf{e}_\uparrow(t)\ \middle|\ \mathbf{R}(t)\ \right]$ (18)
>
> $\mathbf{Q}(t) = \left[\ \mathbf{g}(t)\ \middle|\ \mathbf{U}_\uparrow(t)^H \mathbf{T}(t)\ \right]$ (19)
>
> $\mathbf{U}_\downarrow(t)^\dagger = \mathbf{U}_\downarrow(t-1)^\dagger + \mathbf{R}(t)\,\mathbf{T}(t)^H$ (14)
>
> $\mathbf{\Phi}(t) = \mathbf{\Phi}(t-1) + \mathbf{P}(t)\,\mathbf{Q}(t)^H$ (17)
>
> $\{\hat{z}_m\}_{1 \le m \le r} = \mathrm{eig}\left(\mathbf{\Phi}(t)\right)$

If $\mathbf{U}_\downarrow(t)$ is rank deficient, $\mathbf{\Phi}(t)$ can no longer be updated with equation (17), since $\mathbf{C}(t)$ is singular. In this case, the pseudo-inverse of $\mathbf{U}_\downarrow(t)$ must be explicitly computed. In equation (7), $\mathbf{W}(t)$ is now defined as the pseudo-inverse of $\mathbf{C}(t)$:

$$\mathbf{W}(t) = \mathbf{C}(t)^\dagger.$$

This pseudo-inverse is obtained by computing the EVD of $\mathbf{C}(t)$, which can be achieved in $O(r^3)$ operations. Then $\mathbf{\Phi}(t)$ is given by equation (2), which requires $O(nr^2)$ operations. In the following time steps, $\mathbf{W}(t)$ and $\mathbf{\Phi}(t)$ are computed in the same way, while $\mathbf{C}(t)$ remains singular (note that $\mathbf{C}(t)$ can be updated using equation (8)). When $\mathbf{C}(t)$ becomes full rank again, the algorithm can switch back to the fully adaptive processing.

5. SIMULATION RESULTS

The test signal of Figure 1-a is a sum of $r = 4$ complex sinusoidal sources plus a complex white gaussian noise (the SNR is 5.7 dB). The frequencies of the sinusoids vary according to a jump scenario originally proposed by P. Strobach in the context of Direction Of Arrival estimation [7]: their values abruptly change at different time instants, between which they remain constant. Their variations are represented on Figure 1-b.

Figure 2-a shows the frequency tracking result, with parameters $n = 80$ and $\beta = 0.99$. The dotted line indicates the true frequency parameters, while the solid line indicates the estimated frequencies. Here, an orthonormal version of PAST [12] was used instead of the classical PAST algorithm. Indeed, OPAST outperforms PAST without increasing the computational cost, and it can be used without modifying the ESPRIT main section.

The performance of the subspace estimation is also analyzed in figure 2-b in terms of the maximum principal angle between the true dominant subspace of the data matrix (obtained via an exact singular value decomposition), and the estimated dominant subspace of the same data matrix (obtained with the tracker). This criterion was originally proposed by P. Comon and G.H. Golub as a measure of the distance between equidimensional subspaces [13].

It can be noticed that the adaptive ESPRIT algorithm robustly tracks the abrupt frequency variations. The time delay before convergence in transient regions is mainly due to the exponential forgetting nature of the analysis window and could be strongly shortened with a sliding window (as shown in [10]).

Finally, these results can be compared to that shown in figure 3, obtained with the $O(nr^2)$ adaptive ESPRIT algorithm [7] associated to the $O(nr^2)$ Bi-SVD1 subspace tracker [9] with the same parameters n and β. It can be seen that the performance is very similar to that shown in figure 2. However, the computational cost is an order of magnitude higher, and the faster $O(nr)$ adaptive ESPRIT algorithm proposed in [7], associated to the $O(nr)$ Bi-SVD3 subspace tracker [9], proved to be unstable on this test signal.
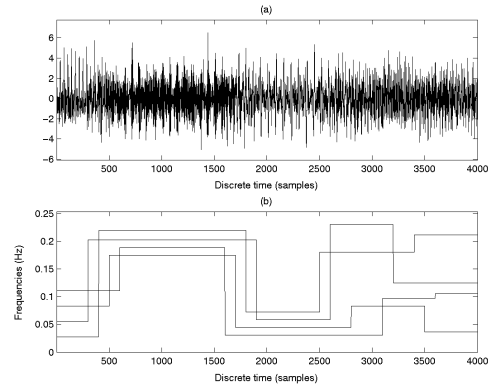

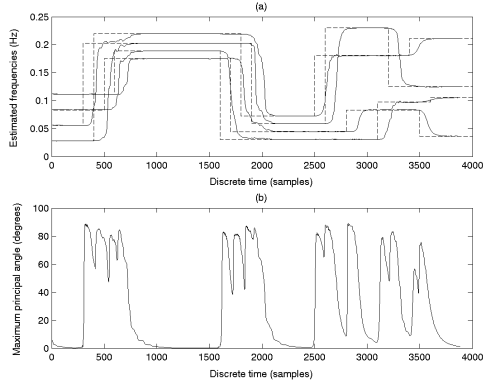**Fig. 1**. (a): Test signal; (b): Normalized frequencies of the sinusoids.

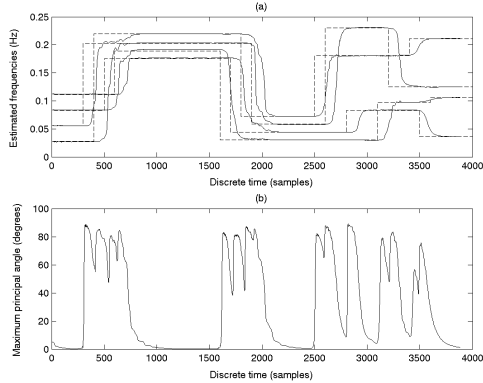**Fig. 2**. OPAST / ESPRIT tracking: (a): Frequency tracking; (b): Maximum principal angle trajectory.

Fig. 3. Bi-SVD1 / ESPRIT tracking: (a): Frequency tracking; (b): Maximum principal angle trajectory.

6. CONCLUSIONS

In this paper, an adaptive implementation of the ESPRIT algorithm for frequency estimation and tracking has been derived, relying on the PAST subspace tracker. This fast algorithm reaches the linear complexity $O(nr)$ (or $O(nr^2)$ in the rank deficiency case, which is seldom encountered in practice), except for the final extraction of the parameters, which requires the eigenvalue decomposition of a small matrix ($O(r^3)$ operations). The technique proved to robustly track abrupt frequency variations. The time delay before convergence in transient regions can be strongly reduced by using a sliding window version of the PAST algorithm (which was not presented here because of a lack of space). Future work will include a fast tracking of the eigenvalues of the small matrix, in order to make the overall complexity linear.

7. APPENDIX: PROOF OF LEMMA 2

Substituting equations (11) and (3) into equation (7) yields

$$
\begin{aligned}
\boldsymbol{U}_{\downarrow}(t)^{\dagger} &= \boldsymbol{U}_{\downarrow}(t-1)^{\dagger} - \boldsymbol{S}(t)\,\boldsymbol{\Lambda}(t)\,(\boldsymbol{U}_{\downarrow}(t-1)\,\boldsymbol{S}(t))^{H} \\
&+ \left(\boldsymbol{W}(t-1) - \boldsymbol{S}(t)\,\boldsymbol{\Lambda}(t)\,\boldsymbol{S}(t)^{H}\right)\boldsymbol{g}(t)\,\boldsymbol{e}_{\downarrow}(t)^{H}.
\end{aligned}
\tag{20}
$$

Note that equation (9) yields $\boldsymbol{g}(t) = \boldsymbol{F}(t)\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Therefore, equation (20) becomes

$$
\begin{aligned}
\boldsymbol{U}_{\downarrow}(t)^{\dagger} &= \boldsymbol{U}_{\downarrow}(t-1)^{\dagger} - \boldsymbol{S}(t)\,\boldsymbol{\Lambda}(t)\,(\boldsymbol{U}_{\downarrow}(t-1)\,\boldsymbol{S}(t))^{H} \\
&+ \boldsymbol{S}(t)\left(\boldsymbol{I}_2 - \boldsymbol{\Lambda}(t)\,\boldsymbol{S}(t)^{H}\,\boldsymbol{F}(t)\right)\begin{bmatrix} 0 \\ 1 \end{bmatrix}\boldsymbol{e}_{\downarrow}(t)^{H}.
\end{aligned}
\tag{21}
$$

In the second member of (21), note that equation (13) yields

$$
\boldsymbol{I}_2 - \boldsymbol{\Lambda}(t)\,\boldsymbol{S}^{H}(t)\,\boldsymbol{F}(t) = \boldsymbol{\Lambda}(t)\,\boldsymbol{J}(t)^{-1}.
\tag{22}
$$

Then, it can be seen that equation (10) yields

$$
\boldsymbol{J}(t)^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\tag{23}
$$

By substituting equations (22) and (23) into equation (21), equation (14) is finally derived.

8. REFERENCES

[1] R. Badeau, R. Boyer, and B. David, "EDS parametric modeling and tracking of audio signals," in *Proc. Int. Conf. on Digital Audio Effects DAFx-02*, september 2002.

[2] R. Roy and T. Kailath, "ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques," *IEEE Trans. on Acous., Speech, and Sig. Proc.*, vol. 37, no. 7, july 1989.

[3] M. Moonen, F.J. Vanpoucke, and E.F. Deprettere, "Parallel and adaptive high-resolution direction finding," *IEEE Trans. on Signal Proc.*, vol. 42, no. 9, september 1994.

[4] M. Moonen, P. Van Dooren, and J. Vandewalle, "An SVD updating algorithm for subspace tracking," *SIAM J. Matrix Ana. Appl.*, vol. 13, no. 4, pp. 1015–1038, 1992.

[5] K.J.R. Liu, D.P. O'Leary, G.W. Stewart, and Y-J.J. Wu, "URV ESPRIT for tracking time-varying signals," *IEEE Trans. on Signal Proc.*, vol. 42, no. 12, december 1994.

[6] G.W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. on Signal Proc.*, vol. 40, june 1992.

[7] P. Strobach, "Fast recursive subspace adaptive ESPRIT algorithms," *IEEE Trans. on Signal Proc.*, vol. 46, no. 9, 1998.

[8] P. Strobach, "Low-rank adaptive filters," *IEEE Trans. on Signal Proc.*, vol. 44, no. 12, pp. 2932–2947, december 1996.

[9] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," *IEEE Trans. on Sig. Proc.*, vol. 45, no. 5, may 1997.

[10] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. on Signal Proc.*, vol. 44, no. 1, january 1995.

[11] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1985.

[12] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Sig. Pr. Letters*, vol. 7, no. 3, 2000.

[13] P. Comon and G.H. Golub, "Tracking a few extreme singular values and vectors in signal processing," in *Proc. IEEE*, august 1990, vol. 78, pp. 1327–1343.