A Subspace Tracking Algorithm for Separating Partially Overlapping Data Packets

Mu Zhou and Alle-Jan van der Veen
Delft University of Technology, Department of Electrical Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands.
Email: m.zhou@tudelft.nl; a.j.vanderveen@tudelft.nl

***Abstract*—Consider a narrowband wireless scenario where a terminal equipped with an antenna array is receiving short packets of equal length sent by users from different directions and transmitted at random time points. These packets can be fully or partially overlapping in time. The partially overlapping packets, called asynchronous interferences, make the scenario nonstationary and thus need to be suppressed. Previously, we proposed a subspace intersection algorithm based on signed URV decompositions (SURV) working on block data from one time slot to solve this problem but it needs many steps. In this paper, we propose a one-step subspace tracking algorithm providing competitive performance at a much lower computational complexity like updating a QR decomposition. This algorithm makes use of the tracking ability of SURV to track the orthonormal basis and the rank of the asynchronous interference-free subspace as its principal subspace tracking version through two subspaces with positive and negative signatures.**

I. Introduction

Co-channel interference mitigation has been discussed for a long time in the field of array signal processing for applications such as cellular communication systems [1], [2], phased array-based radio telescopes [3], [4], radar systems [5], and signal acquisition in global positioning systems [6]. This topic brings a lot of algorithms which aim at the similar objective that they all look for weighting vectors to suppress the undesired interferences and recover the target signals. It is possible because in the typical noiseless model

$$\mathbf{X} = \mathbf{H}\mathbf{S}, \tag{1.1}$$

$\mathbf{X}$ is considered as the addition of source data in the subspaces spanned by the columns of $\mathbf{H}$, where $\mathbf{X}$ is the received data matrix, $\mathbf{H}$ is the channel matrix for sources, and $\mathbf{S}$ is the source data matrix. The weighting vectors can be obtained by computing vectors orthogonal to the interference subspace under various assumptions. These assumptions could be known signal or interference channel matrix, which can be collected from sample data when only signals or interferences present [7], or a large gap in power between signals and interferences so that the signal and interference subspaces can be separated through the principal or minor subspace [8] computed from matrix decompositions or their approximation or tracking algorithms [9]. However, in some scenarios the above assumptions or conditions are not available so these algorithms are not applicable.

For example, consider a narrowband wireless scenario where a (mobile) terminal equipped with an antenna array is receiving short packets of equal length sent by users from difference directions and transmitted at random time points. These packets can fully or partially overlap each other in time and thus can be treated as target packets or interference packets according to the detection time. The channel matrices of signals and interferences are unknown and the scenario is quite nonstationary such that it is hard for the previous algorithms to work well. Previously, we proposed a subspace intersection algorithm (SI+SURV) [10] working on block data collected from one time slot to find the asynchronous interference-free subspace (AIFS) and suppress the partially overlapping packets so that the followed source separation algorithms work on stationary data. This algorithm has two main steps, one to compute the principal subspace of the data matrix, and the other one to do subspace intersection based on signed URV decompositions (SURV) [9] on the projected matrix to compute the orthonormal basis and the rank of AIFS.

In this paper, we propose a subspace tracking algorithm which tracks the orthonormal basis and the rank of AIFS at a computational complexity of $O(M^2)$ per joint vector update like updating a QR decomposition, and provides a competitive performance against SI+SURV. This algorithm makes use of the tracking ability of SURV to track AIFS, not in the sense of "principal" or "minor" but in two subspaces with positive and negative signatures. Inside a time slot, this algorithm uses three sliding windows, one in the center with vectors of positive signatures and the other two with vectors of negative signature in the two ends of the time slot. The three sliding windows slide simultaneously with the time slot.

II. Data Model and Preliminary

We consider a wireless narrowband scenario where packets of equal length N_p are transmitted at random time and received by an antenna array. The received signals are expressed in a classic baseband data model.

$$\mathbf{x}(t) = \mathbf{H}(t)\mathbf{s}(t) + \mathbf{n}(t). \tag{2.1}$$

t denotes the time. $\mathbf{H}(t) = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{d(t)}] \in \mathbb{C}^{M\times d(t)}$ contains the channel vectors for each source. M equals to the number of antennas, and $d(t)$ equals to the number of sources. $\mathbf{x}(t) = [x_1^H(t), x_2^H(t), \cdots x_M^H(t)]^H \in \mathbb{C}^{M\times 1}$ is the received data vector with stacked output from the antennas. $\mathbf{s}(t) = [s_1^H(t), s_2^H(t), \ldots, s_{d(t)}^H(t)]^H \in \mathbb{C}^{d(t)\times 1}$ is the source data vector with stacked input data from the independent sources. $\mathbf{n}(t) \in \mathbb{C}^{M\times 1}$ is the white Gaussian noise vector

978-1-4673-1071-0/12/$31.00 ©2012 IEEE

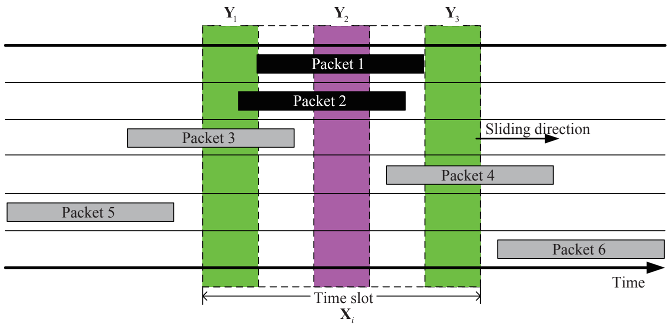

Fig. 2.1. One time slot in the data model.

with covariance $\mathbf{R}_n = \mathrm{E}(\mathbf{n}(t)\mathbf{n}(t)^H) = \sigma_n^2\mathbf{I}$. $\mathrm{E}(\cdot)$ is the expectation operator. σ_n^2 is the noise power.

In this paper, (2.1) is extended and modified as

$$\mathbf{x}(t) = \tilde{\mathbf{H}}(t)\mathbf{G}(t)(\tilde{\mathbf{s}}(t) \odot \boldsymbol{\varphi}(t)) + \mathbf{n}(t), \tag{2.2}$$

where $\odot$ is the Schur-Hadamard (pointwise multiplication) operator, $\mathbf{G}(t) = \mathrm{diag}\{g_1, g_2, \cdots, g_{d(t)}\} \in \mathbb{R}^{d(t)\times d(t)}$ contains the source power, and $\boldsymbol{\varphi}(t) = [\mathrm{e}^{-j2\pi\Delta f_1 t}, \mathrm{e}^{-j2\pi\Delta f_2 t}, \ldots, \mathrm{e}^{-j2\pi\Delta f_{d(t)} t}]^H \in \mathbb{C}^{d(t)\times 1}$ contains the Doppler phase shifts (Δf_k is Doppler frequency shift (DFS) for the k-th packet). For simplicity, we assume the added submatrices are always absorbed into $\mathbf{H}(t)$ and $\mathbf{s}(t)$, i.e., $\mathbf{H}(t) = \tilde{\mathbf{H}}(t)\mathbf{G}(t)$ and $\mathbf{s}(t) = \tilde{\mathbf{s}}(t) \odot \boldsymbol{\varphi}(t)$.

Sample the channel output from (2.1) at sample period $T_s = T/P$, where T is the symbol period and P is the oversampling ratio. A block data

$$\mathbf{X} = \mathbf{HS} + \mathbf{N} \in \mathbb{C}^{M\times PN_s} \tag{2.3}$$

is collected inside a given time interval, called a time slot of length N_s symbols long. The time slot can be set related to moving windows along the time line. Assume that d packets appear in the time slot (See Fig. 2.1). We have $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{PN_s}]$, $\mathbf{H} \in \mathbb{C}^{M\times d}$, $\mathbf{S} = [\mathbf{s}_1^H, \mathbf{s}_2^H, \ldots, \mathbf{s}_d^H]^H \in \mathbb{C}^{d\times PN_s}$ where $\mathbf{s}_k = [s_k(T_s), s_k(2T_s), \ldots, s_k(PN_sT_s)]$, and $\mathbf{N} = [\mathbf{n}(T_s), \mathbf{n}(2T_s), \ldots, \mathbf{n}(PN_sT_s)]$.

The interference packets for the k-th packet are defined as the other packets overlapping it. $\mathbf{H}_k$ is defined as the corresponding interference channel matrix for the k-th packet. $\mathbf{H}$ and $\mathbf{S}$ are all unknown. Our objective is to recover the k-th packet when it presents around the middle of the time slot by using the beamformer $\mathbf{w}_k \in \mathbb{C}^{M\times 1}$ such that

$$\hat{\mathbf{s}}_k = \mathbf{w}_k^H\mathbf{X}, \tag{2.4}$$

where $\hat{\mathbf{s}}_k$ is the estimate of $\mathbf{s}_k$. $\mathbf{w}_k$ is required to suppress the interference packets as deep as possible.

Signed URV decomposition [9]: Given two matrices $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathbb{C}^{M\times N}$, SURV implicitly computes

$$\mathbf{CC}^H = \mathbf{Y}_1\mathbf{Y}_1^H - \mathbf{Y}_2\mathbf{Y}_2^H = \mathbf{AA}^H - \mathbf{BB}^H \tag{2.5}$$

and the column subspace bases of the indefinite matrix $\mathbf{CC}^H$ from the compact factorization

$$[\overset{+}{\mathbf{Y}_1}\ \overset{-}{\mathbf{Y}_2}]\,\boldsymbol{\Theta}' = [\overset{d_1}{\mathbf{Q}_\mathbf{A}}\ \overset{d_2}{\mathbf{Q}_\mathbf{B}}]\,[\overset{\overset{+}{d_1}}{\mathbf{L}_\mathbf{A}}\ \overset{\overset{-}{d_2}}{\mathbf{L}_\mathbf{B}}], \tag{2.6}$$

where the sign $+$ and $-$ above matrices denote the positive and negative signatures of the corresponding columns, $[\mathbf{A}\ \mathbf{B}] = [\mathbf{Q}_\mathbf{A}\ \mathbf{Q}_\mathbf{B}]\,[\mathbf{L}_\mathbf{A}\ \mathbf{L}_\mathbf{B}]$, $\boldsymbol{\Theta}'$ is part of the $\mathbf{J}$-unitary matrix $\boldsymbol{\Theta}$ [11] for corresponding nonzero columns, $[\mathbf{L}_\mathbf{A}\ \mathbf{L}_\mathbf{B}] \in \mathbb{C}^{M\times M}$ is a lower triangular matrix, and $[\mathbf{Q}_\mathbf{A}\ \mathbf{Q}_\mathbf{B}] \in \mathbb{C}^{M\times M}$ is a unitary matrix. We call the subspace $\mathrm{ran}(\mathbf{Q}_\mathbf{B})$ or $\mathrm{ran}(\mathbf{B})$ "the negative subspace" and the subspace $\mathrm{ran}(\mathbf{A})$ "the positive subspace". $\mathbf{Q}_\mathbf{A}$ is the orthogonal complement of the negative subspace. $\mathrm{ran}(\mathbf{Q}_\mathbf{A}) = \mathrm{ran}(\mathbf{Q}_\mathbf{B})^\perp$. d_1 and d_2 are the dimensionalities of the positive subspace and the negative subspace, respectively. $M = d_1 + d_2$. SURV provides subspace estimates with good properties as

$$\mathrm{ran}(\mathbf{Q}_\mathbf{B}) \subset \mathrm{ran}(\mathbf{Y}_2). \tag{2.7}$$

SURV can be easily updated by adding or removing vectors from $\mathbf{Y}_1$ and $\mathbf{Y}_2$.

III. Asynchronous Interference-free Subspace Tracking

In [10], we proposed SI+SURV to find AIFS. SI+SURV is designed for block data and consists of many steps. It needs the help of rank tracking algorithms [9] to determine the start of the time slot. In this section, we propose a more computationally efficient tracking algorithm to track the orthonormal basis and the rank of AIFS (IFST, asynchronous interference-free subspace tracking). This algorithm can automatically determine the optimal output time of subspace and rank estimates.

IFST is designed from the special case that the target packets fully fill up the middle interval centered in the time slot and the remaining two intervals on both sides contain only the asynchronous interference packets. When the time slot slides along the time line, this special structure always exists for each packet and facilitates the suppression of asynchronous interference packets as the interference channel matrix can be obtained from the two side intervals. It is clear that we can define three separate sliding windows based on the three intervals. Our proposed tracking algorithm uses the data matrices of these windows defined as

- Time slot: $\mathbf{X}_i = [\mathbf{x}_{i-PN_s/2+1}, \ldots, \mathbf{x}_{i+PN_s/2}] \in \mathbb{C}^{M\times PN_s}$, $i \geq PN_s/2$, $N_s = N_{y_1} + N_p + N_{y_3}$, $N_{y_1} = N_{y_3}$;
- Left window: $\mathbf{Y}_1 = [\mathbf{x}_{i-PN_s/2+1}, \ldots, \mathbf{x}_{i-PN_s/2+PN_{y_1}}] \in \mathbb{C}^{M\times PN_{y_1}}$;
- Center window: $\mathbf{Y}_2 = [\mathbf{x}_{i-PN_{y_2}/2+1}, \ldots, \mathbf{x}_{i+PN_{y_2}/2}] \in \mathbb{C}^{M\times PN_{y_2}}$;
- Right window: $\mathbf{Y}_3 = [\mathbf{x}_{i+PN_s/2-PN_{y_3}+1}, \ldots, \mathbf{x}_{i+PN_s/2}] \in \mathbb{C}^{M\times PN_{y_3}}$.

i is the sample index which also denotes the center of the time slot, and $\mathbf{Y}_k$ and PN_{y_k} are the data matrix and the length of the k-th window. From these matrices, we form a compact SURV

$$[\overset{+}{\alpha\mathbf{Y}_2}\ \overset{-}{\mathbf{Y}_1}\ \overset{-}{\mathbf{Y}_3}\ \overset{-}{\gamma\mathbf{I}}]\,\boldsymbol{\Theta}' = [\overset{d_{y_2}}{\mathbf{Q}_\mathbf{A}}\ \overset{M-d_{y_2}}{\mathbf{Q}_\mathbf{B}}]\ [\overset{\overset{+}{d_{y_2}}}{\mathbf{L}_\mathbf{A}}\ \overset{\overset{-}{M-d_{y_2}}}{\mathbf{L}_\mathbf{B}}], \tag{3.1}$$

where $\alpha = 1$ and $\gamma = \alpha\beta\sigma_n(\sqrt{PN_{y_2}} + \sqrt{M})$. The value of β is chosen to give no "false alarm" [9]. This single SURV implies two steps:

1) Placing $\mathbf{Y}_1$ and $\mathbf{Y}_3$ in the negative subspace makes the positive subspace asynchronous interference-free (orthogonal to the subspace spanned by the interference channel matrix) [10].
2) Placing $\gamma\mathbf{I}$ in the negative subspace reduces $\mathbf{A}$ to full column rank d_{y_2} (roughly the rank of $\mathbf{Y}_2$). d_{y_2} is the estimated number of target packets. The columns of $\mathbf{Q_A}$ approximately spans AIFS. It is guaranteed by the property that in stationary cases without any interference, using a long sliding window, $\mathrm{ran}(\mathbf{A})$ and $\mathrm{ran}(\mathbf{B})$ asymptotically become orthogonal to each other as a function of the noise power σ_n^2. Similar algorithms can be found in literature such as spherical averaging algorithms designed for cases with white Gaussian noise [12].

The columns of $\mathbf{Q_A}$ are the orthonormal basis of AIFS and d_{y_2} is the rank of AIFS. The output time of the AIFS estimates can be set to $PN_p/4$ samples from the last rank change. The updating of (3.1) is direct and simple as in [9]. There are three incoming and outgoing vectors for the three windows $\mathbf{Y}_k$ per joint vector update when sliding the time slot position by 1 (i.e., $i = i + 1$).

The computational complexity of the proposed algorithm is of $O(M^2)$ per joint vector update. Two $M \times M$ matrices $[\mathbf{Q_A}\ \mathbf{Q_B}]$ and $[\mathbf{L_A}\ \mathbf{L_B}]$ need to be stored, which is the same to the principal subspace tracking version [9].

Remark 1: Larger α is also possible, e.g. $\alpha = 8$, with which our proposed algorithm shows similar behavior to SI+SURV. It is also interesting to see that different α can be applied according to different purposes in selecting packets, which will be discussed in incoming papers.

Remark 2: Slight tolerance on the packet length is also allowed, which could be between $(0.5N_p, N_p]$.

IV. Simulation Results

In this section, we compare the performance of our proposed tracking algorithm (IFST) with SI+SURV and SVD.

The defined data model can be found in a real scenario [13], [14] where a low earth orbit satellite is collecting messages (or packets) sent out by ships at sea in thousands of self-organized ground cells. Taking this scenario as background, we choose the carrier frequency 162.025MHz and modulation scheme 9.6kbps QPSK. Consider 5 packets (see Fig. 4.1) in the scenario. Parameters of these packets are listed in Table 1. A linear antenna array with $M = 10$ elements spaced at half wavelengths is used. We set $N_p = 256$ symbols, $N_s = 512$ symbols, $N_{y_1} = N_{y_3} = 128$ symbols, $N_{y_2} = 64$ symbols, $f_s = 4.8k\mathrm{Hz}$, $P = 1$, $\beta = 1.24$. We use the algebraic constant modulus algorithm (ACMA) to separate user data from the data matrix preprocessed by the compared algorithms. The data for ACMA is 80 samples (Assume $d \leq 8$ in one time slot) in the middle of the time slot.

Signal and interference powers are defined for every symbol. The signal-to-interference ratio SIR $:= 10\log_{10}(\sigma_s^2/\sigma_f^2)$, and SNR $:= 10\log_{10}(\sigma_s^2/\sigma_n^2)$, where σ_s^2 and σ_f^2 are the symbol power of the signals and interferences, respectively. All packets have equal power. We number all packets in the simulation by index k, $1 \leq k \leq 5$. The performance measure is the residual signal-to-interference-plus-noise ratio (SINR) of the output beamformers. For every time slot $\mathbf{X}_i$, the output performance for the k-th packet is found as

TABLE 4.1
PARAMETERS OF PACKETS.

	DOA	DFS(kHz)	Start i	Output i
Packet 1	-10°	3.7	522	660
Packet 2	30°	-3.7	542	660
Packet 3	10°	0.1	702	830
Packet 4	-30°	0.5	882	1025
Packet 5	40°	1.2	902	1025

$$\mathrm{sinr}(\mathbf{h}, \mathbf{w}) := \frac{\mathbf{w}^H(\mathbf{h}\mathbf{h}^H)\mathbf{w}}{\mathbf{w}^H(\bar{\mathbf{H}}\bar{\mathbf{H}}^H - \mathbf{h}\mathbf{h}^H + \sigma_n^2\mathbf{I})\mathbf{w}}, \quad (4.1)$$

$$\mathrm{SINR}(\bar{\mathbf{H}}_k, \mathbf{W}_i) := \max(\mathrm{sinr}(\mathbf{h}_k, \mathbf{w}_{i1}), \ldots, \mathrm{sinr}(\mathbf{h}_k, \mathbf{w}_{id_{y_2}})), \quad (4.2)$$

where $\bar{\mathbf{H}}_k = [\mathbf{h}_k\ \mathbf{H}_k]$ and $\mathbf{W}_i = [\mathbf{w}_{i1}, \mathbf{w}_{i2}, \ldots \mathbf{w}_{id_{y_2}}]$ is the output beamformer matrix for $\mathbf{X}_i$.

In the simulation, the same time slot is used for all algorithms. IFST uses all three windows and keeps tracking on them. SI+SURV uses the center window $\mathbf{Y}_2$ to do rank tracking and then subspace intersection (with $\alpha = 4$) on the entire time slot. SVD uses only the center window $\mathbf{Y}_2$. For initialization of IFST, a buffer of $2PN_p + 10$ samples long is put before the start of the first packet.

Fig. 4.2 shows the rank estimates and SINR of the beamformers for packets in one run. The results in time interval $i \in [450, 1200]$ are shown and for the beamformers, SINR is only shown for the time interval where the corresponding target packet is present. Fig. 4.2(a-1), Fig. 4.2(b-1) and Fig. 4.2(c-1) show the rank estimates (blue line) and the real number of packets (or real rank) d (red line) along the time line. IFST gives correct rank estimates around the middle of target packets, which means IFST can automatically synchronize to the target packets. SI+SURV gives rank estimates more close to the real rank such that it works on any time point but it may provide subspaces with a higher rank. In practice, IFST detects stationary intervals from the rank estimates and outputs the corresponding subspace estimates, while SI+SURV outputs subspace estimates at time determined by the rank estimates from another methods, such as SURV or SVD.

Fig. 4.2(a-2), Fig. 4.2(b-2) and Fig. 4.2(c-2) show the corresponding SINR of the beamformers for packets. IFST and SI+SURV give much better results than SVD because they are designed for finding AIFS. It is seen that IFST gives performance as good as SI+SURV at a much lower computational complexity as it is a tracking algorithm. The dash line is the optimal antenna gain for 10 antennas.

In order to compare the averaged SINR, we do Monte Carlo runs and collect the SINR output at sample indices $i = [660, 660, 830, 1025, 1025]$ (see Table 4.1) which are around the center of the corresponding packets and for IFST, about 50 samples ($\approx PN_p/4$) from the last rank change.

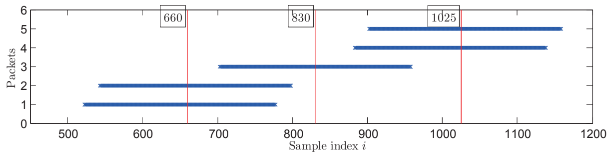

Fig. 4.1. Packet position.

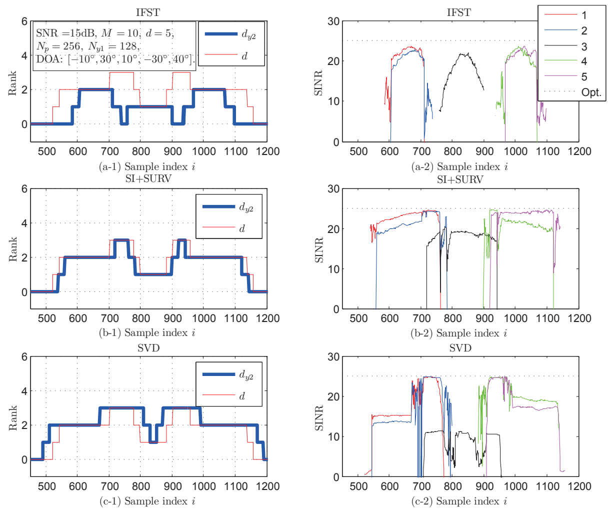

Fig. 4.2. The rank estimates and SINR of the beamformers for packets in one run.

Fig. 4.3 shows the averaged SINR of the beamformers for packets as a function of SNR after 10,000 Monte Carlo runs. The SINR results from standard minimum mean square error algorithm on known $\mathbf{S}$, $\mathbf{\Phi}$ and start sample indices are plotted as an upper bound for the highest performance we can achieve. The dotted line is the optimal antenna gain for 10 antennas. It is seen that IFST gives quite good and even better performance than SI+SURV. The gap between the upper bound and IFST as well as SI+SURV is mainly caused by the short packet length N_p, although their performance are good enough for decoding packets. Subspace estimates from SVD show poor suppression of interferences.

V. Conclusion

We proposed an efficient subspace tracking algorithm for separating partially overlapping data packets. This algorithm tracks the orhonormal basis and the rank of the asynchronous interference-free subspace for the target packets at a complexity of $O(M^2)$ per joint vector update, much lower compared with the previous algorithm working on block data and consisting of many steps. This algorithm can automatically determine the correct output time of the subspace estimates. The algorithm was shown to give quite good subspace estimates compared with the previous algorithm.

Acknowledgment

This work was supported in part by the project BDREAMS at Delft University of Technology in Netherlands, and China Scholarship Council from P. R. China.

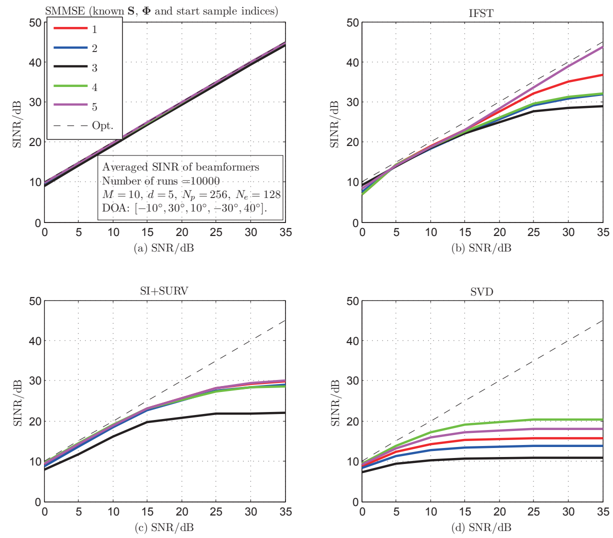

Fig. 4.3. Averaged SINR of the beamformers for packets as a function of SNR.

References

[1] J. Andrews, W. Choi, and R. Heath, "Overcoming interference in spatial multiplexing MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 95–104, Dec. 2007.

[2] S. Karimifar and J. Cavers, "Achieving high-capacity narrowband cellular systems by means of multicell multiuser detection," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 945–953, Mar. 2008.

[3] J. Raza, A.-J. Boonstra, and A. van der Veen, "Spatial filtering of RF interference in radio astronomy," *IEEE Signal Process. Lett.*, vol. 9, no. 2, pp. 64–67, Feb. 2002.

[4] S. Ellingson and G. Hampson, "A subspace-tracking approach to interference nulling for phased array-based radio telescopes," *IEEE Trans. Antennas Propag.*, vol. 50, no. 1, pp. 25–30, Jan. 2002.

[5] M. Zhou and A. van der Veen, "Improved blind separation algorithm for overlapping secondary surveillance radar replies," in *Proc. IEEE CAMSAP*, San Juan, Puerto Rico, Dec. 2011, pp. 181–184.

[6] Y. Morton, M. Miller, J. Tsui, D. Lin, and Q. Zhou, "GPS civil signal self-interference mitigation during weak signal acquisition," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5859–5863, Dec. 2007.

[7] R. Behrens and L. Scharf, "Signal processing applications of oblique projection operators," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1413–1424, Jun. 1994.

[8] H. Subbaram and K. Abend, "Interference suppression via orthogonal projections: a performance analysis," *IEEE Trans. Antennas Propag.*, vol. 41, no. 9, pp. 1187–1194, Sep. 1993.

[9] M. Zhou and A.-J. van der Veen, "Stable subspace tracking algorithm based on signed URV decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 2720–2723.

[10] ——, "Improved subspace intersection based on signed URV decomposition," in *Proc. of the Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, California, USA, Nov. 2011.

[11] A.-J. van der Veen, "A Schur method for low-rank matrix approximation," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 139–160, 1996.

[12] D. Rabideau, "Fast, rank adaptive subspace tracking and applications," *IEEE Trans. Signal Process.*, vol. 44, no. 9, pp. 2229–2244, Sep. 1996.

[13] M. A. Cervera, A. Ginesi, and K. Eckstein, "Satellite-based vessel automatic identification system: A feasibility and performance analysis," *Int. J. Satell. Commun. Network.*, 2009.

[14] M. Zhou, A.-J. van der Veen, and R. van Leuken, "Multi-user LEO-satellite receiver for robust space detection of AIS messages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012.