PARTIAL UPDATE ADAPTIVE TRANSMIT BEAMFORMING WITH LIMITED FEEDBACK

Eduardo Zacarías B., Stefan Werner and Risto Wichman

Signal Processing Laboratory
Helsinki University of Technology
P.O.Box 3000/FIN-02015 HUT

ABSTRACT

This work proposes a new partial update filtering technique tailored for adaptive transmit beamforming with low feedback rate. A signal dependent selection rule is derived that singles out one component of the beamforming vector to be updated. This provides an efficient way to perform the update, which escalates the performance with the number of available feedback bits without increasing the computational complexity. An operation count for the proposed algorithm and existing solutions is provided. Simulations show that the proposed scheme outperforms the tracking capabilities of existing codebook solutions in low mobility scenarios, while having comparable bit error probability performance and a computational complexity reduction of 90%.

1. INTRODUCTION

In this paper we consider systems employing adaptive transmit beamforming with low feedback rate and a single receive antenna. The goal of the transmitter is to adapt the transmission on each of its antennas in order to maximize the received signal power. However, it can only rely on limited information fed back by the receiver. This setup is appropriate for wireless communications when the channel does not have any long-term structure, e.g., in non line of sight scenarios where received signals may have large angular spreads.

The limited number of available feedback bits prevents a solution where the receiver signals back an accurate (i.e., high precision) estimate of the optimal beamforming vector to the transmitter. As a consequence, alternative methods like signed-gradient [1][2][3] and codebook [4] solutions have been proposed. However, the computational complexity of these solutions grows exponentially with the number of feedback bits employed (codebook size or number of stochastic search directions).

In order to lower the computational complexity associated with the search for an update to the complete beamforming vector, we employ a signal dependent criterion to choose one coefficient that is to be updated. Our idea stems from that of partial update (PU) adaptive filters, and allows to approach the performance of the full vector update at a lower computational complexity cost.

The idea of PU has been applied to various adaptive filtering applications [5][6][7], where one (in general a reduced group) of the filter coefficients is updated in order to decrease the computational complexity. However, the transmit beamforming application considered in this paper differs from those in [5][6][7], and therefore it is not possible to apply the conventional PU concept directly.

This work was partially funded by the Academy of Finland, SMARAD Center of Excellence and Grant 205703

In this paper we extend the PU ideas to enable a distributed updating scheme for transmit beamforming. We consider a signal dependent coefficient selection criterion that increases the adaptation capabilities of the algorithm when compared with a trivial round robin update, at the expense of some feedback overhead. Our proposed formulation indeed extends the PU principles to a different setup: the system under consideration is distributed, as opposed to those of [6][7] where the optimal filter coefficients are not available. Furthermore, only the receiver has access to the optimum solution and informs the transmitter which coefficient to update and how to do it. While [6][7] conclude that the coefficient to be updated is the one associated to the strongest input, our system equations lead to the idea that the one with the largest combination of error and channel power must be updated (note that the channel power would not be available in [6][7]).

2. SYSTEM MODEL

Our system consists of a transmitter equipped with N antennas and a single-antenna receiver. The receiver sends messages of n_b bits during each block of L channel samples. Let k denote the sample index and l the block index. The available signal at the receiver is:

$$y(k) = x(k)\mathbf{h}(k)\mathbf{w}(l) + n(k) \tag{1}$$

where $x(k)$ is the complex-valued symbol sent at sample time k, chosen independently of the rest of the variables, $\mathbf{h}(k) = [h_1(k) \dots h_N(k)] \in \mathbb{C}^{1\times N}$ is the channel vector, $\mathbf{w}(l) = [w_1(l) \dots w_N(l)]^T \in \mathbb{C}^N$ is the transmit filter or beamforming vector subject to a normalized total transmit power constraint $||\mathbf{w}(l)|| = 1$, and $n \in \mathbb{C}, n \sim \mathcal{N}(0, \sigma^2)$ is the additive noise at the receiver.

The goal of the transmit beamformer is to maximize the instantaneous received signal-to-noise ratio (SNR). However, $\mathbf{w}$ can only be updated once per block, based on the feedback information. Due to an inherent delay in the feedback channel, the adaptation of $\mathbf{w}$ is forced to be based on the feedback message produced in the previous block. Thus, the cost function is written in terms of the last channel sample of the previous block:

$$J(\mathbf{w}, \mathbf{h}) = |\mathbf{h}(lL-1)\mathbf{w}(l)|^2 \tag{2}$$

Note that the value of $\mathbf{w}(l)$ needs to be known to transmitter and receiver simultaneously.

The optimal $\mathbf{w}(l)$ maximizing (2) is $\mathbf{u}_\delta(l) = e^{j\delta}\ \mathbf{h}(lL-1)^\dagger / ||\mathbf{h}(lL-1)||$, where † denotes Hermitian transpose and δ is an arbitrary phase rotation. We choose δ so that the first component of $\mathbf{u}_\delta(l)$ is real-valued, and thus the optimal solution is:

$$\mathbf{v}(l) = e^{j\angle h_1(lL-1)} \frac{\mathbf{h}(lL-1)^\dagger}{||\mathbf{h}(lL-1)||} \tag{3}$$

where $\angle(\cdot)$ is the argument function.

We are interested in systems where the number of feedback bits n_b is small compared to the number of bits required to quantize the optimal beamforming vector. That is, if n_q bits are employed for quantizing each of the N components of the optimal filter, we assume $n_b \ll n_q N$. The next section proposes a novel solution that approaches (3) with a limited number of bits n_b.

3. THE PARTIAL UPDATE BEAMFORMING (PUB) ALGORITHM

This section proposes a partial update beamformer (PUB), which adjusts a single antenna weight at each update instant. The update made at the transmitter side is based on the information available at the receiver and transmitted through a low-rate feedback channel, assumed error free. In the following subsections we first consider a coefficient selection criterion, tailored for the application at hand, that singles out which beamformer weight to update. Thereafter, the coefficient update strategy and the structure of the feedback message are discussed. Finally, a discussion on the computational complexity of the proposed scheme is provided, and compared to that of an existing solution based on vector quantization.

3.1. Coefficient selection rule

At any given update instance, the idea is to update a single coefficient of the vector to reduce the overall complexity and the length of the feedback message. The simplest scheme is to use a fixed sequence to determine which coefficients to update each block. However, this does not guarantee that the partial update is optimal, in the sense of maximizing the cost function. For moderate values of N, this is not a big problem if the coefficients of $\mathbf{h}$ evolve slowly and at a similar rate. The h_i will have similar variation rates if there is correlation between them, which may be due to insufficient antenna spacing or lack of scatterers in the propagation environment. On the other hand, they will evolve slowly if the Doppler frequency is small (e.g., due to the receiver moving at low speeds). In all other cases it is beneficial to update the coefficient that produces the largest decrease in error as will be shown below.

Consider the error of the received signal:

$$\begin{aligned} e(k) = y_{opt}(k) - y(k) &= x(k)\mathbf{h}(k)[\mathbf{v}(l) - \mathbf{w}(l)] \\ &= x(k)\sum_{i=1}^{N} h_i(k)[v_i(l) - w_i(l)] \end{aligned} \tag{4}$$

Note that the coefficient-error terms in (4) are weighted by the channel magnitudes. Applying the triangular inequality bound in (4), we get

$$|e(k)| \leq |x(k)| \sum_{i=1}^{N} |h_i(k)||v_i(l) - w_i(l)| \tag{5}$$

As can be seen from (5), the bound is made up from one contribution per weight. Therefore, if only a subset of the N weights can be modified at the update instance, the best choices for reducing the error are those weights associated to the dominant terms in (5).

Consequently, the component w_a to be updated is the one associated with the index a obtained from

$$a = \underset{i}{\operatorname{argmax}} \left\{ |h_i(lL-1)|^2 |v_i(l) - w_i(l)|^2 \right\} \tag{6}$$

where the last sample of the block is used to make the decision.

Approaching the problem from a different point of view, we can also consider the gradient of $|e(k)|^2$. For a given weight $w_m(l)$ we have

$$\frac{\partial |e(k)|^2}{\partial w_m^*(l)} = |x(k)|^2 h_m(k) \sum_{i=1}^{N} [w_i(l) - v_i(l)]^* h_i^*(k) \tag{7}$$

It can be seen from (7) that the gradient is proportional to the term $\sum_{i=1}^{N}[w_i(l) - v_i(l)]^* h_i^*(k)$. Therefore, reducing this sum through the update of the chosen weight makes the gradient uniformly smaller. Applying the triangular inequality leads us to (6).

We note that the selection criterion is different than the existing M-Max criterion [6], where the coefficient to be updated corresponds to the input with the largest power. In this case, it would mean adjusting w_a with $a = \operatorname{argmax}_i(|h_i|)$, regardless of the difference between w_a and its optimum value. This is due to the fact that the optimum values are not available in the M-Max setup.

3.2. Single coefficient update

There are two approaches for updating the chosen coefficient. The first one is recursive, where the new value is computed based on the previous value and the feedback message. The second approach is non-recursive and computes a new value, disregarding the previous value. The signed gradient [1] algorithm implements recursive update by perturbing all the weights simultaneously. The codebook solutions [4] produce a non-recursive update to all the coefficients at the same time. We propose a non-recursive scheme for a single coefficient by quantizing the magnitude and phase separately, and sending the indexes of the quantized quantities as part of the feedback message. The update is given by

$$w_a(l+1) = \mathrm{Q}_\rho \left\{ |v_a(l)| \right\} e^{j\mathrm{Q}_\phi\{\angle v_a(l)\}} \tag{8}$$

where $\mathrm{Q}_q\{\cdot\}$ denotes scalar uniform quantization with q bits. Note that the first weight will use $\phi + \rho$ bits for quantizing the magnitude and will not be updated in phase, as we have chosen it to be real-valued, see (3).

Due to the norm-one constraint of vector $\mathbf{w}$, updating the magnitude of one element will affect all the other magnitudes. We propose to update $|w_{i \neq a}(l+1)|$ after applying (8), by scaling the corresponding magnitudes and distributing the "remaining norm" according to the allocations prior to the update. In other words, we get:

$$|w_i(l+1)|^2 = \frac{1 - |w_a(l+1)|^2}{1 - |w_a(l)|^2} |w_i(l)|^2 \quad \forall i \neq a \tag{9}$$

It can be verified that the norm constraint $||\mathbf{w}|| = 1$ is satisfied after the update of (8) and the scalings of (9) have taken place.

3.3. Feedback message structure

In this section we describe the structure of the feedback message with which the transmitter updates its beamforming vector. Let ρ and ϕ be the number of bits used to quantize the squared magnitude (power) and phase of $v_a(l)$, for $a > 1$. For $a = 1$, $\rho + \phi$ bits are used for power only, as w_1 is always real-valued. We denote the sets of quantized values as $\mathcal{M} = \{p_i\}_{i=1}^{2^\rho}$, $\mathcal{A} = \{\theta_i\}_{i=1}^{2^\phi}$ and $\mathcal{M}_1 = \{p'_i\}_{i=1}^{2^{\rho+\phi}}$. Further, assume that the indexes of the quantized values are r, s and t, i.e., $\mathrm{Q}_\rho\{|v_{a>1}(l)|\} = p_r \in \mathcal{M}$, $\mathrm{Q}_\phi\{\angle v_{a>1}(l)\} = \theta_s \in \mathcal{A}$ and $\mathrm{Q}_{\rho+\phi}\{|v_1(l)|\} = p'_t \in \mathcal{M}_1$.

The PUB feedback message is then defined:

$$\mathbf{b}_{\mathrm{R-PUB}}(l+1) = \begin{cases} [\mathcal{B}(a)\ \mathcal{B}(r)\ \mathcal{B}(s)] & a > 1 \\ [\mathcal{B}(a)\ \mathcal{B}(t)] & a = 1 \end{cases} \tag{10}$$

where $\mathbf{b}(l+1)$ is the feedback message that will be used by the transmitter at block $l+1$, $\mathcal{B}(\cdot)$ denotes the binary representation of an integer, and R-PUB refers to the PUB algorithm that uses the ranking function in (6), as opposed to the sequential counterpart, denoted S-PUB hereafter, for which the weights are updated in a round robin fashion. If n_b is the same for ranked and sequential versions, the number of bits used for phase and power must be different, since S-PUB does not signal the index a. Thus $\phi+\rho = n_b$ for S-PUB, while $\lceil \log_2(N) \rceil + \phi + \rho = n_b$ for R-PUB.

Note that the S-PUB is just a natural extension of the standard WCDMA closed loop modes [8], and is defined here for comparison purposes only. In the following, the acronyms PUB and R-PUB are used interchangeably.

The signaling overhead associated with the index a decreases the available bits for quantization by $\lceil \log_2(N) \rceil$. However, in some scenarios this loss is compensated by the faster adaptation capabilities, resulting in net performance gains. This is illustrated in Section 4. Methods for eliminating the feedback overhead are under investigation.

3.4. Computational Complexity

In this section we give an operation count for the PUB algorithm, assuming that the elements of the channel are known in polar form. At a given block index l, the optimal vector $\mathbf{v}(l)$ is built from $\mathbf{h}(lL-1)$, the ranking function of (6) is computed for each weight, the optimal squared modulus and phase are determined and quantized, and the respective indexes are obtained. Finally, the scaling factor of (9) is computed and applied to the other weights. The updated magnitude is computed from the quantized optimal squared modulus.

Let $O_{\mathrm{R-PUB}}$, $O_{\mathrm{S-PUB}}$ and O_{CBBF} be the number of real operations required by R-PUB, S-PUB and the codebook beamforming algorithm [4] (CBBF). Discarding the conversion from integer to binary necessary to build $\mathbf{b}(l)$, the operation count is

$$\begin{aligned} O_{\mathrm{R-PUB}} &= 10N\,\text{r.m.} + (5N+2)\,\text{r.a.} + 2\,\text{r.d.} + 2\,\text{sqrt} \\ O_{\mathrm{S-PUB}} &= (3N-5)\,\text{r.m.} + 2N\,\text{r.a.} + 2\,\text{r.d.} + 2\,\text{sqrt} \\ O_{\mathrm{CBBF}} &= 2^{n_b}[(3N+2)\,\text{r.m.} + (3N-1)\,\text{r.a.}] \end{aligned} \tag{11}$$

where r.m., r.a., r.d. and sqrt denote real-valued multiplications, additions, divisions and square roots, respectively. Note that O_{CBBF} is based on choosing the vector that maximizes the cost function in (2), which is not always the one with minimum Euclidean distance to the optimal vector $\mathbf{v}(l)$. We have neglected the cost of finding the maximum among the candidate values of the cost function in (2).

Thus, the PUB algorithm has complexity $\mathcal{O}(N)$ regardless the value of n_b. In contrast, CBBF results in complexity $\mathcal{O}(N2^{n_b})$. For example, at $n_b = 6$, $\mathcal{O}_{\mathrm{R-PUB}} = 0.016\mathcal{O}_{\mathrm{CBBF}}$. Furthermore, for $N = 4$ transmit antennas, the ratio of operation counts (discarding the 2 roots and 2 divisions) is $O_{\mathrm{R-PUB}}/O_{\mathrm{CBBF}} = 62/1600 \approx 0.039$, and we can say that in this scenario the PUB algorithm achieves a complexity reduction of about 90%.

4. SIMULATIONS

The performance of the PUB algorithm was evaluated in a system consisting of $N = 4$ transmit antennas and one receive antenna. For comparison purposes, the codebook solution of size 64 ($n_b = 6$) proposed in [4] was implemented, based on the vectors made publicly available by one of the authors. Note that single bit algorithms like [1] require to be extended to bit rates (per block) higher than one and have thus been excluded from the simulations.

The channel was modeled as Rayleigh fading with propagation paths spatially correlated according to the covariance matrix $\mathbf{R}$ specified in [9] for $\lambda/2$ antenna spacing and 3GPP cases 1 (uncorrelated), 2 and 4. Three different environments were evaluated with the associated condition numbers of $\mathbf{R}$ given by $\kappa = \lambda_{\max}/\lambda_{\min}(\mathbf{R}) = 1$, 7537 and 37, respectively. The carrier frequency used was $f_c = 2.1$ GHz. Each transmitted block contained $L = 160$ QPSK modulated symbols, and the transmit beamformer was updated at the end of each block. The channel is sampled L times per block.

The uncorrelated scenario produces the vector which is the most difficult to track, due to abrupt phase changes of the $h_i(l)$ (up to 1 radian even at low speeds). As correlation increases, the changes from $\mathbf{h}(l)$ to $\mathbf{h}(l+1)$ become smoother, resulting in better tracking performance.

Figure 1 shows the upper bounds for the mean cost function that can be obtained using the PUB algorithm, when the speed was ramped from 3 to 70 km/h. The bounds were obtained by using the true weights at the transmitter side, i.e., $w_a(l+1) = v_a(l)$. As can be seen from the figures, the coefficient selection in (6) becomes important at speeds higher or equal than 30 km/h.

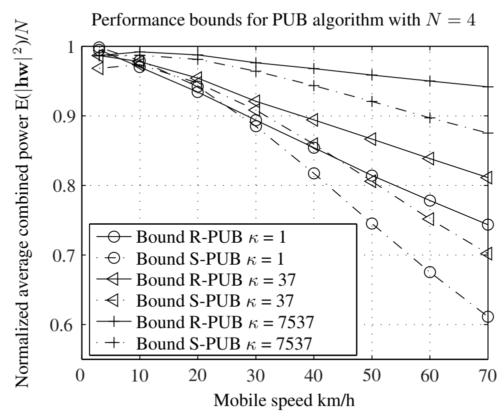

Fig. 1. Performance bound of the PUB algorithm for different correlation scenarios, as function of mobile speed.

The performance of the PUB algorithm improves with the number of available feedback bits, without increasing the computational complexity. Figure 2 shows the performance increase of the PUB algorithm when increasing n_b from 6 to 8 (1 for power and 3 for phase, to 2 for power and 4 for phase). Note that the phase resolution is more important than that of the square magnitude (power). For example, when using R-PUB and $n_b = 6, N = 4$, assigning 1 bit for power and 3 bits for phase quantization outperforms the same scheme with 2 and 2 bits (recall that the signaling overhead associated to a takes 2 bits because $N = 4$).

Figure 2 shows the performance of the R-PUB algorithm compared to that of CBBF, both at the same feedback rate of $n_b = 6$. The PUB algorithm outperforms CBBF in tracking performance, up to speeds of 30 km/h in uncorrelated channels.

It should be noted that at low mobile speeds, the feedback bits reserved to signal the selected coefficient might be better employed increasing the quantization resolution, at least for moderate sizes of

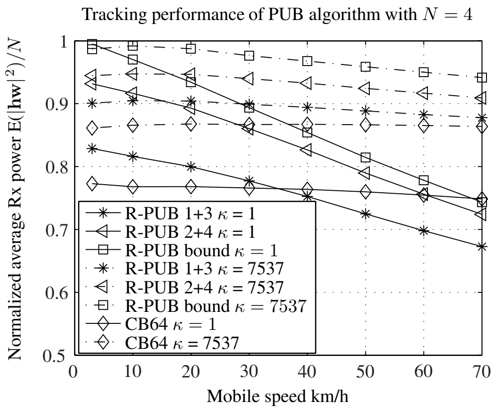

Fig. 2. Tracking performance comparison of R-PUB and CBBF for feedback rate of $n_b = 6$, plus performance of R-PUB with $n_b = 8$ (2+4). "p+q" refers to bits assigned to power and phase, respectively

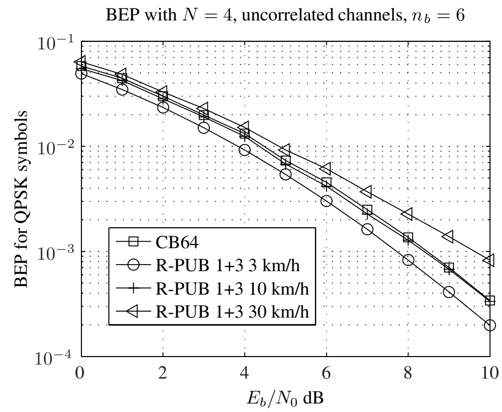

Fig. 4. BEP for PUB algorithm and codebook solution in uncorrelated channels, with $n_b = 6$. "p+q" refers to bits assigned to power and phase, respectively

N such as 4 or 8. Thus, the S-PUB algorithm could outperform the R-PUB algorithm at very low speeds, and a mechanism for switching between them would be required if the speeds are allowed to change.

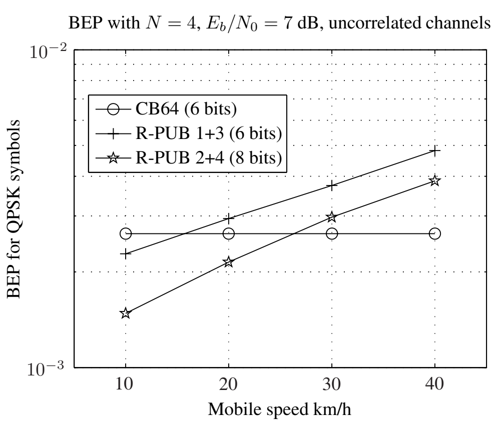

Fig. 3. BEP for PUB algorithm and codebook solution in uncorrelated channels, at E_b/N_0=7 dB. "p+q" refers to bits assigned to power and phase, respectively

The BEP curves for selected configurations are shown in Figure 4. It can be seen that at 10 km/h, both R-PUB and CBBF perform about the same when $n_b = 6$. However, CBBF performs better at 30 km/h and more, but with a much higher computational complexity, as detailed in Section 3.4. Note that CBBF shows almost the same performance for the considered speed range. The performance increase of the R-PUB algorithm due to the addition of 2 bits is shown in Figure 3.

5. CONCLUSIONS

This paper proposed a distributed partial update adaptive filtering scheme suitable for transmit beamforming with low feedback rate. The proposed algorithm features a low complexity update that adapts one coefficient at the time based on a signal dependent selection function. This formulation extends the existing partial update adaptive filtering techniques to a distributed system setup where the optimum filter is known at one end, and must be conveyed to the other end under severe rate restrictions. The complexity is linear with the size of the filter only and does not depend on the feedback rate, which contrasts with existing codebook solutions. Simulations for slowly fading uncorrelated channels show that the PUB has better tracking performance, similar BEP performance, and reduced complexity when comparing to existing codebook techniques [4].

6. REFERENCES

[1] B. C. Banister and J. R. Zeidler, "A simple gradient sign algorithm for transmit antenna weight adaptation with feedback," *IEEE Trans. Signal Processing*, vol. 51, no. 5, pp. 1156–1171, May 2003.

[2] B. Raghothaman, "Deterministic perturbation gradient approximation for transmission subspace tracking in FDD-CDMA," in *Proc. IEEE ICC*, vol. 4, Anchorage, May 2003, pp. 2450–2454.

[3] E. Zacarías B., R. Wichman, and S. Werner, "Filtered gradient algorithm for closed loop MIMO systems," in *Proc. IEEE VTC-Spring*, Stockholm, May 2005.

[4] D. J. Love, R. W. Heath Jr., and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2735–2747, October 2003.

[5] S. C. Douglas, "A family of normalized LMS algorithms," *IEEE Signal Processing Lett.*, vol. 1, no. 3, pp. 49–51, March 1994.

[6] T. Aboulnasr and K. Mayyas, "Selective coefficient update of gradient-based adaptive algorithms," in *Proc. IEEE ICASSP*, vol. 3, Munich, April 1997, pp. 1929 – 1932.

[7] S. Werner, M. de Campos, and P. Diniz, "Partial-update NLMS algorithms with data selective updating," *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 938– 949, April 2004.

[8] 3GPP RAN WG1, "Technical specification TS25.202: Physical layer - general description," 3GPP, available at www.3gpp.org, Tech. Rep., 2002.

[9] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE J. Select. Areas Commun.*, vol. 20, no. 6, pp. 1211–1226, August 2002.