Exploiting Multi-Core SoC Architecture for MU-MIMO Schedulers

Ganesh Venkatraman, Janne Janhunen, Markku Juntti
Centre for Wireless Communications (CWC), Department of Communication Engineering (DCE),
FI-90014, University of Oulu, Finland

***Abstract*—Upcoming standards are moving towards multi-antenna multiple-input multiple-output (MIMO) transmission techniques to harness the benefits of spatial degrees of freedom (DoF) in addition to the conventional time and frequency resources. Even though single user MIMO transmission improves the throughput noticeably, multiplexing different user data streams across the spatial dimension as in multi-user (MU) MIMO enhances the overall cell throughput significantly. However, this improved performance depends on the efficient selection of the users to be multiplexed over the spatial DoF. In this work, we compare the performance of different scheduling schemes in terms of achievable sum throughput and the overall complexity involved in the implementation for the real-time system requirements. The performances of the proposed schemes are evaluated on MATLAB for various MIMO configurations. The complexity analysis is carried out by implementing the scheduler algorithms on TI TCI6636K2H evaluation platform. We evaluate the complexity by sharing the load across eight TMS320C66x DSP core subsystem on the (system-on-chip) SoC.**

I. Introduction

The current wireless standards are moving towards the packet switched networks to improve the system performance and flexibility as compared to the circuit switched networks. The radio access technologies aim at achieving higher throughput and better system performance but still targeting at low energy budget. The use of multi-antenna transmission is *de-facto* in all upcoming wireless access standards, favoring the spatial multiplexing of user data by applying transmit precoders for multi-user multiple-input multiple-output (MIMO) transmission. In order to achieve the best possible benefit of the multi-user (MU) MIMO transmission, the multiplexed users should have channel vectors as linearly independent as possible. By selecting users with the uncorrelated channel vectors, efficient linear precoders can be designed to decouple the user data streams and to avail the benefits of MU-MIMO technique. The selection of users with such a constraint is carried out by the schedulers to utilize the wireless system resources efficiently.

The scheduling algorithms based on the sum rate maximization objective for MU-MIMO were discussed thoroughly in the literature. The search based on successive projections (SP) scheme for single-antenna receiver was presented in [1] and its extension to the multi-antenna receivers was provided in [2]. In [2], the users were selected iteratively by choosing the channel vector with the maximum gain onto the orthogonal subspace. The orthogonal subspace was obtained by evaluating the null space of the stacked channel vectors of the already chosen users from the earlier iterations. Similar algorithms addressing a lower computational complexity were proposed in [3] and [4]. User selection based on the volume maximization metric was discussed in [5]. It is identical to the SP or the block diagonal (BD) scheme, since both are based on Gram-Schmidt (GS) procedure to find the orthogonal vectors.

Even though the scheduling algorithm determines a subset of users from a large set of contending users, precoders are to be designed efficiently to harness wireless resources effectively. The iterative precoder design based on the mean squared error (MSE) reformulation for the weighted sum rate maximization (WSRM) objective provides an efficient transmit precoder design [6]. The precoders can be used to determine the subset of users implicitly by assigning all-zero vector as the precoder for the unscheduled users. However, the complexity involved in the precoder design scales significantly with the number of users for which the precoders are to be designed as compared to the elementary design based on zero-forcing (ZF) criterion. Even though the ZF based precoder design is inferior to the weighted minimum mean squared error (WMMSE) approach, it provides a closed form expression to evaluate the precoder design followed by a bisection search to identify the power for each data stream. The choice of precoder design also plays a major role in the achievable sum rate.

In this paper, we focus on the efficient implementation of various state-of-the-art MU-MIMO scheduling algorithms on a multi-core system-on-chip (SoC) platform and evaluate the system size that can be supported with the current implementation. In order to have a fair comparison, we first analyze the achievable throughput by different algorithms and then we discuss the implementation complexity of the scheduling algorithms on TI TCI6636K2H evaluation platform. This work is a continuation to our previous implementation studies [7]–[9] aiming at finding low-power but high performance solutions for the advanced long term evolution (LTE-A) transceivers. Current work excludes the precoding design, but it is based on the two-step scheduling-precoding approach, where the scheduler finds a subset of users in the first and in the second step, the precoders are designed for the chosen subset.

The paper is organized as follows. A single-cell downlink system model is described in Section II followed by a brief discussion on the scheduler algorithms in Section III. Section IV compares the performance between selected scheduler schemes using link level simulations. In Section V, implementation results of different scheduling algorithms are presented. Section VI summarizes the conclusions.

II. System Model

We consider a single-cell downlink MU-MIMO transmission with N_T transmit antennas and K users with N_R antenna

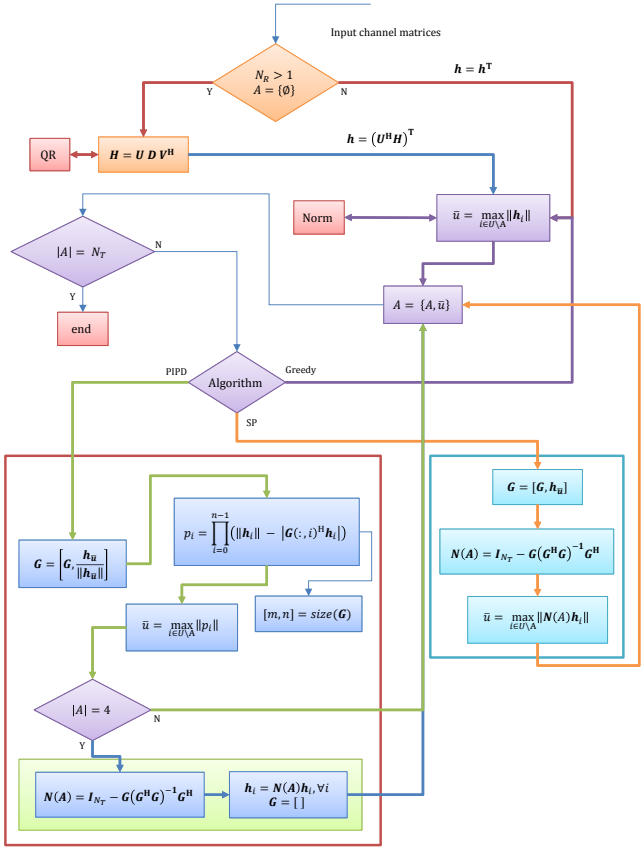

Fig. 1. Scheduler algorithm flow.

each. Let $\mathbf{H}_k \in \mathbb{C}^{N_R \times N_T}$ be the channel matrix seen by user k. Let $\kappa = \min(N_R, N_T)$ be the rank of channel matrix $\mathbf{H}_k$. To utilize the available spatial streams, the channel matrix $\mathbf{H}_k$ is decomposed into N_R virtual single receive antenna channel vectors using singular value decomposition (SVD) as $\mathbf{H}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{V}_k^{\mathrm{H}}$, where $\mathbf{U}_k \mathbf{U}_k^{\mathrm{H}} = \mathbf{I}_{N_R}$ and $\mathbf{V}_k \mathbf{V}_k^{\mathrm{H}} = \mathbf{I}_{N_T}$ are unitary matrices and $\mathbf{D}_k$ is a diagonal matrix. The virtual user or the spatial stream index for the i^{th} column vector of $\mathbf{U}_k \in \mathbb{C}^{N_R \times N_R}$ for the k^{th} user is denoted by $\hat{k} = kN_R + i$ and the equivalent single receive antenna channel is $\mathbf{h}_{\hat{k}} = \mathbf{U}_k(i)^{\mathrm{H}} \mathbf{H}_k$. The total number of virtual single antenna streams available in the system is given by $N_S = N_R K$. Let $\mathcal{U}$ be the set of indices of users present in the system and let $\mathcal{A} \subset \mathcal{U}$ be the set of active users chosen by the scheduling algorithm for a transmission instant. Now, the equalized received symbol $\hat{d}_k$ of user $k \in \mathcal{A}$ for an active spatial stream is given by

$$\hat{d}_k = \mathbf{h}_k \mathbf{m}_k d_k + \sum_{i \in \mathcal{A} \setminus \{k\}} \mathbf{h}_k \mathbf{m}_i d_i + n_k, \tag{1}$$

where $n_k \in \mathbb{C}$ is the equivalent zero mean complex Gaussian noise with variance $\mathcal{N}(0, N_0)$ and $\mathbf{m}_k \in \mathbb{C}^{N_T \times 1}$ is the transmit precoder corresponding to user k. Note that the channel vector $\mathbf{h}_k = \mathbf{U}_{\tilde{k}}^{\mathrm{H}}(i) \mathbf{H}_{\tilde{k}}$, where $\mathbf{U}_{\tilde{k}}(i)$ corresponds to the i^{th} column vector of the left singular matrix $\mathbf{U}_{\tilde{k}}$ and $\tilde{k}$ corresponds to a user and k is the spatial stream index. The data symbol d_k is assumed to be uncorrelated with $\mathrm{E}[|d_k|^2] = 1$.

III. Scheduling Algorithms

The selection of a subset of users $\mathcal{A}$ from the set $\mathcal{U}$ for a sum rate maximization objective requires an exhaustive search, and therefore the complexity is of the order of $O((K \times N_R)^{N_T})$. In order to reduce the complexity involved in the selection procedure, we find a subset of user channel vectors that are as linearly independent as possible to reduce the inter-stream interference caused by the spatial multiplexing. Fig. 1 shows the outline of the scheduling algorithms discussed in this section. Due to the simplicity of the greedy/norm based search, we omit the discussion on the greedy scheduling algorithm. We assume simple linear precoder design based on zero-forcing (ZF) algorithm on the stacked channel vectors of the chosen users by the scheduling algorithm.

A. Successive Projections (SP)

The SP scheduling algorithm selects the users in an iterative manner by finding the user channel vectors that are linearly uncorrelated with the already chosen channel vectors. The compatibility is evaluated by projecting the channel vector onto the null space formed by stacking the existing channel vectors of the chosen users from the earlier iterations [1], [10]. Initially, a user with higher channel gain from the set $\mathcal{U}$ is selected for the transmission set $\mathcal{A}$. Then, the remaining users are selected from $\mathcal{U} \backslash \mathcal{A}$ by projecting the equivalent channel vectors $\mathbf{h}_{\hat{k}} = \mathbf{U}_k(i)^{\mathrm{H}} \mathbf{H}_k$ onto the null space of the user channel vectors in $\mathcal{A}$ as

$$\mathbf{G} = \left[\mathbf{h}_{\mathcal{A}(1)}^{\mathrm{T}}, \ldots, \mathbf{h}_{\mathcal{A}(|\mathcal{A}|)}^{\mathrm{T}}\right] \tag{2a}$$

$$\mathbf{N}(\mathcal{A}) = \mathbf{I}_{N_T} - \mathbf{G}\left(\mathbf{G}^{\mathrm{H}}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathrm{H}} \tag{2b}$$

where $\mathbf{N}$ denotes the null space matrix and $\mathbf{I}_{N_T}$ is the identity matrix of size N_T. The metric for the selection is given by

$$m_i = \|\mathbf{N}(\mathcal{A})\,\mathbf{h}_i^{\mathrm{T}}\|, \ \forall\, i \in \mathcal{U} \backslash \mathcal{A} \tag{3a}$$

$$\bar{u} = \arg\max_i m_i, \quad \mathcal{A} = \mathcal{A} \cup \{\bar{u}\}. \tag{3b}$$

The above metric is evaluated for all users at each iteration to identify a user until the condition $|\mathcal{A}| = N_T$ is satisfied. Even though selecting N_T users is not optimal in the lower signal-to-noise ratio (SNR) regime, the precoders can be recalculated by retaining the channel vectors with non-zero power. To reduce the complexity, we can ignore the square root involved in the channel norm evaluation that is required for the user searching.

B. Product of Independent Projections (PIPD)

The PIPD algorithm requires minimal computational complexity compared to the SP scheduling scheme as discussed in [11]. The complexity involved in selecting the first user remains the same, since the user with the highest channel norm is considered for the set $\mathcal{A}$ in both schemes. The complexity involved in selecting the remaining users for set $\mathcal{A}$ is significantly reduced in the PIPD scheduling scheme by virtue of using the product of independent vector projection displacements metric. A detailed description of the algorithm can be found in [11]. However, for a system configuration of $N_T = 8$ antennas, the performance of the PIPD scheme is significantly inferior to that of the SP scheduling algorithm. To overcome this problem, we modified the PIPD algorithm as highlighted in Fig. 1 by projecting user channel vectors onto the null space of the existing user channels when $|\mathcal{A}| = 4$. Once the channel vectors are projected onto the null space formed by the existing user set $\mathcal{A}$, the algorithm proceeds by finding the remaining users for the active user set $\mathcal{A}$. It

can also be restarted to search the remaining $N_T - 4$ users for the transmission set $\mathcal{A}$, since current $\mathbf{G}$ matrix provides zero valued vector. At this point, the next user selection is simply based on the max norm of the already projected channel vectors and then proceeds as earlier. It reduces the overall complexity involved in the search algorithm by limiting the matrix inversion only to the maximum of 4×4 matrix.

Fig. 1 illustrates the operations involved in the metric calculations for greedy, SP and PIPD scheduling schemes. The blocks mentioned separately for the PIPD and the SP schemes are executed based on the selected algorithm. The greedy algorithm requires only the norm calculation of the virtual channel vectors with which the sorting operation can be used to find the leading N_T users for set $\mathcal{A}$. In case of the SP selection scheme, the users are identified by projecting the corresponding channel vectors onto the null space of the existing users, channel vectors and selecting the one with the maximum projection gain. In order to evaluate the null space, a matrix inversion is needed, which incurs significant amount of processing power in comparison to the vector multiplications involved in the PIPD algorithm in Fig. 1.

IV. Performance of MU-MIMO Scheduling

A. Scheduling Resolution

The user selection is performed over each scheduling block (SB). The number of SBs in the system determines the overall achievable throughput. For instance, if the SB resolution is equal to the physical resource block (PRB), the scheduling complexity increases linearly with the number of SBs and the throughput performance depends on the number of users in the system. Thus, if the number of users is large, the performance will improve by utilizing the multi-user diversity with the help of a proper scheduling scheme. However, if the users are few in number, the performance degradation is significant due to the smaller coding length and lack of channel variations over the smaller SB to extract the channel diversity in the form of coding gain. In case of a larger SB size with grouped PRBs, the scheduling complexity will be reduced significantly, but the network throughput will also reduce noticeably unless the channel fading over frequency domain is flat or nearly flat. In this case, the users with the transmission in such scheduling will attain higher throughput due to the longer coding length which needs to be quantified only by the system level study.

B. Scheduling Performance

Fig. 2 compares the sum rate performance of three different scheduling schemes with $\mathrm{N}_T = 4, 8$ and $K = 100$ users. The sum rate of the PIPD scheme performs significantly close to the SP scheme. Fig. 2 also shows a larger gap (approximately 10 bit/sec/Hz) between Norm/Greedy scheme which selects users based on the channel norm only and PIPD scheduling scheme for a 4×2 configuration. Note that the performance of the scheduling algorithms improves marginally by adding additional receive antennas, due to the channel hardening by the multi-user diversity. Increasing N_T antennas in the system, improves the achievable sum rate of the system by multiplexing more number of users over the spatial dimension, which is evident from Fig. 2.

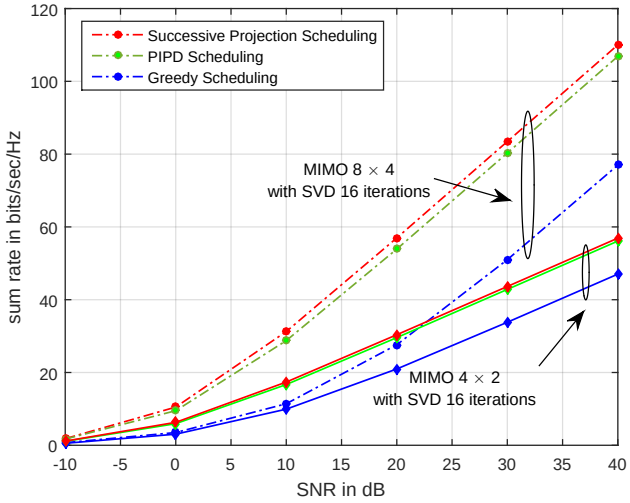

Fig. 2. Comparison of scheduler algorithms for $K = 100$ users.

V. Implementation Results

In this section, we discuss the fixed point implementation of various scheduling algorithms discussed in the Section III on TI's Keystone-II platform. For the current implementation work, we considered only single SB scenario and utilize the available $N_C = 8$ cores in the evaluation platform.

A. SVD Implementation

The SVD is performed by repeated QR factorization to obtain the left and the right singular vectors of the channel matrices. The complex QR decomposition is performed using the Householder transformations [12]. We perform 16 QR factorization for each SVD processing to limit the off diagonal values of the $\mathbf{D}_k$ matrix $\leq 1e^{-4}$. The SVD requires significant computational complexity and time, and therefore, it is ideal to perform in parallel over multiple cores. Since the matrix size is typically limited by the number of transmit and the receive antennas, parallelizing each SVD operation is not an efficient approach due to the overhead involved in setting up the resources for sharing and synchronization. In the current approach, each TI C66x core shares $\lfloor \frac{K}{N_C} \rfloor$ of the available SVD operations to perform in parallel.

The channel matrices are stored in multi-core shared memory (MSMCSRAM), and therefore accessible from all cores in the SoC. In this work, only Core(0) is used to perform scheduling and all other cores are utilized for the SVD processing. Due to the availability of the shared memory MSMCSRAM region, we can synchronize the cores effectively by using shared memory and inter-processor communications (IPC) interrupts available through chip-level interrupt controller (CIC) instead of using OpenMP framework [13]–[15]. Since each core is running separate SYS-BIOS operating system, the shared memory region addressing will be different for each cores. In order to overcome the memory aliasing, we fix the channel buffer to a fixed physical address using the compiler directive *location* so as to be coherent with the channel buffer location. The channels are randomly generated and stored in the shared memory region. An interrupt is given to all the cores to notify the availability of the updated channel state information and to start processing for the SVD operation in parallel.

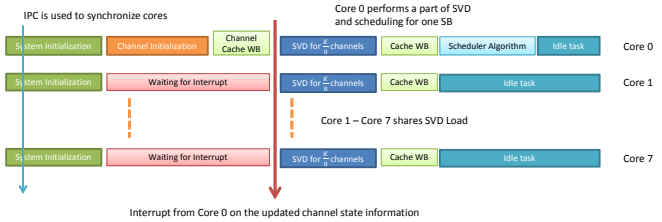

Fig. 3. Task scheduling over $N_C = 8$ cores.

At this point, each core determines the starting offset in the buffer based on the number of cores sharing the computation and the current core number. If the total number of channel matrices is not an exact multiple of the number of cores, then the first $\text{mod}(K, N_C)$ processors perform one additional SVD processing to share the overall load. In the current scenario, Core(0) performs the scheduling of users. Therefore, after completing the SVD of the channel matrices, instead of waiting for other cores to complete the SVD job, Core(0) can proceed with the scheduling algorithm after invalidating the relevant buffers without affecting the overall selection procedure as shown in Fig. 3. Meanwhile, other cores will complete and write back all the virtual channel matrices to the shared MSMCSRAM from the cache before switching to the idle task. We notify the availability of the channels using CIC interrupt to all cores as shown in Fig. 3.

B. Scheduler Implementation on C66x

The scheduling algorithms and the overall control is performed by Core(0). Since we analyze only single SB in the current work, we utilize only C66x Core(0) to perform user scheduling. In order to reduce the computational latency, we implement all the algorithms and the SVD computation in fixed point format. The complex baseband channel is represented in a packed real and imaginary format with 16 bit representation by allocating 15 bits for the fractional part and one bit to represent the sign. Since the output of the SVD is also used in the precoder design, Core(0) writes back the virtual channel buffer before proceeding with the user scheduling. The complex operations involved in the scheduling algorithms are null space evaluation and the matrix-by-vector multiplications. In order to evaluate the actual timing of the scheduling algorithm, we invalidate the cache of Core(0) before the scheduling algorithm. The complexities of various schemes are tabulated in Table I. Note that the overall complexity involved in the scheduler implementation can be reduced by limiting the stream search to the dominant streams only, since the singular values are sorted in the descending order. Due to the available multi-user diversity, limiting the stream search has marginal impact on the sum rate performance.

Table I compares the complexity of various algorithms using the time as the metric for $K = 100$ users. The legend λ in Table I denotes the number of streams included in the scheduler algorithm search. It can take the maximum value of κ, which is the rank of the channel. As seen from Table I, significant amount of time is spent on the SVD computations. From Table I, we can see that the total time taken by performing SVD in parallel is significantly less when compared with the single core performance. Note that the multi-core performance is not equal to the single core timing if it is multiplied by the number cores. It is due to the overhead involved in the cache coherency and also due to the unequal sharing of the load of $K = 100$ channel matrices.

TABLE I. SCHEDULING COMPLEXITY FOR $K = 100$ USERS (msec) WITH C66X OPERATING AT 1.2GHZ

$N_T \times N_R$	λ	SVD (1)	SVD (8)	Greedy	SP	PIPD
8×4	4	22.68	2.90	0.075	0.524	0.469
8×4	2	22.68	2.90	0.064	0.325	0.268
8×2	2	6.055	0.79	0.063	0.325	0.266
8×2	1	6.055	0.79	0.058	0.226	0.166
4×4	4	15.81	2.07	0.045	0.168	0.167
4×4	2	15.81	2.07	0.034	0.102	0.098
4×2	2	4.844	0.64	0.034	0.102	0.097
4×2	1	4.844	0.64	0.029	0.069	0.063

As far as the scheduler complexity is concerned, we can see that all the scheduling algorithms can meet the LTE-A timing requirements for the user scheduling, which happens at the sub-frame interval of 1 msec. The complexity of the SP and the PIPD algorithm is attributed to the null space matrix evaluation. Note that the square root is not performed for the norm calculation in the SP and in the greedy scheduling scheme, which provides additional margin. The complexity of the SP algorithm is due to both null space evaluation and the matrix-by-vector multiplication. The SVD operation can be performed over the radio frame of 10 msec. duration, which can be justified by the nomadic user assumption with the slowly changing channel fading.

The computational figures provided in Table I for the SVD scales linearly with each additional user in the system and it increases with the number of users K with each additional SB. The complexity of the scheduling algorithms scales linearly with the additional number of SBs while with the number of users, it is sublinear. Table I also provides information about the scalability of the system with the number of SBs that can be supported by the SoC implementation without compromising the real-time constraints. For example, assuming the channel is coherent for 10 ms, we can support 8 SBs with 8×2 MIMO configuration by performing all SVDs related to $K = 100$ users in 6.055 msec and any scheduling algorithm at 1 msec interval. In this approach, each SB is served by each core dedicatedly without any real-time performance glitch. Note that if we reduce the number of QR iterations in each SVD, we can support 10 SBs with slight degradation in the performance.

VI. CONCLUSIONS AND FUTURE WORK

We studied the computational complexity of different state-of-the-art MU-MIMO scheduling algorithms. We showed that the complexity is mainly attributed to the SVD decomposition of the channel matrices and the matrix multiplications involved in the metric calculations. We have demonstrated the performance of three different scheduling algorithms on TI TCI6636K2H evaluation platform by sharing the SVD operations over multiple C66x DSP cores. In future, we analyze the performance of the scheduling algorithms on four ARM cores by performing only SVDs on the available C66x DSP cores.

ACKNOWLEDGMENT

This research was done in BaSE (Baseband and System Technologies for Wireless Evolution) project supported by the Finnish Funding Agency for Innovation (Tekes), Nokia, Broadcom, Xilinx and Elektrobit.

REFERENCES

[1] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," in *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3. IEEE, march 2006, pp. 528 – 541.

[2] A. Tolli and M. Juntti, "Scheduling for Multiuser MIMO Downlink with Linear Processing," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 1, Sept 2005, pp. 156–160.

[3] Z. Shen, R. Chen, J. Andrews, R. Heath, and B. Evans, "Low complexity user selection algorithms for multi-user MIMO systems with block diagonalization," in *IEEE Transactions on Signal Processing*, vol. 54, no. 9. IEEE, 2006, pp. 3658–3663.

[4] Z. Youtuan, T. Zhihua, and Z. Jinkang, "An Improved Norm-Based User Selection Algorithm for Multiuser MIMO Systems with Block Diagonalization," in *IEEE 66th Vehicular Technology Conference, VTC-Fall*. IEEE, 2007, pp. 601–605.

[5] L. Jin, X. Gu, and Z. Hu, "A Novel Volume-Based Scheduling Scheme for Multiuser multiple-input multiple-output Downlink System," in *IEEE Conference on Radio and Wireless Symposium (RWS)*. IEEE, 2010, pp. 448–451.

[6] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel," in *IEEE Transactions on Signal Processing*, vol. 59, no. 9. IEEE, 2011, pp. 4331–4340.

[7] J. Janhunen, T. Pitkänen, O. Silvén, and M. Juntti, "Fixed- and Floating-Point Arithmetic Processor Comparison for MIMO-OFDM Detector."

[8] T. Hänninen, J. Janhunen, and M. Juntti, "Novel Detector Implementations for 3G LTE Downlink and Uplink," *Analog Integr. Circuits Signal Process.*, vol. 78, no. 3, pp. 645–655, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10470-013-0128-5

[9] S. Shahabuddin, J. Janhunen, M. Juntti, A. Ghazi, and O. Silvén, "Design of a Transport Triggered Vector Processor for Turbo Decoding," *Analog Integr. Circuits Signal Process.*, vol. 78, no. 3, pp. 611–622, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10470-013-0183-y

[10] A. Tolli and M. Juntti, "Scheduling for Multiuser MIMO Downlink with Linear Processing," in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC)*, vol. 1. IEEE, sept. 2005, pp. 156 –160.

[11] G. Venkatraman, A. Tolli, J. Janhunen, and M. Juntti, "Low Complexity Multi-User MIMO Scheduling for Weighted Sum Rate Maximization," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*. IEEE, 2014, pp. 820–824.

[12] H. Simon, *Adaptive Filter Theory*. Prentice Hall, 2002, vol. 3.

[13] "OpenMP Programming for KeyStone Multicore Processors." [Online]. Available: http://www.ti.com/lit/ml/sprt620a/sprt620a.pdf

[14] "SYS/BIOS Inter-Processor Communication (IPC)." [Online]. Available: http://www.ti.com/lit/ug/sprugo6e/sprugo6e.pdf

[15] "KeyStone Architecture Chip Interrupt Controller (CIC)." [Online]. Available: http://www.ti.com/lit/ug/sprugw4a/sprugw4a.pdf