REAL TIME IMPLEMANTATION OF LMS BEAMFORMER FOR cdma2000 3G SYSTEM USING TI TMS320C6701 DSP

Kerem Küçük[*], Mustafa Karakoç[*], and Adnan Kavak[+]

[*]Kocaeli University
Electronics and Computer Ed. Dept.
İzmit, 41100 Turkey

[+]Kocaeli University
Computer Eng. Dept.
İzmit, 41040 Turkey

E-mails: kkucuk, mkarakoc, akavak@kou.edu.tr

Key words: Beamforming, DSP, LMS, smart antenna, software radio 3G system

ABSTRACT

Digital Signal Processor (DSP) is an emerging technology that offers great opportunities for the development of future wireless communications systems. In this paper, we demonstrate an implementation of real time adaptive beamforming for cdma2000 reverse link using TI C6701 DSP. We consider that the DSP executes least-mean-square (LMS) algorithm for the beamformer weight vector computation. The results show that the system can handle computation of optimum weight vector within reasonable time for 3G applications. The computed weight vector is observed to steer the main beam of the antenna array towards the desired signal direction. Complexity and execution time of the algorithm increase with the number of antenna elements.

I. INTRODUCTION

Smart antennas [1] that compose of an antenna array and advanced signal processing module in the baseband are being regarded by many as the key solution for increasing the spectral efficiency and improving the system performance in mobile communications. In principle, an antenna is smart only when it can recognize and track the signal of a mobile user while suppressing interfering signals. The antenna array employs adaptive beamforming algorithms for recognize, track and suppress the interference. In this study, we investigate the feasibility of implementing an adaptive beamforming algorithm, namely Least Mean Square (LMS) beamforming method, utilizing TI C6701 floating point DSP for real time cdma2000 applications. The simulation system consists of a PC for data generation, Matlab's TI code composer toolbox for interfacing module, and TI TMS320C6701 evaluation module (EVM) for the beamformer weight vector computation.

This work is supported by The Scientific and Technical Research Council of Turkey – TUBITAK under grant EEEAG / 102E015

Simulations are performed as the following steps: First, received data in cdma2000 reverse link format [2] is generated using Matlab and saved in a data file of a PC. In the signal model, wireless propagation channel and interference effects are considered by setting multiple access interference (MAI), additive white Gaussian noise (AWGN), and channel parameters properly. Second, the adaptive algorithm is coded in ASM and compiled finally to be loaded in a memory of DSP board (TMS320C6701). DSP board computed the weight vector that is based on LMS algorithm and beamformer generated the weight vector for multi-users in real time. Third, the weight vector obtained in the beamforming module is feedback to the PC via Matlab's interfacing functions and code composer studio (CCS) to check the performance based on off-line processing.

II. SYSTEM MODELING

Consider a reverse link channel [2], assuming that there are N subscribers in a cell or a sector, the signal received by M element antenna array at the base station can be expressed as,

$$\mathbf{X}(t) = \sum_{\ell=1}^{L} \alpha_{\ell} s(t-\tau_{\ell})\,\mathbf{a}(\theta_{\ell}) + \mathbf{I}(t) + \mathbf{N}(t) \qquad (1)$$

where, s(t) is transmitted signal with complex path attenuation $\alpha_l = \beta_l e^{j\phi_l}$ and time delay τ_{ℓ} for each multipath signal, $\mathbf{I}(t)$ is the multiple access interference (MAI) given by

$$\mathbf{I}(t) = \sum_{q=1}^{N-1} \sum_{\ell=1}^{L_q} \alpha_{q,\ell} s(t-\tau_{q,\ell}).\,\mathbf{a}(\theta_{q,\ell}) \qquad (2)$$

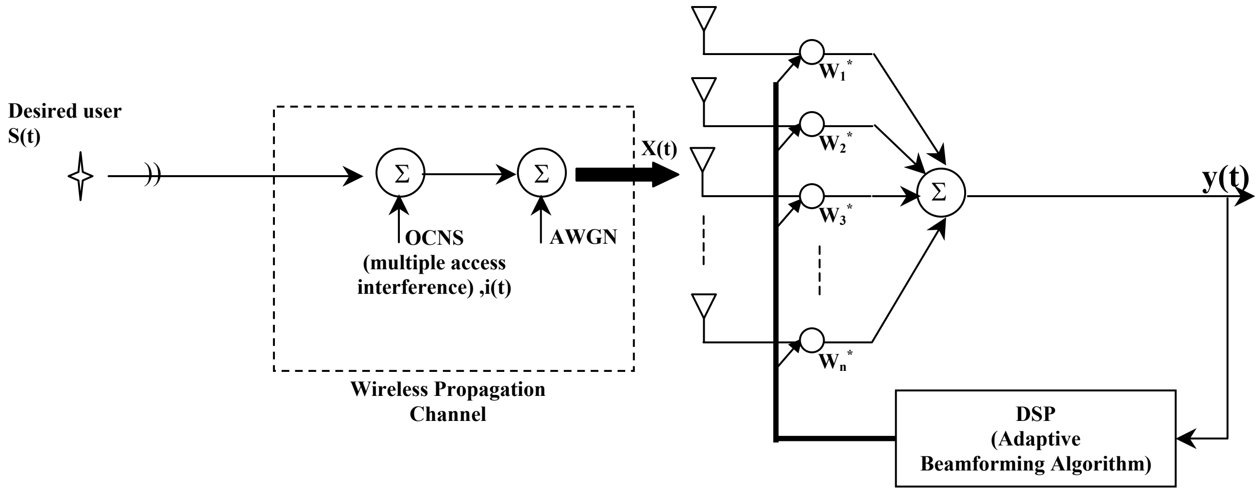

Figure 1 Received signal model at an antenna array.

and $\mathbf{N}(t)$ is the Mx1 complex-valued additive white Gaussian noise vector, $\mathbf{a}(\theta_\ell)$ is the Mx1 array response vector for the multipath arriving at direction of arrival (DOA) θ_ℓ.

At the antenna array output as shown in Figure 1, complex baseband signals are multiplied by complex weight vectors

$$\mathbf{W} = [W_1 \quad W_2 \quad .. \quad .. \quad W_M]^T . \tag{3}$$

the weight vectors W is determined by some optimization (or beamforming) algorithm to yield optimum signal-to-interference (SINR) power ratio or some other objective function. The array output is multiplied by these complex weights to yield

$$y(t) = \mathbf{W}^* \mathbf{X}(t) . \tag{4}$$

The generated signal in (1) is fed to DSP for the computation of the optimum weight vector via LMS algorithm. The DSP processes this signal and calculates a weight vector at each iteration, which converges to optimum weight vector [3].

In the LMS algorithm that is most widely used non-blind type adaptive algorithm introduced by Widrow & Hoff in 1959, antenna weights are recursively obtained to minimize the mean square error using the following equations:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu\, \mathbf{X}(t) e^*(t) \tag{5}$$

$$e(t) = d(t) - y(t) \tag{6}$$

$$y(t) = \mathbf{w}^{\mathbf{H}}(t)\, \mathbf{X}(t) \tag{7}$$

where $\mathbf{w}(t)$ is the antenna weight vector, $\mathbf{X}(t)$ is received signal vector at the antenna output, $e(t)$ is the error signal between the desired response $d(t)$ and weighted antenna output y(t). The superscripts $*$ and H, represent the complex conjugation and Hermitian operation, respectively. μ is the step size parameter chosen as;

$$0 < \mu < \frac{\lambda_{\mathbf{max}}}{10} \tag{8}$$

where the λ is the eigenvalue of pre-correlation matrix. The algorithm does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. The major benefit of the LMS algorithm is its simplicity compared to other adaptive algorithms [4,5].

III. SIMULATION SYSTEM SETUP

The parameters in the signal modeling of eq (1) have been setup as summarized in Table 1.

	LMS Algorithm Conditions		
number of samples	100		
number of elements in the ULA	3	5	7
desired DOA	37		
interference DOAs	45,85,145		
SINR	1,9635	1,9934	2,0021

Table 1 Simulation values

Convergence criteria for the LMS algorithm is determined from the norm of error vector given by

$$\|\Delta \mathbf{w}\| = \|\mathbf{w}(t+1) - \mathbf{w}(t)\| \,. \tag{9}$$

Simulation is performed according to the flow chart outlined in Figure 2. This flow chart is realized in the simulation with Matlab Link for Code Composer Studio. This development tools help use Matlab functions to communicate with Code Composer Studio and with information stored in memory and registers on a target DSP. With these links, it is easy to transfer information to and from the memory and registers in the DSP. Figure 3 illustrates data flow in the simulation system. The signal received by the smart antenna array is modeled and generated in Matlab in a PC. This noisy signal is filtered with the weight vector obtained via the adaptive algorithm executed in C6701 DSP.

.

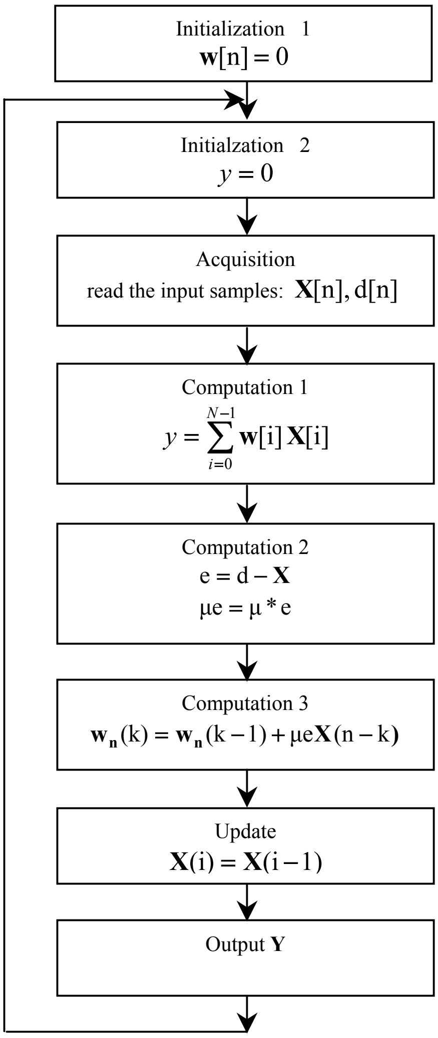

Figure 2 Steps for implementing a LMS adaptive filter

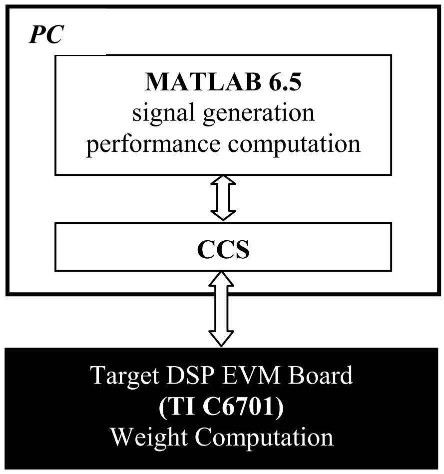

Figure 3 General block structure of simulation system

IV. RESULTS AND DISCUSSION

Tests for 3,5 and 7 antenna elements of uniform linear array (ULA) were performed. We assume that with this number of elements executes beamforming at the system. Table 2 below tabulates the collected performance data. We consider one direct path for the desired user, and three interference signals.

DSP Processing Time			
Antennas	***3***	***5***	***7***
C6701 Clock(Cycle)	2883	5408	7364
C6711 Clock(Cycle)	11737	21311	28583
C6701 Time(msec)	0,01922	0,03605	0,04909
C6711 Time(msec)	0,07824	0,14207	0,19055
Sample	100	100	100

Table 2 DSP beamformer processing times for various antenna numbers with various DSP (C6701 and C6711)

The resulting signal and weight errors obtained with LMS algorithm are shown in Figures 4, 5 and 6 for 3, 5, and 7 ULA, respectively. Spatial spectrums formed with the weight vector resulting from the algorithm outputs are shown in figures 7, 8, and 9. In these figures, we see that antenna array main beam direction is steered towards to DOA of the desired signal. This is because the desired signal was the strongest and the algorithm sense the signal that has the largest power as the desired signal. We note that beamwidth decreases as the number of antenna elements is increased.

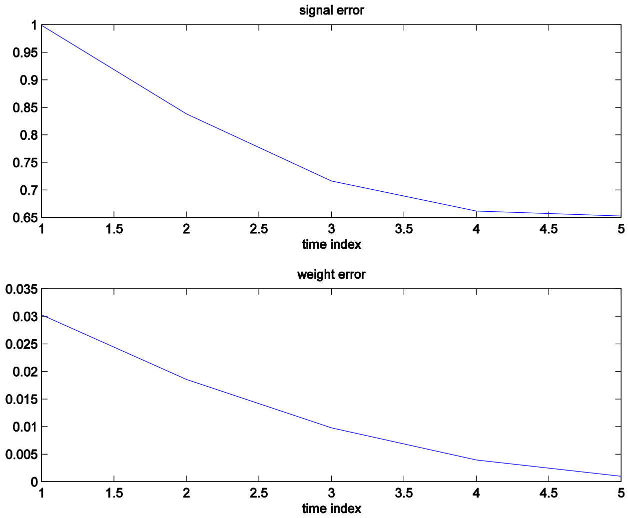

Figure 4 Weight and signal error for m=3 antennas

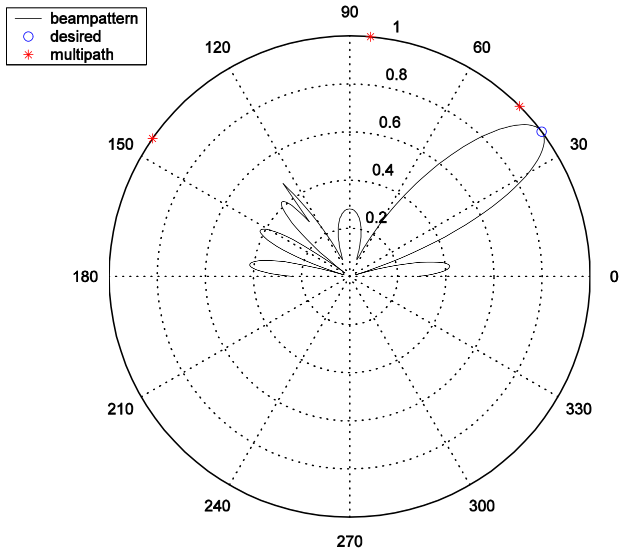

Figure 7 Spatial spectrum with 3 antennas

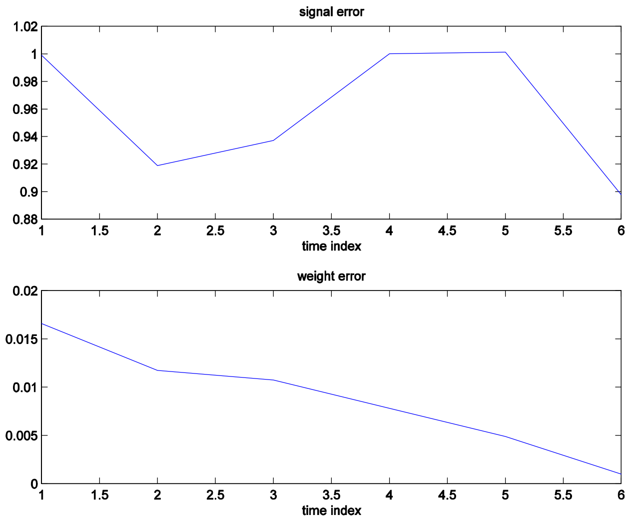

Figure 5 Weight and signal error for m=5 antennas

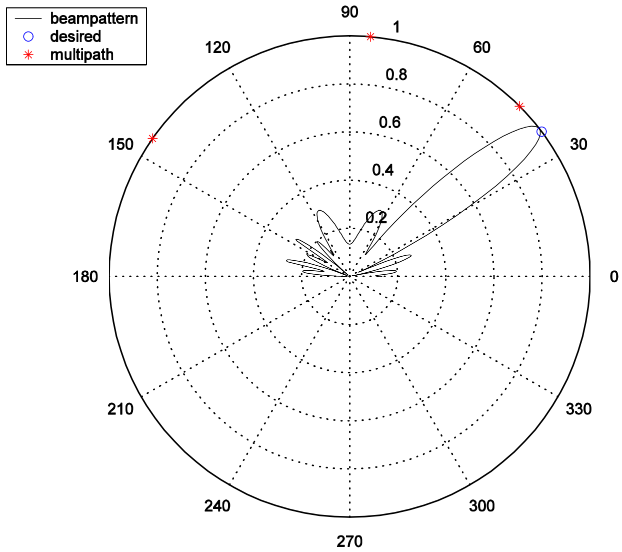

Figure 8 Spatial spectrum with 5 antennas

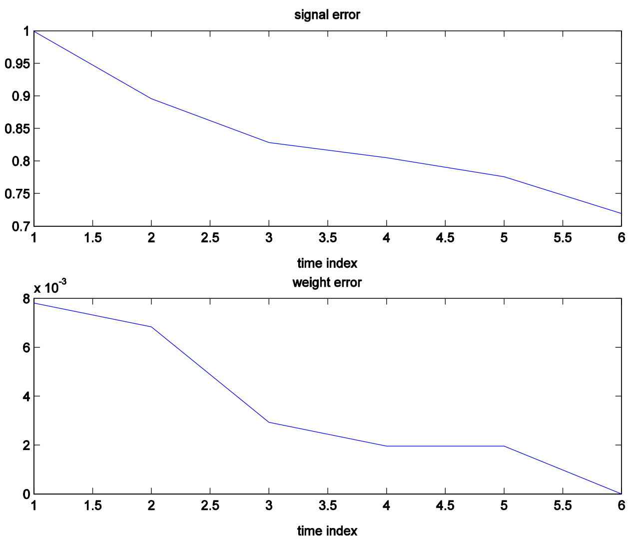

Figure 6 Weight and signal error for m=7 antennas

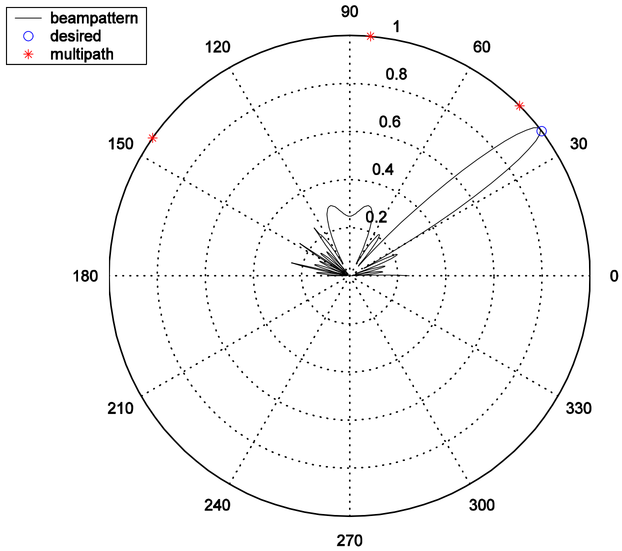

Figure 9 Spatial spectrum with 7 antennas

V. CONCLUSION

In this paper we have demonstrated the use of a digital signal processor in the implementation of real time adaptive beamforming algorithm for cdma2000 3G mobile communication systems. It seems possible from our simulations that the proposed system (TI C6701 DSP and Matlab functions) can handle optimum weight vector computation. We have found that algorithm computation time and accuracy of beam pattern direction are proportional with the number of antenna elements. Although, the results are preliminary they are valuable in terms of finding new directions for optimizing the assembly coding in DSP and thus enabling software radio implementation of various beamformer algorithms.

REFERENCES

[1] T. S. Rappaport, J. C. Liberti, *Smart Antennas for Wireless Communication*, Prentice Hall, NJ, 1999.

[2] TIA/EIA Interim Standard, Physical Layer Standard for cdma2000 spread spectrum systems, TIA/EIA/S-2000-2.

[3] H. Im, S. Choi, and K. Lee, "Implementation of a smart antenna systems test-bed for wide-band CDMA WLL channel," Proc. VTC2000, 0-7803-5718-3/00 2000 IEEE.

[4] Naim Dahnoun, *DSP Implementation using the TMS320C6000 DSP Platform,* Prentice Hall, NJ, 2000

[5] Mohammad Darwish, Cecil Lau, “A Software Radio Architecture for Smart Antennas,” Spectrum Signal Processing, http://www.spectrumsignal.com.