

A Software Radio Architecture for Smart Antennas

Mohammad Darwish and Cecil Lau
Spectrum Signal Processing, Inc.
Vancouver, Canada
Mohammad_Darwish@spectrumsignal.com

Introduction

Smart Antennas are being regarded by many as the key solution to increasing the spectral efficiency and improving the system performance in mobile communication. In principle, an antenna is smart only when it can recognize and track the signal of a particular mobile telephone while suppressing interfering signals. This can be achieved by forming a beam towards the mobile telephone, hence, the term beamforming is widely used in literature.

The evolution of the software radio technology, combined with the recent advances in digital signal processors, has motivated researchers around the globe. Furthermore, the development of VLIW DSP architectures, such as that of the Texas Instruments TMS320C6201 processor, has left researchers curious as to its suitability towards a variety of applications, especially that of smart antennas.

This paper describes the design and implementation of a beamforming system on a Distributed Shared Memory architecture [4]. This architecture relies on a PCI local bus for communication between the TMS320C6201 processors. A specialized DSP-PCI bridge chip provides the interface between the TMS320C6201 local bus and the PCI local bus. The Distributed Shared Memory architecture eliminates the need for global shared memory and, hence, doubles the rate at which data is moved between processors.

The beamforming system is based on software radio where the analog to digital conversion for each antenna element occurs as close as possible to the antenna. Furthermore, the in-phase and quadrature (I and Q) components of each antenna signal are generated in the digital domain using the Harris HSP50214 programmable down-converter.

The beamforming system described in this paper is being used to evaluate the processing requirements for the *recursive least-squares* (RLS) algorithm. We start by giving an overview of beamforming. For illustration purposes, the results of simulating the RLS algorithm in MATLAB are presented. After we give a formal description of the RLS algorithm, we carry on to describe the beamforming system architecture. Finally, we report the results of running the RLS algorithm on the TMS320C6201 processor. Though these results are preliminary, they are valuable in terms of finding new directions for optimizing the overall system performance.

Beamforming Basics

Beamforming is the term used to describe the application of weights to the inputs of an array of antennas to focus the reception of the antenna array in a certain direction, called the look direction or the main lobe [3]. More importantly, other signals of the same carrier frequency from other directions can be rejected. These effects are all achieved electronically and no physical movement of the receiving antennas is necessary. In addition, multiple beamformers focused in different directions can share a single antenna array; one set of antennas can service multiple calls of the same carrier.

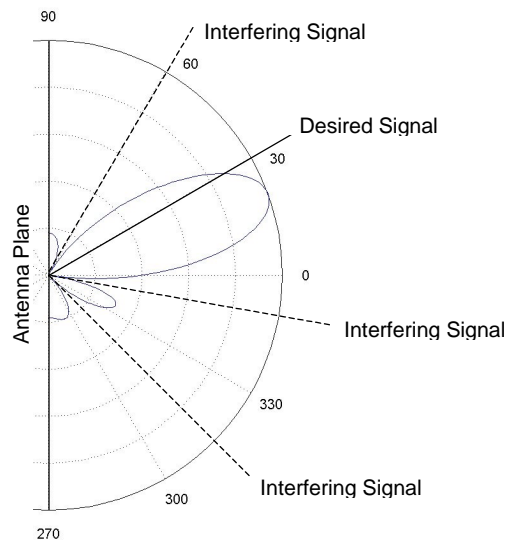


Figure 1: Formed beam of a four element linear antenna array

Figure 1 illustrates the theoretical spatial response of a four element beamformer rejecting signals from -45° , -10° , and $+60^\circ$, and forming a main lobe towards a desired signal at $+30^\circ$. Note that the direction of the main lobe is not exactly at 30° . This phenomena arises from the properties of the beamformer; it is optimized to reject interfering signals rather than to direct the main lobe towards the desired signal.

It is no coincidence that the number of elements in the above diagram equals the number of incoming signals. A beamformer of L antenna elements is capable of accepting one signal and reliably rejecting $L-1$ signals. A greater number of interfering signals will diminish the performance of the beamformer.

Beamforming presents several advantages to antenna design [1]. Firstly, space division multiple access (SDMA) is achieved since a beamformer can steer its look direction towards a certain signal. Other signals from different directions can reuse the same carrier frequency. Secondly, because the beamformer is focused in a particular direction, the antenna sensitivity can be increased for a better signal to noise ratio, especially when receiving weak signals. Thirdly, signal interference is reduced due to the rejection of

undesired signals. For the uplink case of transmitting from the antenna array to a mobile telephone, system interference is reduced since the signal is only transmitted in the look direction.

A digital beamformer is one that operates in the digital domain. Traditionally, beamformers were implemented in analog; the weights were determined and applied to the antenna inputs via analog circuitry. With digital beamforming, the antenna signals are individually translated from Radio Frequencies (RF) to Intermediate Frequencies (IF), digitized and then down-converted to base-band I and Q components. A beamforming algorithm implemented on one or more digital signal processors then processes the I and Q components to determine a set of weights for the input signals. The input signals are then multiplied by the weights and summed to output the signal of interest (SOI). Figure 2 below illustrates the process.

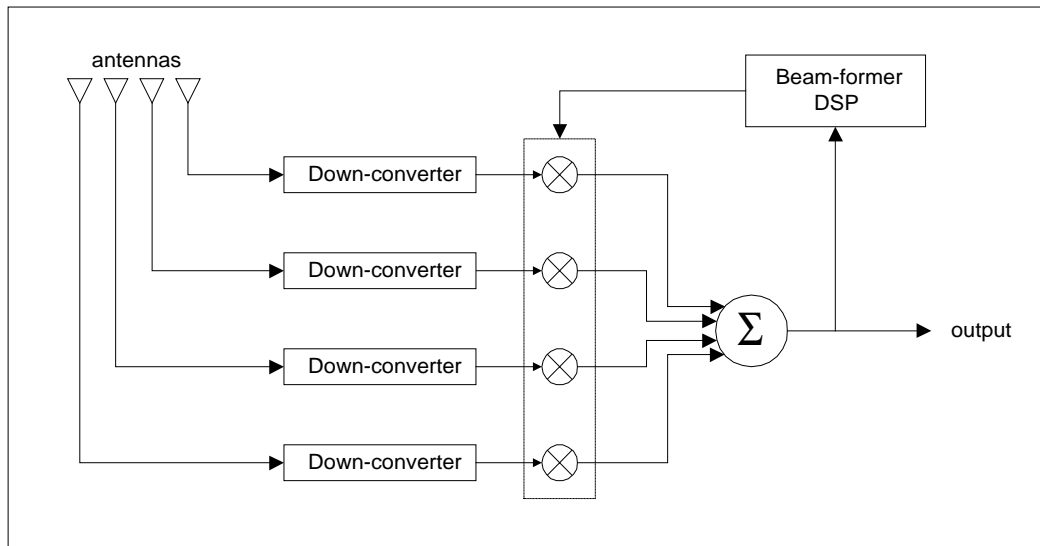


Figure 2: The beamforming process with a 4 element antenna array

An adaptive beamformer reliably updates its set of weights to track the direction of the SOI. Hence, if the SOI is that of a mobile telephone, the beamformer will constantly update its set of weights such that the look direction of the beamformer and more importantly, the directions of signal rejection are steered as the signal sources change in azimuth with respect to the plane of an antenna array.

The RLS Algorithm

One of the foremost advantages offered by the software radio technology is flexibility. Because beamforming is implemented in software, it is possible to investigate a wide range of beamforming algorithms without the need to modify the system hardware for every algorithm. Consequently, researchers can focus their efforts on improving the

performance of the beamforming algorithms rather than on designing new hardware, which can be a very expensive and time consuming process.

A complete description of the RLS algorithm can be found in [2]. This algorithm was chosen for its fast convergence rate and ability to process the input signal before demodulation. While the first reason is important especially when the environment is changing rapidly, the later reason decreases the algorithm dependency on a specific air interface.

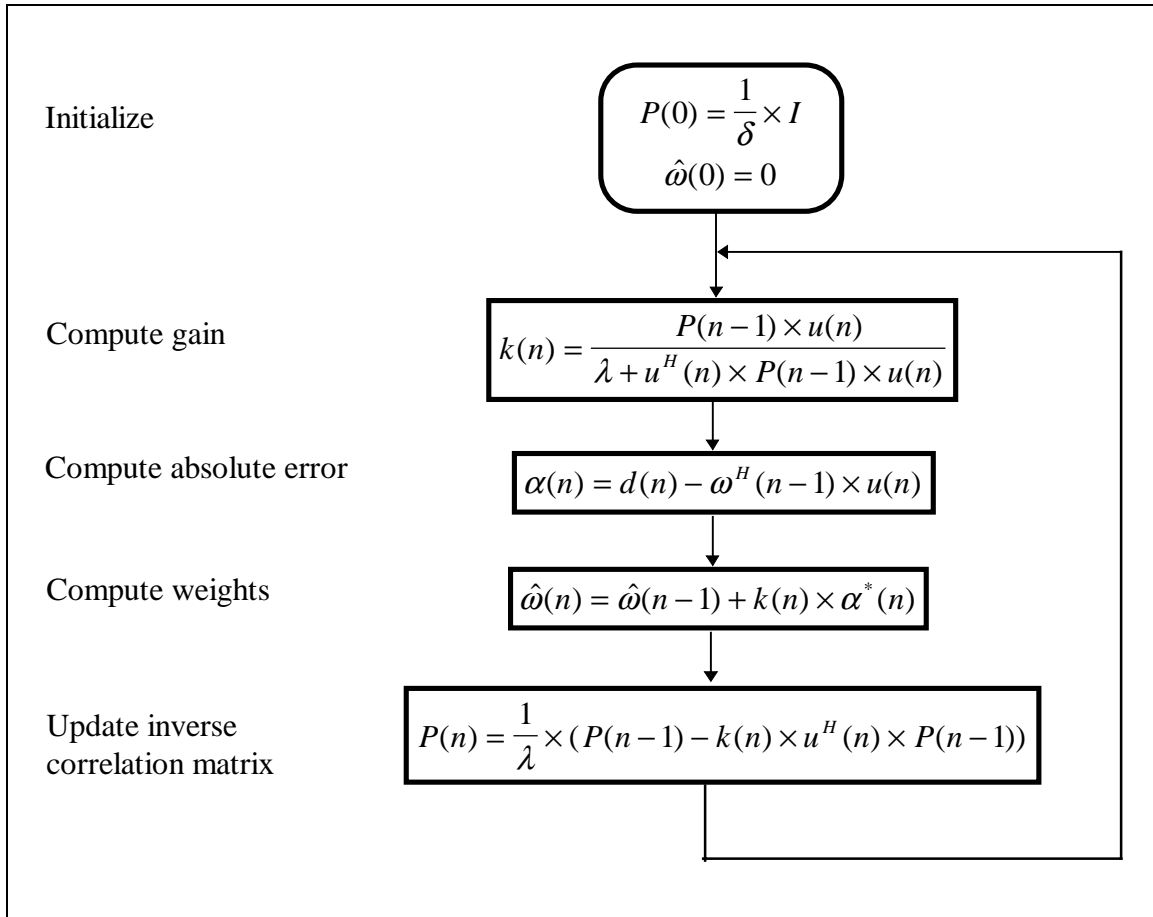


Figure 3: Summary of the RLS algorithm

A summary of the RLS algorithm is shown in Figure 3. Note that applying the RLS algorithm does not require any matrix inversion computations as the inverse correlation matrix is computed directly.

The RLS algorithm recursion is initialized by choosing a starting value for the inverse correlation matrix $P(n)$ that assures the non-singularity of the correlation matrix. The weights vector $\hat{\omega}(0)$ is set to zero. Subsequently, for every input sample n , a gain

factor $k(n)$ and an absolute error $\alpha(n)$ are computed, which in turn are used to compute the weight vector $\hat{w}(n)$.

System Architecture

The beamforming system is illustrated in Figure 4. The system consists of the following:

1. An L -element antenna array;
2. L RF receiver stages;
3. L analog to digital converter stages (PMC-MAI);
4. $L/2$ digital down-converter stages (PEM-2PDC); and
5. $L/2$ Daytona dual TMS320C6201 PCI boards.

Daytona provides two 200 MHz TMS320C6201 DSP processing elements on a single board with industry-standard I/O and a fast PCI interface. Each TMS320C6201 processor provides 400 Mbytes/s access to a local bank of Synchronous Burst SRAM and a local bank of Synchronous DRAM. An 8K bank of 32-bit dual port RAM is shared between the processors for low-latency message passing. Daytona features one Processor Expansion Module (PEM) site supporting two TMS320C6201 processors with a theoretical total transfer rate of 400 Mbytes/s. It also features one PCI Mezzanine Connector (PMC) site.

Antenna signals from the RF band are first down-converted to IF by using conventional analog circuitry, such as local oscillators, band-pass filters and mixers. The IF signal is then passed on to the PMC-MAI module.

The PMC-MAI is a PMC module that samples an IF signal at a rate up to 65 MSPS and converts the 12 bit wide parallel data into a high-speed serial data stream. The PMC-MAI provides high-accuracy, low-noise sampling of signals, different input filtering options and offers six high-speed serial outputs for inter-connection with the PEM-2PDC modules.

The PEM-2PDC is a PEM module that receives high-speed, serial data streams of digitized IF signals and down-converts them to base-band signals. The PEM-2PDC has two Harris HSP50214 programmable down-converter (PDC) chips. Each PDC chip can be tuned to a narrow-band radio signal anywhere within the IF bandwidth. The PDC chip converts the signals to base-band signals centered at 0 Hz and provides complex digital samples at a programmable low rate. The I and Q data is passed from the PDC chips through a set of FIFO chips and is accessible by the DSP through the PEM interface.

In order for any beamforming algorithm to work properly, the PDC chips must be synchronized. This can be achieved by programming one PMC-MAI module to send a reset signal to all of the PEM-2PDC modules. When the reset signal is de-asserted, all of the PDC chips start processing input samples at the same time.

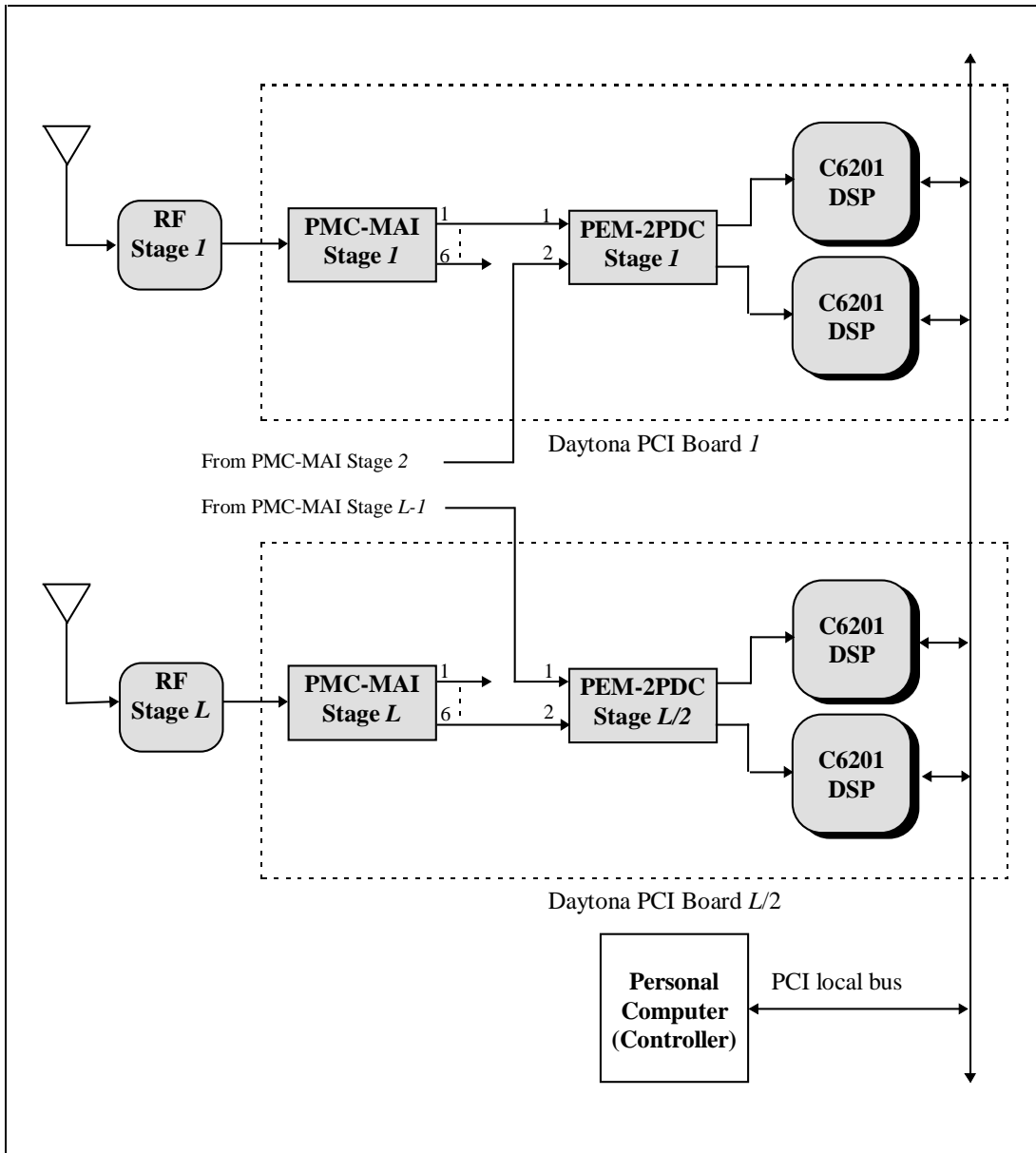


Figure 4: Beamforming system architecture

Experimental Results

Tests for 2, 4, 8, and 16 antenna elements were performed. Table 1 below tabulates the collected performance data. The effective sampling rate of the FIFO's data is 6 kHz. The time it would take for a specified number of samples to arrive from the downconverters to DSP memory is calculated and included in the *Arrival Time* row of Table 1. A plot of the results follows in Figure 5.

Antennas	DSP Processing Time (seconds)					Samples
	64	128	256	512	1024	2048
2	0.062526	0.077718	0.10825	0.16903	0.29113	0.51400
4	0.11310	0.15708	0.24510	0.42133	0.77359	1.47870
8	0.28330	0.41045	0.72775	1.3051	2.5146	4.8854
16	0.76156	1.3646	2.3759	4.6416	8.9706	17.630
Arrival Time	0.010667	0.021333	0.042667	0.085333	0.17067	0.34133

Table 1: DSP Beamformer processing times for various sample sizes

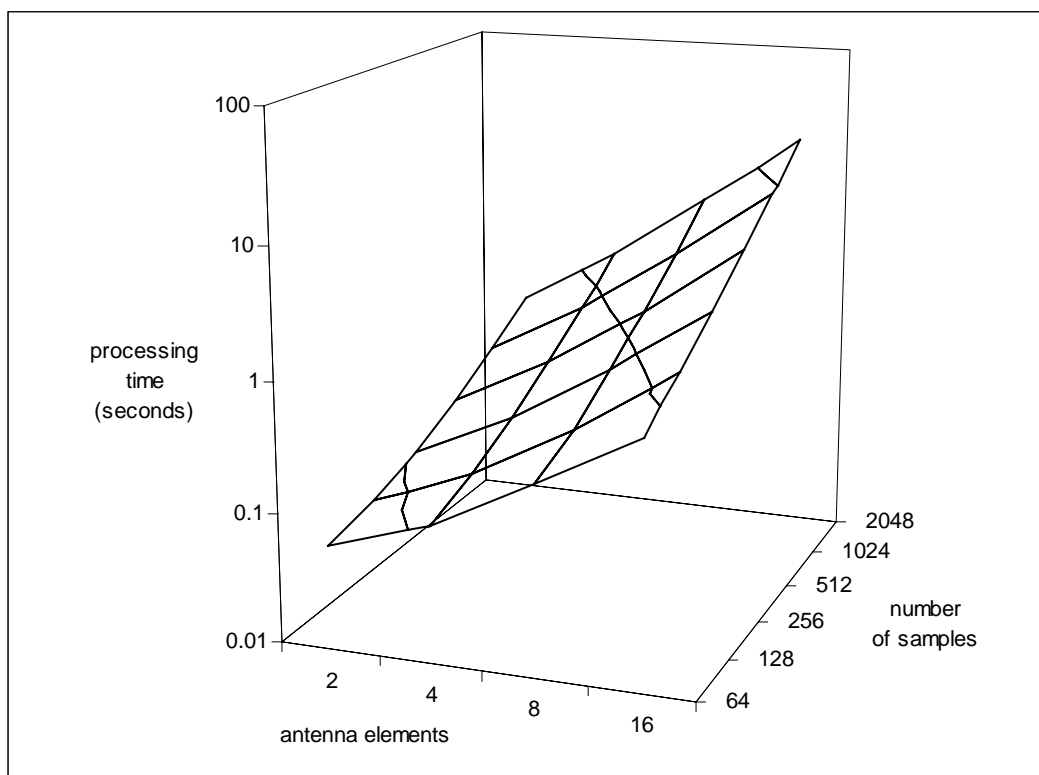


Figure 5: DSP Beamformer processing times for various sample sizes

Theory states that the complexity of the RLS algorithm increases as the square of L , or $O(L^2)$, where L is the number of antenna elements. Though it is difficult to see, Figure 5 reflects this exponential increase. As for the increasing number of samples, the algorithm's complexity should increase linearly. Figure 5 verifies the prediction of an exponential and linear increase in processing time with respect to the number of antenna elements and samples processed, respectively.

Clearly, even the processing time of the two-element scenario is greater than the amount of time it takes for the data to arrive. That is, real time performance cannot be achieved.

The primary explanation is that the floating-point math is running on a TMS320C6201 fixed-point processor. The question at hand is whether we would like to achieve real-time performance for the adaptive beamformer. If not, the advantages of utilizing only one DSP per beamformer are realized. However, if required, then a single DSP running the implemented algorithm may not suffice, even with a floating-point processor. A method of utilizing multiple DSP processors is necessary.

References

- [1] B. Haller, "Algorithms and VLSI Architecture for RLS-Based Time Reference Beamforming in Mobile Communications," Proceedings of the International Zurich Seminar on Broadband Communications IZS'98, Zurich, Switzerland, 1998.
- [2] S. Haykin, Adaptive Filter Theory, Englewood Cliffs, NJ: Prentice-Hall, 2nd Ed. 1991.
- [3] J. Litva and T. Lo. Digital Beamforming in Wireless Communications, Boston, MA: Artech House, 1996.
- [4] M. Darwish, "Distributed Shared Memory - A New High Performance DSP Board Architecture," ICSPAT Proceedings, 1998.