

# Realisation of Radar Signal Processor for the Ground Moving Target Indication (GMTI) mode of SAR on Multiprocessor Hardware

Peter Joseph Basil Morris, Ramakrishnan S, Shayer Dudekula , Ramkumar A, Mahesh Kopp  
Electronics & Radar Development Establishment, Defence Research & Development Organisation,  
Bangalore, India 560093  
[peter.jbm@lrde.drdo.in](mailto:peter.jbm@lrde.drdo.in)

*Abstract: This paper discusses the realization of a radar signal processor for the GMTI mode of operation of SAR on a multiprocessor hardware. The paper discusses at length the algorithms chosen for detection of ground moving targets viz :- Pulse Compression, Doppler processing and CFAR that are employed for the detection of moving targets against the background clutter. The paper also details the design of platform motion compensation algorithm (PMC) along with moving target indication (MTI) used for the mitigation of airborne main-lobe clutter. The paper discusses in depth the implementation of the signal processing chain on a multiprocessor hardware with SIMD architecture, leveraging the vector signal processing capabilities of the hardware and the inherent parallelization of the radar signal processing algorithms. The paper tries to bring out the design methodologies adopted to reduce the execution time, improve the data throughput and to achieve real-time performance. In conclusion, the work brings out the design of the GMTI signal processing solution coupled with the optimized implementation of the same on a multiprocessor hardware.*

**Key Words:** GMTI, SAR, Multiprocessor, SIMD.

## I. INTRODUCTION

Synthetic Aperture Radar (SAR) originally designed as an airborne ground-imaging sensor has long been able to detect, locate and track ground moving objects using the Ground Moving Target Indication(GMTI) mode of operation. GMTI encompass the generic radar modes that allow the detection of ground moving targets against the background of clutter.

GMTI signal processing presents a significant computational challenge due to the real-time processing requirements, compute intensive algorithms and increased computational load due to high data rates. This has led to

traditional high performance computing platforms like clusters of vector or scalar processors, Multicore processors, reconfigurable platforms like FPGA's as suitable solutions to the GMTI signal processing problem. Multicore processors were recently established as the most popular CPU architecture for radar signal processing due to their hardware multithreading capabilities and Single Instruction Multiple Data (SIMD) model.

The paper discusses the realisation of a radar signal processor for the GMTI mode of operation of SAR. It details the algorithm selection, implementation on a multiprocessor hardware with SIMD architecture and the optimisation strategies adopted for the same. Section II describes the GMTI algorithms selected for the implementation. Multiprocessor architecture selected for implementation is brought out in section III. Section IV details the software architecture used for the implementation of the signal processor algorithms. Section V brings out the implementation aspects and the optimization methodologies adopted. The results and conclusion are brought out in Section VI and VII.

## II. GMTI SIGNAL PROCESSOR ALGORITHMS

GMTI forms one of the modes of the SAR, selection of which is based on the radar mode information. GMTI algorithm chain includes :- Pulse compression, Platform Motion Compensation (PMC), n-Pulse MTI (Moving Target Indication), Pulse Doppler processing, Cell averaging CFAR, and Monopulse processing.

The cluster of digitized baseband high data rate samples from a single channel and one pulse are mapped on to the single row and layer of the radar data cube as shown in Figure 1. The cluster of samples from the same channel and the next pulse is stored in the second row at the same layer. The GMTI mode of SAR employs two channel processing thereby degenerating the radar data cube into a

data matrix called the range-Doppler matrix with range-bins (fast-time) and number of pulses (slow-time) as its dimensions.

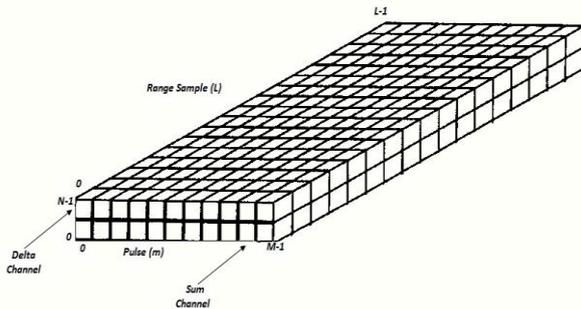


Figure 1 : Radar Data Cube for SAR-GMTI

Range resolution along the fast time dimension is achieved using the traditional pulse-compression techniques. This is followed by the platform motion compensation algorithm which centers the main-lobe clutter Doppler shift induced by the platform motion to the zero Doppler frequency. The MTI processing further applies a linear filter to the slow-time data sequence thereby reducing the main-lobe clutter power which interferes with the target signature. This realisation employs two-pulse and three-pulse MTI cancellers. Doppler processing at the output of MTI performs explicit spectral analysis on the slow-time data for every range bin thereby resolving moving targets and increasing the signal to interference ratio thereby enhancing the probability of detection. The Cell Averaging CFAR (CA-CFAR) detection scheme is employed further for providing predictable detection and false alarm behaviour.

### III. OVERVIEW OF MULTIPROCESSOR HARDWARE ARCHITECTURE

A multiprocessor environment with high number crunching capability and high speed communication fabric has been employed for the compute intensive GMTI processing. Also a high speed low overhead sensor interface has been provided for the transfer of high speed digitized data samples from the analog to digital convertor to the multiprocessor environment. The implementation employs a COTS solution consisting of a pair of multiprocessor boards each with a pair of PowerPC 8640D dual core processors amounting to a total of eight processor cores. A PMC with Serial Rapid I/O capability serves as the ADC sensor interface while an XMC with SFPDP capability acts as the recorder interface for high speed digitized I,Q data recording.

The processor cores are configured under three basic heads. One of the processor core is configured as the Control Node (CN) which receives the digitized down-converted I,Q data samples and distributes the same to the Compute Elements (CE's) or the worker nodes. In addition,

the control node also facilitates the transfer of the high speed I,Q data samples to the high speed data recorder via the Serial-FPDP interface. The Compute Elements (CE's) are the set of processor cores on to which the signal processor chain is fused. Each CE's is fused with the same algorithm chain thereby following a data partitioning approach rather than algorithm parallelization. The number of CE's required for GMTI mode of operation was fixed at six as per the performance requirements. The remaining processor core is configured as the radar controller which performs all the mode specific control operations of the radar and interfaces with the external subsystems of the radar viz:- INS,GPS, Antenna Stabilisation Unit, Transmitter and the Radar Timing I/O module (RTIO).

Each of the six Compute Elements (CE's) are configured with the same GMTI application. Data distribution across the CE's are done in batches from the Control Node (CN) via the Serial Rapid I/O (SRIO) fabric. On the reception of the batch data from the CN the data batches are unpacked and the processing is carried out by the CE's based on the mode information as provided in the mode configuration message. Since every CE is configured with the same application the architecture focuses primarily on data distribution rather than algorithm parallelization to achieve better throughput. The data transfer between the CN and the CE's is accomplished using the DMA transfer facility which places less overhead on the processor.

### IV. SIGNAL PROCESSOR SOFTWARE ARCHITECTURE

The software architecture has been designed to leverage the pipelined nature of the multiprocessor configuration. Identical signal processing chains are ported on to the various CE's. Consecutive batches of data are dispatched to the CE's in a round robin fashion. Hence the total time available for processing is further enhanced by a factor equal to the total number of CE's available for processing. It is also worthwhile to note that an initial latency equal to the processing time of one batch of data is the price paid for this pipelined architecture design. The processed data outputs are dispatched in intervals of the data collection time called the Coherent Processing Interval (CPI). The entire synchronization between the data collection from the sensor interface to the CN, the distribution to the CE's and the dispatching of the processed reports to the radar controller are maintained using semaphores.

### V. IMPLEMENTATION AND OPTIMISATION

#### A. Implementation

The processing dimension of data is of prime importance in obtaining optimal performance for any

processor hardware with cache. Radar data being inherently two dimensional viz. fast-time and slow-time, the processing dimension of data is determined based on the algorithm and data is organised so that consecutive batches of data in the processing dimensions are available in contiguous memory locations, maximizing the locality of reference. The implementation aspects of the various algorithms are discussed below.

- a) **Pulse Compression:** The function performs matched filtering on the digitized down-converted data samples along the fast-time dimension. The algorithm is implemented using the custom designed Fast Fourier Transform (FFT) libraries [1]. A weighted array of twiddle factors are computed prior to performing the FFT operation. Since the number of fast-time samples are known priori the twiddle factor array could be setup before the start of actual processing (outside the hot-loop). This helps in reducing the FFT execution time overheads inside the processing loops. The same twiddle factor array could be used for consecutive FFT operations of sizes smaller than or equal to the weights array size. The algorithm is performed on every pulse data along the fast-time (range) dimension using vector signal processing libraries [1] to leverage the SIMD and strip-mining capabilities of the multiprocessor hardware.
- b) **Platform Motion Compensation (PMC):** The algorithm compensates for the main lobe clutter shift due to the motion of the airborne platform. The algorithm computes the main lobe clutter Doppler frequency [2] and compensates for the same. The operation is performed along the slow-time (pulse) dimension resulting in a corner turn after the pulse compression operation.
- c) **MTI:** MTI processes samples along the slow-time (pulse) dimension. Two and Three pulse MTI are implemented using SAL library [1]. For every fast-time sample the MTI performs subtraction of each complex sample of a pulse with that of the previous pulse. The operation is repeated for every range sample. To leverage the SIMD architecture the entire strip of slow-time data is brought on to the cache memory and is jettisoned only after the completion of the entire algorithm on that data strip, thereby ensuring data locality and faster cache access.
- d) **Pulse Doppler Processing:** Performed along the pulse dimension, the algorithm performs an FFT in an

iterative manner, for all range-bins. A weighted array of twiddle factors are setup prior to performing the FFT operation. Since the number of processing pulses are known priori the twiddle factor array could be setup before the start of actual processing (outside the hot loop). The same twiddle factor array could be used for consecutive FFT operations of size smaller than or equal to the weights array size. The output is passed on to the CFAR block, where the processing dimension is along fast-time. Hence strides of length equal to the pulse length are specified while storing the FFT output thereby arranging the result along the fast-time dimensions further avoiding a corner-turn operation.

e) **Constant False Alarm Rate (CFAR):** Also referred to as adaptive threshold detection, CFAR is used to provide predictable detection and false alarm behavior in realistic interference scenarios. 1-D Cell Averaging CFAR is employed with the processing dimension along the fast-time. A 1-D mask vector with dimensions same as the input data vector is employed for performing a running sum across the input data samples estimating the background information corresponding to every data sample or cell under test (CUT). The running sum is implemented by the convolution of the input data vector with that of the mask vector. The convolution operation is implemented through the FFT libraries [1]. The background information computed is further compared with the input data resulting in a binary data vector of ones and zeros, with ones indicating the declaration of targets.

## B. Optimisation Strategies

Several optimization strategies have been employed to increase the throughput, reduce the execution time and to improve the performance of the GMTI algorithms. The pipelined architecture coupled with the vector signal processing capabilities of the hardware has been leveraged to obtain maximum performance benefits. The following strategies were adopted during the course of the software implementation.

1. **Data Organization:** Organization of data as one dimensional arrays so as to enable faster element access using simple pointer increments
2. **Multiprocessor Hardware:** Leveraging the SIMD capabilities of the multiprocessor hardware through the extensive use of vector signal processing (SAL) [1]
3. **Data Accessing:** Use of unit strides whenever possible, since accessing operands or results with non-unit strides reduces performance of I/O bound SAL functions

4. **Cache Management:** Keeping the data and intermediate results in the L1 and L2 caches as long as possible and use of effective cache management strategies like strip-mining wherein a strip of data is pulled into the super-fast L1 cache, operated on till all the required operations have been completed and then finally loading the next strip of data for processing

## VI. RESULTS

The developed signal processor was evaluated using real-time airborne data for its functionalities, performance and execution timings. Further each intermediate block has been separately timed stamped. The entire chain has been executed over a sizeable number of iterations to check the consistency of the timings. The execution timings for all the processing cores for the GMTI processing chain are tabulated in Table I. The Coherent Processing Interval (CPI) for a batch of data was 70ms. With a total of five processing cores the available time for processing is enhanced to 350ms. As shows in Table I the total processing time on a average is around 215ms which gives a processing margin of around 30 percent. The processing margin enhances the signal processor's capability to further scale up the algorithm chain by the addition of new signal processing algorithms without affecting the execution timings. The algorithm chain was tested on the field with the airborne radar data during field trials Figure 2.

TABLE I. GMTI ALGORITHM EXECUTION TIMINGS

ALGORITHM	CORE 1 (ms)	CORE 2 (ms)	CORE 3 (ms)	CORE 4 (ms)	CORE 5 (ms)	CORE 6 (ms)
Pulse Compression	97	98	97	98	98	99
PMC	21	22	21	22	20	22
MTI	20	22	20	22	21	22
Doppler Processing	15	16	15	16	15	16
CA-CFAR	60	58	60	60	59	58
TOTAL	213	216	213	218	213	217

## CONCLUSION

This paper presented the realisation of a radar signal processor for the GMTI mode of operation of SAR. The paper has primarily focussed on the selection of the signal processing algorithms, the implementation of the same on a multiprocessor hardware and the optimisation strategies adopted to reduce the execution time, improve the data throughput and to achieve real-time performance. A significant improvement in performance was achieved due to the pipelined hardware architecture and the equivalent software organisation. Utilisation of hardware specific optimisation strategies like strip-mining and cache management has further enhanced the throughput of the designed solution. In-short, prudent selection of signal processing algorithms and the multiprocessor hardware coupled with the efficient implementation and optimisation strategies has led to a considerable improvement in the performance figure of the realised signal processor solution.

## REFERENCES

- [1]. *Scientific Algorithm Library (SAL) Reference Manual*, Mercury Computer Systems, 2011.
- [2]. P.Lacomme, JP Hardange, J.C.Marchais, E.Normant, "Air and Spaceborne Radar Systems : An Introduction", NewYork, USA : SciTech,2001.

## BIODATA

Peter Joseph Basil Morris is Scientist 'C' in LRDE, DRDO Bangalore. He has been working in the field of Radar Signal Processing for Airborne radar systems.

Ramakrishnan S is Scientist F in LRDE, DRDO Bangalore. Currently he is Project Director for SAR. His main area of interests includes Active Array radars, Radar target Simulator and SAR.

Shayer Dudekula is Scientist 'C' in LRDE, DRDO Bangalore. He has been working in the field of software development, integration and testing of Radar Signal Processing for Airborne Radar Systems

Ramkumar A is Sc 'C' in LRDE, DRDO, Bangalore. He is working in the areas of embedded systems as well as SAR simulator. He is also involved in the system integration and testing of SAR on FTB.

Mahesk Kopp, is Sc 'E' in LRDE, DRDO, Bangalore. He is System Integration Manager for SAR system. His areas of interests include software development for various radars including SAR.

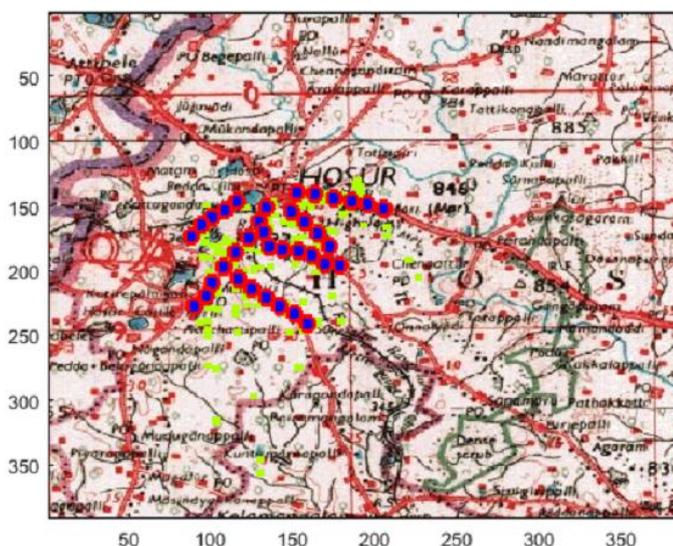


Figure 2 : Ground Moving targets processed from airborne radar data