Design and Implementation of an RFI Direction Finding System for SKA Applications

James Zekkai Middlemost Gowans

A dissertation submitted to the department of Electrical Engineering, University of Cape Town in fulfilment of the requirements for the degree of Masters of Science in Electrical Engineering

Supervisor:
Prof. M. Inggs

June 18, 2015

1 Introduction

1.1 User Requirements

1. A system to perform direction finding of both impulsive and continuous wave (CW) RFI sources is to be designed.
2. They key deliverables of this project are a software package and a thesis report.
3. The software package should have the following functions:
   a) It should take in input from a correlator. This could either be time domain cross correlation for impulsive sources or frequency domain cross correlation for continuous sources.
   b) It should parse a configuration file which contains information about the array configuration and information about the output from the correlator.
   c) The data from the correlator should be used to ascertain the direction of the detected signals.
   d) The software should be designed to fit into a system which has a 100% probability of intercept.
4. The software should have the following user interface features:
   a) The user should connect to it via a web interface.
   b) A streaming waterfall plot of frequency vs amplitude should be displayed to the user to serve as monitoring of the RFI environment.
   c) The user should be able to select a band of interest from the waterfall plot.
   d) The direction finding should then be computed for the signal in that band.
   e) The result of the DF should be presented to the user. An investigation must be done into the best way to present this information to the user.
   f) Where appropriate, additional meta information should be displayed to the user, such as measurement accuracy or signal strength.
5. The system should be designed to find terrestrial RFI sources.
6. The system should be designed to be location independent. It could either be deployed to a fixed location or as a mobile device deployed on a vehicle.
7. This project should be able to interface easily with other systems requiring its data. Specifically, it should be designed to interface with and pass its data on to an allied project which is doing classification of RFI.
8. The system should be real-time, where real-time is defined as having a latency in the order of a few seconds from receiving signals to displaying results to the user.

9. The hardware and software used should be in line with what is used at MeerKAT. This implies the ROACH platform for hardware, Python for back end software and JavaScript for front end software.

10. This system must operate in the context of the MeerKAT site, implying the following:
    a) In general, the RF environment is sparse. While there will be multiple simultaneous transmitters, it can be assumed there will only be one transmitter in a channel and one source of transients at a given time.
    b) The sources of the emissions will be relatively slow moving, up to the maximum speed of a vehicle on a dirt road; 60 km/h.

11. Once the software has been completed, its performance on real life data should be quantified in the following way:
    a) A prototype-stage 4-element antenna array should be connected to a 400 MHz baseband digitiser and correlator.
    b) The correlator need not be real time for the demonstration.
    c) As the goal of this project is not to develop a hardware system, there is no specific requirement on receiver sensitivity or noise figure. Whatever the best available hardware is should be used for the antennas, front end and digitiser.
    d) The performance of the hardware used should be analysed.

12. Mitigation of the effects of performance degradation due to multipath is outside of the scope of this work.

13. The report produced should contain a theoretical analysis of the performance of the system, as well as an analysis of the performance of the prototype on site with real signals.

2 Review of Current Direction Finding Systems

2.1 Introduction

The purpose of this chapter is to provide a discussiong into current strategies and implementations of direction finding systems. An analysis of the advantages and disadvantages of the various systems will take place which will aid in the later descision of which strategy to adopt for this project

2.2 Signals

As discussed by [1]:

We are interested in extracting the parameters of a signal. This is what sensor array signal processing does.

We model the E-field of a narrow-band signal by:

$$E(\vec{r},t) = s(t)\exp\left\{j(\omega t - \vec{r}^{\intercal}\vec{k})\right\} \tag{2.2.1}$$

Where:

- $s(t)$ is the slow (compared to the carrier) modulating signal with bandwidth B
- ω is the carrier frequency
- $\vec{r}$ is the radius vector, of form $\big[x, y, z,\big]$.
- $\vec{k} = \alpha\omega$ which is the wave-vector where $\alpha = \frac{1}{c}$ pointing in the direction of propagation. Note that the magnitude of the wave-vector is known as the wave-number: $|\vec{k}| = k = \frac{\omega}{c} = \frac{2\pi}{\lambda}$. This implies: $\vec{k} = k(\cos\theta \sin\theta)^{\intercal}$ where θ is the angle of the incident wave.

If we have a receiver with a radius vector $\vec{r_r} = \big[x_r, y_r\big]^{\intercal}$

Note that as per the narrowband assumption is is assumed that the array aperture be much less than the inverse relative bandwidth (f/B)

It is shown that the output of an L-element array a L-dimensional vector of the steering vector scalar multiplied by the incident signal, given by

$$\vec{x}(t) = \vec{a}(\theta).s(t) \tag{2.2.2}$$

Here, $\vec{a}(\theta) = \big[a_1(\theta), a_2(\theta), ..., a_L(\theta)\big]^{\intercal}$ which is the steering vector.

Furthermore, it is shown that the principle of superposition applies. If there are M incident signals they are simply summed together:

$$\vec{x}(t) = \sum_{m=1}^{M} \vec{a}(\theta_m)\vec{s_m}(t) \tag{2.2.3}$$

This can be re-written in a more compact form (now adding noise to the model):

$$\vec{x}(t) = \mathbf{A}(\vec{\theta})\vec{s}(t) + \vec{n}(t) \tag{2.2.4}$$

Where:

- we have re-written $\sum_{m=1}^{M} \vec{a}(\theta_m)$ as a matrix of steering vectors

$$\mathbf{A}(\vec{\theta}) = \left[\vec{a}(\theta_1), \vec{a}(\theta_2), ..., \vec{a}(\theta_M)\right] \tag{2.2.5}$$

- we have re-written $\sum_{m=1}^{M} s_m(t)$ as a vector:

$$\vec{s}(t) = \begin{bmatrix} s_t(t) \\ .. \\ s_M(t) \end{bmatrix} \tag{2.2.6}$$

2.3 Overview of Direction Finding

2.3.1 Model

The model which will be discussed here is that presented in [2].

Let there be N individual signal sources, where $\vec{s}(t)$ represents the resultant signal, being

$$\vec{s}(t) = \begin{bmatrix} s_1(t) & s_2(t) & s_3(t) & ... & s_N(t) \end{bmatrix} \tag{2.3.1}$$

Now let there be an array of M antenna elements receiving the signals, where the position of of the ith element is $\vec{x}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^\intercal$. The signal received by this ith element is influenced by the element position. This can be represented as $\vec{s}_i(t, \vec{x}_i)$, showing that the signal at an element is a function of the position of the element. As discussed by [1], as this model contains both spacial and temporal information, it is sufficient to be able to attain spacial information about the signal.

It is shown that the delay time for a signal arriving at the mth element is

$$\tau_m(\vec{\theta}) = \tau_m(\begin{bmatrix} \phi \\ \theta \end{bmatrix}) = \frac{1}{c}[x_m \cos(\phi)\cos(\theta) + y_m \sin(\phi)\cos(\theta) + z_m \sin(\theta)] \tag{2.3.2}$$

Where ϕ is the azimuth angle of the source and θ is the elevation angle. For a 2D space we let $\theta = 0$ and hence simplify to:

$$\tau_m(\phi) = \frac{1}{c}[x_m \cos(\phi) + y_m \sin(\phi)] \tag{2.3.3}$$

The $M \times 1$ steering matrix is

$$\vec{a}_k(\vec{\theta}_k) = \begin{bmatrix} e^{-j\omega_c \tau_1(\phi_k)} \\ e^{-j\omega_c \tau_2(\phi_k)} \\ ... \\ e^{-j\omega_c \tau_M(\phi_k)} \end{bmatrix} \tag{2.3.4}$$

2.4 Antenna Array Fundamentals

Here should be a discussion about how why arrays are necessary for DF. Then a discussion about some of the characteristics of an array.

2.4.1 Array Manifold

As discussed by [3] [4] [5].

The antenna array manifold is said to be useful for direction finding systems, as signal subspace techniques such as MuSIC rely on searching for the best $\vec{a}(p)$ for the detected signal [4].

It is shown by [5] that the output of an array of N sensors receiving M signals in the presence of noise is

$$\vec{x}(t) = \mathbf{A}(\vec{p})\vec{m}(t) + \vec{n}(t) \tag{2.4.1}$$

Where $\vec{x}(t)$ is the N-dimensional output of the array, $\vec{m}(t)$ is the M-dimensional set of signals received by the array, and $\mathbf{A}(\vec{p})$ is a $N \times M$ matrix of source position vectors (SPV). A given SPV, $\vec{a}(p_i)$, shows how the array responds to a source at location p_i, where p_i is often an azimuth and elevation pair: $p_i = (\theta_i, \phi_i)$.

For the case of a terrestrial-only system (which this project will be concerned with), ϕ can be set to 0, meaning that $p_i = \theta_i$, the azimuth angle of source i, typically in the range $[0, 2\pi]$. Here, a SPV can be simplified to $\vec{a}(\theta_i)$.

It is shown that if the N antennas are positioned symmetrically, the antenna array manifold is reduced from complex space $\mathbf{C}^N$ to real space $\mathbf{R}^N$ [5].

The response of the array to a source from a certain location, (θ, ϕ) is:

$$\vec{a}(\theta, \phi) = \vec{g}(\theta, \phi) \odot \exp\left\{-j\mathbf{X}^\intercal \vec{k}(\theta, \phi)\right\} \tag{2.4.2}$$

[5].

Where:

- $\vec{g}(\theta, \phi)$ is a N-dimensional vector of complex number being the gain and phase response of each element in the direction (θ, ϕ).
- $\mathbf{X}^\intercal$ is a $(N \times 3)$ matrix containing the x, y and z coordinates of each of the N elements of the form $\left[\vec{x}, \vec{y}, \vec{z}\right]^\intercal$
- $\vec{k}(\theta, \phi)$ is the wavenumber vector given by $\vec{k}(\theta, \phi) = \pi\left[\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi\right]^\intercal$. Graphically, this equates to the

For the purposes of this research the elements will all be located at the same elevation as we only with to locate terrestrial signals. Hence, this may be simplified to:

$$\vec{a}(\theta) = \vec{g}(\theta) \odot \exp\left\{-j\mathbf{X}^\intercal \vec{k}(\theta)\right\} \tag{2.4.3}$$

Here, $\mathbf{X}^\intercal$ is now a $(N \times 2)$ matrix of the form $\left[\vec{x}, \vec{y}\right]$ and $\vec{k}(\theta) = \pi[\cos(\theta), \sin(\theta)]^\intercal$.

Furthermore, the $\vec{g}$ term may be excluded if we assume omnidirectional elements. Although it is rare to deal with true omnidirectional antennas, for an antenna which is required to receive signals only in the azimuth plane, omnidirectional antennas such as dipoles might very well be used in practice. This hence simplifies to:

$$\vec{a}(\theta) = \exp\left\{-j\mathbf{X}^\intercal \vec{k}(\theta)\right\} \tag{2.4.4}$$

Or, more expressively:

$$\vec{a}(\theta) = \exp\left\{-j\begin{bmatrix} x_1, y_1 \\ x_2, y_2 \\ .., .. \\ x_N, y_N \end{bmatrix}\begin{bmatrix}\cos(\theta) \\ \sin(\theta)\end{bmatrix}\right\} = \exp\left\{-j\begin{bmatrix} x_1\cos(\theta) + y_1\sin(\theta) \\ x_2\cos(\theta) + y_2\sin(\theta)) \\ .., .. \\ x_N\cos(\theta) + y_N\sin(\theta) \end{bmatrix}\right\} \tag{2.4.5}$$

Clearly, this is simply a vector of phase shifts introduced by each element in the array as a function of both the location of the element and the angle of the incident wave relative to some defined zero location and zero direction. It is said by [5] that this array manifold completely characterises the array. That paper goes into additional details on how the manifold may simplified for linear arrays, as well as the special properties which a manifold of a linear array possesses. This will not be discussed here as the array used for this DF system is not likely to be linear.

2.5 Geolocation

Geolocation refers to the process of finding the absolute position of a target, often in terms of a coordinate system like latitude/longitude/elevation. This information is often more useful than only knowing the direction which an emitter lies in. However, it is shown that by having multiple DF stations, the process of triangulation may be used to geolocate a device from direction bearings [2].

This process is shown graphically in Figure 2.1, where multiple DF stations (S_1 through to S_M) are used to locate the x,y,z coordinates of the target x_T. Note that this geolocation process in the figure is for airborne DF systems searching for a ground based target. However, the system could easily be simplified to a purely terrestrial process.

The relevance of this note about geolocation to this work is that it is not necessary to attempt to design a system which can do geolocation natively. Rather, a DF system can be design which can later be duplicated in order to provide geolocation capabilities.

2.6 Overview of Direction Finding

In this section, an analysis is made of different the direction finding techniques which exist. The most applicable of these will be examined in more details following. Classical methods of direction finding algorithms [6].

- Beamforming: by introducing the correct phase delay to each channel of the array, the array factor can be made to be such that the signal in quesetion is added coherently by each element of the array. This phase delay indicates the direction of arrival of the signal. The coherent addition of the signals allows for much higher SNR.

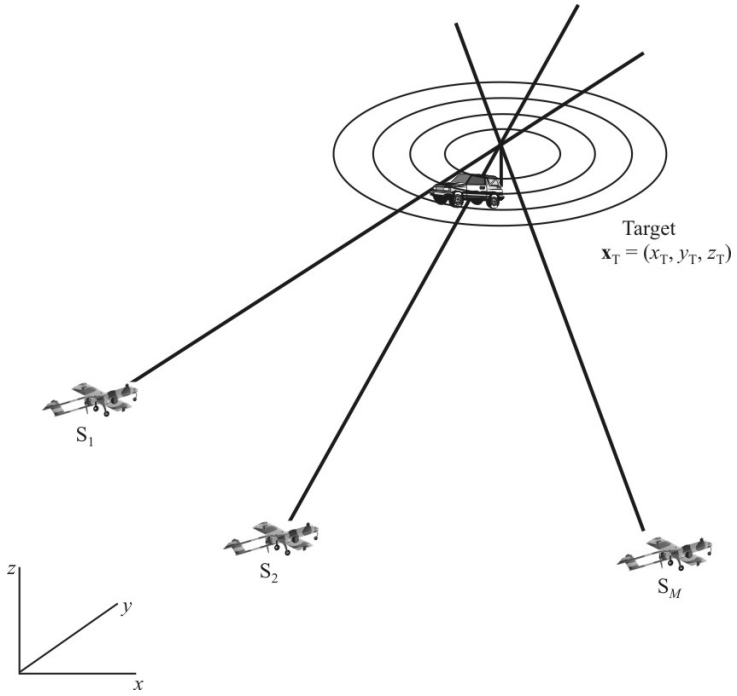

Figure 2.1: Using triangulation from multiple DF stations to ascertain the geographic location of a target. Source: [2]

3 RF Chain

This chapter details the design and characterisation of the chain of antenna, low noise amplifier (LNA), lower pass filters and cabling. The cabling is important as each RF chain needs to have exactly[1] the same phase delay.

The LNAs which are used are the ZLF-500HLN from MiniCircuits. This part operates from 10 MHz to 500 MHz which is ideal for the application. The gain is approximately 21 dB across this band acording to the datasheet. The input current is a maximum of 110 mA.

[1] Some phase difference between the RF chains can and will be calibrated out, but the calibration routine is most robust when the RF chain phases are already closely matches

4 Calibration

There are a few areas where calibration needs to take place in order to ensure that the system performs as expected. What will be discussed here is work on calibrating the ADCs and calibrating the phase of the RF chain from the antennas to the output of the correlator. This RF chain calibration will be done in two stages:

- firstly, the path starting at the input of the LNAs will be calibrated as this can be easily done in the lab by injecting a phase coherent tone into all LNAs simultaneously.
- secondly, the entire RF beginning at the antennas will be calibrated by taking the system out into the field and transmitting tones from known positions

4.1 iADC

The iADCs suffer from a few mismatches between the cores which should be calibrated out for optimal performance. These include:

1. The offsets of each core should be set to 0. Each core can have its offset adjusted from -31.75 LSB to 31.75 LSB in steps of 0.25 LSB.
2. The gains of the cores should be made equal. The gain of each core can be adjusted from -1.5 dB to 1.5 dB in steps of 0.011 dB.
3. The sampling delay between the cores should be adjusted such that the sampling time between each core is as intended. This could be a phase difference of 0 for sampling independent signals in phase, or pi/2 for I/Q sampling or pi for interleaved sampling of a single stream.
4. There are other parameters which can also be calibrated such as the data ready delay adjustment or the gain compensation, but these are not very relevant for this work so will be ignored. Only the analogue gain, offset and sample delay will be calibrated for this work.

5 ROACH Firmware Design

Here are the steps which were done to produce the hardware for testing purposes. Discussion emphasis is on what I changed or modified. Blocks which were already in existance are discussed in less detail.

5.1 Software Setup

CentOS which is the open source version of RHEL. Computer with at least 8 GB of memory as the compile process is memory intensive. Number of CPU cores is not important as the Xilinx compile process only runs single threaded. Xilinx SysGen 14.7. MATLAB R2012B. Casp

5.2 ADCs

The ADCs which were available for use were the CASPER iADCs. These are 8-bit, dual core ADCs, where each core runs at 800 MHz. The cores can either be interleaved to sample a single antenna at 1600 MHz or 2 antennas at 800 MHz each.

5.3 Polyphase Filter Bank

Consists of a polyphase FIR filter which applies a window to the input signal in order to prevent spectral leakage followed by a FFT block. FFT consumes most resources and thus some optimisations had to be done to it. 4K PFB. FFT was a real FFT block meaning it only outputs the upper half spectrum as the lower half is the same due to input signals being real.

Shifting schedule set by software. Bit growth occurs at each stage. If the output of a stage is not shifted down by 1, it risks overflowing. However, if shifting is done unnecessarily, dynamic range is reduced as lower bits are thrown away. Algorithm coded to find optimal shifting. Discuss algorithm here.

5.4 Cross Multiplier

After the FFT, each antenna combination is multiplied together, one being the original signal and one being the complex conjugate. This is somewhat equivalent to dividing the complex numbers, where the key output is that the phase difference between the two antennas is produces. Some maths here to show that this is true.

Optimisations done here: these are fairly large multiplier. Each pair of antennas requires an 18 bit multiplier for the real and imag components, for both simultanious channels. This means 4 18-bit multipliers for 10 combinations. 40 x 18-bit multipliers is a lot of hardware! To mitigate this, I made a change to the complex multiplier block to allow selecting of DSP48E for multipliers. This change was committed back into the centeral code repo for all to use.

Output of a 18_17 x 18_17 is a 37_34.

5.5 Vector Accumulator

The output vector (2K complex elements) is accumulated by summing each element. This is accumulated to a 48 bit number, hence allowing for substantial growth. This is key to getting a very good phase difference approximation as uncorrelated noise is integrated out. The vector accumulator is implemented by two DSP48E blocks, one for the real and one for the imag components. This is followed by a bram which stores and feeds back the vector to the DSP48E adder.

The design is such that 48 bits are continuously accumulated. After the accumulation has run for a configurable number of iterations, the most significant 32 bits are sliced off and snapped. By accumulating 48 bits, no data is thrown away until the snap. Commit XXXXX makes this change to the dsp48_bram_vacc block in the casper library.

6 Software Interface

6.1 Code Structure

It was necessary to write code to allow the computer to interface with the correlator running on the ROACH. The code had to be carefully designed in accordance with good object orientated design methodologies in order to provide a useful, well defined and easily extendible interface to other code which needs to interface with the correlator. As such, there was significant emphasis encapsulating logic into classes which mirrored the physical structure of the correlator in the sense of modularising the key components and writing reusable code. Following is a discussion of the software interface to the correlator with specific focus on the various classes and interfaces.

6.1.1 Control Register

The control register naturally lends itself to a class with interfaces to modify the different bit groups of the register in logical ways. A python module called software_register was written with a single class: SoftwareRegister. This class contains a value parameter which mirrors the value programmed into the FPGA's control register. It then has the following interface methods:

pulse_sync Toggles the syncronisation reset bit low to high and then high to low.

block_trigger ; allow_trigger These methods either set of clear bit the bit that controls the gate which blocks or allows the snapshot blocks being triggered by a complete accumulation. The idea is that this will be used to allow all of the snapshot blocks to be armed sequentially before they can be triggered all at once.

reset_accumulation_counter Pulses the bit which resets the counter which increments every time an accumulation completes and triggers the snap blocks. This counter lets us keep track of how many accumulations have been performed by the system.

pulse_overflow_rst Similar to the above, except this clears the latch which gets latched high whenever the FFT block overflows. This essentially allows us to 'acknowledge' that we have seen the occurrence of the overflow flag.

select_adc Controls which of the four RF inputs is connected to the time domain snapshot block via a multiplexer. By multiplexing the steams, it saves the logic and BRAM of having to have four seperate snap blocks, at the expense of one multiplexer and not being able to synchronously sample the ADC streams in the time domain. This is considered an acceptable tradeoff due to the fact that the time domain snap is only designed to be used for getting an approximation of the signal strength arriving at an antenna and this does not need to be done in parallel for each antenna.

set_shift_schedule This methods takes a 12 bit number and sets the FFT shift schedule to that number. While this design only has a 10 stage FFT, 12 bits is kept in the register for future expansions.

All of the methods in this class log their actions as well as the new value of the control register when it is modified at the debug log level.

7 Field Trials

7.1 Power supply

It was necessary to power the ROACH from a battery in order to allow it to be portable and taken out into the open field. Initially the plan was to power it from an inverter running off of a battery. Here, discuss why the inverter may be too noisy. Can this be shown from reverb chamber measurements?

Instead, an ATX power supply which runs directly from a 12 V battery was made available from the SKA equipment. The power supply is made by Mini-Box and its part number is PicoPSU-80-WI-32V. This can output 80 W which is enough to run the ROACH. To connect it, the traditional mains-powered ATX powersupply is disconnected from the motherboard and this module is plugged into the motherboard. This is shown in (insert figure here!).

Furthermore, a ROYAL 1150K battery was supplied. This is a 105 A h deep cycle calcium battery. As the ROACH draws X amps at 12 V, this implies that we are discharging the battery at $\frac{105}{X} = \frac{C}{Y}$. It is advised to not run down below 70 % to maintain the battery lifespan. As shown by (cite battery charging), this implies that the voltage should not drop to below 12.1 V when discharging at $\frac{C}{Y}$

Testing in the lab showed that the ROACH pulled 3.1 A at 12 V which is 37 W. This can easily be handled by the 80 W ATX power supply. Testing by running the ROACH from the battery overnight. The battery started at 12.6 V and had dropped to 11.9 V 15 h later. The purpose of this test was not to provide a comprehensive report of the capacity of the battery or the requirements of the ROACH, but simply to show that the system will easily be able to run for a few hours in the field during field trials.

Bibliography

[1] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, 1996.

[2] R. Poisel, *Electronic warfare target location methods.* Artech House, 2012.

[3] A. Sleiman and A. Manikas, "Antenna array manifold: A simplified representation," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 5. IEEE, 2000, pp. 3164–3167.

[4] H. Karimi and A. Manikas, "Manifold of a planar array and its effects on the accuracy of direction-finding systems," in *Radar, Sonar and Navigation, IEE Proceedings-*, vol. 143, no. 6. IET, 1996, pp. 349–357.

[5] I. Dacos and A. Manikas, "Estimating the manifold parameters of one-dimensional arrays of sensors," *Journal of the Franklin Institute*, vol. 332, no. 3, pp. 307–332, 1995.

[6] T. E. Tuncer and B. Friedlander, *Classical and modern direction-of-arrival estimation.* Academic Press, 2009.