Self-Stabilizing Switched Beam Offset Reflector (3SBOR) Antenna:

10 GHz Prototype Antenna Solution for mobile radar and communication applications

By

Guillaume Dufour

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the Requirements for the degree of

Master of Applied Science

In

Electrical and Computer Engineering
Department of Electronics
Carleton University
Ottawa, Ontario

ABSTRACT

The purpose of this thesis is to design a solution to the problem of random and fast movement of a ground platform which normally prevents the use of a high gain, pencil beam antenna. Such a problem has prevented the development of a truly mobile ground radar (i.e useable as a vehicle is moving, in contrast to the current concept of mobile ground radar, which translates into a ground radar that can be easily moved between static operation sites.)

In response to this requirement, a self-stabilizing switched offset reflector antenna was designed. Through the use of accelerometers and position filtering, the antenna constantly switches its beam in order to maximise directivity in the intended direction. As a proof of concept, a 10 GHz, single dimension dish was built and tested in laboratory. The results confirmed that the concept is viable and would open enable not only truly mobile radar, but could potentially greatly improve mobile satellite communications by allowing never seen before antenna gain on ground vehicles.

ACKNOWLEDGMENTS

My most sincere thanks go to my wife Chinyere, who gave me the time I needed to complete this research; my infant son Sébastien, who kept me on the schedule, since his "nap time" was the only true "thesis time" for me.

Thanks to my thesis supervisor, Jim Wight, who not only provided me with invaluable advice and great direction at every occasion but was kind and patient to let me bring my little boy to every progress meeting Ihad.

Special thanks to the team at the Quality Engineering Test Establishment (QETE), especially Michael Keller and Jean-Marc Robert whom without their financial support and technical expertise, this thesis would have been just simulation work in Matlab. Thanks for helping me bring this to life!

TABLE OF CONTENTS

ACRONYMS

3SBOR	-Self-Stabilized Switched Beam Offset Reflector
BPM	- Beats Per Minute
HPBW	- Half Power Beam Width
PRI	- Pulse Repetition Interval
LNB	- Low Noise Block
RPM	- Rotation Per Minutes
SMA	- Sub Miniature version A
MTI	- Moving Target Indicator
RF	- Radio Frequency
WR-90	- Waveguide, Rectangular, 900 mils
HFSS	- High Frequency Simulation System
QETE	- Quality Engineering Test Establishment
SLL	- Side Lobe Level
BDF	- Beam Deviation Factor
FEBI	- Finite Element Boundary Integration
HFSS – IE	- High Frequency Simulation System – Integral Equations
MEMS	- Micro Electro-Mechanical Switches
SP3T	- Single Pole Three Throw
SP8T	- Single Pole Eight Throw
Optimetric	- Optimization metrics

LIST OF VARIABLES (excludes sub-section 4.1)

ha	Height of the antenna in meters
φ	Angle between the roll and Z axes
rz	Intersection of the roll and Z axes in meters
ry	Intersection of the roll and Y axes in meters
pz	Z coordinates of the pitch axis in the YZ plane
py	Y coordinates of the pitch axis in the YZ plane
θ	Angle of roll, measured off the Y axis
Ω	Angle of pitch, measured off the Y axis
ρ	absolute angle deviation from the vertical.
Θ	orientation of the absolute angle deviation from the vertical
α	Attributes a weight to the average angular speed used to calculate the position
Δt	Sampling interval
p	Variable number of samples used to calculate the average angular position
s	Variable number of samples used to calculate the average angular speed
z	sensor error function sample size
r	rejection ratio
ri	rejection ratio increment
$L°$	Rejection angle
ρ_T & Θ_T	Calculated using θ_T & Ω_T. (treated signal)
ρ_M & Θ_M	Calculated using θ_M & Ω_M. (untreated, noisy signal)
ρ_R & Θ_R	Calculated using θ_R & Ω_R (reference position)

LIST OF FIGURES

1 INTRODUCTION

1.1 Description of the problem

In accordance with [1], vibration in a comfortable automobile ride is considered to have a natural frequency between 1 and 1.5 Hertz (or 60 to 90 BPM) with an amplitude of less than 2 inches, which is comparable to the step beat of a natural, relaxed walk for an individual.

Two of the many movement mechanisms of a moving vehicle, respectively the body roll and pitch, are determined by the suspension geometry of the vehicle. These translate into roll and pitch axis as shown in the Figure 1 below:

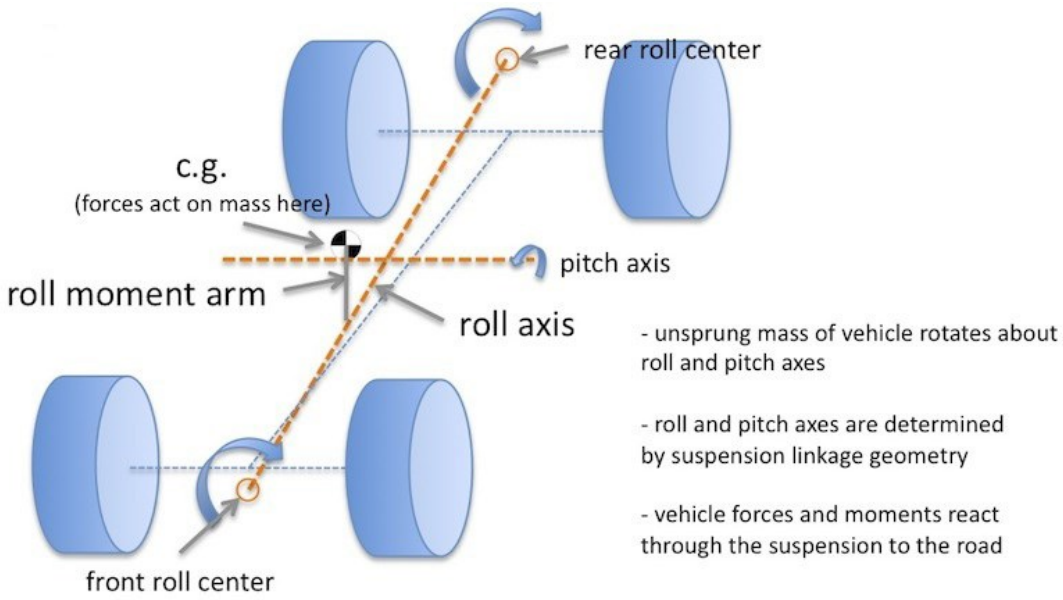


Figure 1: Roll and Pitch Axes (reference [18])

Note that these axes are determined by the suspension geometry. These axes are constant for a given vehicle.

The problem in this chapter will be to define the motion of an antenna located on the roof of a vehicle at a given height of the ground around the roll and pitch axis.

Even a smooth and comfortable ride can easily result in angular variations of about 1º in magnitude. A rougher ride can easily result in angular variations of well beyond 3º. A parabolic antenna can easily have a half power beam width (HPBW) of 2º, such angular variations incapacitate the use of parabolic antennas in mobile applications.

1.2 Target applications

A Self-Stabilizing Switched Beam Offset Reflector (3SBOR) antenna would have applications such as mobile radar systems to satellite communications.

The application for radar systems is the advent of a truly mobile radar. Current mobile radar systems are dubbed “mobile” because they can “easily” relocate between static operation sites. In other words, present ground forces have been blind while moving, relying on outside organizations/forces for radar coverage.

As well, it would be possible to considerably increase a satellite data transfer rate of a moving vehicles or make a stationary satellite communication reflector antenna resistant to jitter caused by crosswind.

1.3 Thesis contributions

The state of the art solution for radar system is the agile beam radar (i.e. phased array). However, this solution only works on mobile platforms that can accommodate the size, computational load and power requirements of a phased array. It works on a plane and on a boat but a ground vehicle is just too small and limited. This is why earth-bound phased-array radar are towed or fixed.

This thesis aims to pave the way for a novel concept for a low cost smart high gain antenna. This antenna will be able to achieve a high gain, comparable to that of a phased array. Just like a phased array can steer its beam to maintain lock on a target, this antenna will be able to "lock" in on a reference point, allowing it to maintain a high gain where it's needed.

This novel antenna concept is a repurposed multi-beam reflector dish.

As you may know, multi-beam reflector antennas are definitively not a novelty. They are commonly used in satellite communication, allowing for coverage of a large area using pencil beams on a single reflector antenna. As for an example, $Bell\ inc$ has multi-beam antenna on every home with satellite TV.

However, my antenna concept differs from the current existing multi-beam reflector design (hence the novelty). It doesn't use all the beams simultaneously; it uses only one beam at any given time. This multi-beam reflector monitors its own position and then selects the beam that provides it with the optimal gain, therefore mimicking the beam steering ability of a phased array, without its disadvantages.

Such antenna concept will have key advantage on ground vehicles. First of all, it will be small enough to me mounted on the vehicle. Second, its cost will be a fraction of that of a phased array. It could very well enable a truly mobile ground radar to be designed or allow a passenger in an RV to watch satellite TV while cruising down the highway.

1.4 Thesis outline

Section 2 presents the model parameters that will dictate the design of the Self-Stabilizing Switched Beam Offset Reflector. A movement model is developed throughout that section to tackle the problem of the vibration of an antenna on a moving ground vehicle.

Section 3 focuses on identifying the technology elements that will guarantee reliable simulation work and a successful prototype.

Section 4 describes the offset reflector design process from the engineering calculations to HFSS simulations. The control algorithm is also covered in this section.

Section 5 provides the results of the antenna system directivity when subject to pseudo-random vibrations and variable noise in Matlab simulations.

Section 6 provides the results of a prototype dish directivity in a live test when subject to oscillations caused by a turntable.

2 REQUIREMENTS

The initial question is quite simple rest in how an antenna moves around while positioned on the roof of a moving vehicle. Many avenues are available, including doing a lengthy movement study of a given vehicle using accelerometer readings. It is necessary to keep in mind that this design must be cost effective, meaning that the movement model must be as simple as possible to reduce calculations requirements and ensure swift operation of the system.

Furthermore, this scope of this thesis does not include the compensation of slow movements of large amplitude resulting from steering, accelerating, braking, going downhill, etc. Compensation mechanisms for these movements (which are currently employed in airborne and shipborne radar system) already exist and will not be considered in this thesis.

Figure 1 shows how ground vehicles jitter and move around the pitch and roll axes. I must define parameters that will let me build a movement model around which I'll establish the antenna design.

2.1 Parameters of the model

Based on the fact that the roll & pitch axes are defined by suspension geometry, it is possible to go forth with the assumption that these constants are known in this study as shown in Figure 2:

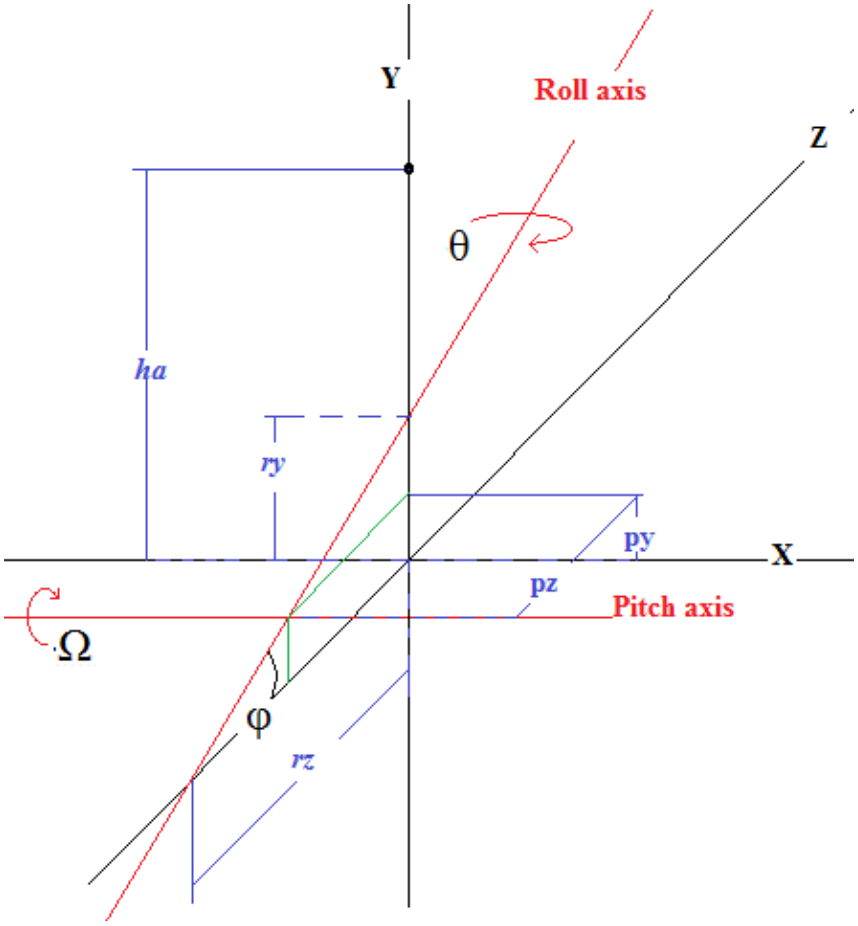


Figure 2: Definition of the roll vector variables

The constants are assumed to be available to define the movement model of the vehicle. In addition, the pitch axis intersects with the roll axis, regardless of the angle of the roll axis and is perpendicular to the YZ plane. Assume the antenna is located at the center of the roof. It is to be noted that the placement in the XZ plane of the antenna isn't relevant as only the height of placement of the antenna will affect its range of angular variations.

Considering a Cartesian coordinates system [X,Y,Z], the following constant will be defined, based on Figure 2:

- ha: height of the antenna in meters. In Figure 2, the antenna is located at [0,ha,0]

-φ: angle between the roll and Z axes;

-rz: intersection of the roll and Z axes in meters. This point is located at [0,0,rz];

-ry: intersection of the roll and Y axes in meters. This point is located at [0,ry,0];

-pz: Z coordinates of the pitch axis in the YZ plane.

-py: Y coordinates of the pitch axis in the YZ plane.

Finally two variables will be defined to establish the movement model:

-θ: angle of roll, measured off the Y axis. Positive θ means clockwise rotation or rotation in the direction of driver. Negative θ means counter clockwise rotation or rotation in the direction of the passenger.

-Ω: angle of pitch, measured off the Y axis. Positive Ω means clockwise or rotation towards the front of the vehicle. Negative Ω means counter clockwise or rotation toward the rear to the vehicle.

Using the above-mentioned constants and axis definitions, it can be established a movement model through vector mathematics for the antenna located at [0,ha,0].

2.2 Roll axis analysis

The methodology employed below to obtain is approximate in nature. As for an example, it was possible to evaluate the minimum and maximum values of the Y component. The Y component would have a maximum value at θ=0 (i.e Y=*ha*) and its values would be the minimum for θ=Pi (i.e. Y= *ry-(ha-ry)·cos(2 φ)*). Using this knowledge, a periodic function was created to match the evaluated minimum and maximum of the Y component.

Accounting for the angle φ, a rotation around the roll axis will introduce movement in three dimensions. Instead of obtaining the exact angle at each point to calculate the exact coordinates in space of the antenna, an approximate function was used.

First the antenna rotates about an arm lever which is anchored to the roll axis at [0,ry,0] and is (ha-ry) meters in length. At θ=0, the angle between the rotation lever and the XY plane is null. It can be determined that at θ=Pi, the angle between the rotation lever and the XY plane is at its maximum value of 2 φ. The angle between the rotation lever and the XY plane can then be approximated as a function of the roll angle θ by the following periodic equation:

Arm lever angle ≈

$$\varphi \cdot (1 - \cos(\theta)) \qquad \textbf{(1)}$$

As one may notice, the value of **(1)** oscillates between 0 and 2·φ. Following this approximate methodology, it's possible to establish the subsequent vector component equations.

2.2.1 Z component

The amplitude of the Z component can be calculated by first understanding that the minimum value of the component is given by taking **(1)** and a roll angle of θ=Pi:

Z component minimum =

$$\frac{-(ha - ry)}{2} \cdot sin(2\varphi) \quad \textbf{(2)}$$

Taking in account this minimum value, the periodic nature of the Z component (it actually oscillates between 0 and **(2)**) I can now substitute the 2*φ term in **(2)** by the angle function **(1).** I can now approximate the Z component with the following equation:

Z component ≈

$$\frac{ha - ry}{2} \cdot sin(\varphi \left(1 - cos(\theta)\right) \cdot (cos(\theta) - 1) \quad \textbf{(3)}$$

2.2.2 Y component approximation

The amplitude of the Y component can be calculated by first understanding that the minimum value is given by:

Y component minimum =

$$ry - (ha - ry) \cdot \cos(2\ \varphi) \quad \textbf{(4)}$$

And of course its maximum value will be *ha*. Taking in account the periodic nature of the Y component (it oscillates between *ha* and **(4)**), replacing the 2*φ term by **(1)** and by adding the appropriate offset (respectively **(4)** plus half the amplitude of the equation) I can approximate the Y component with the following equation (after simplification):

Y component ≈

$$\frac{ha - ry}{2} \cdot \Big((\cos(\theta) - 1) \cdot \big(\cos\big(\varphi \cdot (1 - \cos(\theta))\big)\big) + \cos(\theta)\Big) + \frac{ha + ry}{2} \quad \textbf{(5)}$$

2.2.3 X component approximation

The amplitude of X can be estimated by projecting the length of the arm lever and knowing that the X component will peak at 90º and will be null at 0º and 180º (i.e. periodic in nature), the maximum value of X can be estimated at:

X component maximum ≈

$$(ha - ry) \cdot \sin(\theta) * \cos(\varphi) \quad \textbf{(6)}$$

Unlike other components, the value of φ isn't constant throughout the rotation around the roll axis. By replacing φ with **(1)** I obtain the following component equation:

X component ≈

$$(ha - ry) \cdot \sin(\theta) \cdot \cos(\varphi \cdot (1 - \cos(\theta))) \quad \textbf{(7)}$$

2.2.4 Definition of roll vector.

Now that X, Y, Z components have been defined, it is possible to state the Cartesian [X,Y,Z] position vector that will provide the position in space of the antenna for a given roll angle.

Roll position vector $[X\ Y\ Z]^T$ =

$$\begin{bmatrix} (ha - ry) \cdot \sin(\theta) \cdot \cos\big(\varphi \cdot (1 - \cos(\theta))\big) \\ \frac{ha - ry}{2} \cdot \Big((\cos(\theta) - 1) \cdot \big(\cos\big(\varphi \cdot (1 - \cos(\theta))\big)\big) + \cos(\theta)\Big) + \frac{ha + ry}{2} \\ \frac{ha - ry}{2} \cdot \sin(\varphi\,(1 - cos(\theta)) \cdot (cos(\theta) - 1) \end{bmatrix}$$

(8)

2.2.5 Validation of roll vector model.

In order to confirm the level of validity of the model and that the approximations made aren't detrimental to the precision of the system, system constants were assigned the following values:

Assigned variables:

$$ry = rz \cdot \tan(\varphi)\,, ha = 4\,, rz = 2.7, phi = 0.4187$$

(9)

Given these variables, the following plot is obtained for the roll vector **(8)**:

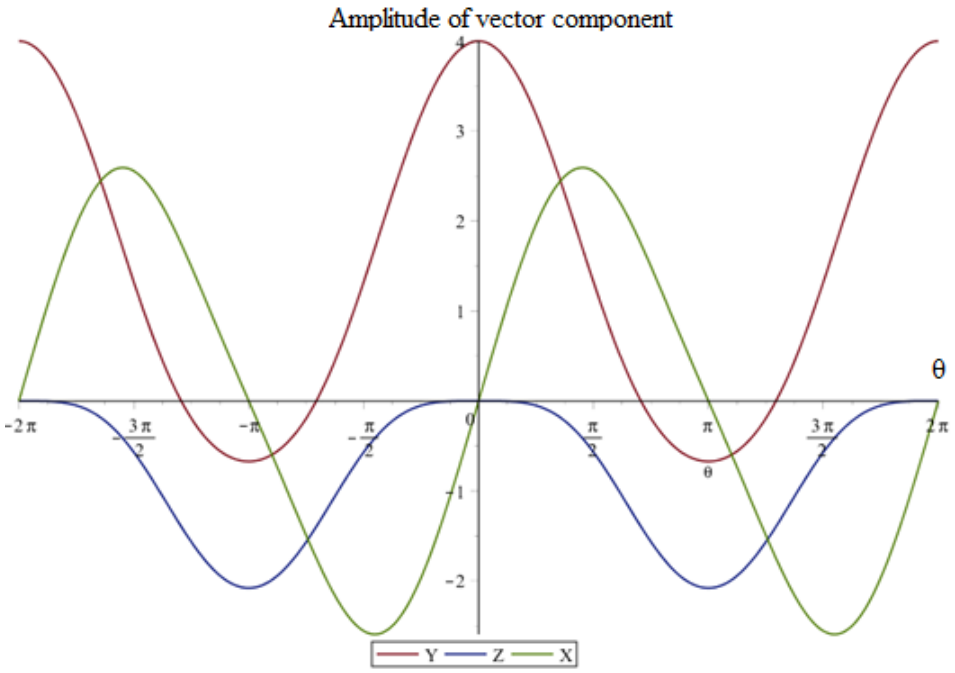


Figure 3 Verification of the roll vector using arbitrary variables.

This graph confirms the physical sense of the component vector. As an additional sanity check, a drawing board was used to estimate the value of Y and Z at a rotation of 180 degree given the variables assigned in ***(9)*** *and the values found corresponded to the roll vector values.*

Considering that a rotation of 90º translates into a vehicle on its sides, the accuracy of the model isn't needed for large value of roll angle θ for the current application or real life. The approximate vector [X,Y,Z] is only required to be accurate for a small range of roll angle.

Knowing that the antenna located at [0,ha,0] is rotating on an arm lever of (ha-ry) meters in length attached at [0,ry,0], it is possible to check the physical correlation of the proposed movement model by calculating the difference between the magnitude of the

roll vector **(8)** for a given range of roll angle and the physical length of the lever. Note that the model is only required to be accurate for a small roll angle (i.e no more than 15º) for the purpose of this thesis.

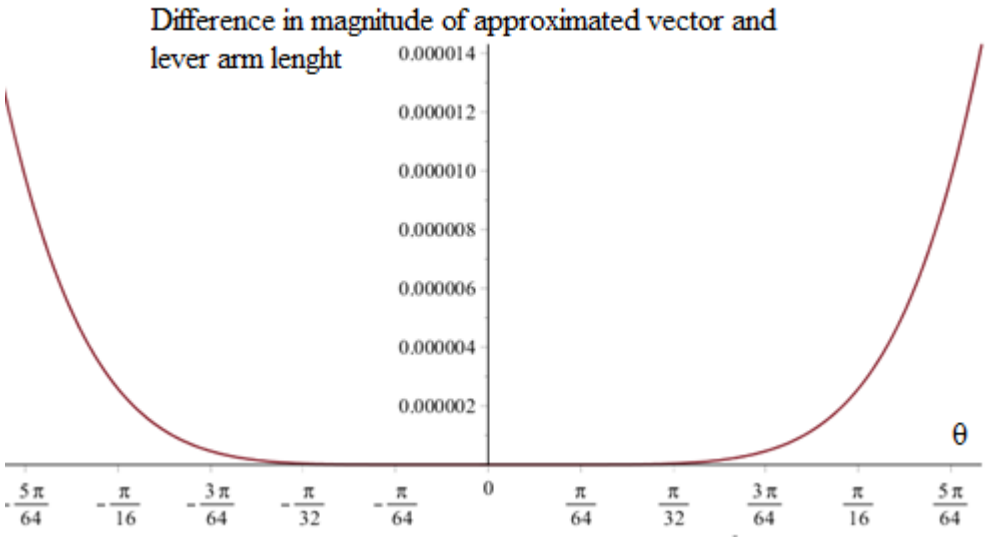


Figure 4: arm lever (ha-ry) length minus the magnitude of the roll vector **(8)** in meters

This graph shows us that the error introduced by the approximate angle function previously used to define the roll vector is negligible (microns are considered negligible). Note that a roll of ±15º would translate into a whiplash inducing ride.

Finally the position roll vector **(8)** was plotted in a three dimensional Cartesian space to confirm that the roll vector was consistent with the physics of the problem (i.e no discontinuities):

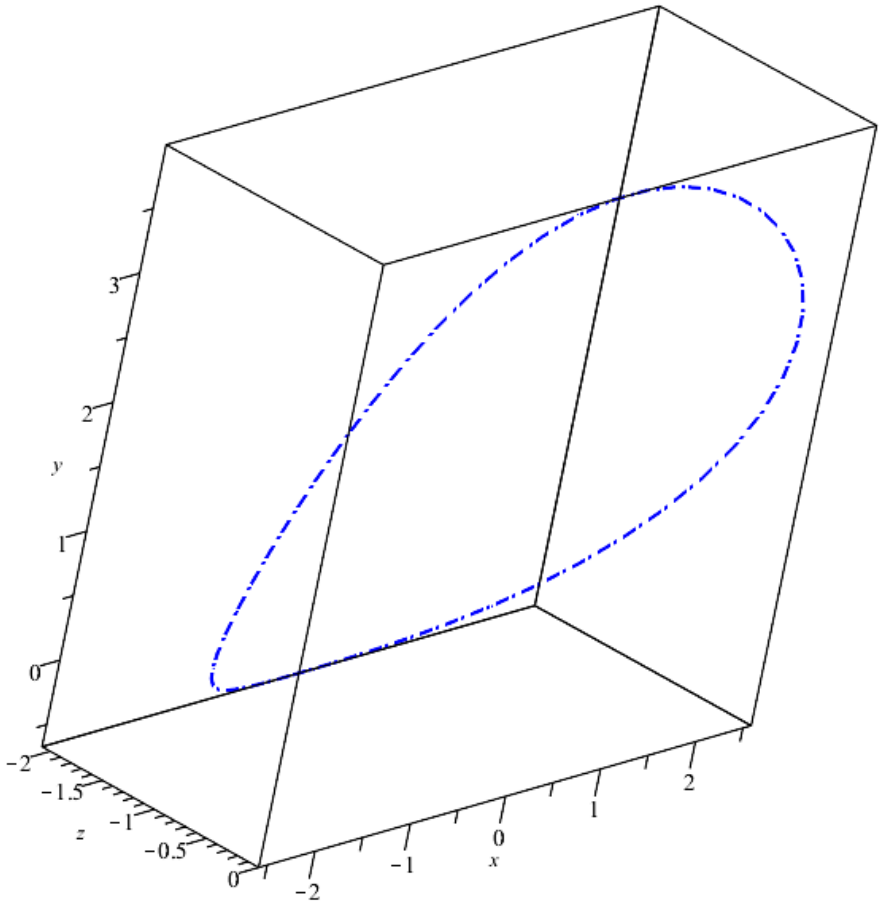


Figure 5: Position roll vector **(8)** in three dimension space in meters.

*The graph above represents a complete rotation of the vehicle around the roll axis. Although not realistic (unless the driver is performing some serious stunts) it shows that the roll vector **(8)** provides a continuous position in space which further validates the model.*

2.3 Pitch axis analysis

The pitch axis intersect with the roll axis at the coordinates [0,py,pz]. Unlike roll, movement around the pitch axis occurs only in the YZ plane. In order to integrate the pitch to the roll vector, it is necessary to consider the rotation around the pitch axis of the arm lever of the antenna for a given pitch angle Ω. This will require us to study the pitch of both antenna location [0,ha,0] (let's call it V1) and anchor point of the arm lever of the antenna [0,ry,0] (let's call this V2). Therefore (V1-V2) will actually be the arm lever rotated around the pitch axis. In addition, the pitch vector must have its origin at [0,0,0] since the true origin point of the position vector is already included in the roll vector **(8).**

2.3.1 Antenna pitch

In order to calculate the pitch of the antenna, it is necessary to calculate the distance (radius) of the antenna [0,ha,0] to the pitch axis at [0,py,pz] and the angle formed by the subtended radius from the antenna location to the pitch axis and the horizontal (i.e. Ω=0º).

R1=

$$\sqrt{(pz^2 + (ha - py)^2} \quad \textbf{(10)}$$

Ω2 =

$$\tan^{-1}\frac{pz}{ha - py} \quad \textbf{(11)}$$

Understanding that the pitch is simply the antenna moving along the circumference of a circle with a radius as defined by **(10)** and a start angle defined by **(11)** is it possible to derive these equations for the pitch vector of the antenna.

Y component, antenna pitch =

$$R1 \cdot \cos(\Omega 2 - \Omega) + py - ha$$

(12)

Z component, antenna pitch=

$$R1 \cdot \sin(\Omega 2 - \Omega) - pz$$

(13)

2.3.2 Arm lever anchor point pitch

In order to calculate the pitch of the arm lever anchor point at [0,ry,0], it is necessary to calculate the distance (radius) of the arm lever anchor point [0,ry,0] to the pitch axis at [0,py,pz] and the angle formed by the subtended radius from the arm lever anchor point to the pitch axis and the horizontal (i.e. Ω=0º).

R2=

$$\sqrt{(pz^2 + (ry - py)^2}$$

(14)

Ω3 =

$$\tan^{-1}\frac{pz}{ry - py}$$

(15)

Understanding that the pitch is simply the arm lever anchor point moving along the circumference of a circle with a radius defined by **(14)** and a start angle of defined by **(15)** it is possible to derive the following equation for the pitch vector of the arm lever anchor point.

Y component, arm lever anchor point pitch =

$$R2 \cdot \cos(\Omega 3 - \Omega) + py - ry$$

(16)

Z component, arm lever anchor point pitch =

$$R2 \cdot \sin(\Omega 3 - \Omega) - pz$$

(17)

2.3.4 Definition of pitch vectors

Now that X, Y, Z components have been established, it is possible to define the Cartesian [X,Y,Z] position vector that will provide the position of the antenna for a given pitch angle. Unlike the roll vector **(8),** there are two pitch vectors to consider; the antenna pitch vector and the arm lever anchor point pitch vector.

Antenna pitch vector $[X,Y,Z]^T$=

$$\begin{bmatrix} 0 \\ R1 \cdot \cos(\Omega 2 - \Omega) + py - ha \\ R1 \cdot \sin(\Omega 2 - \Omega) - pz \end{bmatrix}$$

(18)

Lever arm anchor point pitch vector$[X,Y,Z]^T$=

$$\begin{bmatrix} 0 \\ R2 \cdot \cos(\Omega 3 - \Omega) + py - ry \\ R2 \cdot \sin(\Omega 3 - \Omega) - pz \end{bmatrix}$$

(19)

2.3.5 Validation of pitch vector model

In order to validate the pitch model, we'll use the same variable assignment as in **(9)** in addition to the following:

Pitch vector variable:

$$pz = 1 \quad py = (rz - pz) \cdot \tan(\varphi)$$

(20)

First, it is possible to confirm that for the given set of variables, the pitch is null for a value of Ω=0. Second, the graph in Figure 6 confirms the circular behavior of the antenna around the pitch axis.

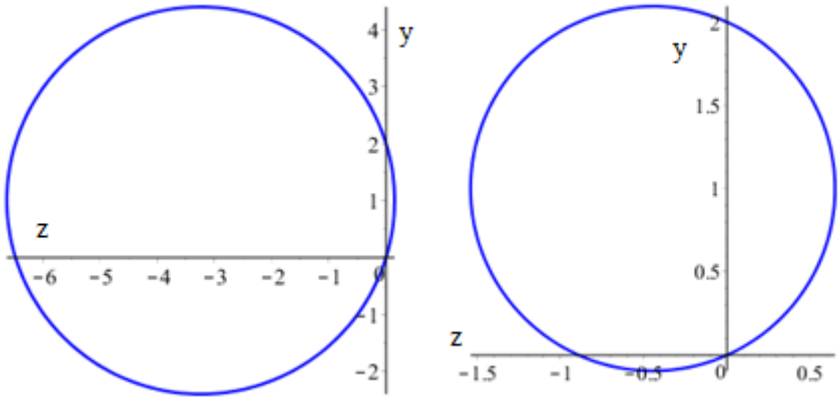


Figure 6: Antenna pitch **(18)** (left) Arm lever anchor point pitch **(19)** (right)

*Both pitch vectors are relative to zero, meaning that for a null pitch angle, their values are null. However, once incorporated with the roll vector **(18)** the origin point will be added meaning that for a null pitch angle the antenna pitch position will be [0,ha,0] and the , lever arm rotation point position will be [0,ry,0].*

Finally, the magnitude of the difference between vector **(19)** and **(18)** was evaluated. To reiterate, this difference corresponds to the length of arm lever of the model and should equate to (ha-ry). Using the variables already established in **(20)** and **(9)**, it can be shown that the error in magnitude is null, which further validate the pitch vector model.

2.4 Antenna position & movement analysis

The antenna position in space as a function of the roll angle and the pitch angle can be represented by adding the roll **(8)** and the antenna pitch **(18)**

Antenna position in space $[X,Y,Z]^T$ =

$$\begin{bmatrix} (ha - ry) \cdot \sin(\theta) * \cos\left(\varphi \cdot (1 - \cos(\theta))\right) \\ \frac{ha - ry}{2} \cdot \left((\cos(\theta) - 1) \cdot \left(\cos\left(\varphi \cdot (1 - \cos(\theta))\right)\right) + \cos(\theta)\right) + \\ R1 \cdot \cos(\Omega 2 - \Omega) + py + \frac{ry - ha}{2} \\ \frac{ha - ry}{2} \cdot \sin(\varphi\left(1 - cos(\theta)\right) \cdot (cos(\theta) - 1) + R1 \cdot \sin(\Omega 2 - \Omega) - pz \end{bmatrix} \quad \textbf{(21)}$$

Using **(21)** it is now possible to plot the position of the antenna in space.

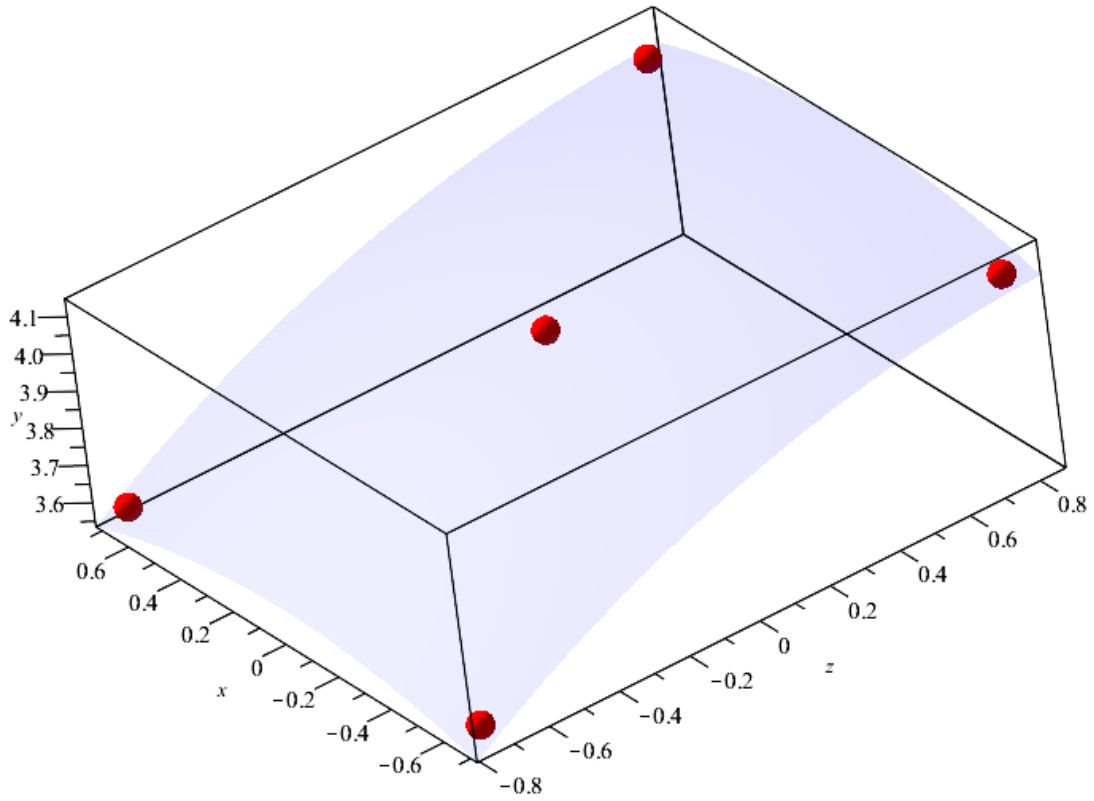


Figure 7: Antenna position in space represented by **(21)**

Note that the graph is not to scale. It shows the position of the antenna in space for roll & pitch angle of ±15° which represents a very, very rough ride. The points represent the origin and the extreme of the graph i.e θ=±15° and Ω =±15°

Even if knowing the actual position of the antenna in space is interesting in itself and does confirm the physics of the model, **(21)** is insufficient to provide angular & orientation information on the antenna. In order to extract this information, it is required to consider the arm lever on which the antenna moves during the roll. Given **(21)** already provides us with the position in space of the tip of the lever arm, it is required to obtain the position vector of the anchor point of the arm lever. This position vector can be obtained by adding the origin ry to the Y component of to the arm lever anchor point position vector (**19):**

Arm lever anchor point position in space $[X,Y,Z]^T=$

$$\begin{bmatrix} 0 \\ R2 \cdot \cos(\Omega 3 - \Omega) + py \\ R2 \cdot \sin(\Omega 3 - \Omega) - pz \end{bmatrix} \quad \textbf{(22)}$$

It is now possible to obtain the vector representing the actual lever arm on which the antenna pivots. This vector will allow us to extract the accurate angular information caused by roll and pitch. The vector can be obtained by subtracting **(22)** from **(21)**

Antenna Arm Lever $[X,Y,Z]^T=$

$$\begin{bmatrix} (ha - ry) \cdot \sin(\theta) \cdot \cos\left(\varphi \cdot (1 - \cos(\theta))\right) \\ \frac{ha - ry}{2} \cdot \left((\cos(\theta) - 1) \cdot \left(\cos\left(\varphi \cdot (1 - \cos(\theta))\right)\right) + \cos(\theta)\right) + \\ R1 \cdot \cos(\Omega 2 - \Omega) - R2 \cdot \cos(\Omega 3 - \Omega) + \frac{ry - ha}{2} \\ \frac{ha - ry}{2} \cdot \sin(\varphi \left(1 - cos(\theta)\right) \cdot (cos(\theta) - 1) + R1 \cdot \sin(\Omega 2 - \Omega) - R2 \cdot \sin(\Omega 3 - \Omega) \end{bmatrix} \quad \textbf{(23)}$$

Given that each component of this vector will be of great use, they can be defined as:

$$\chi = (ha - ry) \cdot \sin(\theta) * \cos\big(\varphi \cdot (1 - \cos(\theta))\big) \quad \textbf{(24)}$$

$$\upsilon = \frac{ha - ry}{2} \cdot \Big((\cos(\theta) - 1) \cdot \big(\cos\big(\varphi \cdot (1 - \cos(\theta))\big)\big) + \cos(\theta)\Big) + R1 \cdot \cos(\Omega 2 - \Omega) - R2 \cdot \cos(\Omega 3 - \Omega) + \frac{ry - ha}{2} \quad \textbf{(25)}$$

$$\zeta = \frac{ha - ry}{2} \cdot \sin(\varphi \left(1 - cos(\theta)\right) \cdot (cos(\theta) - 1) + R1 \cdot \sin(\Omega 2 - \Omega) - R2 \cdot \sin(\Omega 3 - \Omega) \quad \textbf{(26)}$$

2.5 Antenna angular & orientation analysis

As previously cited in [1], the amplitude of movement of a comfortable ride in an automobile is 2 inches or 5 cm. Taking in account the fact that an antenna on the roof of a vehicle will be subject to a greater amplitude of movement than its passenger in the driver seat (being much further away from the roll/pitch axes than the vehicle passengers), I can safely assume (including a safety factor) that the antenna would need to compensate for an amplitude of movement in the 8 inch (20 cm) range, leading us to calculating the corresponding roll and/or inputs that would produce such amplitude of movement.

2.5.1 Amplitude of movement

To obtain the amplitude of the movement of the antenna, it is required to redefine the antenna lever arm **(23)** so that the antenna is the origin point of the coordinates system

(i.e. for θ= Ω=0 then antenna lever arm[X,Y,Z]=[0,0,0]). In order to do so, (ha-ry) must be subtracted from υ **(25).** The magnitude of the resulting vector will represent the amplitude of the movement to which the antenna will be subjected for any value of roll and/or pitch angle.

Amplitude of movement=

$$\sqrt{\chi^2 + \left(\upsilon - (ha - ry)\right)^2 + \zeta^2} \qquad \textbf{(27)}$$

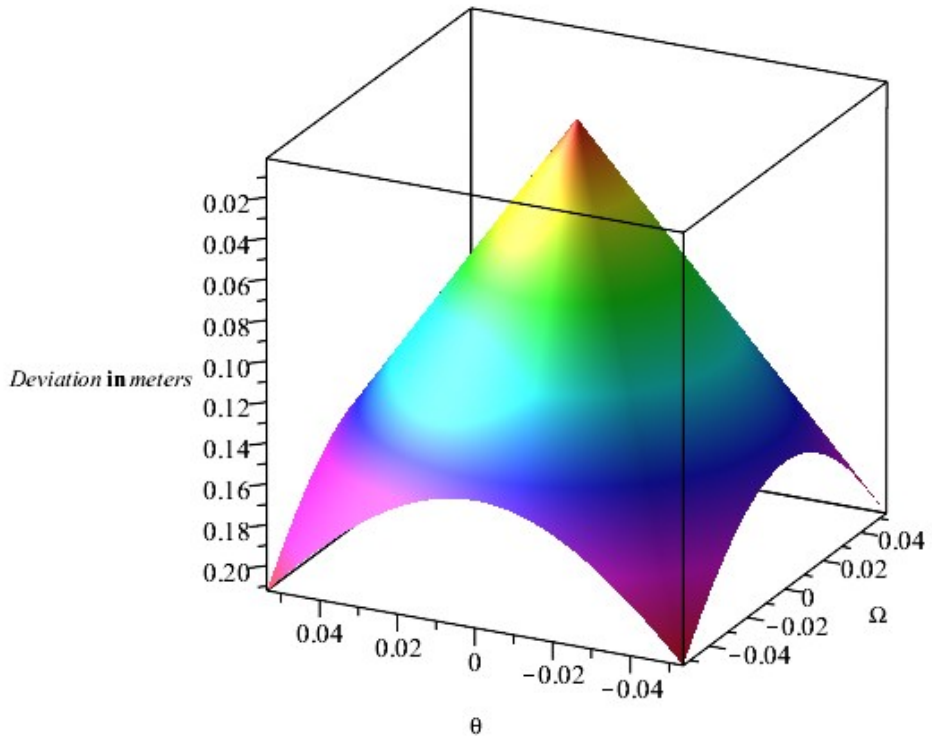


Figure 8: Amplitude of movement of the antenna

In this case, the amplitude of movement is plotted for a value of θ=Ω=±3.07° (displayed in radian of the graph.) which translate into a maximum amplitude of 0.2 meters. Numerical methods were used to establish the values of θ and Ω.

Considering Figure 8 and the antenna arm lever **(23),** it is now possible to obtain angular & orientation information with regards to the antenna. Let us define the following angles:

- ρ which will represent the absolute angle deviation from the vertical. (i.e the magnitude of angular variation of the antenna for a given roll and/or pitch angle)
- Θ which will represent the orientation of the absolute angle deviation from the vertical (i.e. the phase of the angular variation of the antenna for a given roll and/or pitch angle)

2.5.2 Absolute angle ρ

The absolute angle can be calculated by using basic trigonometry and vector components from **(24-26).** Note the conversion to degrees from radians for ρ.

$$\rho = \frac{180}{\pi} \cdot \tan^{-1}\left(\frac{\sqrt{\chi^2 + \zeta^2}}{\upsilon}\right) \quad \textbf{(28)}$$

Using **(28**) in addition of the roll and pitch established in Figure 8 it is possible to obtain the following graph to represent the absolute values of the magnitude ρ.

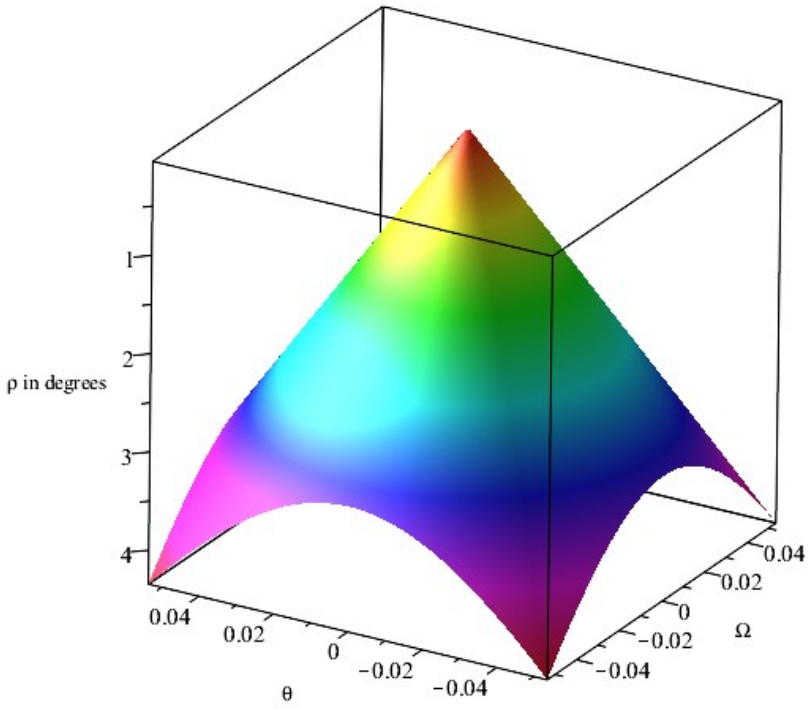


Figure 9: Absolute angle ρ.

Again this graph covers a range of θ=Ω=±3.07° as in Figure 8. Given this information, it is possible to deduce that a self-stabilizing antenna should be able to adjust its beams in a fashion to cover a range of ρ *of 3.5°in order to be efficient. Again these graphs hold truth for an antenna located 2.8 meters off the ground and the other assigned variables in* ***(20)*** *and* ***(9)*** *(note that* ***ha*** *was changed from 4 to 2.8, the height of an armored vehicle).*

2.5.3 Orientation angle Θ

Now that that ρ is defined, it is necessary to outline its orientation. As for ρ, Θ is defined using basic trigonometry and vector components from **(24-26).** Note the conversion to degrees from radians for Θ.

for Ω<0:

$$\Theta = 180 + \left(\frac{180}{\pi} \cdot \cos^{-1}\left(\frac{\chi}{\sqrt{\chi^2 + \zeta^2}}\right)\right) \quad \textbf{(29)}$$

for Ω≥0 :

$$\Theta = \frac{180}{\pi} \cdot \cos^{-1}\left(\frac{\chi}{\sqrt{\chi^2 + \zeta^2}}\right) \quad \textbf{(30)}$$

Using the function **(29-30)** in addition to the roll and pitch angle range established in Figure 8, it is possible to obtain the following graph to represent the orientation Θ.

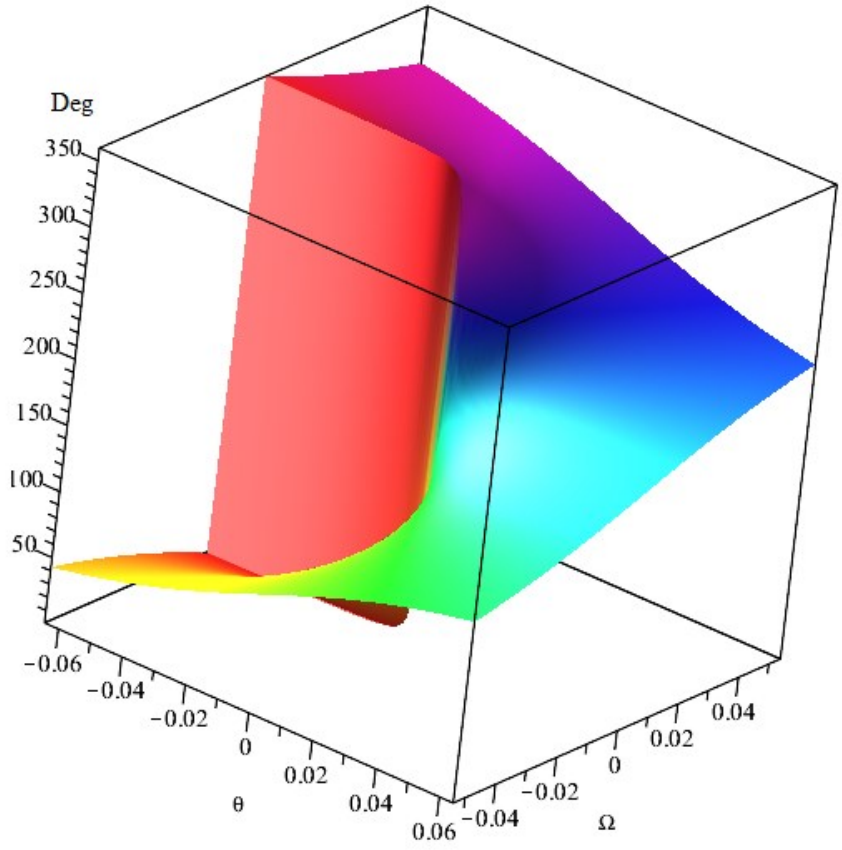


Figure 10: Orientation Θ of absolute angle ρ.

Again this graph covers a range of θ=Ω=±3.07° as in Figure 8. While ρ provides information with regards to the required magnitude of the correction, Θ provides the direction of the required correction..

2.5.4 Final validation of movement model

In reference with 2.2.5, it is necessary to confirm that the error in magnitude of the roll vector will not adversely affect the precision and/or efficiency of the antenna system. In order to do so, the magnitude of the antenna lever arm is measured and subtracted to the actual physical length of the lever arm (ha-ry). The resulting function is then plotted for the range predetermined in Figure 8 of θ=Ω=±3.07º

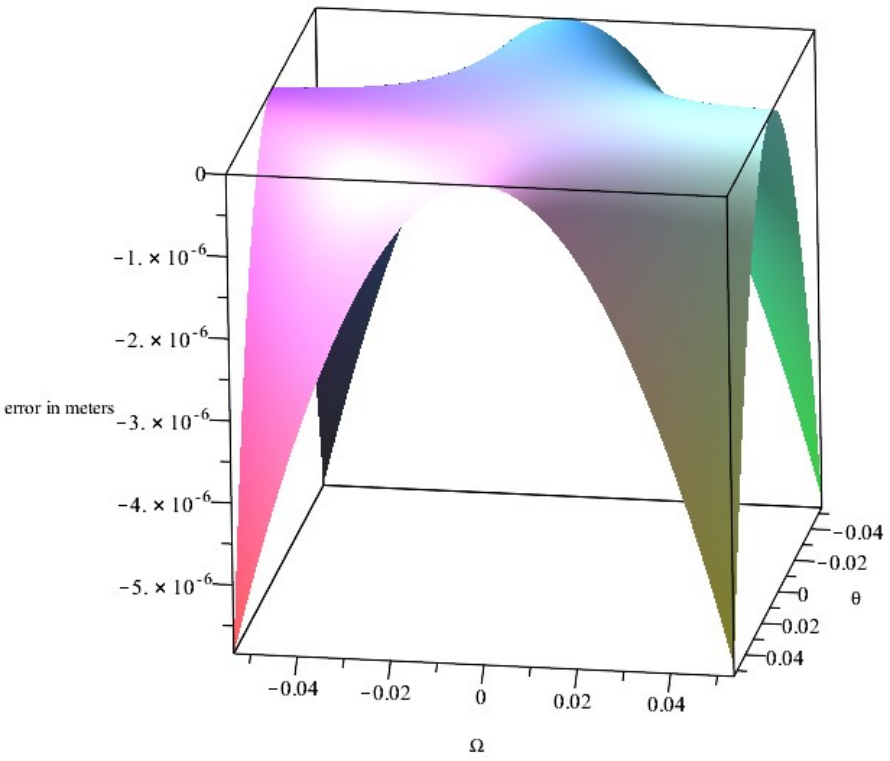


Figure 11: Magnitude error of vector model

Given the angle range of θ=Ω=±3.07º*, it is possible to conclude that the error caused by the roll vector is negligible given the application.*

2.6 Additional requirements

In order to be relevant, this antenna system should at least provide a gain of 30 dB or more over the coverage area. Below that level of gain, other much simpler and lighter antennas can be used to provide comparable gain.

Switching time should be a negligible portion of the operation time as well. Since one of the target applications is radar systems, 3SBOR should be able to switch its beam within a fraction of PRI.

3 TECHNOLOGY SELECTION

This thesis aims to prove the validity of the 3SBOR as a concept. Therefore, the choice of technology is directed by simplicity, affordability and ease of integreation. Optimization of performance is for future work.

3.1 Antenna selection

Given the primary purpose of the antenna system, a parabolic antenna is the best choice. Ranging from a Cassegrain to a dual offset antenna, the choices are multiple. Keeping in mind the multi-beam architecture of the system, having a center fed parabolic antenna is out of the question, since the feed network would create a blind zone on the center beam of a considerable and most likely unacceptable size.

Also, considering the key parameters of this selection, the best candidate is the standard offset reflector antenna. It is by far the simplest design available. The offset reflector antenna is an all-time favorite when it comes to mobile applications, since it allows for a rugged design and a placement of the feed network in a position where its weight and size does not hinder the performance and functionality of the system.

Figure 12: Example of a multi-beam offset reflector antenna

This Figure displays a very common type of multi-beam offset reflector antenna, which is used by Satellite TV providers across the world.

It is to be noted as well that a generic circular profile parabola will be used for the purpose of this thesis. The “orange peel” and other asymmetrical design shape were considered but deemed not required for the purpose of this thesis. Again, the aim is confirming the functionality of the system, not optimizing it. Furthermore, research has shown that a dish with the required diameter and Focal length to Diameter (F/D) ratio isn’t available commercially. Thus a custom offset reflector dish must be designed.

3.2 Horn Feed Selection

Unlike Figure 12, which shows a standard Low Noise Block (LNB) for satellite TV, the antenna system sought doesn't aim to simultaneously receive data from three specific points in space. It aims to maintain a reference position using a series of narrowly positioned pencils beam while the antenna is moving rapidly. This will represent a constraint with regards to the size of the feed elements.

Given the prototype frequency and the requirement for simplicity and affordability, a set of standard smooth walled pyramidal horn are used to feed the antenna. Finally, linear polarization will be used.

A WR-90 Standard Gain Horn Antenna, provided by Pasternack® with a corresponding flange adaptor and a directivity of 10 dB was selected. These standard gain horns were selected as they are readily available off-the-shelf and are the smallest horns available. The flange adaptor includes an End Launch SMA adaptor, which will allow for reliable and cost effective connection. This is contrary to the standard approach for offset reflector design, where the feed size is selected in order to maximize dish efficiency. In this case, simplicity was chosen over efficiency; rather than building an offset reflector and feed horn from scratch, I only have to design and build an offset reflector.

Figure 13: 10 dB standard gain horn antenna with wave guide adaptor (reference [23])

Even if they are the smallest available at 10 GHz, the WR-90 standard waveguide flanges and 10 dB horn antenna dimensions limit how close the feed horns can be "packed" together. This will require the dish to have a greater F/D ratio in order to obtain the required beam squint for continuous coverage. This will result in good illumination and good taper but with large spillover as well.

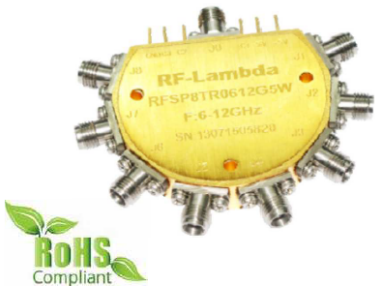
3.3 Feed network selection

Unlike standard multi beam antennas, this system transmits or receives from only one beam at the time. Nor will series or corporate feed be used to feed the horns, which will translate into great cost savings and weight reduction. The feed network solution consists of RF switches. Electromechanical switches, although providing very good insertion loss (in the order of 0.5 dB) are inadequate for this application, given their switching time of the order of ~25 milliseconds. Considering a MTI Radar with a PRI of 10 ms, the antenna system would miss ~2-3 pulses every time it switches, which is unacceptable. Solid state switches, although having a much greater insertion loss (in the order of 2-3 dB) present a

switching time in the range of 50-200 nanoseconds. This is so fast that the switching time would be negligible to any Moving Target Indicator (MTI) Radar.

Given that Iaren't creating a fully functional Radar but are proving the viability of the concept, power handling capability will not be considered. The insertion loss and switching time of the RF switch will be integrated in the comparative evaluation of the active antenna system against a single beam reflector antenna. For the purpose of this thesis, I will use the specifications of the 8 to 1 RF Switch provided by RF Lambda. This will provide a baseline for computer simulation of the system.

Reflective 6-12.0GHz Coaxial SP8T Switch



Features

- Wide Band Operation 6-12.0GHz
- High Power Handle Capability up to 100W upon request.
- TTL compatible driver include
- Fast Switching Speed
- Low Insertion Loss and High Isolation
- Temperature Range -40°C~+85°C
- Customization available upon request

Specification	Reflective type		
	PN: RFSP8TR0612G		
	Low	Med	High
Frequency Range(GHz)	6	9	12
Insertion Loss(dB)	1.6dB typ. @6GHz	2.0dB typ. @9GHz	2.5dB typ. @12GHz
VSWR	1.3	1.3	1.5
Isolation(dB)	85	85	80
Switching Speed(ns)	50	100	150
P1dB Power (dBm)		24	

Figure 14: 8 to 1 RF switch (reference [19])

This RF switch will provide more than sufficient switch speed at 10 GHz. It is to be noted that light radar system such as the MStar-V4 (according to specs sheet [7]) have a transmit power of less than 4 watts. While this is a generic switch, a purpose specific switch would provide improved performance for a given system.

3.4 Sensor selection

In this thesis, a sensor is required that will guarantee a working system. Given a vehicle vibrating at a frequency of 1-2 hertz, the sensor sampling speed should be of the order of 50-200 Hz in order to allow for precise measurement. Additionally, the precision of the sensor should be around ~0.1 of the antenna Half-Power Beam Width (HPBW) in both direction (i.e. roll angle and pitch angle). The MEMS inclinometer from TE Connectivity is shown in Figure 15

- 8 to 30 VDC supply voltage
- Digital signal processing includes
  - filter (e.g. vibration damping)
  - temperature compensation
- 12 bit resolution
- 100 Hz refresh rate
- -40 °C to 85 °C temperature range
- Accuracy typically
  - 0.5° | -40 °C to 85 °C
  - 0.15° | 25 °C

Figure 15: MEMS inclinometer by TE Connectivity (reference [21])

This sensor should provide more than sufficient performance for the prototype antenna system. It has an analog output and the precision and refresh rate will allow for precise position measurement. The range of measurement is 50^{o} [-25,+25].

3.5 Software selection

Matlab will be used to run simulations for the antenna system. A program provided by [2] and written by [8] is used to calculate the position of the feed horn in order to obtain adequate beam squint and continuous beam coverage. Given the hardware selection, a program is written to transition the Matlab simulation into a usable real-time algorithm. The exact program language used is determined by the hardware preference and is described in the following sub-section.

Since there aren't commercially available multi-beam antennas readily available to use as a reference for the simulation, it will be required to design a custom tailored antenna using the horn feed from sub-section 3.2. HFSS by ANSYS is the program of choice for this design.

3.6 Hardware selection

The computing power requirement for this system must be determined. Given that the required input/output are already known, it seems that a standard microcontroller based board with analog & digital I/O would be more than adequate. The concern here is ease of integration, guaranteed performance and widely available documentation and tutorials, not optimization of specific components. Based on the information available at this point the following microcontroller would suit the needs of this design:

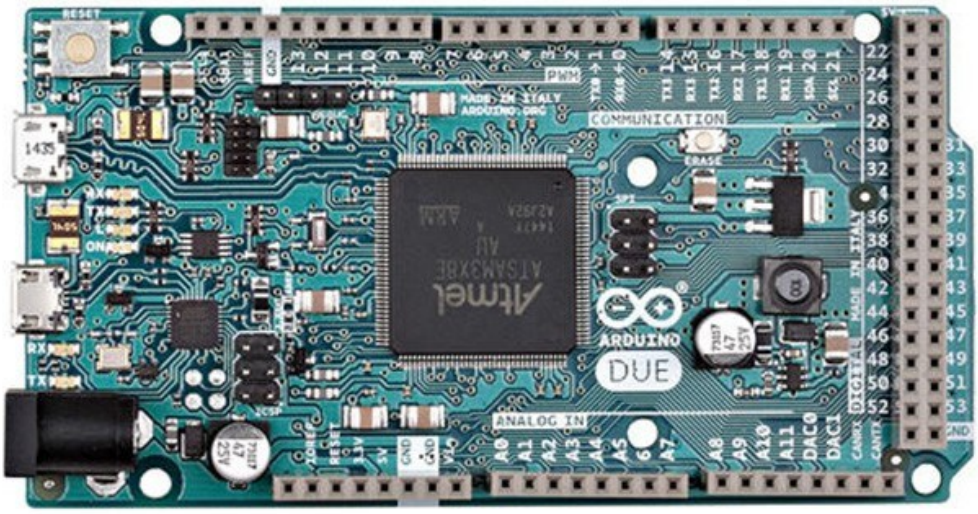


Figure 16: Arduino Due Microcontroller (reference [29])

This microcontroller should provide all the computing power and interface required to connect and operate the RF switch and sensor mentioned in sub-section 3.2 and 3.3 while maintaining the cost to a minimum.

3.7 Smoothing/Prediction filter selection

There are multiple approaches to this problem and the following were considered: Alpha-beta filters, Kalman filters and Particles filters. In order to select the appropriate filter a quick review of the system dynamics & requirements is required.

Firstly, the purpose of the smoothing/prediction filter is to obtain angular data that allows for acceptable performance based on the movement model described in Section 2 by using an inclinometer with a sampling rate of 100 Hz and its noisy output of roll and pitch

angles. The angular data obtained should allow for the selection of the appropriate beam in a timely fashion. It is critical to be at the right place at the right moment for the antenna system to function.

Secondly, the antenna system oscillates around the reference point (or resting position) and always returns to that point. Given this information, it should be possible to remove the mean measurement error by simply calculating a long term average (10-20 seconds) of the measured position. Furthermore, since the measured movement is periodic in nature and always returns to a resting/null position, there is no need to incrementally calculate the position from the starting point. This means that there will be no drift error in this system.

Thirdly, it must be assumed that the noise level of the output will be variable with time. A vehicle driving on a smooth, freshly paved highway will be subject to less vibration (less variance in angular data) than the same vehicle driving on washboard gravel road (loads of variance in angular data). In addition, potholes will represent statistical outliers that could be a source of error (i.e. a large input that throws off the long term average of the mean measurement error). Therefore, the smoothing filter should be able to accommodate a variable noise level and statistical outliers.

Fourthly, the vehicle is oscillating at a frequency of 1-2 Hz while the sampling of its movement is done at a 100 Hz. Given the oversampling, it is possible to assume that roll/pitch angular speed is constant over several samples.

Using the current knowledge of the system and its requirements, the solution seems to point to an Alpha-Beta filter. Its simplicity and low computational requirement should provide adequate performance for the system.

Let's recall the functioning of an Alpha-Beta filter; an Alpha ratio attributes a weight to new positional data being integrated as part of the overall measurement of the position. A Beta ratio attributes a weight to the new speed data being integrated as part of the measurement of the velocity of movement. A Standard low order Alpha-Beta filter adds the positional data with the velocity data to provide a more or less smoothed output.

Some improvements on the basic Alpha-Beta filter will be required to accommodate for mean measurement errors, statistical outliers and variable noise level. The first iteration of the design of the antenna system will focus on this strategy. This thesis focuses on showing the potential of a multi-beam antenna to actively track a reference point in space. The main purpose is not to develop the optimal tracking filter. In the event the Alpha-beta filter provides acceptable performance and allows for a favorable comparative assessment of the antenna system against a single beam offset reflector antenna, developing a Kalman/Particle filter will be future work.

4 3SBOR Antenna Design Process

Designing a multi-beam offset reflector antenna is an iterative process. For given dish dimensions a certain element size is required in order to allow for a certain beam squint. The dish will be designed as a function of the feed element size, sensor precision & movement model.

First, the dish dimensions must be calculated. Reference [4] and [5] provide a simple & efficient approach to designing a multi-beam offset reflector dish. Using these results, [8] will be used to confirm the beam width, gain, feed placement, beam squint and coverage area.

Once successful dish dimensions and feed placement are obtained, the design was simulated in High Frequency Simulation System (HFSS). Prior to starting this design process, an agreement for support was reached with the Quality Engineering Test Establishment (QETE). In accordance with this agreement, the design in HFSS was done assuming that it can be constructed with a half inch thick plywood structure. Additionally, a "one dimension" dish design in HFSS was requested by QETE in case the construction of the "two dimensions" dish would revealed itself to be too complex for their fabrication capabilities (see Section 6). The "one dimension" dish allows for beam squint in only one dimension (i.e. Y axis) whereas the "two dimensions" dish allows for beam squint in two dimensions (i.e. X-Y axis).

Upon successful simulation in HFSS, far field polar directivity data i produced in the form of Excel datasheets for each of the beams available. These datasheets are used to produce

a simulated directivity output in Matlab within a noisy environment on a moving platform.

A Matlab simulation will evaluate the performance of the 3SBOR by using a trio of control signal types; reference orientation signals, noisy sensor measurements of the reference signals and the same noisy measurement signals of the reference signals refined by the control algorithm. Then depending on the type of analysis, the orientation data from the three different types of signals will be used to construct a directivity graph as a function of time. This directivity graph will use the polar data previously obtained from HFSS and will allow an evaluation of the performance of the designed system.

4.1 Establishing Dish Dimensions

Reference [4] provides a straight forward & optimized design methodology for multi-beam offset reflector dish. However, I will have to rework this methodology to suit my needs. An offset reflector dish showing its parameters is displayed in Figure 17.

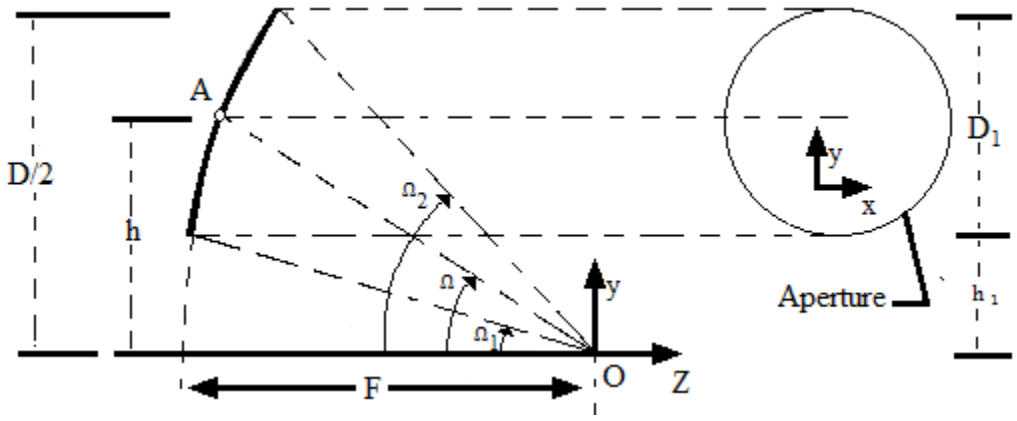


Figure 17: Offset reflector dish parameters & variables (reference [8])

This diagram will be referred to during this design section.

4.1.1 Tapers, efficiency & feed approximation.

The first step of the design is to establish the illumination and edge taper. This will be a key factor throughout the design and will dictate the antenna's efficiency.

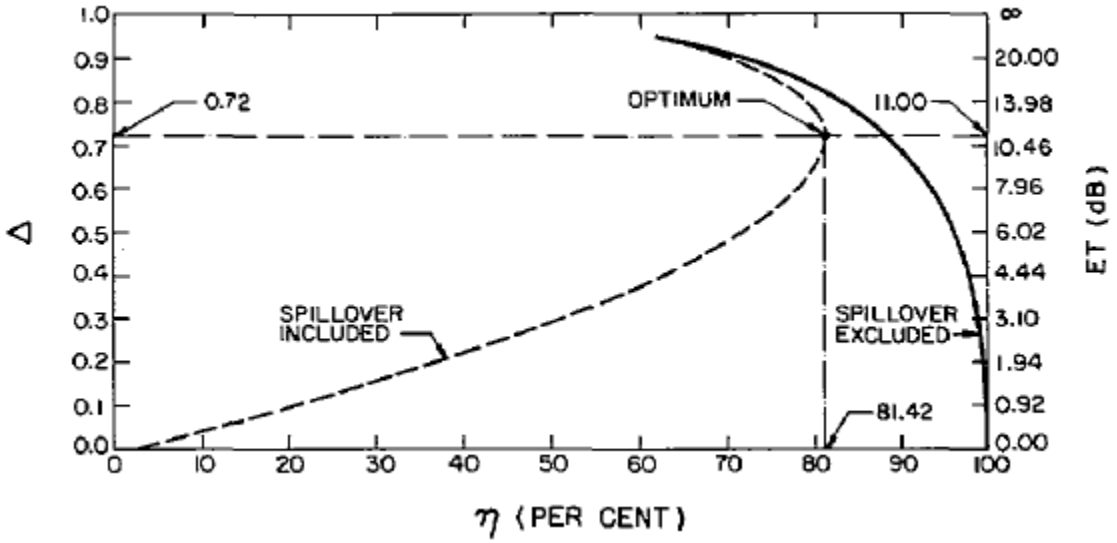


Figure 18: Dish efficiency in function of illumination taper Δ. (reference [8])

In order to maximise dish efficiency (the theory establishes maximum efficiency of a reflector dish at 81.42%) I must choose an illumination taper of 0.72 and an edge taper of 11 dB. However, such values aren't attainable given my equipment.

First the feed source must be characterized. At this time, it will be defined as a point source located at the point of origin O of Figure 17. Its beam pattern will be approximated as such:

Point source=

$$\cos(x)^q \qquad (31)$$

In Section 3, I selected a 10 dB standard gain horn as a feed source; I can therefore calculate a value of q for which **(31)** will approximate my feed horn as accurately as possible. This is done by "matching" the HPBW of **(31)** to that of the gain horn. Given the standard gain horn HPBW is 48.5° and solving the following:

3 dB beamwith:

$$0.5 = \cos\left(\frac{48.5}{2} \cdot \frac{\pi}{180}\right)^{q} \qquad \textbf{(32)}$$

yields a value for q of 7.6656.

Figure 18 shows the ideal illumination taper at 0.72. Such taper cannot be achieved with my current feed source because of feed size. At the end of this design process (according to [4]), it is possible to calculate the value of q by using the illumination taper and the angular size of the dish with reference to the feed source. Based on that value a horn feed would then be tentatively designed to match the approximate beam pattern of **(31)**. My approach is the opposite of this. I need to get the design to match the beam pattern for the horn feed I already have. After a few iterations and a Maple worksheet it was found that with the current requirements and the selected horn feed, the achievable illumination taper will be 0.428 with an edge taper of 4.85 dB. By drawing a line on Figure 18 at the corresponding edge taper and reading the intersection point with the *spillover included* curve, it can be determined that the dish efficiency should be around ~65 %.

4.1.2 SLL, HPBW & F/D ratio.

Using a curve fitted polynomial equation it is possible to relate the side lobe level (SLL) to taper efficiency (Δ) as:

$$\Delta = -19.02056 + 2.562754 \cdot SLL - 0.125929 \cdot SLL^2 + 0.00277094 \cdot SLL^3 - 0.000022869 \cdot SLL^4 \quad \textbf{(33)}$$

By substituting Δ=0.428 and solving for SLL, I find that the first side lobe should be around 20.5 dB. The next step is to find out the half power beam width (HPBW), which can be estimated using the illumination taper, dish diameter, operating wavelength and another curved fitted polynomial equation:

$$\text{HPBW} = \frac{2 \cdot 180}{\pi} \cdot \sin^{-1} \frac{\lambda * (r_3 \cdot \Delta^3 + r_2 \cdot \Delta^2 + r_1 \cdot \Delta + r_0)}{D1 \cdot \pi} \quad \textbf{(34)}$$

By assigning the following design values to **(34)**, it is possible to create a plot of the HPBW as a function of the dish diameter.

The design values are:

$$\lambda = \frac{3}{100} \quad \Delta = 0.428 \quad r_0 = 1.61 \quad r_1 = 0.57 \quad r_2 = -1.43 \quad r_3 = 1.47 \quad \textbf{(35)}$$

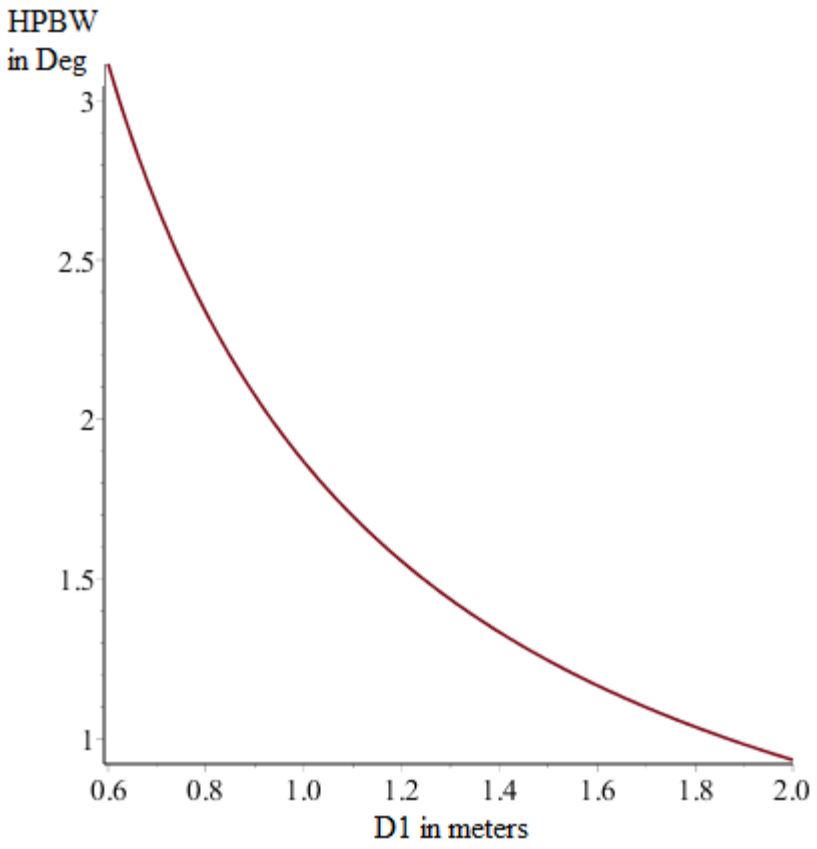


Figure 19: HPBW in function of dish diameter in meters.

This graph allows for a quick prediction of the HPBW in function of the dish diameter. Since 3SBOR is aimed for a mobile application, a valid design should maintain the dish size to less than a meter.

Choosing a HPBW of 2 degrees translates into a dish of 0.934 meters in diameter (represented by D1 on Figure 19.) By choosing this HPBW, I must keep in mind that it must be possible to collocate two horn feed and obtain a beam squint close to HPBW for the antenna system to function as intended.

In addition to than the horn feed spacing, another key factor to control the beam squint is the focal length to diameter ratio (F/D ratio). The higher the ratio, the flatter the dish, the hence greater horn feed spacing allowed for an equivalent beam squint. The ideal F/D

ratio in this case is the one that will allow for a beam squint of one HPBW while keeping the spacing between the horn feed to a minimum (1-2 mm of clearance between the horn feed). A series of iterations was required to find that for this horn feed, a F/D ratio of 1.25 should be adequate. Therefore, the Focal length (F on Figure 17) of the 3SBOR can be calculated as such:

Focal length (F) =

$$D1 \cdot (F/D) \tag{36}$$

which gives a focal length of 1.168 meter. Additionally I can now fix the offset height (h1, Figure 17) of the dish itself. Note that the purpose of the offset height is solely to allow enough clearance to prevent feed blockage. For structural reason, it is better to keep the offset height to a bare minimum. For this design the following is used:

Offset height (h1) =

$$0.05\ meter \tag{37}$$

Thus allowing for the calculation of the diameter (D, Figure 17), which will be used in the subsequent sections.

Diameter (D) =

$$2 \cdot (h1 + D1) \tag{38}$$

4.1.3 Scan width, Gain loss, Angular details & feed source matching

One major consideration for multi beam reflector dish design is the gain loss (in dB) resulting from the scan width. It is possible to calculate the scan width for a given gain loss as:

Scan width=

$$-\sin^{-1}\left(\frac{190\cdot F\cdot\sin(0.5\cdot HPBW)\cdot\left(1-e^{-0.12\cdot\sqrt{\frac{D1}{\lambda}}}\right)\cdot(\cos^{-1}(0.2\cdot GL-1)-\pi)}{D}\right)\cdot\frac{180}{\pi}$$

(39)

Using **(39)**, it is possible to produce the following Figure 20:

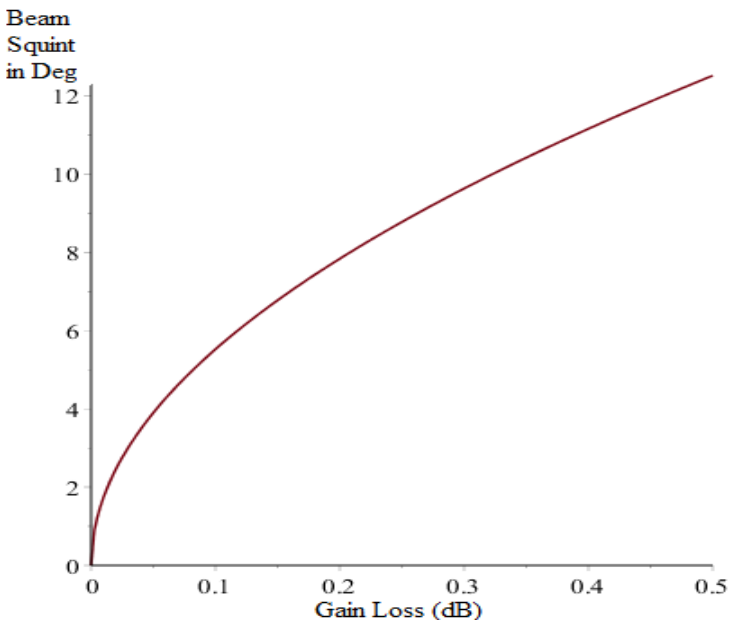


Figure 20: Gain loss in function of scan width

Considering the F/D ratio and HPBW previously chosen, this current design could scan 6 beams wide (~12 degrees) and the peak gain loss 6th beam would be 0.5 dB. However, this graph doesn't account for feed blockage.

This confirms that my current dimensions for the dish will allow for plenty of flexibility with regards to the scan width without the side effect of a large gain loss.

I must establish the angular design of the dish with regards to the feed source: Ω, Ω1 & Ω2 (reference Figure 17). Respectively, Ω is the angle at which the feed will be aimed at, while Ω1 & Ω2 are the bottom and top angle between the feed and the dish.

Ω1=

$$\tan^{-1}\left(\frac{\frac{h1}{F}}{1-\frac{1}{4}\cdot\left(\frac{h1}{F}\right)^2}\right) \quad \textbf{(40)}$$

Ω2=

$$\tan^{-1}\left(\frac{\frac{D}{2\cdot F}}{1-\frac{1}{4}\cdot\left(\frac{D}{2\cdot F}\right)^2}\right) \quad \textbf{(41)}$$

Ω=

$$\tan^{-1}\left(\frac{\frac{h1+0.5\cdot \mathrm{D1}}{F}}{1-\frac{1}{4}\cdot\left(\frac{h1+0.5\cdot \mathrm{D1}}{F}\right)^2}\right) \quad \textbf{(42)}$$

Given the current dimensions, Ω1 is equal to 0.0428 Radian or 2.453º, Ω2 is equal to 0.7977 Radian or 45.702º and Ω is equal to 0.4359 Radian or 24.969º. This data allow us to make a most interesting calculation: the value of q for **(31)**. Given that the value of q is dictated by the beam pattern of the feed horn, which in turn dictates the tapers & efficiency. Also the tapers then allow for the calculation of HPBW & SLL and the HPBW allows for the calculation of the size of the dish, and with a given focal length allows for the calculation of scan width, gain loss and angular details. This finally allows for the

calculation of q. The recursive nature of this design process is clear. A starting point was chosen and multiple iterations were required. Convergence was achieved by matching the following q values with that of **(32)** (i.e q=7.67) and the following:

q (second version)=

$$\frac{\ln(1-\Delta)}{\ln(\cos(0.5\cdot(\Omega 2-\Omega 1))}$$

(43)

Given my current design solution, this gives a value for q (second version) of 7.67 (matching **(32)**).

4.1.4 BDF & Feed blockage.

Calculating the Beam Deviation Factor (BDF) allows a precise position of the feed horn in a fashion that will guarantee an even beam distribution. The goal here is to have the beams crossing each other at the HPBW. The formula is quite cumbersome, so some substitutions are required:

τ=

$$\frac{\cos(\Omega)+\cos(\Omega-\Omega 1)}{1+\cos(\Omega-\Omega 1)}$$

(44)

BDF=

$$\tau\cdot\left(1-0.72\cdot e^{-3.2\cdot\left(\frac{F}{\tau\cdot D1}\right)}\right)$$

(45)

Given the current dimensions, the BDF stands at 0.9411. In order to relate the beam squint to feed spacing, the following is required (with some substitution):

l=

$$\mathrm{F}\cdot\sqrt{\left(1+\left(\frac{h1+0.5\cdot \mathrm{D1}}{2\cdot F}\right)^{2}\right)}$$

(46)

Beam squint=

$$\tan^{-1}\left(BDF\cdot\frac{d}{l}\right)$$

(47)

Given **(47)** I can now plot the beam squint in degrees resulting from the displacement d in meters of the feed horn.

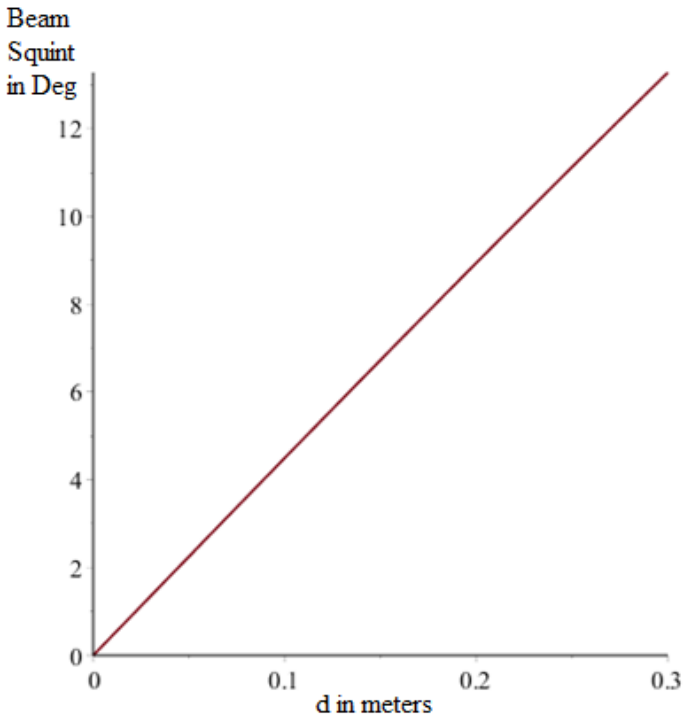


Figure 21: Beam squint is function of feed spacing

The relation between the feed spacing is linear for the scan width shown in Figure 21. Given the horn is placed at O in Figure 17, it will be possible to obtain the a 4 degree beam squint in any direction by moving the feed horn by 10 cm in the opposite direction of the desire beam squint in the plane perpendicular to the line between O and the angle Ω. Additionally, the gain loss would be only 0.1 dB.

Consequently, I can now calculate, using **(47)**, the spacing required to obtain a beam squint of one HPBW; 44 mm. This is perfect since the gain horn from Section 3 is 41.40 X 41.81 mm; giving a clearance of around 2 mm.

Finally, it must be confirmed how far the current dish design can squint a beam without suffering from feed blockage. In other to guarantee that there is no feed blockage, the following condition must be satisfied:

Feed blockage:

$$\mathrm{h1} \geq N \cdot d \cdot \sqrt{\left(1 - \left(\frac{h1 + 0.5 \cdot D1}{l}\right)^2\right)} \qquad \textbf{(48)}$$

where N is the number of feed horn for the current design. Of course, this condition doesn't apply when the beam is squinted laterally. (i.e. if the feed are stacked along x axis on Figure 17.). Given the offset height for this design (h1) is 0.05 meter, a maximum of one beam (i.e horn feed) will be possible in order to avoid feed blockage.

4.1.5 Achievable Gain

According to reference [8] the dish dimensions established throughout this section should provide a maximum gain 33.7 dB with a HPBW of 1.9º.

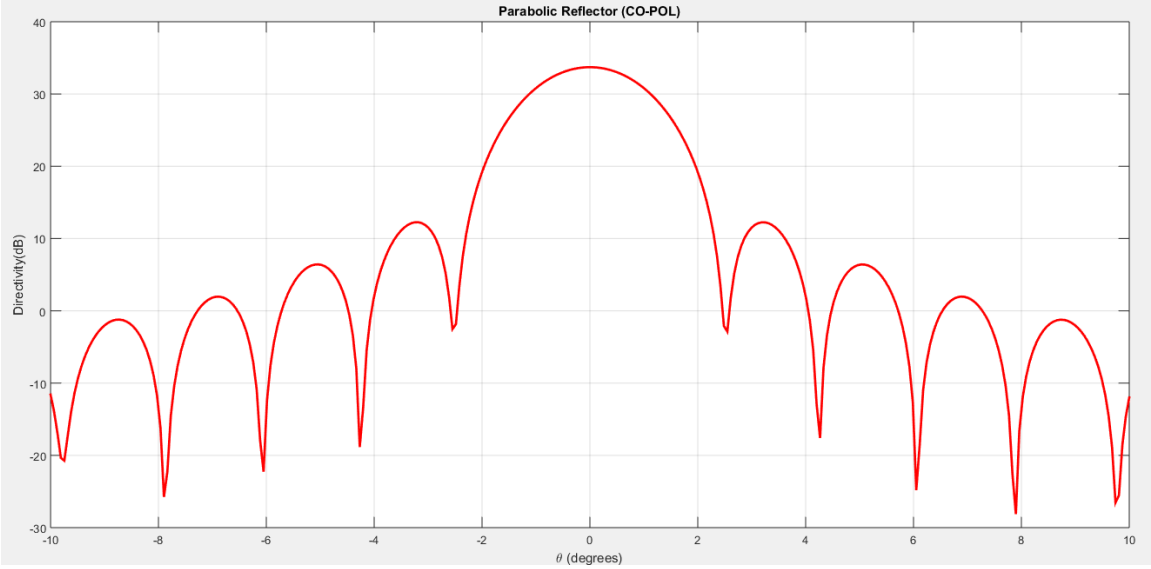


Figure 22: predicted directivity pattern x axis

According to this reference, the SLL should be around 13 dB, much lower than predicted by (33). The curved fitted polynomial predicted 20.5 dB SLL.

4.2 Feed horn placement

3SBOR is an antenna system aimed to be deployed on a moving ground vehicle. The ground vehicle does not yaw. It pitches or rolls. Hence the feed horn should be placed in a fashion to counter pitch and roll or a combination of both. Given this information, I can assume that no feed horn other than the center beam should be placed along the x axis as only yaw movement would make those horn feeds useful. Recalling that the goal is to demonstrate the viability of the 3SBOR, I will aim to keep the number of beams to a

minimum. The final configuration will use a total of seven horn feed, placed as shown in Figure 23:

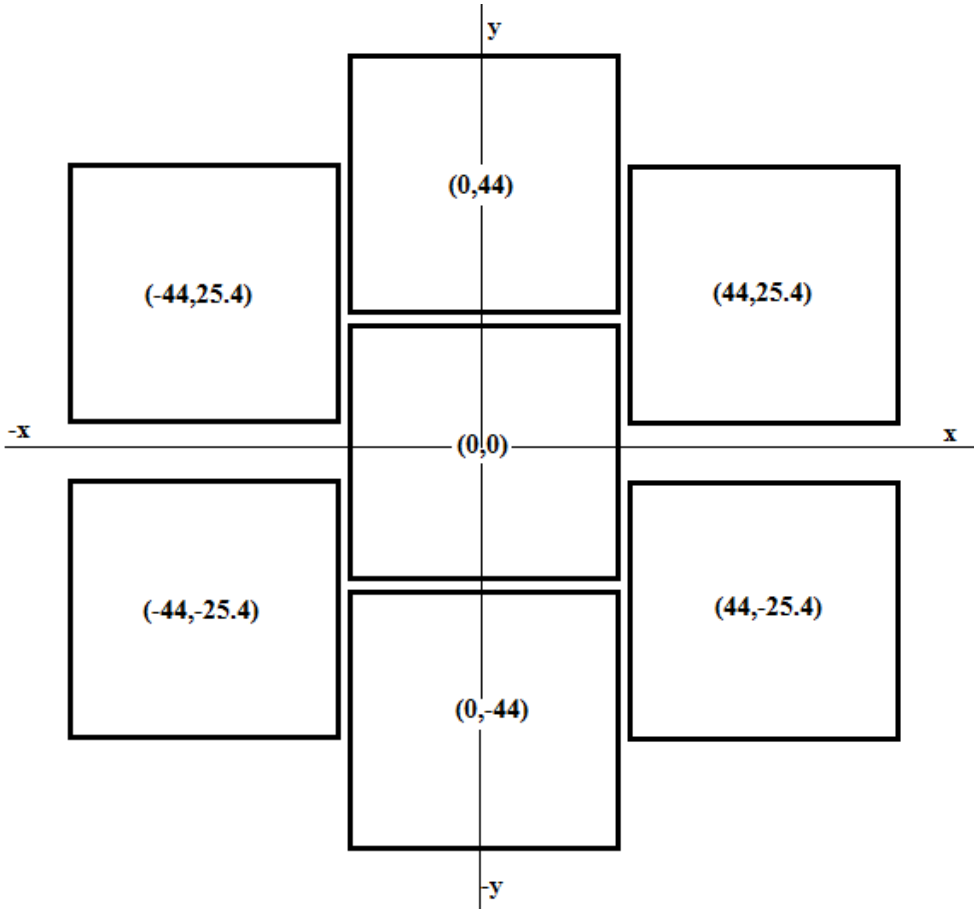


Figure 23: Feed horn placement

This Figure shows the center of each horn feed in Cartesian (X,Y) coordinates, in millimeter. The horns on the vertical axis are spaced by 44 mm from the origin point, which should yield a beam squint of 2°. However, the horns located at 30°, 150°, 210° & 330° are 50.8 mm from the origin, which will result in a beam squint of 2.27°.

On the following Figures, the x axis with equivalent feed spacing is shown.

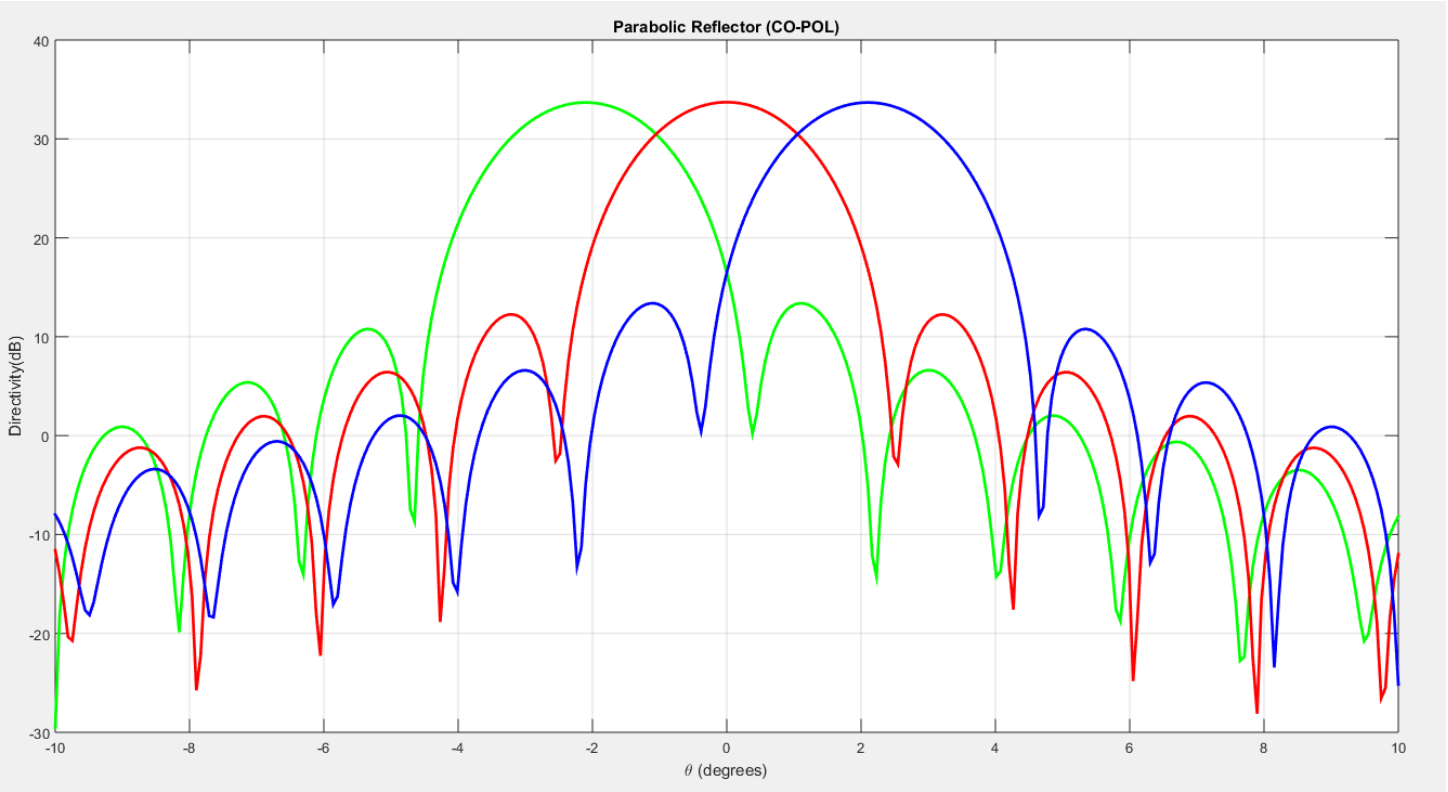


Figure 24: Y axis antenna pattern

Gain loss due to scan width is 0.04 dB and the beams intersect at 30.4 dB and ±1.05º. The peak gain of the main beam is 33.7 dB. The effective HPBW is 6.1º.

The observation angle is again maintained along the x axis in the beam pattern for the horn feed located at 30º & 210º. However the effective feed horn spacing used is 50.8 mm, which is the distance from the center of the center horn feed to the center of the 30º horn. The same graph applies for the feeds located at 150º & 330º.

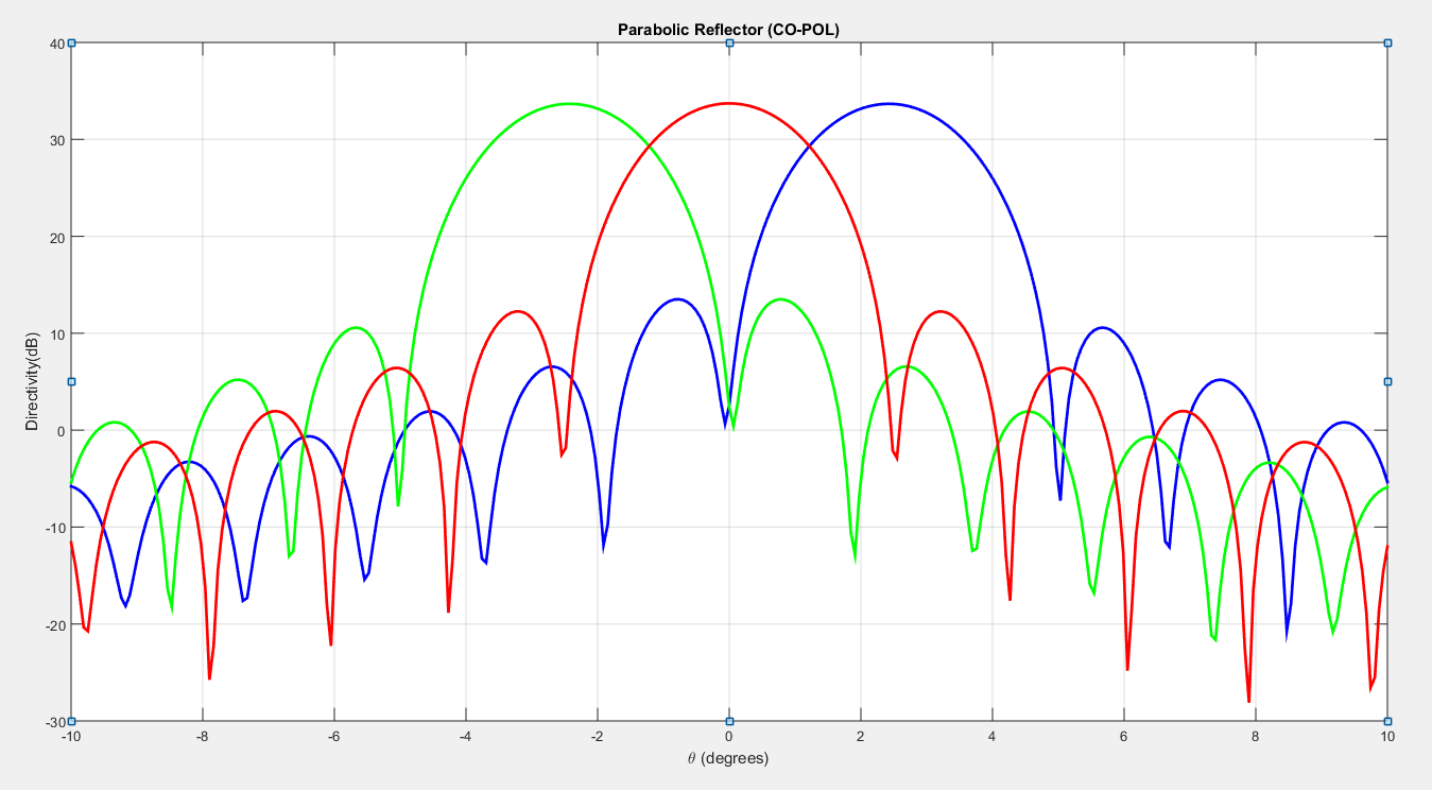


Figure 25: 30º or 150º antenna pattern

Gain loss due to the scan angle is 0.05 dB and the beam intersects at 29.3 dB and ±1.21º. The peak gain of the main beam is 33.7 dB. The effective HPBW is 6.1º.

Figure 24-25 show satisfactory preliminary beam patterns for the 3SBOR. Although some zones are slightly below the HPBW within the coverage area, it is assumed at this point that these areas are small enough that they shouldn't adversely affect the performance of the antenna system.

4.3 3SBOR in HFSS

The goal of constructing the 3SBOR in HFSS is twofold. First it will allow for a reliable 3d beam pattern to be generated and to confirm the dish design of the previous section.

Second, it will allow us to obtain data that can be used for system level simulation. Due to computing resources limitations, all structures in HFSS are assigned as perfect conductors.

4.3.1 Standard Gain Horn Construction

Using the specification of the datasheet available on the *Pasternack* website, the 10 dB Standard Gain Horn was constructed in HFSS:

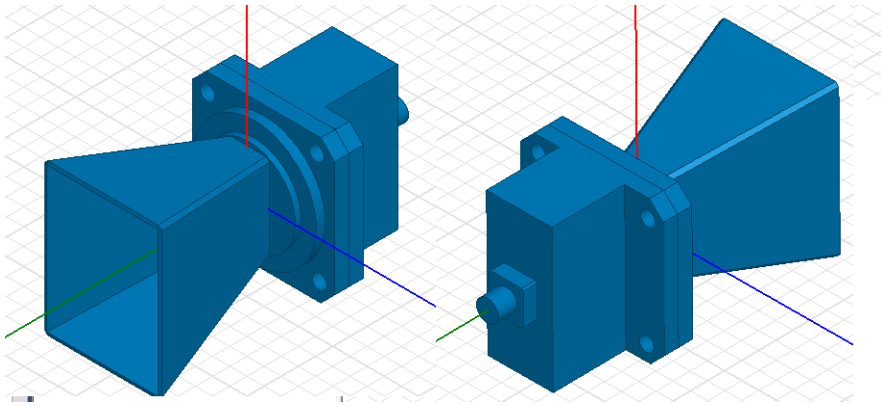


Figure 26: Horn feed in HFSS

Front and rear view of the Horn feed in HFSS. It is excited by a wave port with an integration line centered on the narrow section of the waveguide adaptor.

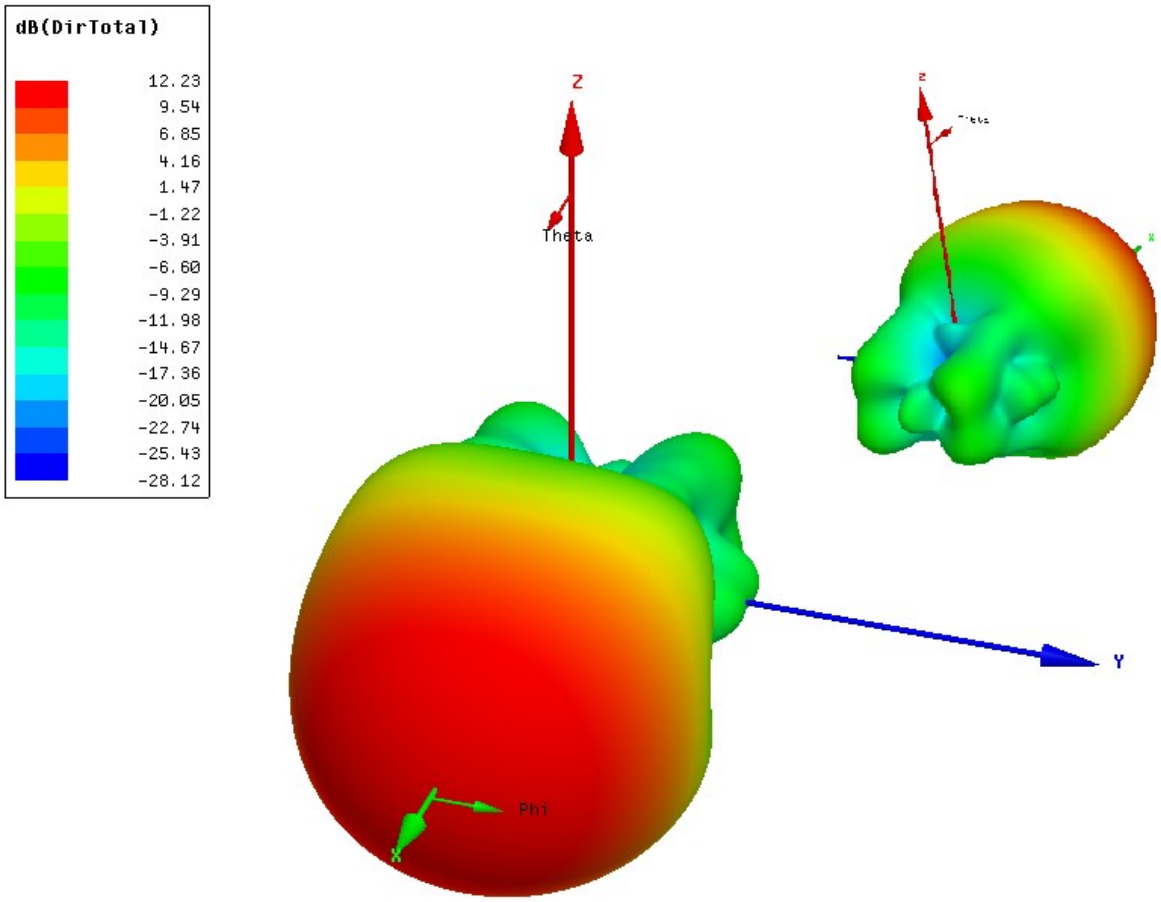


Figure 27: Horn feed in HFSS, 3d Polar plot

The maximum directivity here is 12.23 dB at 10 GHz with HPBW of ~47°. It is possible to see the back view in miniature on the top right of the Figure. This data agrees with the spec sheet provided by Pasternac. The horn is excited by a wave port located in the waveguide adaptor.

While this data seems convincing, it is necessary to evaluate how closely the current standard gain horn directivity pattern matches the approximation of **(32).** To do so a normalized XY plot of the horn directivity pattern (E & H planes) was superposed on a plot of **(32)** and the results are conclusive within the illuminated area of the dish.

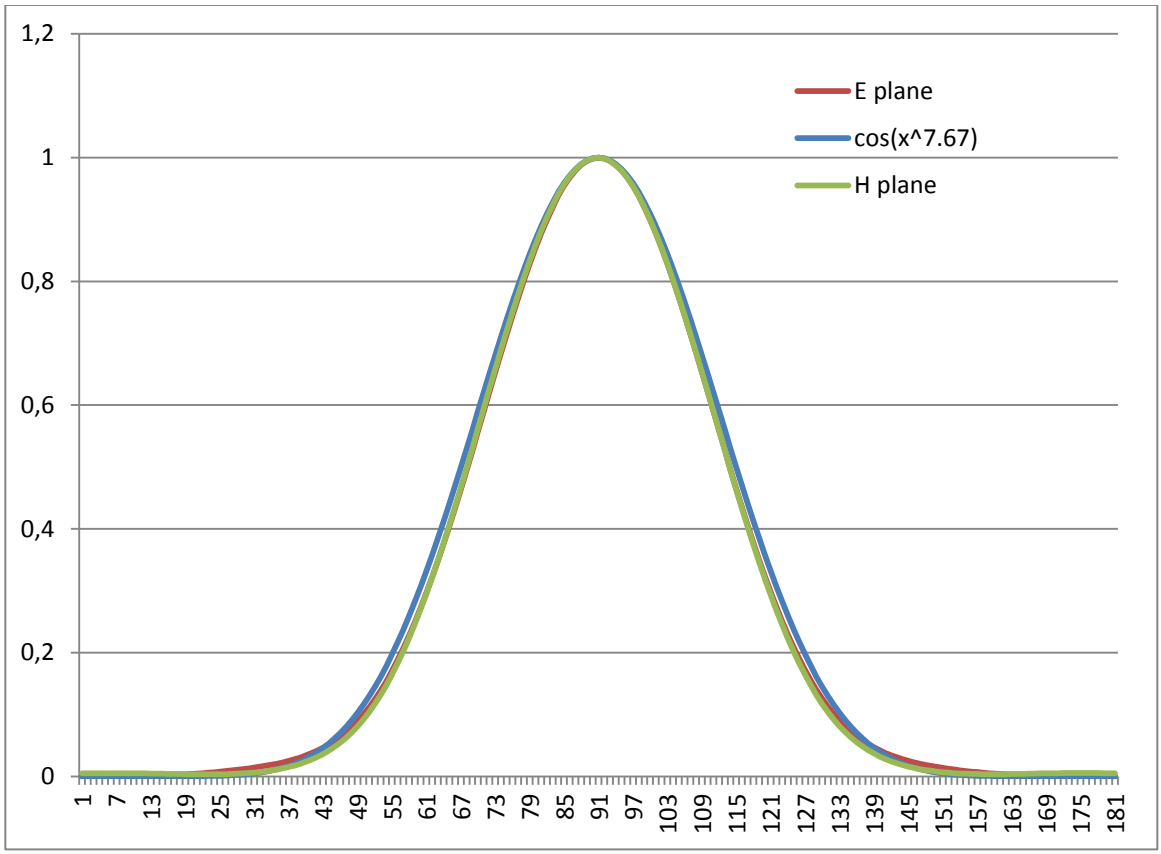


Figure 28: Horn feed in HFSS vs cos(x)^7.67

*The HFSS horn pattern and the cos^q approximation match closely. The center of this graph (90º) will be aimed at the dish at the angle of Ω (*24.969º*) from the horizontal. The dish will be illuminated by this pattern from 67.5º to 112.5º.Since the H-E planes decrease faster that cos^q (7% lower at the edge of the dish) Ican conclude that the illumination taper will improve equally by 7%, increasing from 0.428 to 0.457. The edge taper will increase as well from 4.85 dB to 5.03 dB. Dish efficiency should increase to 68%.*

4.3.2 Dish Construction

Prior to constructing the horn feed as per Figure 23, it is required to confirm the dish design, its gain, HPBW and SLL with a single horn feed. This allows us to correctly evaluate the detrimental effects of mutual interference between the gain horn antennas.

It is first necessary to understand that the construction of an offset reflector differs greatly from that of a center fed parabolic dish. A center fed dish is constructed as a parabolic curve rotated 360º around a center axis. An offset reflector is a two dimensional parabolic surface and is not a portion of a center fed dish. Creating an offset dish in HFSS is a multiple steps process; first the surface equation must be written.

Surface Equation:

$$X(u,v) = v \quad Y(u,v) = u \quad Z(u,v) = \frac{u^2 + v^2}{4 \cdot F} \quad u = [h1, D1] \quad v = \left[-\frac{D1}{2}, \frac{D1}{2}\right] \tag{49}$$

This creates a surface as in Figure 29:

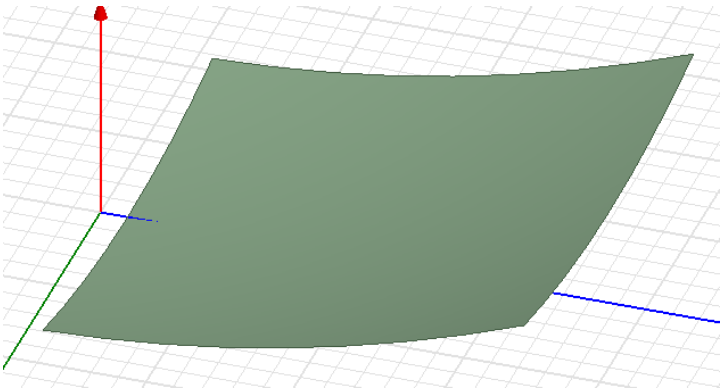


Figure 29: initial plane surface for dish construction

Now that my surface is created, it is necessary to remove the “excess area”. To do so, a relative coordinates system must be defined in HFSS. Two points of reference are necessary.

Reference point #1 (v=0, u=h1) & #2 (v=0, u=D1):

$$(x_0\ , y_{0,}\ z_0) = \left(0 \quad h1 \quad \frac{h1^2}{4 \cdot F}\right) \quad (\ x_1, y_1, z_1) = \left(0 \quad D1 \quad \frac{D1^2}{4 \cdot F}\right)$$

(50)

These points are located in the YZ plane (x=0).Using these two points, Ican now define a vector (Magnitude and Argument). Some simplifications are applied since the vector is located in the YZ plane.

Two dimensional position vector$[\ x_1, y_1, z_1] - [x_0\ , y_{0,}\ z_0]$:

$$Mag = \sqrt{(D1 - h1)^2 + \left(\frac{D1^2 - h1^2}{4 \cdot F}\right)^2} \quad Arg = \tan^{-1}\left(\frac{D1^2 - h1^2}{4 \cdot F \cdot (D1 - h1)}\right)$$

(51)

Using this vector, Ican calculate the origin point of the relative coordinate system, which is located right in the middle of **(51**).

Origin of relative coordinate system:

$$(\ x_2, y_2, z_2) = \left(0 \quad 0.5 \cdot Mag \cdot \cos(Arg) + h1 \quad 0.5 \cdot Mag \cdot \sin(Arg) + \frac{h1^2}{4 \cdot F}\right)$$

(52)

With the established design values, the origin point **(52)** is equal to [0, 0.4917, 0.0936], in meters. A normalized **(51)** defines the y axis of my relative coordinate system.

The following step is to establish the orientation of the x axis. Taking in account the parabolic nature of the surface, it is possible to calculate the intersection point of the parabolic surface with the x axis of the relative coordinate system.

Iare therefore looking for a third reference point that satisfies the following equations:

$$v = x_2 \quad u = \quad y_2 \quad \frac{u^2 + v^2}{4 \cdot F} = z_2$$

(53)

After solving **(53)** for x_2 the third reference point is defined as such:

$$(x_3, y_3, z_3) = \left(\pm\sqrt{4 \cdot F \cdot z_2 - {y_2}^2}, y_2, z_2\right)$$

(54)

This equals to [±0.4417, 0.4917, 0.0936] in meter. Ican now create the relative coordinates system (z axis being perpendicular to the XY plane).

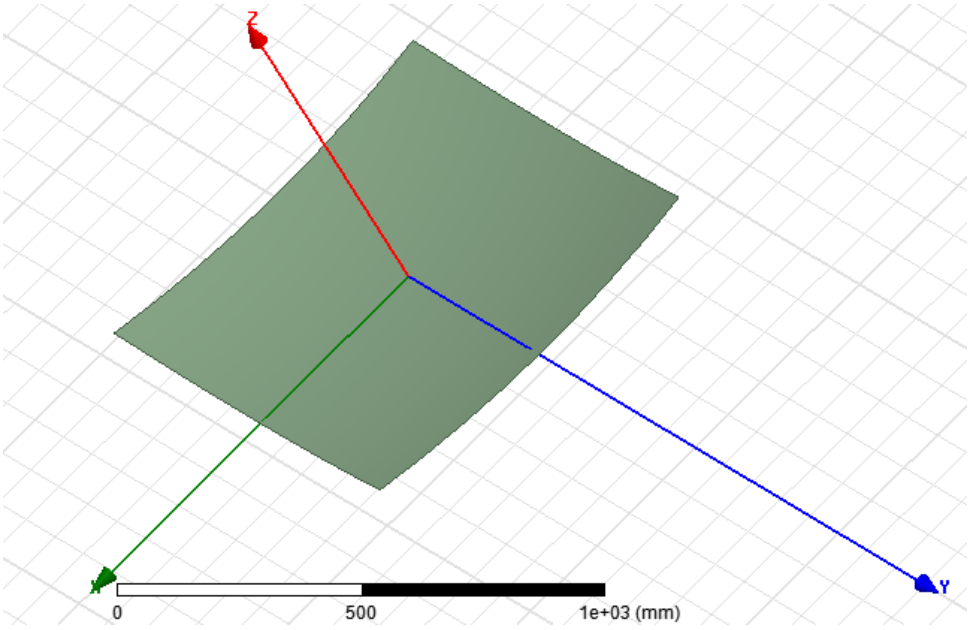


Figure 30: relative coordinates system displayed on dish

Although Ithoroughly calculated this relative system, it is possible to graphically place it in HFSS. This graphical method results in a slightly less accurate aperture size and gain.

The next step is to create a circle of the diameter in **(55)** representing the offset parabola's perimeter centered on the relative coordinate system shown in Figure 31.

Circle diameter=

$$\sqrt{(D1-h1)^2+\left(\frac{D1^2-h1^2}{4\cdot F}\right)^2}$$

(55)

After drawing the circle, Ican simply thicken the circle in both directions to create a cylinder:

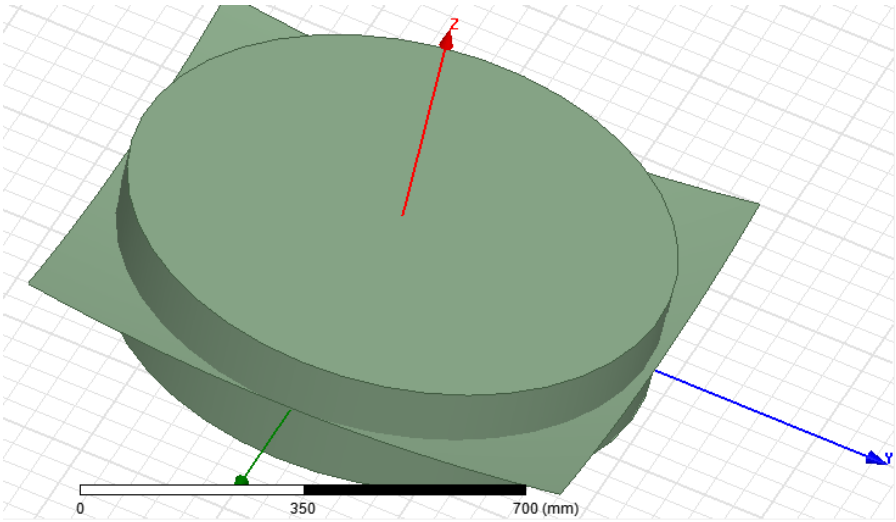


Figure 31: Aperture “cylinder”

With this method, it is possible to “tune” the beam width by changing the shape to an ellipse or by making the circle smaller. Figure 20 can be provided with an estimation of the required ellipse ratio or circle diameter to obtain the desired beam width.

Inow can subtract the Figure 32 to Figure 31, creating a “negative, which will be used to cut out the dish:

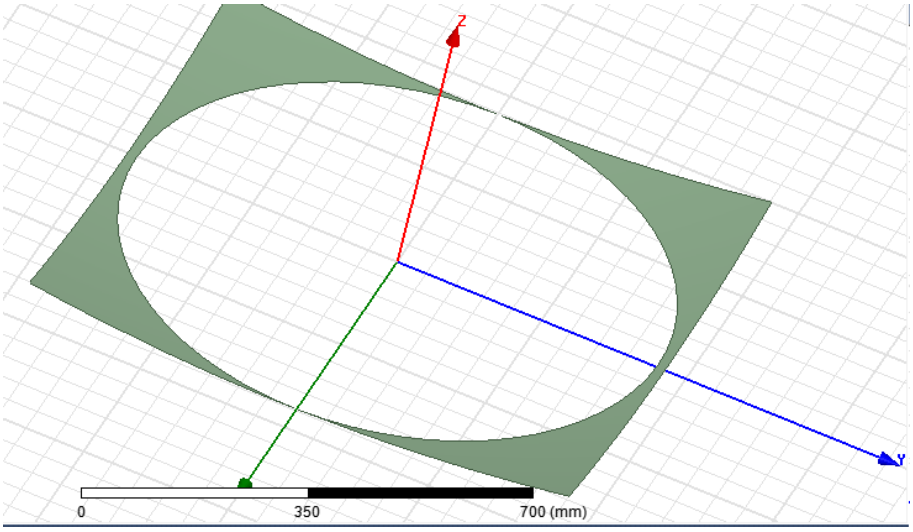


Figure 32 “negative” plane surface

This surface is subtracted from Figure 31. The result is the final offset dish as shown in Figure 33.

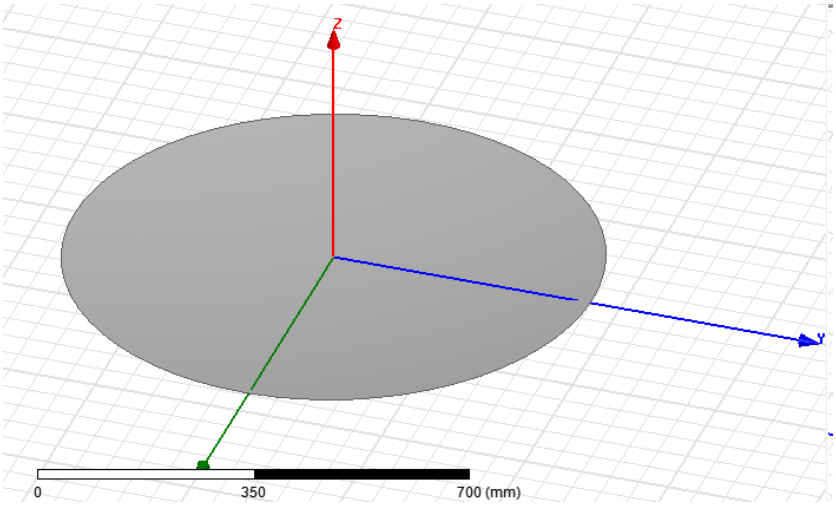


Figure 33 Completed Offset dish.

This dish will be used for future simulations. It was obtained by subtracting Figure 33 from Figure 31

4.3.4 Single horn feed simulation

The HFSS simulation is setup as described below. The single horn feed is placed in an HFSS design and orientated at the angle of Ω. A vacuum region around the horn is created and set as a radiation boundary. The dish is placed in an HFSS-IE design, defined as a perfect electric boundary. The dish is linked to the horn as an external source. The rectangular plot showing the H & E plane is shown below:

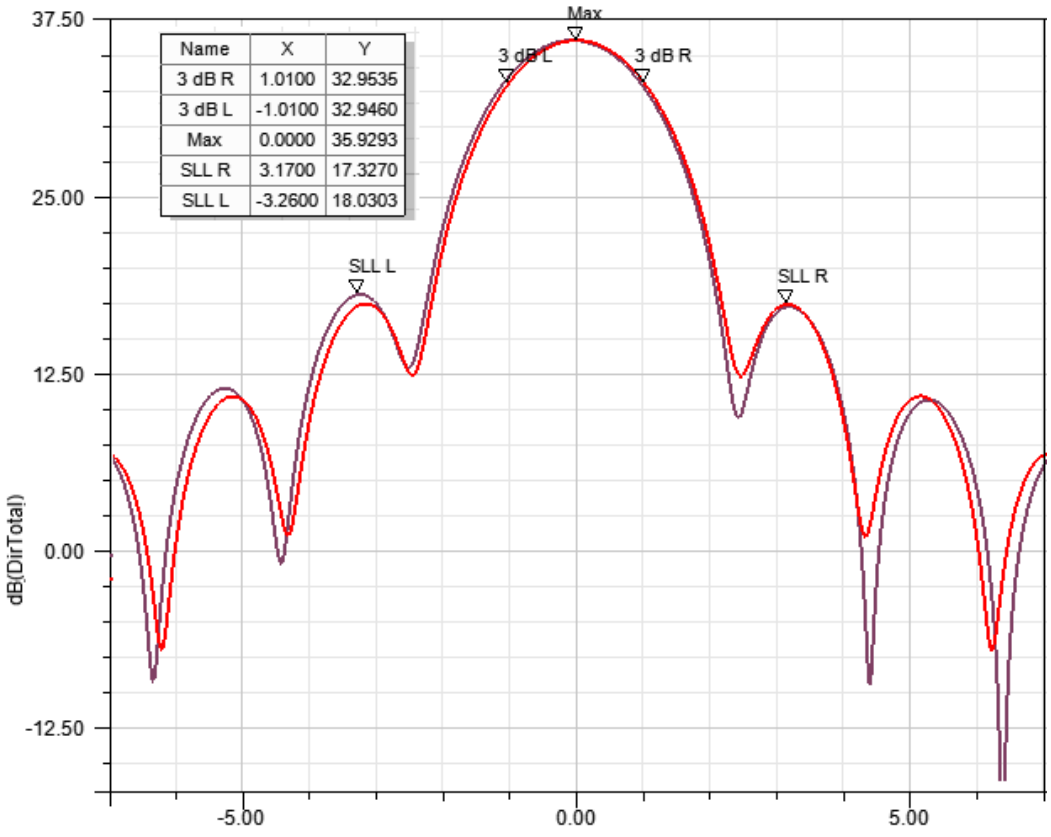


Figure 34 H & E plane rectangular plot

The results are in line with the predicted and exceed expected values. Peak directivity is 35.9 dB, a 2.2 dB improvement over the value predicted by [8], HPBW is in agreement at 2.02°. SLL is 3 dB down from that predicted ***(33)****.*

4.3.5 Horn Feed Construction

As Ihave now confirmed the basic design of the dish and standard gain horn performance, the horn feed will now be assembled as shown in Figure 24.

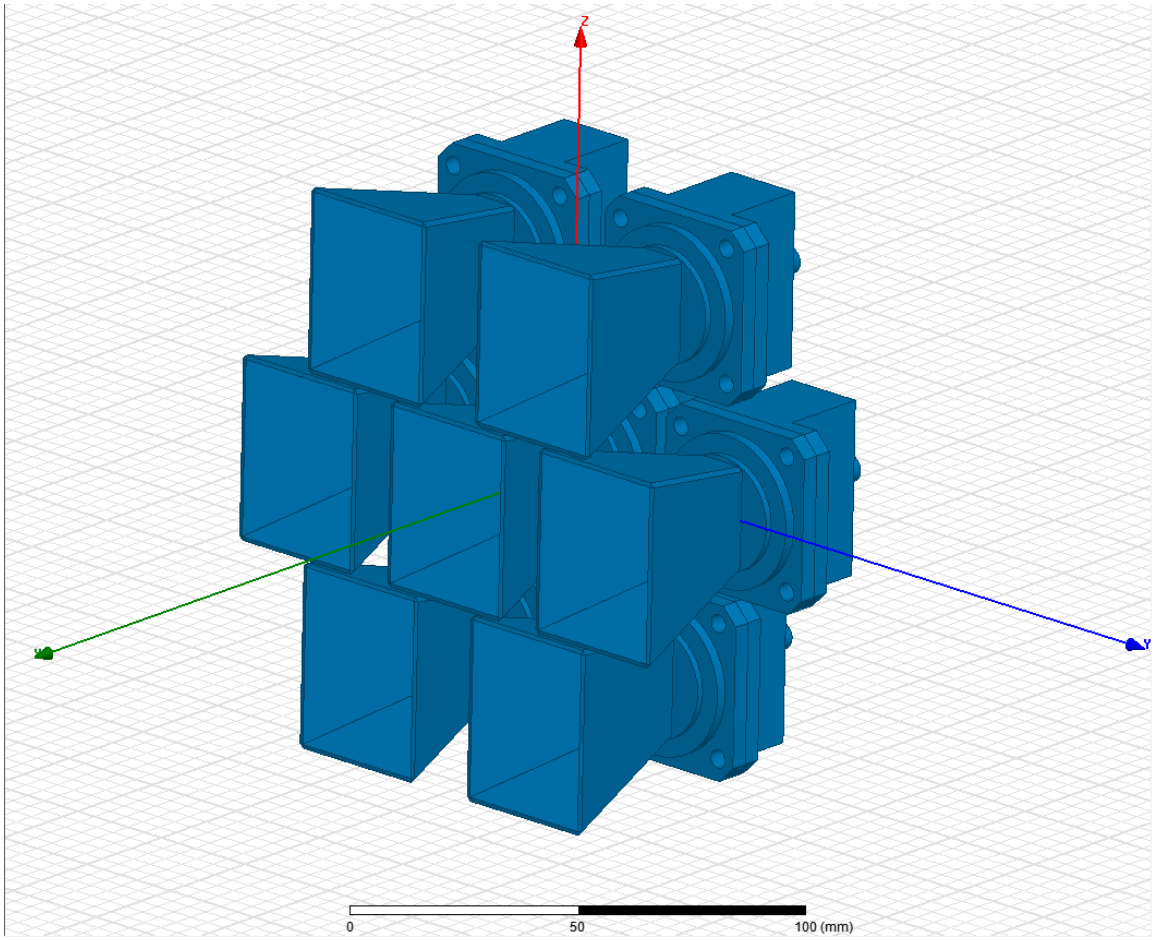


Figure 35 Assembled horn feed

The horn feed is already aligned in the direction of Ω towards the antenna. The origin point of the Cartesian coordinates system on this Figure corresponds with the point O of Figure 17. The X axis of Figure 36 is parallel to Ω. The Z axis of this Figure is perpendicular to the YZ plane of Figure 17. The close proximity of the horn antenna is likely to introduce mutual interference. Further simulation will be required to evaluate this detrimental effect.

In order to evaluate the effect of mutual interference, each horn antenna will be simulated separately (i.e in absence of the presence of the other horn) then the same antenna will be simulated as part of the feed. The results are shown below:

	HPBW		Directivity (dB)		SLL (dB)		Beam Squint	
Posn	Horn	Feed	Horn	Feed	Horn	Feed	Horn	Feed
0	2.05°	2.00°	35.9	35.9	17.6	18.7	N/A	N/A
30	2.02°	2.06°	35.6	35.5	17.3	16.9	2.44°	2.43°
90	2.05°	2.00°	35.8	35.4	17.8	18.4	2.15°	2.14°
150	2.06°	2.07°	35.9	35.6	16.7	17.2	2.42°	2.43°
210	2.06°	2.06°	35.7	35.5	16.4	16.5	2.39°	2.38°
270	2.06°	2.06°	35.4	35.4	17.1	17.3	2.02°	2.05°
330	2.04°	2.03°	35.5	35.4	17.0	16.2	2.38°	2.38°
Avg change	-0.01°		0.2		0.2		0.02°	
Avg change %	-0.5%		-0.4%		1.1%		0.1%	

Figure 36 Mutual interference Measurements

Mutual interference effects are minimal since only one horn antenna is used at any given time. The most noticeable effect is the increased SLL by 1.1% or 0.2 dB

4.3.6 Antenna framing

In order to create a simulation as close as possible to an eventual prototype, a frame is required. Iwill use “plywood” 1/2 inch thick. The dish, rather than being created from a perfectly smooth surface, will be comprised of 18 “pie” sections, with some discontinuities between each of them. Finally, all material will be assigned as a perfect conductor in order to maintain computational requirements within available resources.

This simulation is as follows. Using a HFSS design, a FEBI boundary is defined around the feed and the reminder of the structure (including the dish) will be assigned as an IE region. This will be the 3SBOR prototype that will be used for Matlab simulations.

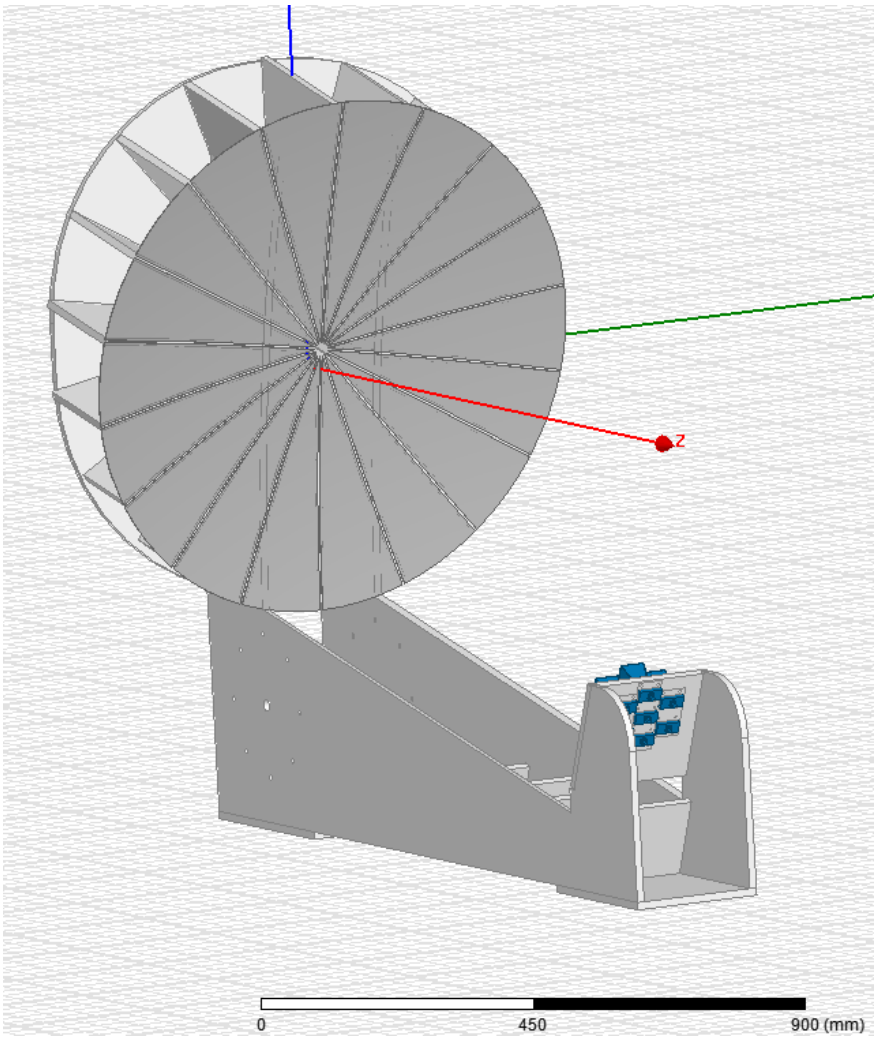


Figure 37 Mutual interference Measurements

Design example for 3SBOR. This structure was added to provide more realistic results to the simulation. In addition, this structure was provided to QETE with the goal to build a prototype. A simplified one dimension 3SBOR prototype was agreed upon and will be discussed in Section 6.

4.3.7 Summary of HFSS simulations results.

The theory behind an offset multi beam reflector design is well documented. Hence the success of this antenna is expected. Excel files providing directivity data with a precision of 0.1° for ±15° in azimuth and longitude were obtained by simulating Figure 38 for each horn antenna. Below is a top view of the 3D polar plot of the seven beams.

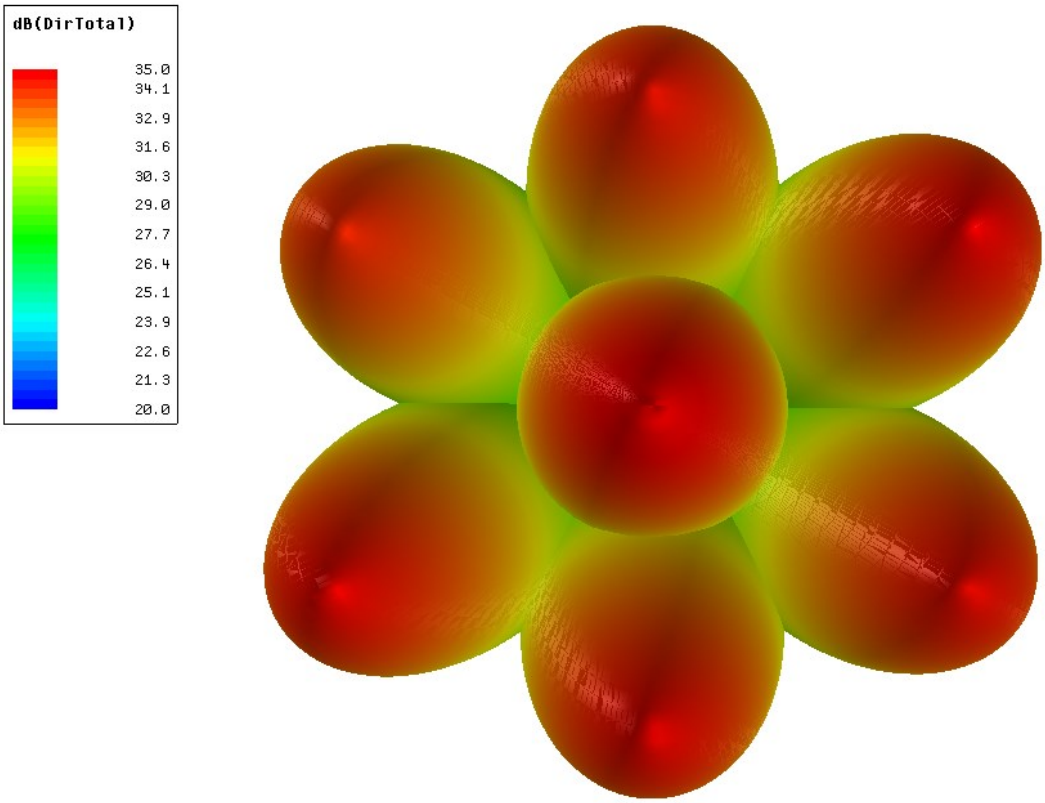


Figure 38 All beams polar plot

This diagram illustrates the coverage provided by the 3SBOR.

Three plane cuts were made to measure the maximum directivity, intersection points and HPBW. The first cut is at 90º, which intersects the peak directivity of the beams of the horns located at 0º, 90º & 270º. The second cut is at 30º, intersecting the peak directivity of the beams of the horn located at 0º, 30º & 210º. The third and final cut is at 150º, intersecting the peak directivity of the beams of the horn located at 0º, 150º & 330º

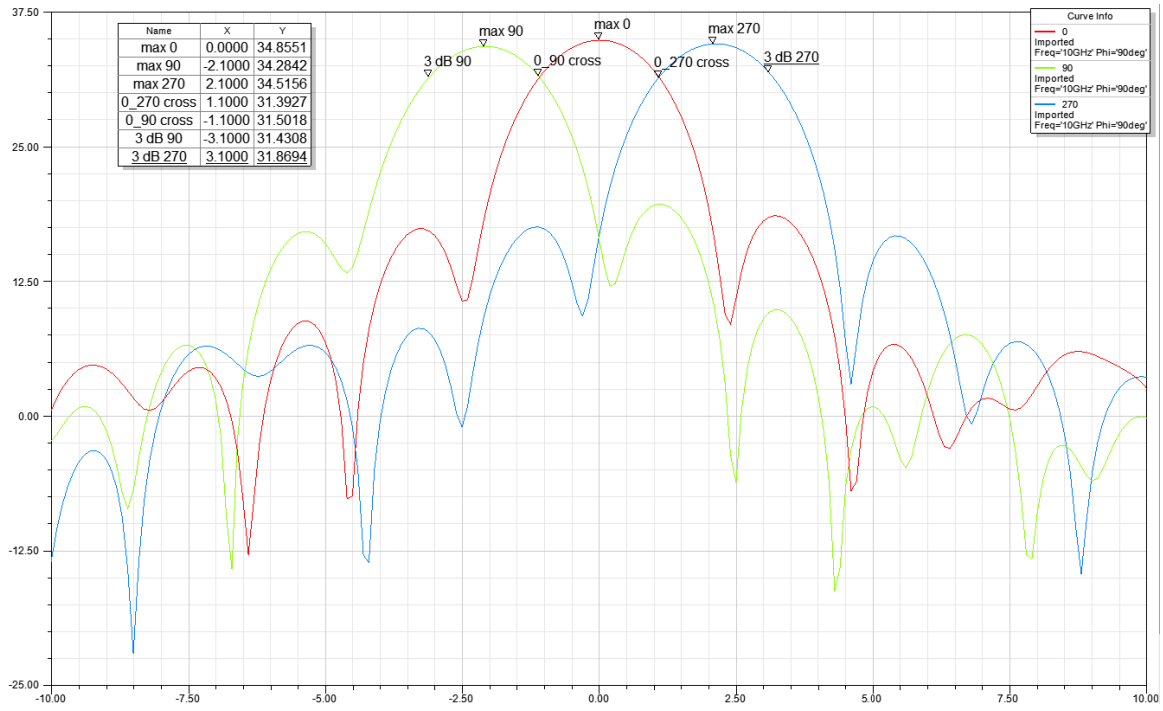


Figure 39 - 90º plane cut

The three beams intersect pretty much at HPBW, guaranteeing a directivity of at least 31 dB for an effective HPBW of 6.2º in that plane cut.

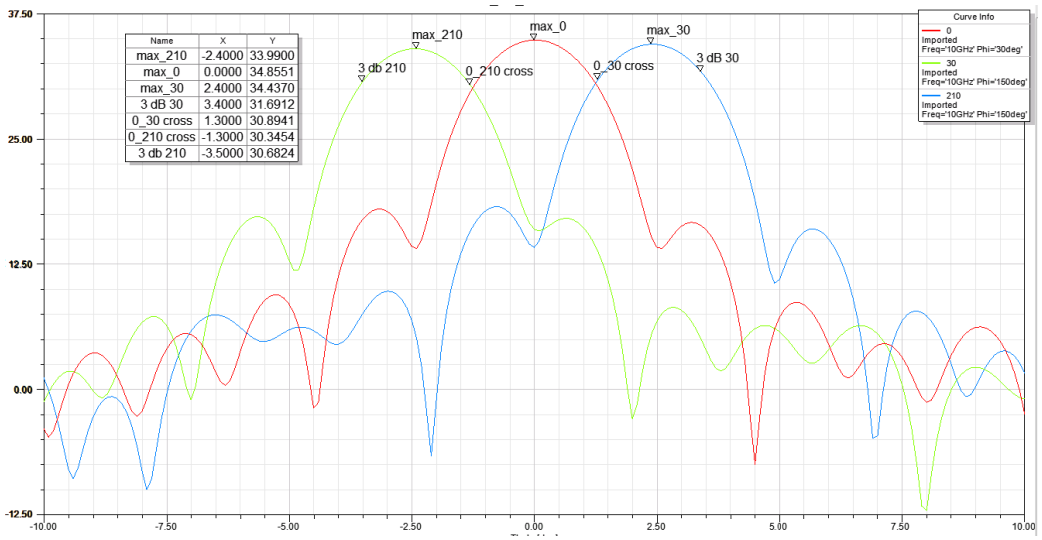


Figure 40 - 30º plane cut

The three beams intersect at 30.3 & 30.9 dB, slightly below HPBW. Each "crossover" is below the HPBW for 0.2º (this will be substracted from the effective HPBW), Ican therefore guarantee a directivity of at least ~31 dB for an effective HPBW of 6.5º in that plane cut.

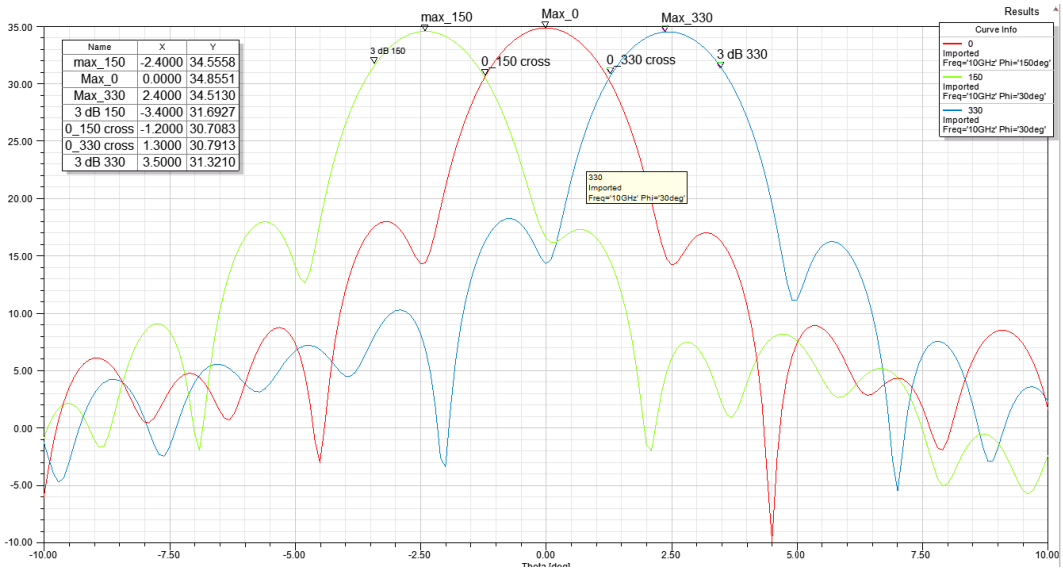


Figure 41 - 30º plane cut

The three beams intersect at 30.7 dB, slightly below HPBW. Each "crossover" is below the HPBW for 0.2º(this will be subtracted from the effective HPBW), Ican therefore guarantee a directivity of at least ~31 dB for an effective HPBW of 6.5º in that plane cut.

4.4 3SBOR Algorithm

The control algorithm's goal is to transfer the noisy pitch & roll angular data (respectively θ & Ω) provided by the sensor into usable angular information for 3SBOR. The output will have a magnitude and an orientation (respectively ρ & Θ). The magnitude will dictate if the main beam is to be used or one of the six side beams while the orientation will select which of the six side beams to use.

4.4.1 Algorithm "backbone"

In Section 3, a modified Alpha Beta filter was selected in an attempt to meet my needs. Alpha-Beta filters are often considered as steady-state Kalman filters. Two major problems of this type of filter is the lack of adaptability to variable noise or mean error and the lag it creates between true position and calculated position.

The 3SBOR algorithm sampling is about 50 times faster than the inherent movement of the vehicle. Therefore, the vehicle's angular speed and angular position can be considered constant for several samples. Consequently, it should be possible to use several position samples to calculate an average angular position and angular speed. As in pulse integration used in Radar, the addition of multiple samples of the noisy data will allow us to separate the true position from the noisy background. The noisier the data sample, the more data samples will be necessary to recover the true position. The following equations will be at the heart of this control algorithm.

Filter equations:

$$\theta_e = \frac{1}{p}\sum_{u=1}^{p}\theta_{t-u} \quad + \frac{\alpha \cdot \Delta t}{s^2} \cdot \sum_{v=1}^{s}(\theta_{t-v} - \theta_{t-v-1}) \tag{56}$$

$$\Omega_e = \frac{1}{p}\sum_{u=1}^{p}\Omega_{t-u} \quad + \frac{\alpha \cdot \Delta t}{s^2} \cdot \sum_{v=1}^{s}(\Omega_{t-v} - \Omega_{t-v-1}) \tag{57}$$

The variables in **(56)** and **(57)** are defined as such:

- - θ_e& Ω_e are the estimated value that will be used to calculate ρ_e & Θ_e using the mathematical model presented in Section 2.
- α is a constant which like the Alpha-Beta filter attributes a weight to the average angular speed used to calculate the position. This allows a synchronization of the calculated position with the measured position.
- Δt is the sampling interval, which in this case is 0.01 second.
- p is the variable number of samples used to calculate the average angular position.
- s is the variable number of samples used to calculate the average angular speed.

The right hand term of **(56)** & **(57)** may be simplified to the simple difference of the first and last term of the summation in a noiseless environment. The assumption here is that like the position, calculating the average speed using multiple noisy measurements will provide greater accuracy than using two samples spaced over the more or less longer period of time.

4.4.2 Sampling size

As it may now become evident, implementing this algorithm will require a series of memory indexes that will be sequentially updated with the most recent angular data and

with the oldest data discarded. The index sizes are controlled by the s & p variables which given the current system, should range from 2 to 15 (greater sampling is detrimental to performance beyond that as it flattens out the measured data, resulting in large measurement error for this simulation). The one reason to increase the sampling size is to reduce the amount of noise in the output. Therefore, Ishould calculate the average and standard deviation of one of the output variable and use it as a criterion for increasing the sample size. Given Θ isn't continuous; the only choice for this approach is ρ. The following equation will be my criteria for increasing the sample size:

Sample size & rejection ratio increase criteria:

$$If\ \rho_e > \frac{1}{m}\cdot\sum_{u=1}^{m}\rho_{t-u}\ + r\cdot\sqrt{\frac{1}{m}\cdot\sum_{u=1}^{m}\left(\rho_{t-u}-\frac{1}{m}\cdot\sum_{u=1}^{m}\rho_{t-u}\right)^2}\ then\ \begin{matrix} s\to s+1 \\ p\to p+1 \\ r\to r+ri \end{matrix} \quad \textbf{(58)}$$

The variables in 58 are defined as such:

- *m* is the constant sampling size of ρ_e
- *r* is the initial rejection ratio (i.e for a Gaussian distribution a *r* value of 3 would represent a rejection of 0.1 % of the data)
- *ri* is the rate of increase of the rejection ratio.

One particularity of **(58)** is that as the sampling size is increased, a greater variance is required to further increase the sampling size. In this case, Ido not attribute a specific probability distribution to the system; Iassume that the variance is variable and therefore the criterion for adjusting the sampling size is variable as well. However, **(58)** introduces some problems; how is the sampling size and rejection ratio reduced in the event of noise level reduction? Increasing the sampling size is beneficial in high noise situation but

detrimental in low noise situation. Therefore, sampling size and rejection ratio should decay over a certain period of time.

Sample size & rejection ratio decrease criteria:

$$After\ d \cdot \Delta t\ has\ passed \begin{matrix} s \to s-1 \\ p \to p-1 \\ r \to r-ri \end{matrix}$$

(59)

Where *d* is the amount of samples (time) before the sampling and rejection ratio are reduced. Establishing the value of *d* will determine the frequency at which the system will oscillate between sampling size.

It is clear that this algorithm is inherently unstable (no steady state). In order for it to work, minimums and maximum will be required for *s, p* & *r.* The challenge here will be to set parameters that will allow for the algorithm to assure adequate 3SBOR performance in the presence of noise.

4.4.3 Mean error & Large inputs

Given the system oscillates around a resting position, calculating the mean error of θ_e & $\Omega_e$will be as simple as calculating the average of each signal over a long period of time (~1000-2000 samples should do) and then removing this average from the actual sensor measurement output.

Mean roll error:

$$\overline{\theta_t} = \frac{1}{z}\sum_{u=1}^{z} \theta_{t-u}$$

(60)

Mean pitch error:

$$\overline{\Omega_t} = \frac{1}{z}\sum_{u=1}^{z} \Omega_{t-u}$$

(61)

By including **(60)** & **(61)** into **(56)** & **(57)** the filter equations with mean error adjustments become:

$$\theta_e = \frac{1}{p}\sum_{u=1}^{p}(\theta_{t-u} - \overline{\theta_t}) \quad + \frac{\alpha \cdot \Delta t}{s^2} \cdot \sum_{v=1}^{s}(\theta_{t-v} - \theta_{t-v-1})$$

(62)

$$\Omega_e = \frac{1}{p}\sum_{u=1}^{p}(\Omega_{t-u} - \overline{\Omega_t}) \quad + \frac{\alpha \cdot \Delta t}{s^2} \cdot \sum_{v=1}^{s}(\Omega_{t-v} - \Omega_{t-v-1})$$

(63)

However these mean error corrections must guard against large inputs; such input (like a pothole) would introduce a measurement error for which 3SBOR would eventually recover over a certain period of time but would certainly hinder overall system performance. In order to "reject" these large inputs, a lockout criteria is required.

Lockout criteria:

$$If\ \rho_e > L^o\ then \begin{matrix} \rho_t = \rho_{t-1} \\ \theta_t = \theta_{t-1} \\ \Omega_t = \Omega_{t-1} \end{matrix}$$

(64)

Therefore, whenever ρ_e is greater than the reject angle, all current data is replaced by the previous data.

4.5 Implementation in Matlab

The complete Matlab code is available in the Appendix A. The simulation is set as a loop for which the sampling is 0.01 second. The following functional diagram will be discussed in details in the following subsections.

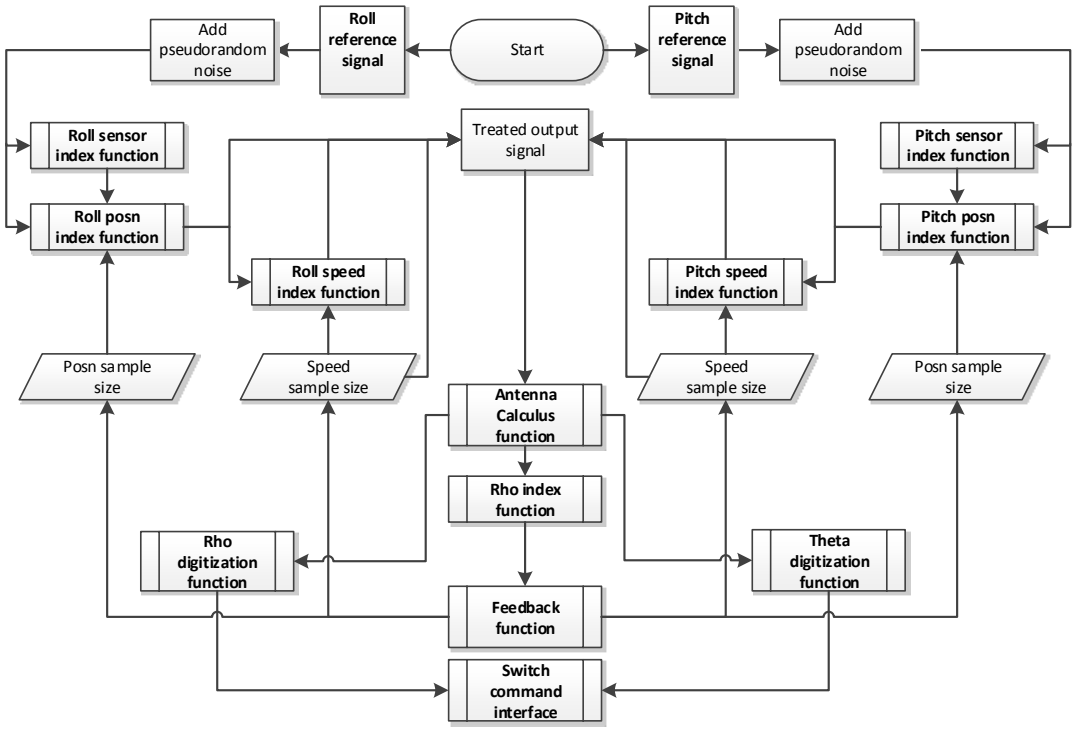


Figure 42 – Block diagram, Matlab simulation – top level

This diagram shows the key relations between the various operations used in the Matlab simulation. There are a total of six operations, which will all be discussed below. The roll and pitch reference signals represent the true position of the antenna in space (θ,Ω). This "true" position will be used later to evaluate directivity. While a beam will be selected using treated/noisy angular data, the actual directivity will be measured with regards to the "true" position.

4.5.1 Reference signals.

Figure 42 includes two reference signals for pitch and roll. These are defined below in the simulation:

Roll reference signal:

$$\theta_R(t) = 0.9 \cdot \left(\cos\left(\frac{\pi}{4}t\right) - \sin\left(\frac{\pi}{2}t\right) + \cos(\pi t) - \sin(3\pi t) + \cos 5\pi t\right) \quad \textbf{(65)}$$

Pitch reference signal:

$$\Omega_R(t) = 0.9 \cdot \left(\sin\left(\frac{\pi}{4}t\right) + \cos\left(\frac{\pi}{2}t\right) - \sin(\pi t) + \cos(3\pi t) - \sin 5\pi t\ \right) \quad \textbf{(66)}$$

Although a simple sine wave with a frequency of 1-2 hertz would have sufficed to demonstrate the system, additional slower & faster harmonics were added to simulate a faulty vehicle suspension. The 0.9 multiplier is an adjustment in amplitude that allows for optimal comparison between the 3SBOR and a single beam reflector (This value was obtained using comparative assessments in Section 5).

Two time-offset square waves with amplitude of 5º were respectively added to **(65)** & **(66)** to create impulses that occur every 100 seconds and last for about 2 seconds. These simulate potholes that the system must be able to recover from quickly.

4.5.2 Added noise

In this case, the randn function from Matlab is used to add pseudorandom noise to the reference signal, giving the measured sensor output:

Measured sensor output:

$$\theta_M(t) = \theta_R(t) + N \cdot randn + \overline{\theta} \quad \textbf{(67)}$$

$$\Omega_M(t) = \Omega_R(t) + N \cdot randn + \overline{\Omega} \quad \textbf{(68)}$$

Where N is the noise multiplier, $\overline{\theta}$ & $\overline{\Omega}$ are unknown mean measurement errors.

Throughout the simulation the level of noise may vary as:

- -N=0.1 would simulate driving on a freshly, smooth paved road.
- -N=1 would simulate driving on a gravel road.
- -N=2 would simulate driving off-road.

As the noise may be constant or not (no data was found with regards to antenna movement on a moving vehicle), this multiplier will allow us to evaluate the improvement in terms of directivity provided by the increase in sampling. Additionally, it is possible to use this particularity to find out the optimal sampling size range.

4.5.3 Roll/pitch sensor index function

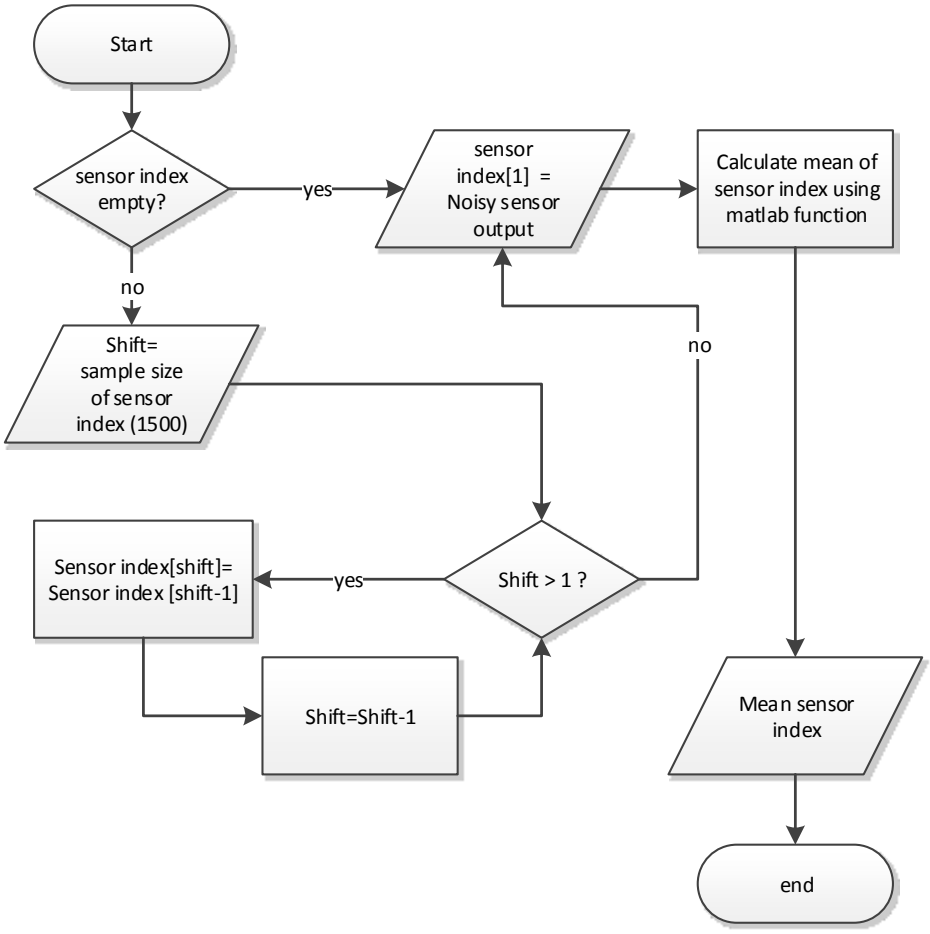


Figure 43 – Block diagram, Roll/Pitch Sensor Index Function

This index (it could referred to as a single row matrix with 1500 columns) continuously updates itself with the most recent information in the first column and the oldest in the last column. Data older than 15 seconds is discarded.

The Sensor Index function aims to calculate the mean error of the sensor in both measured angles. This is possible given the periodic nature of the antenna's movement. In order to do so, ***(60)*** & ***(61)*** now become the following:

Roll sensor index function:

$$\overline{\theta_M} = \frac{1}{1500}\sum_{u=1}^{1500}\theta_{M-u} \quad \textbf{(69)}$$

Pitch sensor index function:

$$\overline{\Omega_M} = \frac{1}{1500}\sum_{u=1}^{1500}\Omega_{M-u} \quad \textbf{(70)}$$

An average of 15 seconds is more than sufficient to calculate the mean error of the measured signal. This methodology however results in a start-up time of 15 seconds for 3SBOR. A wait loop was integrated in the Matlab program to accommodate this.

This function is key to the 3SBOR, as it will allow placing the sensor parallel to the pitch/roll axis plane (ref Figure 2), thus enabling accurate angular measurements. As per Section 2, my current φ is 15º. Therefore, the expected value of **(70)** during simulation should be identical (or very close).

4.5.4 Roll/pitch position &speed index functions

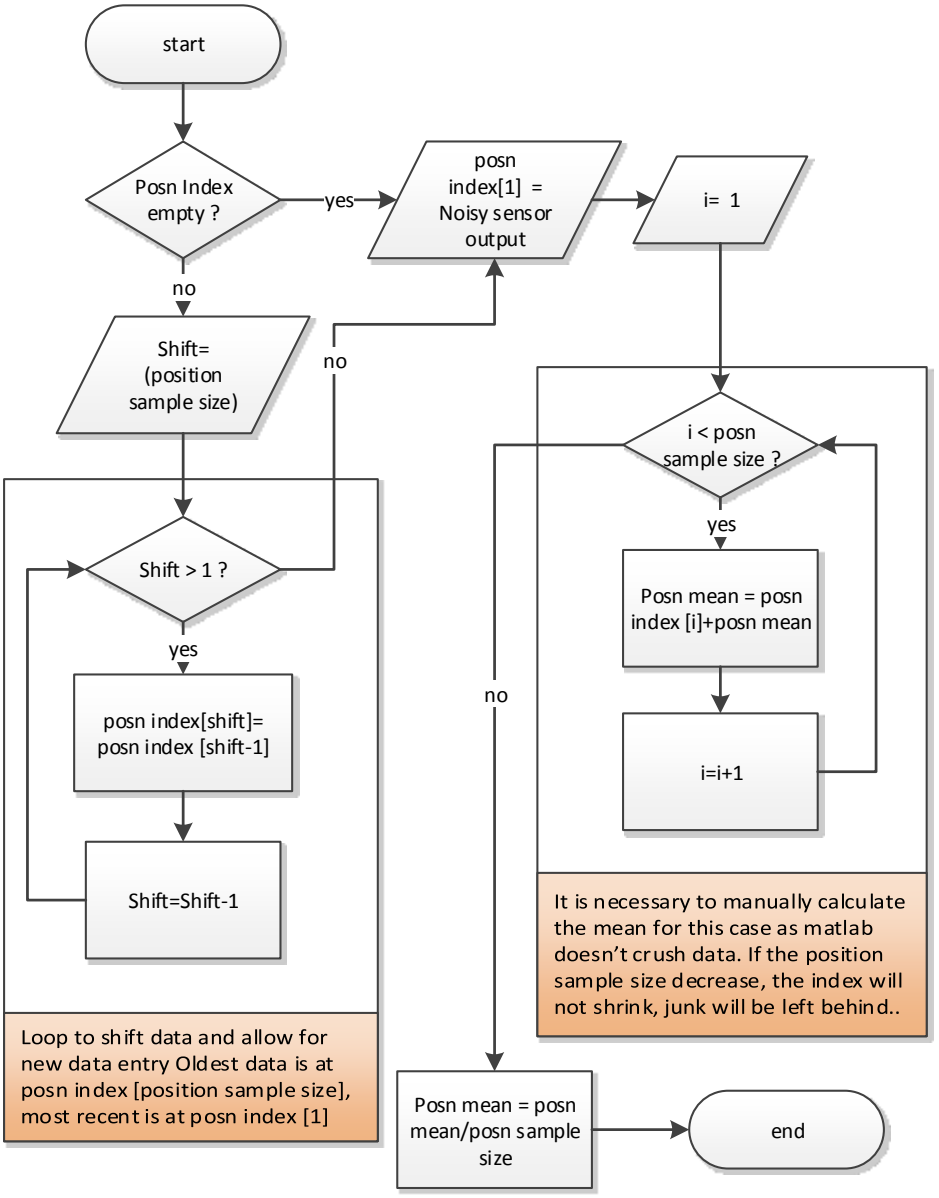


Figure 44 – Block diagram, Roll/Pitch position index function

This memory index has the particularity of being variable in size.

This is the second of three functions required to calculate equations **(62)** & **(63)**. This function provides data for the left hand terms of these equations. Throughout the simulation, the range of sampling size for the angular position was established from a minimum of three to a maximum of eight. A methodology to select an optimal sample size for an angular position approximation is left for future work.

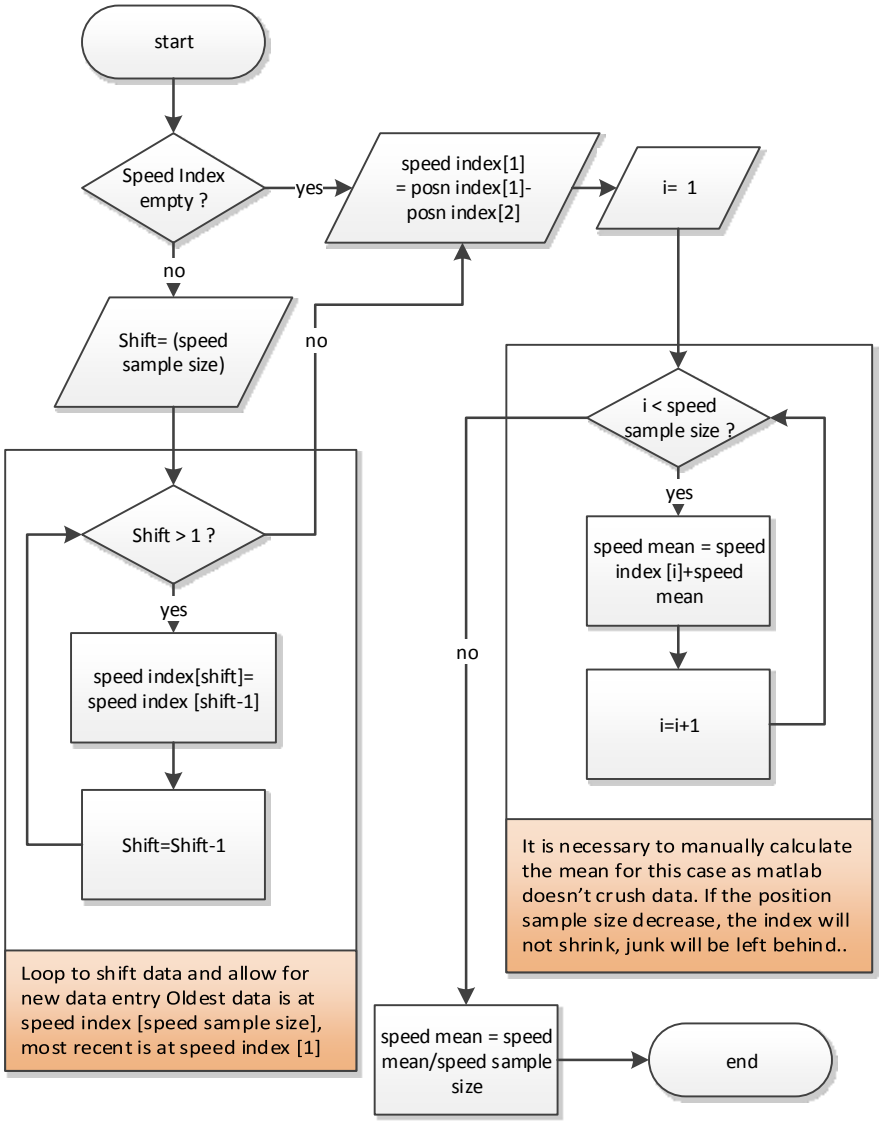


Figure 45 – Block diagram, Roll/Pitch speed index function

This function is very much like the position with the difference of calculating the average difference between samples. Note that the average isn't divided by the sampling time

(which would return a speed…); it is a simplification of ***((62)*** *&* ***(63)*** *which will lose their* Δt *term in the next subsection.*

This is the last of three functions required to calculate equations **(62)** & **(63)**. This function provides data for the right hand terms of these equations. Throughout the simulation, the range of sampling size for the angular speed was established from a minimum of four to a maximum of eleven. A methodology to select an optimal sample size for angular speed approximation is left for future work.

4.5.5 Treated output signal

The parameter α was attributed a value of 0.75. Using three index functions mentioned above, Ican now rewrite **(62)** & **(63)** as such:

Matlab filter equations:

$$\theta_T = \frac{1}{p}\sum_{u=1}^{p}(\theta_{M-u} - \overline{\theta_M}) \quad + \frac{0.75}{s^2}\cdot\sum_{v=1}^{s}(\theta_{M-v} - \theta_{M-v-1}) \quad p = [3,8] \quad s = [4,11] \tag{71}$$

$$\Omega_T = \frac{1}{p}\sum_{u=1}^{p}(\Omega_{M-u} - \overline{\Omega_M}) \quad + \frac{0.75}{s^2}\cdot\sum_{v=1}^{s}(\Omega_{M-v} - \Omega_{M-v-1}) \quad p = [3,8] \quad s = [4,11] \tag{72}$$

Where the notation [x,y] represents an inclusive interval of sample size. The feedback function will output the required sample sizes. As a default start value the sample sizes are set at 3 for position and 4 for speed.

Optimizing the value of α is left for future work. During comparative assessment, the value of 0.75 offered better performance that none, hence the alpha term.

4.5.6 Antenna Calculus

At this point Itake the model from Section 2 and Icalculate three pair of signals

- ρ_T & Θ_T which are calculated using θ_T & Ω_T. (treated signal). This will be used to simulate the overall performance of the system.
- ρ_M & Θ_M which are calculated using θ_M & Ω_M. (untreated, noisy signal) This will be used to evaluate the performance/noise improvement provided by the algorithm.
- ρ_R & Θ_R which are calculated using θ_R & Ω_R (reference position). This will be used to compare overall performance of system with noise free scenario and single beam antenna.

Below is the vehicle parameters used for the simulation

Vehicle parameters:

$$ha = 2.8\,m \quad \varphi = \frac{\pi}{12}\,rad \quad rz = 1.75\,m \quad pz = 1\,m \tag{73}$$

Refer to S ection 2 for detailed Antenna movement analysis and calculus.

4.5.7 Rho index function`

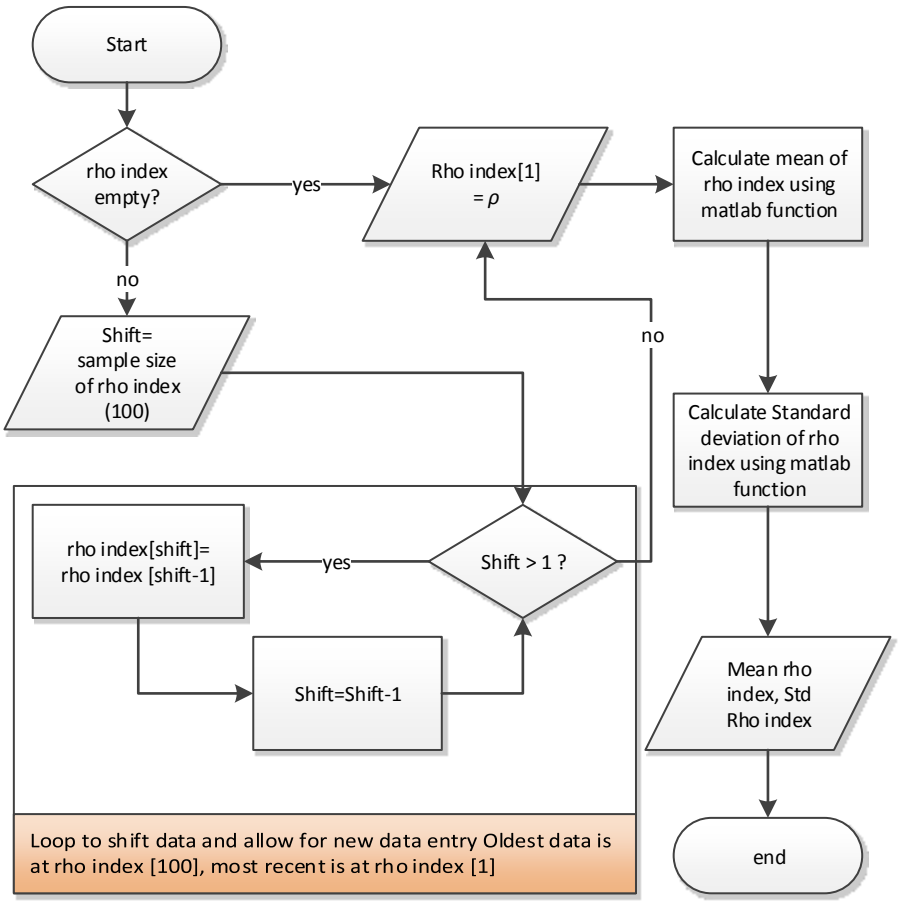


Figure 46 – Block diagram, Rho index function

This function is identical to the sensor index function with the only difference that it monitors only 1 second (i.e 100 samples) of ρ_T.

This function will allow us to run the Feedback Function (section below) which in turn will allow us vary the sample size as a function of the noise level. ρ_T. The size of this index function was determined by comparative assessment during simulation. Optimization of this parameter is left for future work.

4.5.8 Feedback Function

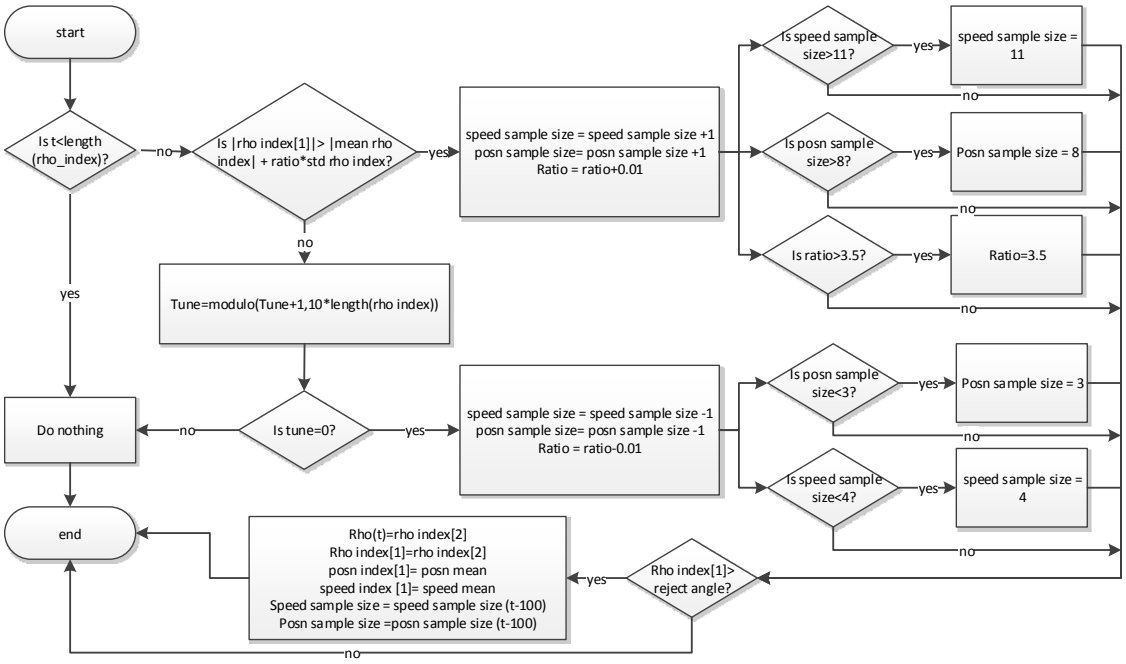


Figure 47 – Block diagram, Feedback function

This function is allows us to increase the size of samples for position and speed measurement in function of noise level of ρ_T

This function sets parameters for **(58) (59)** & (**64**). First, the condition for increasing the sample size is set as such:

Sampling &rejection ratio increase criterion:

$$If\ \rho_T > \frac{1}{100}\cdot\sum_{u=1}^{99}\rho_{T-u}\ + r\cdot\sqrt{\frac{1}{100}\cdot\sum_{u=1}^{100}\left(\rho_{T-u}-\frac{1}{100}\cdot\sum_{u=1}^{100}\rho_{T-u}\right)^2}\ \rightarrow\ \begin{matrix} s\rightarrow s+1 \\ p\rightarrow p+1 \\ r\rightarrow r+0.01 \end{matrix} \quad \textbf{(74)}$$

Note that the rate of increase of the rejection ratio was determined during comparative assessment. This rate allows for repeatable and reliable performance. A ratio increase that is too fast would prevent maxing out the sampling size while too low a rate of increase

would translate into a sampling increase that's too fast. Optimizing the rate increase is left for future work.

Another key part of this function is its decay rate. This is simply a timer (which is only active when the condition **(74)** is false) for which a given count results in a decrease in sample size. In this case the time was set at a total of 10 seconds. Therefore, for every 10 cumulative seconds (not necessarily consecutive), the following is applied:

Decay rate:

$$After\ 1000 \cdot \Delta t\ has\ passed \quad \begin{matrix} s \rightarrow s - 1 \\ p \rightarrow p - 1 \\ r \rightarrow r - 0.01 \end{matrix} \tag{75}$$

This rate was determined during comparative assessment. The optimization of this parameter is left for future work. As this point the performance provided by the current parameters is more than adequate as will be demonstrated in Section 5.

With regards to the lockout criterion, multiple modifications were required to prevent the sampling size to "run off" with each pothole.

Lockout criterion:

$$If\ \rho_T > 6^{o}\ then \quad \begin{matrix} \rho_T = \rho_{T-1} \\ \theta_T = \overline{\theta_{T-1}} \\ \Omega_T = \overline{\Omega_{T-1}} \\ s_T = s_{T-100} \\ p_T = p_{T-100} \\ r = 3 \end{matrix} \tag{76}$$

In addition to replacing the new sample of θ_T & Ω_T with their past average, (replacing them with their respective second last sample caused the algorithm to "flatline"), it was necessary to monitor the sample size and lookback to 1 second prior to ensure the sample

size didn't "run off". The rejection ratio is simply reset to its start value during lockout. (A potential improvement would be to monitor the rejection ratio as for the sample sizes)

4.5.9 Rho Digitization Function

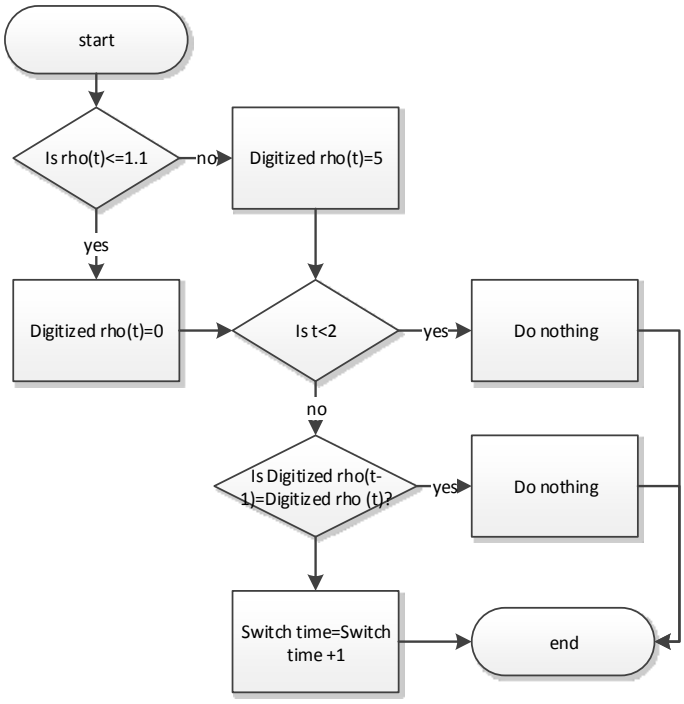


Figure 48 – Block diagram, ρ_T digitization function

The ρ_T "digitization function takes the 100 Hz sampled ρ_T signal and transforms it into a binary signal i.e (0 or 1)

For this specific system, the decision point is 1.1º which is the crossover point between the main beam on boresight and the four beams located respectively at 30º, 150º, 210º & 330º. As per Figure 49, any value of ρ_T that is smaller than or equal to 1.1º returns a value of 0, meaning the centre beam is used and the value of Θ_T is discarded. If the value is greater than 1.1º, the main beam won't be selected and Θ_T will be used to select one the six other beams. This function is used as well to digitize ρ_M & ρ_R for comparative assessment purposes.

4.5.10 Theta Digitization Function

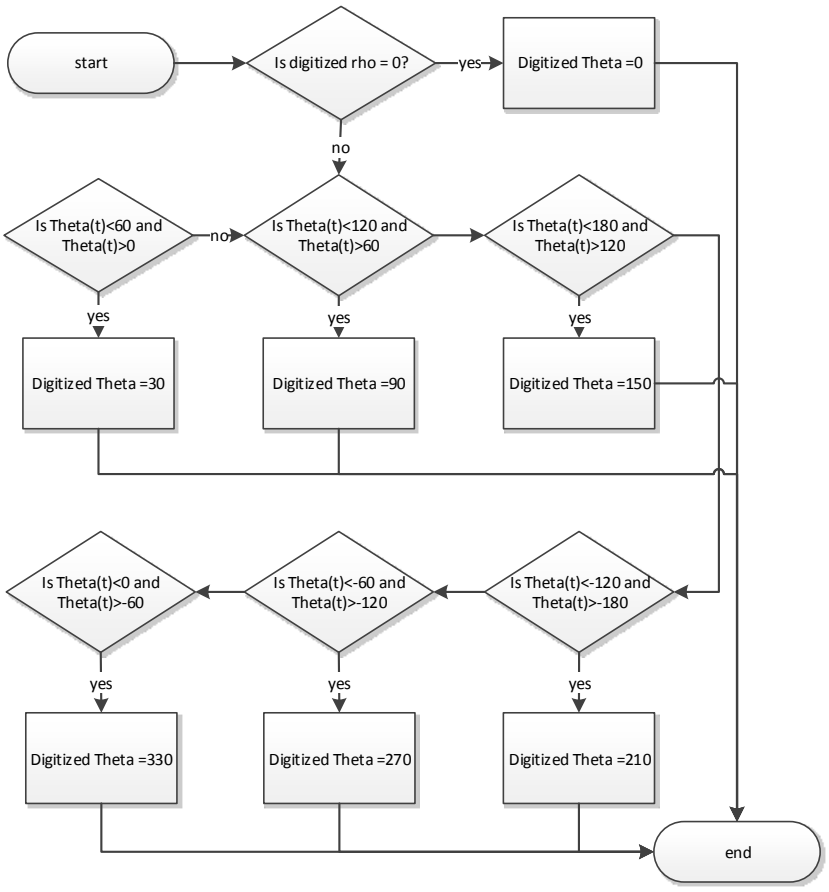


Figure 49 – Block diagram, digitization function

The ρ_T "digitization function takes the 100 Hz sampled Θ_T signal and transforms it into six states signal. In pair with ρ_T digitization function, it will be possible to create a 3 bit signal required to operate the RF switch selected in section 3.

It is clear that the coverage around the center beam isn't even. Therefore Iseparate evenly the decision point in 60º portions, with the center of each portion intersecting with the peak directivity of the selected beams. The results provided are more than acceptable. Optimization of this function is left for future work. This function is used as well to digitize Θ_M & Θ_R for comparative assessment purposes.

5 COMPARATIVE ASSESMENTS

In order to evaluate the performance of the 3SBOR, numerous simulations were required. Two levels of simulations were performed. The low level consisted of evaluating the directivity of 3SBOR over time. The top level simulation ran the low level simulations in a loop while varying a specific variable, thus allowing for testing the limits & performance of the system.

5.1 Low level simulations

The first part of the low level simulations was to monitor the reference position signal θ_R& Ω_R. Simulation time was 360 seconds (6 min).

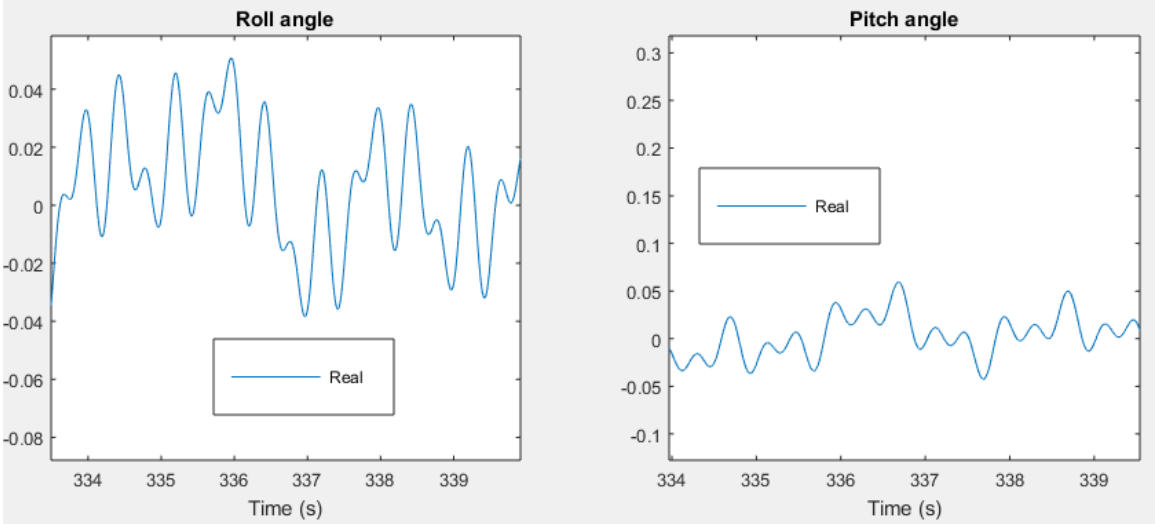


Figure 50 – Zoom inθ_R& Ω_R.

These signals represent the true position of the antenna in space. Note that the vertical axis is in radians. The movement amplitude is about ±4° in roll and pitch directions.

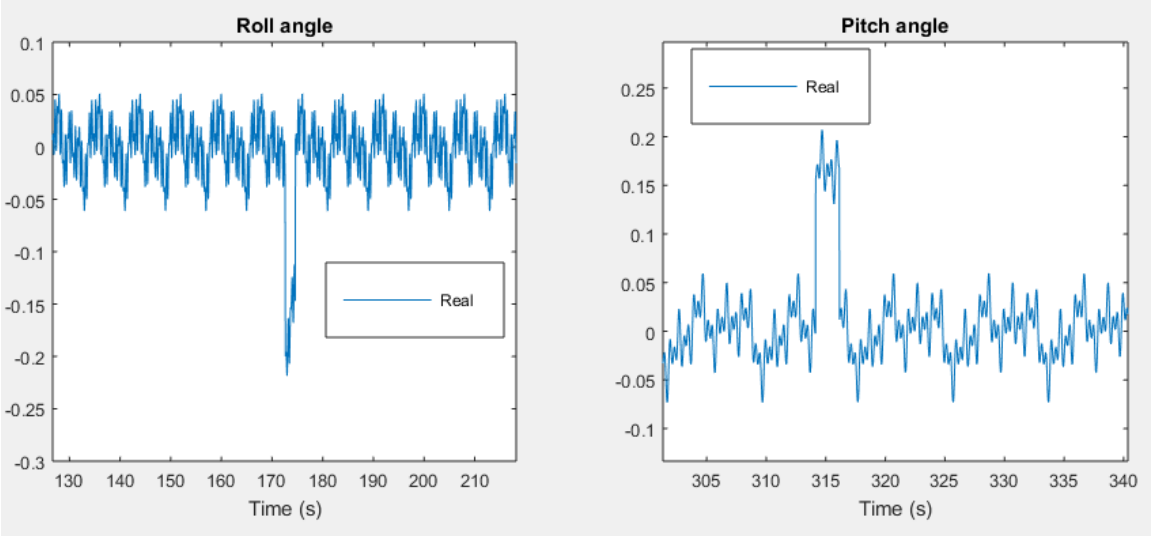


Figure 51, θ_R& Ω_R.with potholes

Here Ican see a broader view including a pothole discontinuity Note that the vertical axis is in radians. The movement amplitude caused by the potholes is ±11° in roll and pitch directions.

This allowed for the calculations of the reference position of the antenna in space (i.eρ_R & Θ_R), which is where the antenna should point at.

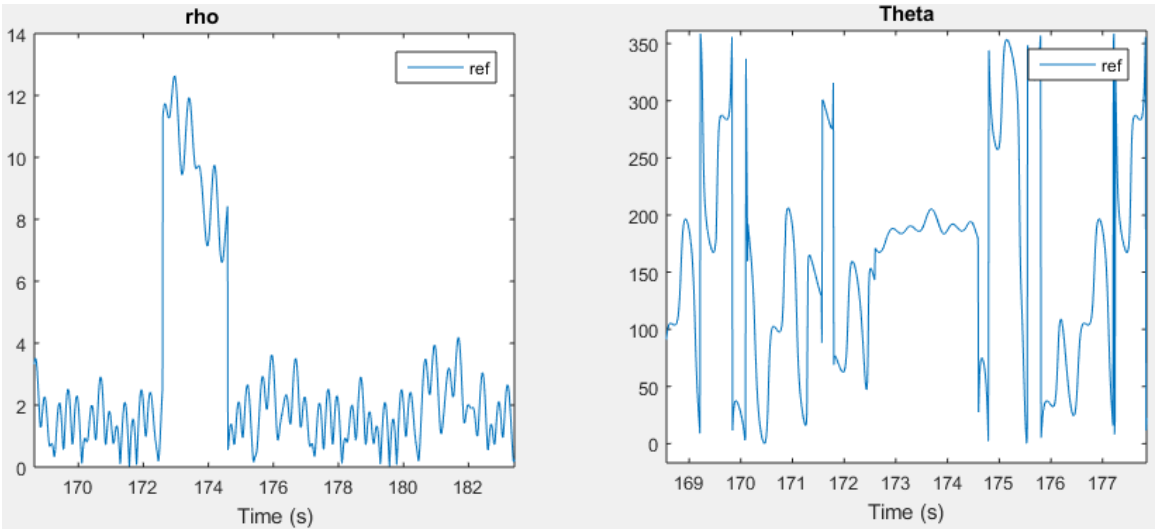


Figure 52, ρ_R& Θ_R.with potholes

The scale is in degrees here. This will be used to measure the performance in noisy conditions.

Subsequently, Ican now digitize $\rho_R$and Θ_R. This will then allow us to read in the directivity files provided by HFSS simulations and produce a directivity graph as a function of time.

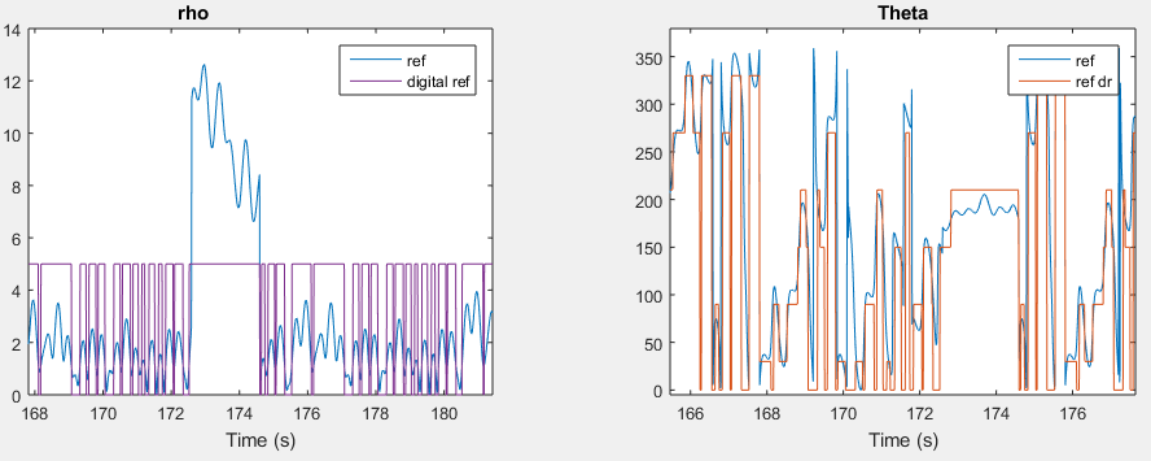


Figure 53, ρ_R& Θ_R. Digitized with potholes

The digitized reference signal will be used to monitor the time the antenna isn't "looking" at the desired location, thus providing an indication of the synchronism of the treated signal with the true position.

From the HFSS results, it is possible to produce two critical results; the directivity as a function of time of the designed antenna with the center beam only (i.e. reference directivity) and the directivity as a function of time of 3SBOR. As a measure of performance, the mean directivity of both graphs was calculated and the difference provides us with the mean improvement in directivity (which does not include the insertion loss of the RF switch proposed in Section 3). In order to evaluate the best case scenario, the potholes have been removed of these results.

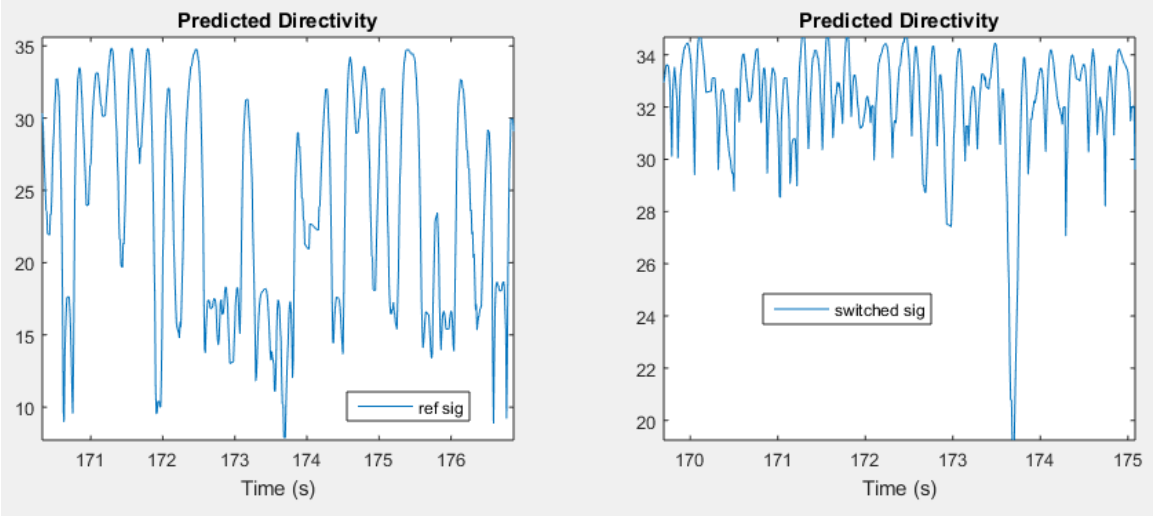


Figure 54, Reference and switched directivity using true position & potholes removed

It can clearly be seen the improvement in directivity between the two results. The mean directivity for the left (reference) is at 24.17 dB (10 dB down from peak!) while the right (switched) averages at 32.30 dB (2 dB down from peak directivity). This provides us with a best case scenario that shows the potential of 3SBOR. Note that these averages discard the first 15 seconds, reducing the simulation time to 345 seconds.

Looking at Figure 54, the achievable directivity improvement is 8.13 dB, in one direction (i.e transmit or receive) for this range of movement. If Iinclude the insertion loss of the RF switch, the maximum drops to about 6 dB, which is still quite considerable. Note that the switch time was accounted for as well for the overall simulation. Given a switch time of 100 ns, the total switching time for a simulation of 345 seconds is 7.765 ms, which is 0.0022 % of the simulation time.

Now that we've presented the ideal scenario, we'll introduce noise, mean measurement errors & the noise reduction algorithm. Then we'll compare these results with the benchmark established here.

5.1.1 Added noise and mean errors

Let's introduce the measured sensor output θ_M& Ω_M which includes mean measurement error and random noise. Note the various levels of noise.

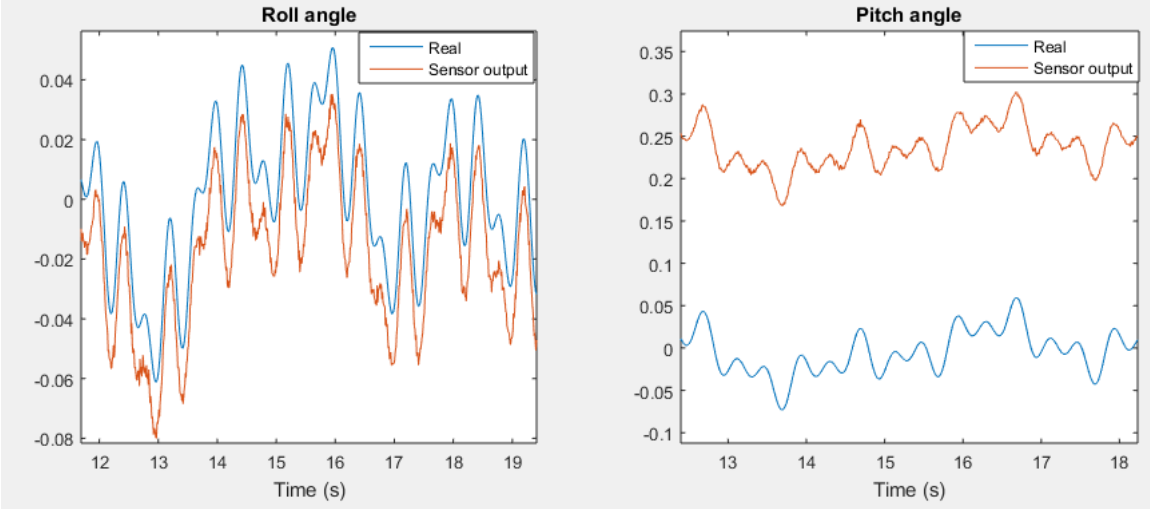


Figure 55 –θ_M& Ω_M.N=0.1

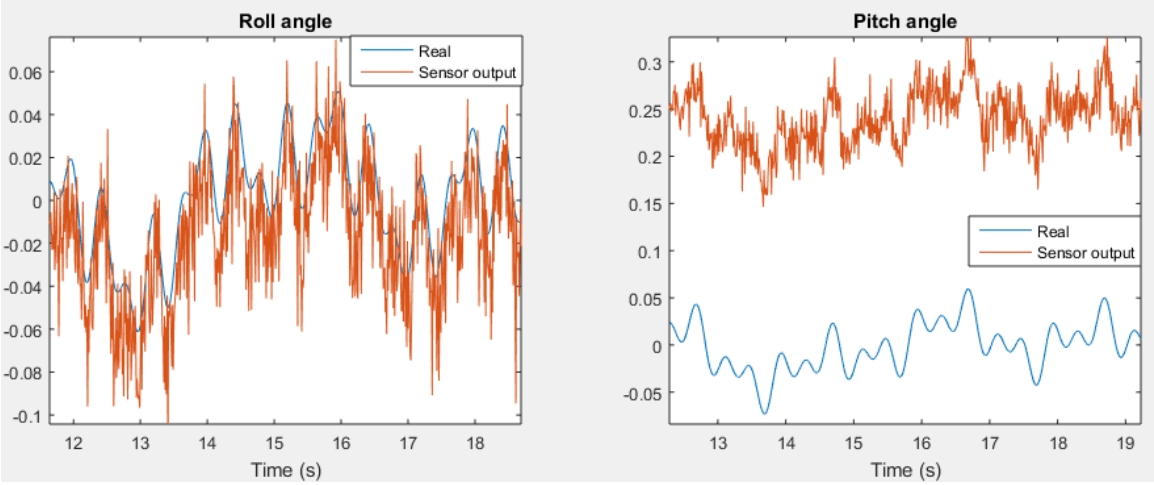


Figure 56 –θ_M& Ω_M.N=1

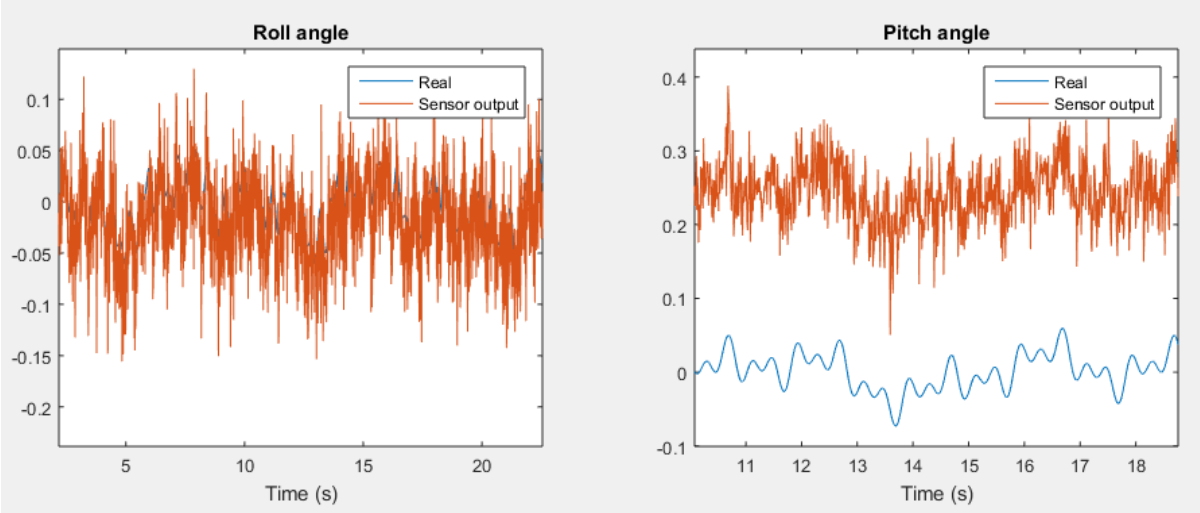


Figure 57 –θ_M& Ω_M.N=2

Notice that the mean error will be constant at all time as it is attributed to an error in position (i.e the sensor isn't perfectly flat). The three Figures above show some examples of noise level. At N=0.1 the sample rate will remain at minimum (i.e p=3, s=4) while at N=2, it will max out (i.e p=8, s=11). N=1 provides a middle point.

Of course it will be possible to vary N during the simulation to evaluate how fast the algorithm can track and adapt to changing conditions.

5.1.2 Treated signal at N=0.1

Iwill look at θ_T& Ω_T during startup and during various operating conditions.

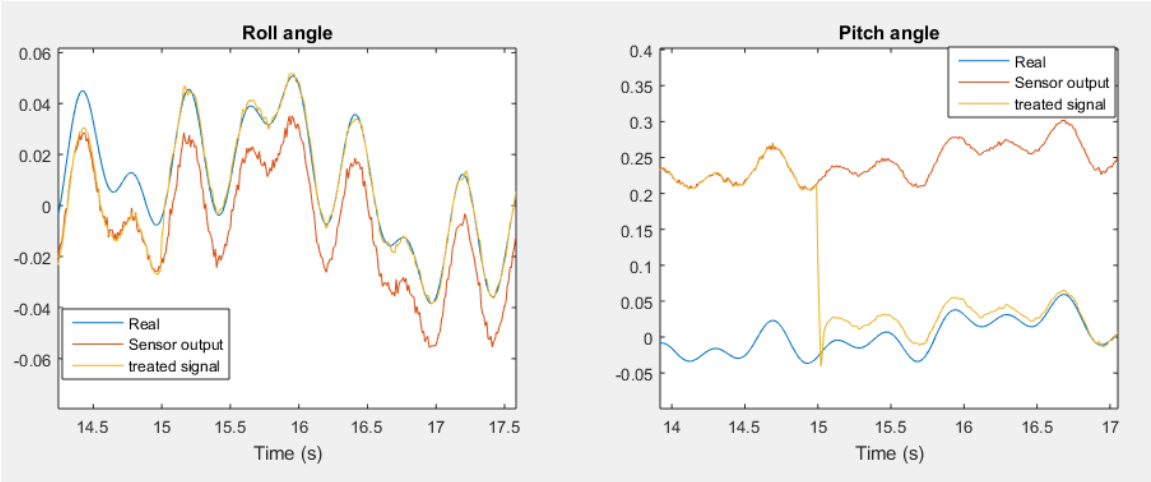


Figure 58 –θ_T& Ω_T.at startup with inclusion of $\overline{\theta_M}$& $\overline{\Omega_M}$ with N=0.1

Here Ican see the inclusion of the mean error correction term in the treated signals at 15 seconds of operation time. Notice the lag of the pitch signal corrects itself after a couple of seconds.

At this point, ρ_T becomes of interest since it provides us with the required data to tune in the sample size.

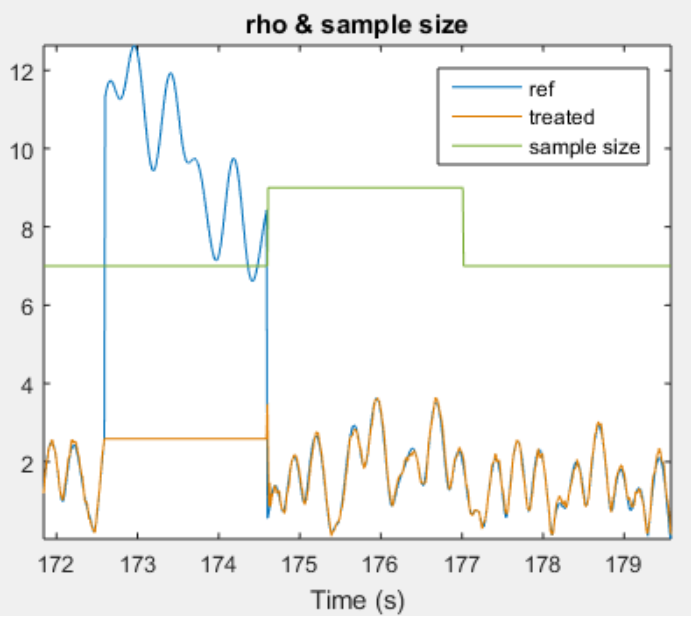


Figure 59 $-\rho_T$, ρ_R and total sample size .at N=0.1

The feedback function is at work in Figure 59. First Ican see ρ_T locking out (treated signal flatline) during the pothole from 172.5 to 174,5 seconds. Notice the total sample size (s+p) does creep up from 7 to 9 but recovers quickly. Another key detail, even if the decay rate is 10 second for the sample sizes, this decay rate is cumulative and calculated regardless of the increase ratio (i.e. an increase might just have occurred and the next loop could cause a decrease, depending on where the decay rate is at.)

Another point to consider is the match of the treated signal to the reference signal. In this case, this "matching" will be evaluated by comparing the two digitized signal. For every sample the treated digital signal isn't equal to the reference signal, this time will be accounted for and added to a total at the end of the simulation. Using this information, an error percentage will be calculated to evaluate the performance of the algorithm.

.

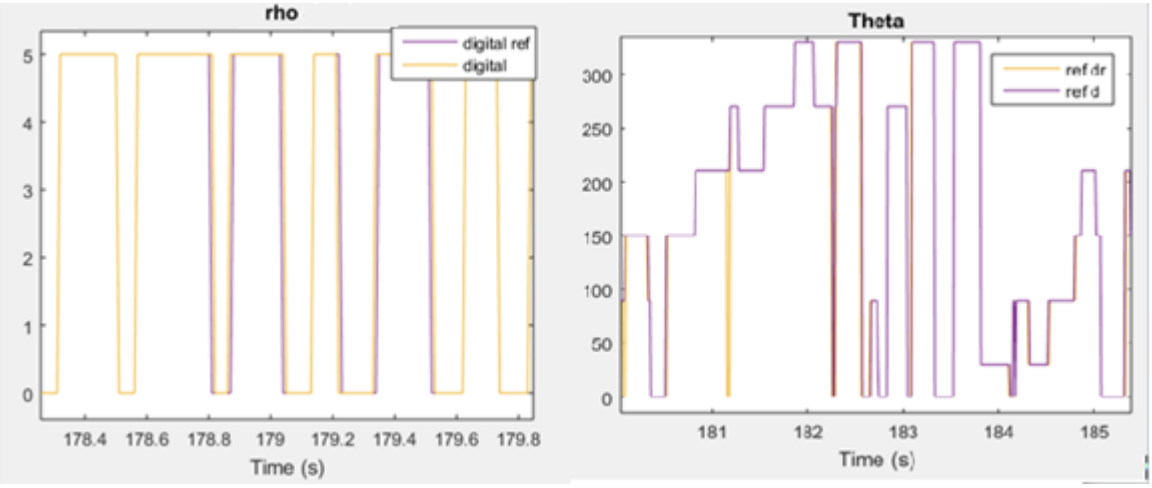


Figure 60 –ρ_T , Θ_T & ρ_R. Θ_R, Digitized at N=0.1

For the total simulation time the total error at N=0.1 stands at 0,53% for ρ_T and 1,16% for Θ_T. This is encouraging, but the real measurement of performance is directivity!

In the end, the mean directivity is little affected by these errors. Mean directivity is measured at 31.88 dB. Note that this includes the potholes! As well, the mean directivity of the single beam antenna has dropped as well from 24.17 to 23.93 dB. Icould therefore attribute a loss of 0.24 dB in average to the potholes. Therefore the actual loss in directivity due to errors in beam selection is 0.19 dB.

5.1.3 Treated signal at N=1

Now Ican look at noisier conditions of operation. The start-up time and condition remains the same. However it does takes some time for the algorithm to home in the proper sampling size for this noise level.

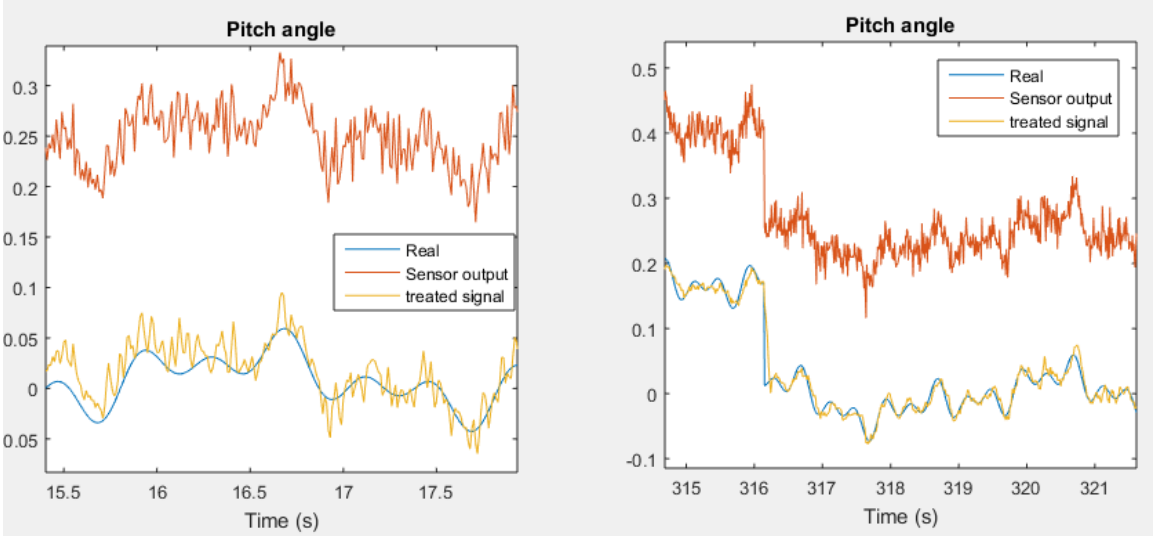


Figure 61 Ω_T At startup and later with N=1

Notice the considerable noise reduction between the early and later measurements of the pitch. This is the result of the increasing sample size. The same applies for the roll angle.

Ω_T "cleans up" as the sample size increase. A look at the total sample size (shown in Figure 62) over the length of this simulation indicates the output of the feedback function.

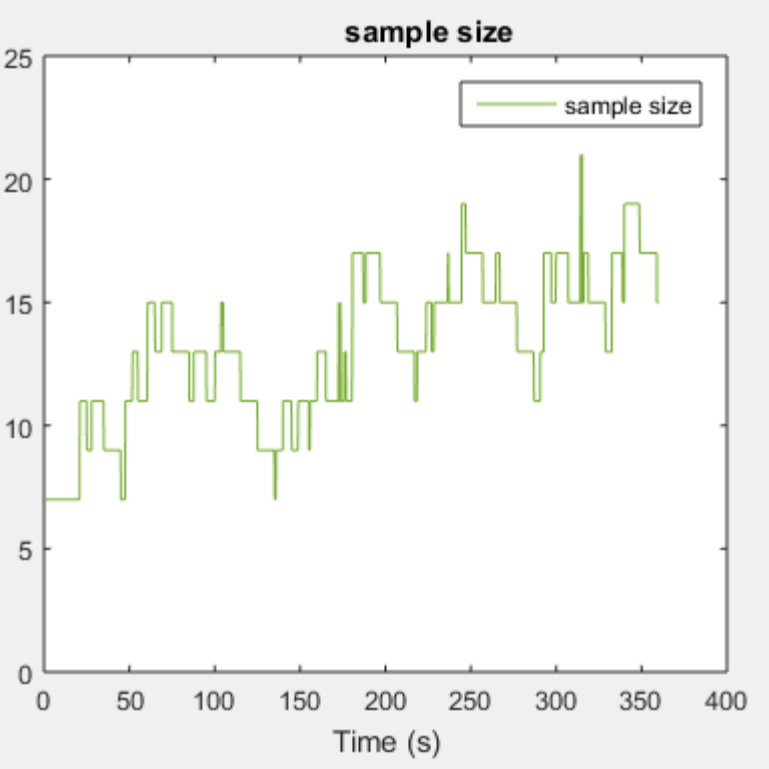


Figure 62 –Total sample size at N=1

Notice the sample size increase up to 19 and then starts to oscillate between 11 and 19. As shown in Figure 62 the initial sample size of 7 is clearly insufficient for these conditions and it quickly increases to compensate.

Applying the same methodology as for the previous noise conditions i.e. (N=0.1), Ican quickly confirm that the increased noise conditions will affect the overall performance in a negative way.

While comparing the digitized versions ρ_T , Θ_T and ρ_R. Θ_R, Ican calculate an error rate of 5.54% for ρ_T and 11.38% for Θ_T. While these are quite an increase, remember that the noise level was increased tenfold. Again the real measure in performance lies in the mean directivity. As per previous simulation the reference mean directivity is 23.93 dB with a loss of 0.24 dB due to potholes. The average directivity for the switched beam has

dropped to 30.67 dB from the nominal 32.30 dB. Removing the “pothole” loss, the increase noise resulted in a net average directivity loss of 1.39 dB

5.1.4 Treated signal at N=2.

At that noise level, the sample size simply maxes out and the algorithm does its best to recover the signal. While comparing the digitized versions ρ_T , Θ_T and ρ_R. Θ_R, Ican calculate an error rate of 7.87% for ρ_T and 20.01% for, Θ_T.

As per previous simulations the reference mean directivity is 23.93 dB with a loss of 0.24 dB due to potholes. The average directivity for the switched beam has dropped to 28.77 dB from the nominal 32.30 dB. Removing the “pothole” loss, the increased noise resulted in a net average directivity loss of 3.29 dB.

5.1.5 Treated signal at variable noise levels (N=[0.1,1])

For good measure, a simulation was setup during which the noise level would vary over time. The noise level was changed four times over the complete simulation:

- 0 to 90 secs N=0.5
- 90 to 180 secs N=1
- 180 to 270 secs N=0.3
- 270 to 360 secs N=0.1

The results were quite satisfying as the sample size tracked the noise accordingly.

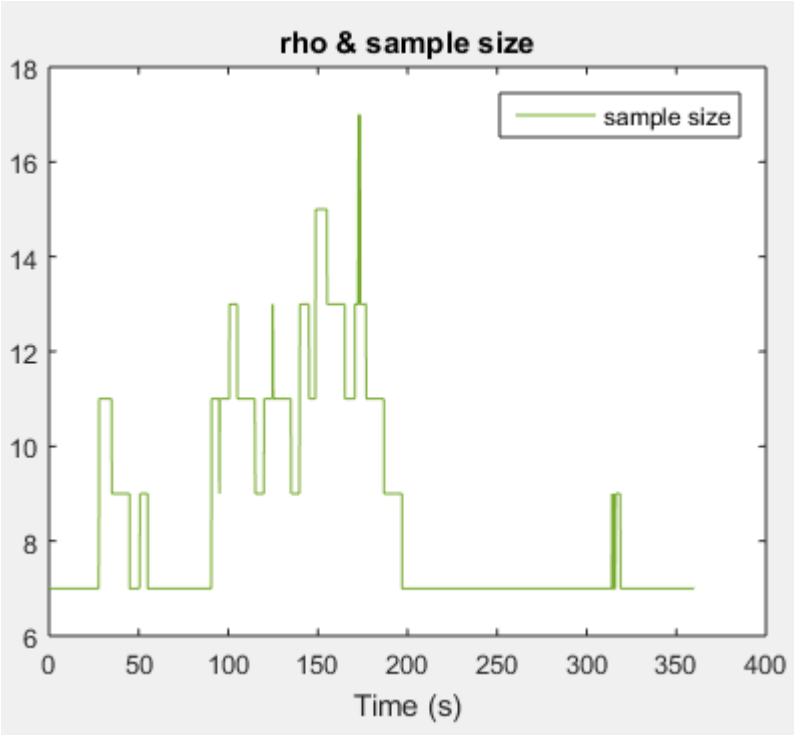


Figure 63 –Total sample size with varying noise levels

The key point is that the parameters of the algorithm allow for adjustment to the noise level in order to maintain performance in various noise conditions.

While comparing the digitized versions ρ_T , Θ_T and ρ_R. Θ_R, Ican calculate an error rate of 2.99% for ρ_T and 5.99% for, Θ_T.

As per previous simulations the reference mean directivity is 23.93 dB with a loss of 0.24 dB due to potholes. The average directivity for the switched beam has dropped to 31.37 dB from the nominal 32.30 dB. Removing the “pothole” loss, the increase noise resulted in a net average directivity loss of 0.69 dB.

5.2 Top level simulations

The top level simulations allow us to evaluate the overall performance 3SBOR by varying certain parameters and running the low level simulations for a repeated amount of time to evaluate the effects of varying said parameter.

The measure of performance here will be the mean directivity of 3SBOR. The following two top level simulations will be performed:

- Range of movement ($\theta_R . \Omega_R$) vs mean directivity
- Mean directivity improvement of algorithm vs noise levels

5.2.1 Amplitude of movement vs Mean directivity

This simulation is setup as the noise level N is set at zero & the potholes are removed to replicate the nominal results presented in the low level simulation. Then the 0.9 factor from **(65)** & **(66)** is replaced a variable dubbed "*ref*". The low level simulation time is 50 seconds. The low level simulation time was reduced to a minimum to save time and computing resources. The top level simulation covers values of "*ref*" ranging from 0.01 to 2 by intervals of 0.1. This corresponds to amplitude of movement (roll and pitch) of ~0º to 8º (0.1 "ref" is about 0.4º).

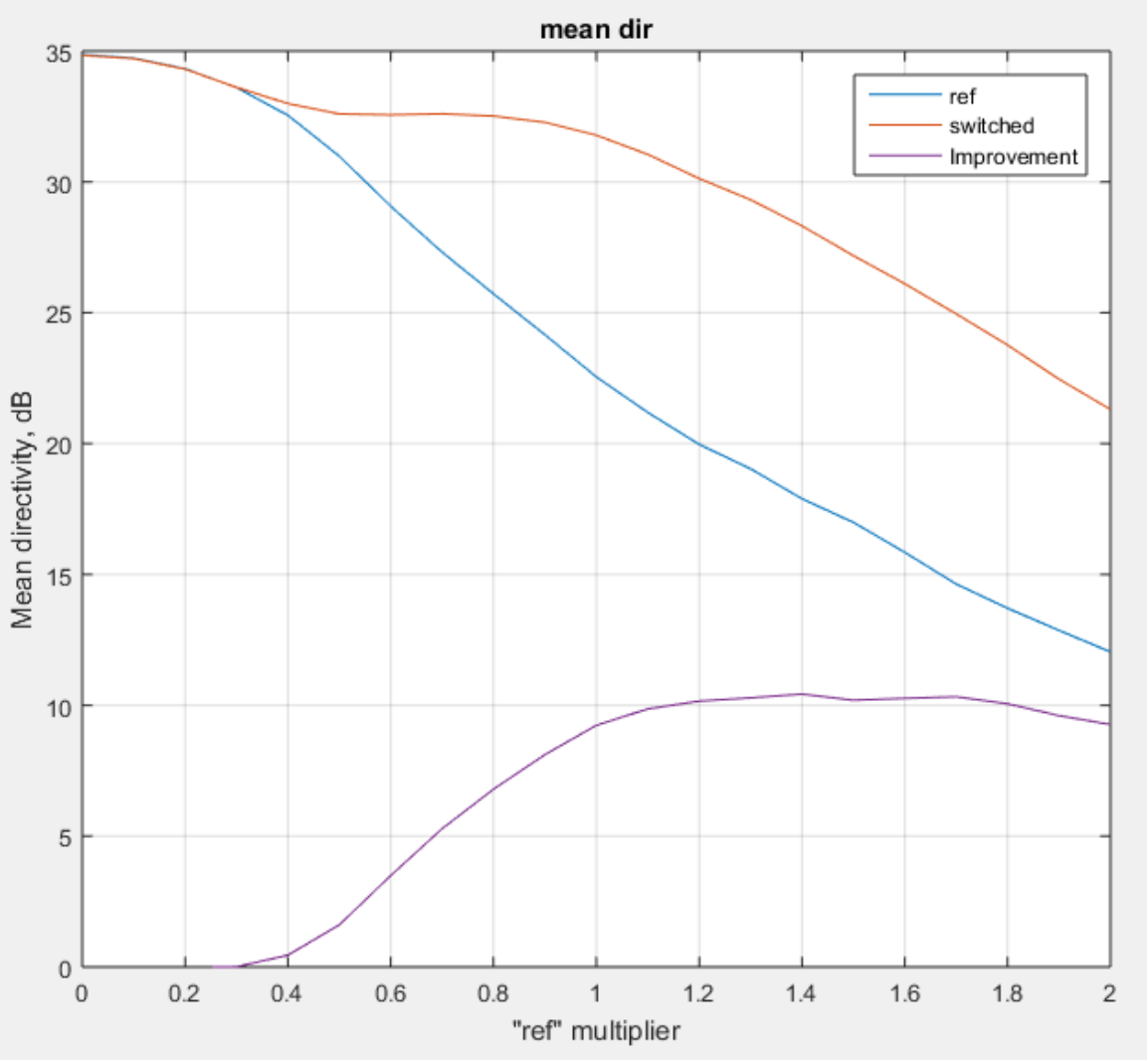


Figure 64 – Mean directivity vs amplitude of movement

Note that below "ref"=0.3, there is no benefit to 3SBOR since the range of movement remains within the HPBW of the center beam. From that point on the 3SBOR activates and effectively maintains the mean directivity up to "ref"=0.9. However the improvement plateaus at "ref"=1.1 (amplitude of movement of 4.4º) and remains constant at around 10 dB up to"ref"=1.8 (amplitude of movement of 7.2º)

This graph provides us with interesting information, and opens the ground for future work. As for an example, could the plateau be stretched by adding more feed horns? Is

there a way to collocate the mean directivity plateau and the maximum directivity improvement values? This will be for future work.

The key result of this top level simulation is that the mean directivity can be maintained above HPBW for an amplitude of movement of 3.5º. This means an effective HPBW of 7º for 3SBOR vs an HPBW of 2º for a similar single beam offset reflector. That's an increase of 350% in HPBW.

5.2.2 Mean directivity improvement of algorithm vs noise levels

Here Iaim at evaluating the mean directivity improvement provided by the algorithm. For this simulation the reference signals amplitude remained constant at "*ref"*=0.9 and the noise level N ranges from 0 to 4. In order to assess the improvement of the algorithm a slight modification was made to the Matlab program. In addition to calculating the directivity using the treated signal, the sensor output (after the mean measurement errors were removed from the sensor output) was used to calculate the directivity, resulting in a mean untreated directivity curve. The difference in mean directivity between the sensor output and algorithm treated output (i.e. treated signal) provided the algorithm improvement as a function of noise.

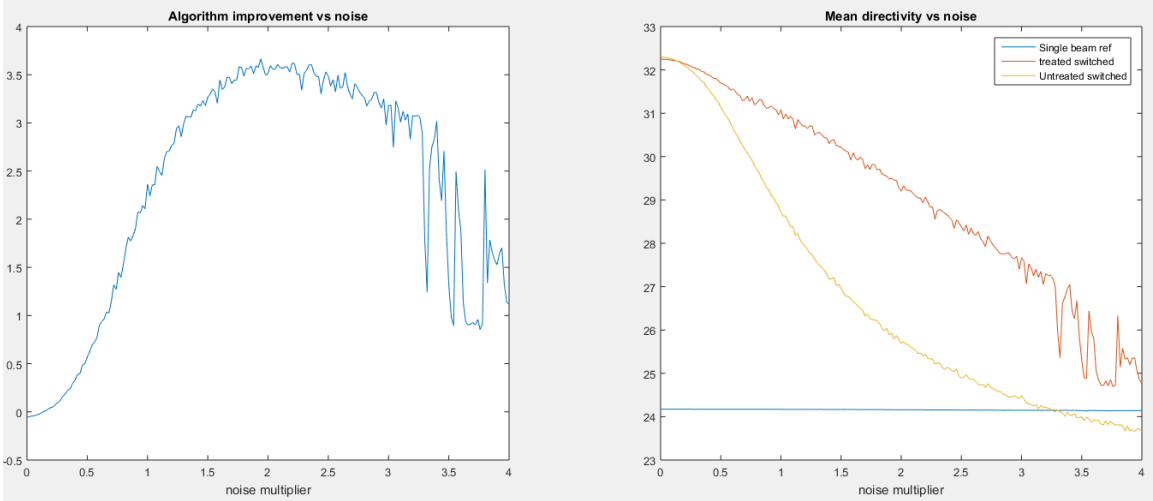


Figure 65 – Mean directivity vs amplitude of movement

The algorithm shows its true performance here, allowing for as much as 3.6 dB of improvement at N=1.9. On the other end (N=0), the treated signal is 0.05 dB below the untreated signal. This is, of course specific to the conditions set in this simulation.

This simulation confirmed that the algorithm behaves as intended; in low noise situation, it almost doesn't interfere with the performance of the system (-0.05 dB mean directivity) and can provide up to 3.6 dB of improvement in mean directivity in high noise (N=2) situation.

Considerable work could be done to optimize the performance and output of the current system. What is the optimal position/speed sample combination? Would adding an acceleration component be beneficial? However, at this point the results serve the purpose of proving 3SBOR as a viable alternative to stabilizing the directivity of a pencil beam antenna without incurring very high costs.

6 3SBOR CONCEPT VALIDATION

In an attempt to fully validate the concept of 3SBOR, the Quality Engineer Test Establishment (QETE) in Gatineau was approached for fabrication and test. The design proposition was simplified from a two dimensions parabola for which the simulation results were presented in Section 5. The prototype is a single dimension dish or in other words, a parabolic curved cylinder section. This greatly simplifies the dish fabrication process. In addition to fabricating the dish, QETE acquired at their cost the following equipment, as presented in Section 3:

- 1X SMA male to SMA male 120 inch RG 400 coax (RF input cable)
- 3X SMA male to SMA right angle male 6 inch 400 coax (switch to horn cables)
- 3X SMA waveguide square cover flange to end Launch SMA Female Waveguide to coax adapter operating from 8.2 to 12.4 GHZ, X band
- 3X WR-90 Standard Gain Horn antenna Operating from 8.2 to 12.4 Ghz with a nominal 10 dB Gain with Square Cover Flange
- 1X ARDUINO DUE microcontroller
- 1X D0G2 MEMS-series inclinometer
- 1X reflective 6-12.0 GHz Coaxial SP3T Switch

An additional advantage of the single dimension dish is cost savings, since a three to one switch is required instead of eight to one (1/3 of the cost) and only three horns vs seven.

6.1 Single dimension design

The key challenge in modifying the design of the 3SBOR to one dimension was to establish the width of the dish itself. Using a methodology similar to that described throughout Section 4, the dish surface was defined using the following plane equation.

Plane equation, 1D dish:

$$X(u,v) = v \quad Y(u,v) = u \quad Z(u,v) = \frac{u^2}{4 \cdot F} \;\; u = [h1, D1] \;\; v = \left[-\frac{1D}{2}, \frac{1D}{2}\right] \tag{77}$$

The problem here is twofold; a dish too narrow will have a poor gain. In addition, a dish that is too wide will have a poor gain as well, as the further away you'd get from v=0, the further away you'd get from the focal point which ultimately results in destructive interference and reduces the gain as well. Therefore Imust determine the optimal value of 1D for the given design.

6.2 Single dimension dish in HFSS

In order to optimize the value of 1D in **(77)**, HFSS's *optimetric* analysis was used to scan a range of 1D values from 10mm to 900mm. The key results are shown in Figure 66.

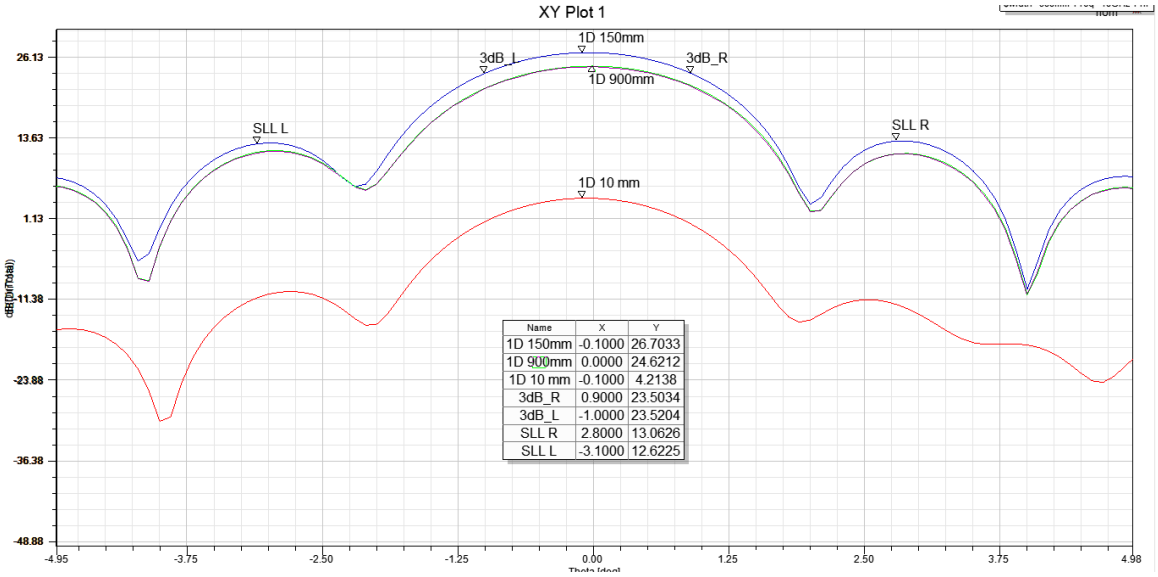


Figure 66 – Optimization of 1D dish width

Increments of 10 mm of the value 1D were made in this case. Directivity peaks at 1D = 150 mm for 26.7 dB and start decreasing as the dish size is increased. SLL are 13 dB down and the beam width is 1.9º. The beam is squinted by 0.1º, even though the location of the center beam is identical to the 2D dish.

Given Figure 66, selecting a value of 150mm for the 1D width yields a peak gain of 26.7 dB. Based on this analysis and the methodology from Section 4, the structure shown in Figure 67 was created in HFSS. The assumption was made that the theory of feed placement would transfer from the 2D to 1D without extensive redesign. The horn feed has only three elements; they are the center, 90º & 270º elements as shown in Figure 24. The other horns were removed.

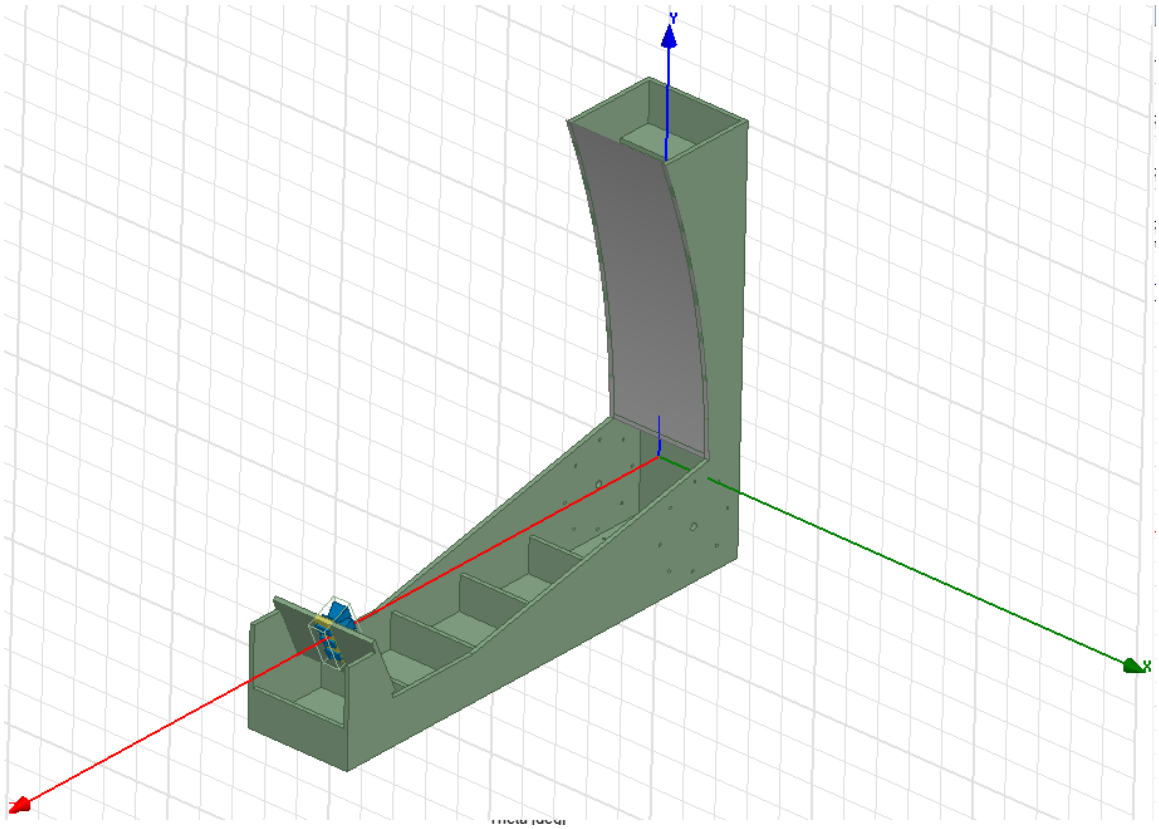


Figure 67 – Design structure of 1D dish width

This design was made so that it would be as simple as possible to build. The side panels provide the curve of the dish and the angle of the horn feed. Back plate and traverse members will guarantee rigidity. The circular holes pattern is for turntable attachment.

As for the 2D design, this structure had to be simulated will all materials as perfect conductors due to computing resource restrictions.

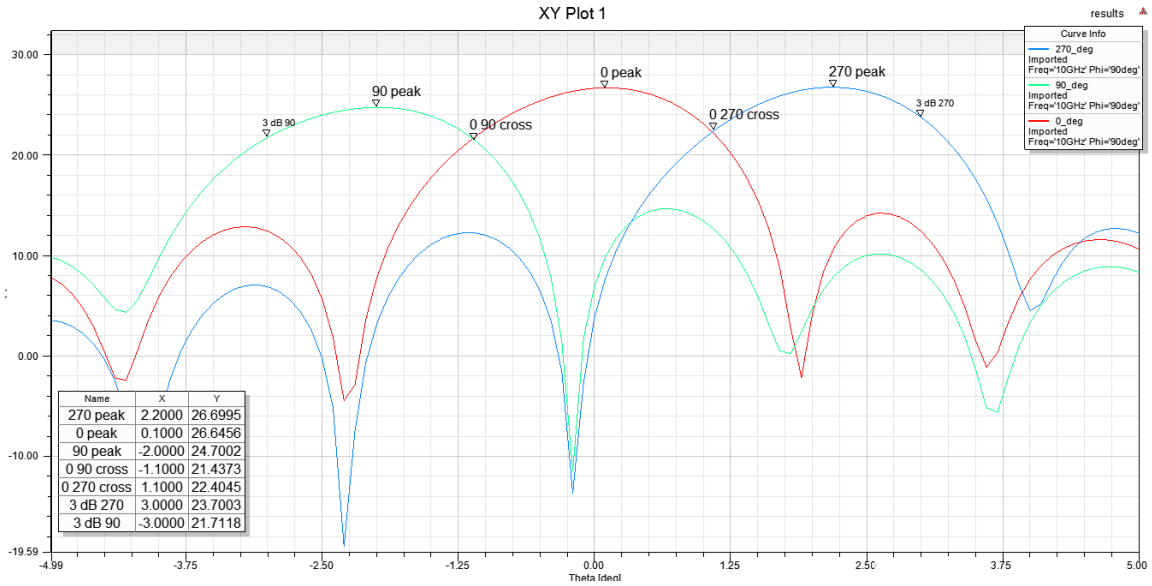


Figure 68 – 1D all beams simulation

While the results may not be as consistent as with the 2D dish, they confirm the simplified design will provide a performance suitable for a proof of concept. The 90° beam lost 2 dB of peak directivity. The crossover point is on par with specifications when looking upwards, (i.e. crossover at HPBW) but not when looking down (crossover 5 dB down from peak). SLL is 12 dB down from peak in average.

Given the simplicity of this design, the results are more than suitable. The lower directivity when looking downward may very well result from the interference of the rather large plate used to support the horn feed. A smaller plate may give in improved directivity, but will result in a more complex design.

Figure 69 shows a 3d directivity plot of the three beams.

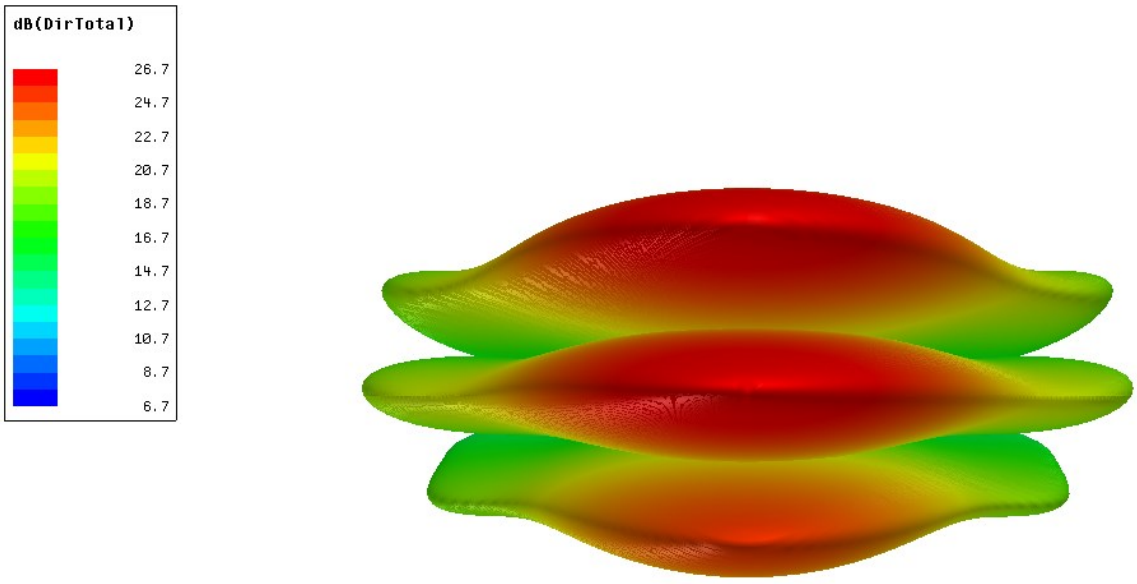


Figure 69 – 1D all beams 3D polar results

Above is the coverage provided by the 1D design. An excel datasheet of this graph was produced to be used in the Matlab simulation. The results of the simulation will be used as a reference during testing.

6.3 Antenna Simulation Setup

As shown in Figure 67, the prototype is made to attach a turntable on the side. Although the movement model wasn't designed for such setup, attempts will be made to use it. Noting the fact that the roll parameters are null, the model should simplify to a circular rotation. For simulation purposes, the axis of the turntable will represent the location of the pitch axis.

Movement model parameters:

$$pz = -43.6\,mm \quad py = 0 \quad ha = 601.7mm \quad rz = ry = \varphi = 0 \tag{78}$$

Note that for this simulation, all roll axis components are set to zero, since the roll component cannot be confirmed with the 1D dish. Such test will be for future work.

In order to control the movement of the antenna an ET-250D turntable will be used. This turntable has a rotation speed of 0.35 RPM, which translate into an angular speed of 2.1º per second. Given this specification it will be possible to recreate the following sine wave using the turntable.

"Turntable" reference signal:

$$\Omega_{Turntable}(t) = 2.1 \cdot \left(\sin\frac{\pi}{2}t\ \right) \tag{79}$$

This will give an oscillation amplitude of ±2.1º with a period of 4 secs. This is about 4-8 times slower than real life application, but it will still allow for measuring the switched directivity and confirm the concept.

6.4 Matlab Antenna Simulation Results

Taking in account the parameters of **(78)**, the signal of **(79)** and a null roll stimulus, a Matlab simulation was done using minimal noise setting (N=0.1) and the results from HFSS for the 1D dish. The simulation time was 72 seconds. The beam switching decision point was changed from $\rho_T = 1.1^o$ to $\rho_T = 1^o$.

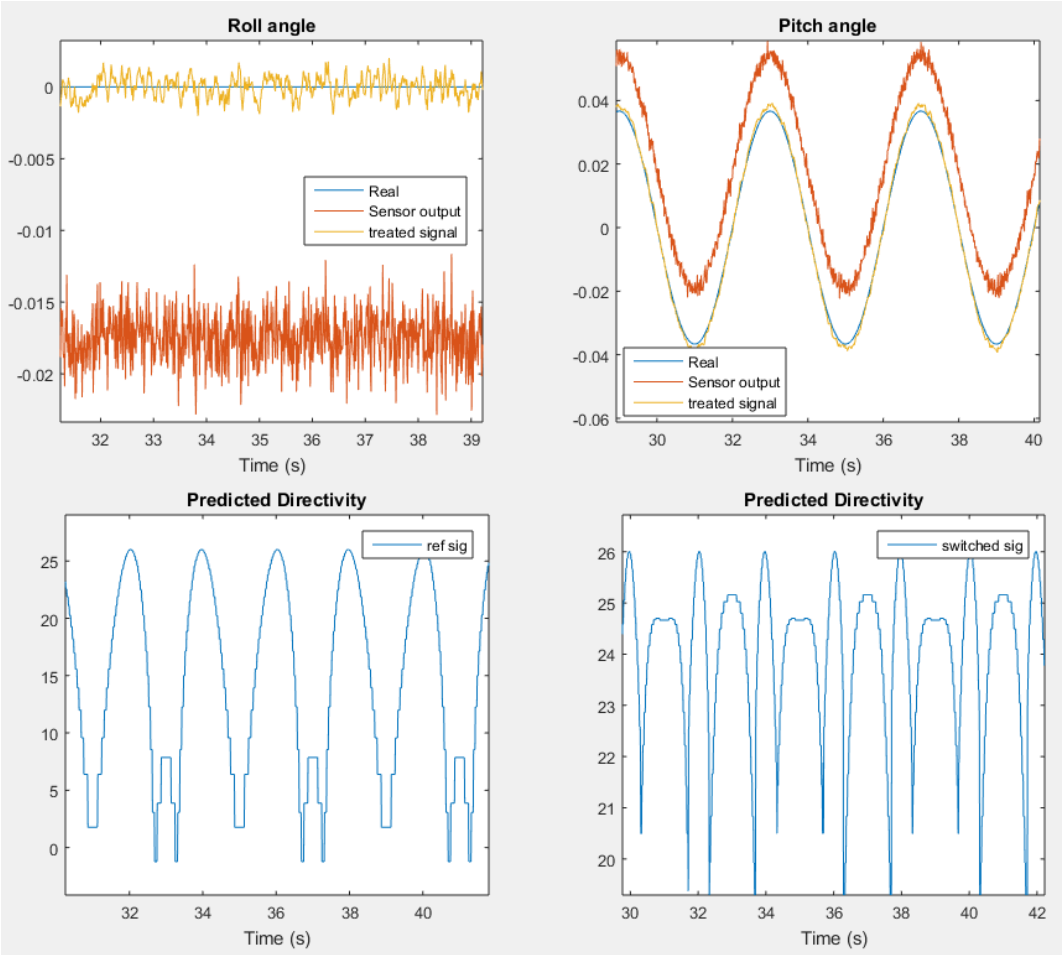


Figure 70 – 1D all beams 3D polar results

*Top left is the roll input. The channel is kept as a noise source. Top right is the pitch input as described in **(79).** Bottom left is the predicted single beam directivity. Bottom right is the predicted switched directivity.*

The results of the 1D simulation are quite encouraging. As expected, the switching error is quite small at 0.153%. The mean directivity of the single beam antenna sits at 15.00 dB while the mean directivity of the switched beam sits at 24.06 dB. The improvement sits at 9.06 dB which is consistent with the data from the 2D dish.

6.5 Test procedure

The goals of the tests with the 1D antenna are:

1. To confirm that the technology selected in Section 3 is adequate and compatible with the requirements. Interfacing these components will be a challenge. Getting the Arduino Due to run the signal treatment algorithm will be another.

2. To confirm the 1D design performance is consistent with what is described in sub-section 6.2; a turn table will be used to characterize the vertical beam pattern of the 1D design. Once measured, these beam patterns will be compared with the HFSS results.

3. To confirm the accuracy and validity of Matlab simulations as described in sub-section 6.4. The conditions of the Matlab simulations will be replicated as closely as possible. While rotating the antenna on the pitch axis, the single beam directivity will be measured by evaluating the received signal strength. The same methodology will be done using switched beams.

6.5.1 Antenna mounting & assembly

The final setup of the Arduino Due controller is shown in Figure 71, as installed in the antenna:

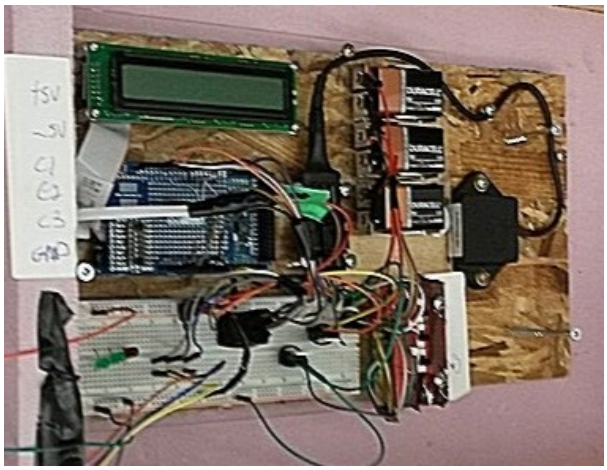


Figure 71 – Arduino Due with inclinometer, breadboard, LCD display & power supplies.

In order to guarantee adequate power supplies, a set of three 9 volts batteries where used. One battery supplies the power to the inclinometer. Using a 5 volts voltage regulator, another one powers the Arduino Due and provides +5 volts to the RF switch. The third one provides -5 volts to the RF switch with the use of a voltage regulator. A logic level translator is used to adjust the digital I/O of the Arduino (3.3 volts) to 5 volts (required to operate the RF switch). Three flip switches are used to interrupt the power supplies in order to follow the power up/power down sequence of the RF Switch.

The Matlab code was translated in order to use Arduino libraries and I/O. The code is provided in the Appendix B. One main difference between the two codes is that the Roll/Pitch Sensor functions as described in Section 4.5.3 were modified from constantly updating themselves with new data to a fixed sensor average of 1500 samples measured at boot. Once the code is loaded in the flash memory of the Arduino Due, it will run in a loop on power up, without requiring to be connected to a computer. The final code was loaded as shown Figure 72:

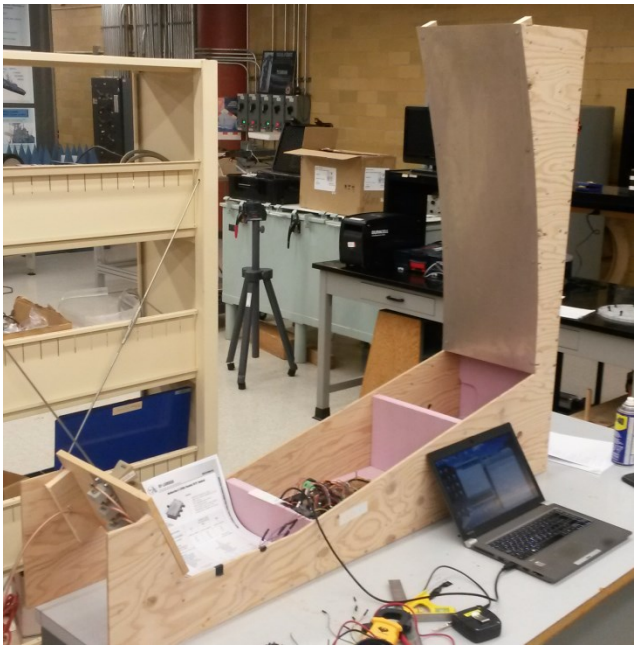


Figure 72 – Final loading of code in Arduino Due.

The assembled 1D prototype with the control board installed between support traverses is shown here. Note that the support members are made of 1 inch Styrofoam instead of ½ inch plywood, making the antenna much lighter. The side panels are made of ¼ inch plywood, further reducing the weight of the assembly.

The antenna mounted to an ET-250D turntable as shown in Figure 73:

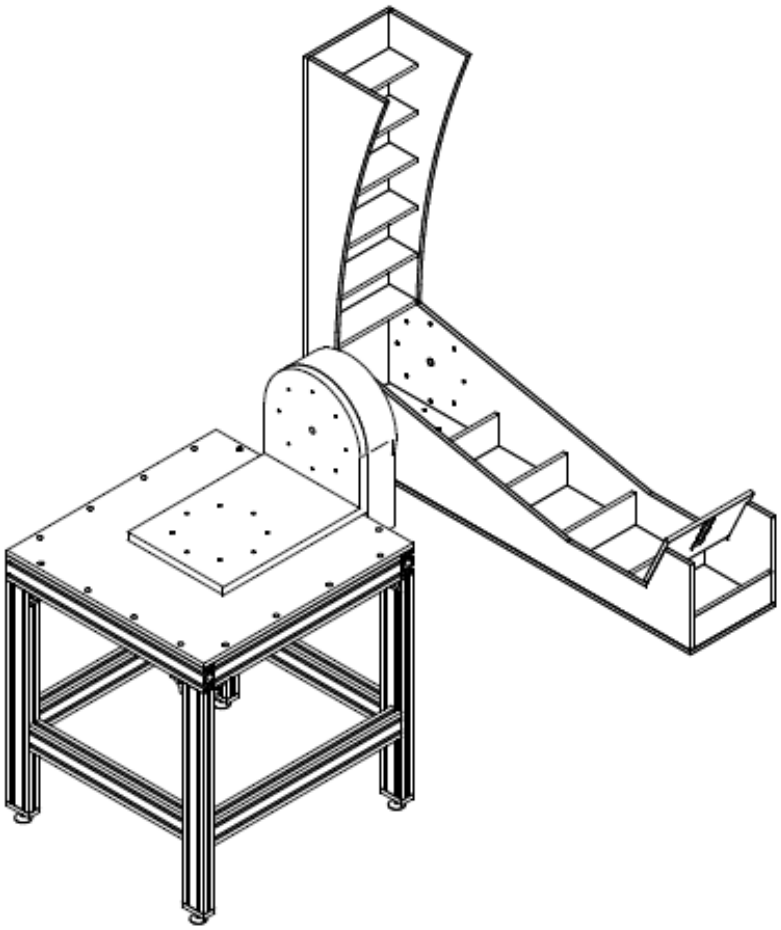


Figure 73 – 1D antenna mounted to turntable

This installation will allow us to precisely control the "pitch" angle of the antenna. A 10 GHz plane wave is used for transmission tests. The accuracy of the accelerometer is confirmed before the transmission/reception tests.

Shown in Figure 74 is the 1D prototype mounted on the turntable in a laboratory setting:

Figure 74 – 1D antenna mounted to turntable

Using a digital level, it was possible to confirm the precision of the inclinometer and control algorithm. The measurement error is well within that of the inclinometer specifications and test requirement.

6.5.2 Placement of testing equipment

The aim is to confirm the far-field beam patterns of the 3SBOR. Considering the current frequency of operation and size of the dish, the minimum distance for far field is of 54 meters. QETE has recommended that the test be performed on the rooftop of the National printing bureau.

Figure 75 shows a satellite view of the test setup.

Figure 75 – Test location.

The rooftop structures provide adequate reference points, allowing for accurate alignment of the 3SBOR with the transmit antenna. The free space loss for 72m at 10 GHz is 89.59 dB

The transmit antenna is setup on a tripod as shown in Figure 76.

Figure 76 – Transmit antenna setup.

The wave generator is set at a frequency of 10 GHz, continuous wave. The cable losses were measured at 1.7 dB. The power output of the generator is set at 21.7 dBm, giving us 20 dBm at the antenna wave port.

The receiver antenna (3SBOR) is setup as shown in Figure 77.

Figure 77 – Transmit antenna setup.

The spectrum analyser is used in zero span mode to record the maximum power received at 10 GHz in function of time. The cable and the switch insertion losses were measured at 7.7 dB

6.5.3 Locked beam test

The locked beam test aims at characterizing each of the three beam pattern of the 1D prototype. Only the E-plane is measured, the H-plane being of no concern in this case.

The following procedure was used to characterize each beam pattern.

1. The RF switch is replaced by a connector (this connector happened to have a default, resulting in an additional insertion loss of 2.3 dB, resulting in a total 8.3 dB of line losses).
2. The test transmitter as shown in Figure 76 sends a 10 GHz plane wave at the antenna.
3. Using the turn table, the antenna shown in Figure 77 is rotated from -20º to +20º to measure the beam pattern.
4. Using the spectrometer in zero span mode at a center frequency of 10 GHz and a resolution bandwidth of 10 KHz, the maximum signal strength as a function of time is recorded for a total of 19 seconds. (The turntable rotates at a constant speed of 0.35 RPM or 2.1º per second.)
5. The directivity of antenna is evaluated using the Friis equation.
6. This is repeated for each the side beam.
7. The results is compared with those shown in Figure 68.

6.5.4 Locked beam test results

The following parameters will be used for the Friss equation to calculate the directivity of the 1D prototype:

$$free\ space\ loss = 89.59\ dB \quad Line\ losses = 8.3\ dB$$
$$Transmitted\ power = -10\ dBW \quad Transmitter\ gain = 15\ dB$$

(80)

Using these parameters the center beam was scanned, giving the results shown in Figure 78.

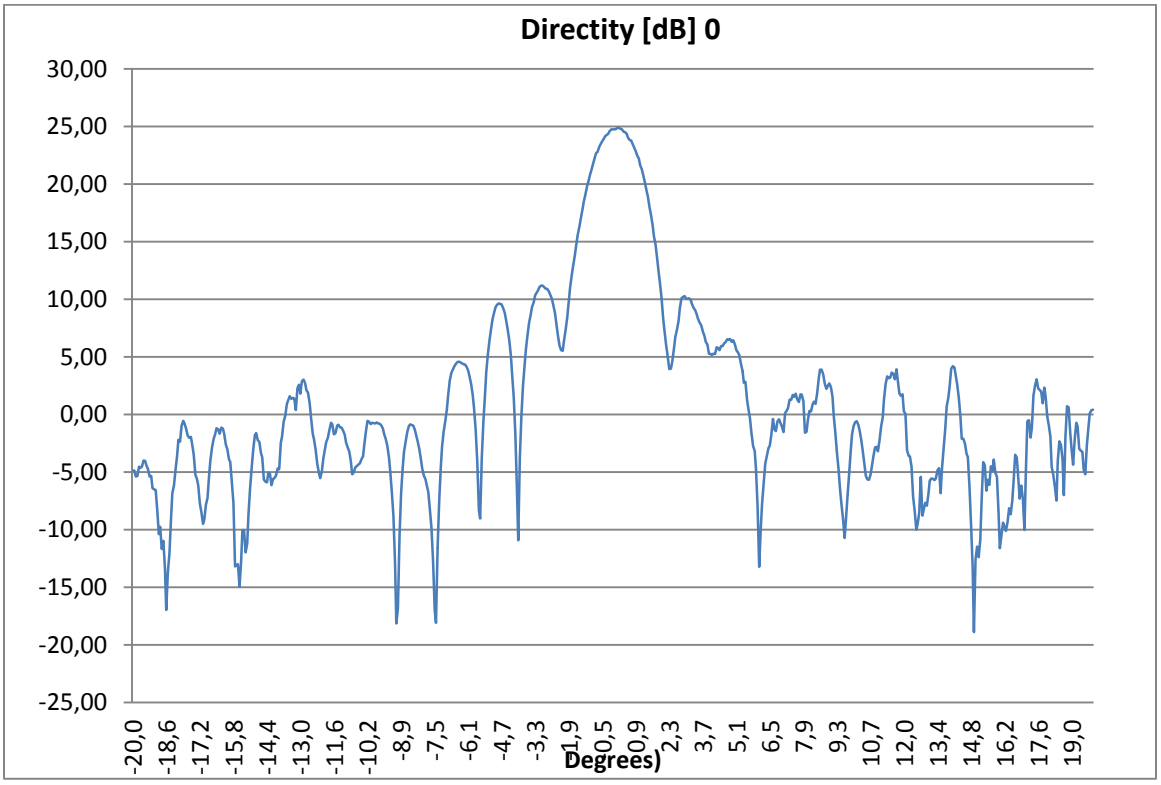


Figure 78 – main beam (0) measured directivity

Note that peak directivity is 1.7 dB below predicted maximum.

These results were compared with the simulation results of HFSS as shown in Figure 79.

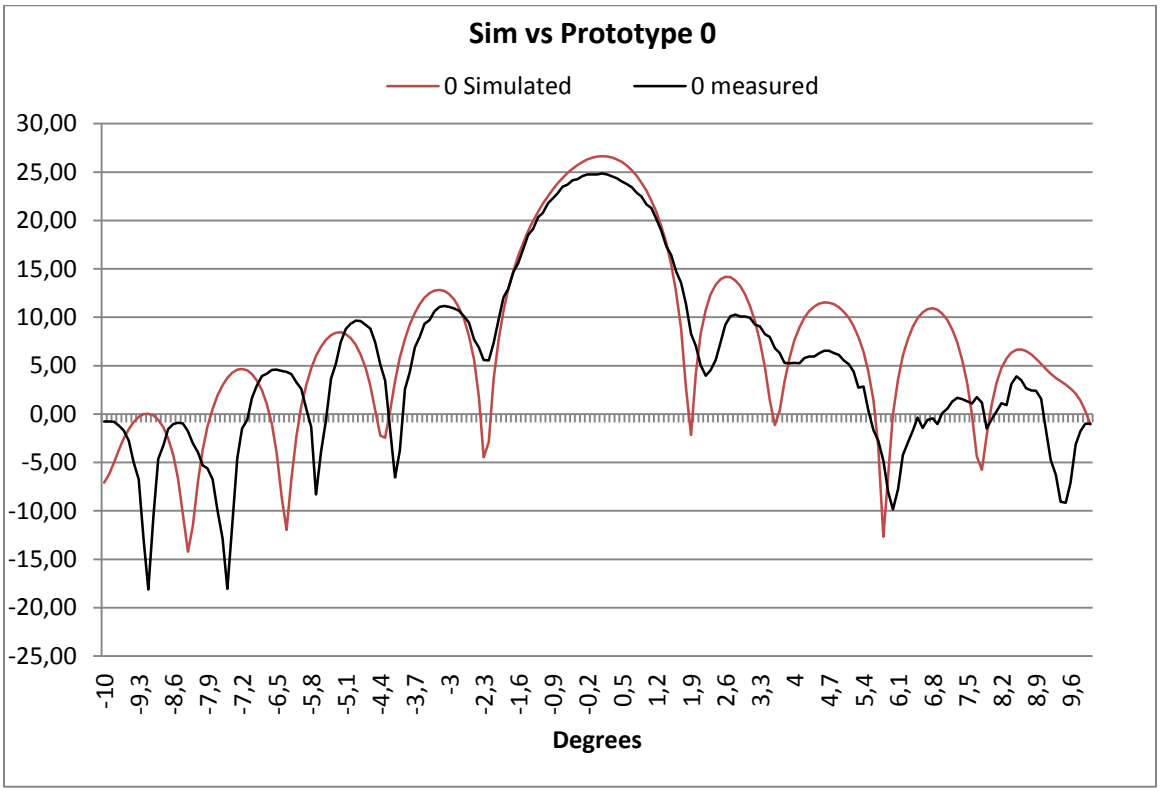
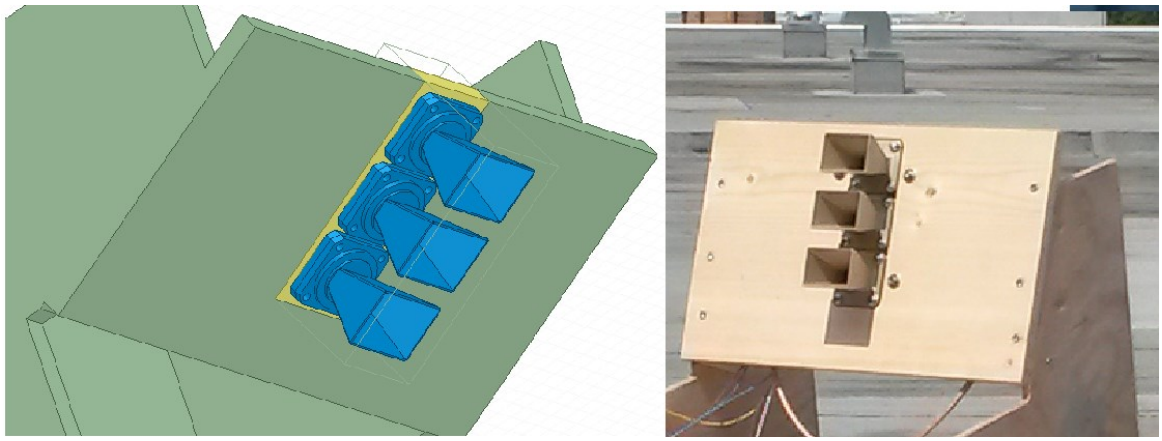


Figure 79 – Main beam (0) measured directivity vs HFSS results

The lower gain is caused by an error in the assembly of the prototype. Nevertheless, the results are more than conclusive with the simulated data. The first nulls are not nearly as deep on the measured data. The side lobes past +4° are blocked by the antenna structure.

Figure 80 shows the error that caused the drop in maximum directivity. Whereas the submitted design mounted the horn feed directly on the support plate made of ½ inch plywood, the engineers at QETE modified the design. The horns were first mounted on a 1/8 inch thick aluminium plate and that plate was then screwed in a cavity in the support plate. This caused the horn feed to be recessed by 9.525 mm from the simulated position.



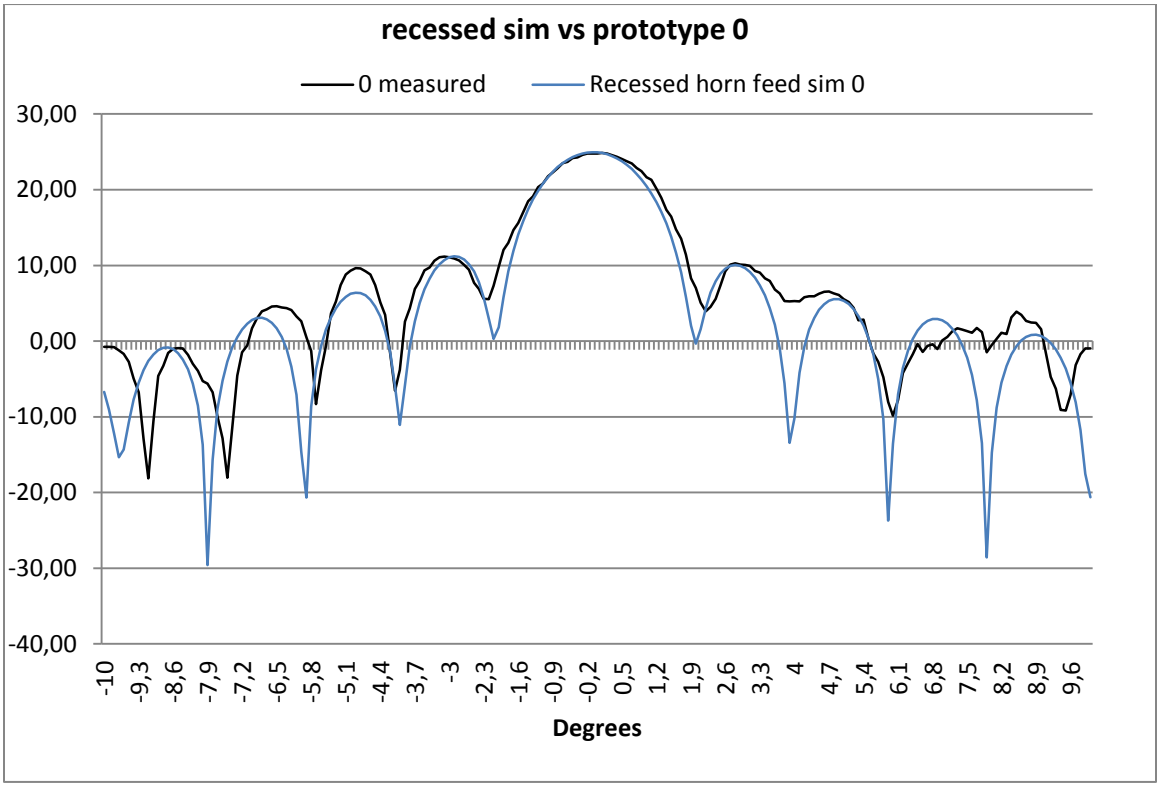


Figure 80 – Prototype assembly error & results

The submitted horn feed design is shown on the left and the assembled horn feed is shown on the right. The prototype feed is recessed by 9,525 mm from the focal point of the dish. A simulation in HFSS was done with the horn feed in this position. It shows a drop in maximum directivity of 1.7 dB. The bottom graph shows the results of the HFSS simulation with the horn feed in recessed position. The close match between the beam patterns confirms the cause of the lower directivity of the prototype.

Figure 81 shows the beam pattern of the horn located at the top (90 degrees).

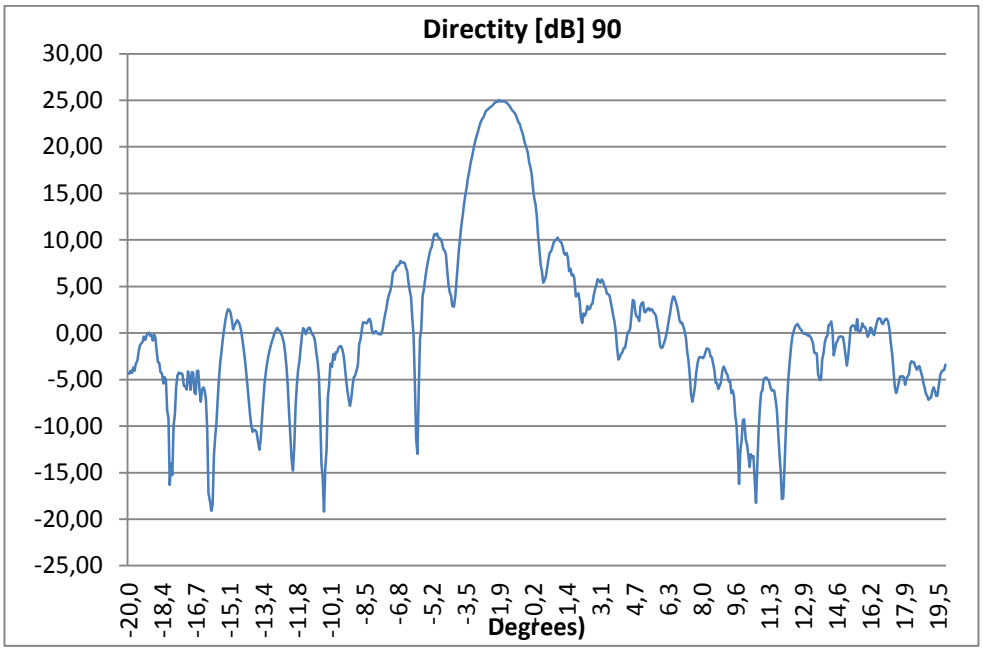


Figure 81 – Top beam (90) measured directivity

These results were put against the simulation results of HFSS as shown in Figure 82:

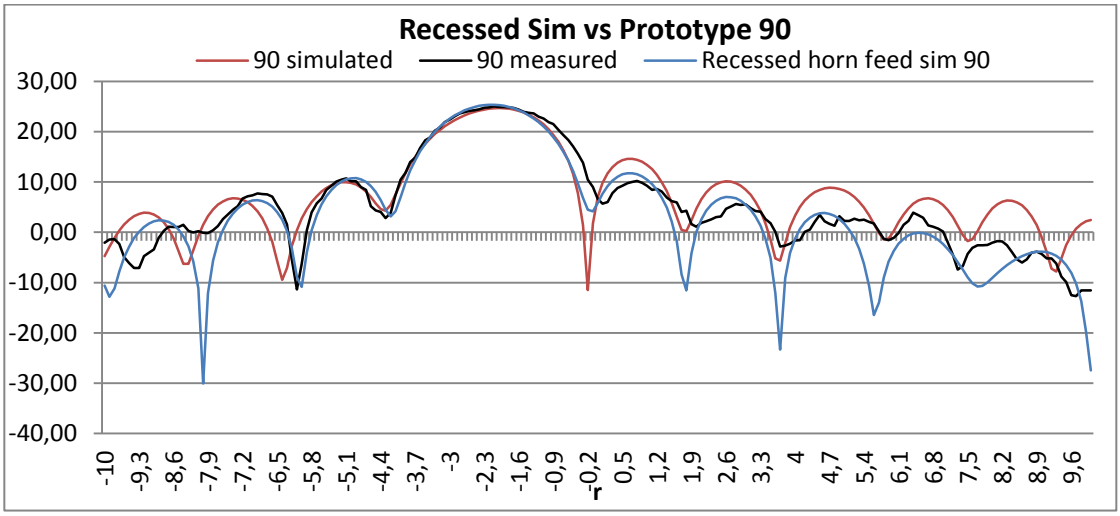


Figure 82 – Top beam (90) measured directivity vs HFSS results

Unlike the center beam, the top beam (90) is not affected by the recessed horn shown in Figure 80.

Figure 83 shows the beam pattern of the horn located at the bottom (270 degrees).

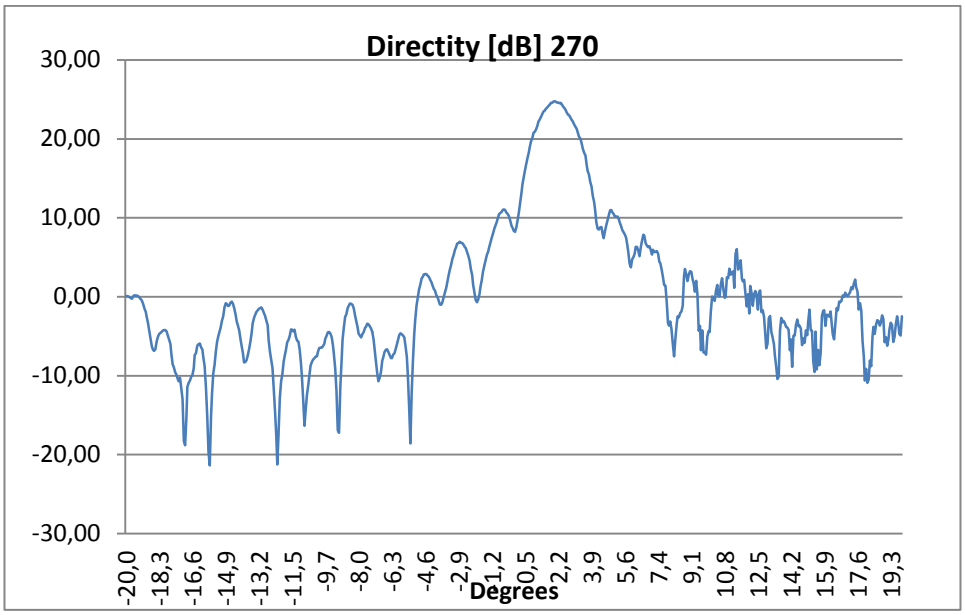


Figure 83 – Bottom beam (270) measured directivity

These results were put against the simulation results of HFSS as shown in Figure 84.

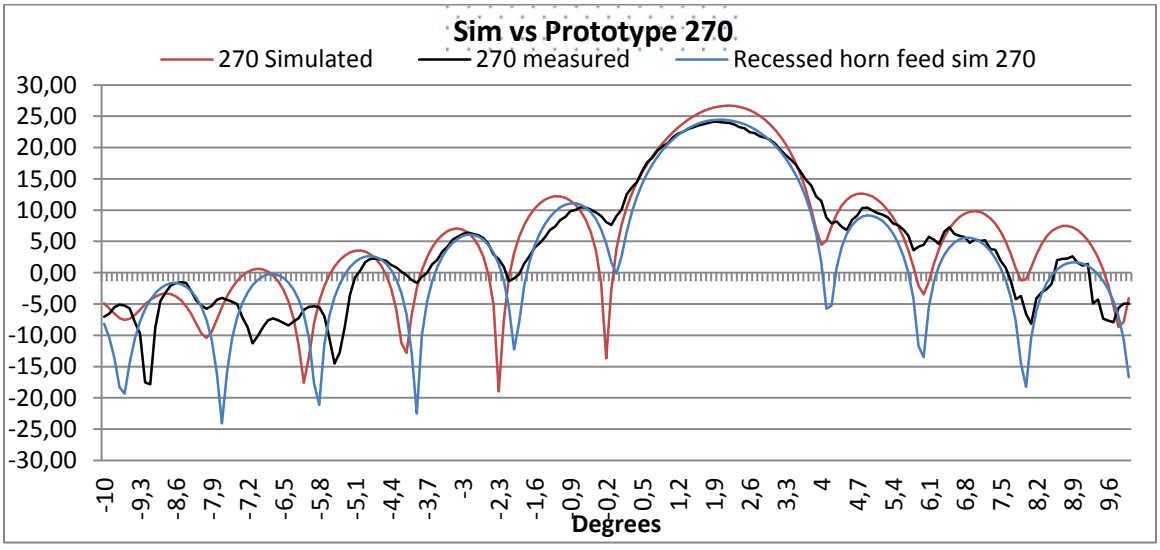


Figure 84 – Bottom beam (270) measured directivity vs HFSS results

This beam is the most affected by the assembly error, causing a loss of 2.5 dB in peak directivity.

Figures 78, 81 & 83 were combined to reproduce the simulated results shown in Figure 68. Figure 85 display the results:

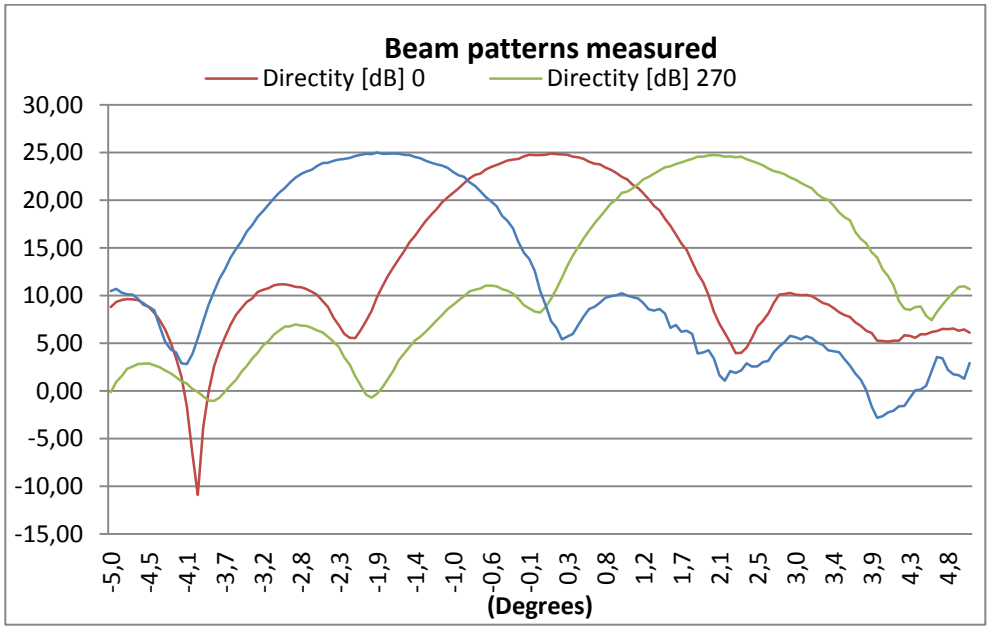


Figure 85 – Bottom beam (270) measured directivity vs HFSS results

*The assembly error had the unforeseen and beneficial effect of averaging the peak directivity of the three beams, which are now all slightly below of 25 dB. The beams intersect at -0,9º and 1,1º. Each beam's peak is offset by approx. +0,1º which indicates that the support plate was mounted 2,5mm too high (according to **(47)**). HPBW is constant for each beam at 2º. SLL is 11 dB on average.*

Although the assembly errors resulted in a lower maximum directivity and squinting the beams by 0.1º, these errors didn't prevent the system to work as intended during the switched beam test.

6.5.5 Switched beam test

The switched beam test aims at characterizing the active performance the 1D antenna design. Only the E-plane was measured, the H-plane being of no concern in this case. The following procedure was used:

1. The sensor and Arduino are brought online; the RF switch connects to the optimal beam depending on the antenna orientation.
2. The test transmitter as shown in Figure 76 directs a 10 GHz plane wave at the antenna.
3. Using the spectrometer in zero span mode with a center frequency of 10 GHz and a resolution bandwidth (RBW) of 10 KHz, the maximum signal strength is recorded while the turn table rotates the antenna from -10º to 10º in order to confirm the prototype is working as intended.
4. Using the spectrometer in zero span mode with a center frequency of 10 GHz and a resolution bandwidth (RBW) of 30 KHz, the maximum signal strength is recorded while the turn table oscillates the antenna from 2.5º of amplitude to a maximum 4º (in a fashion to replicate **(79)**) for a total of 60 secs. This will allow for calculation of average directivity in function of movement amplitude.
5. The results are compared with the simulations.

6.5.6 Switched Beam test results

The following parameters are used for the Friss equation to calculate the directivity of the prototype during the angular sweep. The insertion loss of the switch was increased to match other tests. The same battery powers the +5 volts of the RF switch and the Arduino due. As the voltage dropped below 5 volts the Arduino due kept operating for the test but the switch insertion loss increased. The insertion loss of the switch was estimated to match the other results of the test.

$$free\ space\ loss = 89.59\ dB \quad Line\ losses = 6\ dB \quad Switch\ insertion\ loss = 2.8\ dB$$
$$Transmitted\ power = -30\ dB \quad Transmitter\ gain = 15\ dB$$

(81)

Figure 86 shows the results of the angular sweep, calculated with the parameters of **(81)**.

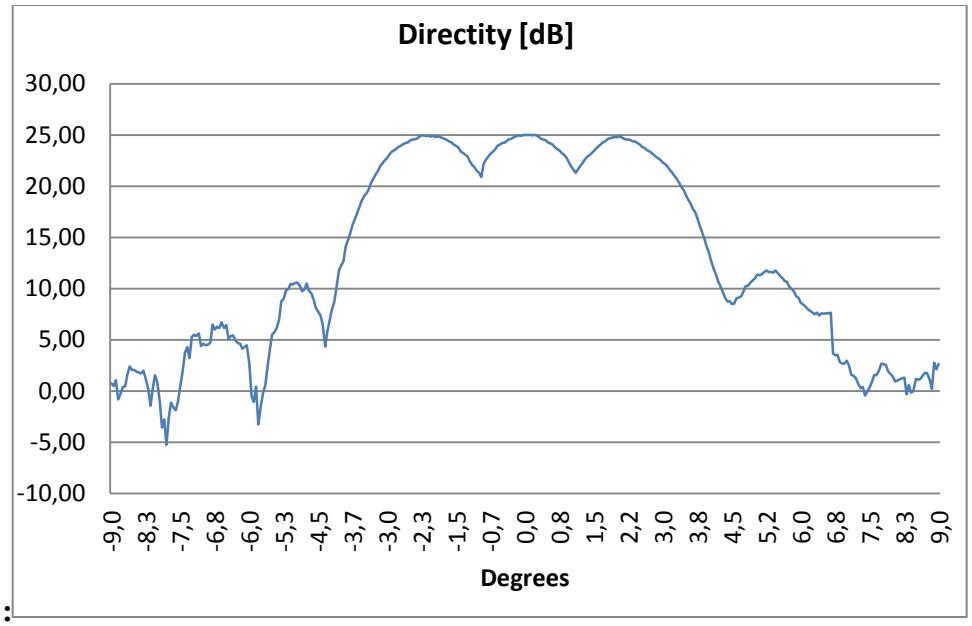


Figure 86 – 3SBOR in operation, single sweep vs angle

The controller is operating as specified, switching beams at 1°. This result is in line with expectations, taking in account the variation induced by the assembly error.

Given the conclusive results of the angular sweep the test moved forward to the repeated oscillation of the antenna. This allows for evaluating the average directivity of the switched beam antenna. Unfortunately, it wasn't possible to replicate **(71)** exactly as the smallest angular step of the turn table is 0.5º.

New batteries were installed, thus bringing back the insertion loss of the RF switch to 1.7 dB. Additionally, the transmit power was increased from 0 dBm to 20 dBm.

$$free\ space\ loss = 89.59\ dB \quad Line\ losses = 6\ dB \quad Switch\ insertion\ loss = 1.7\ dB$$
$$Transmitted\ power = -10\ dBW \quad Transmitter\ gain = 15\ dB$$

(82)

Figure 87 shows the directivity of the antenna oscillating from -2.5º to 2.5º:

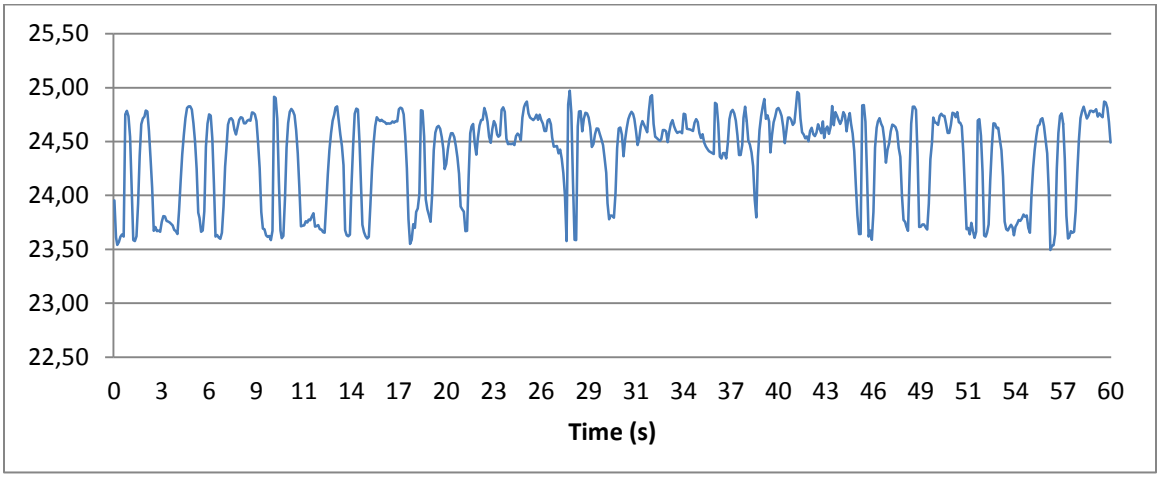


Figure 87 – Directivity, 3SBOR oscillating from -2.5 º to 2.5 º

The average directivity sits at 24.37 dB while the simulated average directivity is 24.15 dB.

Figure 88 shows the directivity of the antenna oscillating from -3º to 3º:

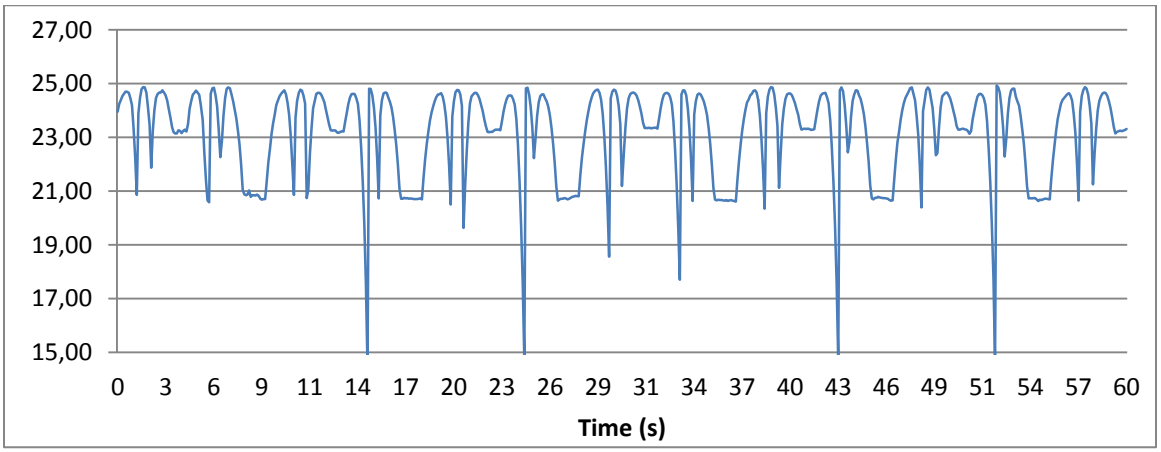


Figure 88 – Directivity, 3SBOR oscillating from -3 º to 3 º

The average directivity sits at 23.18 dB while the simulated average directivity is 23.57 dB.

Figure 89 shows the directivity of the antenna oscillating from -3.5º to 3.5º:

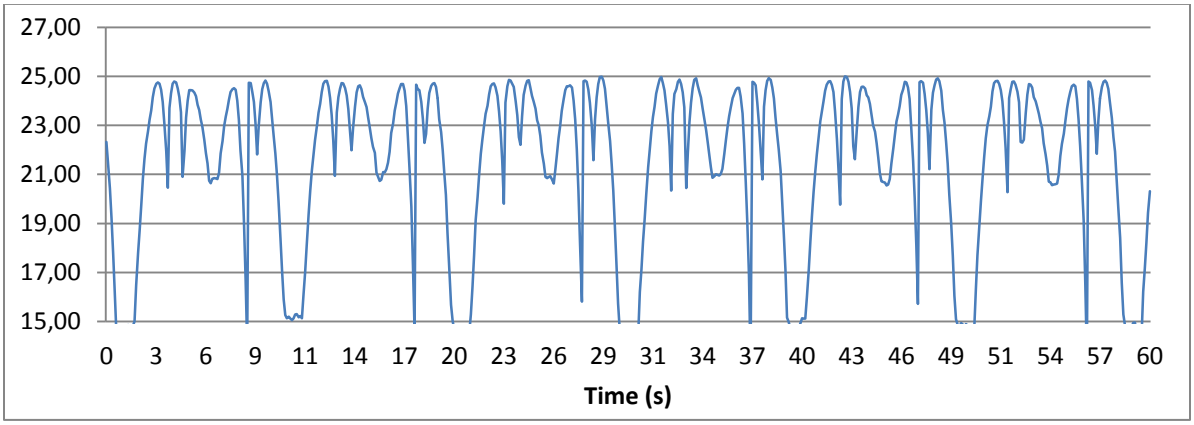


Figure 89 – Directivity, 3SBOR oscillating from -3.5 º to 3.5 º

The average directivity sits at 21.51 dB while the simulated average directivity is 22.03 dB.

Figure 90 shows the directivity of the antenna oscillating from -4º to 4º:

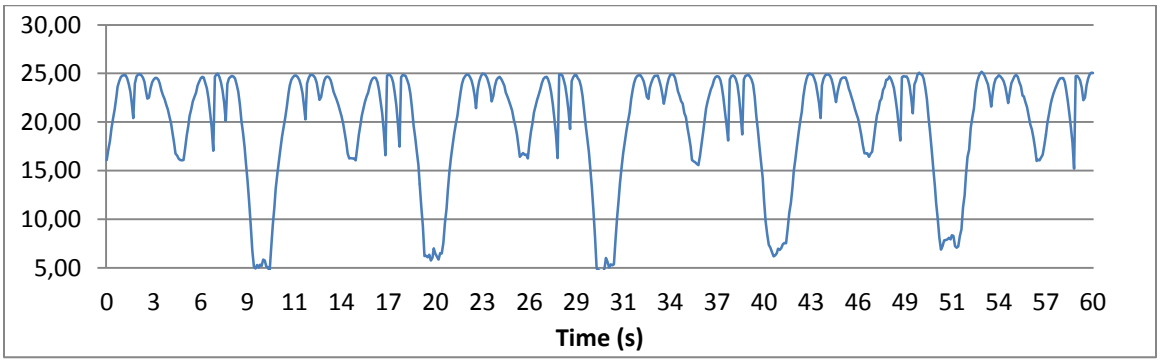


Figure 90 – Directivity, 3SBOR oscillating from -4 º to 4 º

The average directivity sits at 19.86 dB while the simulated average directivity is 18,38 dB.

Using the statistical data provided by Figure 87-90, it is possible to establish a relation between the average directivity of the prototype and the amplitude of movement. In addition, the average directivity of the single locked beam prototype was extrapolated by averaging the measured directivity of the main beam over a given range of movement.

Figure 91 confirms that the 3SBOR functions as predicted and that the 1D design behaves very much alike the 2D design described in Section 4 & 5:

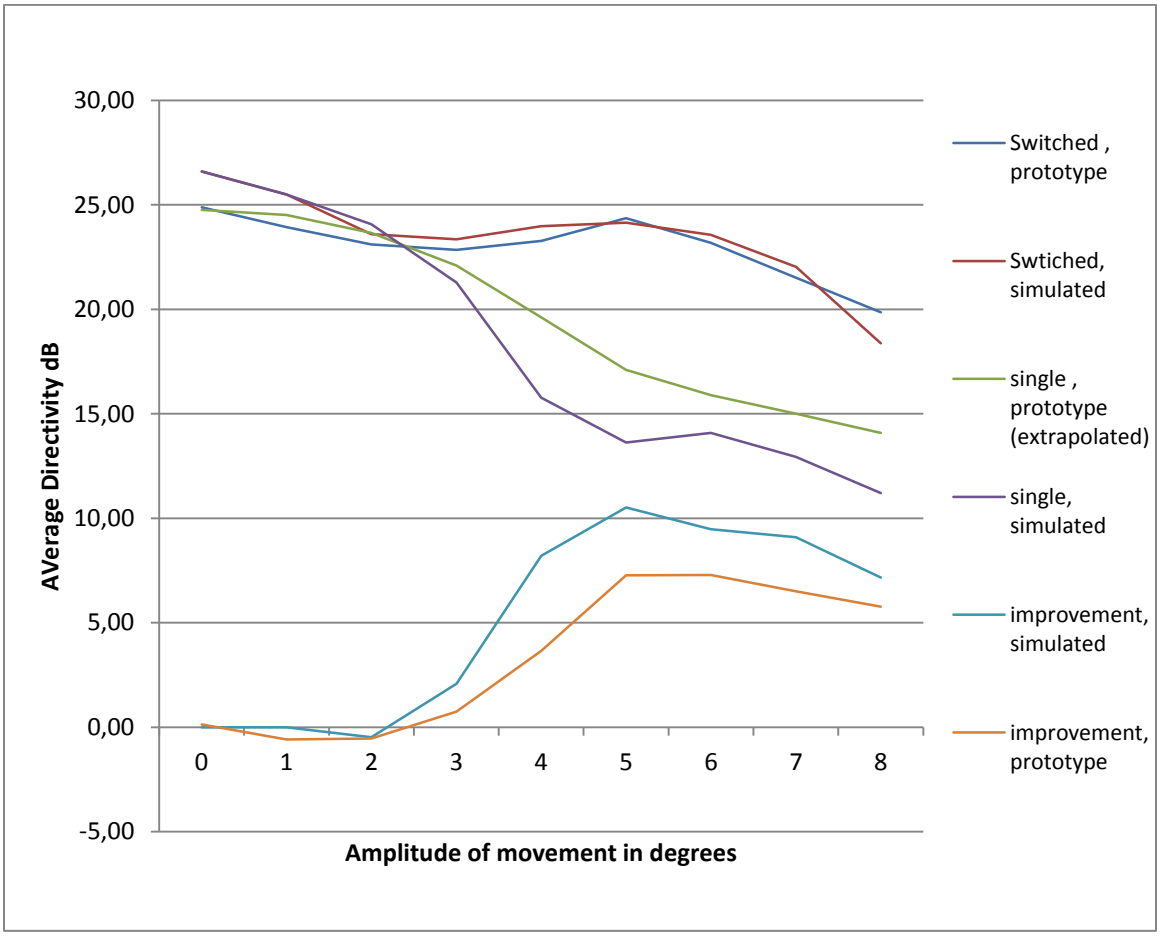


Figure 91 – Average directivity vs amplitude of movement

The curves display the same pattern as those shown in Figure 64. In addition, the match between the switched prototype/simulated average directivity is near exact. The large difference between single beam prototype/simulated average directivity is due to the deep nulls that exist only in the simulations. This caused the single beam simulated average directivity to decrease much faster than its counterpart, leading to a maximum directivity improvement of 7,25 dB for the tested prototype and 10,28 dB for the simulated 3SBOR.

7 CONCLUSION AND FUTURE WORK.

7.1 CONCLUSION

This thesis successfully confirmed the feasibility and potential performance of a self-stabilizing switched beam offset reflector. It concentrated on the problem in which an offset reflector antenna is positioned on the roof of a moving ground vehicle, with two assumptions:

- Large amplitude movements (i.e. steering, large incline) were compensated for by others means.
- The system was to correct the fast oscillations around the pitch and roll axes of the moving ground vehicle in question.

A movement model was determined in order to accurately calculate the angular variation and the orientation of that angular variation for the antenna positioned on a moving vehicle's rooftop.

Two variations of the 3SBOR were designed. The first one focused on solving the problem pertaining to the moving ground vehicle. The designed dish obtained a maximum directivity of 35 dB and a HPBW of 2.1º per beam and was fitted with a feed made of seven 10 dB horn antennas disposed in fashion to counter the oscillations of the moving ground vehicle. The simulations indicated that this design would improve the HPBW by 350% and provide a directivity improvement of up to 10 dB over a single feed offset reflector of same design, dimensions and situation.

A modified Alpha Beta filter control algorithm using variable sampling was devised to improve the response of the dish in a noisy environment (noisy with regards to the angular measurement of the sensor). Simulations showed that a 3SBOR with this control algorithm would achieve an average directivity up to 3.6 dB greater than that of a 3SBOR using untreated sensor data.

The second dish is a simplified version of the first one (single dimension parabolic reflector rather than two). This design was done to accommodate for the production constraints and to add validity to the simulations of the first dish. This reflector was parabolic along the vertical axis and was fitted with a feed composed of three 10 dB horn antennas disposed along the vertical axis. It used the same control algorithm as its counterpart. The maximum directivity of this antenna was 26 dB with a vertical (E-plane) HPBW of 2º. Simulations showed that the HPBW (E-plane) was improved by 300% and that it could provide a directivity improvement of up to 10 dB over a single feed offset reflector of same design.

A prototype of the single dimension parabolic reflector (dubbed 1D design) was built and tested to confirm the veracity of the simulations. The results were in line with the simulations and the concept was undeniably confirmed. The increase in HPBW was on par with the simulations and the directivity improvement was measured at 7.25 dB.

7.2 FUTURE WORK.

As this thesis successfully demonstrated the capability of the concept that is the 3SBOR, it did so in a manner to guarantee simplicity, ease of integration and success. The following items are to be considered for future work:

- Given the proper feed element, it would possible to redesign the dish with a much smaller footprint;
- A ground vehicle movement study would allow for improved control algorithm and optimized dish design. Furthermore, this would allow testing of the control algorithm in a noisy environment;
- Numerous factors of the control algorithm were “picked” as they allowed the system to operate as intended. Numerous optimizations are possible with regard to this matter;
- A full scope design (i.e 2D) has yet to be built and tested; and
- A live test on a moving ground vehicle should be performed. Watching Satellite TV in a moving vehicle could be a very good proof of concept.

To conclude, the 3SBOR could very well open the door to a variety of new applications where a high gain antenna, pencil beam is required but the instability of the moving platform prevents the use of such antennas.

References

[1] Constantine A. Balanis, Antenna Theory, Analysis and Design, Third edition, Wiley 2005, Chapter 13-15

[2] C.K Chui and G. Chen, Kalman Filtering With Real time Application, Third Edition, Springer 1998, Chapter 1-2

[3] Mohinder S. Grewal and Angus P. Andrews, Kalman Filtering, Theory and Practice, Prentice Hall 1993, Chapter 1-2

[4] Merill l. Skolnik, Introduction to Radar Systems, Third edition, McGraw Hill, 2001, Chapter 3 - 5

[5] Chester L. Parsons, Edward J. Walsh and Douglas C. Vandemark, Topographic Mapping Using a Multibeam Radar Altimeter. IEEE Transactions on Geosciences an Remote Sensing, November 1994

[6] Alan W. Rudge and Nurdin A. Adatia Offset-Parabolic-Reflector Antennas: A Review, Proceedings of the IEEE Vol. 66 No. 12, December 1978

[7] A.V. Shishlov, I.L. Vilenko and Yu. V. Kivosheev, Asymptotic Theory, Design and Efficiency of Array-Fed Reflector Antennas, Moscow Institute of Physics and Technology, IEEE, 2013

[8] Shung-Wu Lee and Yahya Rahmat-Samii, Comments on “Simple Formulas for Designing an Offset Multibeam Parabolic Reflector” IEEE Transactions on Antennas and Propagation, Vol AP-30 No.2, March 1982

[9] Quinghua Lai, Pei Li, Xiaopeng Lu, Chu Gao, A Prototype of Feed Subsystem for a Multiple-Beam Array Fed Reflector Antenna, Department of Microwave System, East China Research Institute of Electronic Engineering, IEEE 2015

[10] Shung-Wu Lee and Yahya Rahmat-Samii, Simple Formulas for Designing an Offset Multibeam Parabolic Reflector, IEEE Transactions On Antenna and Propagation Vol AP-29 No 3 May

[11] Lee S. Miller, Gary S. Brown and Lawrence and W. Choy, Multibeam Radar Altimetry: Spaceborne Feasibility, IEEE Transactions On Geosciences and Remote Sensing, Vol 29, No 3 May 1991

[12] Nelson J.G. Fonseca and Jacques Sombrin, Multi-Beam Reflector Antenna System Combining Beam Hoping and Size Reduction of Effectively Used Spot, IEEE Antennas and Propagation Magazine, Vol 54. No 2. April 2012

[13] Peter J.B. Clarricoats and Geoffrey T. Poulton, High-Efficiency Microwave Reflector Antennas- A review, Proceedings of IEEE, Vol 65 No. 10 October 1977

[14] Seunghwan Yoon, Matlab Reflector.m program Arizona State University, 12 June 2006

[15] 1981Ansys HFSS for Antenna Design, Workshop 8-1 Hybrid Simulation of a Reflector Antenna, 2015 release

[16] Ishar Pratap Singh, A High Precision Indoor Tracking System, Dalhousie University Halifax, Nova Scotia, July 2013

[17] DRS Technologies, Manportable Surveillance And Target Acquisition Radar (MSTAR, Specifications sheet www.leonardodrs.com/media/2245/mstar_v4_datasheet.pdf [Online; accessed 10 Sept 2016]

[18] Robert Q. Riley Enterprises Product design and development. Automobile Ride, Handling and Suspension Design with implications for low-Mass vehicle. http://www.rqriley.com/suspensn.htm [Online; accessed 8 July 2016]

[19] RF-Lambda, RFSP8TR0612G Reflective 6-12 Ghz Coaxial SP8T Switch Specifications Sheet, www.rflambda.com [online accessed December 2016]

[20] RF-Lambda, RFSP3TR0612G Reflective 6-12 Ghz Coaxial SP3T Switch Specifications Sheet, www.rflambda.com [online accessed December 2016]

[21] TE Connectivity sensor solutions, DOG2 MEMS-series Inclinometer Specifications Sheet, www.TE.com/sensorsolutions [online accessed December 2016]

[23] Pasternac, WR-90 Standard Gain Horn Antenna, Operates from 8.2 Ghz to 12.4 Ghz with a Nominal 10 dB gain SMA Female Input Connector Specifications Sheet www.pasternack.com [online accessed October 2016]

[24] Wikipedia, Alpha Bet Filter, https://en.wikipedia.org/wiki/Alpha_beta_filter

[25] Marco Borges, Ph. D. Student Computer/Biomedical Engineer, Alpha-beta filter for linear state and velocity estimation, https://www.mathworks.com/matlabcentral/fileexchange/42223-

alphabetafilter?focused=3797521&tab=function&requestedDomain=www.mathworks.com. Version V1, June 2013 [online accessed October 2016]

[26] Marco Borges, Ph. D. Student Computer/Biomedical Engineer, alpha-beta-gamma filter for linear state estimation of Velocity and Acceleration, https://www.mathworks.com/matlabcentral/fileexchange/43570-abgfilter?focused=3797493&tab=function Version V1, June 2013 [online accessed October 2016]

[27] Outline S.R.L., ET250-3D electronic turntable operation manual, 2015 www.outline.it

[28] ANSYS HFSS, Dish IE PO Help Files Version 2016.1

[29] Arduino help and tutorials located at https://www.arduino.cc/

A Appendix – Matlab code – 3SBOR

```
% algorithm for sensor tracking
% some modifications are required for high level and 1D simulations
% declaring variables required for simulation program.


dt = 0.01;
T = 0:dt: 10*3600*dt;
shift=0;
mean_roll= -1*pi/180;
mean_pitch=pi/12;
Ref_sig = zeros(length(T),2);
output_sig=zeros(length(T),2);
sample_size=3;
speed_sample=4;
roll_index=zeros(1,sample_size);
roll_speed_index=zeros(1,speed_sample);
pitch_index=zeros(1,sample_size);
pitch_speed_index=zeros(1,speed_sample);
roll_mean=0;
roll_speed_mean=0;
pitch_mean=0;
pitch_speed_mean=0;
Noisy_sensor_output = zeros(length(T),2);
noise=1*pi/180;
switch_time=0;
rho_index=zeros(1,100);
sample_total=zeros(length(T),1);
speed_size=zeros(length(T),1);
```

```
posn_size=zeros(length(T),1);

rho_mean=0;

rho_std=0;

ratio=3;

tune=0;

error_rho=zeros(length(T),1);

error_rho_ref=zeros(length(T),1);

error_Theta=zeros(length(T),1);

roll_sensor_index=zeros(1,1500);

pitch_sensor_index=zeros(1,1500);

roll_sensor_mean=0;

pitch_sensor_mean=0;

ratio_error_rho=0;

On_target_single_beam=0;

ratio_error_Theta=0;

On_target_3SBOR=0;

reject_angle=6;
```

```
%declaration of variables for antenna orientation calculation.


ha=2.8;

phi=pi/12;

rz=1.75;

ry=rz*tan(phi);

pz=1;

py= (rz-pz)*tan(phi);

theta_r=0;

theta=0;

Omega_r=0;

Omega=0;

rho_r=zeros(length(T),1);

rho=zeros(length(T),1);

rho_d=zeros(length(T),1);

rho_dr=zeros(length(T),1);

Theta_r=zeros(length(T),1);

Theta=zeros(length(T),1);

Theta_d=zeros(length(T),1);

Theta_dr=zeros(length(T),1);

Radii=(pz^2+(ha-py)^2)^.5;

Radii_anchor=(pz^2+(ry-py)^2)^.5;

Omega_2=atan(pz/(ha-py));

Omega_anchor=atan(pz/(ry-py));

roll_vector=zeros(1,3);

pitch_vector=zeros(1,3);

anchor_point=zeros(1,3);

antenna_position=zeros(1,3);
```

```
roll_vector_r=zeros(1,3);

pitch_vector_r=zeros(1,3);

anchor_point_r=zeros(1,3);

antenna_position_r=zeros(1,3);


sim=0 ;


%input the results files from HFSS


beam_0=xlsread('beams.xlsx',1,'A2:C72562');

beam_30=xlsread('beams.xlsx',2,'A2:C72562');

beam_90=xlsread('beams.xlsx',3,'A2:C72562');

beam_150=xlsread('beams.xlsx',4,'A2:C72562');

beam_210=xlsread('beams.xlsx',5,'A2:C72562');

beam_270=xlsread('beams.xlsx',6,'A2:C72562');

beam_330=xlsread('beams.xlsx',7,'A2:C72562');


ref_dir=zeros(length(T),1);

switched_dir=zeros(length(T),1);

improvement_db=zeros(length(T),1);

mean_ref_db=0;

mean_switched_db=0;

mean_improvement_db=0;
```

```
for t = 1:length(T)


%changing the random conditions


if sim==1
    if  t<(length(T)/4)
        noise=0.5*pi/180;
    else
        if  t>(3*length(T)/4)
            noise=0.1*pi/180;
        else
            if t>=(length(T)/4)&&t<(2*length(T)/4)
                noise=1*pi/180;
            else
                noise=0.3*pi/180;
            end
        end
    end
end


if sim==0
    noise=0.1*pi/180;
end
if sim==2
   noise=1*pi/180;
end
```

```
if sim==3
   noise=2*pi/180;
end


    sample_total(t,1)=sample_size+speed_sample;
    posn_size(t,1)=sample_size;
    speed_size(t,1)=speed_sample;


%creating a reference signal

    Ref_sig(t,1) = 0;
    Ref_sig(t,2) = sin(pi/2*T(t));


    Ref_sig(t,:)=2.1*Ref_sig(t,:)*pi/180;


%creating noisy measurement signal off the reference signal


    sensor=Ref_sig(t,:);
    sensor(1) = sensor(1) + mean_roll + noise*randn;
    sensor(2) = sensor(2) + mean_pitch +noise*randn;
    Noisy_sensor_output(t,:)=sensor;


%filling the roll sensor measurement index,mean


    if  roll_sensor_index(1,1)==0
        roll_sensor_index(1,1)=Noisy_sensor_output(t,1);
    else
        shift=length(roll_sensor_index);
```

```
        while shift>1
              roll_sensor_index(1,shift) =roll_sensor_index(1,shift-1);
              shift=shift-1;
        end
        if   abs(Noisy_sensor_output(t,1))>=reject_angle*pi/180
             roll_sensor_index(1,1)=roll_sensor_mean;
        else
             roll_sensor_index(1,1)= Noisy_sensor_output(t,1);
        end
        roll_sensor_mean=mean(roll_sensor_index);
    end

%filling the roll measurement index,mean

    if  roll_index(1,1)==0
        roll_index(1,1)=Noisy_sensor_output(t,1);
    else
        shift=sample_size;
        while shift>1
              roll_index(1,shift) =roll_index(1,shift-1);
              shift=shift-1;
        end
        if   t>length(roll_sensor_index)
             roll_index(1,1)= Noisy_sensor_output(t,1)-
             roll_sensor_mean;
        else
             roll_index(1,1)= Noisy_sensor_output(t,1);
        end
```

```
        i=1;
        while i<sample_size
              roll_mean=roll_index(1,i)+roll_mean;
              i=i+1;
        end
        roll_mean=roll_mean/sample_size;
    end

%filling the pitch sensor measurement index,mean

    if  pitch_sensor_index(1,1)==0
        pitch_sensor_index(1,1)=Noisy_sensor_output(t,2);
    else
        shift=length(pitch_sensor_index);
        while shift>1
              pitch_sensor_index(1,shift) =…
              pitch_sensor_index(1,shift-1);
              shift=shift-1;
        end
        if  abs(Noisy_sensor_output(t,2)-phi)>=reject_angle*pi/180
            pitch_sensor_index(1,1)=pitch_sensor_mean;
        else
            pitch_sensor_index(1,1)= Noisy_sensor_output(t,2);
        end
        pitch_sensor_mean=mean(pitch_sensor_index);
     end
```

```
%filling the pitch measurement index, mean


    if  pitch_index(1,1)==0
        pitch_index(1,1)=Noisy_sensor_output(t,2);
    else
        shift=sample_size;
        while shift>1
              pitch_index(1,shift) =pitch_index(1,shift-1);
              shift=shift-1;
        end
        if   t>length(pitch_sensor_index)
             pitch_index(1,1)=…
            Noisy_sensor_output(t,2)-pitch_sensor_mean;
        else
             pitch_index(1,1)= Noisy_sensor_output(t,2);
        end
        i=1;
        while i<sample_size
              pitch_mean=pitch_index(1,i)+pitch_mean;
              i=i+1;
        end
        pitch_mean=pitch_mean/sample_size;
    end
```

```
%fill the speed index for roll speed and roll speed mean


    if  roll_speed_index(1,1)==0
        roll_speed_index(1,1)=(roll_index(1,1)-roll_index(1,2));
    else
        shift=speed_sample;
        while shift>1
              roll_speed_index(1,shift) =roll_speed_index(1,shift-1);
              shift=shift-1;
        end
        roll_speed_index(1,1)=(roll_index(1,1)-roll_index(1,2));
        roll_speed_mean=mean(roll_speed_index);
        i=1;
        while i<speed_sample
              roll_speed_mean=roll_speed_index(1,i)+roll_speed_mean;
              i=i+1;
        end
        roll_speed_mean=roll_speed_mean/speed_sample;
    end


%fill the speed index for pitch speed and pitch speed mean


    if  pitch_speed_index(1,1)==0
        pitch_speed_index(1,1)=(pitch_index(1,1)-pitch_index(1,2));
    else
        shift=speed_sample;
        while shift>1
              pitch_speed_index(1,shift) =pitch_speed_index(1,shift-1);
```

```
            shift=shift-1;
        end
        pitch_speed_index(1,1)=(pitch_index(1,1)-pitch_index(1,2));
        i=1;
        while i<speed_sample
            pitch_speed_mean=pitch_speed_index(1,i)+pitch_speed_mean;
            i=i+1;
        end
        pitch_speed_mean=pitch_speed_mean/speed_sample;
    end


    output_sig(t,1)=(roll_mean(1,1))+(3/speed_sample)*...
                    roll_speed_mean(1,1);
    output_sig(t,2)=(pitch_mean(1,1))+(3/speed_sample)*...
                    pitch_speed_mean(1,1);


    theta=output_sig(t,1);
    Omega=output_sig(t,2);
    theta_r=Ref_sig(t,1);
    Omega_r=Ref_sig(t,2);
```

```
%begin calculus of antenna position
%antenna orientation calculus based off measured and signal

   roll_vector(1,3)=0.5*(ha-ry)*…
                   sin(-phi*(cos(theta-1)))*(cos(theta)-1);
   roll_vector(1,2)=(1/2*(cos(theta)-1))*(ha-ry)*...
                    cos(-phi*(cos(theta)-1))...
                    +((ha-ry)*(1/2))*cos(theta)+(ha+ry)*(1/2);
   roll_vector(1,1)=(ha-ry)*sin(theta)*cos(-phi*(cos(theta)-1));
   pitch_vector(1,3)=-Radii*sin(Omega_2-Omega)+pz;
   pitch_vector(1,2)=Radii*cos(Omega_2-Omega)+py-ha;

   anchor_point(1,3)=-Radii_anchor*sin(Omega_anchor-Omega)+pz;
   anchor_point(1,2)=Radii_anchor*cos(Omega_anchor-Omega)+py;
   antenna_position=roll_vector+pitch_vector-anchor_point;

   rho(t,1)=atan(((antenna_position(1,1)^2+...
             antenna_position(1,3)^2)^0.5)/...
            (antenna_position(1,2)))*(180/pi);
        if sign(Omega)>0
   Theta(t,1)= acos(antenna_position(1,1)/...
              ((antenna_position(1,1)^2+...
               antenna_position(1,3)^2)^0.5))*(180/pi);
        else
              Theta(t,1)= 360-(acos(antenna_position(1,1)/...
              ((antenna_position(1,1)^2+...
               antenna_position(1,3)^2)^0.5))*(180/pi));
        end
```

```
%antenna orientation based off reference signal

   roll_vector_r(1,3)=0.5*(ha-ry)*sin(-phi*(cos(theta_r-1)))*...
                      (cos(theta_r)-1);
   roll_vector_r(1,2)=(1/2*(cos(theta_r)-1))*(ha-ry)*...
                       cos(-phi*(cos(theta_r)-1))...
                       +((ha-ry)*(1/2))*cos(theta_r)+(ha+ry)*(1/2);
   roll_vector_r(1,1)=(ha-ry)*sin(theta_r)*cos(-phi*(cos(theta_r)-1

   pitch_vector_r(1,3)=-Radii*sin(Omega_2-Omega_r)+pz;
   pitch_vector_r(1,2)=Radii*cos(Omega_2-Omega_r)+py-ha;

   anchor_point_r(1,3)=-Radii_anchor*sin(Omega_anchor-Omega_r)+pz;
   anchor_point_r(1,2)=Radii_anchor*cos(Omega_anchor-Omega_r)+py;

   antenna_position_r=roll_vector_r+pitch_vector_r-anchor_point_r;

   rho_r(t,1)=(atan(((antenna_position_r(1,1)^2+...
               antenna_position_r(1,3)^2)^0.5)/...
              (antenna_position_r(1,2)))*(180/pi));
   if (sign(Omega)>=0)
       Theta_r(t,1)=acos(antenna_position_r(1,1)/...
                     ((antenna_position_r(1,1)^2+...
                     antenna_position_r(1,3)^2)^0.5))*(180/pi);
```

```
    else
        Theta_r(t,1)=360-(acos(antenna_position_r(1,1)/...
                    ((antenna_position_r(1,1)^2+...
                    antenna_position_r(1,3)^2)^0.5))*(180/pi));
    end
dgitial_rho_calculation


    if    rho(t,1)<=1.1
          rho_d(t,1)=0;
    else
          rho_d(t,1)=5;
    end


    if  t<2
    else
        if rho_d(t-1)~=rho_d(t)
           switch_time=switch_time+1;
        end
    end


    if    rho_r(t,1)<=1.1
          rho_dr(t,1)=0;
    else
          rho_dr(t,1)=5;
    end
```

```
%digital Theta calulations


  if  rho_d(t,1)==0
      Theta_d(t,1)=0;
  else
      if (Theta(t,1)>0)&&(Theta(t,1)<60)
          Theta_d(t,1)=30;
      end


      if (Theta(t,1)>60)&&(Theta(t,1)<=120)
          Theta_d(t,1)=90;
      end


      if (Theta(t,1)>120)&&(Theta(t,1)<180)
          Theta_d(t,1)=150;
      end


      if (Theta(t,1)>300)&&(Theta(t,1)<360)
         Theta_d(t,1)=330;
      end


      if (Theta(t,1)>=240)&&(Theta(t,1)<=300)
          Theta_d(t,1)=270;
      end
```

```
    if (Theta(t,1)>180)&&(Theta(t,1)<240)
       Theta_d(t,1)=210;
    end
end

if  rho_dr(t,1)==0
    Theta_dr(t,1)=0;
else
    if (Theta_r(t,1)>0)&&(Theta_r(t,1)<60)
        Theta_dr(t,1)=30;
    end

    if (Theta_r(t,1)>=60)&&(Theta_r(t,1)<=120)
        Theta_dr(t,1)=90;
    end

    if (Theta_r(t,1)>120)&&(Theta_r(t,1)<180)
        Theta_dr(t,1)=150;
    end

    if (Theta_r(t,1)>300)&&(Theta_r(t,1)<360)
       Theta_dr(t,1)=330;
    end

    if (Theta_r(t,1)>=240)&&(Theta_r(t,1)<=300)
        Theta_dr(t,1)=270;
    end
```

```
      if (Theta_r(t,1)>180)&&(Theta_r(t,1)<240)
         Theta_dr(t,1)=210;
      end
  end

%feedback loop, adjust sampling in function of level of noise in output
%filling a rho index for stochastic

  if  rho_index(1,1)==0;
      rho_index(1,1)=rho(t,1);
      rho_mean=mean(rho(t,1));
  else
      shift=length(rho_index);
      while shift>1
            rho_index(1,shift) =rho_index(1,shift-1);
            shift=shift-1;
      end
      rho_index(1,1)= rho(t,1);
      rho_mean=mean(rho_index);
      rho_std=std(rho_index);
  end
```

```
%adjustment of sampling size in function of output noise, based on
standard
%deviation of the output from short term output mean
  if  t<2*length(rho_index)
  else
      if  rho_index(1,1)>(abs(rho_mean)+ratio*rho_std)
          speed_sample=speed_sample+1;
          sample_size=sample_size+1;
          ratio=ratio+0.01;
      else
          tune=mod(tune+1,10*length(rho_index));
          if tune==0
             ratio=ratio-0.01;
             speed_sample=speed_sample-1;
             sample_size=sample_size-1;
          end
      end

      if sample_size<3
         sample_size=3;
      end

      if speed_sample<4
         speed_sample=4;
      end
```

```
    if sample_size>10
       sample_size=10;
    end

    if speed_sample>13
       speed_sample=13;
    end

    if ratio>3.5
       ratio=3.5;
    end

    if rho_index(1,1)>reject_angle
       rho(t,1)=rho_index(1,2);
       rho_index(1,1)=rho_index(1,2);
       roll_index(1,1)=roll_mean;
       pitch_index(1,1)=pitch_mean;
       roll_speed_index(1,1)=roll_speed_mean;
       speed_pitch_index(1,1)=pitch_speed_mean;
       speed_sample=speed_size(t-100,1);
       sample_size=posn_size(t-100,1);
       ratio=3;
    end
end
```

```
if t<length(roll_sensor_index)
else
    if (rho_r(t,1)>=1)&&(rho_d(t,1)==0)
        error_rho(t,1)=(abs(rho_dr(t,1)-rho_d(t,1)))/5;
        if (rho_r(t,1)<1)&&(rho_d(t,1)==5)
            error_rho(t,1)=(abs(rho_dr(t,1)-rho_d(t,1)))/5;
            if  (rho_r(t,1)>3.1)
                 error_rho(t,1)=1;
            else
                 error_rho(t,1)=0;
            end
        end
    end

    if  rho_r(t,1)>=1
        error_rho_ref(t,1)=1;
    else
        error_rho_ref(t,1)=0;
    end

    if  (Theta_dr(t,1)-Theta_d(t,1)~=0)&&(rho_r(t,1)>=1 )
         error_Theta(t,1)=1;
    else
         error_Theta(t,1)=0;
    end
end

%------------
```

```
   if t<length(roll_sensor_index)
  else
      i=1;
      while i<72562
          if
(beam_0(i,1)==round(Theta_r(t,1)))&&(beam_0(i,2)==round(rho_r(t,1),1))
              ref_dir(t,1)=beam_0(i,3);
          end


          if rho_d(t,1)==0
           if (beam_0(i,1)==round(Theta_r(t,1)))&&…
              (beam_0(i,2)==round(rho_r(t,1),1))
                switched_dir(t,1)=beam_0(i,3);
             end
          else
              if Theta_d(t,1)==30
                 if (beam_30(i,1)==round(Theta_r(t,1)))&&…
                    (beam_30(i,2)==round(rho_r(t,1),1))
                    switched_dir(t,1)=beam_30(i,3);
                 end
              end
              if Theta_d(t,1)==90
                 if (beam_90(i,1)==round(Theta_r(t,1)))&&…
                    (beam_90(i,2)==round(rho_r(t,1),1))
                    switched_dir(t,1)=beam_90(i,3);
                 end
              end
              if Theta_d(t,1)==150
```

```
        if(beam_150(i,1)==round(Theta_r(t,1)))&&…
          (beam_150(i,2)==round(rho_r(t,1),1))
           switched_dir(t,1)=beam_150(i,3);
        end
    end
    if Theta_d(t,1)==210
        if (beam_210(i,1)==round(Theta_r(t,1)))&&…
           (beam_210(i,2)==round(rho_r(t,1),1))
           switched_dir(t,1)=beam_210(i,3);
        end
    end
    if Theta_d(t,1)==270
        if (beam_270(i,1)==round(Theta_r(t,1)))&&…
           (beam_270(i,2)==round(rho_r(t,1),1))
           switched_dir(t,1)=beam_270(i,3);
        end
    end
    if Theta_d(t,1)==330
      if (beam_330(i,1)==round(Theta_r(t,1)))&&…
         (beam_330(i,2)==round(rho_r(t,1),1))
           switched_dir(t,1)=beam_330(i,3);
        end
    end

end
```

```
            improvement_db(t,1)=switched_dir(t,1)-ref_dir(t,1);
            i=i+1;
        end

    end

end

ratio_error_rho=((sum(error_rho(:,1))/t)*100);
ratio_error_Theta=(sum(error_Theta(:,1))/t)*100:
On_target_single_beam=100-(sum(error_rho_ref(:,1))/t)*100:
switch_time=switch_time*(50*10^-9)*100;
On_target_3SBOR=((((100-((ratio_error_Theta)))/100)*t-...
                switch_time/100)/t)*100;
mean_ref_db=mean(ref_dir)*length(T)/(length(T)-
length(roll_sensor_index));
mean_switched_db=mean(switched_dir)*length(T)/(length(T)-…
                 length(roll_sensor_index));
mean_improvement_db=mean(improvement_db)*length(T)/(length(T)-…
                    length(roll_sensor_index));
table(ratio_error_rho,ratio_error_Theta,On_target_3SBOR,...

On_target_single_beam,switch_time,mean_ref_db,mean_switched_db,mean_impr
ovement_db)
```

```
Figure('units','pixels','Position',[0 0 1600 900]);


subplot(2,3,1);
plot(T,Ref_sig(:,1),T,Noisy_sensor_output(:,1),T,output_sig(:,1));
xlabel('Time (s)'); title('Roll angle');
legend('Real','Sensor output','treated signal');


subplot(2,3,2);
plot(T,Ref_sig(:,2),T,Noisy_sensor_output(:,2),T,output_sig(:,2));
xlabel('Time (s)'); title('Pitch angle');
legend('Real','Sensor output','treated signal');


subplot(2,3,3);
plot(T,rho(:,1),T,rho_r(:,1),T,rho_d(:,1),T,rho_dr(:,1),T,sample_total(:
,1));
xlabel('Time (s)'); title('rho & sample size');
legend('treated','ref','digital','digital ref','sample size');


subplot(2,3,4);
plot(T,Theta(:,1),T,Theta_r(:,1),T,Theta_dr(:,1),T,Theta_d(:,1));
xlabel('Time (s)'); title('Theta');
legend('measured','ref','ref dr','ref d');


subplot(2,3,5)
plot(T,ref_dir(:,1));
xlabel('Time (s)'); title('Predicted Directivity');
legend('ref sig');
```

```
subplot(2,3,6);
plot(T,switched_dir(:,1));
xlabel('Time (s)'); title('Predicted Directivity');
legend('switched sig');
```

B Appendix – Arduino code – 1 D Prototype

```
#include <LiquidCrystal.h>
#include <Math.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

//declaring variables required for algorithm

int shift = 0;
int sensor_sample = 1500;
int sample_size = 3;
int speed_sample = 4;
int hold = 0;
int calibrated = 0;
float ratio = 3;
int   tune = 0;
float reject_angle = 5;

float roll_index[11];
float roll_speed_index[14];
float roll_mean = 0;
float roll_speed_mean = 0;
float roll_sensor_index[1500];
float roll_sensor_mean = 0;
float roll_calibrated = 0;

float pitch_index[20];
float pitch_speed_index[30];
```

```
float pitch_mean = 0;
float pitch_speed_mean = 0;
float pitch_sensor_index[1500];
float pitch_sensor_mean = 0;
float pitch_calibrated = 0;

float rho_index[100];
float rho_mean = 0;
float rho_std_temp = 0;
float rho_std = 0;

//declaring variables required for movement model

double pi = 3.141592;
double ha = 0.6017;
double phi = 0;
double rz = 0;
double ry = rz * tan(phi);
double pz = -0.0436;
double py = 0;
double theta = 0;
double omega = 0;
double radii = sqrt(sq(pz) + sq((ha - py)));
double radii_anchor = sqrt(sq(pz) + sq(ry - py));
double omega_2 = atan(pz / (ha - py));
double omega_anchor = atan(pz / (ry - py));
double roll_vector[3] = {0, 0, 0};
double pitch_vector[3] = {0, 0, 0};
double anchor_point[3] = {0, 0, 0};
```

```
double antenna_position[3] = {0, 0, 0};
double rho = 0;
double Theta = 0;
byte   rho_d = 0;
int   Theta_d = 0;
byte  C1 = 8;
byte  C2 = 10;
byte  C3 = 13;

// initiliaze the board and lcd

void setup() {

  Serial.begin(9600);
  analogReadResolution(12);
  lcd.begin(16, 2);
// initialize lcd display
  lcd.print("theta");
// display headers
  lcd.setCursor(0, 1);
  lcd.print("omega");
  delay(1000);
  pinMode(C1, OUTPUT);
  pinMode(C2, OUTPUT);
  pinMode(C3, OUTPUT);

}
```

```
// initiliaze the algorithm and lcd
void loop() {

  delay(10);
  float roll = analogRead(A0);
  float pitch = analogRead(A1);
  roll = (roll - 3120) / 107;
  pitch = (pitch - 3025) / 107;

  // This portion calibrate the sensor roll & pitch output to near zero.

  if (abs(roll) > reject_angle)
  {
    roll_index[0] = 0;
    roll_speed_index[0] = 0;
  }
  else
  {
    roll_sensor_index[0] = roll;
    roll_index[0] = roll - roll_sensor_mean;
    roll_speed_index[0] = (roll_index[1] - roll_index[2]);
  }
  if (abs(pitch) > reject_angle)
  {
    pitch_index[0] = 0;
    pitch_speed_index[0] = 0;
  }
```

```
 else
  {
    pitch_sensor_index[0] = pitch;
    pitch_index[0] = pitch - pitch_sensor_mean;
    pitch_speed_index[0] = pitch_index[1] - pitch_index[2];
  }

  shift = sensor_sample;
  while (shift > 0 && calibrated < 1500)
  {
    roll_sensor_index[shift] = roll_sensor_index[shift - 1];
[1499])
    pitch_sensor_index[shift] = pitch_sensor_index[shift - 1];
[1499])
    shift = shift - 1;
    roll_sensor_mean = roll_sensor_mean + roll_sensor_index[shift];
    pitch_sensor_mean = pitch_sensor_mean + pitch_sensor_index[shift];
  }
  if (calibrated < 1500)
  {
    roll_sensor_mean = roll_sensor_mean / sensor_sample;
    roll_calibrated = roll_sensor_mean / sensor_sample;
    pitch_sensor_mean = pitch_sensor_mean / sensor_sample;
    pitch_calibrated = pitch_sensor_mean / sensor_sample;
    calibrated++;
  }
```

```
// time to calculate the algorithm! position averages first

  shift = sample_size;
  while (shift > 0)
  {
    roll_index[shift] = roll_index[shift - 1];
[sample_size])
    pitch_index[shift] = pitch_index[shift - 1];
[sample_size])
    shift = shift - 1;
    roll_mean = roll_mean + roll_index[shift];
    pitch_mean = pitch_mean + pitch_index[shift];
  }
  roll_mean = roll_mean / sample_size;
  pitch_mean = pitch_mean / sample_size;

  // time to calculate the algorithm! speed averages now!

  shift = speed_sample;
  while (shift > 0)
  {
    roll_speed_index[shift] = roll_speed_index[shift - 1];
    pitch_speed_index[shift] = pitch_speed_index[shift - 1];
    shift = shift - 1;
    roll_speed_mean = roll_speed_mean + roll_speed_index[shift];
    pitch_speed_mean = pitch_speed_mean + pitch_speed_index[shift];
  }
  roll_speed_mean = (roll_speed_mean / speed_sample);
  pitch_speed_mean = (pitch_speed_mean / speed_sample);
```

```
theta = roll_mean + (3 / speed_sample) * roll_speed_mean;
omega = pitch_mean + (3 / speed_sample) * pitch_speed_mean;
theta = theta * pi / 180;
omega = omega * pi / 180;

//time for model movement calculus madness!!!
roll_vector[2] = 0.5 * (ha - ry) * sin(-phi * (cos(theta - 1))) *
                 (cos(theta) - 1);  //z component due to roll
roll_vector[1] = (0.5 * (cos(theta) - 1)) * (ha - ry) * cos(-phi *
                 (cos(theta) - 1)) + ((ha - ry) * (0.5)) * cos(theta)
                 + (ha + ry) * (0.5);
roll_vector[0] = (ha - ry) * sin(theta) * cos(-phi * (cos(theta) -1));
pitch_vector[2] = -radii * sin(omega_2 - omega) + pz;
pitch_vector[1] = radii * cos(omega_2 - omega) + py - ha;
anchor_point[2] = -radii_anchor * sin(omega_anchor - omega) + pz;
anchor_point[1] = radii_anchor * cos(omega_anchor - omega) + py;
antenna_position[0] = roll_vector[0];
antenna_position[1] = roll_vector[1] + pitch_vector[1] -
                      anchor_point[1];
antenna_position[2] = roll_vector[2] + pitch_vector[2]
                      anchor_point[2];
rho = (atan((sqrt(sq(antenna_position[0]) + sq(antenna_position[2])))/
      (antenna_position[1]))) * 180 / pi;
if (omega >= 0)
{
  Theta = (acos(antenna_position[0] / (sqrt(sq(antenna_position[0]) +
          sq(antenna_position[2]))))) * (180 / pi);
}
```

```
  else
  {
    Theta = 360-(acos(antenna_position[0]/ (sqrt(sq(antenna_position[0])
            + sq(antenna_position[2]))))) * (180 / pi);
  }

  // time to write the rho based feedback function. first let's fill the
rho index

  if (abs(rho) > reject_angle)
  {
    rho_index[0] = 0;
    rho = rho_index[99];
    speed_sample = 4;
    sample_size = 3;
    ratio = 3;
  }
  else
  {
    rho_index[0] = rho;
  }
  shift = 100;
  while (shift > 0)
  {
    rho_index[shift] = rho_index[shift - 1];
    shift = shift - 1;
    rho_mean = rho_mean + rho_index[shift];
  }
  rho_mean = rho_mean / 100;
```

```
shift = 100;
while (shift > 0)
{
  rho_std_temp = sq(rho_index[shift] - rho_mean);
  rho_std = rho_std + rho_std_temp;
  shift = shift - 1;
}
rho_std = sqrt(rho_std / 100);
if (hold < 1500);
else
{
  if (rho > (abs(rho_mean) + ratio * rho_std))
  {
    sample_size++;
    speed_sample++;
    ratio = ratio + 0.01;
    if (sample_size > 10)  sample_size = 10;
    if (speed_sample > 13) speed_sample = 13;
    if (ratio >= 3.15)  ratio = 3.15;

  }
  if (tune > 1000)
  {
    tune = 0;
    sample_size--;
    speed_sample--;
    ratio = ratio - 0.01;
    if (sample_size < 3)   sample_size = 3;
    if (speed_sample < 4)  speed_sample = 4;
```

```
    if (ratio < 3) ratio = 3;
  }
}
if (hold < 1501)hold++;
tune++;
if (rho > 1)
{
  rho_d = 1;
  if (Theta > 0 && Theta <= 180)
  {
    Theta_d = 1;
    digitalWrite(C1, HIGH);
    digitalWrite(C2, HIGH);
    digitalWrite(C3, LOW);
  }
  if (Theta > 180 && Theta < 360)
  {
    Theta_d = 2;
    digitalWrite(C1, LOW);
    digitalWrite(C2, HIGH);
    digitalWrite(C3, HIGH);
  }
}
else
{
  rho_d = 0;
  Theta_d = 0;
  digitalWrite(C1, HIGH);
  digitalWrite(C2, LOW);
```

```
    digitalWrite(C3, HIGH);
  }
// feedback function as per matlab code.

    Serial.println(Theta_d, 'DEC');
    lcd.setCursor(6, 0);
    lcd.print(theta * 180 / pi, 2);
    lcd.setCursor(6, 1);
    lcd.print(omega * 180 / pi, 2);
    lcd.setCursor(13, 0);
    lcd.print(sample_size, 'DEC');
    lcd.setCursor(13, 1);
    lcd.print(speed_sample, 'DEC');
    lcd.setCursor(15, 1);
    lcd.print(ratio);
    lcd.setCursor(15, 0);
    lcd.print(rho_d);

    }
```