Design and Development of Ultra Wide-Band Microwave Components for Snow-Probing Radars

By

Renish Thomas

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Chairperson: Dr. Carlton Leuschen

_____

Co-Chairperson: Dr. Fernando Rodriguez Morales

_____

Committee member: Dr. Rongqing Hui

Date Defended: 10 August 2018

The thesis committee for Renish Thomas

certifies that this is the approved version of the following thesis:

Design and Development of Ultra Wide-Band Microwave Components for Snow-Probing Radars

_____

Chair: Dr. Carl Leuschen

_____

Co-Chair: Dr. Fernando Rodriguez Morales

Date Approved: 10 August 2018

Abstract

This thesis describes the design and development of two different ultra-wideband microwave circuits for snow-probing radars. First, a broadband, low-loss planar quadrature hybrid coupler for the 2-20 GHz range is presented. The coupler offers better performance than commercially available options in terms of phase/amplitude imbalance and form factor. Next, a broadband, high-power T/R module with fast switching and integrated LNA is demonstrated to enable high altitude and multi-channel modes of operations of the CReSIS airborne snow radar along with automated surface tracking ability. The modules include a custom medium-power switch with an overall order of magnitude performance increase compared to commercially available duplexers/SPDT switch solutions.

Pulsed mode operations at peak power levels exceeding 100 Watts (conservatively) can be supported with these devices and a demonstrated switching speed of less than 400 ns.

Acknowledgements

I thank the Almighty for everything in my life, both in the good and the bad times and for having the best plans for me. "Seek the Lord and he shall make your paths straight".

I thank Dr. Carl Leuschen for providing me an opportunity to work as a Graduate research assistant at CReSIS. This was one of the best things to ever happen to me. This place truly molded me as an engineer. I sincerely thank Dr. Fernando Rodriguez Morales for being an exceptional teacher, a guide and an amazing adviser. I might have bothered him with too many questions sometimes, but he always took the time to explain in spite of his busy schedule.

I thank Paulette Place for teaching me practical skills in soldering and reworking and also for being like a mother figure to everyone in the lab. I thank Aaron Paden for all the machining help and also for being patient and considerate when having to do repeated design modifications. I thank Jennifer Laverentz for helping out with the logistics during field deployments and for all administrative support. I sincerely thank Teja, Hara, Shravan, Calen and other CReSIS colleagues who helped out with this project when I was deployed on field.

I thank NASA OIB (Operation Ice Bridge) which funded this project and also for the amazing field deployments I had with the NASA P-3 and DC-8 polar expeditions. I specially want to thank Bryan Townley for teaching me a lot regarding radar installations on airborne platforms.

Lastly I want to thank my parents Licy Thomas and Thomas Kutty for bringing me up and guiding me up in the right ways. I also want to thank my brother Anish Thomas for his support.

Table of Contents

List of Figures/Tables

Chapter 1: Introduction

1.1   Background

Snow cover over land ice and on the surrounding sea ice are important components of the cryosphere. Snow cover acts as an insulator for the ice underneath and protects it from excessive melt. Dry snow reflects a large portion of the incident sun rays back into space due to the albedo effect and hence helps regulate the climate. Snow fall adds to the ice mass on land ice and contributes to thickening of the ice sheets over time and compensates for the melting of ice sheets which is a major contributor to sea level rise. Also snow melt water accounts for a large proportion of accessible fresh water for human consumption and is a vital target to study and understand to better manage limited water resources. Several remote sensing techniques are employed for the periodic and systematic monitoring of snow including ground, airborne and space borne sensors. At the Center for Remote Sensing of Ice Sheets (CReSIS) at the University of Kansas, we have developed various renditions of ultra-wideband frequency modulated continuous wave (UWB-FMCW) radars for snow studies [1, 2]. These systems, referred to as "Snow radars" have been deployed onboard multiple airborne platforms and used to quantify the snow cover thickness and seasonal snow accumulation in the Arctic and Antarctic. Other groups around the world have developed microwave systems for the same application, including for helicopter based snow pack measurements and georadar [3, 4].   An example of the snow cover echogram data product produced using the snow radar over the Pine island glacier is shown in Figure 1.1
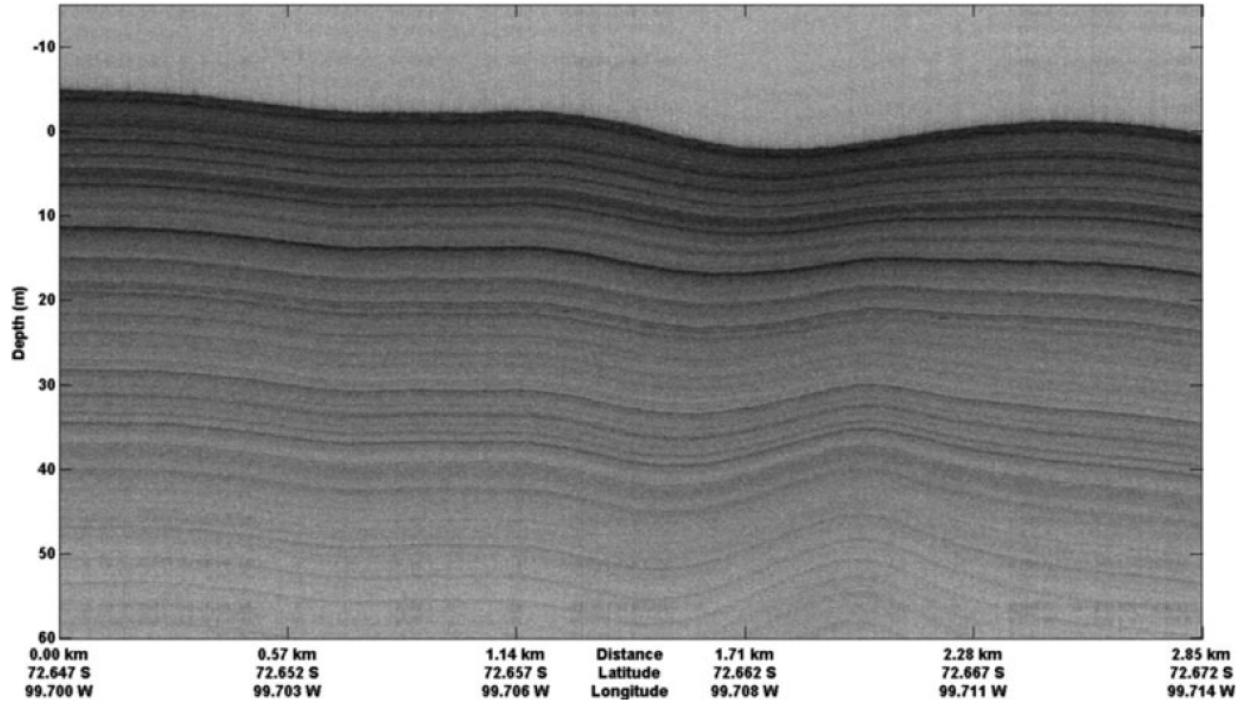
Figure 1.1: Image of the snow layering echogram from the snow radar over the Pine island glacier[1].

1.2 Motivation

The current configurations of the CReSIS airborne snow radar operating over a 2-18 GHz bandwidth utilizes a dual antenna setup, in which one element is dedicated for transmit (Tx) and one for receive (Rx). The antennas are typically attached to the airframe and separated by a fixed distance set by the minimum isolation requirement in order to reduce short-range leakage (Figure 1.2.1.a). Short-range leakage caused by the direct coupling between the antennas is a well-known limiting factor in the performance of FMCW systems[5] .Besides introducing physical separation, metal enclosures lined with microwave absorbing material are often used to minimize reflections in the setup and to further reduce coupling [5, 6]. Furthermore, radars operating in this configuration use long chirp pulses and the de-chirping of the received signal is done in hardware. For long-range operations, this can result in a beat frequency signal that falls outside the operating range of the digitizer.

In contrast to the above configuration, if the Tx and Rx channels share the same antenna (Figure 1.2.1.b) it would be possible to have the transmitter disabled during receive. Such a configuration requires the use of a radar duplexer device or transmit/receive switch. The radar duplexer allows for short chirp pulses be used for transmit mode and the de-chirping of the receive signal can be accomplished by mixing it with delayed copy of the transmit signal. This allows the IF signal to be within the frequency range of the digitizer. Such an approach is particularly advantageous for high-altitude operations. This thesis thus addresses the need of a medium-power T/R switch for UWB microwave radar, capable of fast switching and low-loss. To reach the implementation of the switch, a set of custom UWB couplers were designed, fabricated and characterized. As will be discussed in the ensuing sections, the development of the switch not only addresses challenges related to antenna location, transmit power and high altitude operation but has the potential to enable added capabilities not possible in the dual antenna setup. Applications other than snow sensing can benefit from the developments presented here.
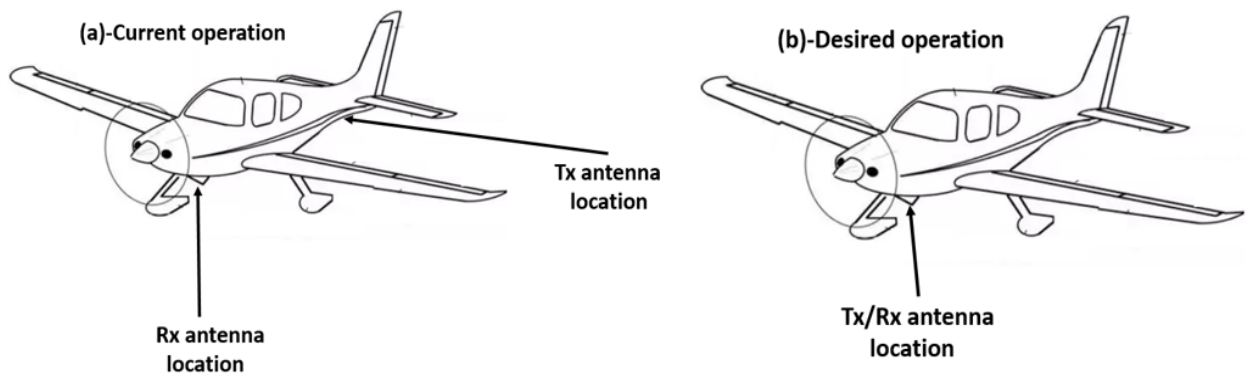
Figure 1.2.1: Illustration of the typical transmit and receive antenna arrangement for the snow radar on an airborne platform. (b): Co-located transmit and receive antenna arrangement enabled by a T/R switch.

1.3 Advantages of the single antenna configuration

The airborne platform typically flies at a set altitude AGL (above ground level) while probing the snow cover with nadir-looking antennas. During the transmission of signals, the duplexer device should provide for a low loss path from the transmit port to antenna port while also providing good isolation between the transmitter and the receiver as to not cause damage to the sensitive receiver electronics such as the LNA (low noise amplifier). During reception of return signals from the target, the duplexer should provide for a low loss path from the antenna port to the receiver port, while also isolating the transmitter and the receiver to prevent possible damage to the receiver chain. Ideally the power amplifier at the transmission port should remain turned off in the receive mode. In the case of non-pulsed amplifiers, the isolation during transmit will ensure that no noise from the power amplifier is coupled to the receiver.

A graphic illustration of the operation of the radar in single antenna configuration is shown in Figure 1.3.1, along with the basic radar-duplexer block and timing diagrams. The lowest operational altitude of the science flights sets the minimum switching speed of the duplexer device. A typical low altitude for the snow probing science flight is about 500 meters AGL. To account for the blind range and other considerations at this altitude the duplexer must be able to switch within 2μs for a 1μs short pulse. A control signal from the radar switches between the duplexer transmit and receive modes. The duplexer transmits when the control is high and receives when control is low. The control signals are typical digital logic voltage of 5 volts for high and 0 volts for low. The turn on time 't_1'is defined and measured as the time when the RF output power at the transmitter port is 90% from the duplexer control 'ON' signal. The turn off time 't_2'is defined and measured as the time when the RF output power at the receiver port is 90% from the duplexer control 'OFF' signal. Ideally the turn on time has to be low for the radar transmission to have a

low wait time from when the control signal turns 'ON' and the turn 'OFF' time has to be low to minimize the blind range distance and to enable target detection from a closer range. These times are determined by the diode carrier lifetimes (τ) [7], and a safe assumption is that the switching can be done in $\tau/10$.
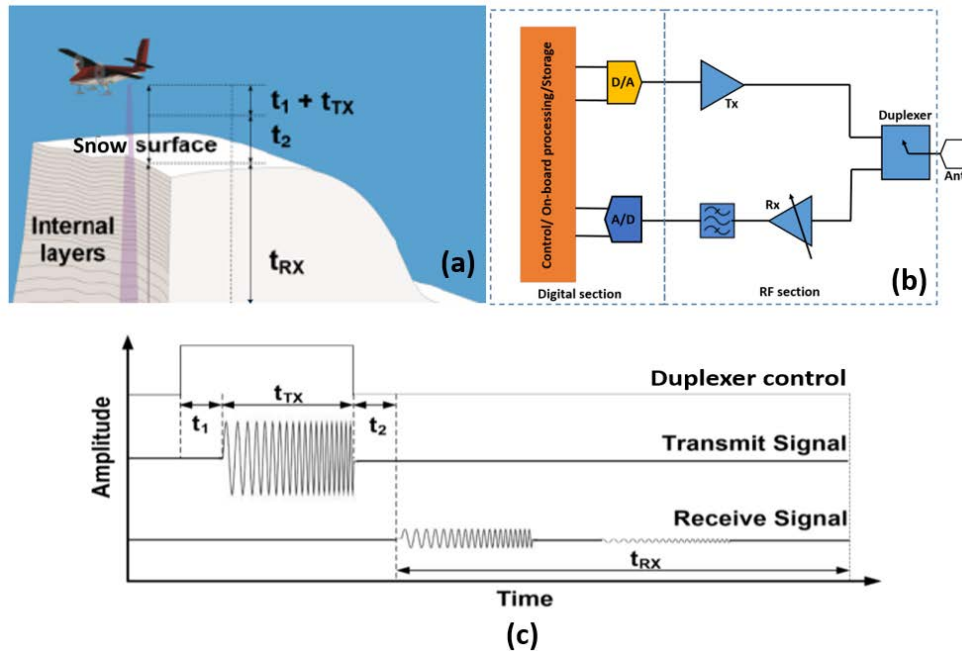
Figure 1.3.1: (a): Illustration of the airborne snow radar with the duplexer switching time shown related to radar pulse propagation from antenna to snow target and back. (b): Simple single channel block diagram of the snow radar system with the duplexer. (c): Transmit and receive timing diagram of the duplexed snow radar system (adapted from [8]).

Aside from eliminating the antenna isolation challenges and enabling the capability of a higher power to be transmitted, the switch may allow for multiple channels to be used in an array configuration. In Figure 1.3.1.1, each channel can have its own power amplifier thereby increasing the transmit power and improving SNR and detection of weaker targets (for example, deeper layers of snow and firn)
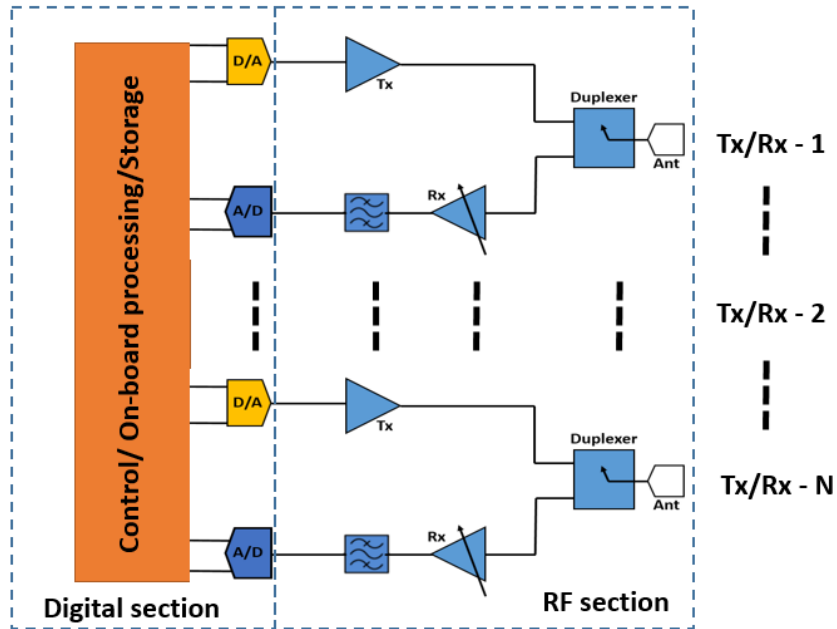
Figure 1.3.1.1: Block diagram of a configuration with multiple transmit-receive / duplexer modules to excite an antenna array.

Multiple antenna elements could be used as an array with the common antenna element mode enabled by the duplexer, which enables array processing techniques. The short pulsed mode which is enabled by the use of the radar duplexer, which can result in an increase of the PRF (pulse repetition frequency) and thereby SNR improvements as more coherent averaging can be done. Moreover, the radar duplexer can enable operations across a wider array of airborne platforms including UAV (unmanned aerial vehicles) also can be integrated on current platforms such as the NASA P-3 and DC-8 (Figure 1.3.4.1)
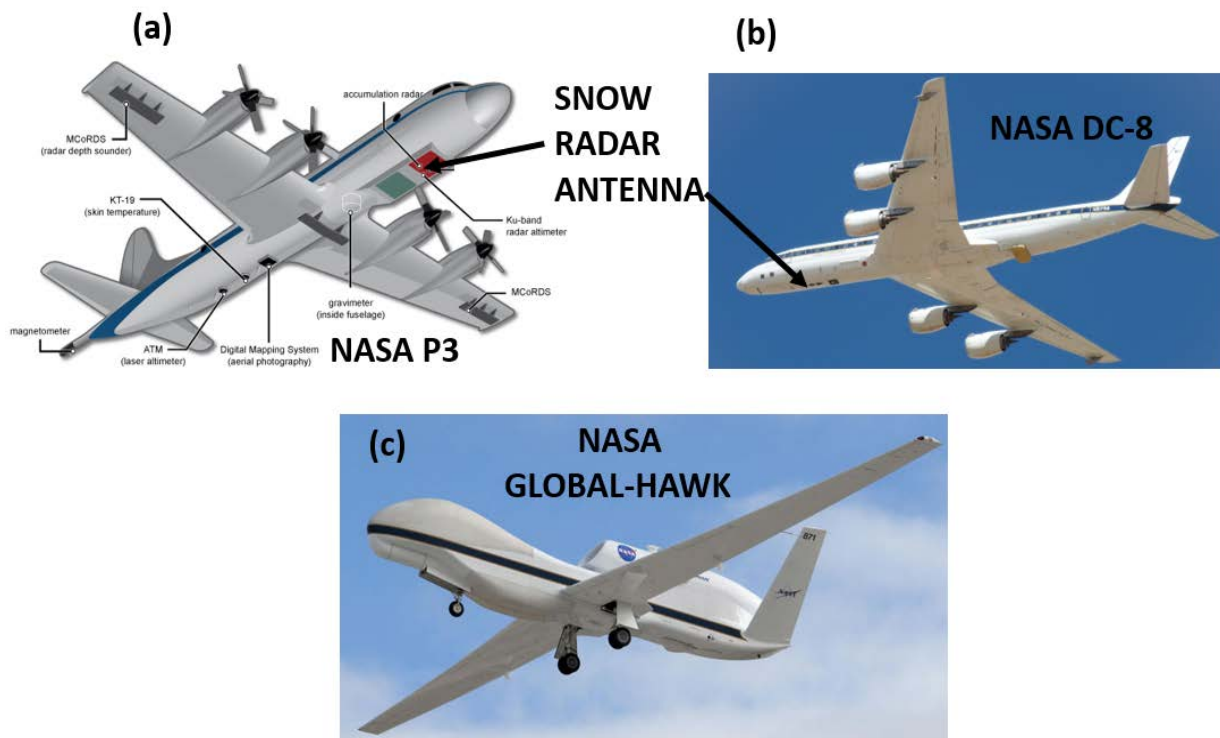
Figure 1.3.4.1: (a): NASA P-3 Airborne science laboratory platform used typically for Greenland survey. (b): NASA DC-8 Airborne science laboratory platform used typically for Antarctic survey. (c): NASA Global hawk Airborne science platform for potential use of the snow radar for unmanned and long duration surveys. Picture credits: NASA Goddard.

The co-locating of the Tx/Rx paths enables the RF section to be right at the antenna port which can significantly improve the noise figure, cut down coaxial cable lengths thus the insertion loss of the system, among other benefits. Also down-conversion of the received signal to IF band can be done right at the antenna feed point which enables lower frequency signals to be sent to the digitizer section thus eliminating the need for expensive high frequency coaxial cables to be used.

1.4   Duplexer solutions

There are several possible radar duplexer solutions available to date. Examples include circulators, electro-mechanical switches, PIN diode SPDT switches, COTS duplexers/SPDT switches. To

7

understand the drawbacks and advantages of each implementation, we present a brief summary in the ensuing sections.

1.4.1 Circulators as duplexers

Earlier radar systems and some recent ones use microwave circulators as duplexer devices (Figure 1.4.1.1). These devices do not have to switch and are instantaneous as the signals are discerned by the port it arrives from and thus eliminates the problem of blind ranges. While blind range may not be an issue, there are other severe disadvantages of using a circulator as the duplexer for our application.
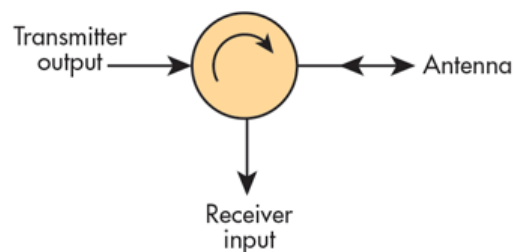
Figure 1.4.1.1: Microwave circulator as a radar duplexer device

Pros: No switching is required for duplexing and hence does not have blind range issues. Cheap and generally simplistic to implement. No mechanical moving parts.

Cons: Ultra-wide band circulators are not available. Poor isolation from transmitter to receiver, and hence the likely possibility of receiver blinding and damage. Poor return loss in wide band designs and can cause reflections at the antenna port and the transmitted energy to reflect back to the receiver port and degrades effective transmitter-receiver isolation. Bulky devices as it is mostly ferrite based. Not suitable for high powered applications. Insertion loss is significant in high frequency applications.

1.4.2 Electro-mechanical SPDT switches as duplexers

Electro-mechanical SPDT (Single pole double throw) switches gives superior performance in terms of isolation, return loss and bandwidth as the switch works literally by breaking and making connections using mechanical actuators. But unfortunately these devices cannot be used in our application as the very nature of mechanical switching process requires unacceptably high switching times of about 20-ms, which means the blind range is in the thousands of miles and unusable.

Pros: Very low insertion loss up until 50 GHz range or more. Extremely good isolation is possible between the transmitter and receiver. Return loss is exceptionally good and the switch operates right from DC to 50 GHz or more. Cheap compared to other solutions. Suitable for medium/high power applications.

Cons: Switching time is about 20-ms, rendering the switch useless for radar duplexer application. The switch is prone to electric arcing problems when an energized inductor path is abruptly broken. Mechanical actuator parts have a maximum of about 1 million switching cycles and wears out quick.

1.4.3 PIN diode based SPDT switches as duplexers

PIN (positive intrinsic negative) diodes are widely used as RF switch elements in series and shunt diode configurations[9] . These have fast switching times of a few nanoseconds and are fairly easy to design and implement. Two kinds of PIN diode SPDT switch architectures exist, the series PIN diode connected SPDT switch (Figure 1.4.3.a) and the shunt PIN diode connected SPDT switch (Figure 1.4.3.1.b). Even though these are solid state devices and fast switching, they cannot be used in our application as the cons of each of the SPDT switch architecture shows.
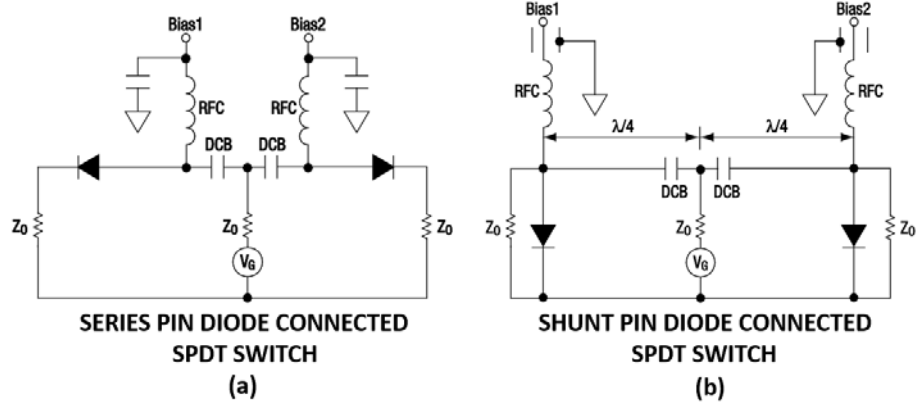
Figure 1.4.3.1:(a) Series PIN diode connected SPDT switch. (b) Shunt PIN diode connected SPDT switch [9].

Series PIN diode connected SPDT switch pros: Faster switching time compared to shunt PIN diode connected case. Wide bandwidth achievable. Solid state switch and the mean time between failures is small.

Series PIN diode connected SPDT switch cons: Low isolation between transmit and receive ports as series diode essentially behaves as a lossy capacitor in reverse bias. Low power handling capacity and poor heat sinking ability. Large negative voltages are required for biasing. Resonant free RF chokes and wide band DC blocks are needed.

Shunt PIN diode connected SPDT switch pros: Good isolation is achievable between the transmit and receive ports. Heat sinking is good as shunt diodes are heat sinked at one end. Solid state switch and the mean time between failures is small. Higher power handling than the series diode connected case.

Shunt PIN diode connected SPDT switch cons: The transmission line between the connected shunt diode arms needs to be spaced at half a wavelength apart at the operating frequency to achieve a proper match and this prevents wideband application of this switch architecture. Large

negative voltages are required for reverse biasing the shunt diodes. Resonant free RF chokes and wide band DC blocks are needed.

1.4.4 COTS SPDT switches as duplexers

As we have looked through potential duplexer solutions and found that none satisfy our application needs properly, we also look at one of the best performing SPDT switch found on the market. The best one out of an overall performance comparison is found to be the Triquint TGS2353-2 (Figure 1.4.4.1) which is a marketed as a 0.5 – 18 GHz high power SPDT Switch and is based on an MMIC (Monolithic Microwave Integrated circuit) GaN on SiC chip process [10]. Let us go through the pros and cons of this SPDT switch.

Pros: Relatively fast switching switch with a speed of about 45 ns. Covers our wide bandwidth application. Small form factor.

Cons: Large external negative voltage of -40 Volts is required to switch which causes the switching speed to be limited by the speed of the switching supply and hence is impractical. Only supports 10 Watts continuous incident power.
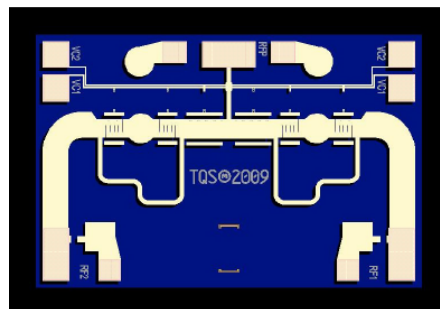
Figure 1.4.4.1: Triquint TGS2353-2 MMIC SPDT switch [10]

1.5   T/R switch based on a balanced duplexer design as the duplexer solution.

For this work we consider a radar duplexer solution based on a balanced configuration transmit receive switch design proposed by Cofrancesco et al [11] which was previously implemented by the CReSIS group for various radar depth sounders of VHF/UHF/HF frequency bands to image the Arctic and Antarctic ice sheets and to map the underlying topography [12]. This solution seemed the most suitable for our application, based on its power handling capability and relatively fast switching speeds. A summary of the different duplexer solutions available is shown in Table 1.5.1

Duplexer type	Switching speed/ Biasing supply voltages	Insertion loss	Isolation	Bandwidth	Power handling
Circulators	N/A / NA	Poor at high frequencies	Poor isolation over wide bandwidths	Narrow	Low/moderate at higher frequencies
Electro-mechanical SPDT	Extremely slow 20-ms or more / Typically 24 volt for switching	Excellent compared to other options	Excellent isolation over wide bandwidths	Very wide (DC-50 GHz or more)	Moderate/high powered applications

Series PIN diode SPDT	Fast 20-ns/ High negative voltage required for high power applications.	Moderate at high frequencies	Moderately low isolation over wide bandwidths	Wide bandwidth possible	Low powered applications
Shunt PIN diode SPDT	Fast 20-ns	Moderate at high frequencies	Moderately high isolation over wide bandwidths	Inherently narrow-band	Moderate powered applications
COTS Qorvo TGS2353-2 SPDT switch	Fast 4-ns	Moderate (1.5-dB)	Moderate (30dB)	Wide bandwidth 0.5-18GHz	10 Watts (CW) max
Balanced duplexer based TR switch	Moderately Fast 500-ns achievable	Moderate (1.5-3 dB)	Moderately high (30-50 dB)	High bandwidth possible if carefully implemented	High power rating of 100 Watts pulsed

Table 1.5.1: Summary of the different duplexer technologies

1.6 This thesis work

As mentioned above, this thesis work deals with developing a solid state ultra-wideband T/R switch based on a balanced duplexer design that employs quadrature hybrid couplers as the splitting and combining components and actively biased wideband shunt PIN diodes as the switch elements. First, a set of custom quadrature hybrid couplers are designed by using optimization and

dimension generation algorithms. The designed quadrature hybrid couplers are then fabricated and tested. Improvements are made in the fabricated couplers in the insertion loss characteristics by analyzing the cause of the loss and then using superior materials to compensate. Next, a T/R switch with an integrated LNA is designed and assembled to meet design requirements. Analysis of biasing requirements and power handling was done to find the best architecture and components to implement the balanced duplexer design. The duplexer was assembled and tested for switching, loss, isolation performance.

1.7   Thesis outline

This thesis document contains six chapters. Chapter 2 deals with the operation of the balanced duplexer and initial trade studies to demonstrate the feasibility of the design. Chapter 3 presents the design of the custom low-loss planar quadrature hybrid couplers while Chapter 4 is concerned with the physical implementation of the quadrature hybrid couplers. Chapter 5 delves into the T/R switch component selection and assembling details along with low- and high-level simulations and measured performance of the switch.

Chapter 2: Proof of concept and design requirements of the T/R switch/duplexer.

This chapter describes the working principle of the balanced duplexer configuration based T/R switch and goes through the previous implementations of the switch to radar and nuclear magnetic resonance (NMR) applications. We did perform some mock up tests with commercial quadrature hybrid couplers to verify if the balanced duplexer based T/R switch could be implemented at our

high frequency and wide bandwidth application. The design requirements for the TR switch is also

given.

2.1   Working principle of the balanced duplexer T/R switch

The working principle of the T/R switch based on a balanced configuration is illustrated in Figure

2.1.1. The circuit employs two identical hybrid couplers (denoted A and B) and a set of actively

biased shunt PIN diodes as the switching elements (denoted D1-D3). In the transmit mode of

operation, the quadrature hybrid coupler A splits the signal from the power amplifier into equal-

power paths with $0°$ and $90°$ phase, respectively. During this state, the diode D1 is reverse biased,

providing a low impedance path from the power amplifier signal. Concurrently, the diodes D2 and

D3 are forward biased, which provides a reflective impedance to the signals at the output of the

hybrid coupler A. These signals are reflected back to the coupler, and interfere constructively and

destructively at the isolation and summation ports, respectively. The signal from the power

amplifier is thus injected into the antenna.
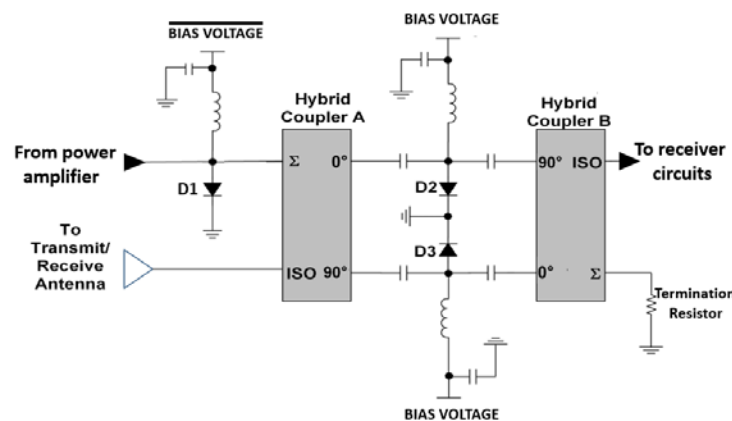
Figure 2.1.1: Simplified block diagram of the T/R switch based on the balanced duplexer configuration using shunt PIN diode switching elements

15

In the receive mode of operations the diode D1 is forward biased which provides a reflective impedance to incoming signals. Diodes D2 and D3 are reverse biased and provides a capacitive impedance to incoming signals. The signal from the antenna enters the isolation port of the quadrature hybrid coupler A and is split with 0° and 90° phase respectively which passes into the ports of the quadrature hybrid coupler B, where the signals interfere constructively at the isolation port and destructively at the summation port. A low-loss path between antenna and receiver is thus achieved. Any residual power caused due to phase and amplitude imbalance is dissipated into the termination resistor. The shunt diode D1 provides a reflective impedance to isolate the transmitter and receiver ports effectively and turns the T/R switch into a reflective SPDT switch. The DC blocks are provided to isolate the receiver, transmit and antenna ports from applied bias voltages. The RF chokes provides a way for the DC bias currents while blocking the AC currents.

2.2 Previous T/R switch implementations based on the balanced duplexer design

The use of this configuration at lower frequencies is well documented. The earliest implementation of the balanced duplexer design was used for NMR (Nuclear magnetic resonance) imaging applications from 90-500 MHz with a power handling of 500 Watts [11]. This device had 1.3-dB insertion in receive mode and 7 dB insertion loss in transmit mode. An atmospheric radar implementation of the T/R switch by Agarwal et al [13] was capable of 120 kW peak power at 53 MHz. Another switch implementation for NMR application is described by Vaughan et al [14] and is capable of 15 kW of peak power for 40-80 MHz and 160-180MHz band with a switching time of 10 μs .Lemette *et al.* [15] describe a T/R switch built for defense radar applications. The T/R switch was capable of handling more than 1.2 kW peak power. More recent implementations for

radar applications at frequencies ranging from 14 MHz to 900 MHz and peak powers up to 1-kW are documented by Arnold *et al.* [8, 12, 16-18]

2.3 Proof of concept demonstration with COTS quadrature hybrid couplers

A commonality of all the implementations discussed in Section 2.2 is the operation at relatively low frequencies (maximum 1 GHz). To our knowledge this duplexer design has not been implemented for microwave frequencies or multi-GHz bandwidths. Such configuration would be more challenging to implement than its lower frequency counterparts, because of sensitivity to phase and amplitude mismatches in the couplers. To verify that this concept could work at microwave frequencies we put together a simple mock-up structure using COTS quadrature hybrid couplers (Marki-QH-0226) of 2-26 GHz operating frequency (Figure 2.3.1.a). Two cases were analyzed: in the first case a single coupler was used with its outputs terminated in short circuit to simulate the transmit state (Figure 2.3.1.b); in the second case two couplers were placed back to back (Figure 2.3.1.c) to simulate the receive state of the switch. This mock-up structure does not show the switching aspect, insertion loss and isolation by using the shunt diodes and other biasing components. Nevertheless, it gives an initial estimate of the expected insertion loss in the transmit and receive modes by the using the quadrature hybrid couplers alone and allowed us to asses if the dispersion in the circuit was acceptable. Circuit simulations were performed for the 2-18 GHz range using ADS (Advanced design system) by using s-parameters from the manufacturer. The test fixture was characterized in terms of its scattering parameters, obtained with a VNA (Vector network analyzer). Figure 2.3.2 shows the simulated and measured results for the two cases under study. Both ADS simulations and measurements agree with one another. The response for the transmit mode show a maximum insertion loss approaching 2.5 dB (0.5 dB at 2 GHz). The insertion loss for the receive mode ranges between 0.5 dB at 2 GHz to and about 2 dB at 18 GHz.

These losses are comparable to those of COTS switches with much lower power handling capacity (e.g. the Qorvo TGS2353-2.). To ensure that the dispersion within the circuit would be acceptable, we also performed a delay line test with the CReSIS snow radar to quantify potential degradation of the main lobe in the impulse response when using the T/R switch. For this test, a delay line of 1.7 μs was placed between the transmitter and receiver to simulate a specular target at a range of 255 m. The radar was set to operate across the 2-18 GHz range with and without the switch mock-up in the signal path. Figure 2.3.3 shows a comparison of the system's impulse responses for four different test cases. Case-1 is the delay line response without the mock up switch setup inserted. Case-2 is the delay line response with the Tx mode mock up setup. Case-3 is the delay line response with the Rx mode setup. Case-4 is a combination of the Tx mode and Rx mode mock up setup in a bank to bank connection to determine the worst case scenario effect. From Figure 2.3.3 it can be seen that the switch would not have significant effect in the response, aside from the insertion losses, which are inherent to the incorporation of a T/R switch. This simulation and measurements validate that this approach for the radar duplexer is viable.

While the connectorized coupler would perform satisfactorily within the switch, it is not ideal for a compact module. A custom quadrature hybrid coupler with lower insertion loss and suitable for a compact module. Such approach would allow us to compensate for the insertion loss and the effect of adding the shunt PIN diodes and biasing circuitry.
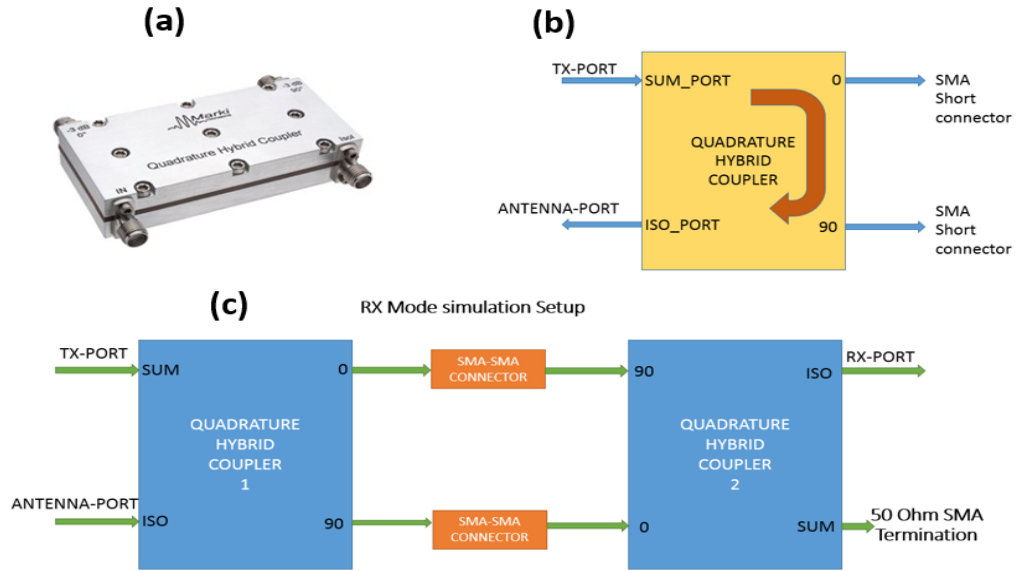
Figure 2.3.1: (a): The Marki QH-0226 ultra-wide band quadrature hybrid coupler used for the mock setup. (b): The shorted outputs set up to simulate the transmit mode operation. (c): The back to back connected set up to simulate the receive mode operation
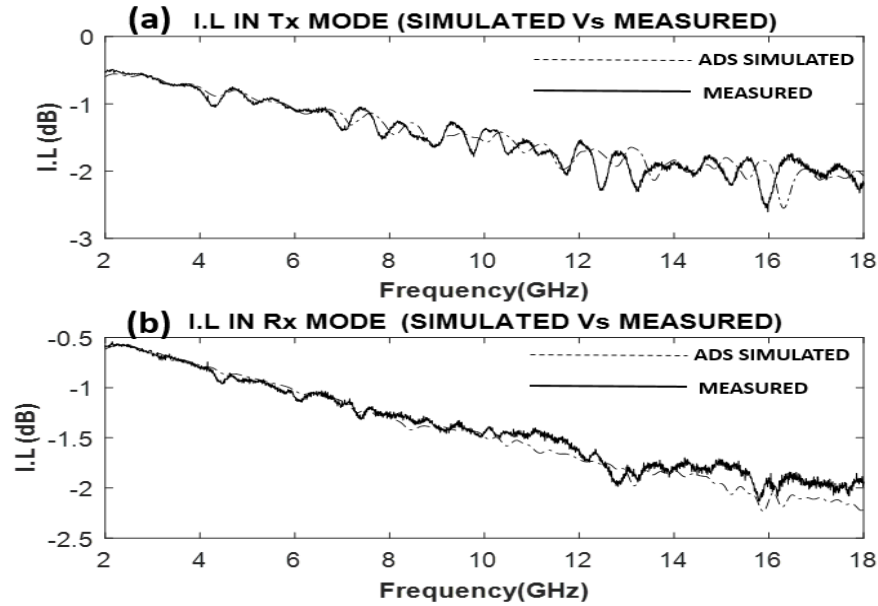
Figure 2.3.2: (a): ADS simulated vs measured insertion loss s-parameters from transmit to antenna port in transmit mode. (b): ADS simulated vs measured insertion loss s-parameters from antenna to receive port in transmit mode.
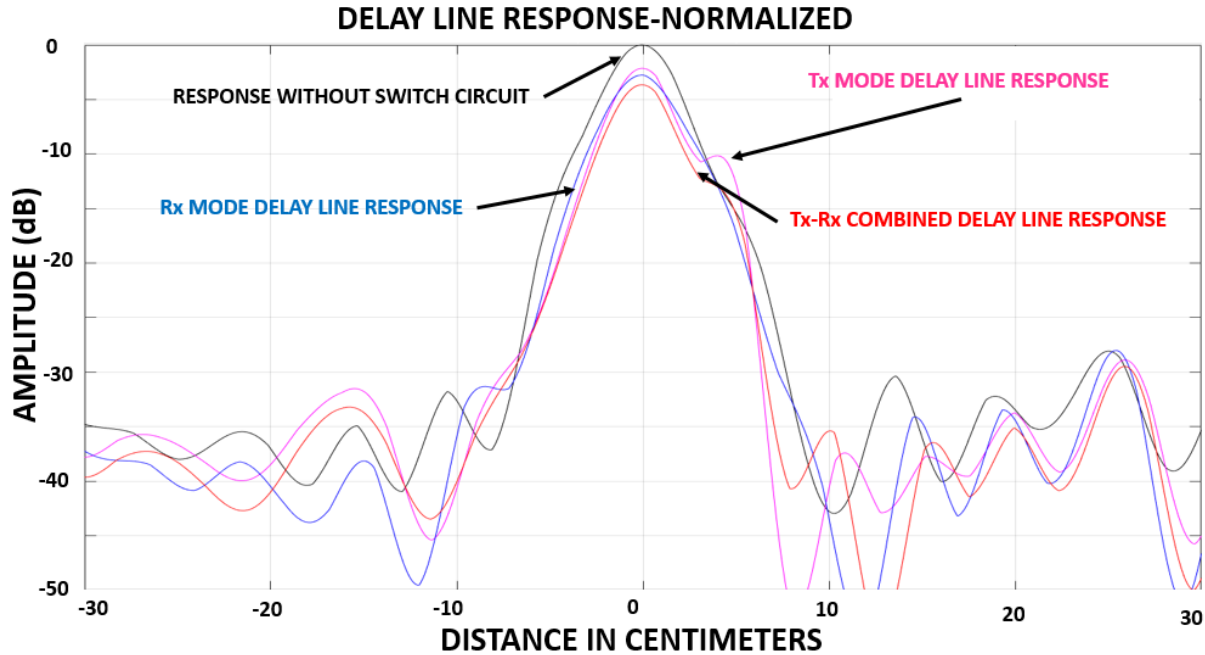
Figure 2.3.3: Measured delay line response with Tx mode / Rx mode /Combined Tx/Rx mode showing no significant degradation in the main lobe when the switch mockup is used.

2.4 Design requirements

Table 2.4.1 presents a summary of design requirements for the T/R switch.

Operational bandwidth	2-18 GHz
Peak pulsed power	> 25 Watts (desired) (10 Watts minimum)
Insertion loss during transmit	< 2 dB
Isolation during transmit	> 50 dB
Insertion loss during receive	< 2 dB
Isolation during receive	> 50 dB
Switching time between transmit and receive	< 2.3 µs
Biasing voltage supplies	+5 Volt and -24 Volt standard power supplies

Table 2.4.1: Summary of design requirements for the T/R switch circuit

20

The minimum bandwidth to be covered was determined by the operating range of the CReSIS UWB snow radar: 2-18 GHz [1, 2, 19].The T/R switch must be able to handle at least 10 Watts in continuous wave (CW) operations. This will allow operation from 60,000 feet using a single channel configuration and achieving better than 15 dB of signal-to-noise-ratio (SNR) [19]. The insertion loss in the transmit and receive path has to be ideally less than 2 dB as the COTS solutions available in the market have an insertion loss from 1 to 3.5-dB in the 2-18 GHz bandwidth[10, 20]. The insertion loss considering the fact that the receiver and transmitter are often connected to the antennas by means of long cable runs, the eventual integration of a transmit/receive module right at the antenna feed-point would result in further performance improvement. For example, a typical cable run from the radar to antenna in the NASA P3 aircraft is 20 feet and assuming and the cables have a loss of about 0.3-dB per feet (at 18 GHz), this corresponds to a total cable loss of 12-dB including transmit and receiver feed lines.

Another relevant parameter for the switch is the isolation. High isolation is required to protect the receiver from damage and even saturation. 50-dB is the minimum target for isolation as this would ensure that the integrated LNA at the receiver does not saturate [21] and keep from reaching the P1db compression of the receiver. As for the switching time, the circuit has to be able to toggle between transmit and receive states in less than 2.3 μs. to keep a blind range of less than 500 meters Lastly, the bias voltages supplies are standard COTS supplies of +5 Volt and -24 Volts to facilitate integration within the system.

Chapter 3: Quadrature hybrid coupler design

One of the critical microwave components in the balanced duplexer based T/R switch design is the quadrature hybrid coupler whose phase quadrature, amplitude imbalance, insertion loss, isolation and other properties are to be well designed over the wide operating bandwidth to ensure good performance of the T/R switch. The motivation to design this coupler in house is to implement a customized and integrated switch solution, which is better matched to the operating bandwidth, insertion and return losses and other requirements than commercially available solutions. This approach also facilitates for a PCB board solution, which can be easily modified for future miniaturization and integration. This chapter goes through an in-depth analysis on the selection, design and fabrication of the required quadrature hybrid couplers. This chapter goes through the various ways to achieve an amplitude split and phase quadrature, then goes through the basics of the quadrature hybrid coupler, dielectric material selection among others. Algorithm for generating the coupling profile for the required quadrature hybrid coupler is described along with the algorithms used for generation of the coupling profile dimensions along with basic equations for the described algorithms. A custom algorithm was developed in Matlab for the quadrature coupler design based on least squares fitting type optimization, also an algorithm to generate the quadrature hybrid's dimensions based on magnetostatic and electrostatic potential analysis was implemented.

3.1   Ways to implement analog power split with phase quadrature

In this section we discuss some of the ways to achieve an equal power split with phase quadrature between the output arms, the pros and cons of each realization and why we choose the quadrature hybrid coupler over other methods [22].

3.1.1 Power splitter with a delay

This method is primitive and straight- forward as it involves sending a signal through a power divider and passing one of the outputs through a delay line to achieve a phasing of 90° with respect to the other. This is essentially a time shifter which creates a phase shift.

Pros: Easily achievable. Can be tuned.

Cons: Limited to a single frequency. Delay has to be adjusted each time for the required frequency of operation.

3.1.2 Branchline coupler

This method can be easily implemented on a microstrip planar circuit (illustrated in Figure 3.1.1). It has a ring structure, return and insertion loss can be excellent over a narrowband of operation.

Pros: Easily achievable on a microstrip planar circuit. Excellent return and insertion loss.

Cons: Narrowband operation. Cannot be tuned.
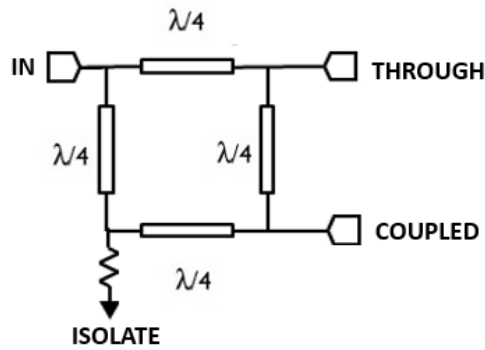
Figure 3.1.1: Branchline coupler

3.1.3 Schiffman phase shifter

This method requires a pre-existing power split and can be used on differential signals and provides for a broadband phase shift. Differential signals are split, one is sent through a delay line and the other through the Schiffman phase shifter to achieve phase quadrature.

Pros: Broad banding can be achieved.

Cons: The signals have to be pre-split.

3.1.4 Polyphase filter quadrature splitter

The polyphase filter quadrature splitter (illustrated in Figure 3.1.2) is easily implemented on CMOS chips for cellular applications, where the LO (Local Oscillator) signals are phase shifted to achieve quadrature phasing. Wider bandwidths can be achieved by cascading section, at the expense of poor return and insertion loss.

Pros: CMOS compatible. Inexpensive solution for large-volume production.

Cons: Low power handling. Poor return and insertion loss. Low frequency of operation.
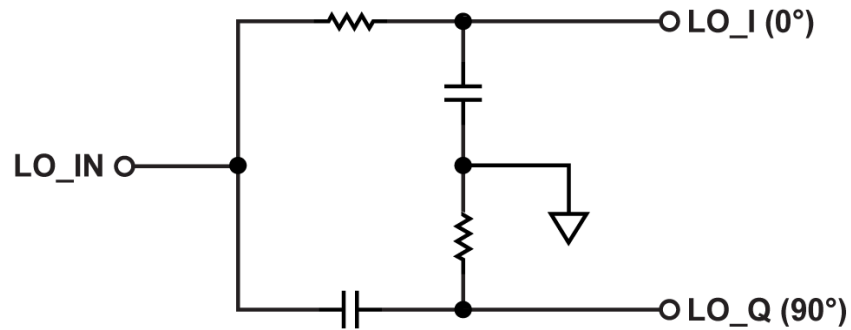
Figure 3.1.2: Polyphase quadrature splitter

3.1.5 Lange coupler

The Lange coupler (illustrated in Figure 3.1.3) is an excellent MMIC (Monolithic Microwave Integrated circuit) compatible solution. The coupling strength can be made substantial and the signal jumpers are typically gold bond wires. Wider bandwidth is obtained by increasing the number of fingers at the expense of a degraded phase balance.

Pros: MMIC compatible. Wide bandwidth.

Cons: Requires wire bonding. Low power rating.
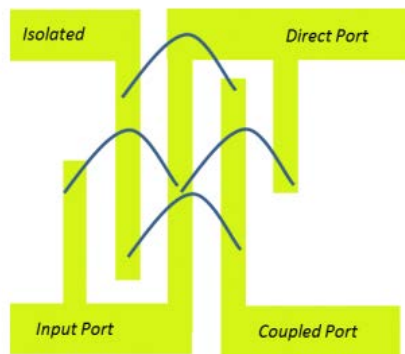
Figure 3.1.3: Lange coupler

3.1.6 3-dB Quadrature hybrid coupler

The 3-dB quadrature hybrid coupler (illustrated in Figure 3.1.4) consists of closely coupled quarter wave TEM (Transverse Electro Magnetic) line sections to achieve equal-power coupling and quadrature phase. This is the best approach to all of the aforementioned approaches, providing for superior bandwidth and phase quadrature. The bandwidth can be increased by increasing the number of quarter wave sections. At microwave frequencies, these couplers are typically implemented in a triplate stripline configuration.

Pros: Multi-octave operating bandwidth. Excellent phase and amplitude balance. Can handle much higher power over a wide bandwidth.

Cons: Superior low loss dielectric materials required. Challenging mechanical integration. Difficult to design and characterize.
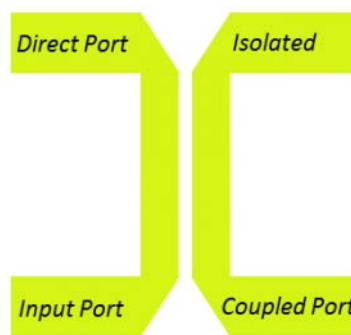
Figure 3.1.4: Quadrature hybrid coupler

3.2   A brief history of quadrature hybrid couplers

Quadrature hybrid couplers belong to a class of passive microwave devices which include, power dividers, directional couplers, phase shifters and microwave filters among others. Quadrature

hybrid couplers get their name from the property of the device to have phase quadrature between the through and coupled ports across the specified bandwidth as well as the equal power split between them. The precursor to the quadrature hybrids is the directional coupler, whose invention is credited to H. A. Affel of A.T.T. Even though the name directional coupler was used much later, the principle of operation remains very much the same as the primitive setup of Affel. He used a quarter wave parallel wire conductor, with one side terminated on a resistor and the other connected to a detector placed in close proximity to a two wire transmission line, the coupling of signals was measured at the detector [23].

The earliest mathematical theory on directional couplers was published by H.J Riblet in the 1940s. It showed that sections of holes, probes, transmission lines and other conductors of quarter-wave length sections ($\lambda o/4$ with λo being the operating wavelength) placed between parallel waveguides, can couple at the selected frequency f_o. The coupling strength is furthermore increased by using 'n' identical sections of the same length in series, with offsets from the center. The resulting coupled powers at the coupled line and through line is proportional to $\cos^2(n\text{K})$ and $\sin^2(n\text{K})$ where K is an integer. There are two additional subclasses to the directional coupler, which are the symmetrical and asymmetrical. The symmetrical directional coupler (illustrated in Figure 3.2.2) uses $(2n+1)$ quarter-wave sections to achieve similar bandwidth of an n section asymmetrical directional coupler (illustrated in Figure 3.2.1), which increases the conductor losses but enables a useful property of phase quadrature between the through and coupled ports. This property was not tested experimentally then but was evident with the scattering parameter matrix of the symmetrical directional coupler. The coupler achieves isolation property as the backward travelling coupled waves are 0° offset with respect to each other. They constructively interfere at the coupled port

while destructively interfering at the isolation port as it is 180° phase shifted with respect to each other [24].
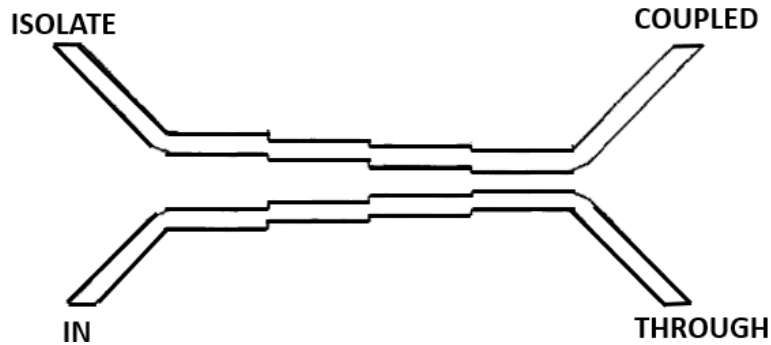
Figure 3.2.1: Asymetrical directional coupler
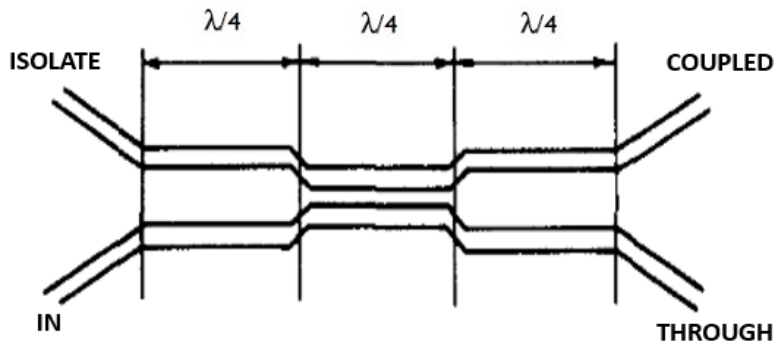
Figure 3.2.2 Symmetrical directional coupler

Henceforth we are going to deal with proximity coupled quarter wave TEM line based directional couplers/quadrature hybrid couplers, as that would be our preferred direction of design and development. Some fundamental properties of parallel coupled TEM lines need to be mentioned before we delve into further discussion. A way to mathematically analyze these coupled lines can be in terms of its even (illustrated in Figure 3.2.3) and odd mode (illustrated in Figure 3.2.4)

analysis and the corresponding even mode (Z_{oe}) and odd mode impedances (Z_{oo}). The wave induced in the coupled line propagates antiparallel to the direction of the wave in the through line. Perfect directivity and unity VSWR (Voltage Standing Wave Ratio) is achieved if $Z_{oe}.Z_{oo} = Z_o^2$ is maintained throughout the length of the line, where Z_o is the characteristic impedance of the termination transmission line which is typically chosen to be 50 Ohms. Another important property of the directional coupler is the voltage coupling constant k which is equal to the reflection coefficient of the even mode or negative of the odd mode at each point along the direction of propagation of the coupler if the line is terminated in Z_o. The coupler design can be completely defined in terms of k and the ratio Z_{oe} / Z_o.
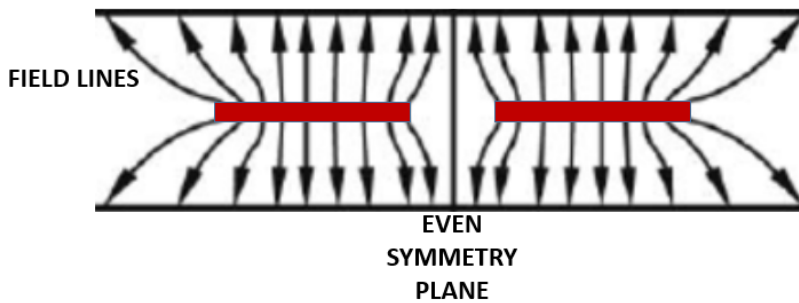
Figure 3.2.3: Even mode analysis.
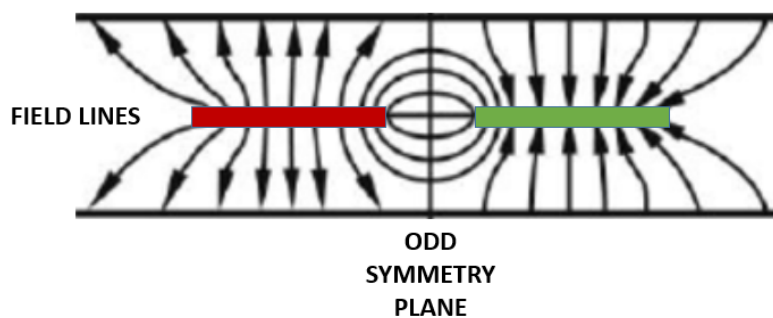
Figure 3.2.4: Odd mode analysis.

The earliest practical implementation of a symmetrical directional coupler was a three section coupler (illustrated in Figure 3.2.2), by E. F. Barnett et al. in the 1950s, who demonstrated a remarkably large bandwidth improvement compared to previous approaches. Further

29

improvements to facilitate for easier 3-dB coupling by using a tandem connection was advanced by G. D. Monteath. The tandem crossover method (illustrated in Figure 3.2.6) required an offset parallel stripline arrangement than the earlier edge coupled stripline versions demonstrated. Early versions of the tandem stripline quadrature couplers used thick copper stripline suspended in air, supported by dielectric tabs, these structures had poor performances mechanically speaking as the dielectric tabs had to be small to have minimal mismatch from the air dielectric. Electrically too this structure suffered from low performance due to thick copper strips being used for mechanical stability reasons which created undesired fringing fields at the strip edges. Advancements in material technology enabled for superior low loss dielectric materials with low dielectric constant Dk/Er, which could be used instead of the air dielectric providing good mechanical stability and enabling the use of thin copper strips. These materials offer negligible performance deterioration with respect to air as the dielectric material.

The properties of the symmetrical directional coupler such as the *'n'* sectional quarter wave coupling and phase quadrature property is exploited to create the 3-dB quadrature hybrid coupler. The idea to use multi-sectional coupled TEM lines in a crossover fashion with offsets varying from closely coupled at the center to the farthest apart with essentially no coupling at the ends of the coupler was conceived by J. Paul Shelton, Jr in the mid 1960's. One of the biggest impediments to achieve a broad bandwidth with 3-dB coupling was the extremely tight center coupling required. This is a physical implementation problem as it requires unpractically extreme coupling of the center strips and needing extremely thin dielectrics and copper thickness and etching tolerances. This problem can be alleviated by using two $-8.343 \, dB$ symmetric directional couplers in a tandem crossover fashion (illustrated in Figure 3.2.7), which produces a 3-dB quadrature hybrid as shown by J. P. Shelton and J. A. Mosko. The equation *(3.2.1)* shows that by using two of the

$-8.343\ dB$ couplers in tandem we can achieve a $-3.01\ dB$ quadrature hybrid coupler. Also another implied characteristic of this double symmetric device arrangement is that its symmetry is maintained irrespective of port numbering. Assuming any of the ports as the reference input port, the port adjacent to it becomes the isolated port, the port directly opposite to it becomes the through signal port, and the port diagonally opposite to it becomes the coupled port.

$$-3.01\ dB = 20 * \log\left(\sin\left(\frac{\pi}{4}\right)\right) \quad - (3.2.1)$$

$$-8.343\ dB = 20 * \log\left(\sin\left(\frac{\pi}{8}\right)\right) - (3.2.1)$$
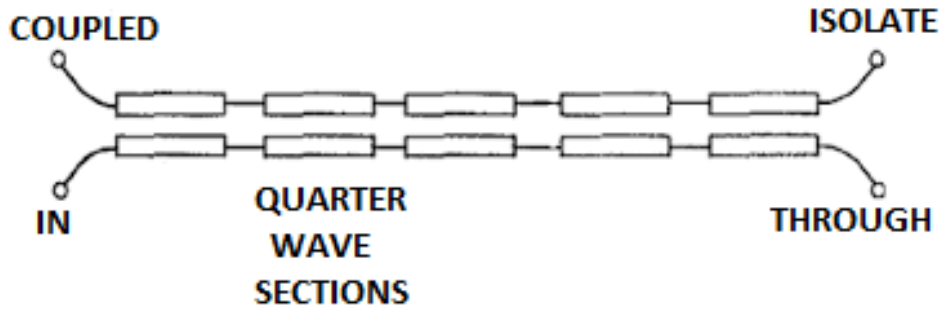
Figure 3.2.5: N-sectional TEM coupler
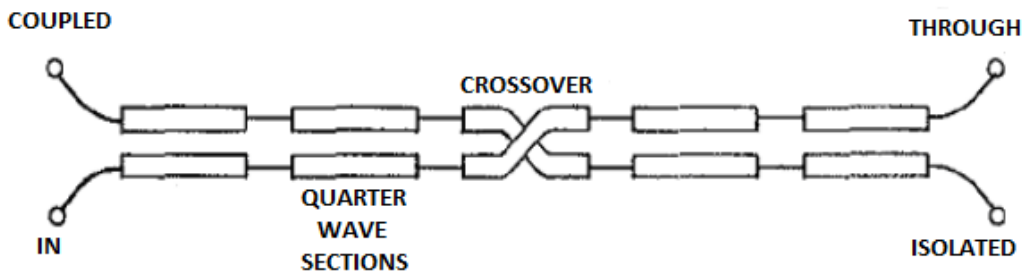
Figure 3.2.6: N-sectional symmetrical coupler with cross-over

31
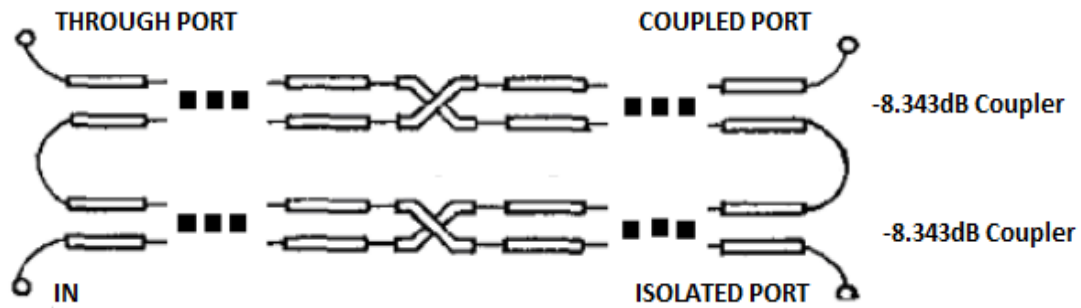
Figure 3.2.7: Tandem crossover connection from two -8.343-dB symmetrical coupler

3.3 Figures of merit for quadrature hybrid couplers

In this section we define some of the important figures of merit of a directional coupler/quadrature hybrid in order to characterize the device's performance [25].

3.3.1 Operating bandwidth

This indicated the frequency range over which the device operates, providing for 3-dB coupling and phase quadrature.

3.3.2 Splitting ratio

Splitting ratio is defined as the ratio of input to output power, in the quadrature hybrid coupler case it is a half power split between the through and coupled ports of 2:1splitting ratio.

3.3.3 Coupling ratio

Coupling ratio is defined as the ratio of input to output power, in the quadrature hybrid coupler case it is a half power split between the through and coupled ports, hence a -3-dB coupling.

3.3.4 Relative phase shift

This refers to the phase difference between the through and coupled ports. For an ideal quadrature hybrid this is a 90° phase lead of the coupled port signal over the through port signal.

3.3.5 Amplitude unbalance

Amplitude unbalance is a measure of how well the power is split between the through and coupled ports over the bandwidth, well designed quadrature hybrids will have a very low value of typically less than 0.5-dB for narrow band and less than 1 dB for wideband.

3.3.6 Amplitude ripple

A perfect quadrature hybrid would have a flat response in band, and amplitude ripple is a measure of how much a practical coupler deviates from the ideal case, generally speaking bandwidth can be traded to achieve lower ripple.

3.3.7 Phase unbalance/imbalance

Phase unbalance is a measure of how well relative phase shift is achieved between the through and coupled ports over the operational bandwidth. Ideally a phase shift of 90° is desired over the operational bandwidth. Here too bandwidth can be traded for a lower phase unbalance.

3.3.8 Phase ripple

Analogous to amplitude ripple, this property is a measure of the ripple in the phase quadrature between the coupled and through ports over the entire operational bandwidth and how much it deviates from the ideal case, here too lower phase ripple can be traded for lower bandwidth.

3.3.9 Insertion loss

This is a measure of the loss in power from input to the output ports, ideally a 3 dB coupler has a $-3.01\ dB$ loss in each of the arms provided the ports are terminated on the characteristic impedance, but in the real world other losses such as dielectric loss, mismatch loss, skin effect,

radiation loss and conductor loss adds onto into this to make it worse, also higher operating frequencies mean that these losses are more severe, careful design and material selection is required to mitigate these effects.

3.3.10 Return loss

This is a measure of loss resulting from reflections caused by mismatches in impedances along the direction of propagation. These losses can be reduced by perfectly terminating the ports to its characteristic impedance of typically 50 Ohms and also by careful design to reduce impedance mismatches along the coupler.

3.3.11 Coupler isolation

The quadrature hybrid is a four port device and coupler isolation is a measure of how well the signal from the input port is isolated from the arm of the adjacent isolation port. A perfect quadrature hybrid coupler would allow no signal to reach the isolation port and generally speaking higher bandwidth degrades the isolation and careful optimization is required to obtain good performance over the entire operational bandwidth.

3.4   Discrete vs tapered TEM directional couplers

One of the issues encountered with an 'n' sectional discrete TEM line coupler (illustrated in Figure 3.4.1) is the evident impedance discontinuities along the direction of propagation due to the abrupt changes in the coupling strength and dimensions between the coupling sections. While these may not be detrimental at low frequencies, at higher operational frequency these contribute to increased return losses, distortion and degradation of isolation and directivity. These effects are can be compensated for adequately if the condition $b * \frac{\sqrt{Er}}{\lambda} < 0.1$ is maintained, where b is the spacing between the stripline conductors and Er/Dk is the dielectric constant of the stripline material and λ is the wavelength of operation . These bounding conditions cannot be satisfied in our application

of 2-20 GHz+. Another problem in the discrete TEM line coupler approach is the parasitic capacitance and inductance induced at the discontinuities [26], which leads to excitation and propagation of unwanted modes, this is mainly caused by the discontinuities due to the conductor trace offsets becoming comparable to the size of the conductor widths as seen in the figure below.
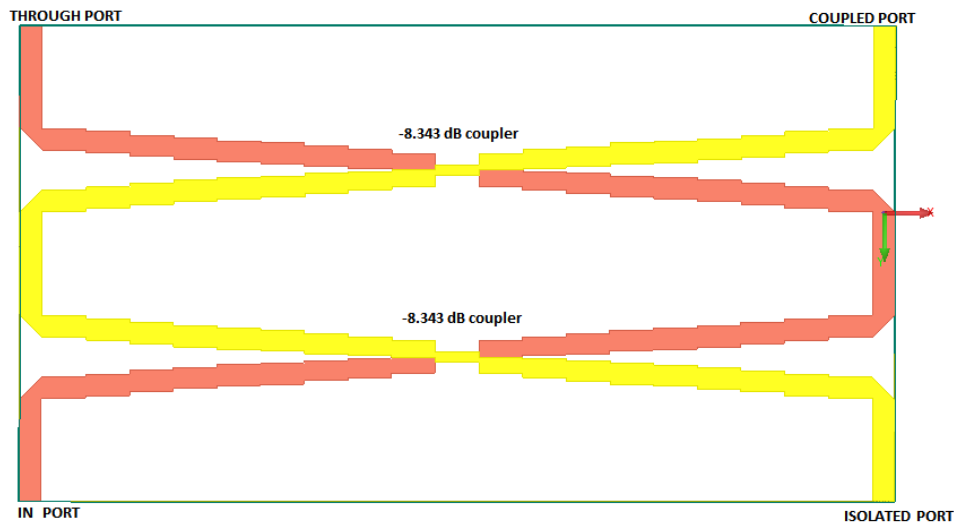
Figure 3.4.1: Discrete N section 3-dB Quadrature hybrid coupler

One of the approaches suggested to circumvent this problem was to simply taper off at the discontinuities [27] [28], but this approach resulted in a drop of coupling strength at the tapers, which results in issues like lowering of operational bandwidth and phase imbalance. Another approach was to divide the discrete coupling sections into smaller sections to approximate a desired taper smoothing effect [29] [30] and have an equal-ripple response across the band, this improved performance but really didn't capture the physics behind the coupler response and hence was not the best solution available. A better approach is to use an optimization technique suggested by D.W. Kammler [31]which involves Remez type numerical optimization methods to provide for

a continuously tapered coupler of equal-ripple response and coupling errors within a desired tolerance. This method provides for good performance at the expense of significantly higher design complexity, time and computational resources required. Further sections deal with the design of this tapered quadrature hybrid coupler required for our T/R switch application.

3.5 Dielectric material and laminate thickness selection for the quardrature hybrid.

Dielectric material selection plays a prominent part in the design of the quadrature hybrid coupler, the material selected needs to have properties as close to that of vacuum to have the best achievable coupling and low insertion loss, while still having the required mechanical stability to enable the use of thin conductors and other properties which helps out in the fabrication and operational stage. Two of the materials suitable for our application was the RT/Duroid 5880 laminate from Rogers corporation and the TLY 5 from Taconic, both of which had a low Dk of 2.2 and dielectric loss tangent of 0.0009. The RT/Duroid 5880 was a better material in terms of dielectric uniformity as it has a random chopped fiberglass PTFE (Polytetrafluoroethylene) composition and the Taconic TLY 5 is a PTFE material with reinforced fiber glass for mechanical stability, we chose the latter as it has a better coefficient of thermal expansion (CTE) in all axes than the RT/Duroid 5880. CTE is detrimental when the duplexer operates for extended periods of time and under high power which cause copper trace heating resulting in trace registration misalignment and consequently degrades the coupler performance. The chosen material also has a better breakdown voltage rating which helps out in peak pulsed power handling capability. *The properties of the Taconic TLY-5 material are shown in Table 3.5.1*

TLY Typical Values					
Property	Test Method	Unit	Value	Unit	Value
Dk @ 10 GHz	IPC-650 2.5.5.5		2.20		2.20
Df @ 10 GHz	IPC-650 2.5.5.5		0.0009		0.0009
Moisture Absorption	IPC-650 2.6.2.1	%	0.02	%	0.02
Dielectric Breakdown	IPC-650 2.5.6	kV	>45	kV	>45
Dielectric Strength	ASTM D149	V/mil	2,693	V/mm	106,023
Volume Resistivity	IPC-650 2.5.17.1 (after elevated temp.)	Mohms/cm	10^{10}	Mohms/cm	10^{10}
Volume Resistivity	IPC-650 2.5.17.1 (after humidity)	Mohms/cm	10^{10}	Mohms/cm	10^{9}
Surface Resistivity	IPC-650 2.5.17.1 (after elevated temp.)	Mohms	10^{8}	Mohms	10^{8}
Surface Resistivity	IPC-650 2.5.17.1 (after humidity)	Mohms	10^{8}	Mohms	10^{8}
Flex Strength (MD)	IPC-650 2.4.4	psi	14,057	N/mm²	96.91
Flex Strength (CD)	IPC-650 2.4.4	psi	12,955	N/mm²	89.32
Peel Strength (1/2 oz. ED copper)	IPC-650 2.4.8	lbs./inch	11	N/mm	1.96
Peel Strength (1 oz. CL1 copper)	IPC-650 2.4.8	lbs./inch	16	N/mm	2.86
Peel Strength (1 oz. CV1 copper)	IPC-650 2.4.8	lbs./inch	17	N/mm	3.04
Peel Strength	IPC-650 2.4.8 (at elevated temp.)	lbs./inch	13	N/mm	2.32
Young's Modulus (MD)	ASTM D 3039/IPC-650 2.4.19	psi	1.4×10^{6}	N/mm²	9.65×10^{3}
Poisson's Ratio (MD)	ASTM D 3039/IPC-650 2.4.19		0.21		0.21
Thermal Conductivity	ASTM F 433	W/M*K	0.22	W/M*K	0.22
Dimensional Stability (MD, 10 mil)	IPC-650 2.4.39 (avg. after bake & thermal stress)	mils/inch	-0.038	mm/M	-0.038
Dimensional Stability (CD, 10 mil)	IPC-650 2.4.39 (avg. after bake & thermal stress)	mils/inch	-0.031	mm/M	-0.031
Density (Specific Gravity)	ASTM D 792	g/cm³	2.19	g/cm³	2.19
CTE (x) (25 - 260˚C)	ASTM D 3386 (TMA)	ppm/°C	26	ppm/°C	26
CTE (y) (25 - 260˚C)	ASTM D 3386 (TMA)	ppm/°C	15	ppm/°C	15
CTE (z) (25 - 260˚C)	ASTM D 3386 (TMA)	ppm/°C	217	ppm/°C	217
NASA Outgassing (% TML)			0.01		0.01
NASA Outgassing (% CVCM)			0.01		0.01
NASA Outgassing (% WVR)			0.00		0.00
UL-94 Flammability Rating	UL-94		V-0		V-0

Table 3.5.1: Taconic TLY-5 dielectric laminate properties [32]

The next consideration after fixing on the dielectric material was to choose the thickness of the laminates to be used. For the strongest coupling a low value of 's' (illustrated in Figure 3.5.2) which is the thickness of the center dielectric sheets has to be chosen. The thinnest dielectric laminate sheet available with the selected dielectric was 3.5 mils thick. The next thickness in consideration was that of the outer dielectric layers which has a thickness of $(b - s)/2$ where b is the top and bottom ground plane spacing. The ground plane spacing is limited at the lower end by the coupling strength required and parallel plate radiation effects and at the upper end by the excitation of higher order mode, an empirical formula (equation 3.5.1) where W is the maximum

conductor width sets limit on maximum usable frequency in GHz. We chose 30 mils outer
dielectric layer thickness form the available selection from Taconic, this corresponds to a *b*= 63.5
mils, which satisfies our design requirements as well as meet the bounds for the operating
bandwidth.

$$Fmax = \frac{30}{\sqrt{Er} * (2W + \frac{\pi b}{2})} \quad (GHz) \quad - (3.5.1)$$
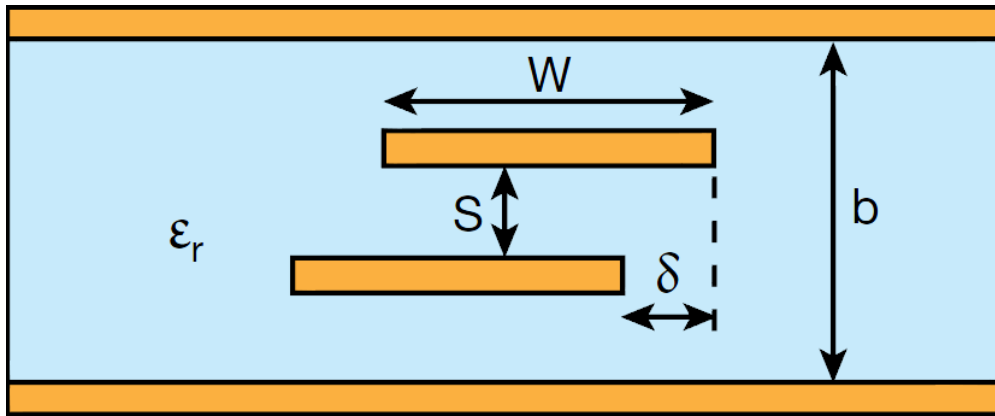
Figure 3.5.2: Broadside offseted stripline crosss section view

3.6 Algorithm for generation of coupling profile for continuously tapered quadrature hybrid couplers

Let us now briefly discuss the design algorithm for generation of the coupling profile required for
the quadrature hybrids. The procedure used to synthesize the couplers is based on a computer aided
design method suggested by David W. Kammler which involves non-linear optimization
techniques based on a generalized Remez type optimization and curve fitting type techniques[33].
The software used to implement the algorithm is Matlab, which stands for matrix laboratory and

38

is a matrix based scientific computing language well suited for our optimization procedure. A brief explanation of the optimization procedure is given along with basic flowcharts of the implemented code used to obtain the coupling response. The complete code is given in the appendix, which can be modified to be used to generate tapered coupling profiles for any given mean coupling strength, coupler length and for a given operating bandwidth and within an acceptable error tolerance limit.

The basic idea behind the optimization algorithm is to convert a given coupling strength profile along the length of the coupler to a corresponding power coupling response in the frequency domain, and to tune and tweak the coupling profile till a target output response is achieved in the frequency domain, which is the domain we are interested in. This problem can be thought of analogously as an input excitation function convolved through an unknown black box function which results in an output response function, since the desired output response function is known and determined by our design requirements the goal is to find the optimum input function which causes this desired response. Let us now go briefly through the building ideas and mathematical concepts of this algorithm.

3.6.1 Spectrum analysis block

The most commonly used block in the algorithm is the spectrum analysis block, which provides for a way to convert a given continuously tapered coupling strength profile to a corresponding power coupling response in the frequency domain. A continuously tapered coupling strength profile can be approximated by a large number of infinitesimally small elemental sections (illustrated in Figure 3.6.1.1).
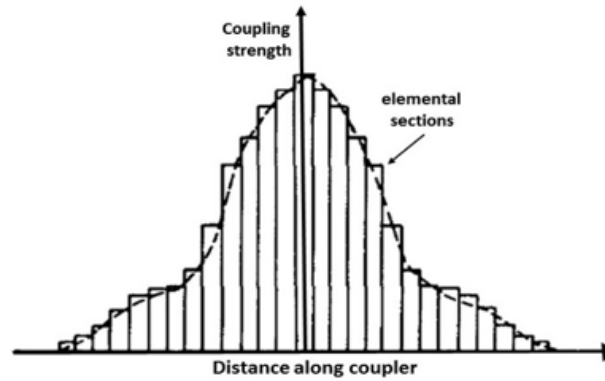
Figure 3.6.1.1: Elemental TEM coupling sections approximation

Each element can be thought of as a uniformly coupled TEM line pair (illustrated in Figure 3.6.1.2). with its of incidence wave coefficient of *a1* and *a4* and reflected wave coefficient of *b2* and *b3*.
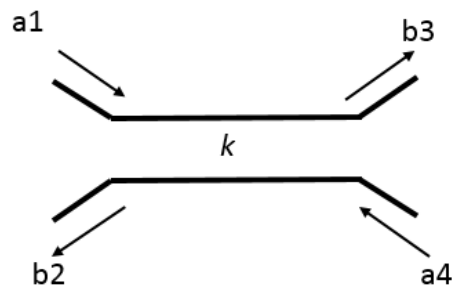
Figure 3.6.1.2: Elemental TEM coupling sections for analysis

These wave coefficients of the elemental section are related by the matrix equation

$$\begin{bmatrix} b2 \\ a1 \end{bmatrix} = \Delta T(k, \Delta L) \begin{bmatrix} a4 \\ b3 \end{bmatrix}$$

in which the elemental transfer matrix $\Delta T(k, \Delta L)$ is given by

$$\Delta T(k, \Delta L) = \begin{bmatrix} 1/\overrightarrow{C'} & C/C' \\ \overrightarrow{C}/\overrightarrow{C'} & 1/C' \end{bmatrix}$$

40

Where C is the coupled transfer function (equation 3.6.1.1) and C' is the through transfer

function (equation 3.6.1.2) and k is the coupling coefficient between the TEM elemental

coupled lines.

$$C = \frac{jk * \sin(\beta \Delta L)}{\sqrt{(1 - k^2)} * \cos(\beta \Delta L) + j\sin(\beta \Delta L)} \quad - (3.6.1.1)$$

$$C' = \frac{\sqrt{(1 - k^2)}}{\sqrt{(1 - k^2)} * \cos(\beta \Delta L) + j\sin(\beta \Delta L)} \quad - (3.6.1.2)$$

Also β is the propagation constant in the dielectric medium given by

$$\beta = 2\pi f / v$$

Where f is the operating frequency and v is the velocity of propagation in the dielectric medium given by .

$$v = c/\sqrt{Er}$$

Since we formulated a way to find the response of an elemental TEM line section we can now

cascade the elemental sections which are assumed to have no cross coupling in series to any desired

length (illustrated in Figure 3.6.1.3).
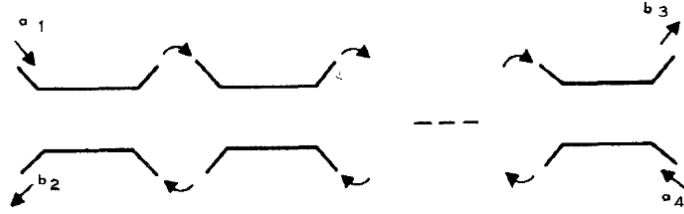
Figure 3.6.1.3: Cascading of elemental sections to form the complete coupler

The resulting transfer function is a product of the individual elemental transfer functions (equation 3.6.1.3). The continuously tapered coupler case can be achieved by applying a limiting condition as $Ns \to \infty$ where Ns is the number of elemental sections. Practically about 100 to 500 elemental sections can be used as a reasonable limiting case for most tapered couplers.

$$T = \Delta T(k1, \Delta L) * \Delta T(k2, \Delta L) * \Delta T(k3, \Delta L) * \dots \dots \dots . \Delta T(kNs, \Delta L) \; - (3.6.1.3)$$

Since the coupler is symmetric about the center having $(2s - 1)$ elemental sections, we can transform the Transfer matrix equations into a convenient form for ease of computations *(3.6.1.4)*

$$T_{sym} = \Delta T(ks, \Delta L) * \Delta T(ks - 1, \Delta L) * \dots . \Delta T(k2, \Delta L) * \Delta T(k1, \Delta L)$$

$$* \dots \Delta T(k2, \Delta L) \dots \dots . \Delta T(ks - 1, \Delta L) * \Delta T(ks, \Delta L) \quad - (3.6.1.4)$$

The equations can be expressed in an inductive form equivalent (equation 3.6.1.5) , which decomposes the transfer matrix T_{sym} to a problem in terms of its electrical length θ. The coefficients ρ_s, σ_s, π_S are real valued trigonometric polynomials that depends on the coupling coefficients $k1, k2, k3, k4, \dots \dots \dots . ks$ of each section.

$$T_{sym} = \begin{bmatrix} \rho_s \cos \theta - j\sigma_s \sin \theta & j\pi_S \sin \theta \\ -j\pi_S \sin \theta & \rho_s \cos \theta + j\sigma_s \sin \theta \end{bmatrix} \; - (3.6.1.5)$$

The initial values of the coefficient are

$$\rho_1 = 1 \ , \ \ \sigma_1 = \frac{1}{\sqrt{(1-k_1^2)}} \ \ , \ \ \pi_1 = \frac{k_1}{\sqrt{(1-k_1^2)}}$$

Using the recursive relationships of equation *(3.6.1.6)*, the polynomials of the next stages

s=2,3,..Ns where *Ns* is the number of sections in each of the symmetrical side of the coupler and

the coupler has *(2*Ns-1)* sections in total.

$$\rho_s = \rho_{s-1} - (2sin^2\theta) \ \langle \rho_{s-1} + \left(\frac{1}{\sqrt{(1 - k_1^2)}} \right) \sigma_{s-1} - \left(\frac{k_s}{\sqrt{(1 - k_1^2)}} \right) \pi_{s-1} \ \rangle$$

$$\sigma_s = \sigma_{s-1} + \left(\frac{1}{\sqrt{(1-k_1^2)}} \right) (\rho_s + \rho_{s-1}) \qquad - (3.6.1.6)$$

$$\pi_s = \pi_{s-1} + \left(\frac{k_s}{\sqrt{(1-k_1^2)}} \right) (\rho_s + \rho_{s-1})$$

This recursive calculation is done until we obtain the final coefficients of ρ_{Ns} , π_{Ns} , σ_{Ns} and the

procedure is repeated for all frequencies of interest. From the T_{sym} matrix we have by comparison

$$\frac{C}{C'} = j\pi_{Ns} \sin\theta$$

$$\frac{1}{C'} = \rho_{Ns} \cos\theta + j\pi_{Ns} \sin\theta$$

Also for a loss less coupler assuming the law of conservation of energy is not violated we have

$$|C'|^2 + |C|^2 = 1$$

Now from comparisons of the above mentioned equations we have the coupling power response

as given

$$|C|^2 = \frac{\pi_{Ns}{}^2 \sin^2 \theta}{1 + \pi_{Ns}{}^2 \sin^2 \theta}$$

The coupling power frequency response in dB as a function of frequency

$$Power(f) = 10 log 10 |C(f)|^2$$

Now we have a way to analyze the power coupling frequency response at desired frequencies (illustrated in Figure 3.6.1.4) for a given coupling strength function along the direction of propagation.
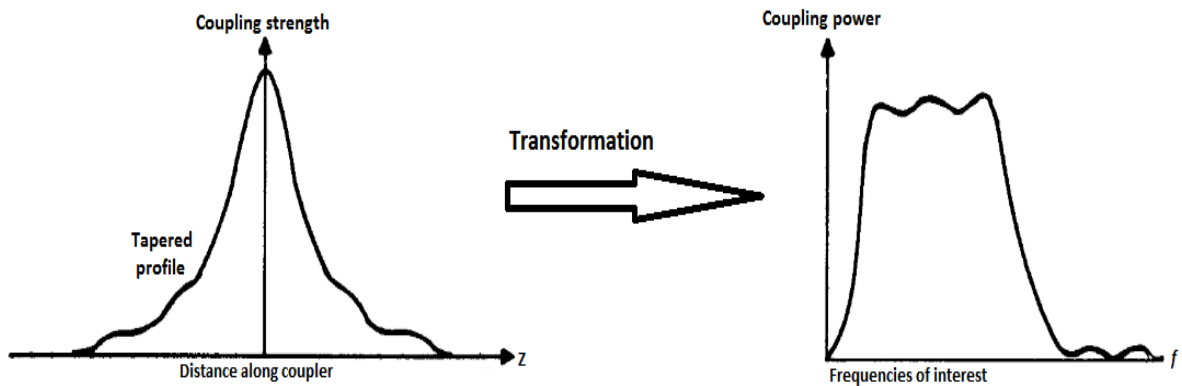
Figure 3.6.1.4: Coupling profile to coupled power frequency response

3.6.2 Parametrization of the coupling profile- Spline parametrization

Before we go into further discussion on the optimization procedure, we need a way to parameterize the coupling profile to reduce the number of independent variables of optimization. Letting all the elemental sections to be independent optimization variables would create a computationally unpractical and inefficient problem as there are typically more than 500 elemental sections in our problem and the analysis is over a wide bandwidth which creates large 3 dimensional matrices for computations. Initial estimation of processing time on a typical high end desktop computer by

letting all the variables free for a wide band width optimization was found to require more than a week for each iteration which was obviously absurd and unpractical.

There are a number of ways to parameterize and fit the coupling profile curve so as to have a smooth monotonic shape and to cause a desirable output response. The best way is to use a smooth fitting high order natural spline curve which is shown to give lower amplitude ripple response. The symmetrical profile is divided into equally spaced intervals (illustrated in Figure 3.6.2.1) with the first interval being the central point of the tightest coupling to the end points being of zero coupling for a finite length coupler. Since the coupler is symmetric about the central position the parameters are also mirrored, hence also arises the added benefit of needing to optimize only one half of the coupling profile. The coupling profile is then a spline curve fit between the parameters with the coupling strength itself acting as the parameters. Too few parameters don't afford enough degrees of freedom and too many parameters create numerical errors rendering the optimization technique useless. About ten to twenty parameters are found sufficient enough to accurately characterize and parameterize a given coupling profile.
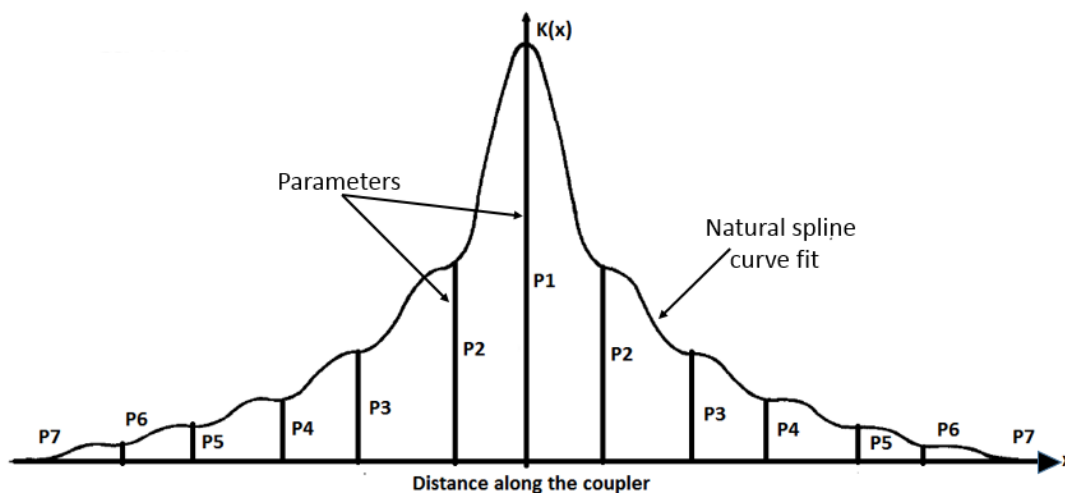
Figure 3.6.2.1: Spline parametrerization of coupling profile

Another important consideration while applying symmetrical parameterization is to prevent abrupt discontinuity at the tightest coupling point, as this corresponds to an abrupt change in line dimensions and impedances at the crossover points in the physical coupler. This can be achieved by setting the differential change constraint at this point to zero. Also another consideration is to fix the tightest coupling strength possible as this is bound by the dielectric material properties and the spacing between the broadside stripline conductors. This parameter is fixed[34] and cannot be an optimization variable and is set fixed to K_{max}, where K_{max} is set based on the characteristic impedance and the maximum possible center section coupling with the chosen dielectric laminate thickness.

3.6.3 Optimization of the coupling profile

Now since we have a way to analyze the coupler frequency response from any given arbitrary tapered symmetrical coupling profile and a way of parametrizing the profile for computational efficiency let us now look briefly into the optimization procedure, further equations and detailed explanation are mentioned in the paper by D.W. Kammler[31]. The working idea of the optimizer is to use least squares curve fitting to find the optimum coupling profile for the target coupled frequency response. The frequency coupling spectrum is calculated for equal intervals for a sufficiently large number of frequency points to get a continuous response, about 500-1000 frequency points was found to be good enough in our case to approximate for a good enough response curve. Now the desired mean coupling response of the symmetrical directional coupler is fixed as this would be our target to achieve.

Once the frequency spectrum is calculated for the corresponding coupling profile the extremums in the spectrums are taken (illustrated in Figure 3.6.3.1) and an estimate of the error is calculated. The error in this case is the extremal deviation from the desired mean coupling value. The

optimization problem now turns into the form of a Chebyshev type optimal compression problem where the optimal parameters are obtained when the minimal value of the error curve is reached.

$$Maximum_{f\min<f<f\max} \left| C(f, Parameters_{Optimum}) - Mean\ coupling \right| = Minimize$$
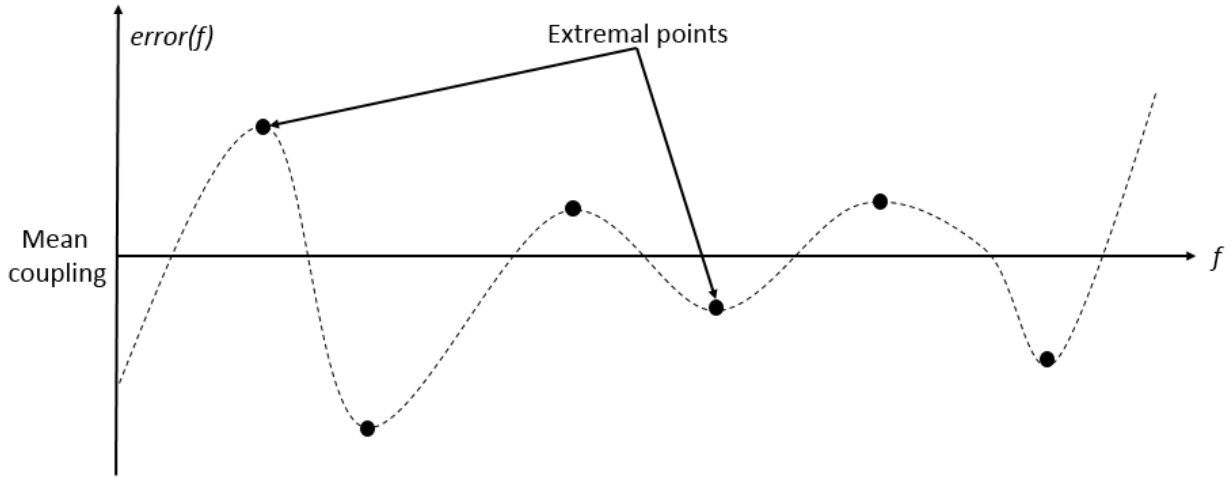
Figure 3.6.3.1: Coupling error response curve

After estimation of the extremums from the error curve the average extremal deviation (equation 3.6.3.1) is calculated ,where $\bar{P} = (P_1, P_2, P_3.....P_n).$ is the vector set of complete parameters and m is the number of extremums in the frequency band of operation.

$$d_{avg} = \frac{1}{m} \sum_{i=1}^{m} |\varepsilon(f, \bar{P})| \quad - (3.6.3.1)$$

Approximate sensitivity Jacobian matrix 'D' is calculated to determine the sensitivity of the frequency response to each of the optimization parameters, where the elements of the matrix D_{ij} (equation 3.6.3.2) is obtained by a differential variation in each of the parameter and finding its gradient. The resulting matrix is a sensitivity gradient matrix of dimension $m \times n$.

$$D_{ij} = \frac{\varepsilon\left(f_i, P_1, P_2, \dots P_{j-1,} P_j + \delta P_j, P_{j+1,} \dots \dots \dots P_{n,}\right) - \varepsilon\left(f_i, P_1, P_2, \dots P_{n,}\right)}{\delta P_j} - (3.6.3.2)$$

Once the gradient matrix is obtained we define the error vector $\Delta\varepsilon$ (equation 3.6.3.4) .

$$\Delta\varepsilon \equiv \varepsilon(f_i, \bar{P} + \delta\bar{P}) - \varepsilon(f_i, \bar{P}) \quad - (3.6.3.4)$$

$$\Delta\varepsilon = \sum_{j=1}^{n} D_{ij}\, \delta P_j \quad - (3.6.3.5)$$

Now we can formulate our inverse matrix problem to find the change in parameters vector $\Delta\bar{P}$

(equation 3.6.3.5) .required to achieve the desired target response. The goal is to estimate from

the sensitivity gradient matrix the required change in parameter needed to achieve a given error

vector $\Delta\varepsilon$ .We got to compress and level the average error to get to the mean desirable target value.

Halving the error with each iteration is an acceptable solution, as reduction of error by a big margin

in a single iteration can cause numerical instability and noise. Let us define the future error vector

$\overline{\varepsilon_{future}}$ as half the current error vector $\overline{\varepsilon_{current}}$ (equation 3.6.3.6) .

$$\Delta\overline{\varepsilon_{future}} = \frac{1}{2}\sigma_i d_{avg} - \overline{\varepsilon_{current}}(f_i, \bar{P}) \quad - (3.6.3.6))$$

Due to the nonlinear nature of the optimizations three cases can occur, the underdetermined case

where the extremums are less than the parameters, the over determined case where the extremums

are more than the parameters and the exactly determined case where the extremum's are equal to

the parameters. Using a Penrose pseudo matrix inversion, we can estimate our desired change in

parameters. The underdetermined case can be solved (equation 3.6.3.7) using the half error

approach, and the overdetermined case can be solved using the modified least squares pseudo-

inverse equation (equation 3.6.3.8), and occasionally when this inverse problem encounters

instability it can be reduced to an under determined or exactly determined problem by deleting some of the extremums. The exactly determined problem can be solved using a square nonsingular matrix of $m \times m$ dimensions with direct matrix inversion. These methods can reduce the error curve to determine the optimum parameters till numerical errors and residual matrix noise effects start to dominate and render the optimization technique useless.

$$\Delta \bar{P} = \ \mathrm{D}^T (\mathrm{DD}^T)^{-1} \Delta \overline{\varepsilon_{future}} \quad - (3.6.3.7)$$

$$\Delta \bar{P} = \ (\mathrm{DD}^T)^{-1} \mathrm{D}^T \ \Delta \overline{\varepsilon_{future}} \quad - (3.6.3.8)$$

Once the change matrix is obtained the current parameters vectors are updated to a better solution. (equation 3.6.3.9). This iteration is continued till the desired response is obtained.

$$\overline{P_{optimized}} = \ \overline{P_{current}} + \Delta \bar{P} \quad - (3.6.3.9)$$

The initial parameters to start the optimization is derived from a parameterization method known as the standard parameterization which is based on monotonic energy integrals and captures the physics behind the working of the symmetrical directional coupler, the formulation of these parameterizations for optimization are mentioned in D.W. Kammler's paper, and are not discussed here as only the basic ideas and concepts of the optimization procedure is mentioned.

3.7 Basic flow charts and working of the optimizer code.

The coupling profile generation and optimization methods mentioned in the previous section was implemented in Matlab for generating optimized symmetrical directional coupler which is used in assembling our quadrature hybrid coupler. This section contains the basic flow charts of the main

body, the flow charts of the of the three sub optimizer blocks, the initial parameter generation block and a demonstration of the optimization process. The Matlab code is attached in the appendix section.

The initial parameter generation code (illustrated in Figure 3.7.1) takes the input variables such as operating frequency band of the coupler, number of elemental sections, number of optimization iterations among other starting parameters. Then a weighting function is calculated based on the standard parametric form. The weighted function is to compensate for the limited length of the coupler. A set of monotonic integral functions is generated which gives the initial estimate of the starting coupling profile needed for the spline parameterized optimization.

The main body of the optimizer code (illustrated in Figure 3.7.2) takes in parameters such as maximum coupling, dimensions of the dielectric laminate, dielectric constant among other properties and generates the initial parameters from the initial parameters generator code. Coupling error curve of the coupling profile is estimated and the extremum points are picked out and the error deviation calculated. Now the optimizer sends the coupling profile to three different optimization blocks based on the error matrix size and number of extremums. The blocks are the under determined optimizer, exactly determined optimizer and the over determined optimizer. These sub blocks optimize the profile giving a refined coupling profile output back to the main function. Once the optimization process is completed the optimizer exits with the optimized coupling profile as the output.

The under determined optimizer (illustrated in Figure 3.7.3) gets called from the main optimizer function and calculates the sensitivity and error matrix using a pseudo Penrose matrix inverse and halves the coupling error with each function call. In case the coupling error is not halved the target

50

error reduction target is halved to reduce numerical instabilities and the process is retried. This process quits either when maximum iterations limit is reached or when the optimization is successful. The output of this function call is a refined coupling profile.

The exactly determined optimizer (illustrated in Figure 3.7.4) gets called from the main optimizer function and calculates the sensitivity and error matrix using a linear matrix inverse process and halves the coupling error with each function call. In case the coupling error is not halved the target error reduction target is halved to reduce numerical instabilities and the process is retried. This process quits either when maximum iterations limit is reached or when the optimization is successful. The output of this function call is a more refined coupling profile.

The overly determined optimizer (illustrated in Figure 3.7.5) gets called from the main optimizer function and calculates the sensitivity and error matrix using a pseudo Penrose inverse matrix inverse process and halves the coupling error with each function call. In case the coupling error is not halved the target error reduction target is halved to reduce numerical instabilities and the process is retried. In some cases, the matrix is completely unstable and unusable caused by the over determined matrix inversion. When this stage is encountered some of the extremums are deleted to make the inverse matrix process into an exactly determined optimization and the process is passed onto the exactly determined optimizer block. This process quits either when maximum iterations limit set is reached or when the optimization is successful. The output of this function call is a more refined coupling profile.
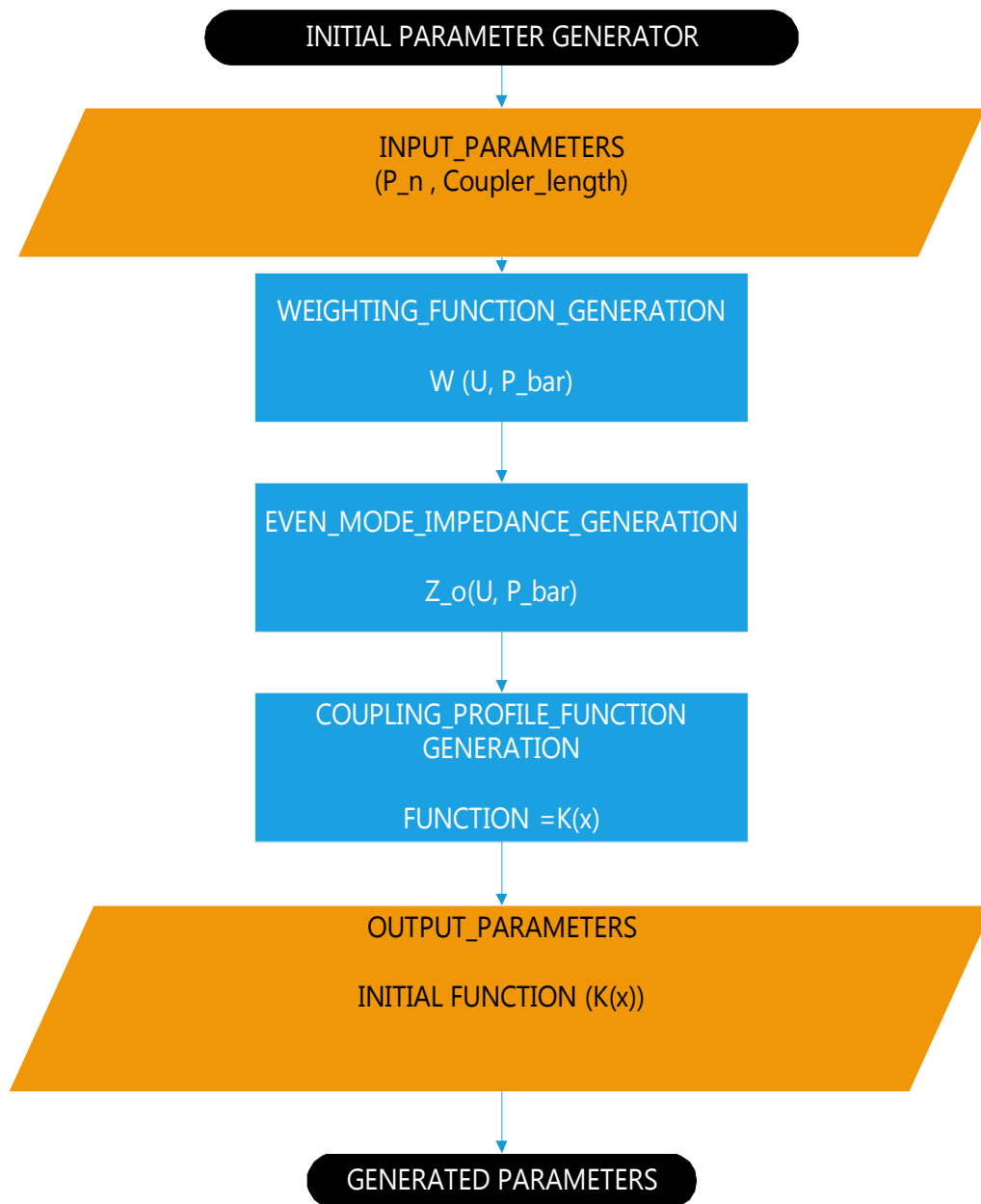
Figure 3.7.1: Flow chart of the initial parameter generator based on standard parameterization
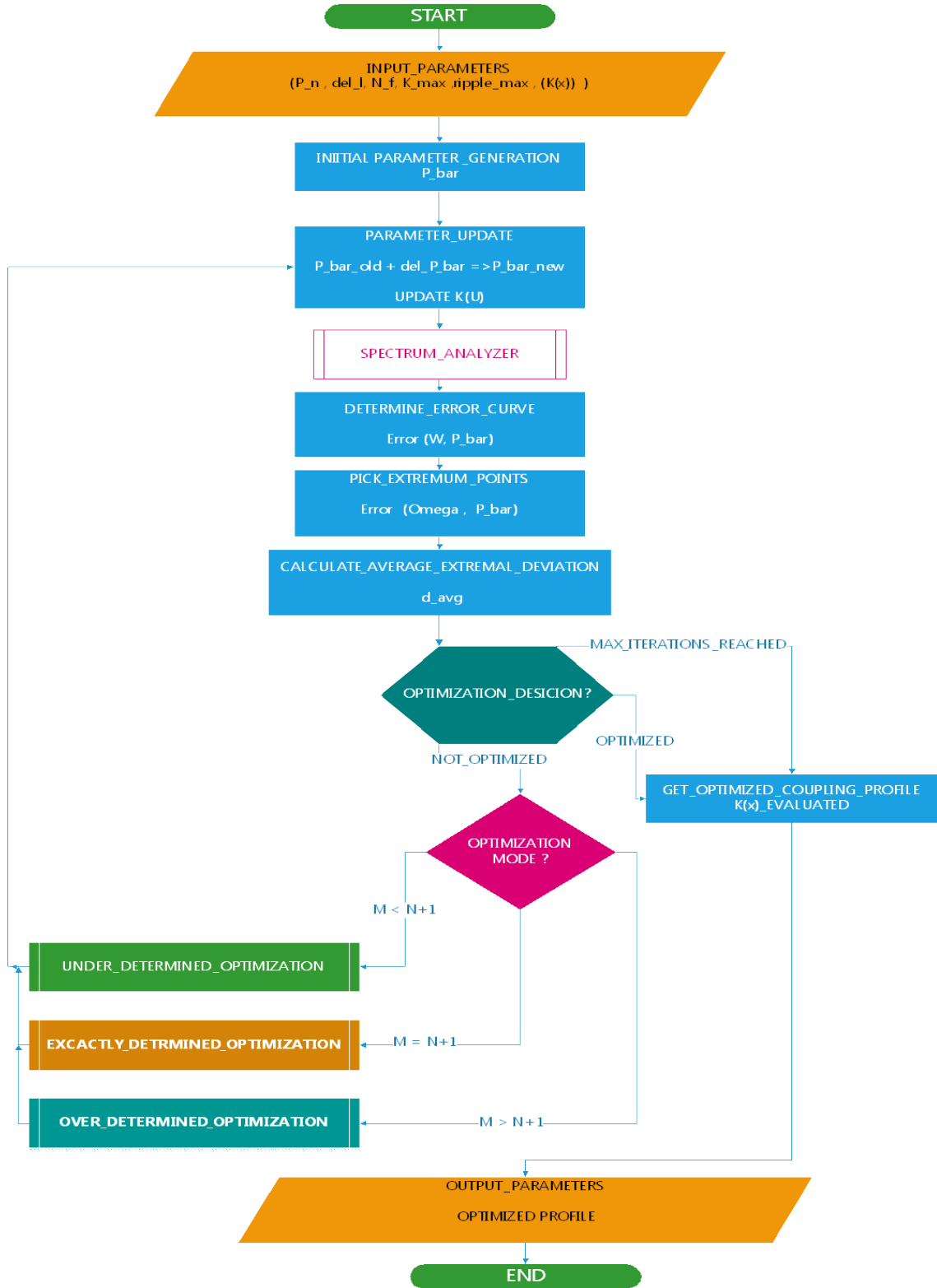
Figure 3.7.2: Flow chart of the main optimizer code employing a spline parameterization procedure
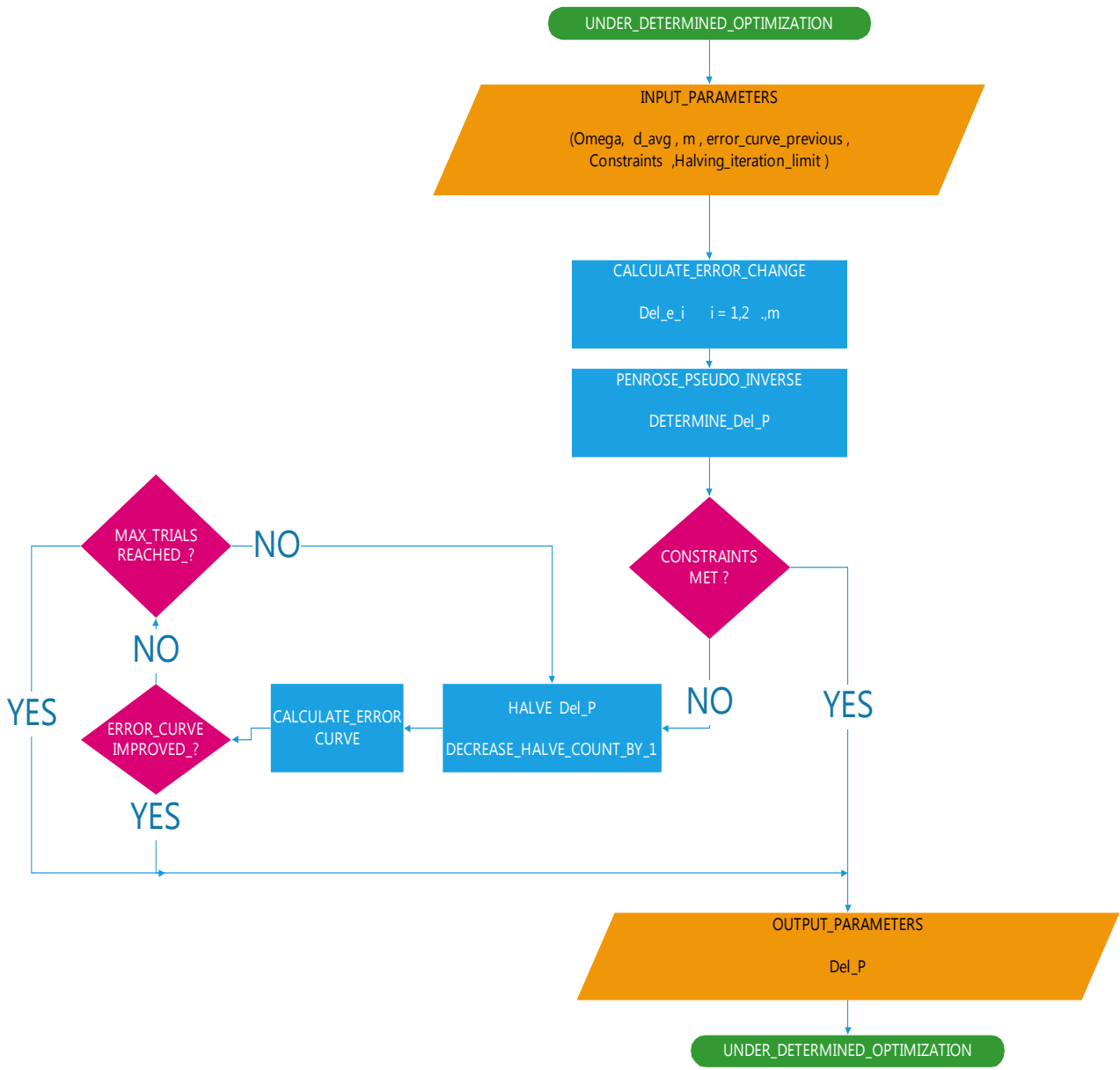
Figure 3.7.3: Flow chart of the underdetermined optimizer block
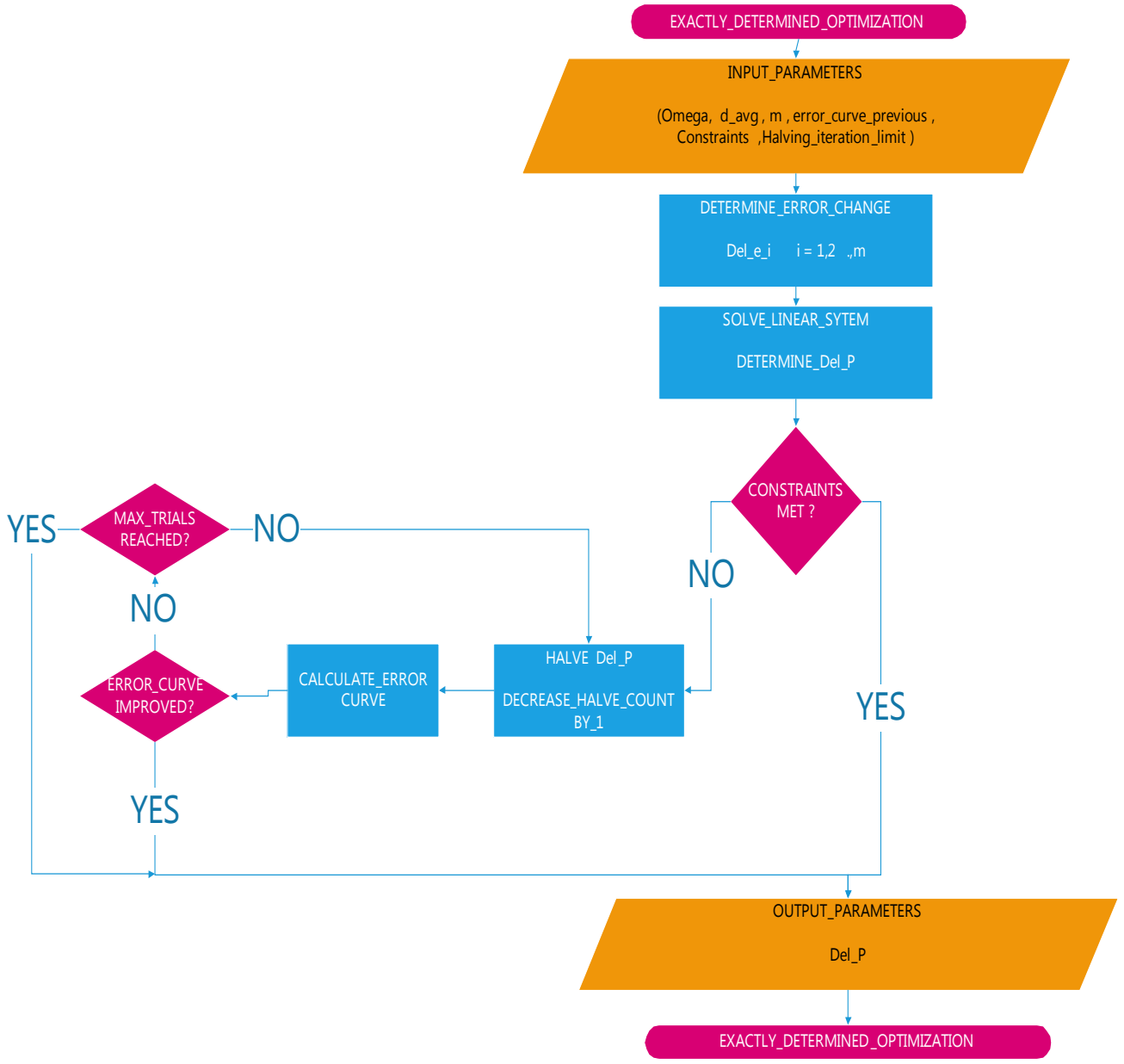
Figure 3.7.4: Flow chart of the excatly determined optimizer block
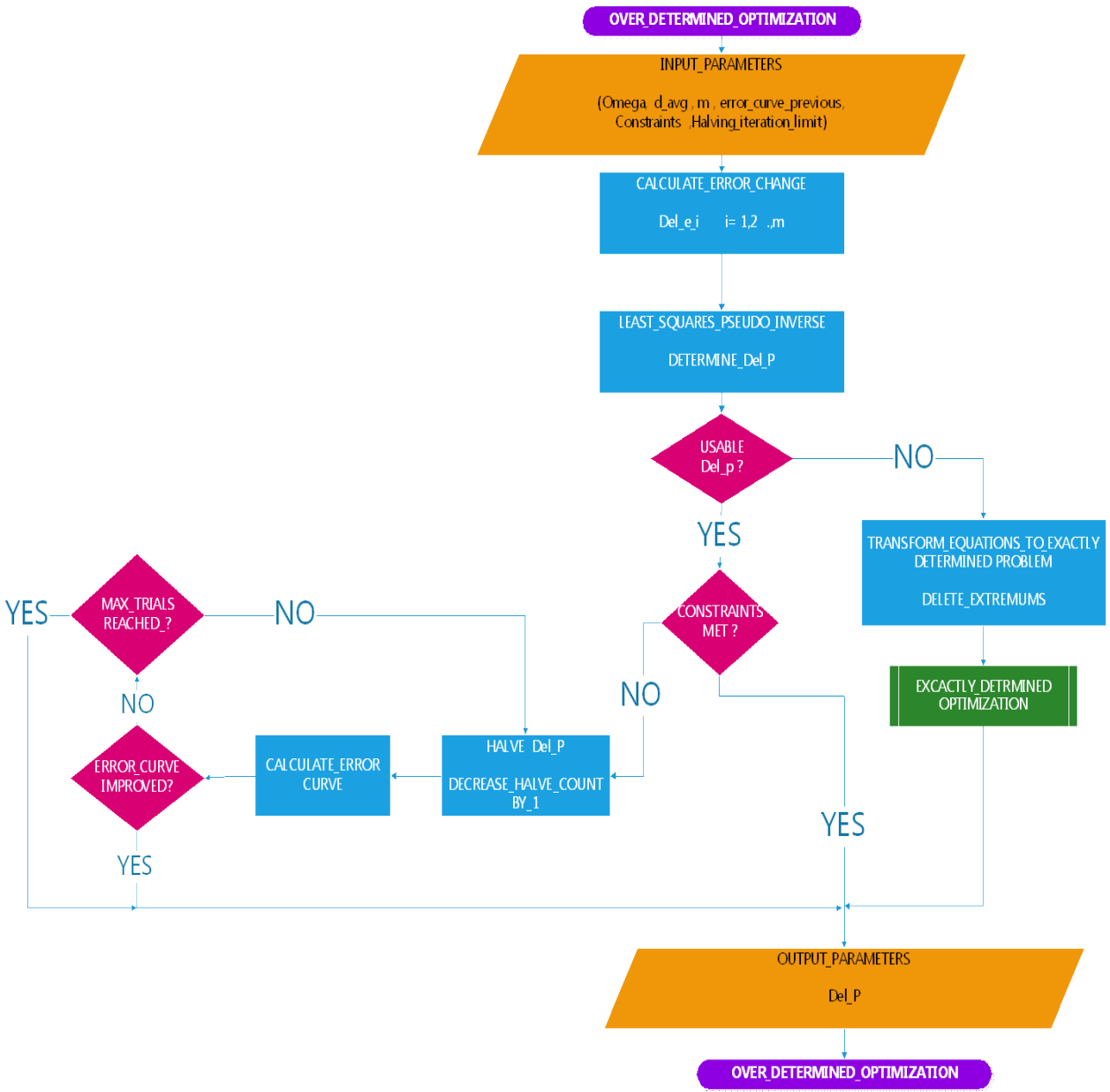
Figure 3.7.5: Flow chart of the overly determined optimizer block

56

3.7.6 Optimizer working demonstration

In this subsection, we show the working demonstration of the optimizer code for a 5 iteration run of a sample 13 quarterwave length -8.343-dB symmetrical coupler having a total of 26 quarter-wave sections. The optimization is performed for the 2-20 GHz operational bandwidth. The figures show (illustrated in Figure 3.7.6.1) how the coupling arm power response tends to approach the desired response as the iteration runs progress. Also given is the corresponding optimized coupling strength profiles.
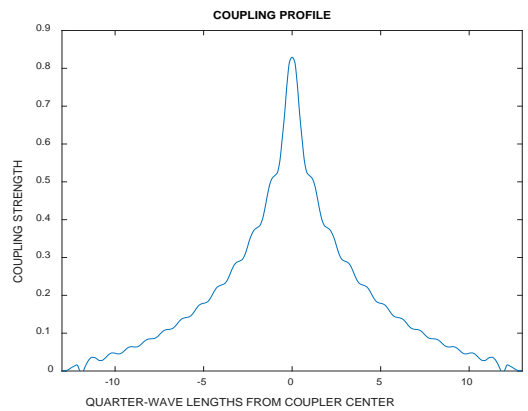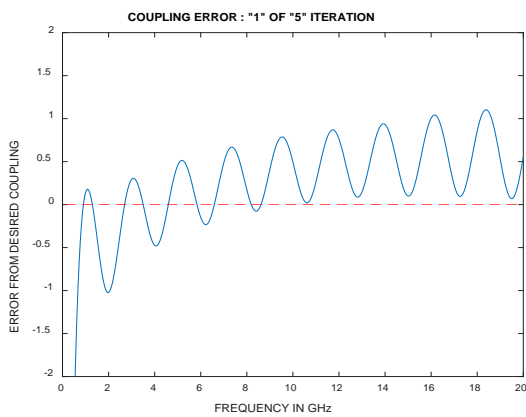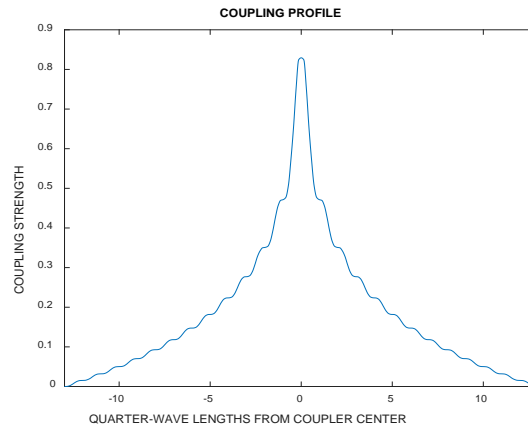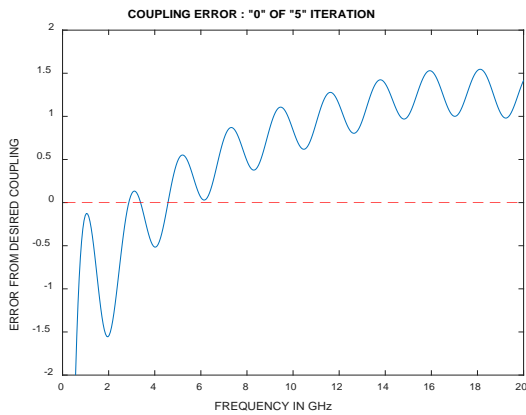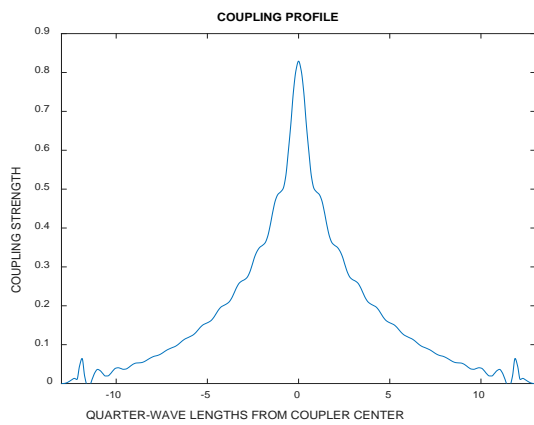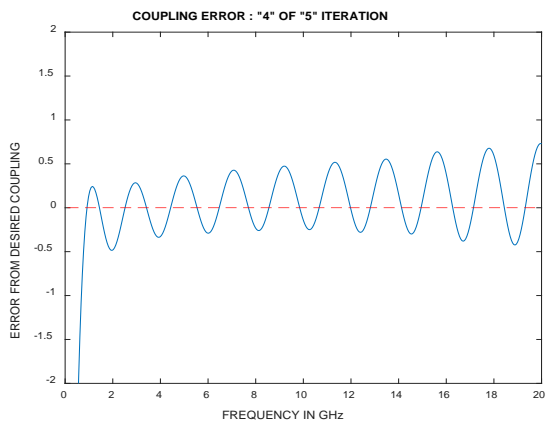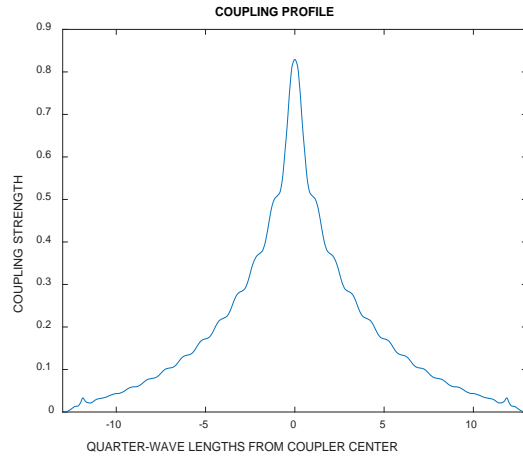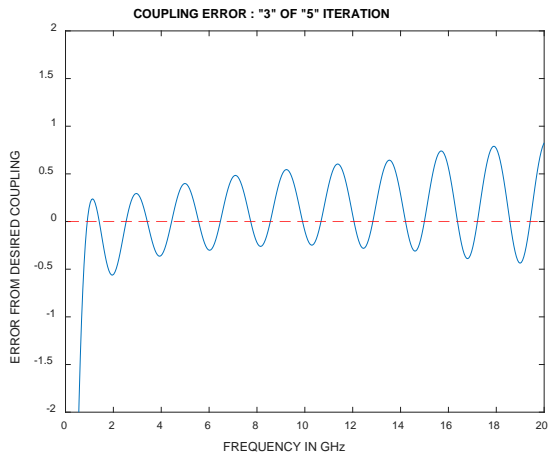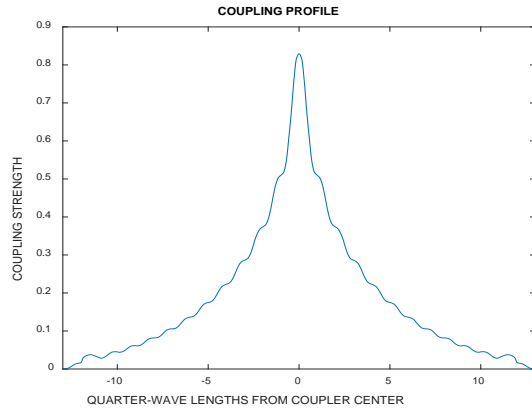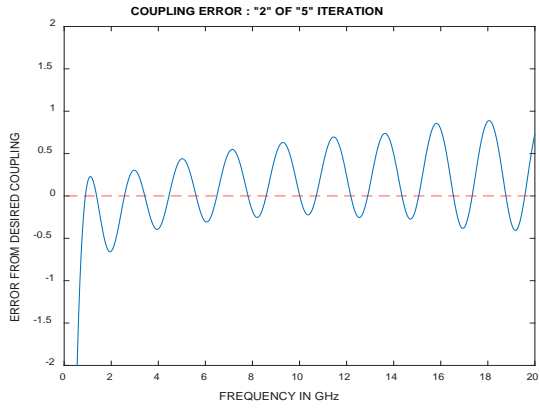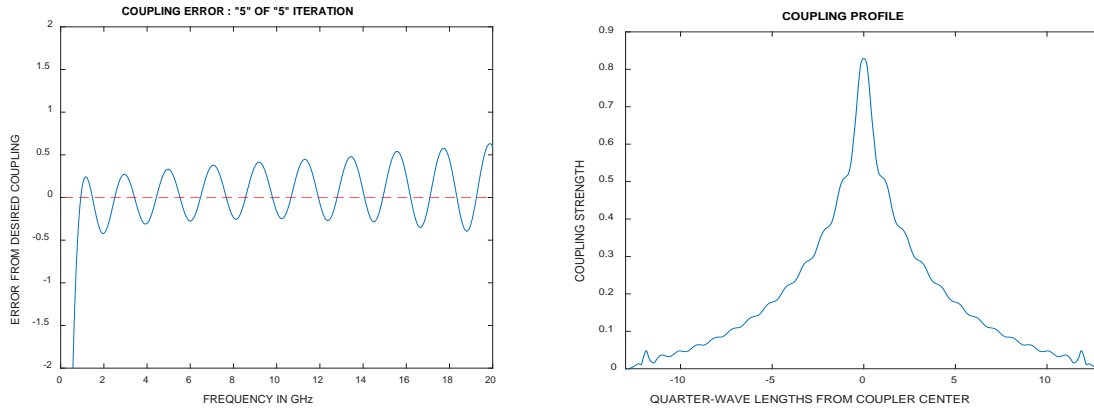
Figure 3.7.6.1: Coupling error and coupling profile for different iterations to demonstrate the spline optimization procedure. The coupling error changed from 3-dB in the 0th iteration to under 1-dB in the 5th iteration.

The initial extremal error in the coupling power response is about 3-dB as seen from the figures (illustrated in Figure 3.7.6.1), and after 5 iterations the error has been reduced to about 1-dB, the error reductions in the beginning is drastic and iterations are faster and slows down during the later iterations as the error matrix starts to get dominated by residual noise and numerical instability effects which gives erroneous results. Once the inverse matrix can no longer be estimated accurately the optimization process ends and the corresponding coupling profile is the best possible minimal error coupling response for that problem setup. Different numbers of parameters have to be tried with to see which captures the most degrees of freedom for a given number of quarterwave lengths.

3.8   Coupling strength to dimensions using equations and its limitations

Once the parameterized coupling strength profile has been optimized the next step is to convert the profile to its corresponding broadside stripline dimensions in terms of corresponding trace widths and conductor offsets.

One of the limitations with the methods in current literature is exact formulas to cover the cases of general broadside offset coupled stripline from coupling coefficient to dimensions. There exists Shwartz-Christoffel transformations and other closed elliptical integral methods [35] [36] which either is a quasi-empirical approach or assumes zero conductor thickness and completely disregards conductor strip side fringing capacitance from the traces, although these approaches are acceptable for low coupling values at the edges of the coupler, at the center sections these formulations could give erroneous estimates of more than 10%. Thickness correction and compensation empirical formulas exists [37] but only for broadside coupled stripline conductors and not for the general offset lines as required in our case. Other approaches include finding the charge on the conductors and determining the capacitance and corresponding even and odd mode impedances, but these have no upper and lower bounds hence are not variational expressions and are prone to bounding errors. What we need now is a way to accurately determine odd and even mode impedance for broadside coupled offset stripline with conductor strips of finite thickness to physically realize the coupler.

3.9   Analysis of coupled TEM line conductors using a two dimensional finite difference/finite elements electro-magnetic solver.

R.R Gupta [38] has suggested a way to accurately determine the odd and even mode impedance of coupled TEM conductors by using finite differences [39] method to solve for a two dimensional Laplace equation of the form $(\nabla^2 \emptyset_{e/m} = 0)$ where $\emptyset_e$, $\emptyset_m$ are the scalar electric and magnetic potentials in the dielectric region which can be applied to analyze any arbitrary coupled TEM lines of finite conductor thickness and offsets with sufficiently good estimation accuracy [40, 41]. In this section we go through briefly the approach of using finite elements/finite difference methods to determine the odd and even mode impedance using variational methods. Although the

underlying mathematical concepts remain similar we have used finite elements over finite differences as Matlab is better suited for finite element meshing that we require. Each elemental section of the optimized coupler is projected onto a two dimensional space and analyzed. We make some reasonable assumptions in order to design this two dimensional electromagnetic solver.

- The signal and ground plane conductors are perfect conductors and no electric fields can exist inside the conductors which forces boundary conditions at the conductor dielectric interface.

- No cross coupling exists between the elemental conductor sections, i.e. the field lines stay within the plane of the two conductor element pair. This effect is generally negligible and won't distort our results noticeably.

- The dielectric material is perfect, isotropic, homogeneous and lossless and assumes the Laplace condition $(\nabla^2 \emptyset_{e/m} = 0)$ is true always at all points in the medium.

A pair of coupled TEM lines can be analyzed in terms of its odd and even modes where the odd mode occurs when the direction of currents is opposite in the coupled TEM mode conductors and the even mode occurs when the currents are along the same direction, all other combinations in between are a weighted and linear combination of these modes. The impedances of these modes are related to the corresponding static capacitance and inductances (equation 3.9.1 and 3.9.2) where C_{oe} and C_{oo} are unit length static capacitance of odd and even modes and where L_{oe} and L_{oo} are unit length inductance of odd and even modes respectively.

$$\sqrt{\varepsilon_r}\, Z_{oe} = \frac{120\pi}{C_{oe}/\varepsilon} = \frac{120\pi}{\mu_0/L_{oe}} \quad -(3.9.1)$$

$$\sqrt{\varepsilon_r}\, Z_{oo} = \frac{120\pi}{C_{oo}/\varepsilon} = \frac{120\pi}{\mu_0/L_{oo}} \quad - (3.9.2)$$

The mode impedances can also be written it terms of characteristic impedances Z_o (equation 3.9.3) and coupling coefficient between the conductors k (equation 3.9.4).

$$Z_o = \sqrt{Z_{oe}\; Z_{oo}} \quad - (3.9.3)$$

$$k = \frac{Z_{oe} - Z_{oo}}{Z_{oe} + Z_{oo}} \quad - (3.9.4)$$

Since we require the unit length even and odd mode capacitances and inductances to estimate the corresponding impedances, a variational formula involving integral over a closed region (equation 3.9.5) is used to get these estimates. Where W_e and W_m are the unit length electrostatic and magnetostatic energies. dA is the cross section differential area, V and I are the voltages and currents on the conductors and is normalized to 1Volt and 1 Amp for calculations.

$$2C = \frac{2\, W_e}{V^2} = \frac{\varepsilon}{V^2} \iint (\nabla\emptyset_e . \nabla\emptyset_e) dA \quad - (3.9.5)$$

$$2L = \frac{2\, W_m}{I^2} = \frac{\mu_0}{I^2} \iint (\nabla\emptyset_m . \nabla\emptyset_m) dA$$

The approximate scalar electrostatic $\emptyset_e$ and magnetostatic fields $\emptyset_m$ are calculated by solving for $(\nabla^2 \emptyset_{e/m} = 0)$ in the dielectric region numerically by finite elements methods and by applying boundary conditions at the conductors and the ground plates. The magnetostatic fields (equation 3.9.7) gives an under estimated energy as compared to the true solution and the electrostatic scalar field (equation 3.9.6) gives an over estimated energy which leads to the variational upper and lower bounds that is desirable, which then can be averaged to get an estimate of the true value

of the desired impedances (equation 3.9.8). The error on these estimates can be reduced by having a fine finite element meshing at the strong electric field regions.

$$\sqrt{\varepsilon_r}\, Z \geq \sqrt{\varepsilon_r}\, Z^L = \frac{120\pi}{\frac{C^U}{\varepsilon}} = \frac{240\pi V^2}{\iint (\nabla\emptyset_e . \nabla\emptyset_e)dA} \quad - (3.9.6)$$

$$\sqrt{\varepsilon_r}\, Z \leq \sqrt{\varepsilon_r}\, Z^U = \frac{120\pi}{\frac{\mu_0}{L^U}} = \frac{60\pi \iint (\nabla\emptyset_m . \nabla\emptyset_m)dA}{I^2} \quad - (3.9.7)$$

$$\overline{Z_{estimate}} = \frac{Z^U + Z^L}{2} \quad - (3.9.8)$$

3.10   Finite elements meshing and solving

In this section we go through a basic demonstration of how finite elements method is used to create a 2D planar EM solver. The meshing we chose was triangular as they seemed to produce lesser numerical singularities and error when the partial differential equations are solved as compared to square or other polygonal meshing methods. The meshing can be sparse at the low field strength area far from the TEM conductors and ground plane and has to be concentrated at the areas of strong electric fields between the conductors and the sides of the conductor traces to account for fringing field capacitance. We, however, used uniform fine meshing (illustrated in Figure 3.10.1) throughout the area as the availability of computing resources was not an issue. No meshing is needed inside the conductors as we assume perfect boundary conditions at the dielectric-conductor interfaces. A magnetic wall boundary condition is placed sufficiently apart at the sides to limit the area of calculation and also to have a closed region to solve the equations. The magnetic wall is set at about 2-3 trace widths apart in either direction from the center of the coupler where it is assumed the electric fields die out exponentially from the TEM conductors and by the magnetic wall distance it should be insignificant.

The mesh is refined until its residual noise is negligible. The Laplacian partial differential equations are solved in the mesh. This is done under the assumption that there is no variation in potentials within each mesh element and only varies between the elements. Examples for solved even mode electrostatic potential distribution (illustrated in Figure 3.10.2) and odd mode electrostatic potential distributions (illustrated in Figure 3.10.3) along with an example of overlay of mesh with the electrostatic scalar potential solution of even mode (illustrated in Figure 3.10.4) is shown.
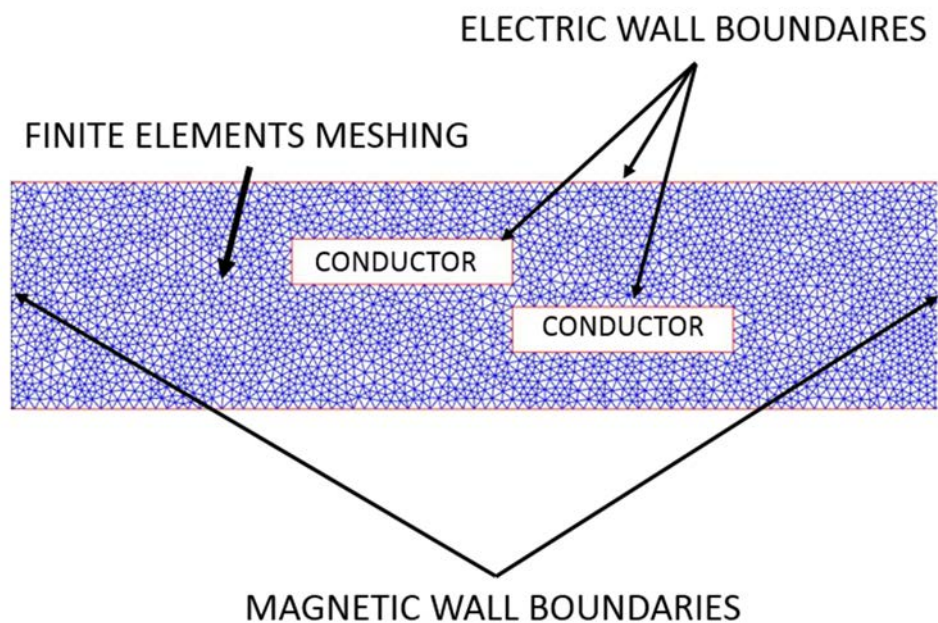
Figure 3.10.1: Illustration of the finite elements meshing done for a pair of coupled TEM conductors for the even and odd mode electro-static and magneto-static potential analysis.
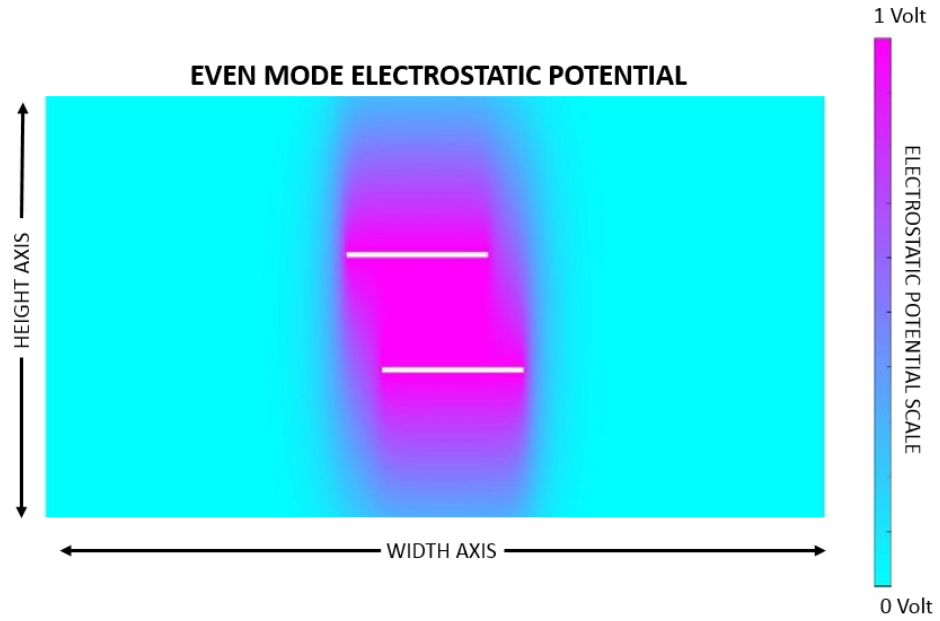
Figure 3.10.2: Calculated even mode electrostatic potential field between an elemental section coupled TEM conductor pair.
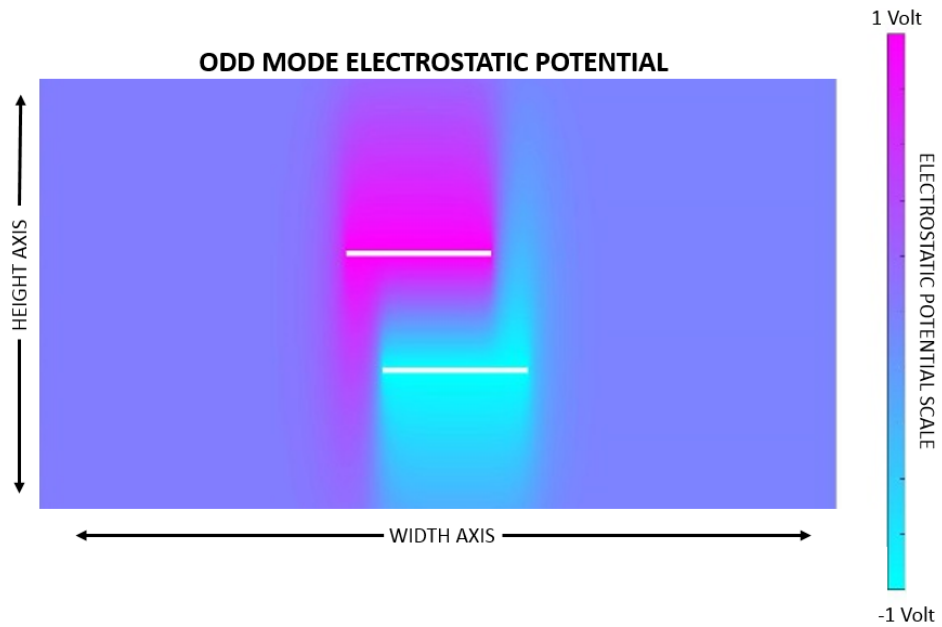
Figure 3.10.3: Calculated odd mode electrostatic potential field between an elemental section coupled TEM conductor pair.
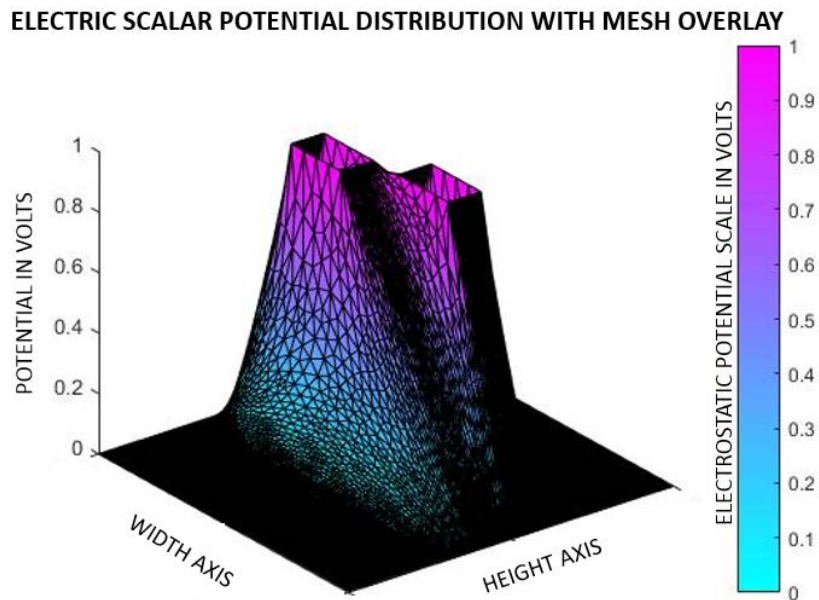
Figure 3.10.4: Computed even mode electric scalar potential distribution overlaid with finite elements mesh between an elemental section coupled TEM conductor pair.

3.11   2-D to pseudo 2.5-D EM solver/tuner to convert coupling profile to dimensions

Now as we have found a way to accurately analyze the even and odd mode impedances for a given set of elemental section TEM coupled conductors of specified thickness, offset and width, we develop an algorithm which essentially is a pseudo 2.5-D solver/tuner to tune the dimensions of the required physical coupler until it matches the optimized coupling profile within an acceptable tolerance limit.

The tuning begins with the first elemental section and an initially estimated offset and trace width is analyzed and if the coupling constant or the impedance is not within the required tolerance bound, another set of tweaked dimensions are analyzed, this continues until the dimensions produce the desired characteristic impedance and coupling constant within an acceptable tolerance.

66

Then the next elemental TEM section is taken and analyzed, here the dimensions do not change drastically from the previous section due to the tapered nature of the sections and hence the convergence to desired dimensions is much faster. This process continues till all the elemental section dimensions are tuned. A set of vectors of trace widths and offsets is then obtained which can be used to formulate the physical coupler. The basic flow chart of the algorithm is given (illustrated in Figure 3.11.1).
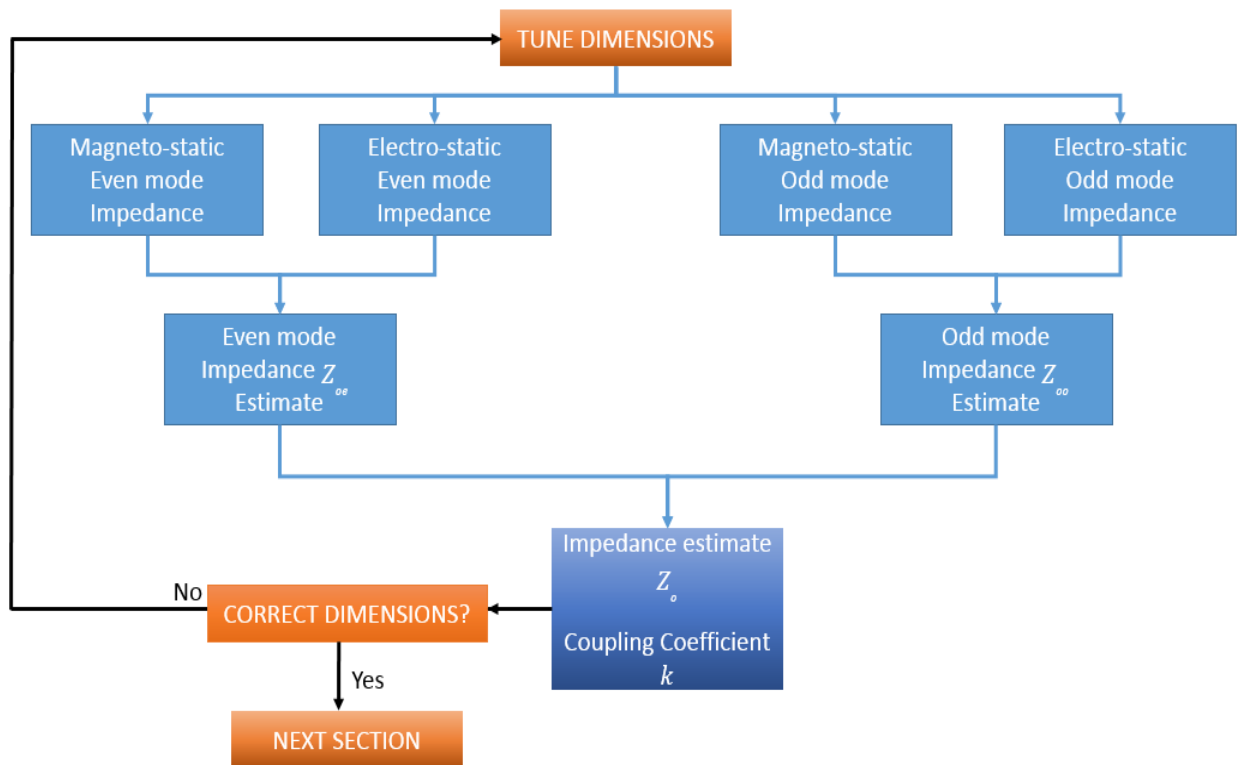
Figure 3.11.1: Flow chart of the algorithm employed in the generation of the coupler dimensions

Chapter 4: Physical realization of the optimized quadrature hybrid coupler

Once the symmetrical directional coupler has been designed and optimized, the next step was to physically realize the quadrature hybrid coupler. In this chapter we describe the three-dimensional (3-D) electromagnetic model validation of our design as well as additional compensational structures to improve phase, return loss and isolation properties of the quadrature hybrid coupler. We also discuss the issues related to mechanical integration, material imperfections, fabrication tolerances and other non-ideal effects in the real world and how to best compensate for and/or mitigate some of them to meet the design specifications.

4.1 Full-wave EM simulations and verification of the developed symmetric directional coupler and the formulated quadrature hybrid

We used the techniques and algorithms explained in the previous chapter (Section 3.6) to generate a 10 quarterwave sectional $-8.343\,dB$ symmetrical directional coupler after a 7 iteration optimization process of the symmetrical coupler with the total number of sections in each symmetrical coupler being twenty. Ten sections were chosen as they found to be an acceptable compromise between the design bandwidth, insertion loss, coupler length and other design considerations which was validated using the high frequency structure simulator (HFSS) 3-D electromagnetic simulator, which can account for dielectric and copper foil material losses and non-idealities not included in the design and optimization phase. The length of each of the symmetric coupler is 38 mm from end to end. The resulting optimized symmetric coupler is connected in a tandem fashion to formulate a 3-dB quadrature hybrid (Figure 4.1.1) and with the spacing in between them tuned to be such that it is not too close to induce cross coupling between the symmetric coupler outer arms and degrade the response as well as not too far apart as to add to the conductor trace length which increases device size and insertion loss. This model was then

simulated for verification in HFSS software terminated at 50 Ohm wave ports. The simulated response of the coupling arm (Figure 4.1.2) shows that both the amplitude for both arms are within 0.9-dB from -3-dB all the way to 20 GHz. The number of distinct passband ripples is seen to be about 7 and the phase quadrature imbalance (Figure 4.1.3) between the coupled and through ports is within 1º from the desired 90º, which is an acceptable tolerance. Also, the return loss (Figure 4.1.4) of all the ports is better than 20 dB in simulations, and the isolation between the input ports and its adjacent ports (Figure 4.1.5) for all the ports is found to be better than 20 dB, which are satisfy design requirements.
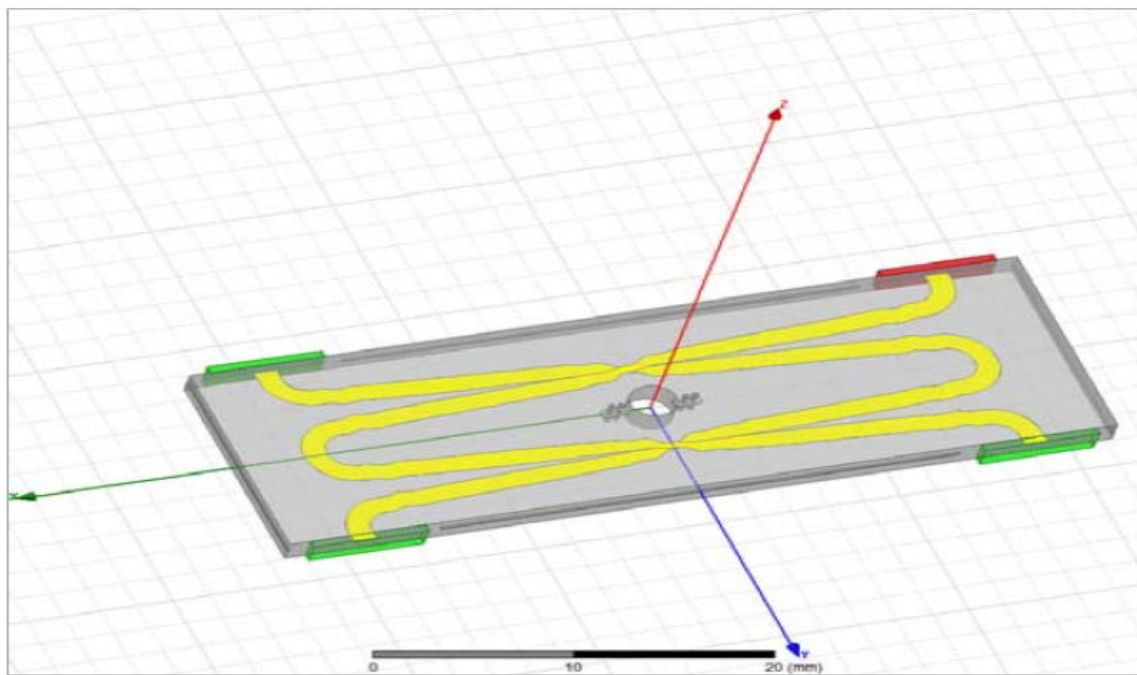
Figure 4.1.1: HFSS model of optimized 3-dB quadrature hybrid coupler

COUPLING RESPONSE
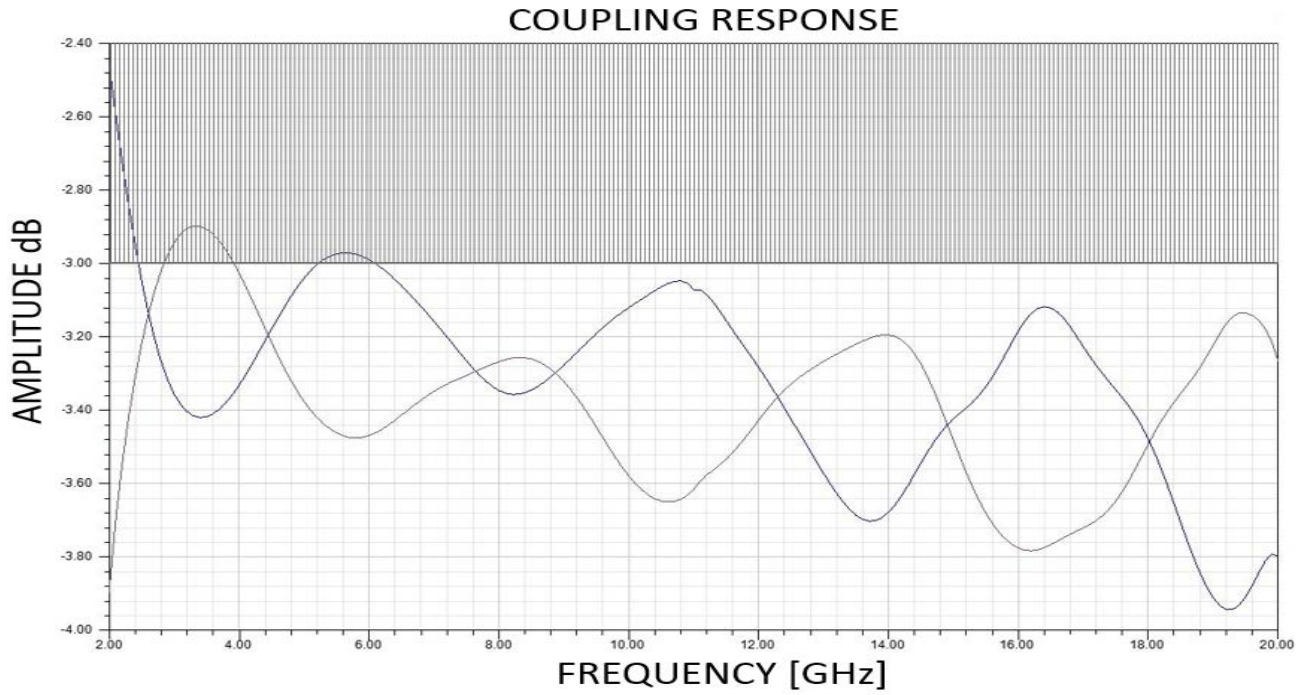
Figure 4.1.2: HFSS simulation of the coupled and through port amplitude response

PHASE QUADRATURE RESPONSE
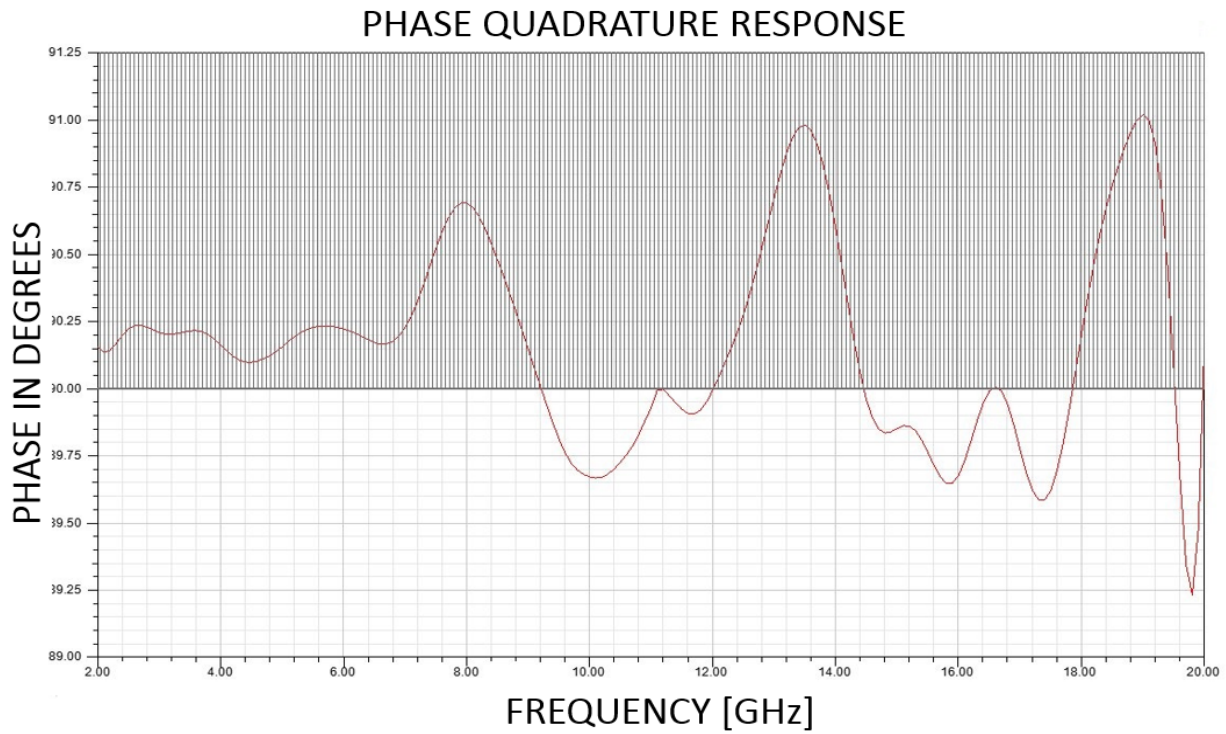
Figure 4.1.3: HFSS simulation of the phase quadrature between the coupled and through wave ports
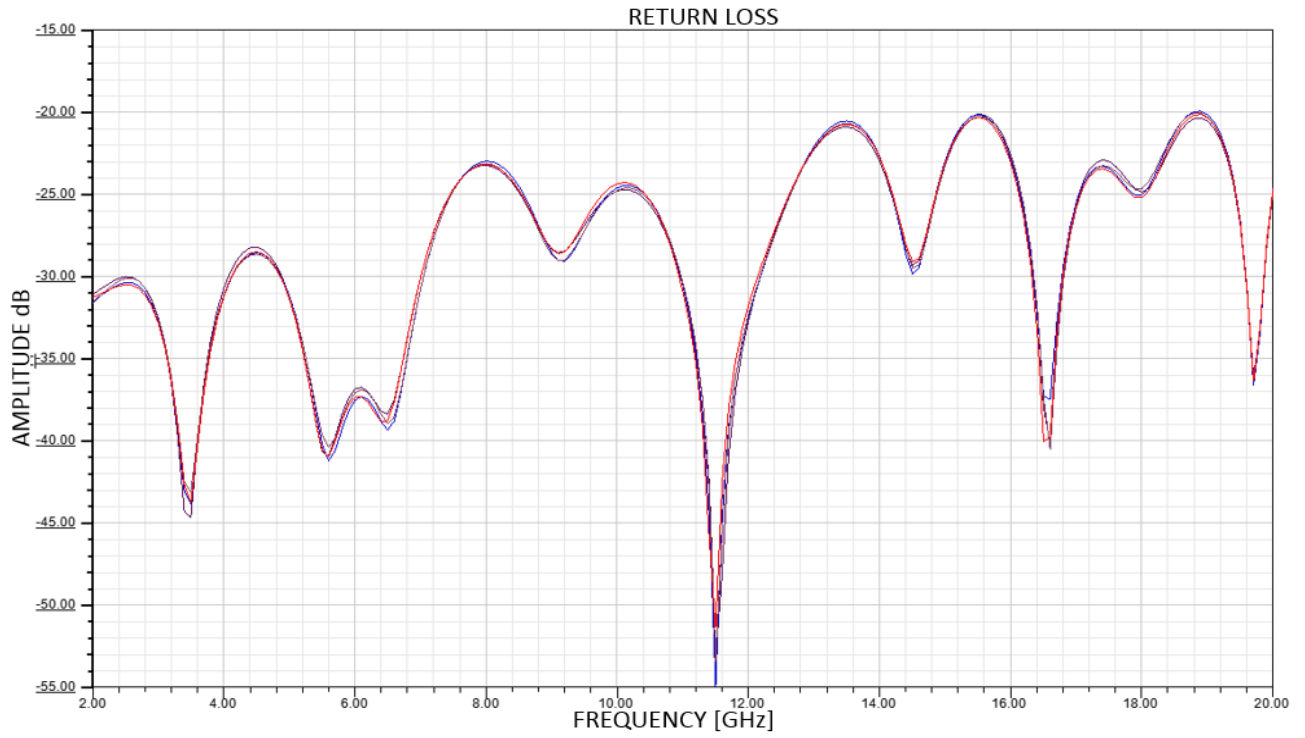
Figure 4.1.4: HFSS simulation of the return losses at the wave ports of the quadrature hybrids

Figure 4.1.5: HFSS simulation of the isolation between the wave ports and corresponding isolation wave ports

71

4.2 Phase quadrature, return loss and isolation improvements by appending compensating structures

When using the standard parametric form for optimization which is a parameterization based on monotonic energy integral functions which does sufficiently capture the physics behind the working of the coupler, an observation is made which suggests that the optimizer finds a better amplitude and phase response when there exist discontinuities spaced quarterwave lengths apart, this becomes evident as the iterations proceed, starting with the smooth response of an un-optimized coupler and ending up with stronger discontinuities as the optimization runs go. To demonstrate this observation an example of a 5 section symmetrical coupler optimization using the standard parametric profile is run (Figure 4.2.1). As seen in the figures the un-optimized coupling profile (Figure 4.2.1.a) has a smooth monotonic character, and as the iteration runs progress the discontinuities at quarter wave length spacing's begin to appear even though not completely as seen in the first iteration. By the fifth iteration (Figure 4.2.1.d) the discontinuities are well pronounced and clearly seen protruding out.

STANDARD COUPLING PROFILE: "1" OF "5" ITERATIONS

DISCONTINUITIES
START TO
EMERGE

COUPLING STRENGTH

QUARTERWAVE LENGTHS FROM COUPLER CENTER

STANDARD COUPLING PROFILE: "2" OF "5" ITERATIONS

STRONGER
DISCONTINUITIES AS
ITERATION
PROGRESSES

COUPLING STRENGTH

QUARTERWAVE LENGTHS FROM COUPLER CENTER

73

STANDARD COUPLING PROFILE: "5" OF "5" ITERATIONS
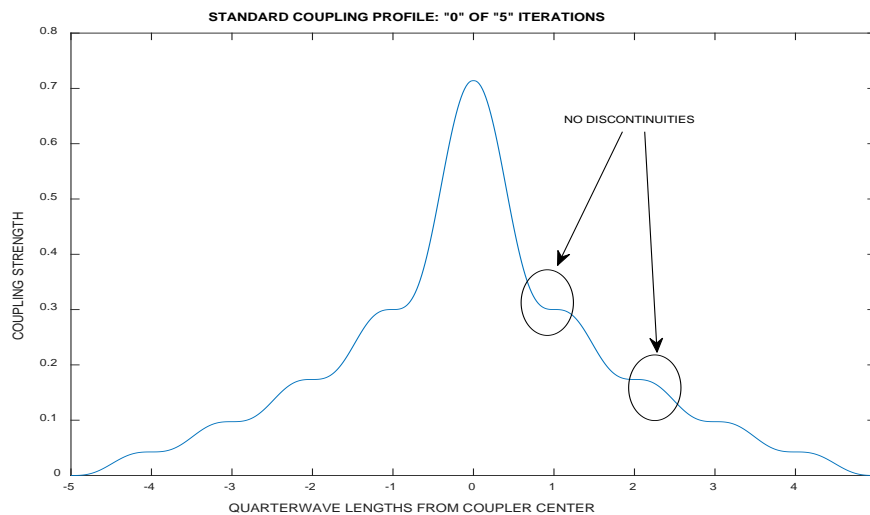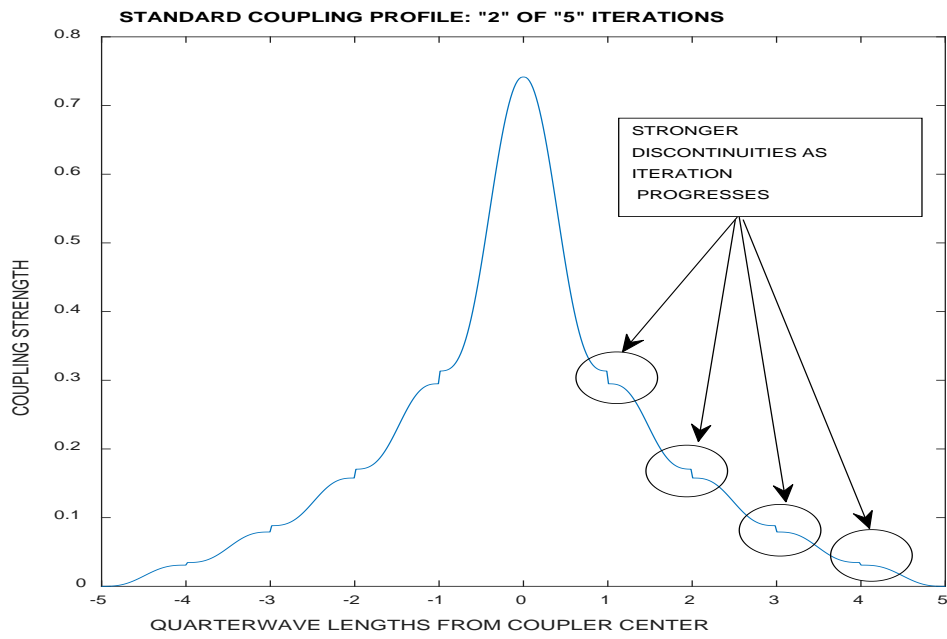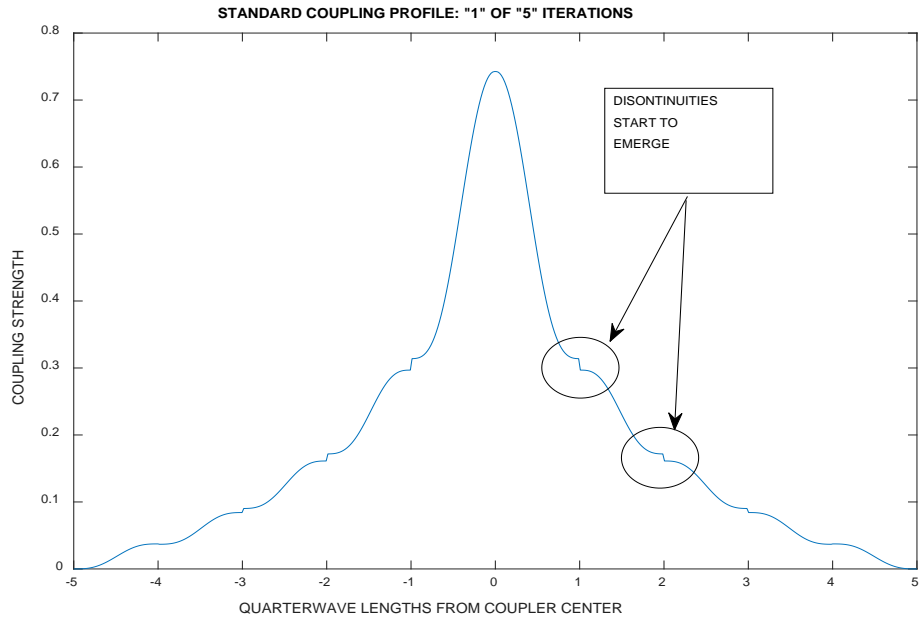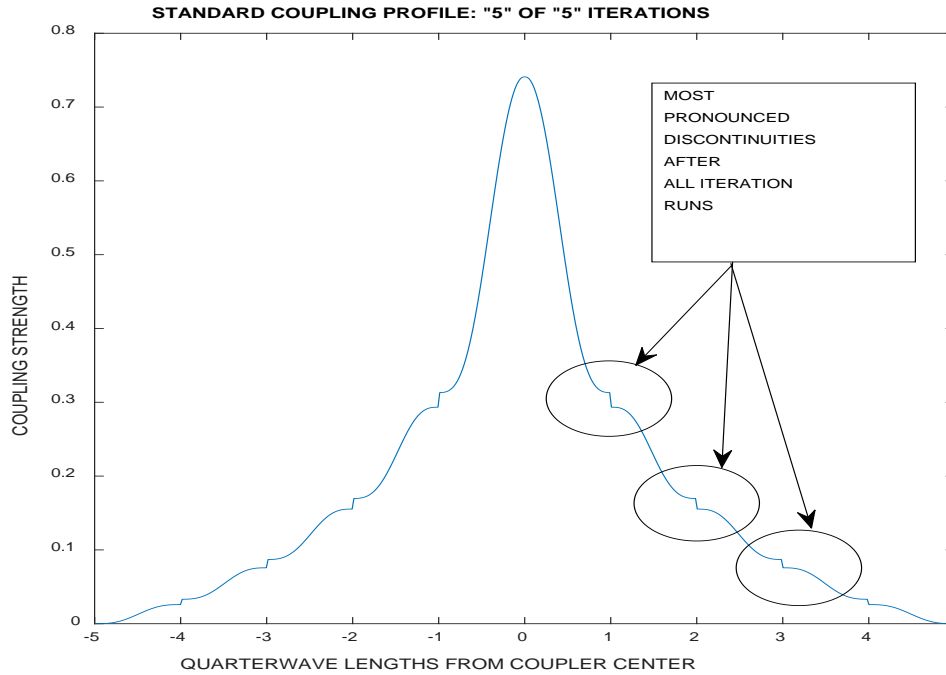
Figure 4.2.1: Discontinuities of the coupling profile during standard parameterization optimization; (a) Unoptimized standard parameterization of the coupling profile showing a smooth monotonic response; (b) Optimized standard parameterization of coupling profile for 1 iteration; (c) Optimized standard parameterization of coupling profile for 2 iteration; (d) Optimized standard parameterization of coupling profile for 5 iteration showing distinct quarterwave discontinuities

We used the above suggested idea that quarter wavelength spaced coupling profile discontinuities seems to improve the coupling arm response in the optimizer and appended stub like compensation structures on the outside of the spline optimized coupler arms (Figure 4.2.2) which were spaced quarterwave lengths apart and then tuned the lengths and widths of the structures to improve the phase quadrature response, which seems to agree in the simulations. Care has to be taken so that the protruding structures do not act like resonant short or open stubs at the frequencies of operation, these stubs were tuned and sized based on these considerations. The phase improved to within 0.5° of deviation from phase quadrature in the simulations (Figure 4.2.4), and the return loss (Figure 4.2.5) and isolation (Figure 4.2.6) improved by about 2-dB as compared to the uncompensated

quadrature coupler case. The improvement in isolation is a direct consequence of a better phase quadrature response of the device, ideal phase quadrature property by default ensures that the input signals are well discriminated and directed and don't reach the isolated ports. There was however a slight negative impact in the simulated amplitude response (Figure 4.2.3) which degraded the amplitude response of the output arms to about 1-dB from 0.9-dB in the uncompensated case and the number of evident ripples in the passband remained the same. The reason we appended the compensating structures to the outsides of the coupler rather than between the coupled conductor trace arms is to not inadvertently induce any unwanted propagating modes and fringing field effects in the strong electromagnetic field area. These phase improvement effects seen in the simulations could be explained on the fact that while a tapered coupler reduces the return loss, cuts down unwanted modes of propagation and smoothens out the ripples in the passband, quarterwave discontinuities acts like strong resonators/antenna elements which helps in better coupling between the TEM conductor elements. So this compensated coupler is technically a hybrid between a tapered and a discreet coupler taking most of the advantages of the tapered coupler as well as adding in the strong quarter-wave antenna effect from the discrete coupler approach. We decided to add in the compensation structure in our build to improve phase quadrature, return loss and isolation even though amplitude response takes a slight hit as the phase balance, return loss and isolation of the coupler was a more critical property to our application. If amplitude response is more preferred than the phase quadrature property, then the compensation structures need not be added. By this we have completed the design and verification of the required quadrature hybrid coupler. The non-idealities of cross coupling between the elemental TEM sections resulting from using the optimizer and 2D solver/tuner ideal assumptions are found to be within an acceptable tolerance.
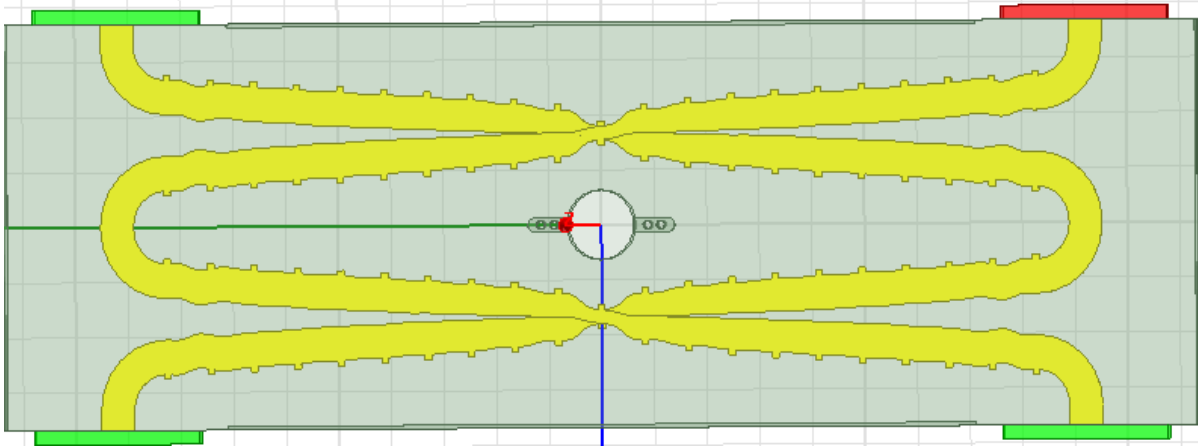
4.2.2: HFSS model of optimized 3-dB quadrature hybrid coupler with phase compensating structures and center screw hole for mounting
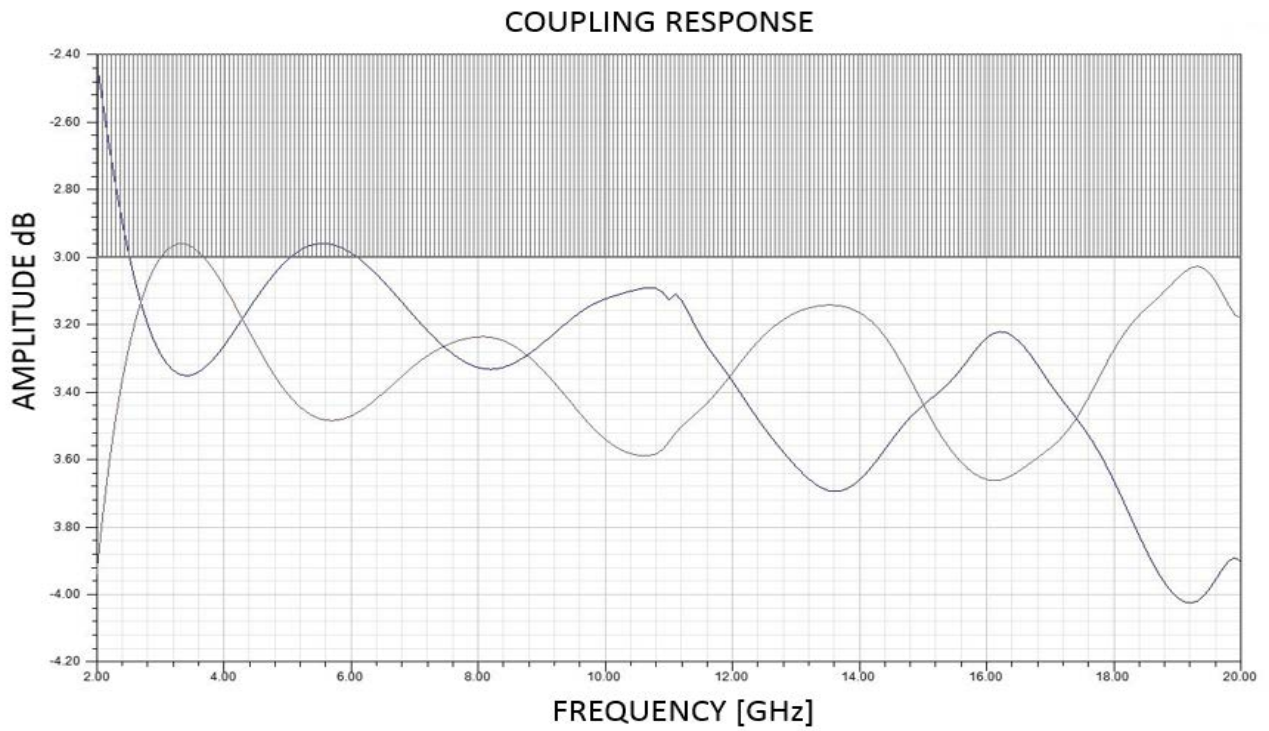
Figure 4.2.3: HFSS simulation of the coupling amplitude response of the through and coupled ports for the phase compensated quarature hybrid coupler
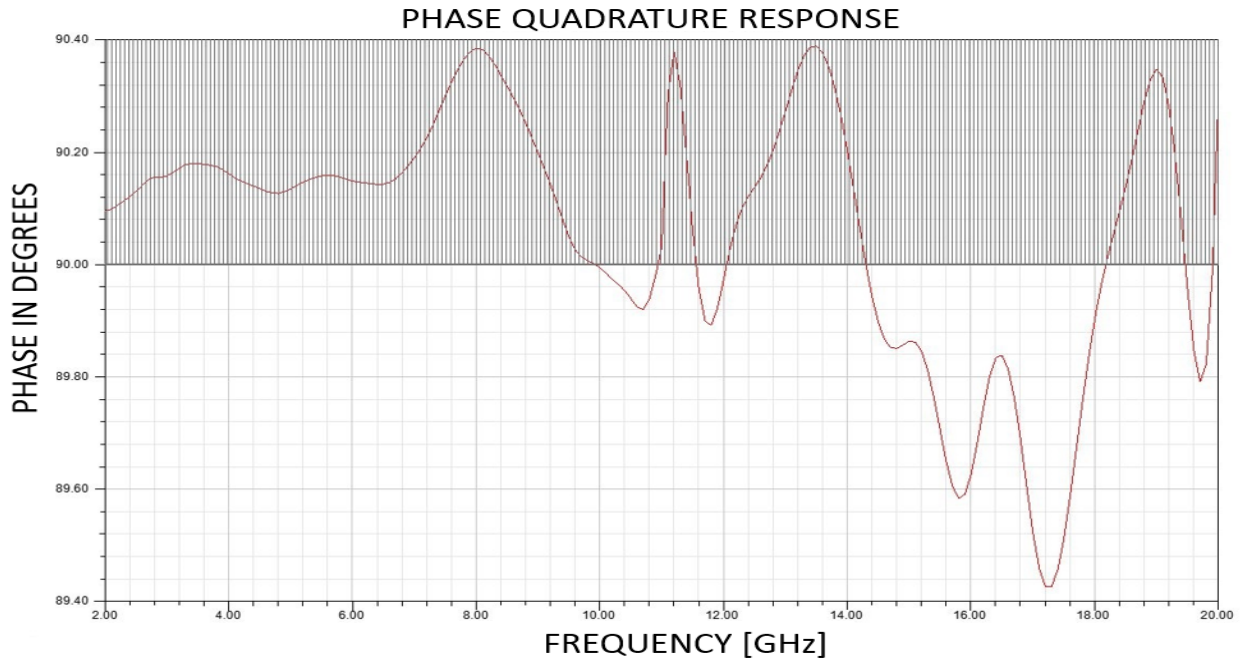
Figure 4.2.4: HFSS simulation of the phase quadrature between the coupled and through wave ports for the phase compensated quadrature hybrid coupler
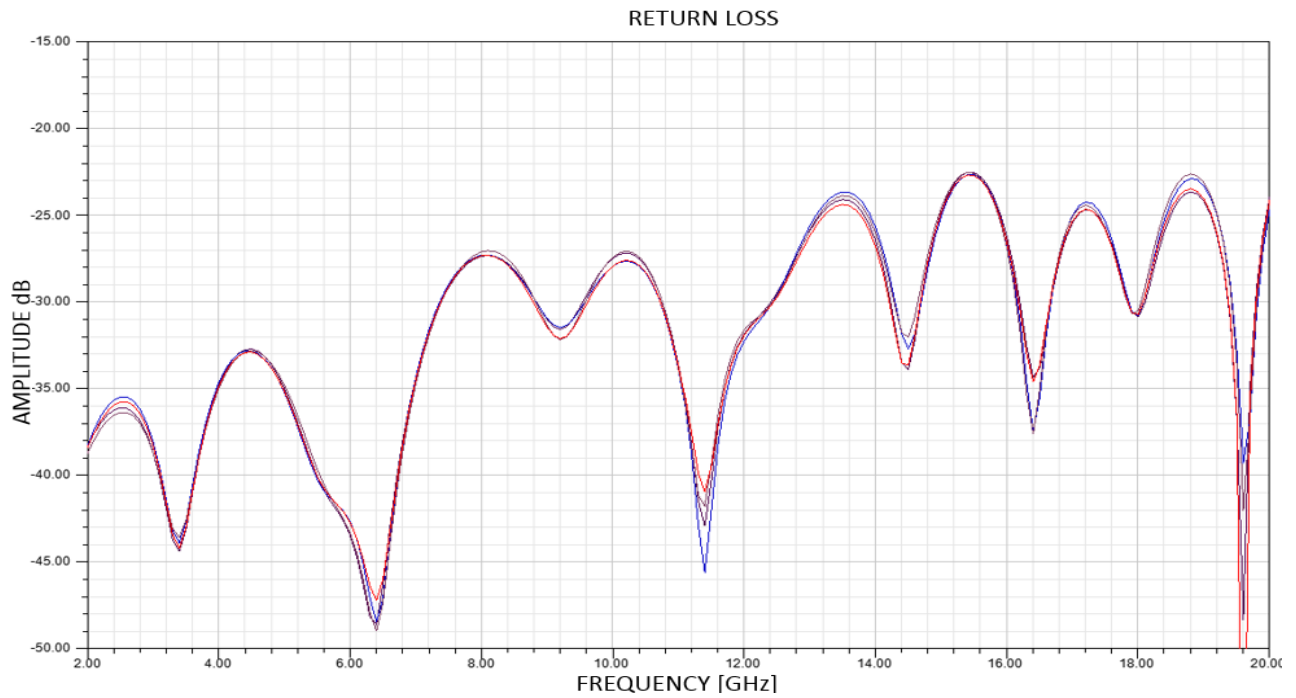
Figure 4.2.5: HFSS simulation of the return loss at the wave ports for the phase compensated quadrature hybrid coupler
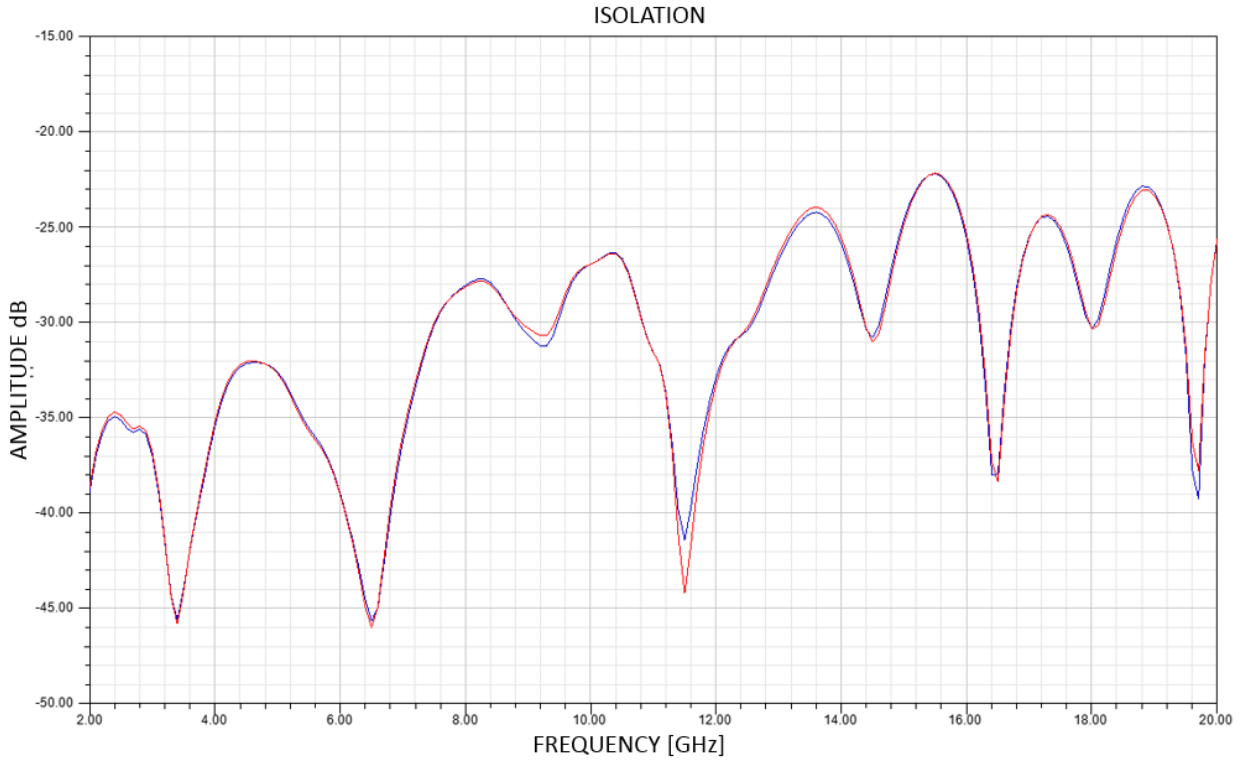
Figure 4.2.6: HFSS simulation of the isolation between the input wave ports and corresponding isolation wave ports for the phase compensated quadrature hybid coupler.

4.3   Fabrication and mechanical integration of the quadrature hybrid coupler.

The Gerber design files was sent out to a microwave PCB fabrication house with a pattern etching tolerance of ± 1 mil and a positional registration accuracy of ± 3 mils on the 3.5 mil center dielectric layer. The copper conductor trace was etched without a finish on both sides, as a finish on the copper such as ENIG (Electro-less nickel immersion gold) would be mechanically desirable to mate the conductors to an external trace or an SMA launch but would have degraded the insertion loss due to the nickel coating. Also a finish on the copper would have degraded the copper etch accuracy due to uneven coating. The copper used was reverse treated electrodeposited ½ Oz copper which was standard issue copper foil for the TLY-5 dielectric laminates.

Gluing down the boards to get the outer and middle layers to behave as a single multi-layered board was a challenge as prepregs used to glue down multi-layer boards could not be used in this application due to dielectric constant mismatch between the dielectric layers which could perturb the stripline electric fields significantly (Figure 4.3.1). Also it was important for the dielectric and the prepreg to have a CTE (Coefficient of thermal expansion) as close as possible for mechanical and performance reasons, which was not the case.
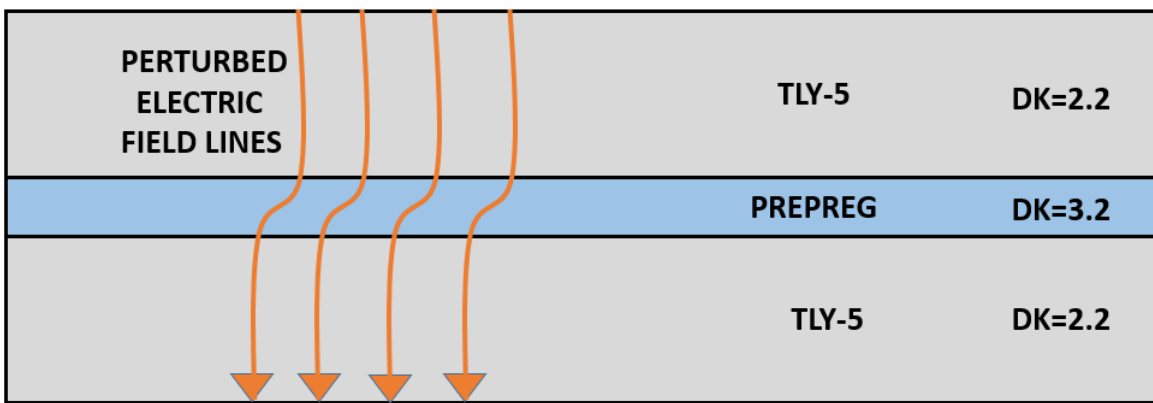
Figure 4.3.1: Perturbation of electric fields due to dielectric constant mismatch by the use of prepregs
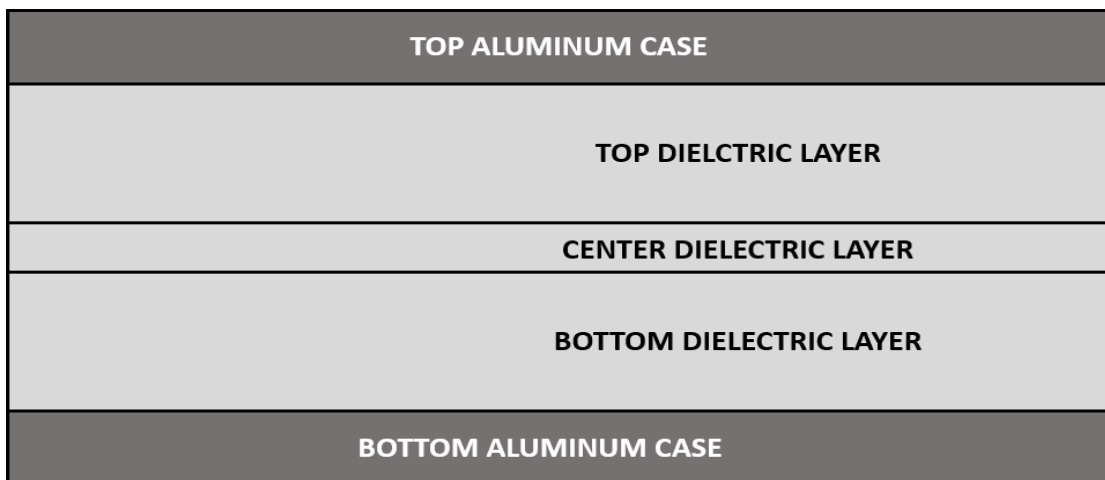
Figure 4.3.2: Mechanical integration of the quadrature hybrid by applying pressure on aluminum case

Aluminum casing was used to press down the dielectric layers (Figure 4.3.2), which would get rid of the need for prepreg use, so the dielectric layers was fabricated as separate boards (Figure 4.3.3). Additional care was taken while clamping down the aluminum casing as not to compress the Teflon dielectric material which is fairly compressible as this would reduce the ground to ground spacing unevenly which would cause deviation of the actual coupling profile from the designed profile and significantly affect performance and might cause resonance due to parallel ground plate radiation effects, which was observed during the integration. Once the coupler has been screwed down and electrical performance ensured using a VNA (Vector network analyzer) the clamping screws have to be locked down by using thread locker or epoxy resins so that the device stays fixed in place. Southwest microwave SMA female 2-hole flange connectors (214-510SF) were used to interface the coupler port traces to measurement devices. The launch pins used were 1090-05G from Southwest microwave and the port traces were rounded off at the launch interface to lower impedance mismatches with the signal launch pin.
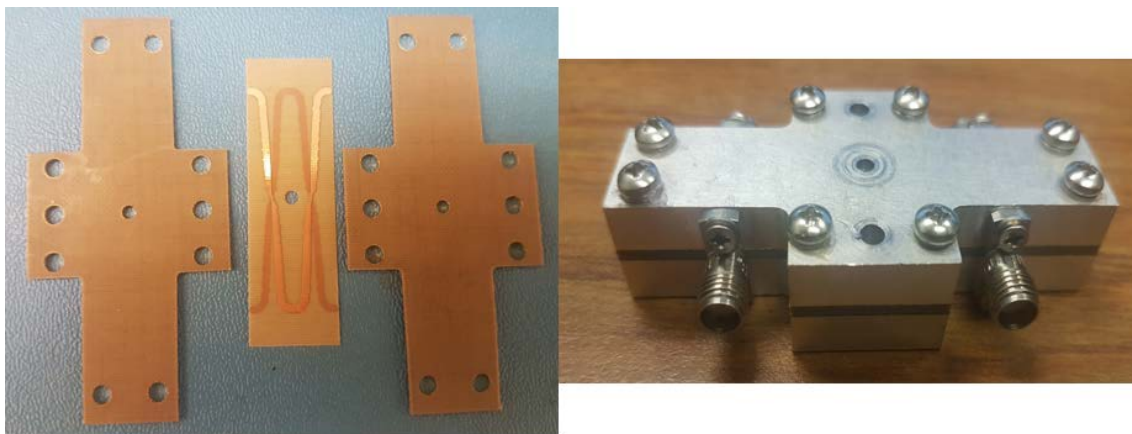
Figure 4.3.3: Photographs of the fabricated quadrature hybrid dielectric laminate layers (left) and finished coupler with aluminum casing (right)

4.4 Simulated vs measured performance of the assembled quadrature hybrid coupler with SMA launches

We simulated the performance of the assembled quadrature hybrid coupler (Figure 4.4.1) with the 2-hole SMA flange connectors with HFSS 3D models provided by Southwest microwave with assumptions of perfectly homogenous and isotropic dielectric material and smooth copper conductor arms and ground plane.
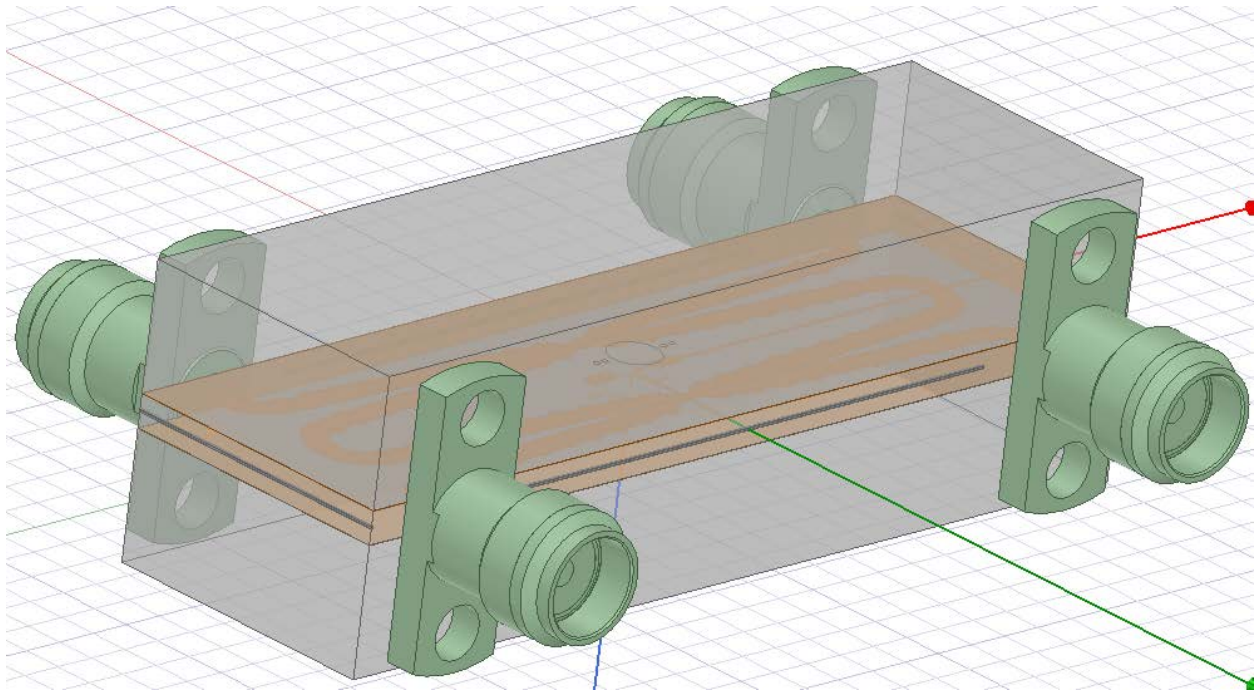
Figure 4.4.1: HFSS model of assembled quadrature hybrid coupler with 2-hole SMA flange mount female SMA connectors

The simulated performance with SMA connectors is as shown with the connectors terminated at 50 Ohm wave ports. The simulated coupling arm amplitude response (Figure 4.4.2) shows that both the output arms have increased insertion losses due to the connector and are within 1.25-dB from -3-dB all the way until 20 GHz, the number of distinct passband ripples is still seen to be about 7 but the ripples after 12 GHz seems to have some distortions and non-smoothness which

can be attributed to the SMA connector-launch pin-trace interface effects. The phase quadrature imbalance (Figure 4.4.3) between the coupled and through ports has degraded to $\pm 2^\circ$ from the desired 90°. The return loss (Figure 4.4.4) of all the ports has degraded by 5-dB from the case without the connectors which can be explained by the impedance mismatches at the SMA launch pin-copper trace interface, but is still better than 15-dB across the frequency band in the simulations, and the isolation between the input ports and its adjacent ports (Figure 4.4.5) for all the ports is found to be better than 15 dB in simulations.
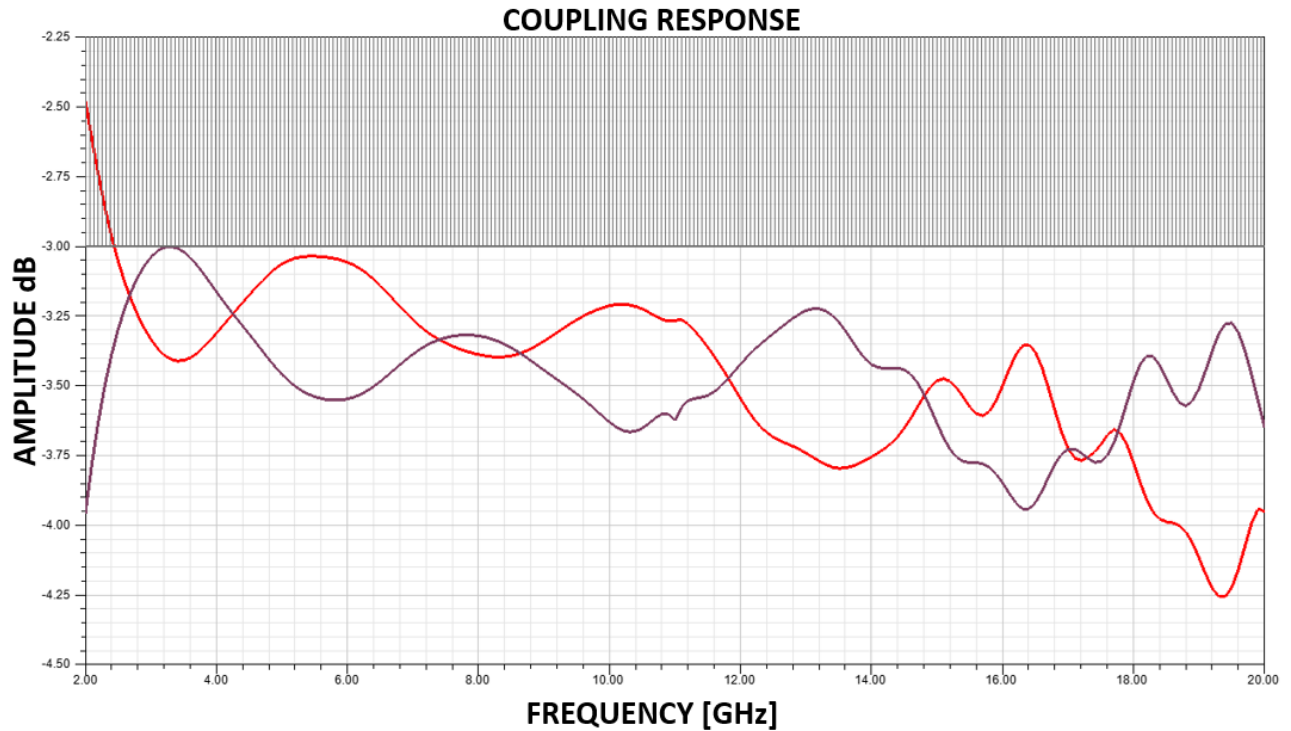
Figure 4.4.2: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with smooth copper assumptions
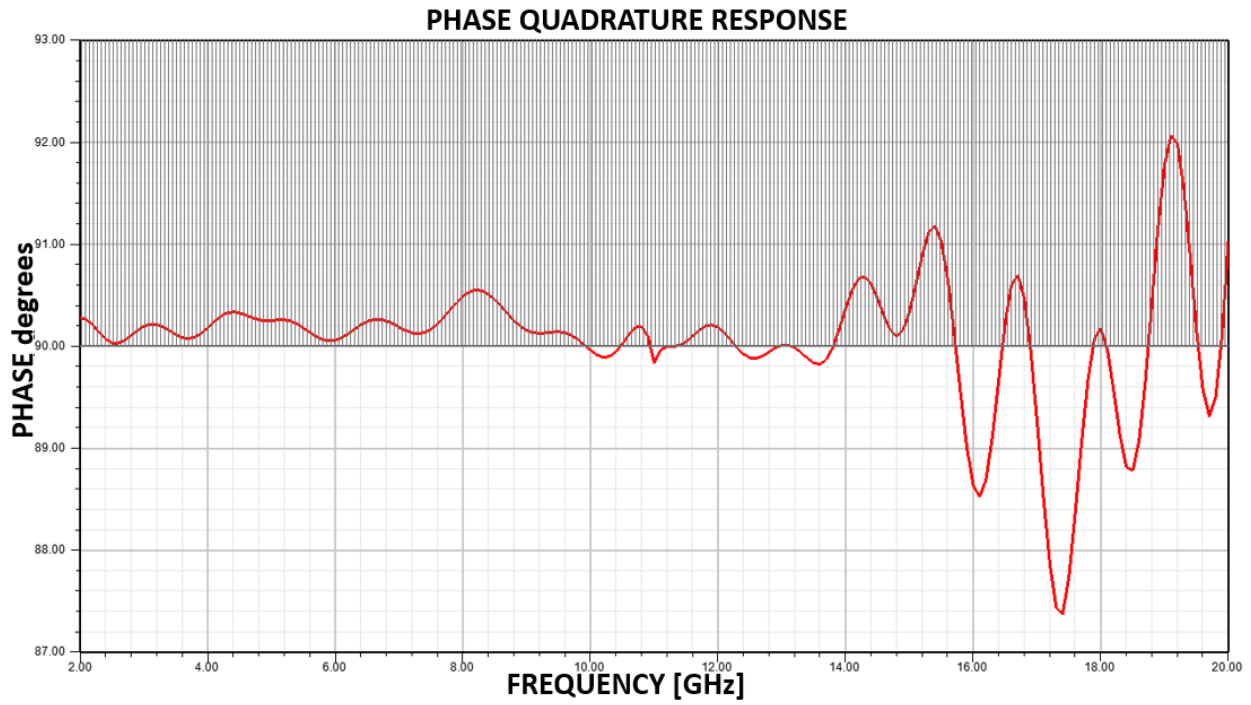
Figure 4.4.3: HFSS simulation of phase quadrature between the output arms of the assembled quadrature hybrid coupler with smooth copper assumptions
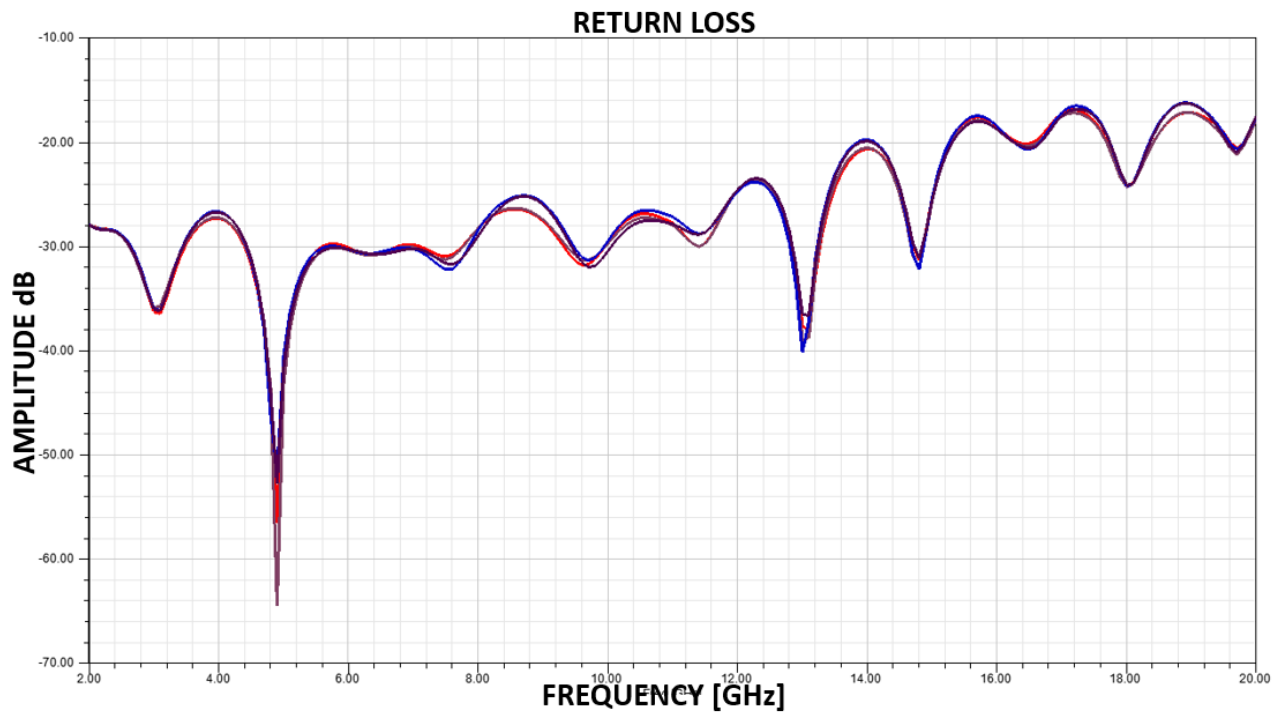
Figure 4.4.4: HFSS simulation of the return loss at all ports of the assembled quadrature hybrid with smooth copper assumptions
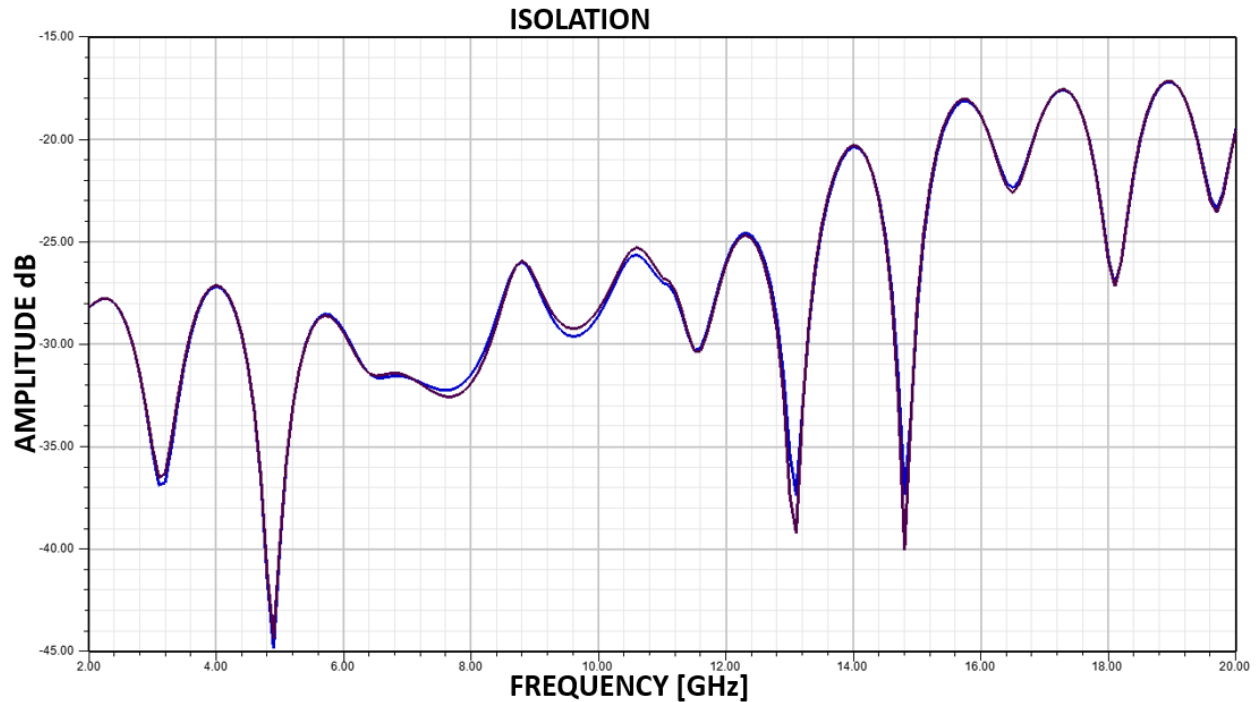
Figure 4.4.5: HFSS simulation of the isolation between at all ports and its adjacent ports of the assembled quadrature hybrid with smooth copper assumptions

The assembled quadrature hybrid coupler is measured using a 4-port vector network analyzer (VNA) from Keysight (PNA N5222A) calibrated with an electronic calibration module (Keysight N4433A) up to 20 GHz and with the measurement cables tied down such that the calibration is minimally biased on all the quadrature hybrids couplers measured. All further measurements of 10 MHz-20 GHz are calibrated using this setup (Figure 4.4.6) with the measured device SMA ports ensured to be on the same plane as the electronic calibration kit SMA ports. The cables used for the measurements are Micro-Coax utiflex UFA147A.

The measured coupling arm amplitude response (Figure 4.4.7) shows that both the output arms have significantly increased insertion losses due to the connector and the conductor losses and have losses approaching 2-dB from -3-dB at around 16 GHz which is unacceptable. The number of distinct passband ripples is seen to be about 5 with the 2 last ripples being smothered into one,

but the ripples after 12 GHz seems to have some distortions and non-smoothness which can be attributed to the SMA connectors, launch pin solder, conductor etching effects and other imperfections. The phase quadrature imbalance (Figure 4.4.8) between the coupled and through ports has degraded to $\pm\,3^{\rm o}$ from the desired $90^{\rm o}$. The return loss (Figure 4.4.9) of all the ports has degraded slightly from simulations, but is still better than 15 dB across the operational frequency band in the simulations, and the isolation between the input ports and its adjacent ports (Figure 4.4.9) for all the ports is found to be better than 15 dB in measurements.
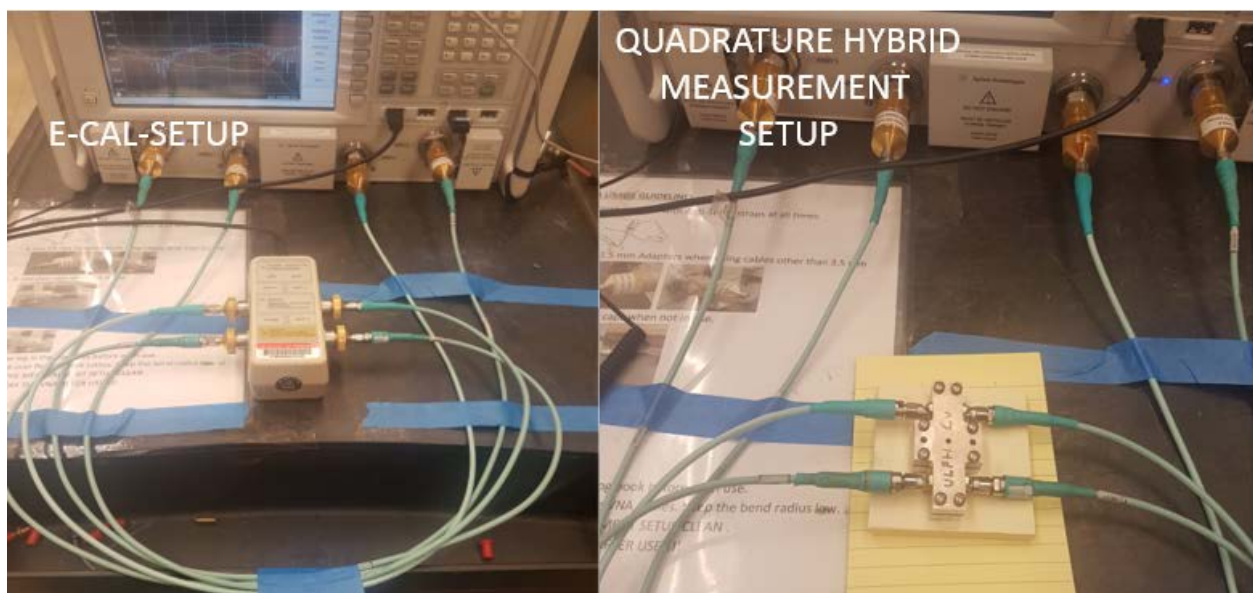
Figure 4.4.6: Photographs of the 4-port VNA calibration with the electronic calibration setup (left) and the quadrature hybrid coupler measurement setup (right)
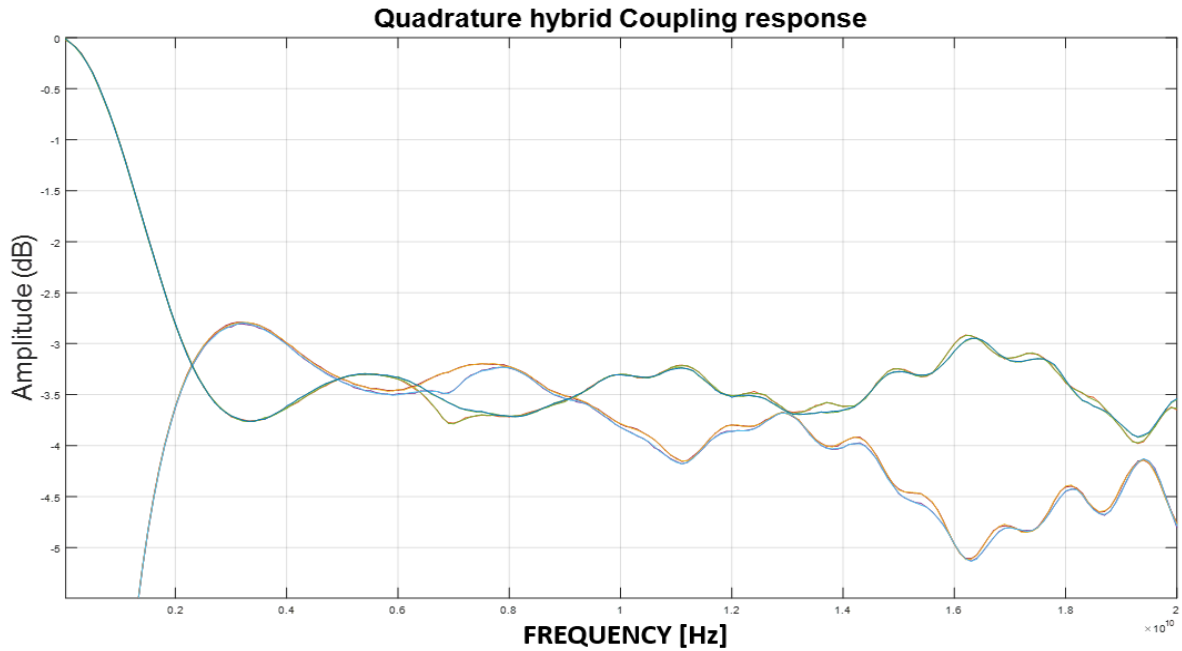
Figure 4.4.7: Measured scattering parameters of coupling response of all the ports of the assembled quadrature hybrid using reverse treated electro deposited 0.5 Oz copper foil
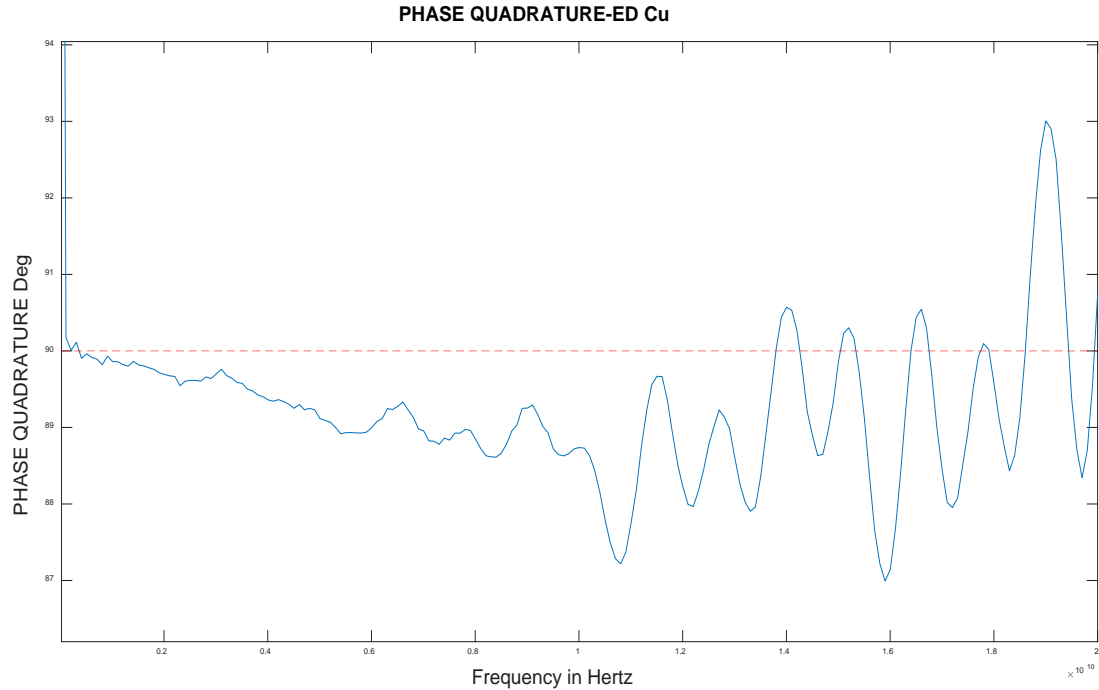
Figure 4.4.8: Measured phase imbalance in the phase quadrature between coupled and through ports of the assembled quadrature hybrid using reverse treated electro deposited 0.5 Oz copper foil
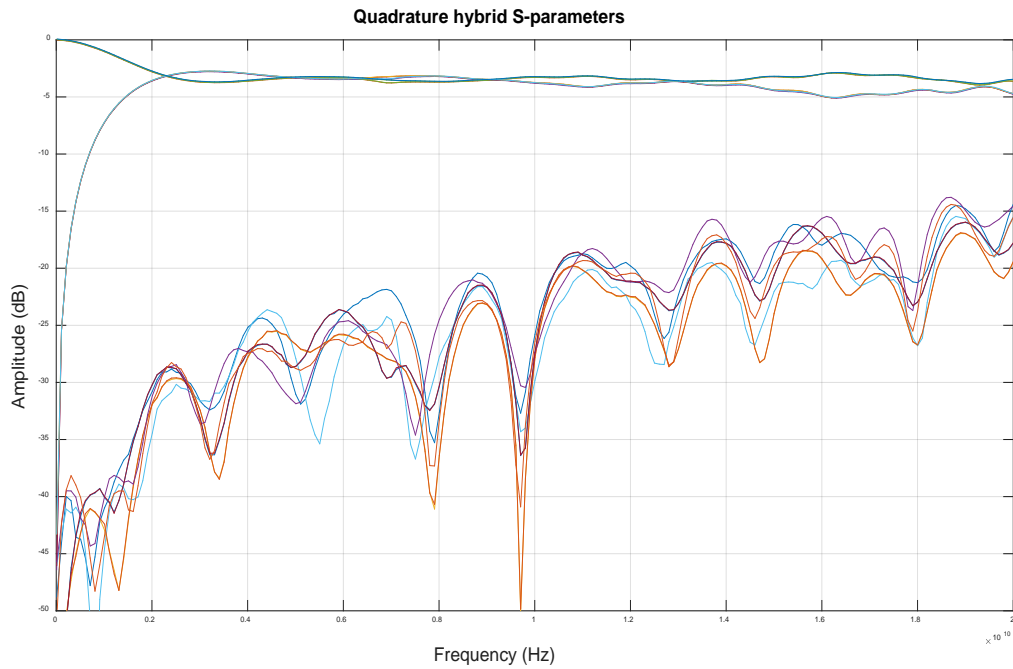
Figure 4.4.9: Measured scattering parameters of coupling response, return losses,isolation of all the ports of the assembled quadrature hybrid using reverse treated electro deposited 0.5 Oz copper foil

4.5   Effects of non-ideal copper foils and corrective solutions.

We now look into the primary reason for the much higher than expected measured insertion loss for the first quadrature hybrid version we assembled and ways to minimize the effects. We did use the standard issue 0.5 Oz copper foil with the TLY-5 laminate which was the reverse treated electrodeposited copper (CLH), which has a copper surface roughness of about 1.16 μm RMS on the treated/bonding side and 1.12 μm RMS on the untreated/masking side. This is due to dielectric laminate manufacturing processes and usage of adhesion promoters for the coper foil to adhere to the laminate[42-45]. This is a significant deviation from the perfectly smooth copper that we assumed in simulations and does contribute negatively at our frequency band of operations. An example of various copper surface roughness on 0.5 Oz Copper foil as seen under a scanning electron microscope is shown (Figure 4.5.1).
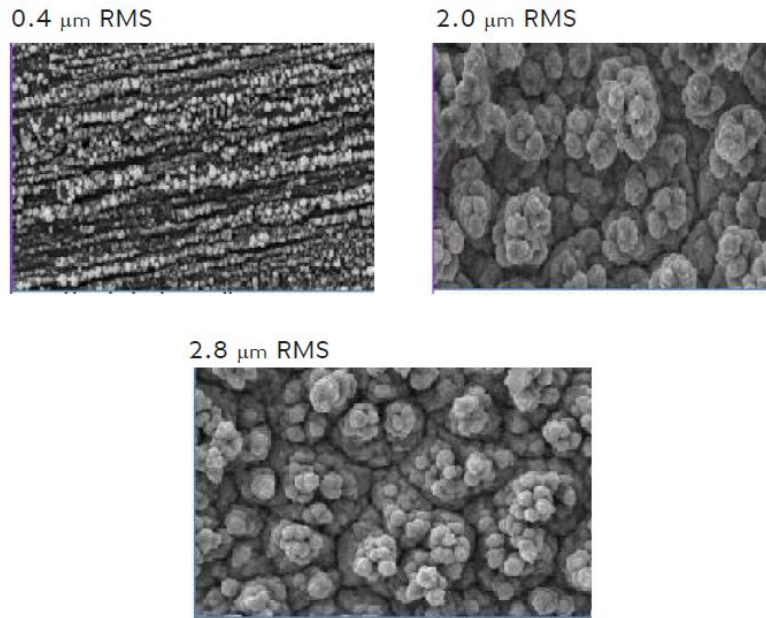
Figure 4.5.1: 0.5 Oz Copper foil surface roughness irregularities of 0.4 μm , 2.0 μm and 2.8 μm as seen under a scanning electron microscope [46]

The most detrimental effect due to surface roughness is the skin effect, where the currents in a low frequency wave propagation tends to flow more on inside the core of the conductor and a higher frequency wave current tends to flow more on the outskirts of the conductor (Figure 4.5.2) which reduces the effective cross sectional area of the conductor and hence an increase in resistive line losses. The skin depth is a good measure of the skin effect and is the distance inside the conductor where the amplitude of currents has reduced to $1/e$ of the maximum current from the outskirt of the conductor. For example, a 100 MHz low frequency wave has a skin depth of about 6.52 μm in copper which is acceptable while using the electrodeposited foil as most of the currents would be enclosed in the core of the conductor and would have similar path lengths through all the ports, whereas in our application of lower and upper frequencies of 2 GHz and 20 GHz the skin depth is about 1.461 μm and 0.461 μm respectively and causes the different signal frequencies to take different propagation paths and add to the path lengths as the propagation is not on a straight line

88

anymore. The added path length contribute to insertion loss as the copper length is effectively increased which also degrades the phase quadrature property as the paths length are not equal and symmetrical at all the coupler arms due to random manufacturing process variations. Copper surface roughness also induces localized surface eddy current's [47] which can cause heating and power dissipation and contributes to the overall insertion loss and also can be detrimental in peak power applications where excessive heating would cause misalignment in the coupler traces. An example of insertion loss for various surface roughness currents for 0.5 Oz copper is shown, and measured in differential microstrip lines on LCP material (Figure 4.5.3). These assumptions were simulated in HFSS using a Groisse surface roughness model of $1.16 \ \mu m$ RMS to show that indeed these effects were significant at our frequencies. These models are not perfectly modelled to estimate these effects as equal surface roughness randomness assumptions are applied to all the surface which is truly not the case, nevertheless it shows some of the effects.
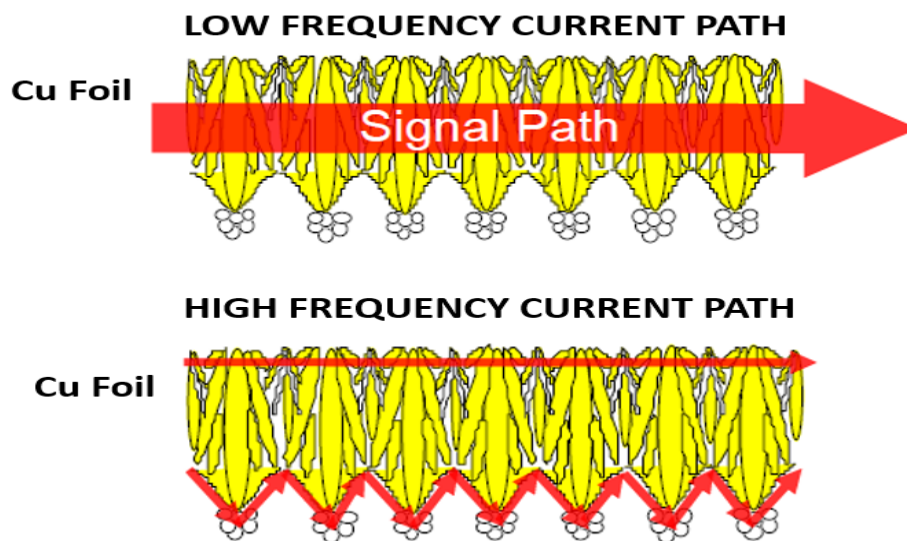
Figure 4.5.2: Signal propagation paths of low and high frequency currents in laminate copper foil [46]

89
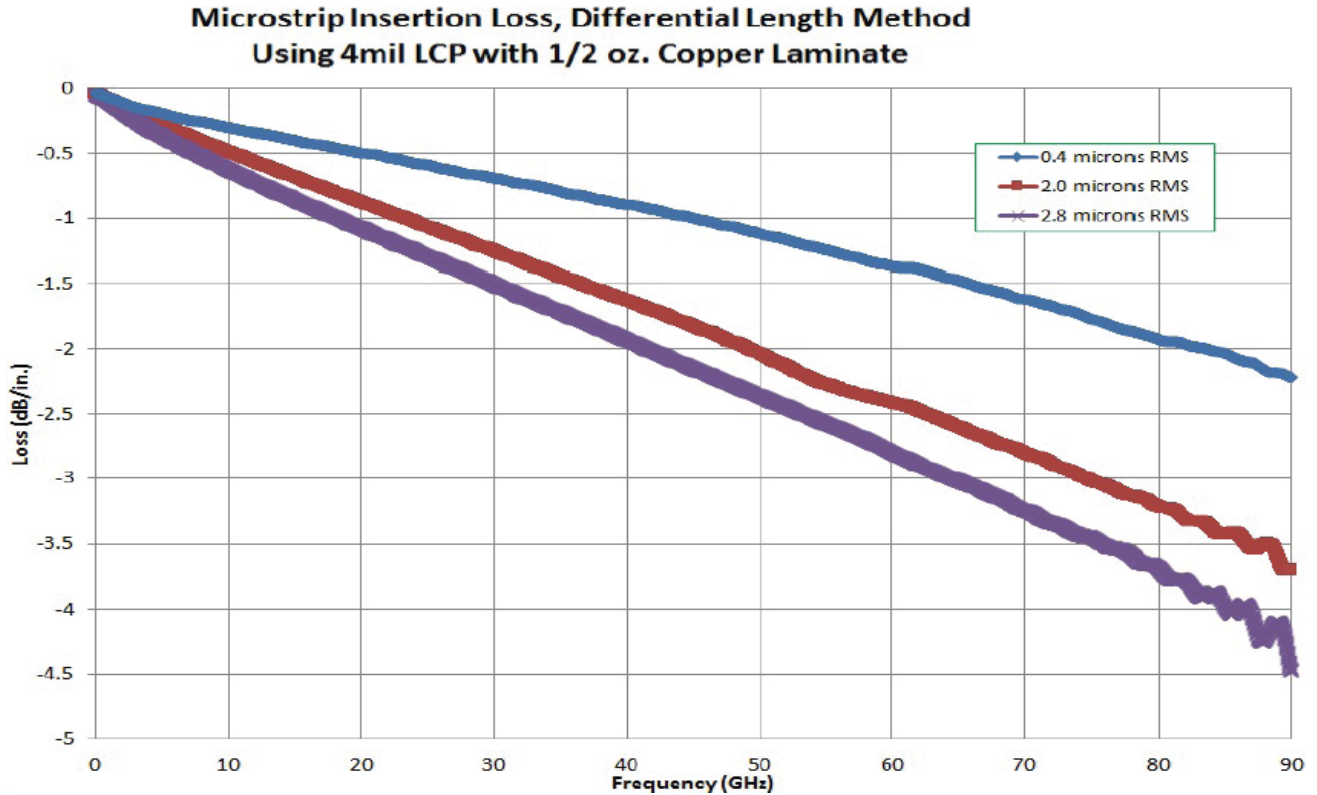
Figure 4.5.3: Insertion loss in differential microstip lines of 0.4 μm , 2.0 μm and 2.8 μm copper foil surface roughness for half ounce thickness [46].

The HFSS simulated performance for reverse treated electrodeposited half ounce foil which was used in our first build shows that the coupling arm response (Figure 4.5.4) has an additional dB of loss as seen from our measured results. Even though the phase quadrature (Figure 4.5.5) shows only a slight degradation from the smooth copper assumptions, it is expected to degrade much more as the surface roughness assumptions are assuming equal randomness at all the coupler arms and still maintains symmetry. The return loss (Figure 4.5.6) at the ports shows degradation but still is better than 15-dB across the operating band. Isolation at the ports (Figure 4.5.7) is also better than 15 dB at the ports, even here the assumptions of copper surface roughness are not perfect.

90
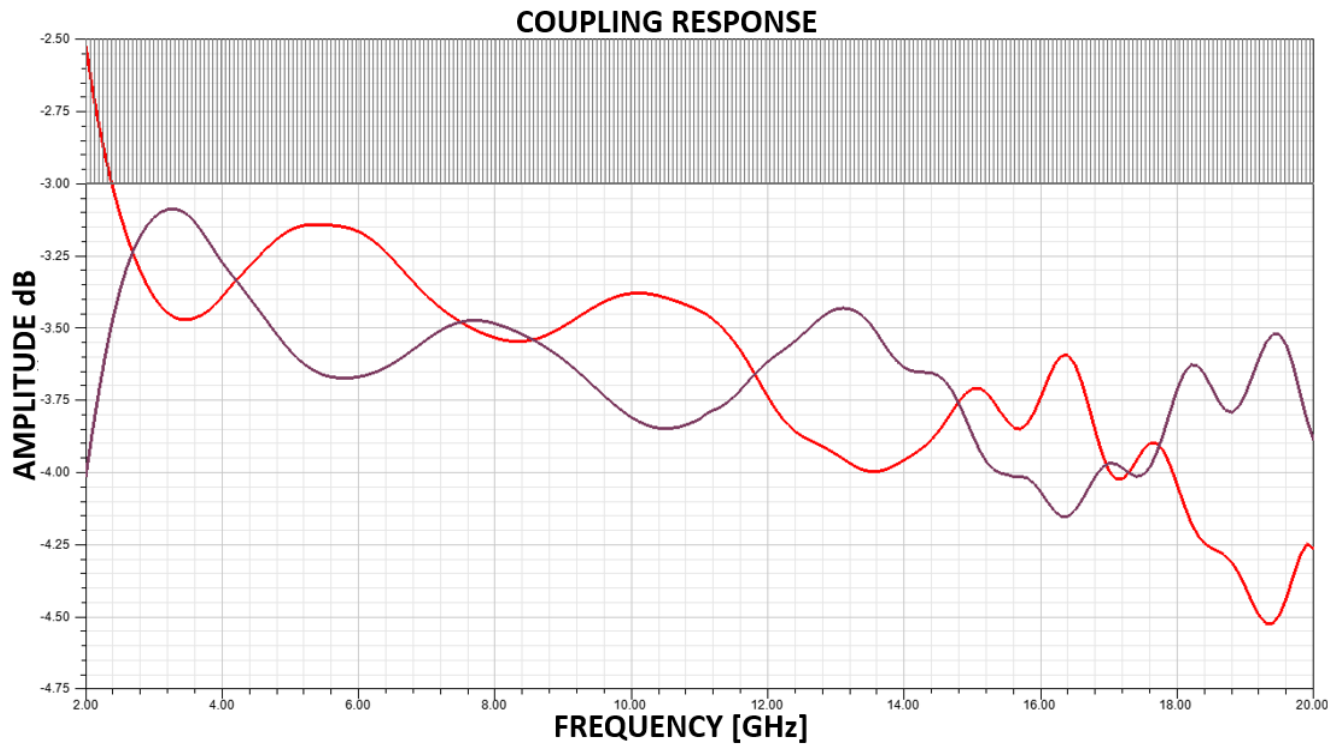
Figure 4.5.4: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 1.16 μm rough copper surface assumptions
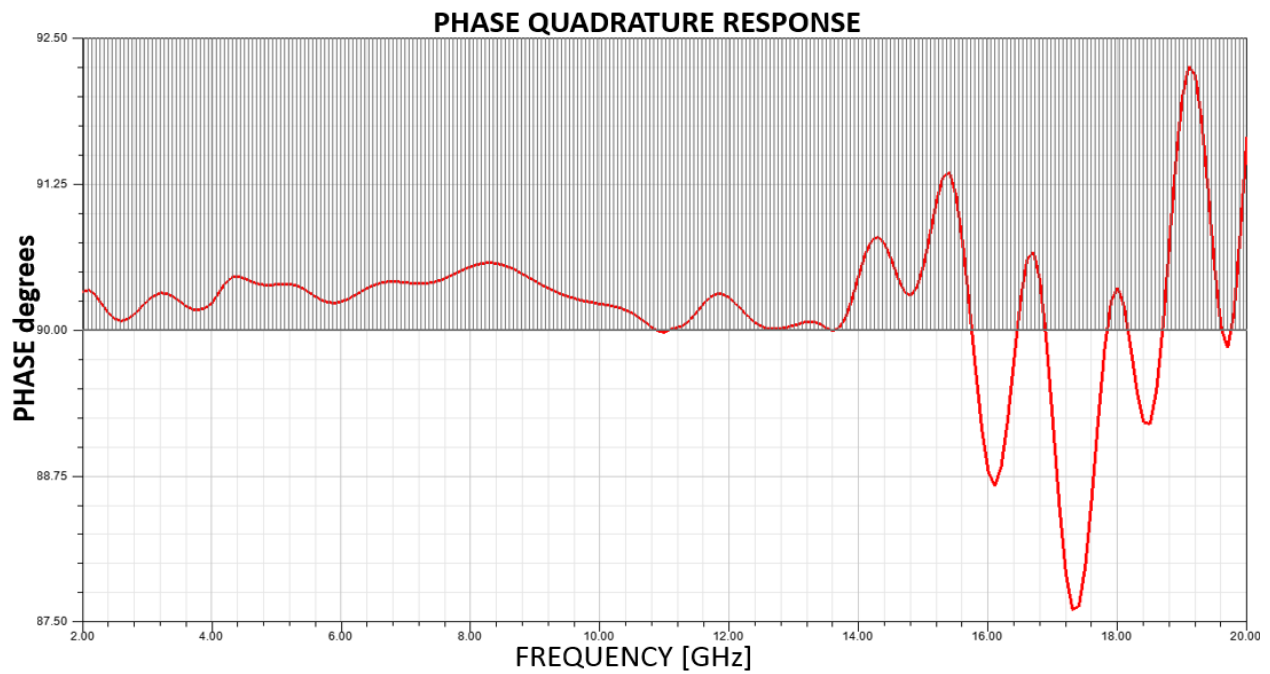
Figure 4.5.5: HFSS simulation of phase quadrature between the output arms of the assembled quadrature hybrid with 1.16 μm rough copper surface assumptions
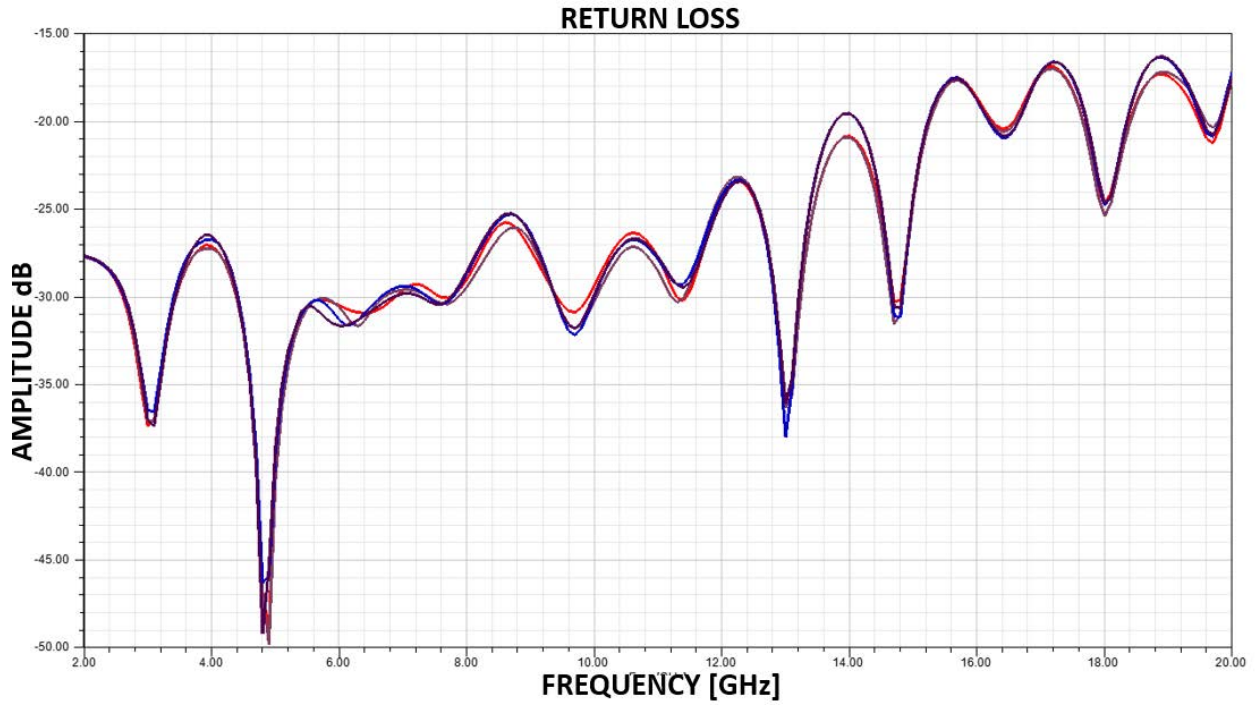
Figure 4.5.6: HFSS simulation of the return loss at all ports of the assembled quadrature hybrid with 1.16 μm rough copper surface assumptions
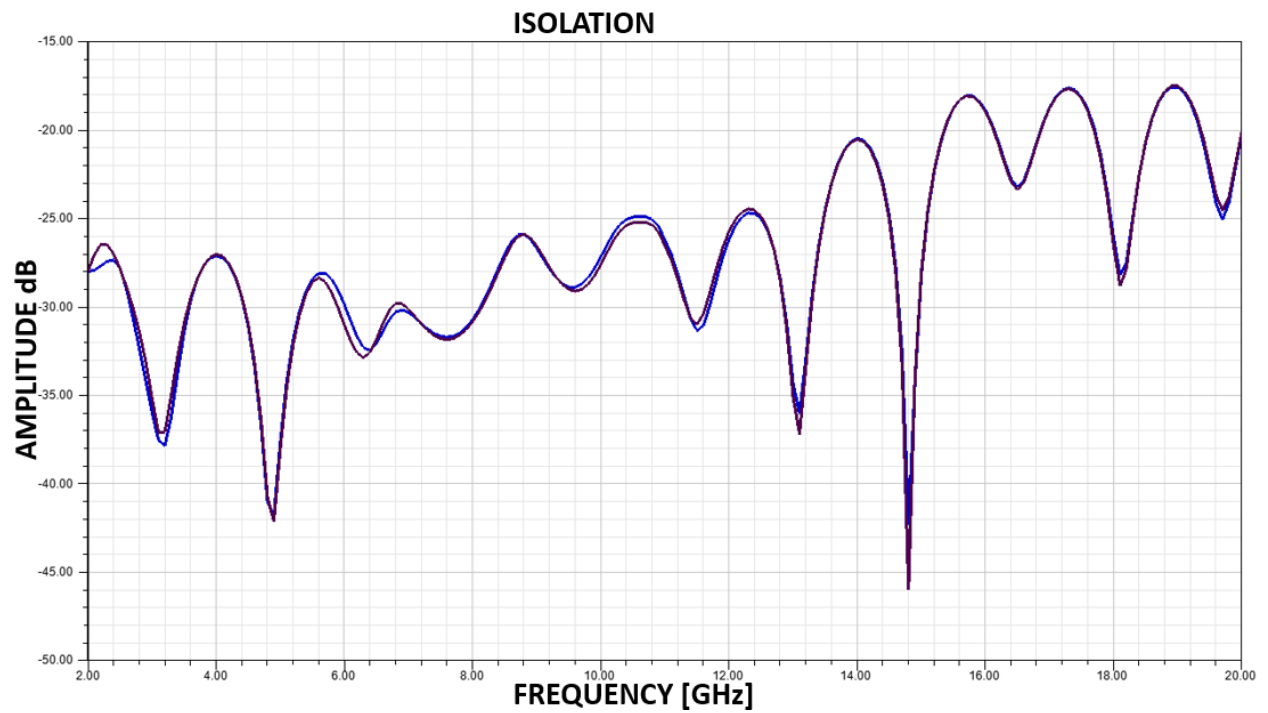
Figure 4.5.7: HFSS simulation of the isolation between at all ports and its adjacent ports of the assembled quadrature hybrid with 1.16 μm rough copper surface assumptions

As we have seen the effects of surface roughness in copper we now require a way to mitigate these effects. Fortunately for us there exists specialty copper foil materials designed for low loss/high frequency microwave applications. Taconic TLY-5 offers materials with designations as shown (Figure 4.4.1). We chose the ULPH (Ultra low profile half ounce) copper foil material which was among the foils from Taconic with the lowest copper surface roughness for our second build of the quadrature hybrid coupler.

Copper Cladding (Bruker Optical Measurement)					
Copper Designation	Surface Roughness R_{MS} Treated side		Surface Roughness R_{MS} Untreated side		Description
	Microinches	Microns	Microinches	Microns	
ULPH	*Comparable to rolled copper				Ultra Low Profile ½ oz
ULP1					Ultra Low Profile 1 oz
RH	15	0.37	12	0.31	Rolled-Annealed ½ oz
R1	11	0.28	9	0.24	Rolled-Annealed 1 oz
CFH	29	0.74	13	0.34	HVLP Electrodeposited ½ oz
CF1	27	0.69	16	0.40	HVLP Electrodeposited 1 oz
CLH	46	1.16	44	1.12	Reverse Treated Electrodeposited ½ oz
CL1	39	1.00	52	1.31	Reverse Treated Electrodeposited 1 oz
CH (CVH)	66	1.68	18	0.46	High Ductility Very Low Profile Electrodeposited ½ oz
C1 (CV1)	60	1.54	19	0.49	High Ductility Very Low Profile Electrodeposited 1 oz

Table 4.5.1: Table showing copper foil materials available from Taconic and the corresponding copper surface roughness on the treated and untreated sides[48].

Before our final build we did an HFSS simulation run with 0.37 μm copper surface roughness to simulate ULPH copper foil material to see if the performance improvements were noticeable and in agreement with our reduction in insertion loss assumptions. As expected and seen in the coupling arm response (Figure 4.5.8) the insertion loss has improved and even the phase response (Figure 4.5.9) is better compared to the electro-deposited copper case. The return loss (Figure 4.5.10) and isolation (Figure 4.5.11) improves as a consequence of a better phase quadrature

property. The actual improvements are expected to be much more significant as the modelled surface roughness do not capture the real surface roughness.
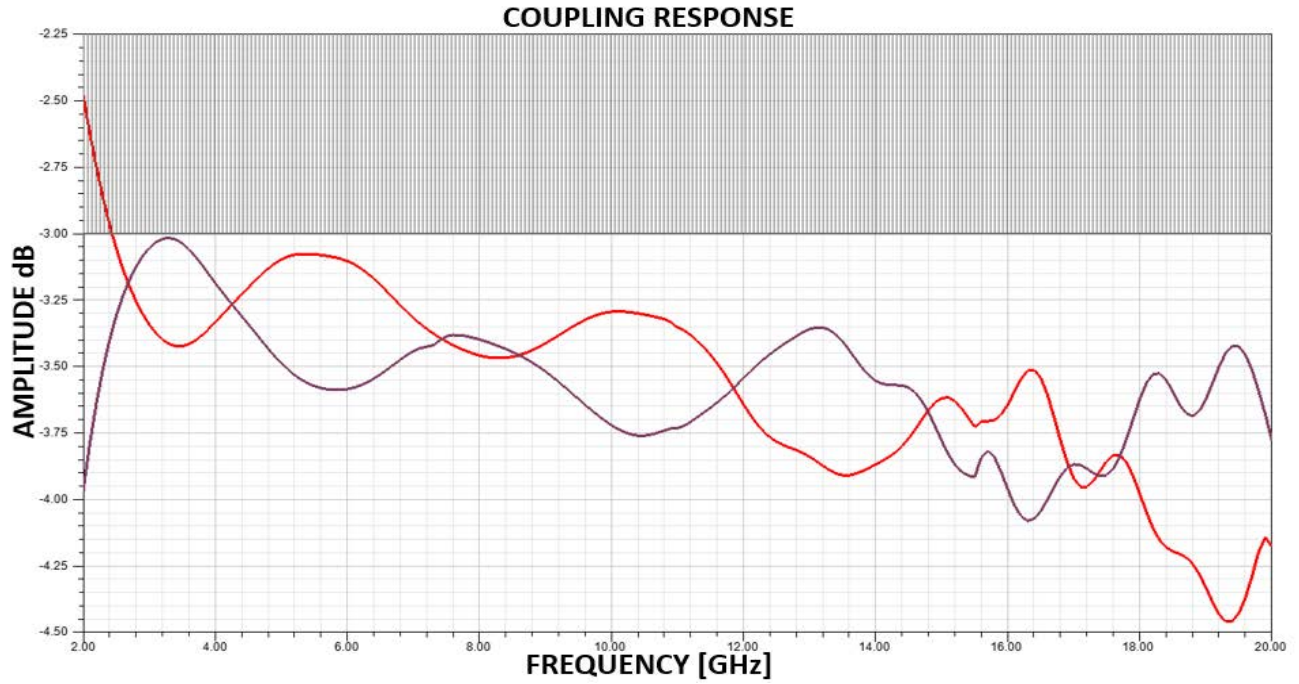
Figure 4.5.8: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions
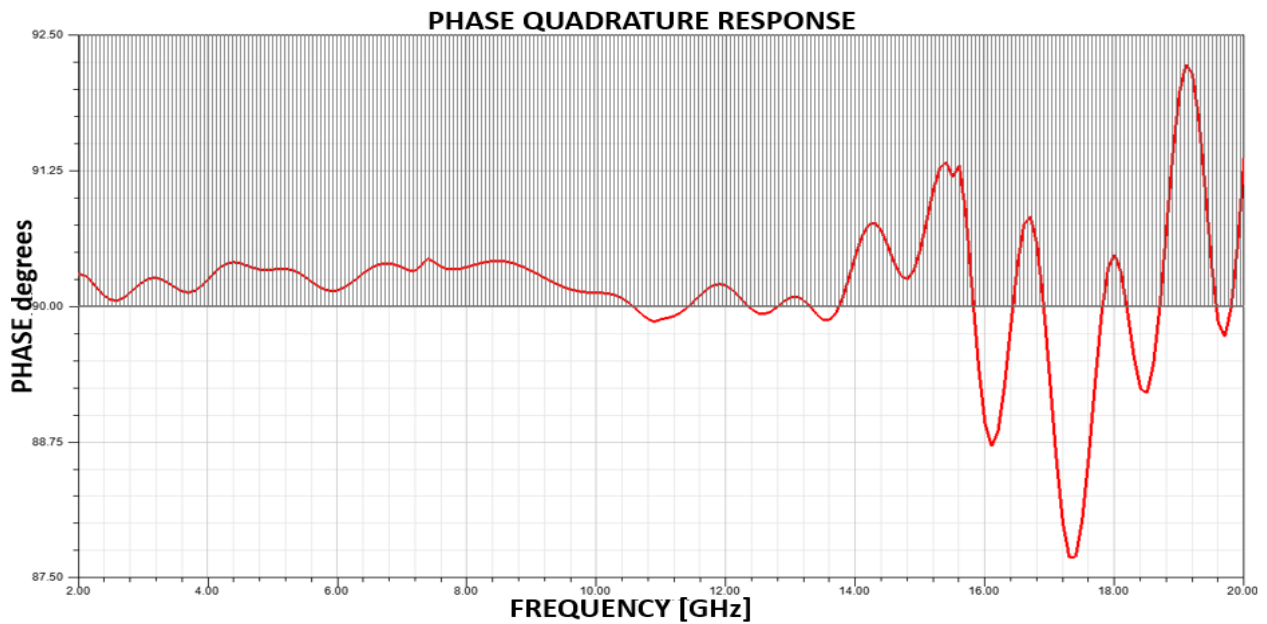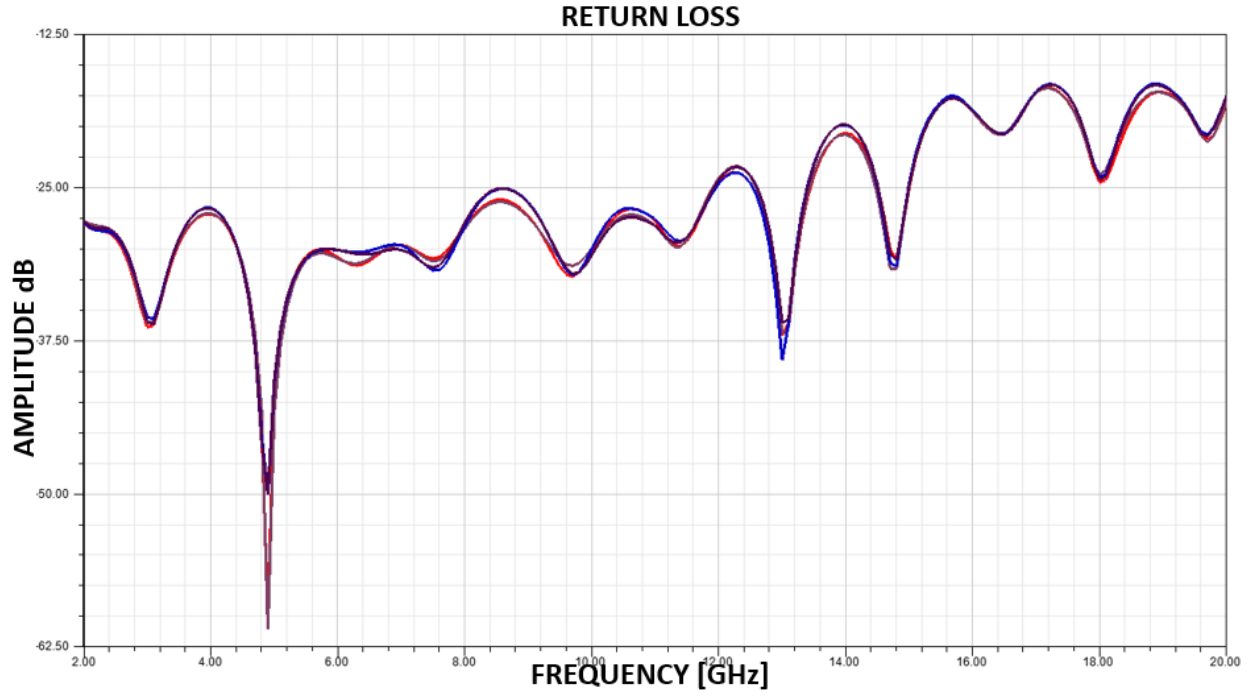
Figure 4.5.10: HFSS simulation of the return loss at all ports of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions

95
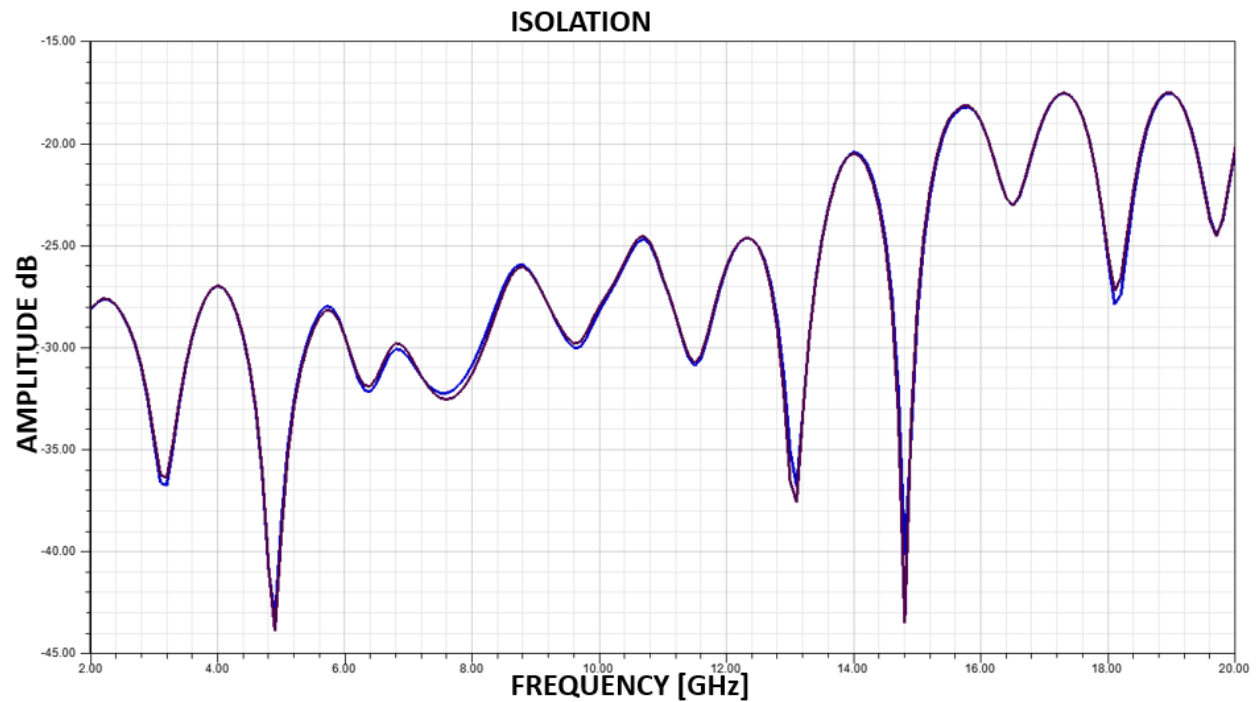
The final quadrature hybrid coupler was assembled and measured and the results are concluded here. The coupling response (Figure 4.5.12) has significantly improved by at least over a dB across the operating bandwidth, although the dip at 16.2 GHz and slight non-conformity with the simulations after about 15 GHz which is also seen in the electrodeposited copper case is of a concern and the next section would deal with the probable causes of these effects. The phase quadrature plot (Figure 4.5.13) shows a maximum phase deviation of +6 degrees until 20 GHz, this however is probably contributed significantly by the misalignment in launch pin arrangement and imperfect mechanical etch at the trace edge of the coupler as the coupling, return loss and isolation plot shows that not all port responses are aligned more or less identically as seen in the electro-deposited copper case and so neglecting the SMA flange connector launch imperfections the phase quadrature should be better balanced than the electro deposited hybrid case. The return loss (Figure 4.5.13) at all the ports and isolation between the ports and its adjacent port is all better than 15dB across the operational band. This final build quadrature hybrid coupler performance is verified to be acceptable in our design requirements, and we decided to use the ULPH copper for the snow duplexer build.
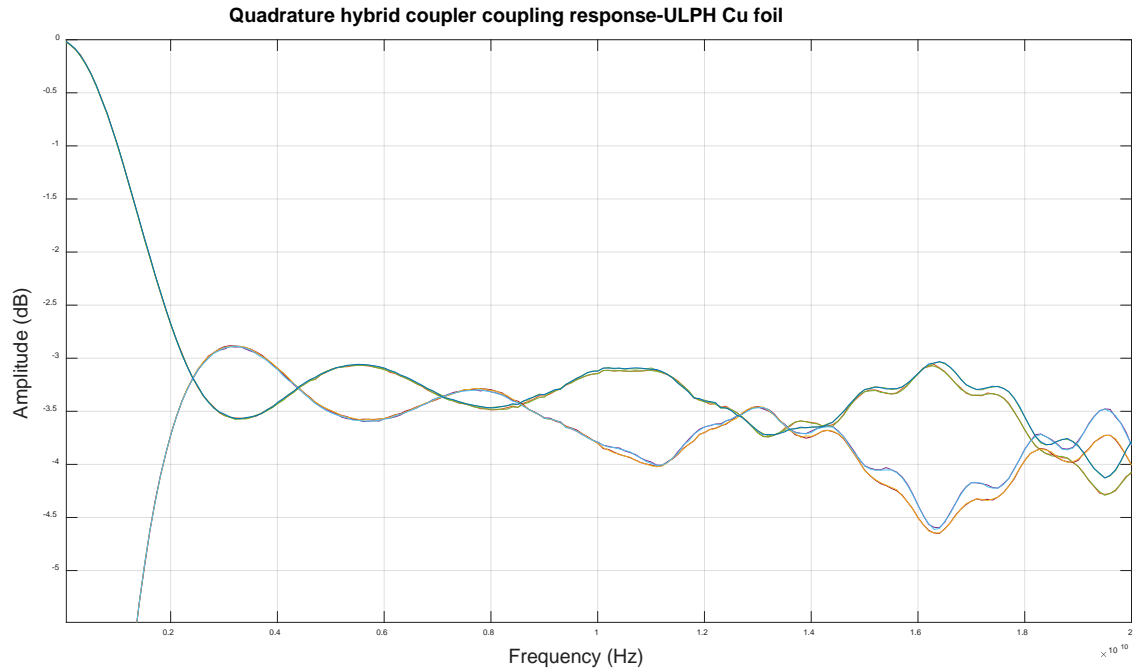
Figure 4.5.12: Measured scattering parameters of coupling response of all the ports of the assembled quadrature hybrid using ULPH copper foil
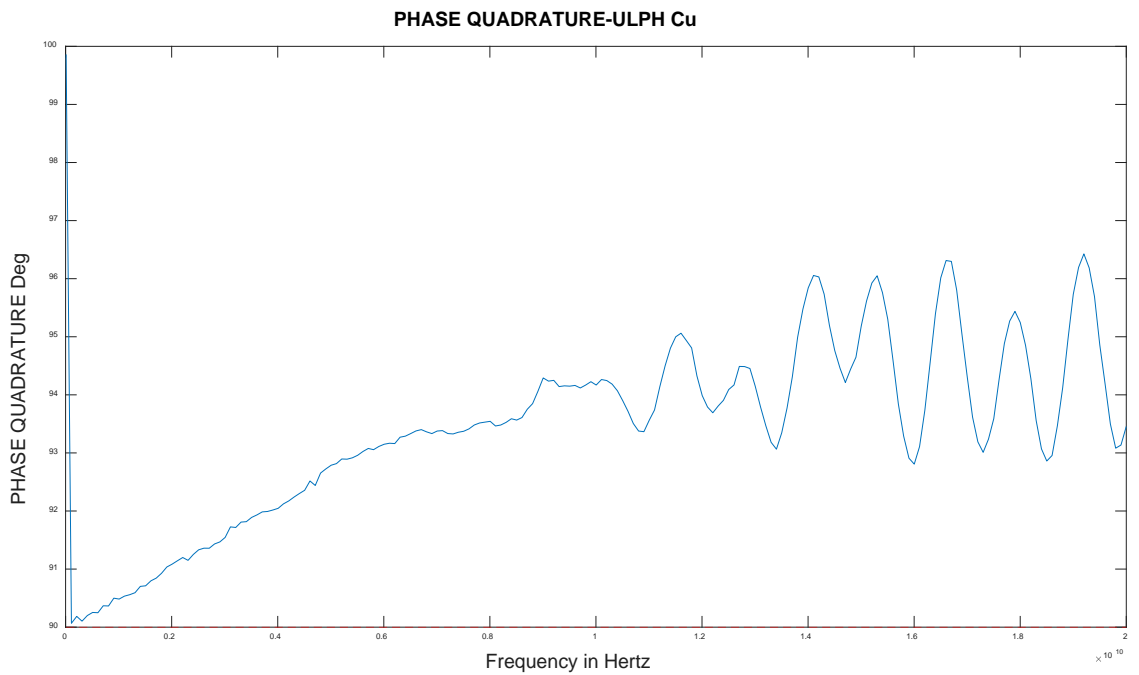
Figure 4.5.13: Measured phase imbalance in the phase quadrature between coupled and through ports of the assembled quadrature hybrid using ULPH copper foil
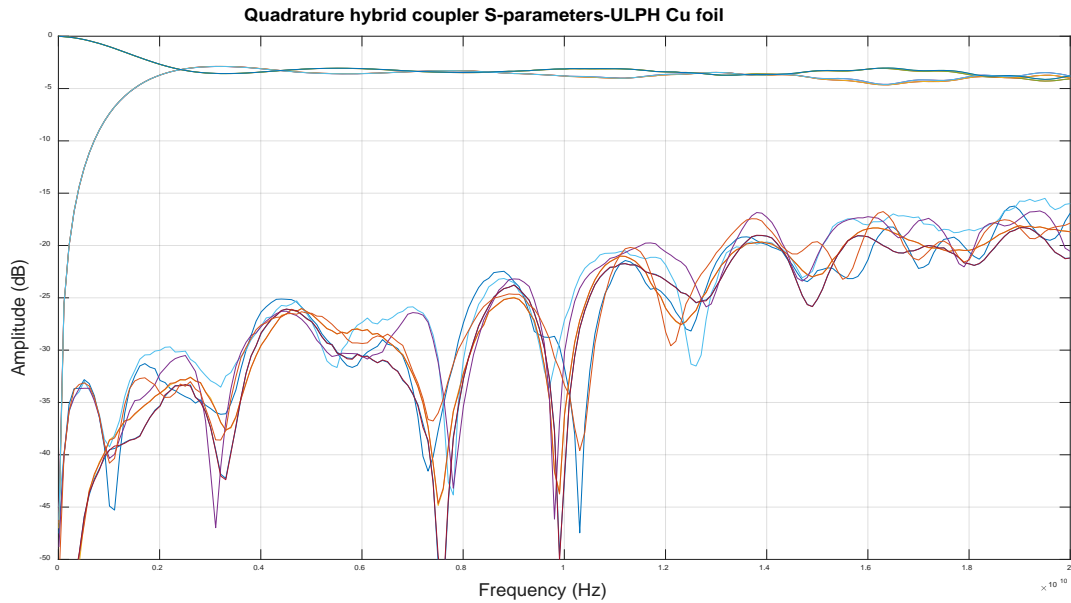
Figure 4.5.14: Measured scattering parameters of coupling response, return losses, isolation of all the ports of the assembled quadrature hybrid using ULPH copper foil

4.6   Fiber glass weave, non-homogenous, anisotropic and other non-ideal effects

This section would briefly go through the effects of other non-idealities resulting from the dielectric material chosen, fabrication and mechanical integration which causes the measured performance to deviate from the HFSS simulated performance of the quadrature hybrid coupler.

4.6.1 Fiberglass weave effects of the dielectric laminate material

The Taconic TLY-5 Teflon dielectric laminates are fiberglass weave reinforced (Figure 4.6.1.1) for mechanical rigidity, while this is a good thing for ease of manufacturing and PCB board strength it causes non-idealities due to the very nature of the weaves. Two main weave effects are the location of the copper trace in alignment to the conductor (Figure 4.6.1.2.a) and the angle of the copper trace with the fiberglass weave (Figure 4.6.1.2.b). The location of the copper trace edge if located on the weave has a higher dielectric constant Dk than the surrounding gaps and causes a lower characteristic impedance and hence non-uniform perturbations in the field lines just based

98

on the location which cannot be properly controlled in the board fabrication process. The angle of the copper trace with the fiber weaves causes periodic loading resonances with the strength and frequency of resonance depending on the angle.
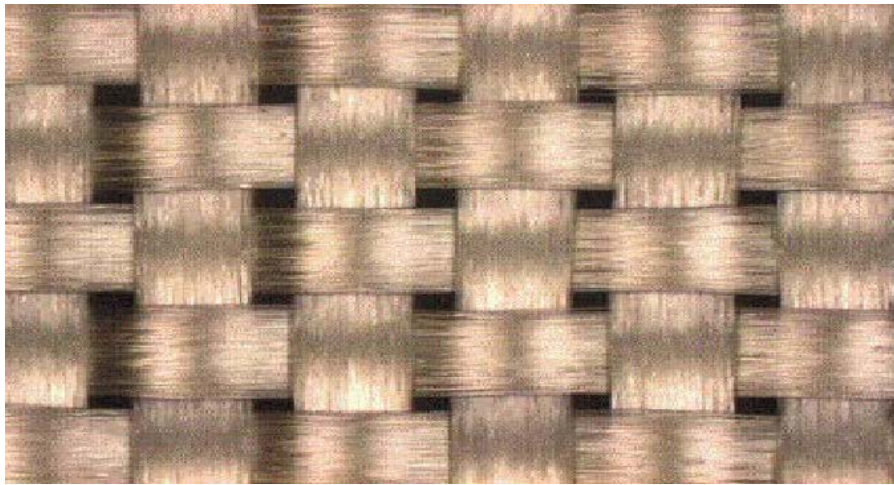
Figure 4.6.1.1: Fiberglass weaves used for reinforcing the PTFE low Dk dielectric material laminate
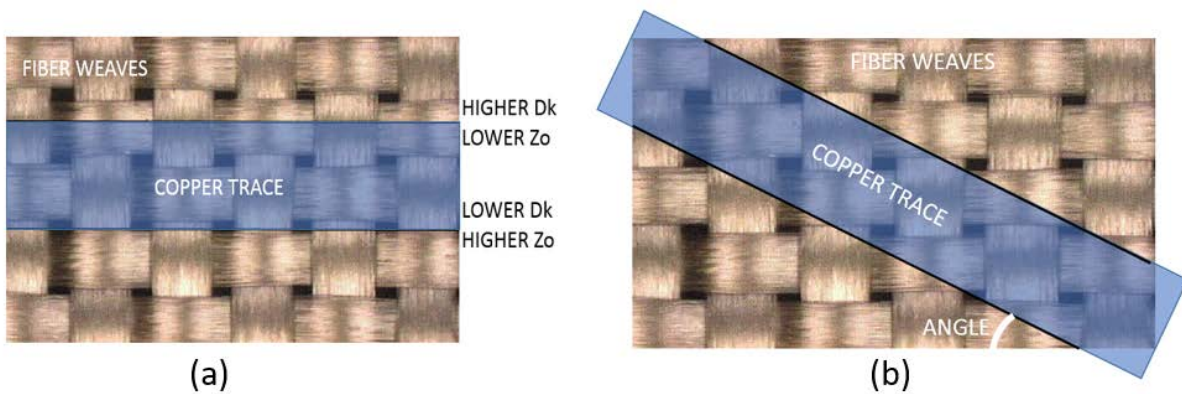
Figure 4.6.1.2: (a): alignment of copper trace with the fiber weaves ; (b): angle of copper trace with the fiber weaves

Isola corp [46] did an analysis of the angle resonance effects for a 4-inch trace and sample measurements of insertion loss (Figure 4.6.1.3) and return loss (Figure 4.6.1.4). There looks to be resonance in band at 16 -20 GHz for an angle of 7 degrees. This could explain the dips at 16 GHz

99

and slight non-conformity with simulations from 15-20 GHz as the designed quadrature hybrid as an inherent property has to be offset and angled to achieve the desired coupling profile. The alternative dielectric material to use is the Rogers RT/Duroid 5880 which do not have the fiber weave enforcements hence reducing the resonance effects but unfortunately is not available for a center dielectric thickness of 3.5 mil and has lesser mechanical rigidity, CTE and strength which means that we have to stick to this material option.
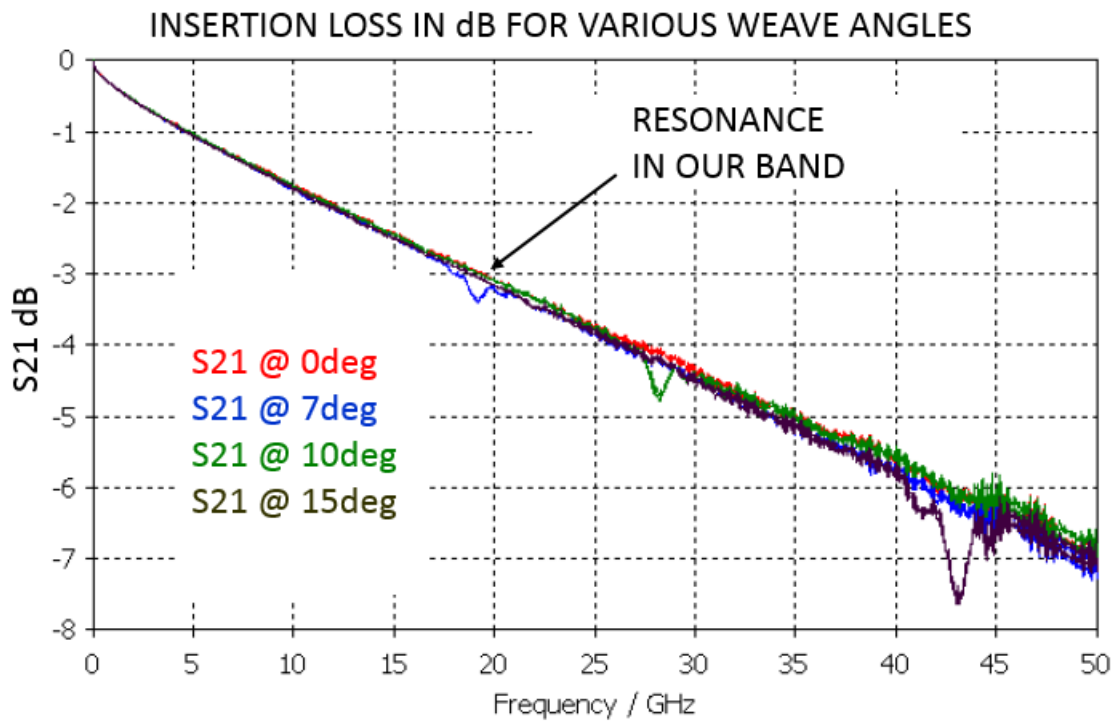
Figure 4.6.1.3: Insertion loss (dB) for a 4 inch sample trace to demonstrate the resonance effects due to the fiber weave angle [46] .
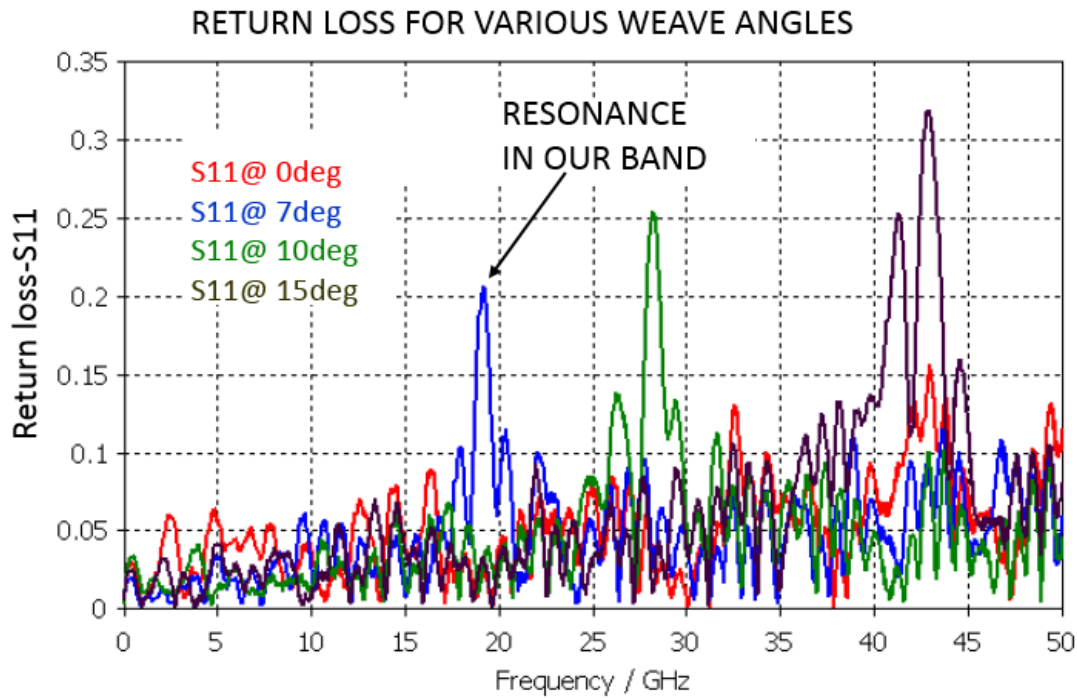
Figure 4.6.1.4: Return loss for a 4 inch sample trace to demonstrate the resonance effects due to the fiber weave angle [46].

4.6.2 Anisotropic and non-homogenous dielectric laminate effects

The other significantly contributing non-idealities in the dielectric laminate is the anisotropic and non-homogeneous effects which mainly arises from the manufacturing process. Anisotropicity in the material arises primarily from the stacking of dielectric sheets with resin/glue to form laminates of desired thickness (Figure 4.6.2.1) and hence creates a slightly different effective dielectric constant property along the height, width and length axis of the laminate. This causes the fields in the various directions to travel with different velocities which results in phase non-linearity and dispersion effects. Anisotropic effects were simulated with the quadrature hybrid coupler model without the flange connectors and terminated on 50 Ohm wave ports. Anisotropic dielectric with a 10% dielectric constant variation along the height axis which can demonstrate the effects of this non-ideal property were used. As can be seen this induced anisotropicity degrades the coupling

response (Figure 4.6.2.2) and the passband ripples is seen to be severely smeared and distorted compared to the isotropic case. Phase quadrature (Figure 4.6.2.3) has degraded due to unequal mode velocities. Isolation (Figure 4.6.2.5) and return loss (Figure 4.6.2.4) has degraded by about 5dB as compared to the smooth copper with isotropic dielectric case. Usually the effective dielectric along the width and length axis of the laminate is similar due to the uniform weaving in the dielectric sheets and is assumed same in the simulations. Non–homogeneous property arises from the issue of the Teflon fiberglass weaving and the applied resin and hence causing slightly different dielectric constants and hence distorting the propagating electric fields ever so slightly but definitely. Non-homogeneity can be neglected when the wave-length in the dielectric material is much larger than the fiber weave structures.
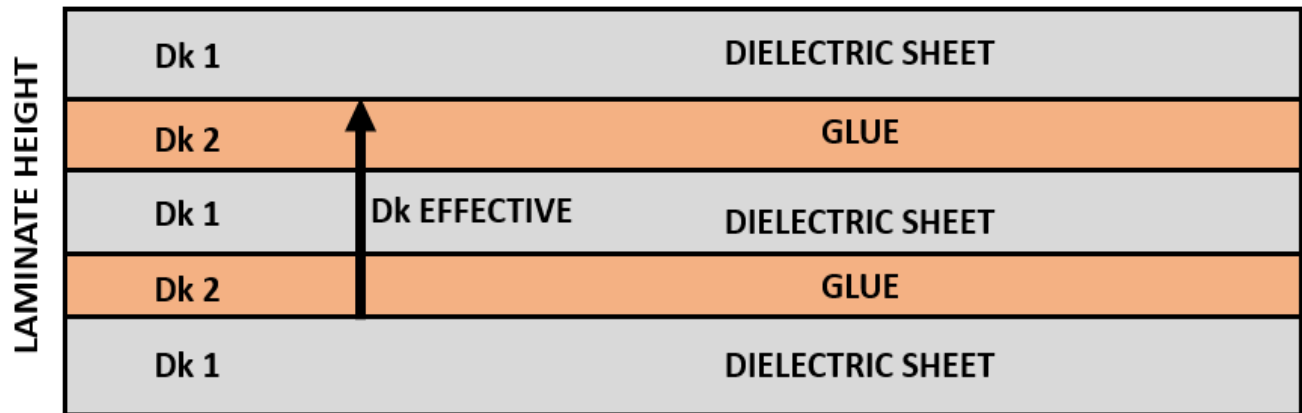
Figure 4.6.2.1: Dielectric sheet stacking with resin/glue in the laminate for achieving desired board thickness.
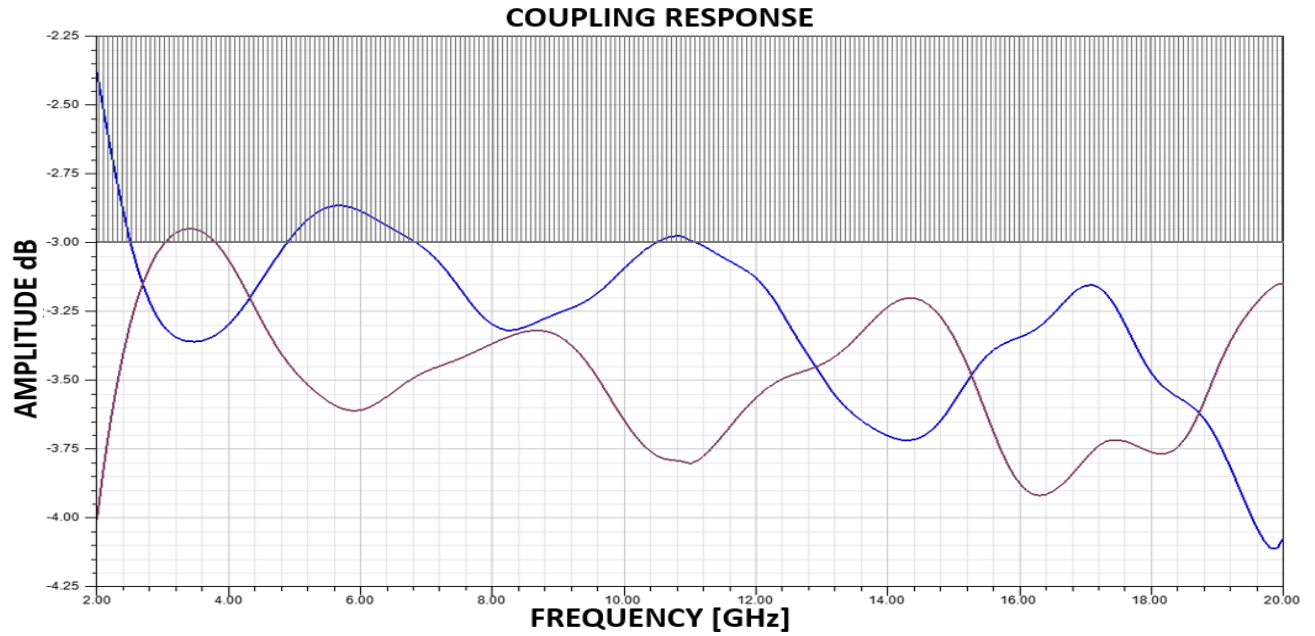
Figure 4.6.2.2: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions
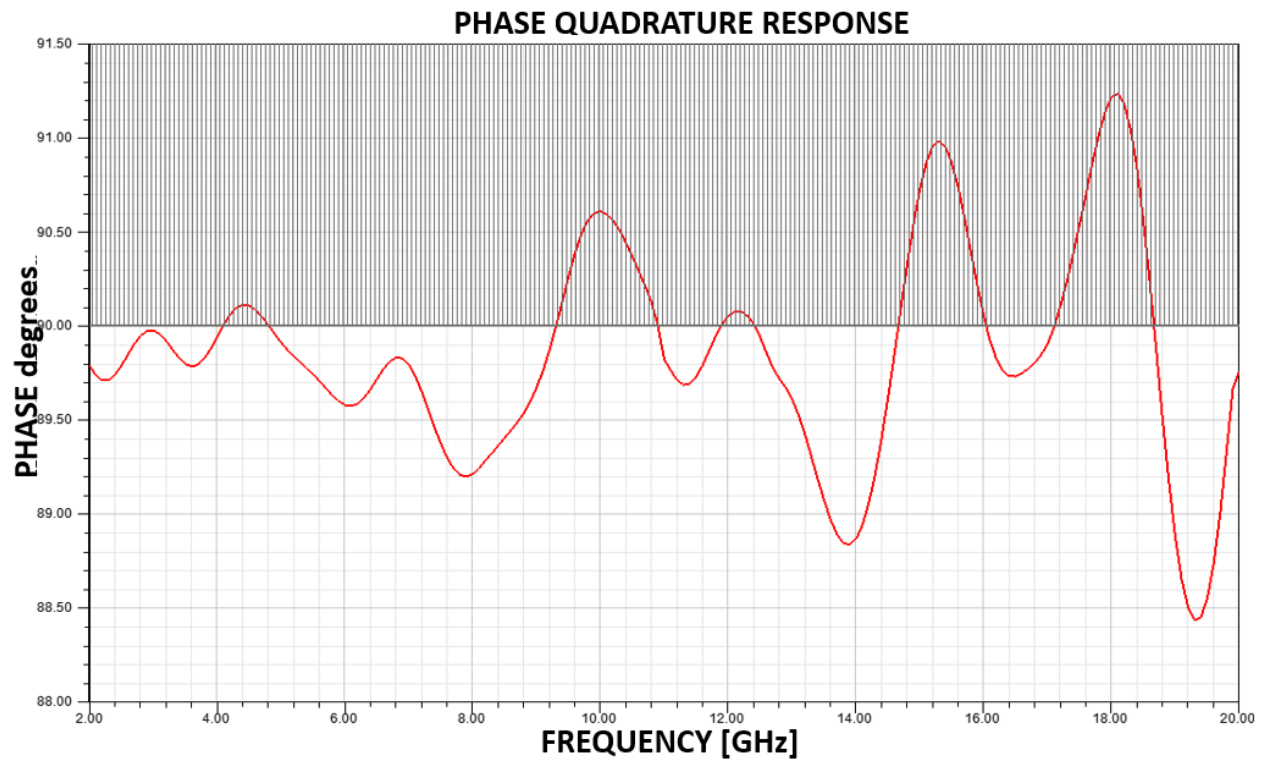
Figure 4.6.2.3: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions

Figure 4.6.2.4: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions
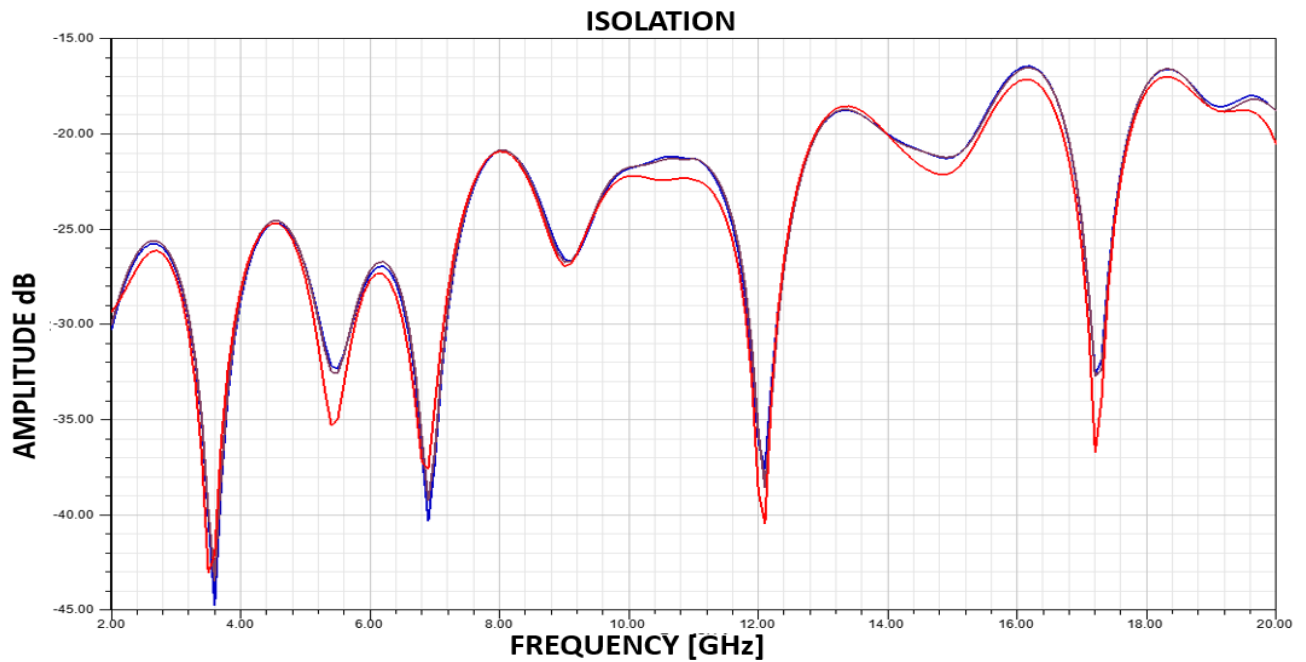
Figure 4.6.2.5: HFSS simulation of coupling response of the output arms of the assembled quadrature hybrid with 0.37 μm rough copper surface assumptions

4.6.3 Copper trace shape distortion non-idealities

Besides the dielectric laminate non-idealities, the copper foil shape distortions caused by the laminate manufacturing and PCB fabrication processes also contribute to the non-ideal behaviors leading to differences in simulated and measured results. The main contributor here is the ridging in the copper foil caused by the fiber weaves imprinting its shape on the foil. These ridges deviate the copper trace from the perfectly planar assumption to a pseudo planar one where the electric fields perturb and cause unequal velocity of the even and odd modes and also deviation from the designed coupling profile. Adding to this is the shape distortion is the coupler arm crossover points at the center of the symmetric couplers where when the dielectric layers are pressed down to assemble the coupler, the traces press on each other and distort the desired location and designed center coupling strength value. The pressing of the dielectric layers down also creates air pockets as shown which causes slight dielectric mismatch and field perturbations. Last but not the least is the deviation from the designed coupler shape caused by etching and registration tolerances which causes deviation from the desired coupling profile, and can be improved by fabricating the PCB board at a high etching accuracy PCB fabrication house, which would be a thing to do in the future to enhance performance.

RIDGES IN THE SIGNAL TRACE DUE TO FIBER WEAVING

Figure 4.6.3.1: Ridges on the copper trace caused due to reinforcement fiber weaves in the dielectric laminate material
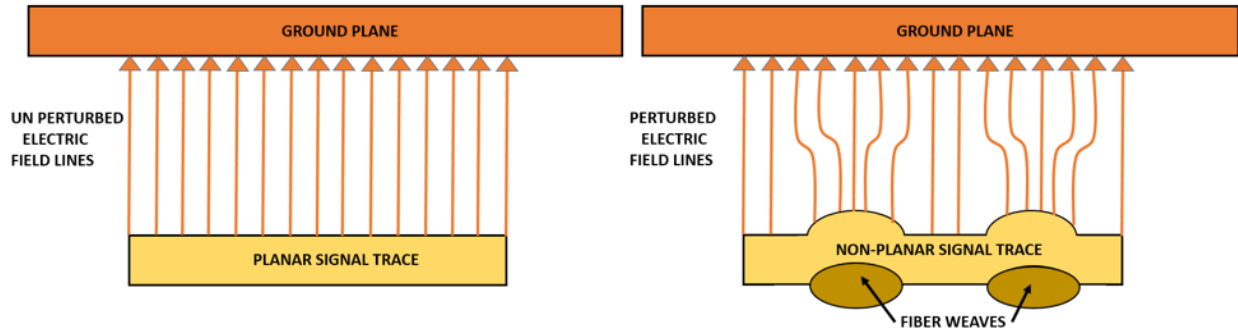
Figure 4.6.3.2: Fiberglass reinforcement weaves causing ridges on the copper trace and inducing electric field distortions
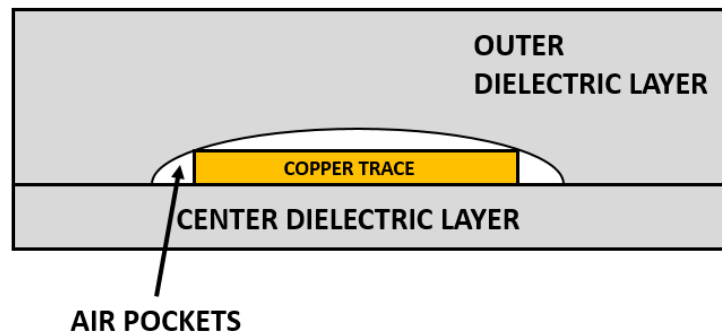
Figure 4.6.3.3: Air pockets around copper signal trace caused by the pressing of dielectric layers during assembly.

4.7 Size, cost consideration and power handling of the final standalone quadrature hybrid coupler with a high performance COTS quadrature hybrid

The final version of the low loss, low form factor planar quadrature hybrid coupler is assembled and measured and has a size of 2.6 inches in length and 1.3 inches in width which includes the SMA flange connector protrusions and a height of about 0.57 inch. The weight of the device is about 2.08 ounce. This prototype quadrature hybrid coupler was found to be much smaller and lighter than commercial standalone coaxial connected quadrature hybrid couplers with similar bandwidth. The price of getting this coupler fabricated and assembled (*ignoring the engineering and design cost*) is about $150 apiece including the dielectric laminate layers, SMA launches, launch pins and the aluminum casing. This pricing can be further reduced if the device is mass

produced as the materials can be procured for much cheaper rates. A similar device by Marki microwave (QH-0226) which is heavier and larger using the same SMA connectors is retailing for $995 apiece.

 The power handling of this device in pulsed mode, which would be our mode of application is limited by the smallest dielectric spacing between the signal conductors which is the center section having 3.5 mil spacing where the dielectric strength is 9425 V-peak and can safely handle at least half a Kilowatt (conservatively) in short duty cycles. In continuous wave (CW) applications the power handling is limited by the heating in the narrowest conductor and the thermal expansion effect which can dislodge the signal trace location. This material has a good CTE along all the axis due to the fiberglass reinforcement and the ULPH coper foil dissipates much lesser heat than regular copper foil due to lower AC resistance and hence it is estimated to safely handle at least 50 Watts of CW(conservatively) power without performance degradation. The comparison of the designed vs COTS quadrature hybrid coupler is shown in table (Figure 4.7.5). Also, shown is the couplers side by side with measured weights (Figure 4.7.1)   and the s-parameters of the COTS quadrature hybrid which was measured using the same calibrated setup used for the designed couplers (Figure 4.7.2). Phase quadrature (Figure 4.7.3) of the COTS coupler is shown which seems poorer than our case. Coupling response of the COTS coupler (Figure 4.7.4)   shows higher insertion loss than the designed coupler.
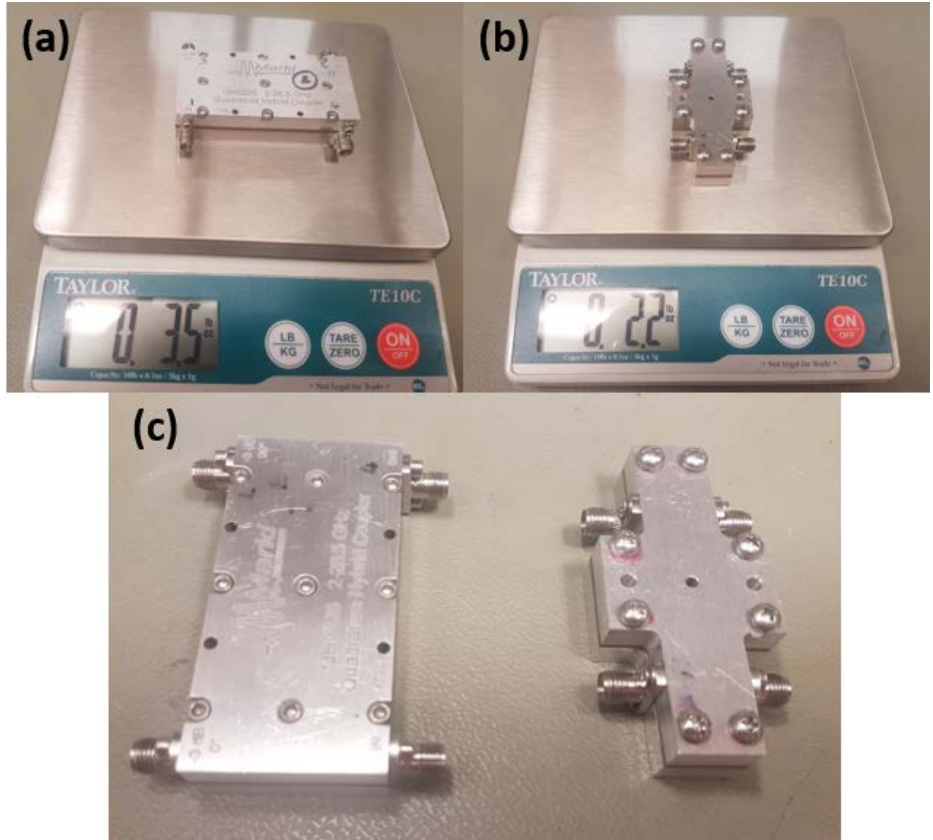
Figure 4.7.1: (a): Marki QH-0226 quadrature hybrid coupler being weighed. (b): Designed ULPH quadrature hybrid coupler being weighed. (a): picture of COTS and designed coupler side by side .
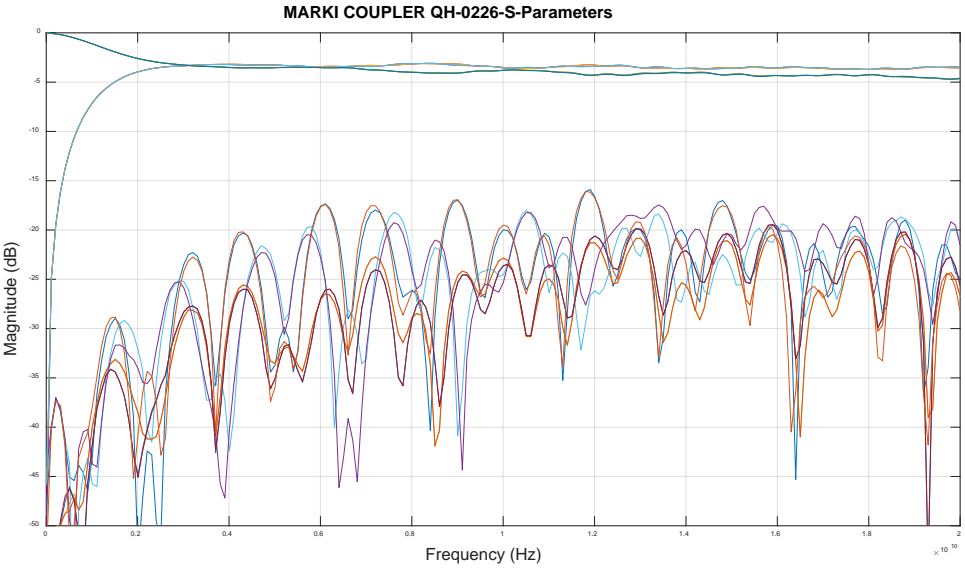
Figure 4.7.2: Measured scattering parameters of coupling response of all the ports of Marki QH-0226
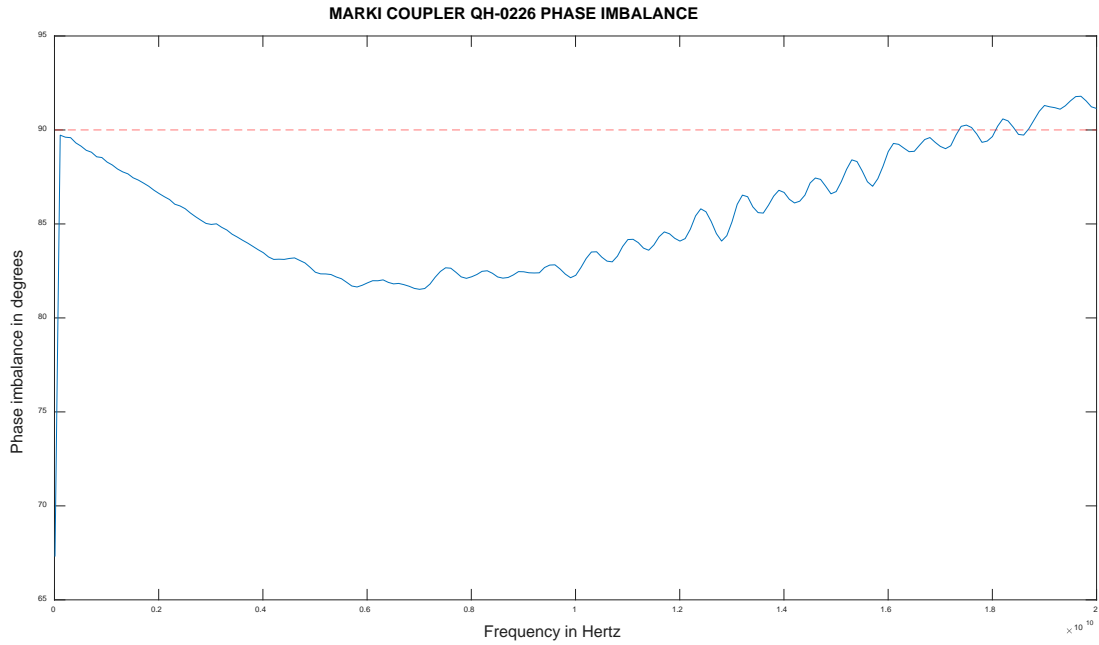
Figure 4.7.3: Measured phase imbalance in the phase quadrature between coupled and through ports of Marki QH-0226
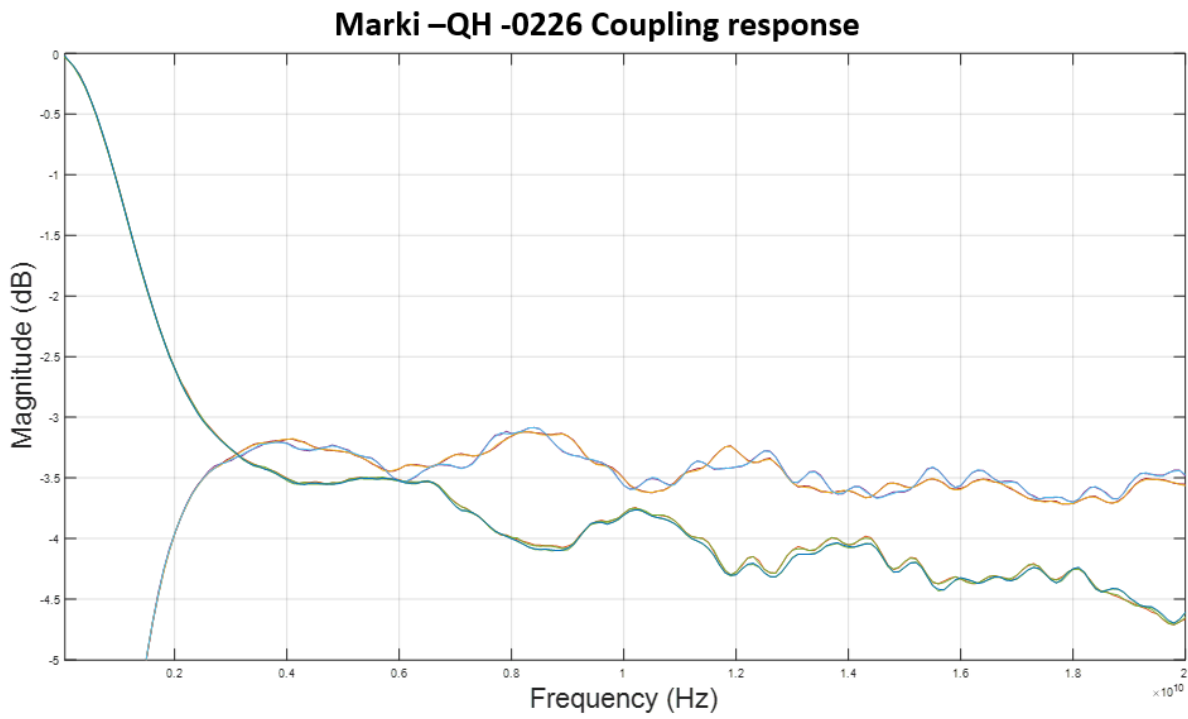
Figure 4.7.4: Measured scattering parameters of coupling response of all the ports of the assembled Marki QH-0226

Property name	ULPH Quadrature hybrid	Marki QH (02-26)
Weight (Oz)	2.2	3.5
Size (inches)	2.6 x 1.33	2.7 x 2.0575
Cost	150 $ (Material cost)	995 $ (Retail price)
Frequency of operation measured (GHz)	2-20 GHz designed usable to 25 GHz (Measured)	2-26.5 GHz (manufacturer's specification)
Material	Taconic TLY-5	Rogers-5880
Typical excess insertion loss (Measured)	0.7 dB	1.3 dB
Maximum phase quadrature deviation (degrees) 2-20 GHz (Measured)	+6	-8.5 to +1.7
Power handling (CW)	50 Watts (estimated)	20 Watts (Manufacturer specified)
Isolation (dB) minimum (measured)	16.39	15.89
Return loss (dB) minimum(measured)	16.39	15.89

Table 4.7.5: Table showing the properties of the designed vs COTS quadrature hybrid coupler

Chapter 5: T/R Switch Development

This chapter presents the design of the T/R switch module using the balanced configuration discussed in previous chapters, which incorporates the custom quadrature hybrid couplers presented in Chapter 4. The design considerations for a single PCB solution are discussed along with the justification for component selection in the various sections of the circuit. Moreover, power handling considerations are discussed in detail.

5.1 PIN diode selection, power handling and biasing requirements

One of the critical components in the T/R module after the quadrature hybrid couplers are the switching elements. Careful selection and integration is needed to achieve the best performance in terms of switching speeds, insertion loss, isolation, required biasing voltages and power handling capability. Shunt PIN diodes were the preferred choice as they offer superior isolation compared to series PIN diode, which behave as lossy capacitors at high frequencies when reverse-biased. Shunt diodes also offer greater average power handling capacity. This is because one end of the diode can be directly attached to a heat-sink. The PIN diode selected was the MMPN080045, a silicon shunt diode from Macom. This diode is a distributed PIN switch element in a 50 Ohm matched microstrip format, which is capable of operating from 2-20 GHz. This device has a gold ribbon bondable surface to make jumper connections to other components. High-performance diodes operating at these frequencies are difficult to package in a surface mount device (SMD) format. We had to opt for the wire/ ribbon jumper design to maintain good electrical performance, although this increased the complexity of the board. The power handling of these diodes in continuous-wave (CW) mode is 40 Watts. The power handling of the device when operated in pulsed mode with a pulse duration of 2 µs (short pulsed mode) is limited by the

breakdown voltage of the diode (-200 Volts)[1] and is estimated to be about 400 Watts (56 dBm). This is considering a 50 Ohm system assuming a perfect impedance match and proper heat sink. The switching speeds for these diodes is about 450 ns, which is 1/10 of the carrier lifetime (τ) of 4500 ns provided by the manufacturer.

One of the main issues which prevented us from using a shunt diode in the transmit path in lower-frequency implementations of the balanced T/R switch design was the requirement for a large reverse voltage to prevent the diode going into conduction in the presence of very high RF power[49]. In this application, the device selected stands at an advantage of being used at a much higher frequency range and having a comparatively thick intrinsic region (80 μm). Therefore the input RF signal cannot make the minority charge carriers flood fast enough into the intrinsic region to drive the device into conduction [49]. We calculated the power handling for the diodes considering the lowest frequency of operation (2 GHz), assuming a biasing voltage of -20 V and typical properties of silicon [49]. Since the T/R switch sweeps from 2 GHz to 18 GHz during the pulse duration the actual required negative bias will be lower because the power handling capability for a given negative bias voltage improves as the operating frequency increases. Figure 5.1.1 shows the minimum power handling calculated for -20 Volt reverse bias in CW mode as 25 Watts and for a 2 μs pulsed duration with a 200 μs pulse period it is about 276 Watts, which is well above our design requirements. Figure 5.1.2 shows the variation of maximum power handling for a -20 Volt reverse bias with pulse durations of 2,5,10 μs for a 200 μs pulse period. From these plots it can be seen that as the duty cycle lowers, the diode requires much less reverse bias to remain in the off state.

---

[1] The expected pulse duration is much smaller than the thermal time constant of 15 μs for the selected PIN diode.

Also plotted in figure 5.1.3 is the typical power handling, calculated for a -20 Volt bias limit for a high altitude operational mode in airborne platforms such as the global hawk where duty cycle close to 50% might be used. The power handling capability has considerably decreased compared to the low duty cycle operations, but is still within the power handling requirements.

Another factor to consider when dealing with high-power signals in this circuit is that even though the thicker intrinsic region is advantageous for its lower minimum negative biasing requirements and a higher power handling capacity, it increases the insertion loss and switching speeds of the PIN diode.
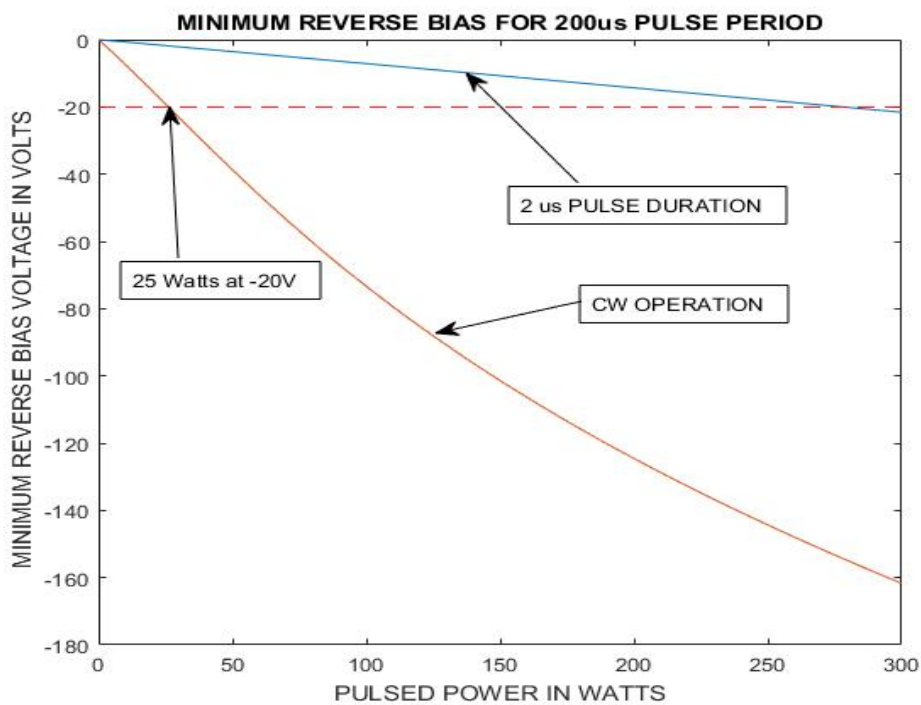
Figure 5.1.1: Calculated pulsed power vs required reverse bias voltage for a 2-μs pulse with a pulse period of 200 μs.
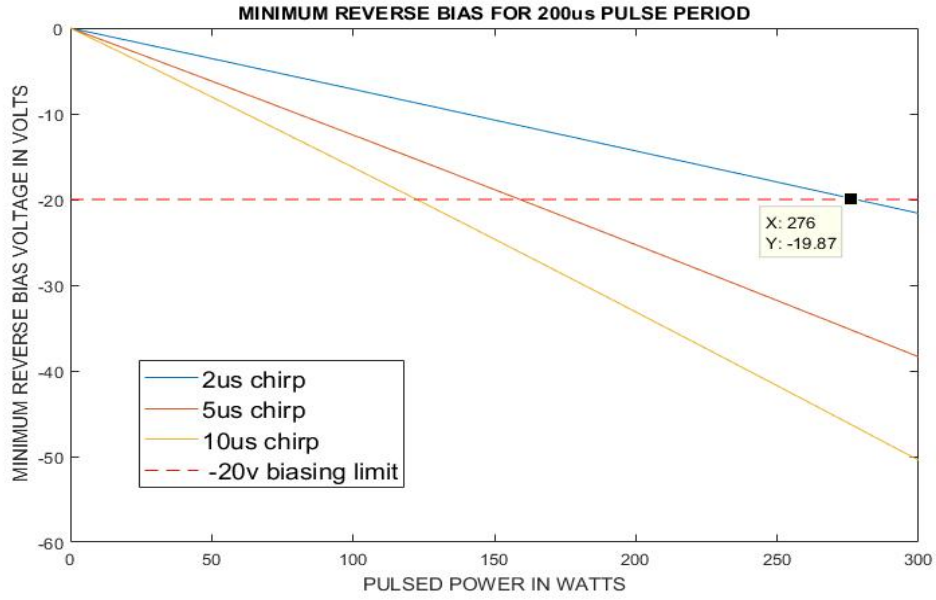
Figure 5.1.2: Calculated pulsed power vs required reverse bias voltage illustrating the effect of pulse duty cycle on the minimum reverse bias requirements
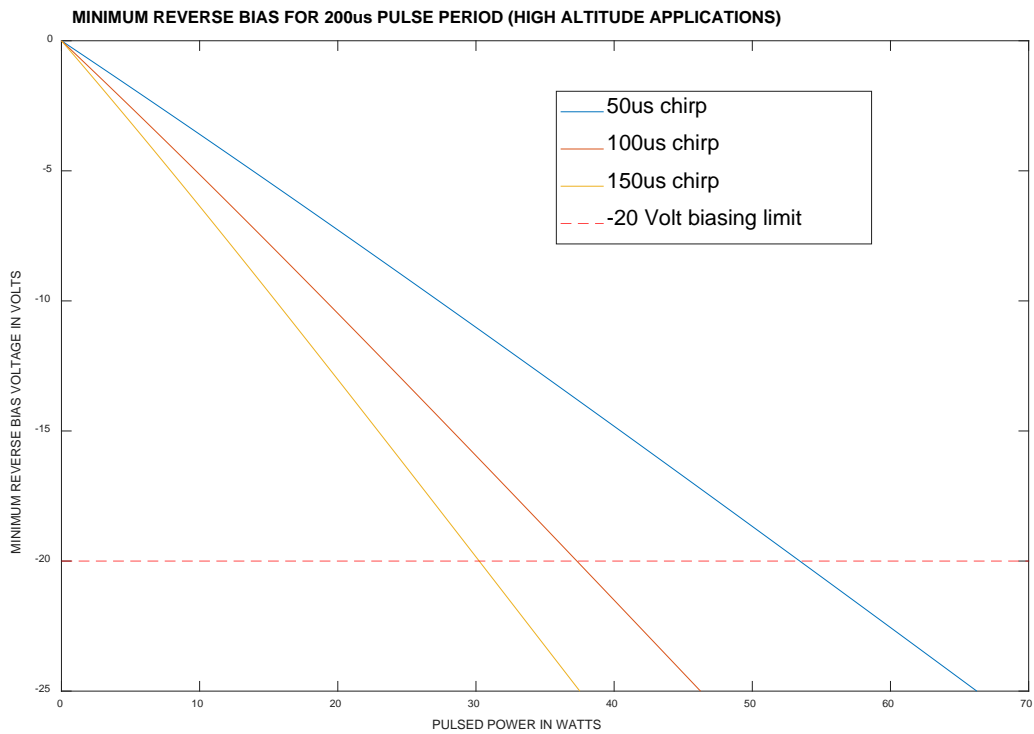
Figure 5.1.3: Calculated pulsed power vs required reverse bias for a high altitude operation mode using long duty cycle.

5.2 DC blocks, RF chokes, termination resistor, ribbon bond, SMA interface connector, copper surface finish and transmission line type considerations

 The DC blocking capacitors used for the bias tee circuit for each diode have to possess wide-bandwidth and low-loss characteristics. High-performance capacitors P42BN820Z5ST from Dielectric Labs (Figure 5.2.1.a) were found to be the lowest loss DC blocks in the market with the required frequency specifications and also has gold terminations which makes jumpering to the PIN diodes easy. These DC blocks have a VDC rating of 50 Volts and can handle the -20 Volt bias applied across it. The pulsed RF peak voltage across these capacitors is limited by the dielectric breakdown voltage of 125 Volts. The actual dielectric breakdown is about 40% higher than the specified breakdown (verified by the manufacturer) and this sets the upper limit on the peak RF voltage applied and corresponding power handling of the T/R switch. The maximum power handling of the T/R switch is set by the DC blocks and is about 158 Watts. To be conservative, taking into account reflections and other effects we can set the maximum power handling of the switch to be about 100 Watts.

 The other component of the bias tee circuit is the RF chokes, which should present a high reactance over the frequency range of interest. The RF chokes needed to be series resonance free for multi octave bandwidths while having low insertion loss when used in a shunt configuration. SMT broadband conical inductors from Piconics satisfy this requirement [50] (CC45T47K240G5-C2) and were chosen for the design (Figure 5.2.1.b). These inductors are capable of handling a DC current of 160 mA in continuous operation. The DC resistance for this inductor is about 1.6 Ohms due to the thin wires used and large number of turns. The switching transients produced by the inductors were also verified using ADS simulation models provided by Modelithics[51]. The inductors needed to be placed as close to the signal trace as possible to prevent stubbed resonances

by the trace. These inductors are specifically designed to be placed directly over the CPW (Coplanar wave guide) transmission lines or micro-stripline and interact with the electric fields from the signal trace [52]. The termination resistor used in the isolated port of the second hybrid coupler was the R2B130850R0J4J0 from Anaren, which is rated for operation up to 18 GHz and is in a SMT format.
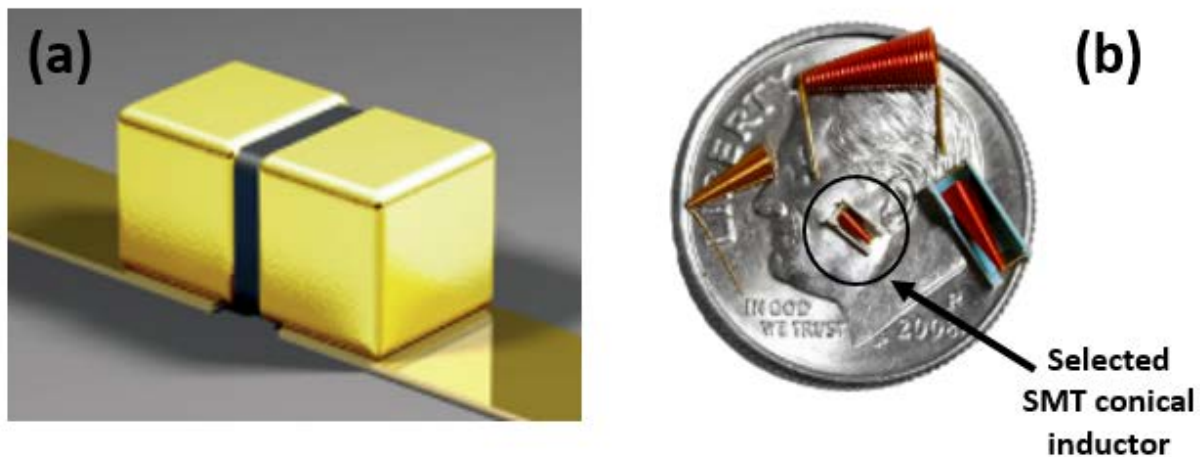
Figure 5.2.1: (a): Picture of the ultra-wide band DC blocks with gold terminations from Dielectric labs/ Knowles capacitors [53] and (b): Broadband conical inductors from Piconics on a quarter dollar coin [54].

Gold ribbon were selected as interconnects from the DC blocks to the PIN diodes and the PIN diodes to the RF traces. Gold ribbons reduce the inductance and increases effective cross sectional area for current conduction and increases the power handling capacity as compared to gold wire. A ribbon width of 3 mils was chosen to match the 3 mil wide gold trace on the PIN diode and was verified to safely handle at least 200 Watts of CW power for a 20 mil length [55].

Other important RF components were the SMA connector interfaces. The connectors selected needed to be of low loss over the operating frequencies and well matched to the selected dielectric

116

thickness and trace width considerations. The selected connectors at the transmitter and receiver ports were 1092-02A-6 end launches from Southwest connectors employing 2.92 mm pins. The antenna port SMA interface connector was a two-hole flange female SMA connector 214-510SF from Southwest connectors.

Once that the critical microwave components for the T/R switch design were selected, the next step was to fix the PCB copper surface finish. The primary factor here is the gold ribbon bond-ability. ENIG (electro-less nickel immersion gold) and ENIPEG (electro-less nickel electro-less palladium immersion gold) are the commonly used surface finishes for wire and ribbon bonds. ENIPEG offers a much higher bond strength than an ENIG finish. However, ENIPEG finish has a much higher insertion loss than ENIG due to the thicker gold and palladium layers [56, 57]. We decided to go with the ENIG finish as the ribbon bonds used are wide enough at the surface contact area to compensate for the loss in strength as compared to the ENIPEG finish.

The last consideration in this section is the selection of the transmission line configuration for the PCB RF traces. GCPW (grounded conductor backed coplanar wave guide) was chosen over microstrip in spite of design complexity to achieve better dispersion properties [58] and to keep the electric fields confined to the trace area to prevent coupling effects with the surrounding traces and components.

5.3 Receiver circuit selection (LNA and Sequencer)

The T/R switch is designed with integrated LNA at the receiver port to achieve improvements in SNR while keeping the added noise minimal. The selected LNA was the QPM1000 from Qorvo which comes in a QFN (quad flat no leads) package. The LNA has a typical gain of greater than 17 dB, a noise figure of less than 2 dB from 3-12 GHz and less than 4 dB toward the band edges.

The QPM1000 has a RF power limiter integrated into it to protect the LNA from switching transients and leakage from the transmitter. The RF input and output ports are DC blocked. The functional diagram of the LNA is shown in Figure 5.3.1 while the basic parameters are given in Table 5.3.1
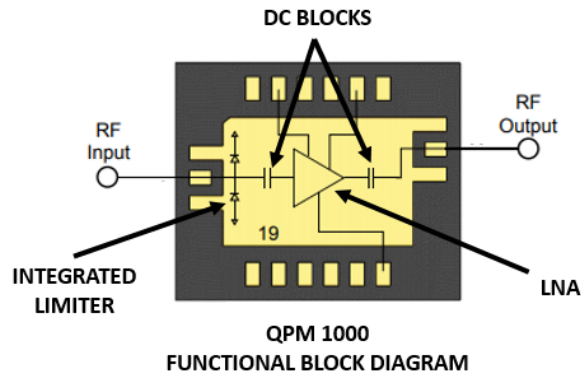
Figure 5.3.1: Functional diagram of the QPM1000 LNA/ limiter [59]

QPM1000 parameter	Value
Frequency range	2-20 GHz
Input power CW survivability	4 Watts
Gain	>17 dB (Adjustable using V_{G2})
Noise figure	< 2 dB (3-12 GHz), < 4 dB (outer frequencies)
IM3	< -21 dBc
Bias voltages	V_D=5V, I_D=100mA, V_{G1}=-0.6V(typical),V_{G2}=1.3V

The LNA supply and bias voltages have to be turned on in a specific sequence to set the drain current without damaging the device. The voltage sequencer used for this purpose was the HMC981LP3E, an active bias controller which is capable of supporting the sequencing requirements and supply voltages for the LNA. The schematic diagram for the LNA-sequencer section of the circuit is shown in Figure 5.3.2.
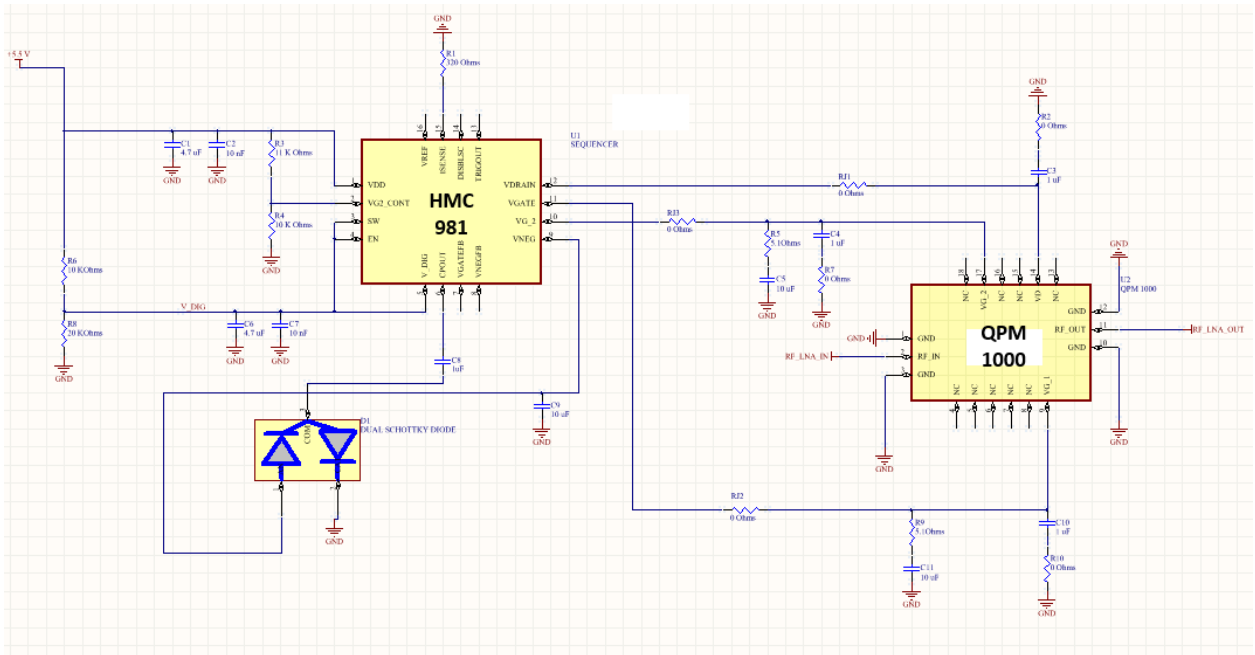
Figure 5.3.2: Schematic diagram of the LNA/sequencer receiver section

5.4 T/R switch RF devices placement considerations

Careful arrangement of selected RF and bias components was done to minimize the number of components, achieve minimal insertion losses and to facilitate fabrication. The DC block DCB 1 isolates the transmitter from the bias voltages applied to shunt diode D1. The DC block at the input of the quadrature coupler 90° port is removed and the DC blocking is now achieved by the

integrated DC block of the LNA RF input. The DC block at the 0° port of the LNA is moved to the summation port of the quadrature hybrid B. This was done to maintain symmetry at the ports. The biasing circuitry for the diodes D2 and D3 is placed in proximity to the respective diodes. This placement ensures that the signal from the transmitter does not leak into the bias circuit and only the residual signal after the isolation from the shunt diode encounters the bias inductors. This change reduces the insertion loss in the transmission mode. BC 1, BC 2, BC 3 are bypass capacitors for the biasing supply. The capacitors used for bypass are the same as the DC blocks.
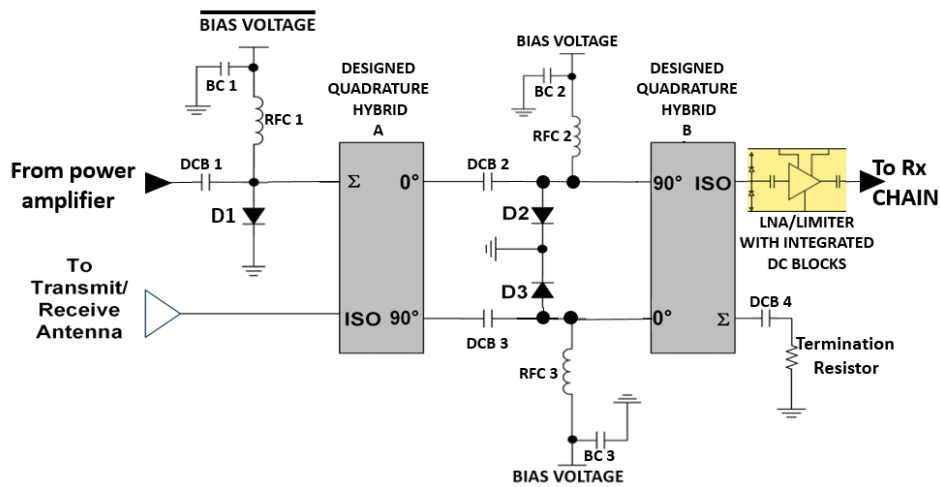
Figure 5.4.1: Finalized RF component placement architecture

5.5 PIN diode driver circuit

The driver circuitry is required to switch the PIN diodes between the ON and OFF states under the 2.3 µs switching requirements to keep the blind range low. The following sections explains the description and functionality of the PIN diode driver.

5.5.1 PIN diode driver description

120

The switching control input to the T/R switch is TTL signals of (+5 V and 0 Volt standard) and cannot drive the PIN diodes with the high current and relatively high negative voltage requirements. Transistors are used to provide the required drive currents to switch the PIN diode states. During the turn 'on' of the diodes the driver needs to provide an inrush current of at least 100 mA of current to reduce the forward series resistance and provide for a low reflective impedance path. The transistors used must have a low junction capacitance to flood the intrinsic region of the PIN diode fast enough. During the turn 'off' state the driver must apply the required negative bias voltage fast enough to completely turn off the diode and provide a closed circuit path. The T/R switch is in transmission mode when the control signal goes high and the switch is in receive mode in the low control state. The driver for the transmitter port diode is complementary to the control signal and turns off during the transmit state to close the circuit and turns on during the receiver state to provide a reflective impedance to leakages from the power amplifier stage.

5.5.2 PIN diode driver functionality

The PIN diode driver is based on a design recommended by Borroson[60] and was adapted by Ahmed[16] for T/R switch drivers used for depth sounding radars. The adapted Borroson PIN diode driver schematic is shown in Figure 5.5.2.1.c. When the TTL control at the input of the driver goes low the transistor Q1 goes into saturation and connects the +5 Volts supply path to the load. The diode D6 turns on and sends the transistor Q2 into cutoff mode and isolates the load from the negative supply. The capacitor C51 speeds up the switching by acting as a charge reservoir. The resistor R28 limits the drive current to maximum of 100 mA.

When the TTL input to the driver goes high, Q2 is in saturation mode and provides a path from the -20 Volt to the load, C53 acts as the charge reservoir to speed up the switching time. The diode D6 turns off which causes the transistor Q1 into cutoff. At any instance the positive and negative supplies are prevented from seeing each other by this design. The schottky diodes D5 and D7 maintain the base emitter voltage at 0.4 Volts.

The TTL input is first buffered using a non-inverting buffer chip (Figure 5.5.2.1.a) The buffer/transceiver chip used is SN74ABTH25245DWR. The important characteristic of this transceiver is shown in Table 5.5.2.1. This buffer provides a high current output to drive the next stages and does not load the input TTL signals. One of the issues with the buffer chip is the low output voltage of 3.3 Volts which does not turn on the transistors and the other issue is that the PIN diode is complimentary in nature which means that it turns on the PIN diodes in the TTL off state and turns off the diodes the TTL on state. To solve this, issue an inverting buffer chip EL7232 is used which has a low delay of 25 ns. The important parameters of the inverter chip are given in Table 5.5.2.2.

The transistors used are MMBT2907A PNP transistors. The transistors have a collector to emitter voltage of 60 Volts. The maximum current rating is 800 mA which is well above the required 100 mA turn on current for the PIN diodes. The transistor delay is 10 ns which keeps the driver switching time low. The diodes D6 and D7 are 1N4148 by Fairchild semiconductors with a current handling capacity of 150 mA in continuous operation and 300 mA maximum. The breakdown voltage for the didoes is 100 volts and reverse bias current of 100 micro amps. The diode D5 is a schottky diode BAT42XV2 from Fairchild semiconductors with a 30-volt breakdown and reverse bias current of 100 micro-amps. The switching time for this device is less than 5 ns.

Figure 5.5.2.1: Schematic of the PIN diode driver circuit (a): The inverter circuit driving the PIN diode driver (b): The transceiver chip used to buffer the input TTL control signal. (b): The PIN diode driver circuit.

Parameter	Characteristics	Unit
V_{CC}	4.5 – 5.5 (used at 5)	V
V_{IH}	2	V
V_{IL}	0.8	V
I_{OH}	-80	mA
I_{OL}	188	mA
t_{PHL}	4.3	ns

Table 5.5.2.1: Important parameters of the SN74ABTH25245DWR transceiver chip

Parameter	Characteristics	Unit
V_{CC}	4.5-16 (used at 5)	V
V_{IH}	2.4	V
V_{IL}	0.8	V
I_{PK}	2	A
t_{PHL}	25	ns

Table 5.5.2.2: Important parameters of the EL7232 inverting buffer chip

5.5.3 PIN diode driver switching times

We performed timing simulations in ADS using transient analysis to estimate the switching times of the PIN diode driver with and without the diode/bias tee load. The PIN diode driver switching times were simulated by ADS transient analysis tool using spice models for the transistor and diode devices and ideal capacitor and resistor elements. The TTL control input was used to simulate a standard control signal from the radar digital system with a logic high voltage of 5 Volts and a low voltage logic of 0 Volts. The rise and fall times of the TTL control signals are 5 ns each. The pulse duration is 2 µs (for a 1 µs short pulsed mode). The un-loaded waveforms of the PIN diode driver are shown in figure 5.5.3.1. The time taken for transition of the PIN diode driver output from -20 Volts to about 5 Volts is about 330 ns. The time taken for the PIN diode driver to switch from +5 Volts to -20 Volts is about 120 ns.
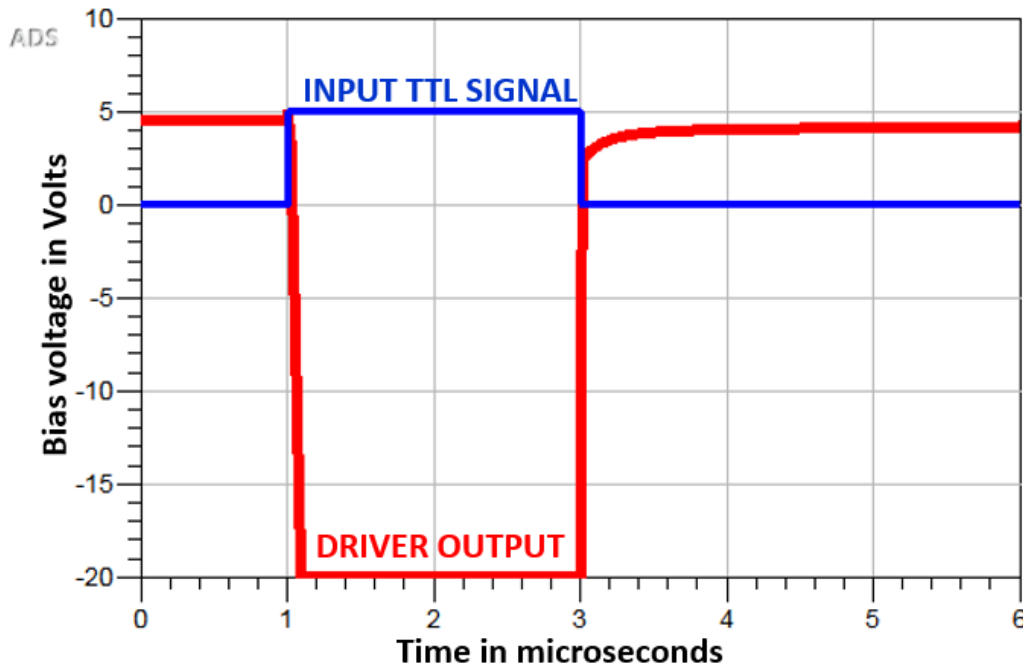
Figure 5.5.3.1: ADS simulated un-loaded switching time of the PIN diode driver driven by a TTL control

The transient analysis of the PIN driver output under load from the shunt diode (ADS PIN diode model) and the bias tee (Piconics simulation model) is shown in figure 5.5.3.2. The output voltage high voltage doesn't reach all the way down to 0 Volts due to the DC resistance of the inductor. The time for switching 'off' the PIN diode is about 250 ns and the switching 'on' is about 340 ns in simulations.
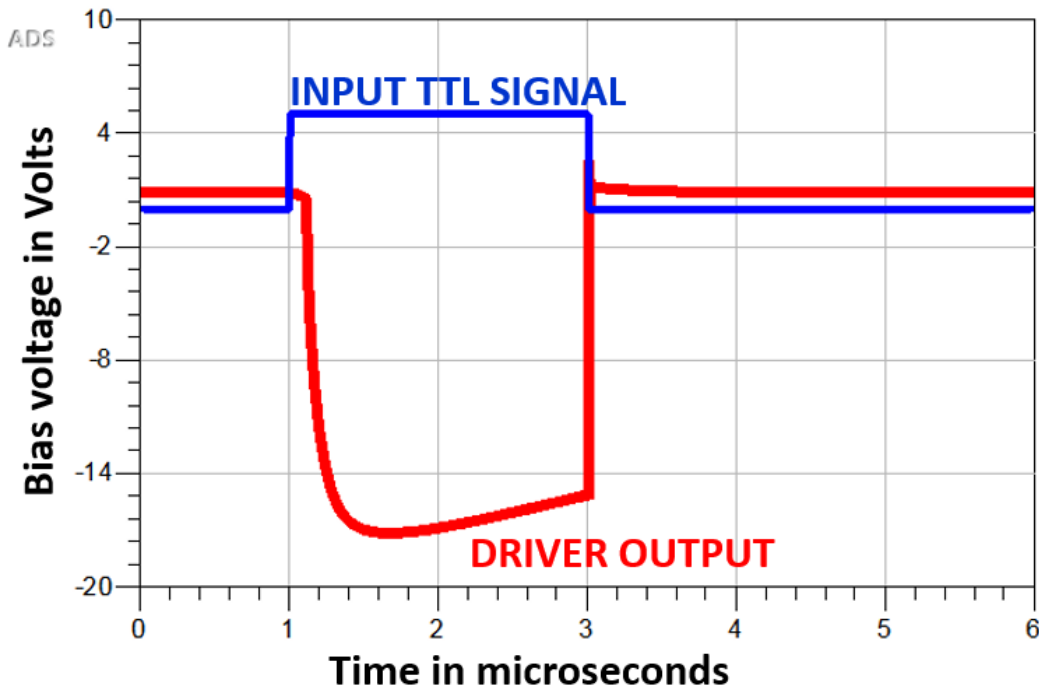
Figure 5.5.3.2: ADS simulated loaded switching time of the PIN diode driver driven by a TTL control

We measured the switching times for the loaded and unloaded PIN diode driver with oscilloscopes. The TTL controls were provided by a pulse generator and was set to 2 µs pulse duration (for a 1 µs short pulsed mode) and a 200 µs pulse period for typical the typical operating mode. The logic high and low levels were 5 Volts and 0 Volts respectively. The unloaded switching time of the PIN diode driver measured using a digital oscilloscope is shown in figure 5.5.3.3. The turn 'off' time for the driver is about 318 ns and the turn 'on' time is about 118 ns.
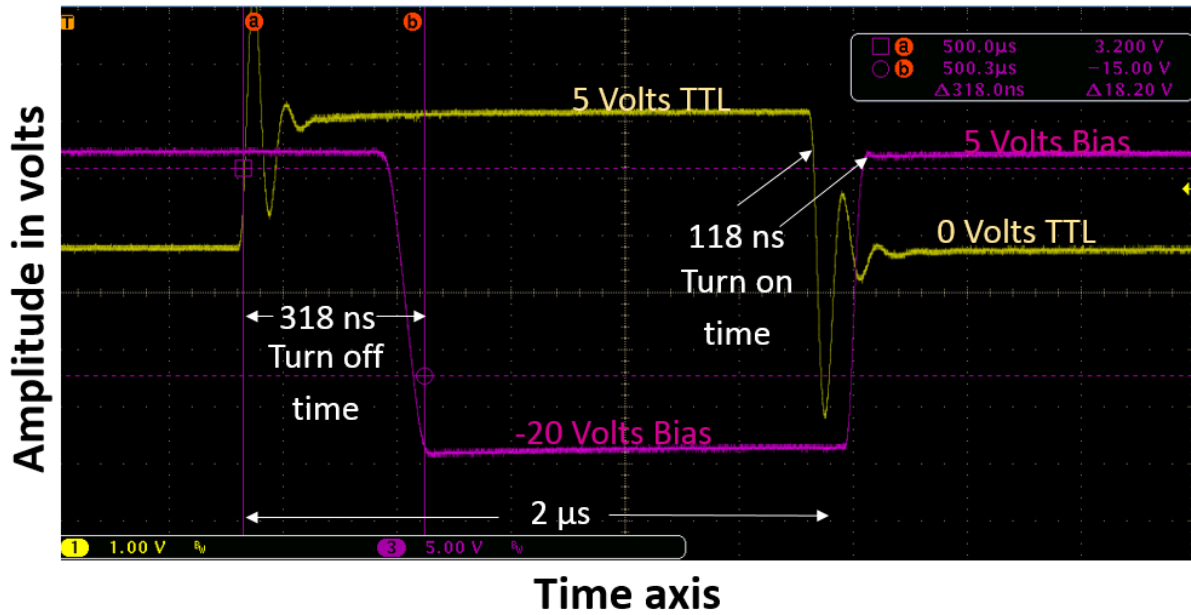
Figure 5.5.3.3: Measured un-loaded switching time of the PIN diode driver driven by a TTL control line
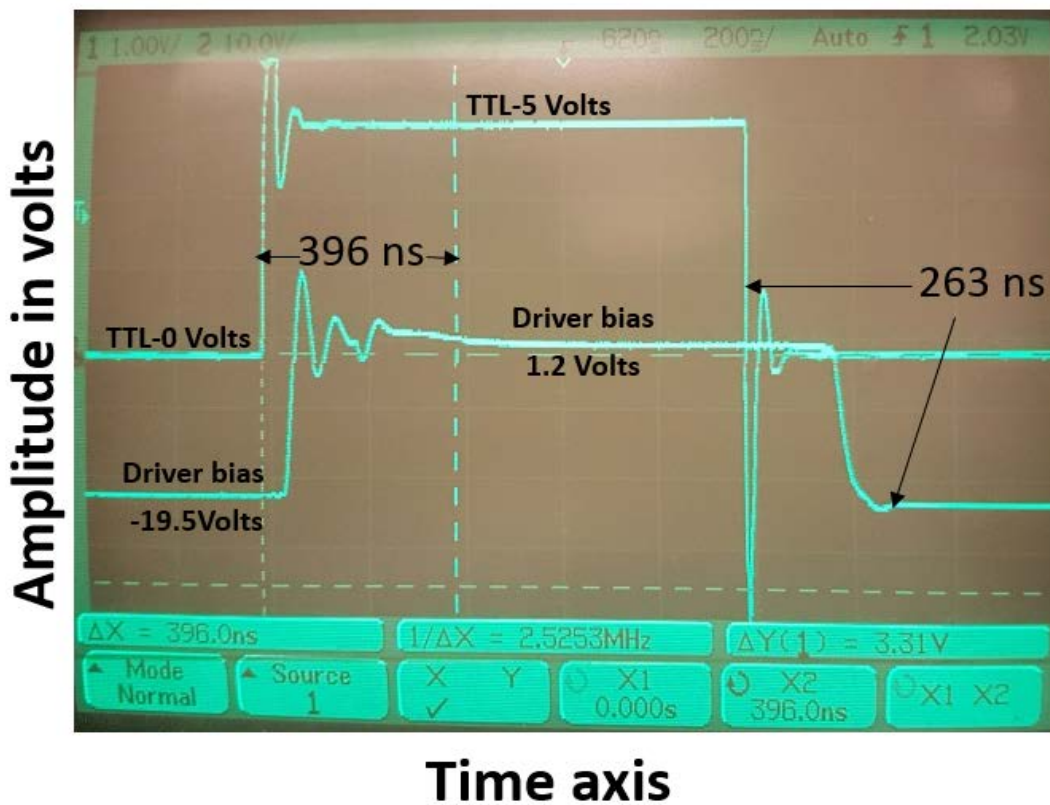
Figure 5.5.3.4: Measured loaded switching time of the PIN diode driver at the receive path diodes D2 and D3 driven by a TTL control line
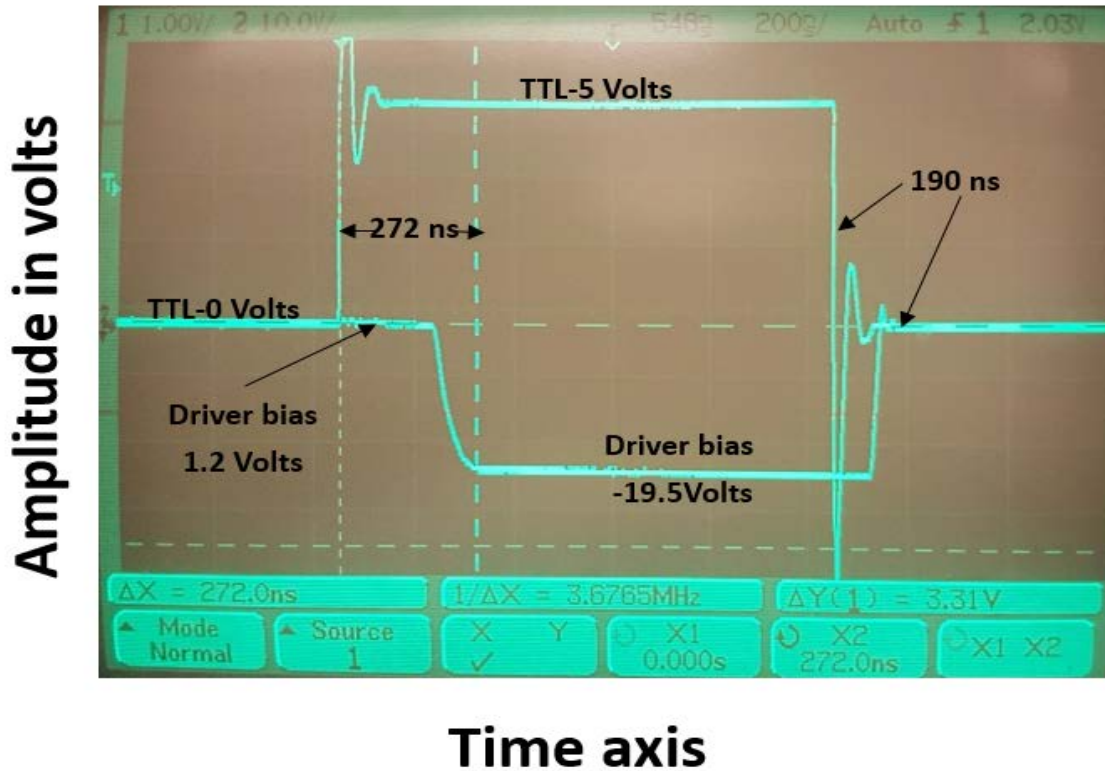
Figure 5.5.3.5: Measured loaded switching time of the PIN diode driver at the transmit path diode D1 driven by a TTL control line

The loaded driver measurements were done by a CRT (Cathode ray tube) Oscilloscope after connecting the PIN diode driver output to the shunt diodes and the bias tee components. The driver output at the receiver path diodes D2 and D3 shows a turn 'on' time of about 396 ns and a turn 'off' time of about 263 ns as shown in figure 5.5.3.4. The driver output at the transmitter path diode D1 shows a turn 'off' time of about 272 ns and a turn 'on' time of about 190 ns as shown in figure 5.5.3.5. These switching times were well under the requirements.

5.5.4 TR switch switching time constraints.

From the measured switching times of the loaded PIN driver circuits we can set switching time constraints for the TR switch. The largest switching time sets the wait time constraint in each case.

127

In transmit mode the time is set by the diodes D2 and D3 turn 'on' time of 396 ns. In receive mode the constraint is set by the turn 'off' time for the diodes D2 and D3 which is 263 ns. Table 5.5.4.1 shows the time constraints in the receive and transmit modes of operation.

Mode of operation	Minimum wait time constraint (from TTL control signal)
Transmit mode	396 nanoseconds
Receive mode	263 nanoseconds

Table 5.5.4.1: Measured loaded switching time of the PIN diode driver at the transmit path diode D1 driven by a TTL control line

5.6 PCB board layout and RF trace optimization

One of the critical aspects in the TR switch implementation is to have properly designed GCPW transmission lines and have a good impedance match at all points along the signal trace path. This section describes the optimizations done on the PCB boards to ensure good return loss and insertion loss characteristics for the TR switch. This section also shows the final designed TR switch PCB board layout.

5.6.1 SMA end launch trace tapering

The signal trace at the SMA end launch connector pin to trace interface had to be tapered to minimize reflections caused due to impedance mismatch. The trace has to be tapered to reduce the excess capacitance from the signal trace to ground plane along the length of the launch pin. Care had to be taken as to not to make the tapers too narrow and increase the inductance at the launch

pin interface. HFSS tuning was done to find optimal tapering at these points and keep return loss to a minimum.
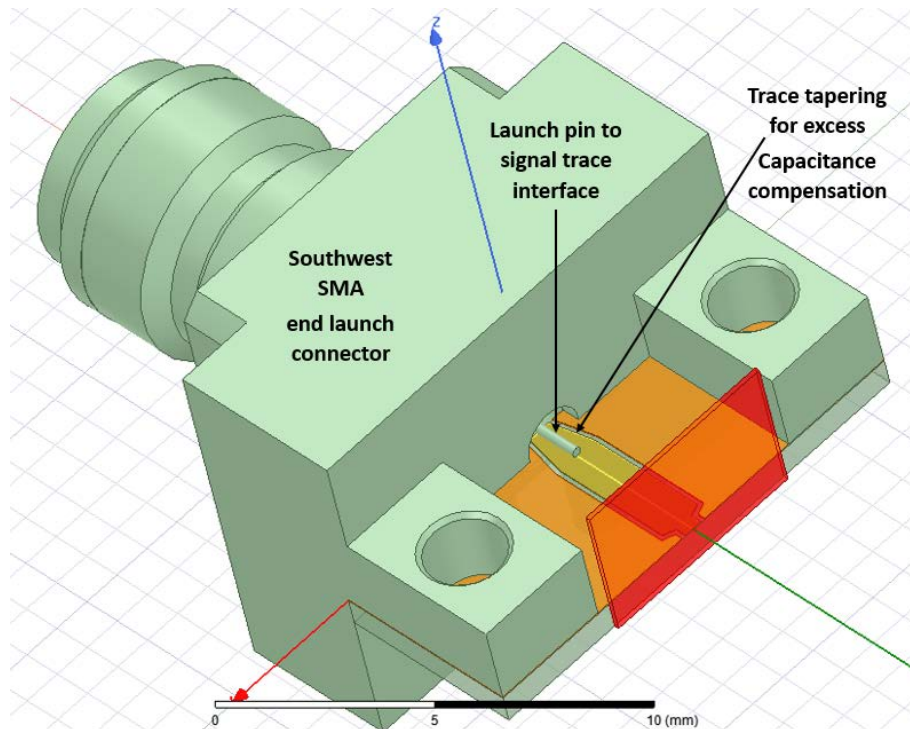
Figure 5.6.1.1: HFSS model of SMA end launch pin to signal trace interface tapering to compensate for excess capacitance at the launch.

5.6.2 Via spacing considerations

One of the assumptions of grounded conductor backed coplanar waveguide (GCPW) transmission line is the microwave guiding metal walls between the ground planes[61] as shown in figure 5.6.2.1.a. This microwave walled effect can be achieved by using closely spaced via's. The via's have to be spaced under a λ/20 spacing between the via's over the whole frequency of operation as shown in figure 5.6.2.1.b. The spacing of the via's from the start of the ground trace have to be as minimal as possible. Closest possible spacing under the PCB board fabrication house tolerance was a 20 mil spacing from one via hole center to the next. The via diameter was chosen to be 10

mils which was the smallest via feasible for the dielectric laminate thickness. The spacing between

the via and the start of the ground trace was selected as10 mils.
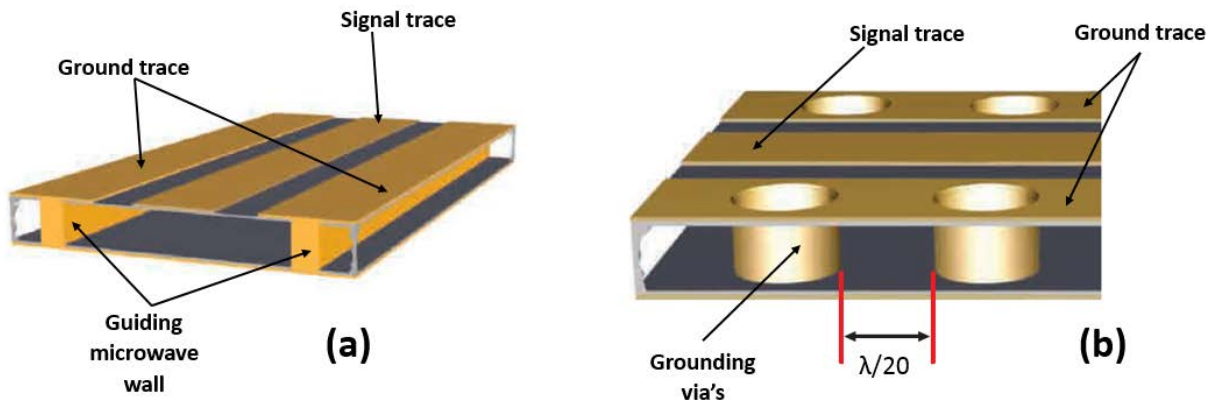
Figure 5.6.2.1: (a): Microwave guiding wall between the grounding traces[61]. (b): Via's spaced closely to
approximate a microwave guiding wall in a GCPW transmission line[61].

5.6.3 Stripline-GCPW transitions

A critical design consideration for the RF signal routing was the transition from the stripline mode

of the coupler to GCPW mode for RF component placement. Several architectures were explored

and HFSS simulations performed and the best transition was found to be a direct transition with a

continuity in the ground paths as shown in figure 5.6.3.1. The signal trace is jumpered using a

circular tab which can be solder bridged. The propagation modes between these transitions are not

disturbed by a large margin. The return loss for this transition was simulated to be better than 20

dB over the operating bandwidth.
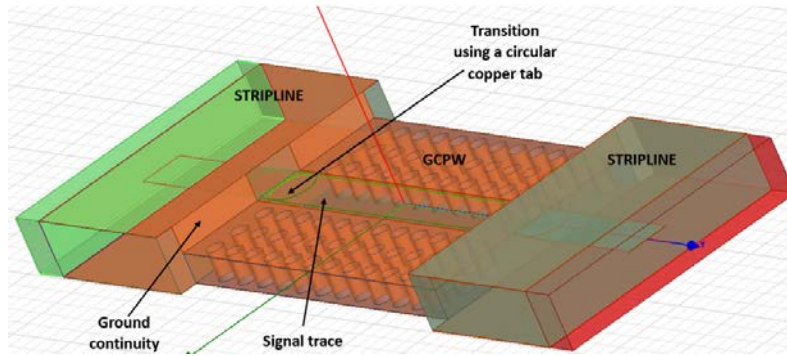
Figure 5.6.3.1: HFSS model of stripline to GCPW transition with ground path continuity preserved.

5.6.4 Signal trace tapering for RF components

The RF component footprints are not exactly matched to the signal trace and proper tapering is required to guide the signals onto the components without inducing significant reflections. Trace tapering's such as Klopfenstein tapers, triangular tapers were simulated. The shortest tapering was obtained by using triangular tapers with the taper transitions being much shorter than $\lambda/20$.

5.6.5 Ground path continuity

The return paths for the signal have to be well routed at these high frequencies. Discontinuities in the round paths could cause reflections and resonances. Ground return currents pick up structures were appended to the top dielectric layer PCB to transfer the ground plane routing between the stripline and GCPW modes. Figure 5.6.5.1 shows the ground return paths provided using grounding via's.
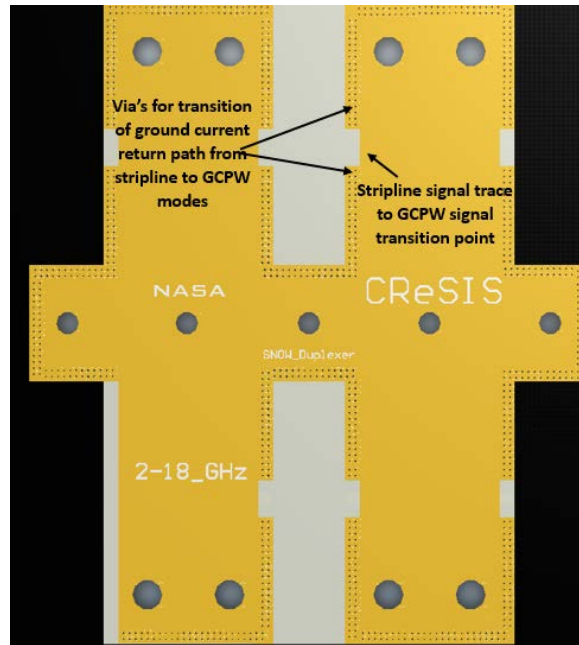
Figure 5.6.5.1: Altium 3D PCB board model showing ground return current pick up points provided on the top layer PCB of the TR switch.

5.6.6 PCB board and TR switch assembly

The TR switch PCB boards was designed using Altium designer. The three dielectric laminate layers were fabricated as 3 PCB boards. The bottom PCB board contains the switching circuitry, power regulator stage, LNA and sequencer stage and the RF components (shown in figure 5.6.6.1.a). The center PCB board contains the coupler (shown in figure 5.6.6.1.b). The top PCB board contains the top dielectric layer and ground current transition via structures (shown in figure 5.6.6.1.c).

The fabricated boards are assembled into a single unit using top and bottom aluminum covers similar to the assembled quadrature hybrid couplers. The arrangements of the aluminum cover and fabricated PCB board arranged in the order of assembling from top to bottom is shown in figure 5.6.6.2. The assembled unit can be easily integrated into a radar chassis.

(a)

PIN diode drivers , LNA stage, power regulation stage along
with RF components

Quadrature coupler a and b

(b)

Top dielectric layer with
ground path transitions

(c)
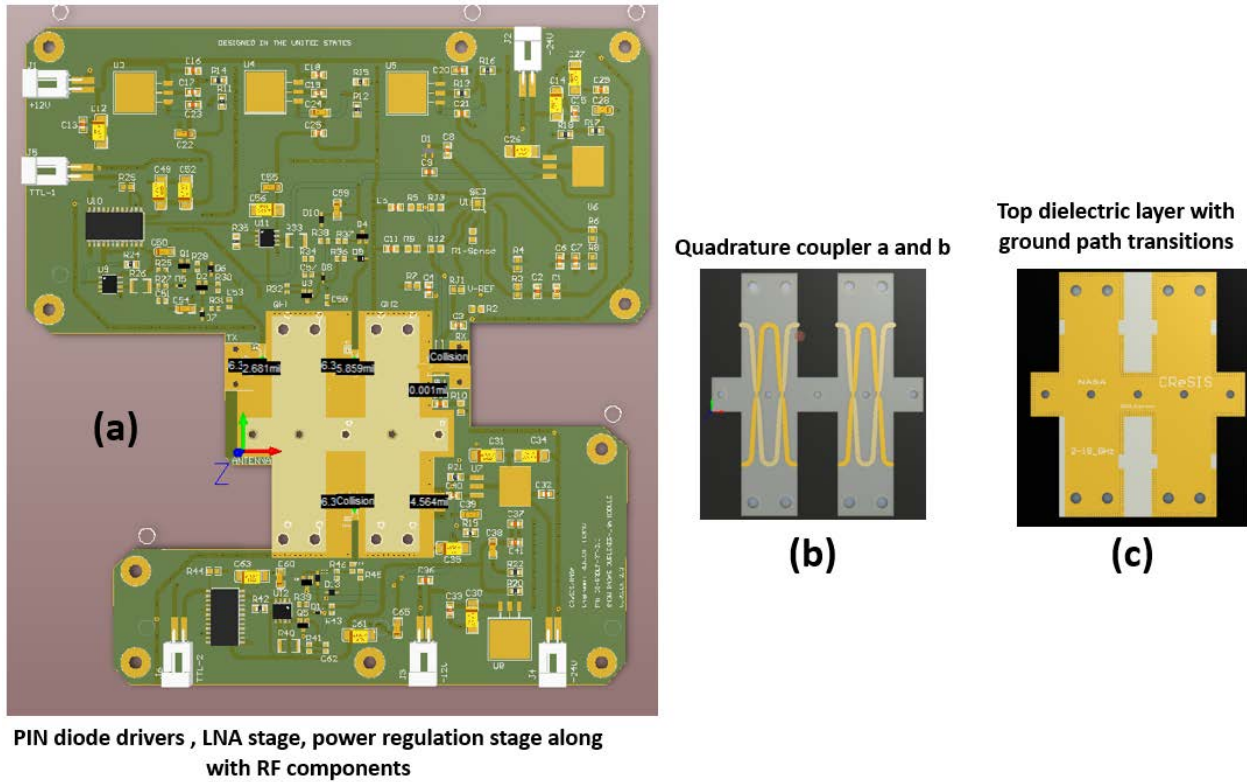
Figure 5.6.6.1: Altium 3D PCB board models showing (a): Bottom PCB board, (b) Center PCB board, (c) Top PCB board.

Bottom to top layer arrangement

Bottom aluminum cover

Bottom PCB board

Middle
PCB board

Top
PCB board

Top
aluminum cover

Layer-1

Layer-2

Layer-3
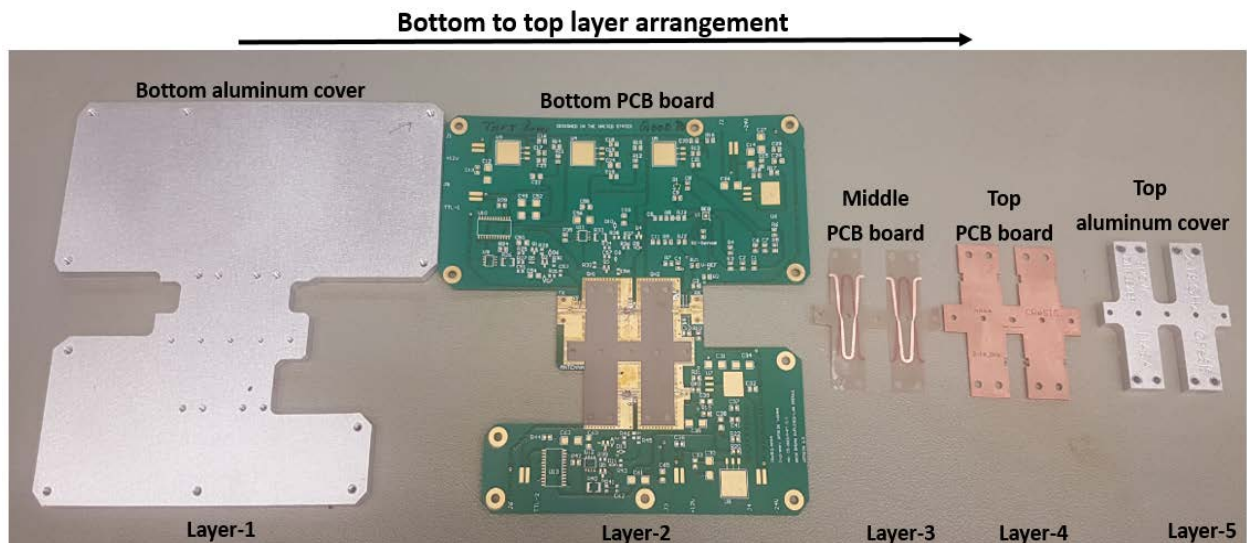
Layer-4

Layer-5

Figure 5.6.6.2: Fabricated PCB boards along with the aluminum covers to integrate the PCB boards into a single TR switch unit, where the arrangement is from bottom to top layer.

133
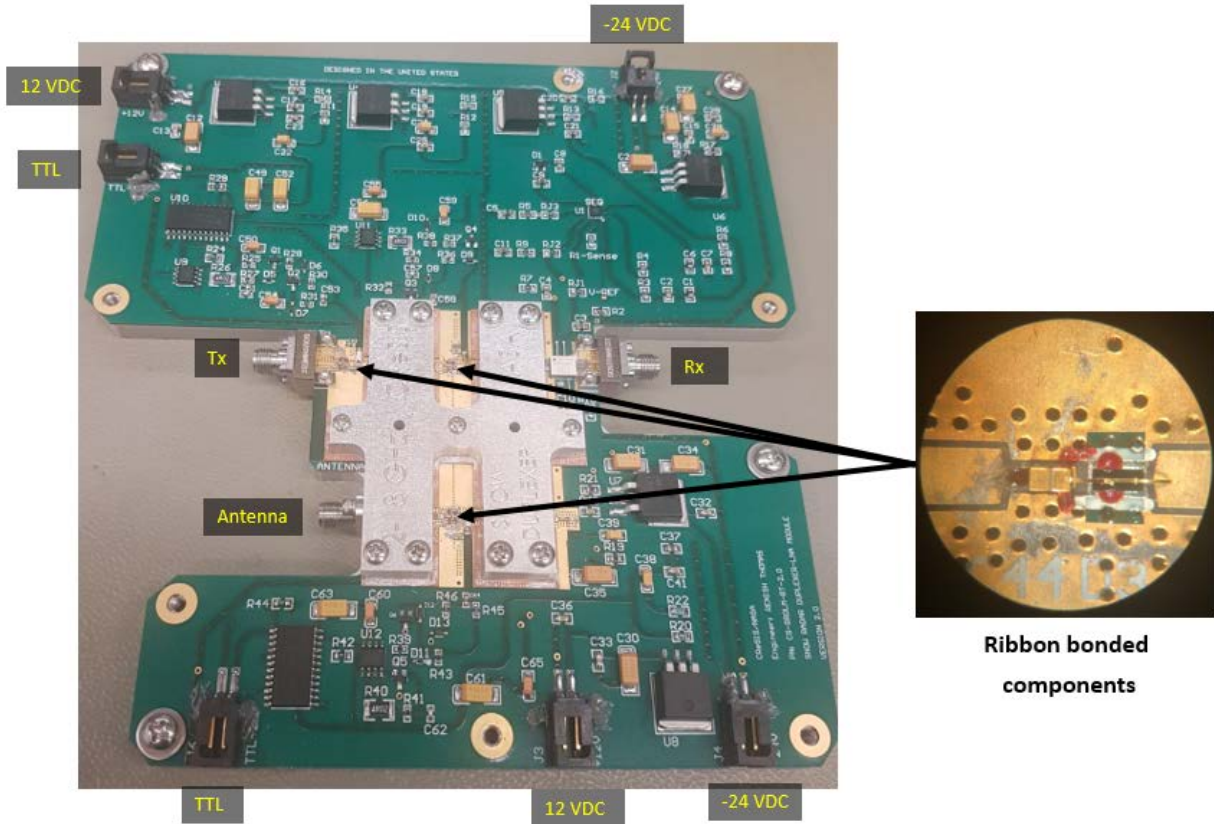
Figure 5.6.6.3: Fully assembled TR switch module with a zoom in of the ribbon bonded components.
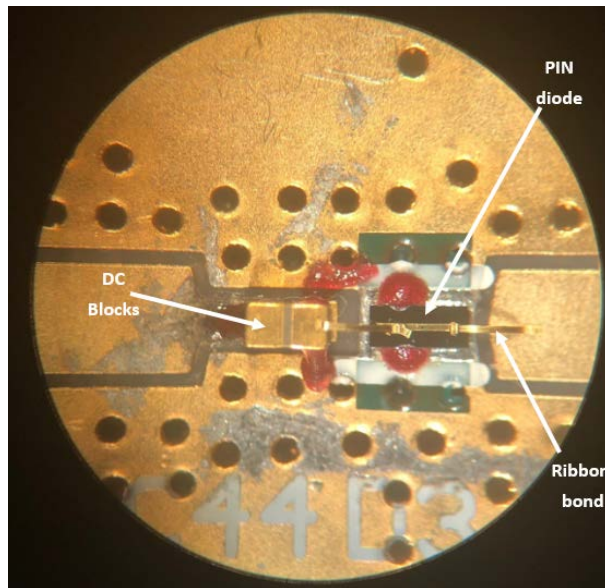
Figure 5.6.6.4: A close up view of the ribbon bonded RF components under the microscope.

The fully assembled TR switch module along with SMA end launches and SMA flange connector interface is shown in figure 5.6.6.3. C-grid type launch connectors provide connections for the +12 and -24 Volt supplies and for the TTL controls. A close up of the ribbon bonded parts is shown in figure 5.6.6.4. The ribbon bonding occurs at 3 locations.

5.7 Simulations of performance

Advanced designs systems (ADS) was used to perform circuit and electromagnetic (EM) co-simulations of the TR switch. The scattering parameters of the quadrature hybrid couplers used in these simulations is from the HFSS simulations of the designed stubbed quadrature hybrid couplers. The other component's scattering parameters are provided by the manufacturers.

5.7.1 Circuit simulation

The circuit simulations using just the s-parameters of the components were performed from 2 GHz to 18 GHz. Figure 5.7.1.1 shows the circuit simulation setup in ADS to perform the simulation using all the RF components and the HFSS optimized stubbed quadrature hybrid coupler s-parameters.

Figure 5.7.1.1: ADS 2-18 GHz simulation setup for the transmission and receive modes for circuit simulation of the RF components

135

The simulated receive mode antenna to receiver port gain is shown in figure 5.7.1.2. The lowest gain is at 18 GHz at about 15.146 dB. The isolation in the receive mode between the transmitter and antenna shown in figure 5.7.1.3 is greater than 60 dB across the frequency band. The isolation in the receive mode between the transmitter and receiver port shown in figure 5.7.1.3 is greater than 50 dB across most of the operating frequency band and approaches 40 dB nearing 2 GHz.

The simulated transmit mode transmitter to antenna port insertion loss is shown in figure 5.7.1.4. The highest insertion loss is at 16.2 GHz at about 3 dB. The isolation in the receive mode between the transmitter and receiver shown in figure 5.7.1.5 is greater than 20 dB across the frequency band and greater than 30 dB across most of the band. This lower than expected isolation is due to the LNA gain at the receiver. An SPST (Single pole single throw) switch can be used after the LNA stage to improve this isolation if the need arises.
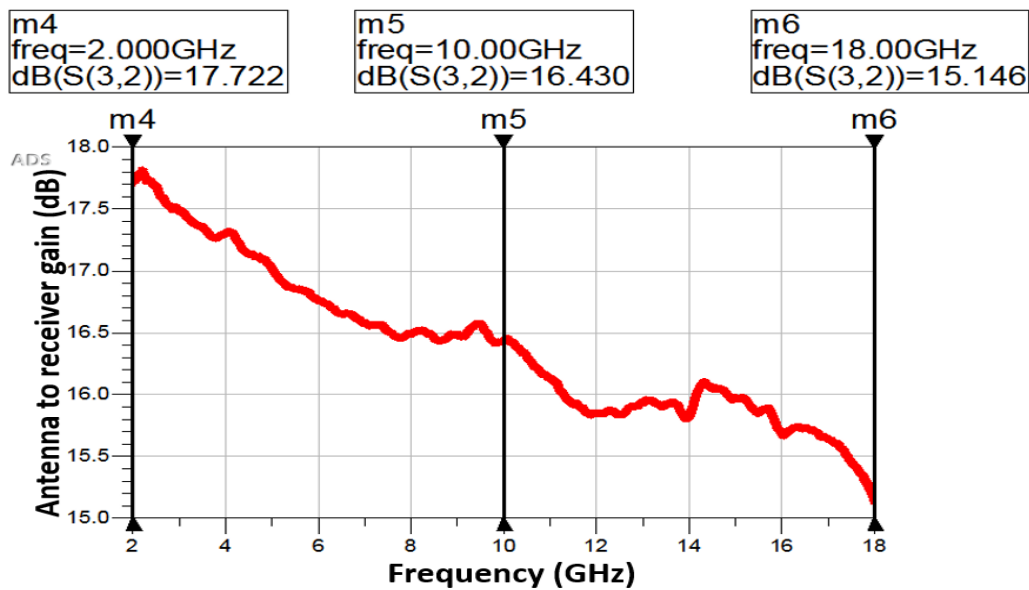
Figure 5.7.1.2: ADS 2-18 GHz simulation of the antenna to receiver gain during the receive mode.
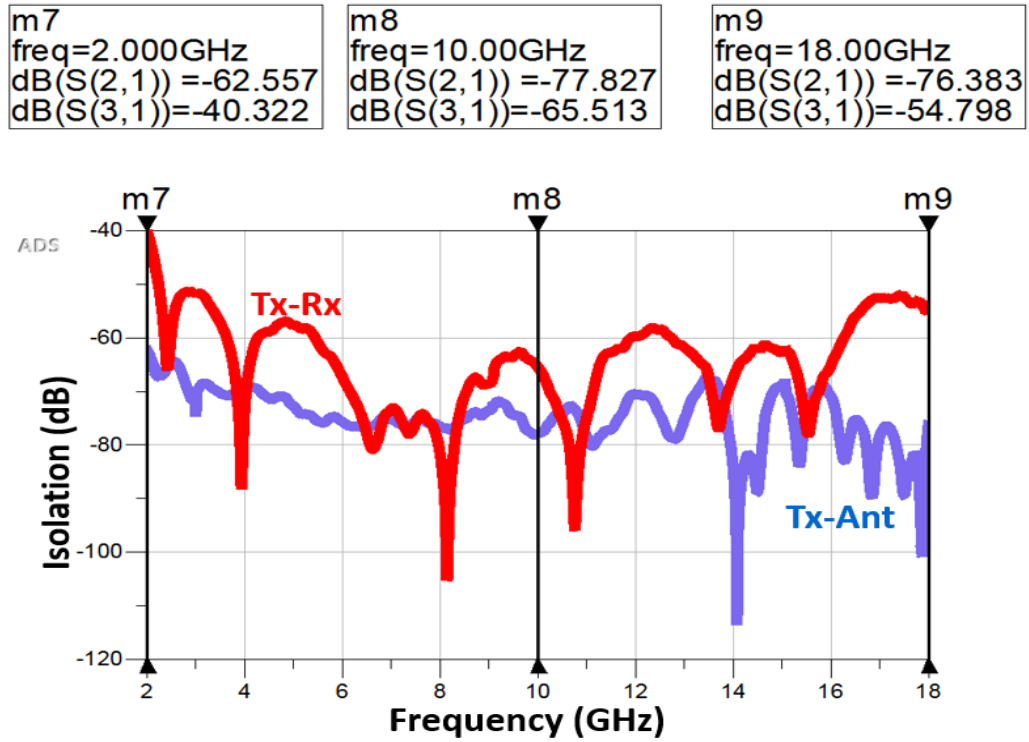
Figure 5.7.1.3: ADS 2-18 GHz simulation of the isolation between the transmitter and receiver and between the transmitter and antenna during the receive mode

137
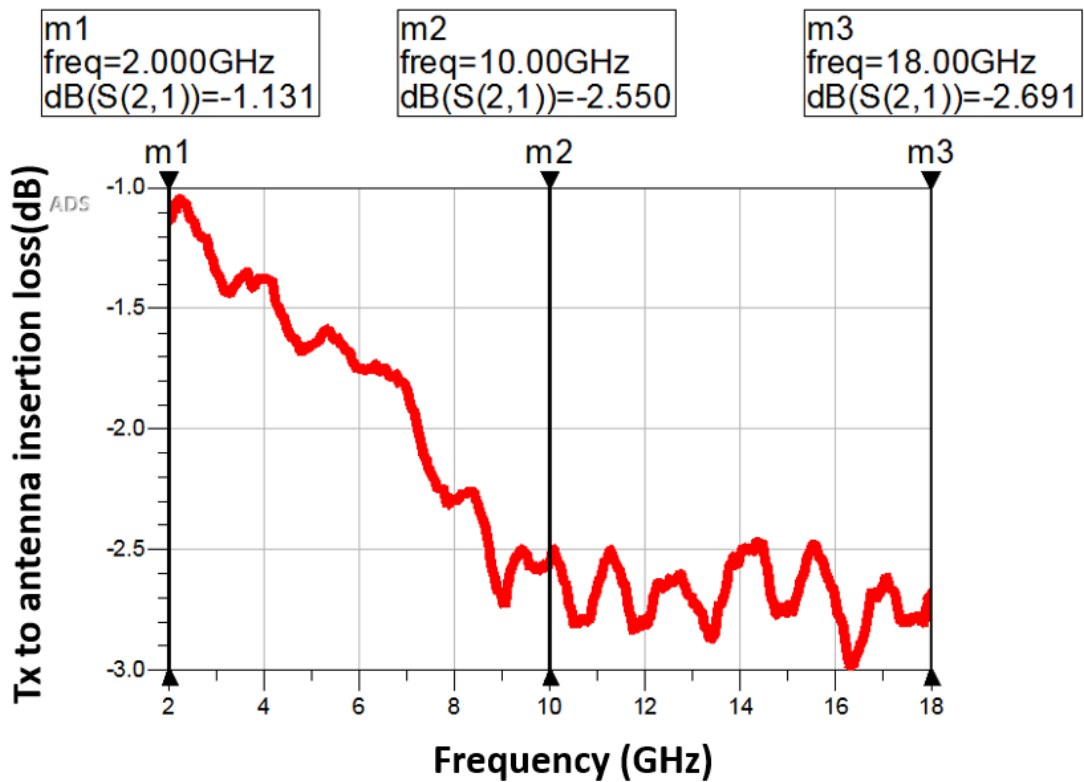
Figure 5.7.1.4: ADS 2-18 GHz simulation of the transmitter to antenna insertion loss during the transmit mode.
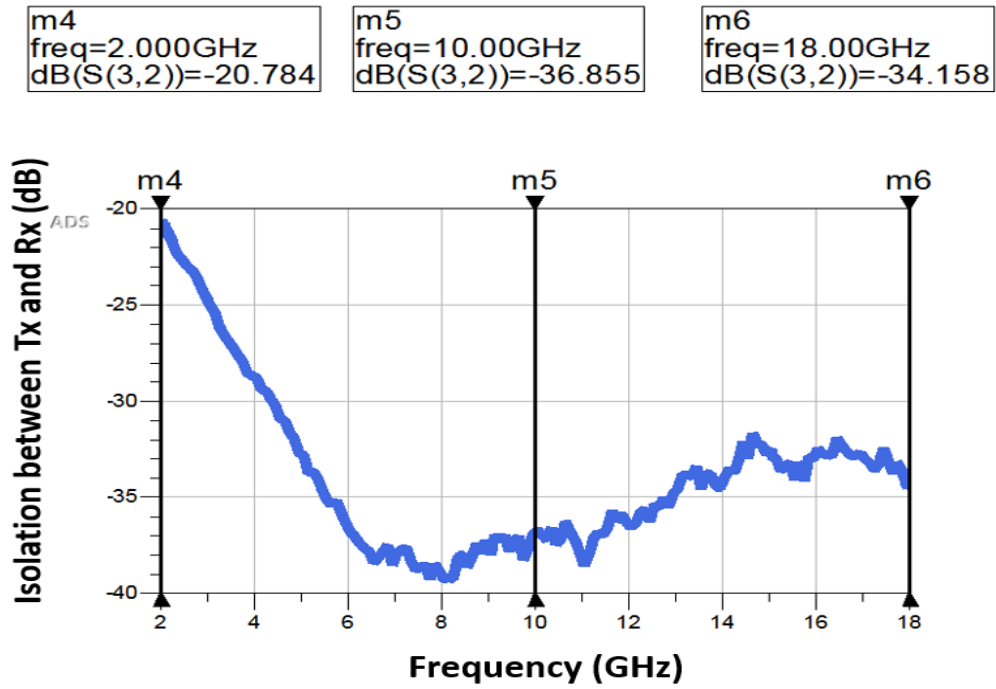
Figure 5.7.1.5: ADS 2-18 GHz simulation of the transmitter to receiver isolation during the transmit mode.

7. REFERENCES

1.   Panzer, B., et al., *An ultra-wideband, microwave radar for measuring snow thickness on sea ice and mapping near-surface internal layers in polar firn.* Journal of Glaciology, 2017. **59**(214): p. 244-254.

2.   Yan, J.-B., et al., *Airborne Measurements of Snow Thickness: Using ultrawide-band frequency-modulated-continuous-wave radars.* IEEE Geoscience and Remote Sensing Magazine, 2017. **5**(2): p. 57-76.

3.   *al,* H.-P.M.*e.*, *HELICOPTER-BASED MICROWAVE RADAR MEASUREMENTS IN ALPINE TERRAIN.* Proceedings of the 2008 International Snow Science Workshop, Whistler, British Columbia, 2008.

4.   Sand, K. and O. Bruland, *Application of Georadar for Snow Cover Surveying.* Hydrology Research, 1998. **29**(4-5): p. 361-370.

5.   Melzer, A., et al., *Short-Range Leakage Cancelation in FMCW Radar Transceivers Using an Artificial On-Chip Target.* IEEE Journal of Selected Topics in Signal Processing, 2015. **9**(8): p. 1650-1660.

6.   Mashury Wahab , Y.P.S., Yuyu Wahyu, Achmad Munir, *Isolation Improvement for X-Band FMCW Radar Transmit and Receive Antennas.* International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, 2016.

7.   Singh, J., *CARRIER DYNAMICS IN SEMICONDUCTORS.* Semiconductor devics basic priciples by Jasprit Singh.

8.   F. Rodriguez-Morales, S.G., F. Ahmed, C. Carabajal, A. Paden,, et al., *T/R Switches and Modules for Ice Sounding/Imaging Radar.* Aerospace & Defense E L E C T R O N I C S, 2016.

9.   Skyworks Solutions, I., *Design With PIN Diodes.* APPLICATION NOTE, 2012.

10.   TriQuint, *TGS2353-2   0.5 – 18 GHz High Power SPDT Switch.* Product Data sheet.

11.   al, P.C.e., *A new NMR duplexer made with quadrature couplers.* Measurement science and technology, 1991. **2147**.

12.   Wang, Z., et al., *Multichannel Wideband Synthetic Aperture Radar for Ice Sheet Remote Sensing: Development and the First Deployment in Antarctica.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2016. **9**(3): p. 980-993.

13.   ARVIND AGARWAL, S.G., K. AURBINDO, B. K. SARKAR, *HIGH POWER, LOW RECOVERY TIME BALANCED DUPLEXER FOR VHF APPLICATIONS.* Society for Applied Microwave Electronics Engineering & Research IIT Campus, Powai BOMBAY - 400 076, INDIA.

14. J.T. Vaughan, D.N.H., P.J. Noa', and G.M. Pohost, *RF Front End for a 4.1 Tesla Clinical NMR Spectrometer.* IEEE, 1995.

15. 2.1, J.L.-C.S.-T.S.-P.E.T.C.A., *New Generation of S band T/R Module.* IEEE, 1999.

16. Ahmed, S.F., *High-Power T/R circuits for a Multichannel VHF/UHF/HF Ice Imaging Radar.* Masters Thesis, 2015.

17. Arnold, E., et al., *HF/VHF Radar Sounding of Ice from Manned and Unmanned Airborne Platforms.* Geosciences, 2018. **8**(5).

18. KARIDI, K.T., *IMPROVEMENTS TO THE CReSIS HF-VHF SOUNDER & UHF ACCUMULATION RADAR.* Masters Thesis, 2018.

19. al, J.P.e., *ADAPTATION OF THE SNOW RADAR FOR NASA GLOBAL HAWK AND IKHANA UNMANNED AIRCRAFT IN SUPPORT OF OPERATION ICE BRIDGE.* 2017.

20. Pasternack, *PE71S2017.* Data sheet SPDT switch.

21. Qorvo, *QPM1000* Product Data sheet-LNA.

22. Jorgesen, D., *Top 7 Ways to Create a Quadrature (90˚) Phase Shift – Marki Microwave RF & Microwave.* April 2015.

23. SEYMOUR B. COHN, *History of Microwave Passive Components with Particular Attention to Directional Couplers.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1984. **MTT-32**(9).

24. Kaarsberg, R., *RF fundamentals: The 90-degree coupling phase shift.*

25. Microwave, M., *Microwave Power Dividers and Couplers Tutorial.*

26. Mauro Anselmi, M.P., Antonio Manna, Roberto Flamini, Luigi Cosmi, *Design and Realization of 3 dB hybrid stripline coupler in 0.5 - 18.0 GHz.* Proceedings of the 44th European Microwave Conference.

27. E.G. Crystal and L. Young, *Theory and Tables of Optimum Symmetrical TEM-Mode Coupled-Transmission-line Directional Couplers.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1965. **MTT-13**(5).

28. Levy, R., *Tables for Asymmetric Multi-Element Coupled-Transmission-Line Directional Couplers.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE.

29. C P. TRESSELT, *The Design and Construction of Broadband, High~Directivity, 90~Degree Couplers Using Nonuniform Line Techniques.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1966. **MTT-14**(12).

30.   J. P. SHELTON and J.A. MOSKO, *Synthesis and Design of WideJ3and Equal~Ripple TEM Directional Couplers and Fixed Phase Shifters.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1966. **MTT-14**(10).

31.   W.KAMMLER, D., *The Design of Discrete N-Section and Continuously Tapered Symmetrical Microwave TEM Directional Couplers.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1969. **MTT-17**(8).

32.   Taconic, *Taconic TLY Technical Data Sheet.*

33.   DAVID W. KAMMLER , P.H.L., *Analog generation of 1/ t and 1/t.* ANALOG COMPUTER TECHNIQUES application notes

34.   GEORGE L. MILLICAN and ROBERT C. WALES, *Practical Strip-Line Microwave Circuit Design.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, 1969. **MTT-17**(9).

35.   COHN, S.B., *Characteristic Impedances of Broadside-Coupled Strip Transmission Lines*.* IRE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1960.

36.   J. PAUL SHELTON, J., *Impedances of Offset Parallel-Coupled Transmission Lines.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1966. **MTT-14**(1).

37.   COHN, S.B., *Thickness Corrections for Capacitive Obstacles and Strip Conductors.* IRE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1960.

38.   GUPTA, R.R., *Accurate Impedance Determination of Coupled TEM Conductors.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1969. **MTT-17**(8).

39.   DONALD H. SINNOTT, et al., *The Finite Difference Solution of Microwave Circuit Problems.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES 1969. **MTT-17**(8).

40.   C.M. Potter, P. and P. G. Hjipieris, *Improvements in ultra-broadband TEM coupler design.* IEE PROCEEDINGS-H, 1992. **139**(2).

41.   GREEN, H.E., *The Numerical Solution of Some Important Transmission-Line Problems.* lEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1965. **MTT-13**(5).

42.   Allen F. Horn III, et al., *Effect of conductor profile on the insertion loss, phase constant, and dispersion in thin high frequency transmission lines.* DesignCon, 2010.

43. Inc., A.I.M., *A Designed Experiment for the Influence of Copper Foils on Impedance, DC Line Resistance and Insertion Loss.*

44. Paul G. Huray, et al., *Impact of Copper Surface Texture on Loss: A Model that Works.* DesignCon, 2010.

45. Rogerscorp, *Copper Foils for High Frequency Materials.*

46. corp, I., *PCB-Material-Selection-for-High-speed-Digital-Designs.*

47. Morgan, S.P., *Effect of Surface Roughness on Eddy Current Losses at Microwave Frequencies.* Journal of Applied Physics, 1949. **20**(4): p. 352-362.

48. Taconic, *ProductSelectionGuide-High performance materials.*

49. ROBERT H. CAVERLY , G.H., *Establishing the Minimum Reverse Bias for a p-i-n Diode in a High-Power Switch.* IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 1990. **38**(12).

50. Inc, P., *A-Broadband-Microwave-Choke-piconics.* MICROWAVE JOURNAL, December 1999.

51. Modelithics, *Modelithics® COMPLETE Library for Keysight Advanced Design System (ADS).* Software product guide.

52. Piconics, *Conical-Inductor-Mounting-Instruction.* Application notes.

53. www.dilabs.com, D.l., *Milli-Cap® Ideal SMT Capacitor.* product guide.

54. Piconics, *http://www.piconics.com.* Company website.

55. Shah, J., *ESTIMATING BOND WIRE CURRENT-CARRYING CAPACITY.* WWW.POWERSYSTEMSDESIGN.COM, JULY/AUGUST 2012.

56. *al.,* O.H.*e.*, *Impact of different surface fishes on insertion loss for microstrip resonators up to 100 GHz.*

57. *al.,* o.H.*e.*, *Optimized Launching Pads fro Investigating Transmission Line losses for Different Surface Finishes of RF-PCBs up to 110 GHz.* IEEE International Microwave and RF Conference, 2015.

58. John Coonrod, B.R., *Comparing Microstrip and CPW Performance.* MICROWAVE JOURNAL, July 2012.

59. Qorvo, *QPM1000 Data Sheet.* Data sheet LNA/Limiter.

60.   BRORSSON, C., *PIN diode drive circuits optimized for fast switching.* Masters Thesis, 2011.

61.   Bill Rosas, P.E.M., Southwest Microwave, Inc.. www.southwestmicrowave.com, *Optimizing Test Boards for 50 GHz End Launch Connectors.* Application notes.

8. APPENDIX

8.1 Matlab code for quadrature coupler optimization and generations of physical dimension

```matlab
function [k_estimated_vector,Z_estimated_vector,TRACE_WIDTH_VECTOR
,TRACE_OFFSET_VECTOR  ] = COUPLER_DIMENSION_GENERATOR_TOP_LEVEL(  )
%% ======COUPLER_DIMENSION_GENERATOR_TOP_LEVEL============================%

% =======================================================================
% CLEARS_PREVIOUS_VARIABLES_AND_ADDS THE_FOLDER_PATH_TO_THE_FUNCTION
% =======================================================================
clc;
clearvars;
currentFolder = pwd;
addpath(genpath(currentFolder));

%_____LOCATION_OF_THE_PATH_TO_THE_FUNCTION_AND_ITS_SUBFUNCTIONS
%--------------------------------------------------------------------------

%% ======LOAD_PARAMETERS_GENERAL_AND_STANDARD===========================%

% =======================================================================
% CLEAR_OUT_IMAGE_FOLDER_STANDARD
% =======================================================================

warning('off','all')
rmdir('STANDARD\ITERATIVE_SPECTRUM_IMAGES_STANDARD', 's')%___DELETE_FIGURES
mkdir STANDARD\ITERATIVE_SPECTRUM_IMAGES_STANDARD%_____MAKE_FOLDER
warning('on','all')

% =======================================================================
% PHYSICAL_PARAMETERS
% =======================================================================

global E_R  V_PROP N_QW_LEN ;

E_R =2.2;%_____DIELECTRIC_CONSTANT_OF_SUBSTRATE
PROP_CONST =((sqrt(E_R))^-1);%_____VELOCITY_FACTOR_IN_THE_STRIPLINE
V_PROP = PROP_CONST*physconst('LightSpeed');%___VELOCITY_OF_PROPAGATION_M/S
N_QW_LEN = 13;%_____HALF_LENGTH_OF_COUPLER_IN QUARTER_WAVE_LENGTH
N_PARAM_EXTRA=0;%__NUMBER_OF_PARAMETERS_MORE_THAN_THE_QUARTER_WAVE_SECTIONS

% =======================================================================
% ELECTRICAL_PARAMETERS
% =======================================================================

global F_MIN  F_MAX  P_NOMINAL RIPPLE_OPTIM  F_CENTER;
```

```matlab
F_MIN
=0.01E9;%_____MINIMUM_FREQUENCY_OF_OPERATION_IN_HERTZ
F_MAX =26.77E9;%_____MAXIMUM_FREQUENCY_OF_OPERATION_IN_HERTZ
F_CENTER=(F_MIN + F_MAX)/2;%_____CENTER_FREQUENCY_IN_HERTZ
P_NOMINAL =-8.343;%_____DESIRED_COUPLING_AMPLITUDE_IN_dB
RIPPLE_OPTIM = 1E-6;%_____MAXIMUM_RIPPLE_TOLERANCE_IN_dB



% =======================================================================
% ANALYSIS_RESOLUTION_PARAMETERS_STANDARD_AND_COMMON
% =======================================================================

global N_SEC N_F  DELTA  N_PARAMS ;

N_SEC =200;%_____ELEMENTAL_ANALYSIS_SECTIONS_PER_QUARTER_WAVE_SECTION
N_F = 1000;%_____NUMBER_OF_FREQUENCY_POINTS_FOR_ANALYSIS
DELTA =1e-7;%_____INTINITESIMAL_VALUE_FOR_COMPUTING_THE_JACOBIAN_MATRIX
N_PARAMS =N_QW_LEN+N_PARAM_EXTRA;%____NUMBER_OF_PARAMETERS_FOR_OPTIMIZATION


% =======================================================================
% ITERATION_AND_PARAMETERS_VARIABLES_STANDARD
% =======================================================================

global ITER_MAX HALF_ITER_MAX PARAMETER_CURRENT ITER_COUNT
HALF_ITER_TOLERANCE   ;


ITER_MAX =0;%_____MAXIMUM_OPTIMIZATION_ITERATIONS
ITER_COUNT =0;%_____CURRENT_ITERATION_NUMBER
HALF_ITER_MAX = 1000;%___MAXIMUM_HALVING_ITERATIONS_IN_CONSTRAINTS_CORRECTION
HALF_ITER_TOLERANCE=1E-14;%_____MINIMUM_HALVING_TOLERANCE
PARAMETER_CURRENT =double(zeros([1 N_PARAMS]));%_____PARAMETER_REGISTER



% =======================================================================
% CONSTRAINT_PARAMETERS
% =======================================================================

global   STD_PARAMETER_MIN_BOUND  STD_PARAMETER_MAX_BOUND


STD_PARAMETER_MIN_BOUND= -10;%_____MINIMUM_BOUND_ON_STANDARD_PARAMETERS
STD_PARAMETER_MAX_BOUND= 1;%_____MAXIMUM_BOUND_ON_STANDARD_PARAMETERS


% =======================================================================
% CHECK_CONDITIONS_FLAGS
% =======================================================================

global INITIAL_ITERATION STD_OPTIMIZATION_COMPLETE    K_STD_OPTIMIZED

INITIAL_ITERATION ='TRUE';%___THIS_IS_REQUIRED_TO_SIGNAL_FOR_THE_NEXT_STAGE
STD_OPTIMIZATION_COMPLETE='NOT_COMPLETED';%_____STD_COMPLETION_CHECK
K_STD_OPTIMIZED ='NOT_COMPLETED';%_____REGISTER_FOR_STD_COUPLING_PROFILE
```

```matlab
% =====================================================================
% CALCULATION_OF_TOTAL_COUPLER_LENGTH
% =====================================================================

global COUPLER_LENGTH

COUPLER_LENGTH=200*N_QW_LEN*V_PROP/(4*F_CENTER);%TOTAL_COUPLER_LENGTH_IN_CM



%---------------------------------------------------------------------

%% ======GENERATE_STANDARD_PARAMETRIC_FORM===============================%

[k_std_form,~,P_BAR] =STANDARD_PARAMETRIC_FORM_GENERATOR( );

%---------------------------------------------------------------------

%% ======STANDARD_OPTIMIZATION_PROCEDURE===============================%

tic;
OPTIMIZER_STANDARD(k_std_form,P_BAR);
TIME_STD_OPTIMIZATION =toc;



%----------------RETURN_VALUE_TEST_FOR_STAGE_2----------------------------
%  k_std_form_substituted=subs(k_std_form,P_BAR,PARAMETER_CURRENT);
%  [~,~,~,~,~,~] =SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted);
%---------------------------------------------------------------------

%% ======LOAD_PARAMETERS_SPLINE_AND_DIMENSIONS_GENERAL===================%

% =====================================================================
% CLEAR_OUT_IMAGE_FOLDER_SPLINE
% =====================================================================

warning('off','all')
rmdir('SPLINE\ITERATIVE_SPECTRUM_IMAGES_SPLINE', 's')%_____DELETE_FIGURES
mkdir SPLINE\ITERATIVE_SPECTRUM_IMAGES_SPLINE%_____MAKE_FOLDER
warning('on','all')

% =====================================================================
% DIMENSIONS_FOR_K_MAX_DETERMINATION
% =====================================================================

global  K_MAX_DETERMINED   Z_O;



Z_O =50;%_____CHARACTERISTIC_IMPEDANCE_OF_TX_LINE

% =====================================================================
% SOLVER_AND_DIMENSIONAL_PARAMETERS
```

```matlab
% =========================================================================
global MATCHED_LOOK_UP

LOOK_UP_FILE_1=load('LOOK_UP_TABLE_MATCHED_(63.5mil)_0_200_2000.mat');%_LOAD_
50_OHM_MATCHED_LOOK_UP_TABLE
MATCHED_LOOK_UP=LOOK_UP_FILE_1.LOOK_UP_TABLE_MATCHED;
%-------------------------------------------------------------------------

K_MAX_DETERMINED=MATCHED_LOOK_UP.k(2);


% =========================================================================
% ANALYSIS_RESOLUTION_PARAMETERS_SPLINE
% =========================================================================

global   N_PARAMS_SPLINE K_MAX    ;


K_MAX
=K_MAX_DETERMINED;%_____MAXIMUM_COUPLING_COEFFICIENT_DETERMINES_CONDUCTOR_SP
ACING
N_PARAMS_SPLINE_MINIMUM=2*N_QW_LEN;%____MINIMUM_NUMBER_OF_PARAMETERS_SPLINE
EXTRA_PARAMETERS_SPLINE=1;%_____EXTRA_DEGREES_OF_FREEDOM_SPLINE
N_PARAMS_SPLINE=N_PARAMS_SPLINE_MINIMUM+N_QW_LEN*EXTRA_PARAMETERS_SPLINE;%TOT
AL_PARAMETERS_SPLINE

% =========================================================================
% ITERATION_AND_PARAMETERS_VARIABLES_SPLINE
% =========================================================================

global   ITER_COUNT_SPLINE   ITER_MAX_SPLINE   ;

ITER_MAX_SPLINE =9;%_____MAXIMUM_OPTIMIZATION_ITERATIONS_SPLINE
ITER_COUNT_SPLINE =0;%_____CURRENT_ITERATION_NUMBER_SPLINE


% =========================================================================
% CHECK_CONDITIONS_FLAGS
% =========================================================================

global   SPLINE_OPTIMIZATION_COMPLETE K_SPLINE_OPTIMIZED
INITIAL_ITERATION_SPLINE   INTERPOLANT_MODE

SPLINE_OPTIMIZATION_COMPLETE='NOT_COMPLETED';%_____SPLINE__COMPLETION_CHECK
K_SPLINE_OPTIMIZED ='NOT_COMPLETED';%__REGISTER_FOR_SPLINE_COUPLING_PROFILE
INITIAL_ITERATION_SPLINE ='TRUE';%_____FLAG_FOR_SPLINE_OPTIMIZER
INTERPOLANT_MODE
='FALSE';%__FLAG_FOR_SPLINE_OPTIMIZER_INTERPOLATION_TRUE_IF_SPLINE_IS_USED


% =========================================================================
% REGISTERS_FOR_SPLINE
% =========================================================================

global     SPLINE_INITIAL_PARAMS PARAMETER_CURRENT_SPLINE
```

```matlab
SPLINE_INITIAL_PARAMS =zeros([1
N_PARAMS_SPLINE]);%_____PARAMETER_REGISTER_SPLINE
PARAMETER_CURRENT_SPLINE=zeros([1
N_PARAMS_SPLINE]);%_____PARAMETER_REGISTER_SPLINE


%-------------------------------------------------------------------------

%% ======GENERATE_SPLINE_INITIAL_PARAMETERS============================%

 SPLINE_INITIAL_PARAMETER_GENERATOR( );


%-------------------------------------------------------------------------

%% ======SPLINE_OPTIMIZATION_PROCEDURE================================%

tic;
OPTIMIZER_SPLINE();
TIME_SPLINE_OPTIMIZATION =toc;


%-------------------------------------------------------------------------

%% ======OPTIMIZATION_TEST_CHECKS====================================%
% %---------------TEST_FOR_SPLINE_PROCEDURE------------------------------
%
%  [~,~,~,~,~,~] =SPECTRUM_ANALYZER_SPLINE( PARAMETER_CURRENT_SPLINE,1);
%  plot(K_SPLINE_TO_NUMERIC_CONVERTER(PARAMETER_CURRENT_SPLINE ));
%
% %---------------STANDARD_VS_SPLINE_COMPARISON_COUPLING_PROFILES-----------
%
%  plot(K_SPLINE_TO_NUMERIC_CONVERTER(PARAMETER_CURRENT_SPLINE ));hold
on;plot(K_STD_OPTIMIZED);
%
% %---------------STANDARD_VS_SPLINE_COMPARISON_FREQUENCY_SPECTRUM---------
% k_std_form_substituted=subs(k_std_form,P_BAR,PARAMETER_CURRENT);
% [~,~,~,~,~,~] =SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted);
%
% [~,~,~,~,~,~] =SPECTRUM_ANALYZER_SPLINE( PARAMETER_CURRENT_SPLINE,1);
% %-----------------------------------------------------------------

%% ======MANUFACTURING_AND_PHYSICAL_PARAMETERS=========================%


% =======================================================================
% PHYSICAL_DIMENSIONAL_PARAMETERS
% =======================================================================

K_PROFILE=K_SPLINE_OPTIMIZED;
% K_PROFILE=PARAMETER_CURRENT_SPLINE;

[k_estimated_vector,Z_estimated_vector,TRACE_WIDTH_VECTOR
,TRACE_OFFSET_VECTOR]=COUPLER_DIMENSION_LOOK_UP(K_PROFILE);


%-------------------------------------------------------------------------

%% ======LOOK_UP_TEST_CHECKS==========================================%
```

```matlab
% =========================================================================
% TEST_LOOK_UP
% =========================================================================

figure(1);
plot(sqrt((K_PROFILE-k_estimated_vector).^2));
figure(2);
plot(sqrt((Z_O-Z_estimated_vector).^2));
figure(3);
plot(TRACE_WIDTH_VECTOR);
figure(4);
plot(TRACE_OFFSET_VECTOR);


%-------------------------------------------------------------------------

%% ======COUPLER_CURVE_FIT========================================%

%-------------------------------------------------------------------------

end


function [TRACE_WIDTH_VECTOR
,TRACE_OFFSET_VECTOR]=COUPLER_CURVE_FIT(TRACE_WIDTH_VECTOR,TRACE_OFFSET_VECTO
R)
%2D_IMAGE_AND_CURVE_FIT_THE_GENERATED_DIMENSIONS



%-------------------------------------------------------------------------
%% ======COUPLER_IMAGE=========================================%

TRACE_WIDTH_CENTERED=TRACE_WIDTH_VECTOR-(TRACE_WIDTH_VECTOR/2);

%-------------------------------------------------------------------------

end



clc;
clearvars;
currentFolder = pwd;
addpath(genpath(currentFolder));
 global s b max_mesh_size TRACE_THICKNESS  solver_lengths
interp_mesh_resolution  E_R  change_step_width change_step_offset ;


s=(3.5);
b=(63.5);
TRACE_THICKNESS=(0.7);
max_mesh_size=b/30;
solver_lengths =3;
```

```matlab
interp_mesh_resolution =TRACE_THICKNESS/4;
E_R=2.2;




% ========================================================================
% WIDTH_RANGE_SPECIFICATION
% ========================================================================

Z_O=50;%_____TARGET_IMPEDANCE
IMPEDANCE_TOLERANCE=1;%_____IMPEDANCE_TOLERANCE
WIDTH_STEP=0.1;%_____STEP_SIZE


% ========================================================================
% OFFSET_RANGE_SPECIFICATION
% ========================================================================
change_offset_start=(0);
change_offset_stop=(250);

change_step_offset= linspace(change_offset_start,change_offset_stop,2000);



%------------------------------------------------------------------------
%% ======RUN_DIMENSIONAL_RUN_AND_UPDATE_LOOKUP_TABLE====================%
global LOOK_UP_TABLE


width_init=19;%_____INITIAL_TABLE_WIDTH
[ k_initial,Z_O_initial ] = LAPLACE_SOLVER(width_init ,0);


LOOK_UP_TABLE =
table(k_initial,Z_O_initial,width_init,0);%_____CREATE_AND_UPDATE_INITIAL_
VALUE
LOOK_UP_TABLE.Properties.VariableNames =
{'k','Z','width','Offset'};%_____LOOK_UP_TABLE_VARIABLE_NAMES


LOOK_UP_TABLE_MATCHED =
table(k_initial,Z_O_initial,width_init,0);%_____CREATE_AND_UPDATE_INITIAL_
VALUE
LOOK_UP_TABLE_MATCHED.Properties.VariableNames =
{'k','Z','width','Offset'};%_____LOOK_UP_TABLE_VARIABLE_NAMES


 len_w_o=length(change_step_offset);


for n=1:len_w_o47

    width_in=LOOK_UP_TABLE.width(end);
    while(1)
```

```matlab
    [ k_estimate,Z_O_estimate ] = LAPLACE_SOLVER(width_in
,change_step_offset(n))

        if((Z_O*IMPEDANCE_TOLERANCE/100) >(abs(Z_O_estimate-Z_O)))

APPEND_TABLE_MATCHED={k_estimate,Z_O_estimate,width_in,change_step_offset(n)}
;
            LOOK_UP_TABLE_MATCHED= [LOOK_UP_TABLE_MATCHED
;APPEND_TABLE_MATCHED];
            break;
        elseif(Z_O_estimate > Z_O)
            width_in=width_in+WIDTH_STEP;
        elseif(Z_O_estimate < Z_O)
            width_in=width_in-WIDTH_STEP;
        end
    end
end


%---------------------------------------------------------------------


function [LINE_WIDTH_STD ,LINE_OFFSET_STD] =DIMENSION_GENERATOR(K_NUMERICAL)
%THIS_FUNCTION_GENERATES_PHYSICAL_DIMENSIONS_FOR_A_GIVEN_SYMMETRIC_COUPLING_P
ROFILE

global s    E_R    Z_O

%---------------------------------------------------------------------

%% ======CALCULATION_OF_RHO_(EVEN_MODE_TO_ODD_MODE_IMPEDANCE)=============%

K_NUMERICAL=K_SPLINE_OPTIMIZED;%_____REMOVE_AFTER_TEST

% =========================================================================
% CALCULATE_RHO
% =========================================================================

syms k r;

eqn= k == ((sqrt(r))-1)/((sqrt(r))+1);

rho = solve(eqn,r);
rho =double(subs(rho,k,K_NUMERICAL));

% =========================================================================
% CALCULATION_OF_EVEN_AND_ODD_MODE_IMPEDANCE
% =========================================================================

ZO_e=sqrt(rho)*Z_O;%_____EVEN_MODE_IMPEDANCE_ALONG_COUPLER

ZO_o=Z_O./sqrt(rho);%_____ODD_MODE_IMPEDANCE_ALONG_COUPLER

%---------------------------------------------------------------------
%% ======DETERMINE_MAXIMUM_COUPLING================================%
```

```matlab
rho_max= max(rho);


% ========================================================================
% CHECK_VALIDITY_CONDITION_FOR_USE_OF_SHELDONS_TIGHT_COUPLING_FORMULA
% ========================================================================

condition_value = sqrt(rho_max*E_R);
condition_value =condition_value*Z_O*log(4);
condition_value =condition_value/(60*(pi^2));

if ((condition_value<0.5)&&(rho_max>2))
    condition_test=true;
else
    condition_test=false;
end

 end

function [K_MAX_APPROXIMATED] =K_MAX_APPROXIMATOR( )
%THIS_FUNCTION_GENERATES_PHYSICAL_DIMENSIONS_FOR_A_GIVEN_SYMMETRIC_COUPLING_P
ROFILE

global    E_R    s    b    Z_O

%--------------------------------------------------------------------------
%% ======DETERMINE_FRINGING_CAPACITANCE================================%

%
% ========================================================================
% DETERMINE_FRINGING_CAPACITANCE_FROM_APPROXIMATE_FORMULA
% ========================================================================


syms r   k w;
r_sub=s/b;%_____RATIO_OF_CONDUCTOR_TO_GROUND_SPACING
k_sub=w/b;%_____RATIO_OF_CONDUCTOR_WIDTH_TO_GROUND_SPACING



% ========================================================================
% DETERMINE_EVEN_FRINGING_CAPACITANCE_TO_ABSOLUTE_PERMITIVITY
% ========================================================================

EVEN_DEN = (r/(1-r))*log(1/r);
EVEN_DEN = EVEN_DEN +log(1/(1-r));
EVEN_DEN= 0.4413+((1/pi)*EVEN_DEN);


% ========================================================================
% DETERMINE_ODD_FRINGING_CAPACITANCE_TO_ABSOLUTE_PERMITIVITY
% ========================================================================
```

152

```matlab
ODD_DEN = (r/(1-r))*log(1/r);
ODD_DEN = ODD_DEN +log(1/(1-r));
ODD_DEN=(1/r)*(1/pi)*ODD_DEN;


%--------------------------------------------------------------------------
%% ======DETERMINE_EVEN_AND_ODD_MODE_IMPEDANCE===========================%


ZO_e_num = 188.3/sqrt(E_R);
ZO_e_den =( k/(1-r))+ EVEN_DEN;
ZO_e= ZO_e_num/ZO_e_den;


ZO_o_num = 188.3/sqrt(E_R);
ZO_o_den =( k/(1-r))+ (w/s)+ODD_DEN;
ZO_o= ZO_o_num/ZO_o_den;


eqn=Z_O==sqrt(ZO_e*ZO_o);
eqn=subs(eqn,k,(k_sub));
eqn=subs(eqn,r,(r_sub));


%--------------------------------------------------------------------------
%% ======DETERMINE_WIDTH_OF_CENTER_SECTION=============================%


width_center = vpasolve(eqn,w);


%--------------------------------------------------------------------------
%% ======DETERMINE_K_MAX_REQUIRED=====================================%


ZO_e=subs(ZO_e,k,(k_sub));
ZO_e=subs(ZO_e,r,(r_sub));

ZO_o=subs(ZO_o,k,(k_sub));
ZO_o=subs(ZO_o,r,(r_sub));



ZO_e=double(subs(ZO_e,w,width_center));
ZO_o=double(subs(ZO_o,w,width_center));


rho_max=ZO_e/ZO_o;


% =======================================================================
% CHECK_CONDITIONS
% =======================================================================



COND_1_VAL=double(width_center/s);
COND_2_VAL=double((width_center/b)/(1-(s/b)));



% =======================================================================
% CHECK_CONDITIONS_FOR_SOLUTION_VALIDITY_1
% =======================================================================
```

153

```matlab
if(COND_1_VAL >= 0.35)
    COND_1=true;
else
    COND_1=false;
end
% =========================================================================
% CHECK_CONDITIONS_FOR_SOLUTION_VALIDITY_2
% =========================================================================

if(COND_2_VAL >= 0.35)
    COND_2=true;
else
    COND_2=false;
end


% =========================================================================
% OUTPUT_SOLUTION_BASED_ON_CHECKS
% =========================================================================

if(COND_1 && COND_2)
%     K_MAX_APPROXIMATED=((rho_max)-1)/((rho_max)+1);%___USE_CURRENT_SOLUTION
    K_MAX_APPROXIMATED =(ZO_e-ZO_o)/(ZO_e+ZO_o);
else
    K_MAX_APPROXIMATED=K_MAX_EXACT;%_____USE_EXCACT_SOLUTIONS
end
%-------------------------------------------------------------------------
end


function [K_MAX_DETERMINED] =K_MAX_EXACT( )
%THIS_FUNCTION_GENERATES_PHYSICAL_DIMENSIONS_FOR_A_GIVEN_SYMMETRIC_COUPLING_P
ROFILE

global   E_R   s   b  Z_O

%-------------------------------------------------------------------------
%% =====DETERMINE_MAXIMUM_COUPLING=====================================%


% =========================================================================
% DETERMINE_RHO_MAX_FROM_COHN_BROADSIDE_FORMULAS
% =========================================================================

syms k ;
k_prime = sqrt(1-(k^2));
ZO_e = (188.3/sqrt(E_R)) * (ellipke(k_prime)/ellipke(k));
ZO_o = (296.1*s)/(sqrt(E_R)*b*atanh(k));


eqn=(Z_O^2)==ZO_e*ZO_o ;


k_factor_eval = (vpasolve(eqn,k,0.5));


ZO_e_max=double(subs(ZO_e,k,k_factor_eval));
```

```matlab
rho_max=(ZO_e_max/Z_O)^2;


% ========================================================================
% DETERMINE_K_MAX_TO_SET_THE_SYSTEM
% ========================================================================

syms c_profile  r;

c_profile = (((r))-1)/(((r))+1);

K_MAX_DETERMINED =(double(subs(c_profile,r,rho_max)));

%------------------------------------------------------------------------

end



function [Z_L_EVEN,Z_L_ODD] =LAPLACE_DIMENSION_SOLVER_ELECTRO(TRACE_WIDTH_IN
,TRACE_OFFSET_IN  )
%THIS_FUNCTION_ESTIMATES_THE_LOWER_BOUNDS_ON_EVEN_AND_ODD_MODE_IMPEDANCES_USI
NG_ELECTRO_STATIC_METHOD

global  s    b    max_mesh_size    TRACE_THICKNESS    solver_lengths
interp_mesh_resolution E_R;


TRACE_OFFSET = TRACE_OFFSET_IN;

TRACE_WIDTH =TRACE_WIDTH_IN;


WIDTH_OUTER= (TRACE_WIDTH+TRACE_OFFSET)/2;
solve_length= solver_lengths*b +2*WIDTH_OUTER;

%------------------------------------------------------------------------
%% ======CREATION_OF_2D_GEOMETRY_AND_MODEL===============================%

g=@COUPLER_GEOMETRY_FUNCTION_ELECTRO;
emmodel = createpde;
geometryFromEdges(emmodel,g);


% figure(1)
% pdegplot(emmodel,'EdgeLabels','on');
% title 'Block Geometry With Edge Labels Displayed';
%------------------------------------------------------------------------

%% ======SET_UP_THE_MESH================================================%


% ========================================================================
% GENERATE_MESH_TO_SOLVE_FOR
% ========================================================================
```

155

```matlab
generateMesh(emmodel,'Hmax',max_mesh_size);

% figure(2)
% pdeplot(emmodel);
% axis equal;
% title 'Block With Finite Element Mesh Displayed';


%--------------------------------------------------------------------------
%% ======SET_BOUNDARY_CONDITION_AND_SOLVE_FOR_ELECTROSTATIC_POTENTIAL=====%



specifyCoefficients(emmodel,'m',0,'d',0,'c',1,'a',0,'f',0);%__equation_coeffi
cients


% =========================================================================
% APPLY_BOUNDARY_CONDITION_FOR_EVEN_MODE
% =========================================================================
applyBoundaryCondition(emmodel,'dirichlet','Edge',([1 2 3
4]),'r',0,'h',1);%__Boundary_for_outer_conductor

applyBoundaryCondition(emmodel,'dirichlet','Edge',([5 6 7
8]),'r',1,'h',1);%__Boundary_for_upper_conductor


applyBoundaryCondition(emmodel,'dirichlet','Edge',([9 10 11
12]),'r',1,'h',1);%__Boundary_for_lower_conductor

results_even_electro = solvepde(emmodel);


% u_even_electro = results_even_electro.NodalSolution;
% figure(3);
% pdeplot(emmodel,'XYData',u_even_electro);
% title('EVEN MODE ELECTRO STATIC POTENTIAL');
% xlabel('x')
% ylabel('y')


% =========================================================================
% APPLY_BOUNDARY_CONDITION_FOR_ODD_MODE
% =========================================================================
applyBoundaryCondition(emmodel,'dirichlet','Edge',([1 2 3
4]),'r',0,'h',1);%__Boundary_for_outer_conductor

applyBoundaryCondition(emmodel,'dirichlet','Edge',([5 6 7
8]),'r',1,'h',1);%__Boundary_for_upper_conductor

applyBoundaryCondition(emmodel,'dirichlet','Edge',([9 10 11 12]),'r',-
1,'h',1);%__Boundary_for_lower_conductor

results_odd_electro = solvepde(emmodel);

% u_odd_electro = results_odd_electro.NodalSolution;
```

```matlab
% figure(4);
% pdeplot(emmodel,'XYData',u_odd_electro);
% title('ODD MODE ELECTRO STATIC POTENTIAL');
% xlabel('x');
% ylabel('y');


%-----------------------------------------------------------------------

%-----------------------------------------------------------------------
%%
=====QUERY_POINTS_MESH==========================================%%_____T
EMPORARY_CHECK


% =======================================================================
% SET_UP_THE_QUERY_MESH
% =======================================================================


xpoints=(2*solve_length)/interp_mesh_resolution;
ypoints=b/interp_mesh_resolution;

x_mesh = linspace(-solve_length,solve_length,xpoints);
y_mesh = linspace(-(b/2),(b/2),ypoints);

[X,Y] = meshgrid(x_mesh,y_mesh);
querypoints = [X(:),Y(:)]';


% =======================================================================
% SET_UP_THE_MASKING_MESH_FOR_ODD_MODE
% =======================================================================


% x_lower_start=ceil((solve_length-
0.5*TRACE_WIDTH+0.5*TRACE_OFFSET)/interp_mesh_resolution+0);
% y_lower_start=ceil(((b/2)-(s/2)-TRACE_THICKNESS)/interp_mesh_resolution+1);
%
% x_lower_end=x_lower_start+ceil((TRACE_WIDTH)/interp_mesh_resolution)-1;
% y_lower_end=y_lower_start+ceil(TRACE_THICKNESS/interp_mesh_resolution)-1;


%% ======ELECTROSTATIC_EVEN_INTEGRAL==================================%%
% =======================================================================
% NORMAL_MESH_SOLUTIONS
% =======================================================================
%
% figure(5);
% pdeplot(emmodel,'XYData',u_even_electro,'ZData',u_even_electro,'Mesh','on')
% xlabel('x')
% ylabel('y')


%
===========================================================================%REM
OVE_LATER
% INTERPOLATED_SOLUTION_FOR_POTENTIAL
% =======================================================================


u_even = interpolateSolution(results_even_electro,querypoints);
```

```matlab
% =========================================================================
% BACK_UP_SOLUTION
% =========================================================================
% u_even_back_up=u_even;
% u_even_back_up = reshape(u_even_back_up,size(X));


% =========================================================================
% APPLY_MASKING_AND_RESHAPE_ARRAY
% =========================================================================
u_even(isnan(u_even))=1;
u_even = reshape(u_even,size(Y));

% figure(6);
% mesh(X,Y,u_even)
% xlabel('x')
% ylabel('y')


% =========================================================================
% ENERGY_INTEGRAL_ELECTROSTATIC
% =========================================================================

M=length(y_mesh);
N=length(x_mesh);

u_even_append=horzcat(u_even,zeros(M,1));
u_even_append=vertcat(u_even_append,zeros(1,(N+1)));




  for y =1:M
      for x =1:N
          grad_even(y,x)=((u_even_append(y,x)-
u_even_append(y+1,x+1))).^2+((u_even_append(y+1,x)-u_even_append(y,x+1))).^2;
      end
  end

% mesh(X,Y,grad_even)
% xlabel('x')
% ylabel('y')

INTEGRAL_EVEN_ELECTRO=sum(sum(grad_even));

% =========================================================================
% EVEN_MODE_IMPEDANCE_LOWER_ESTIMATE
% =========================================================================
Z_L_EVEN=(480*pi)/(INTEGRAL_EVEN_ELECTRO* sqrt(E_R));
%-------------------------------------------------------------------------
%% ======ELECTROSTATIC_ODD_INTEGRAL===================================%%
% =========================================================================
% NORMAL_MESH_SOLUTIONS
% =========================================================================
%
% figure(7);
```

```matlab
% pdeplot(emmodel,'XYData',u_odd_electro,'ZData',u_odd_electro,'Mesh','on')
% xlabel('x')
% ylabel('y')


%
==============================================================================%REM
OVE_LATER
% INTERPOLATED_SOLUTION_FOR_POTENTIAL
% ==============================================================================


u_odd = interpolateSolution(results_odd_electro,querypoints);
% ==============================================================================
% BACK_UP_SOLUTION
% ==============================================================================
% u_odd_back_up=u_odd;
% u_odd_back_up = reshape(u_odd_back_up,size(X));


% ==============================================================================
% APPLY_MASKING_AND_RESHAPE_ARRAY
% ==============================================================================
u_odd = reshape(u_odd,size(Y));

[row_len,~]=size(u_odd);

if(mod(row_len,2)==0)
row_mid=row_len/2;
else
row_mid=ceil(row_len/2);
end


u_odd_negative=u_odd((1:row_mid),:);
u_odd_positive=u_odd(((row_mid+1):end),:);

u_odd_negative(isnan(u_odd_negative))=-1;
u_odd_positive(isnan(u_odd_positive))=+1;

u_odd=[u_odd_negative;u_odd_positive];


% figure(12);
% mesh(X,Y,u_odd)
% xlabel('x')
% ylabel('y')


% ==============================================================================
% ENERGY_INTEGRAL_ELECTROSTATIC
% ==============================================================================

% M=length(y_mesh);
% N=length(x_mesh);

[M,N]=size(u_odd);
```

```matlab
u_odd_append=horzcat(u_odd,zeros(M,1));
u_odd_append=vertcat(u_odd_append,zeros(1,(N+1)));



 for y =1:M
     for x =1:N
         grad_odd(y,x)=((u_odd_append(y,x)-
u_odd_append(y+1,x+1))).^2+((u_odd_append(y+1,x)-u_odd_append(y,x+1))).^2;
    end
 end



% mesh(X,Y,grad_odd)
% xlabel('x')
% ylabel('y')


INTEGRAL_ODD_ELECTRO=sum(sum(grad_odd));


% =========================================================================
% ODD_MODE_IMPEDANCE_LOWER_ESTIMATE
% =========================================================================
Z_L_ODD=(480*pi)/(INTEGRAL_ODD_ELECTRO* sqrt(E_R));



clearvars -except Z_L_ODD Z_L_EVEN;%_____CLEAR_SOLUTIONS
%-------------------------------------------------------------------------



%% ======COUPLER_GEOMETRY_FUNCTION=====================================%%
function [ x,y ] = COUPLER_GEOMETRY_FUNCTION_ELECTRO( bs,t )
%This_function_generates_the_2D_Geometry_for_the_hybrid_coupler


%-------------------------------------------------------------------------
%% ======PASS_ARGUMENTS_BASED_ON_QUERY=================================%

switch nargin

  case 0
% =========================================================================
% RETURN_THE_NUMBER_OF_EDGES
% =========================================================================
     x=12;  %_____ twelve_edge_segments_in_a_section
  return
%-------------------------------------------------------------------------
  case 1
% =========================================================================
% DEFINE_THE_2D_DIMENSION_OF_STRUCTURE
% =========================================================================
```

160

```matlab
      OUTER_COND = [(-b/2),(-solve_length),(b/2),(solve_length); % start
parameter values

                   (b/2),(solve_length),(-b/2),(-solve_length); % end
parameter values

                   0,0,0,0; % region label to left

                   1,1,1,1]; % region label to right

    UPPER_COND =     [(s/2),(-
TRACE_WIDTH/2),((s/2)+TRACE_THICKNESS),(TRACE_WIDTH/2); % start parameter
values

                   ((s/2)+TRACE_THICKNESS),(TRACE_WIDTH/2),(s/2),(-
TRACE_WIDTH/2); % end parameter values

                   1,1,1,1; % region label to left

                   0,0,0,0]; % region label to right

    LOWER_COND =     [ (-(s/2)-TRACE_THICKNESS),(-TRACE_WIDTH/2),(-
s/2),(TRACE_WIDTH/2); % start parameter values

                   (-s/2),(TRACE_WIDTH/2),((-s/2)-TRACE_THICKNESS),(-
TRACE_WIDTH/2); % end parameter values

                   1,1,1,1; % region label to left

                   0,0,0,0]; % region label to right

    STRUCTURE=[ OUTER_COND,UPPER_COND, LOWER_COND ];

   x = STRUCTURE(:,bs);
   return
%-------------------------------------------------------------------------
 case 2
% ========================================================================
% RETURN_X_Y_COORDINATES
% ========================================================================

 x = zeros(size(t));
 y = zeros(size(t));
 if numel(bs) == 1 % Does bs need scalar expansion?
     bs = bs*ones(size(t)); % Expand bs
 end




% ========================================================================
% OUTER_CONDUCTOR_SPECIFICATION
```

161

```matlab
% =========================================================================

 cbs = find (bs == 1); %_____EDGE_1
 y(cbs) = t(cbs);
 x(cbs) =(-solve_length)*ones(size(cbs));

cbs = find(bs ==2); %_____EDGE_2
 x(cbs) = (t(cbs));
 y(cbs) = (+b/2)*ones(size(cbs));

cbs = find(bs ==3); %_____EDGE_3
 y(cbs) =(t(cbs));
 x(cbs) = (solve_length)*ones(size(cbs));

cbs = find(bs ==4); %_____EDGE_4
 x(cbs) =(t(cbs));
 y(cbs) = (-b/2)*ones(size(cbs));


%-------------------------------------------------------------------------



% =========================================================================
% UPPER_CONDUCTOR_SPECIFICATION
% =========================================================================

 cbs = find (bs == 5); %_____EDGE_5
 y(cbs) = t(cbs);
 x(cbs) =(-(TRACE_WIDTH+TRACE_OFFSET)/2)*ones(size(cbs));

cbs = find(bs ==6); %_____EDGE_6
 x(cbs) = (t(cbs)-(TRACE_OFFSET/2));
 y(cbs) = ((s/2)+TRACE_THICKNESS)*ones(size(cbs));

cbs = find(bs ==7); %_____EDGE_7
 y(cbs) =(t(cbs));
 x(cbs) = ((TRACE_WIDTH-TRACE_OFFSET)/2)*ones(size(cbs));

cbs = find(bs ==8); %_____EDGE_8
 x(cbs) =(t(cbs)-(TRACE_OFFSET/2));
 y(cbs) = (s/2)*ones(size(cbs));


%-------------------------------------------------------------------------




% =========================================================================
% LOWER_CONDUCTOR_SPECIFICATION
% =========================================================================

cbs = find (bs == 9); %_____EDGE_9
 y(cbs) = t(cbs);
 x(cbs) =((TRACE_OFFSET-TRACE_WIDTH)/2)*ones(size(cbs));
```

```matlab
cbs = find(bs == 10); %_____EDGE_10
 x(cbs) = (t(cbs)+(TRACE_OFFSET/2));
 y(cbs) = (-s/2)*ones(size(cbs));

cbs = find(bs == 11); %_____EDGE_11
 y(cbs) =(t(cbs));
 x(cbs) = ((TRACE_WIDTH+TRACE_OFFSET)/2)*ones(size(cbs));

cbs = find(bs == 12); %_____EDGE_12
 x(cbs) =(t(cbs)+(TRACE_OFFSET/2));
 y(cbs) = (-s/2-TRACE_THICKNESS)*ones(size(cbs));
%-----------------------------------------------------------------------

  end

end



end

  function [ k_estimate,Z_O_estimate ] = LAPLACE_SOLVER(TRACE_WIDTH
,TRACE_OFFSET)

%_ESTIMATES_THE_IMPEDANCE_AND_COUPLING_COEFFICIENT
global LOOK_UP_TABLE

%% ======IMPEDANCE_AND_COUPLING_ESTIMATOR=============================%

[Z_EVEN,Z_ODD] =LAPLACE_DIMENSION_SOLVER_ELECTRO(TRACE_WIDTH ,TRACE_OFFSET);


Z_O_estimate=sqrt(Z_EVEN*Z_ODD);%_____ESTIMATE_OF_THE_IMPEDANCE
k_estimate=(Z_EVEN-Z_ODD)/(Z_EVEN+Z_ODD);%___ESTIMATE_OF_THE_COUPLING_COEFF


% =======================================================================
% UPDATE_LOOKUP_TABLE_EVERYTIME_CALCULATED
% =======================================================================


APPEND_TABLE={k_estimate,Z_O_estimate,TRACE_WIDTH,TRACE_OFFSET};

LOOK_UP_TABLE= [LOOK_UP_TABLE ;APPEND_TABLE];


%-----------------------------------------------------------------------

end
```

```matlab
function [k_estimate_vector,Z_estimate_vector, TRACE_WIDTH_VECTOR
,TRACE_OFFSET_VECTOR ] = COUPLER_DIMENSION_LOOK_UP( k_profile )
%THIS_FUNCTION_GENERATES_THE_COUPLER_DIMENSIONS

global   MATCHED_LOOK_UP
%  k_profile=K_STD_OPTIMIZED;
%-----------------------------------------------------------------------
%% ======FIND_BEST_FITTING_DIMENSIONS==================================%

len_k=length(k_profile);

for n=1:len_k

[~,find_index(n)]=(min(abs((MATCHED_LOOK_UP.k)-k_profile(n))));
 k_estimate_vector(n)=MATCHED_LOOK_UP.k(find_index(n));
 Z_estimate_vector(n)=MATCHED_LOOK_UP.Z(find_index(n));
 TRACE_WIDTH_VECTOR(n)=MATCHED_LOOK_UP.width(find_index(n));
 TRACE_OFFSET_VECTOR(n)=MATCHED_LOOK_UP.Offset(find_index(n));
end


%-----------------------------------------------------------------------


%% ======GET_THE_2-D_IMAGE_FORM=======================================%

% ======================================================================
% CLEAR_OUT_IMAGE_FOLDER_STANDARD
% ======================================================================

TRACE_INNER=(TRACE_OFFSET_VECTOR/1)-(TRACE_WIDTH_VECTOR(1)/1);

TRACE_OUTER= TRACE_INNER+TRACE_WIDTH_VECTOR;

figure(6);
plot(TRACE_INNER);
hold on;
plot(TRACE_OUTER);
y = [0 0];
x = [(0) (length(TRACE_INNER))];
line(x,y,'Color','red','LineStyle','--')
hold off;


%-----------------------------------------------------------------------


end

function [   ] = EXCATLY_DETERMINED_OPTIMIZER_SPLINE( )
%THIS_FUNCTION_PERFORMS_EXACTLY_DETERMINED_MODE_OF_OPTIMIZATION

global DELTA     HALF_ITER_MAX      ITER_COUNT_SPLINE
PARAMETER_CURRENT_SPLINE     N_PARAMS_SPLINE;
global K_MAX       HALF_ITER_TOLERANCE


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

```matlab
%% ======CALCULATION_OF_THE_JACOBIAN_MATRIX_D============================%

% =========================================================================
% CREATE_DELTA_PARAMETER_SPACING_MATRIX
% =========================================================================

clearvars EXTREMUM_NEW EXTREMUM_CURRENT;%_CLEARING_VARIABLES_FOR_SIMPLICITY

DELTA_parameter= double(DELTA.*ones(1,N_PARAMS_SPLINE));

% =========================================================================
% CALCULATE_THE_INITIAL_EXTREMUM_ERROR
% =========================================================================

[EXTREMUM_CURRENT,~,SIGMA,~,~,~] =SPECTRUM_ANALYZER_SPLINE(
PARAMETER_CURRENT_SPLINE,0);
EXTREMUM_CURRENT=transpose(EXTREMUM_CURRENT);
EXTREMUM_CURRENT = repmat(EXTREMUM_CURRENT,1,N_PARAMS_SPLINE);

% =========================================================================
% CALCULATE_THE_DELTA_EXTREMUM_ERROR_FROM_CURRENT_PARAMETER_POINTS
% =========================================================================

INCREMENTED_PARAMETER=PARAMETER_CURRENT_SPLINE   ;
INCREMENTED_PARAMETER = repmat(INCREMENTED_PARAMETER,N_PARAMS_SPLINE,1);

for j =1:(N_PARAMS_SPLINE)%_____MATRIX_FOR_INCREMENTED_PARAMETERS
    INCREMENTED_PARAMETER(j,j)=
INCREMENTED_PARAMETER(j,j)+DELTA_parameter(1,j);
end

% =========================================================================
% CALCULATION_OF_DIFFERENTIAL_ERROR_CURVE
% =========================================================================

for j =1:(N_PARAMS_SPLINE)%_____ERROR_CHANGE_CALCULATIONS_FOR_ALL_EXTREMUMS

 DELTA_PARAMETER_CHANGE= INCREMENTED_PARAMETER(j,:);
 [EXTREMUM_NEW(j,:),~,~,~,~,~] =SPECTRUM_ANALYZER_SPLINE(
DELTA_PARAMETER_CHANGE,0);

end

EXTREMUM_NEW=transpose(EXTREMUM_NEW);

% =========================================================================
% CALCULATE_THE_JACOBIAN
% =========================================================================
```

165

```matlab
D=EXTREMUM_NEW-EXTREMUM_CURRENT;%_____CALCULATION_OF_DIFFERENTIAL_ERROR
D=D./DELTA;




%-------------------------------------------------------------------------
%% ======REFORMATTING_THE_MATRIX_FOR_LINEAR_SYSTEM=======================%


% ========================================================================
% FORMING_OF_THE_KNOWN_OF_LEFT_HAND_SIDE
% ========================================================================

SIGMA_APPEND= transpose(-SIGMA);
SIGMA_APPEND=SIGMA_APPEND((1:(N_PARAMS_SPLINE+1)),:);
LHS_KNOWN=horzcat(D,SIGMA_APPEND);


% ========================================================================
% FORMING_OF_THE_KNOWN_OF_RIGHT_HAND_SIDE
% ========================================================================

RHS_KNOWN=(-
EXTREMUM_CURRENT((1:(N_PARAMS_SPLINE+1)),1));%____RIGHT_HAND_SIDE_OF_LINEAR_S
YSTEM_TO_SOLVE



%-------------------------------------------------------------------------
%% ======SOLVING_LINEAR_SYSTEM_FOR_CHANGE_IN_PARAMETERS==================%


PARAMETER_CURRENT_BACKUP_SPLINE=PARAMETER_CURRENT_SPLINE;%_____PARA
METER_BACKUP


% ========================================================================
% DETERMINE_CHANGE_IN_PARAMETERS_USING_LINEAR_SYSTEM
% ========================================================================



del_p = (pinv(LHS_KNOWN))*RHS_KNOWN;
del_p=del_p(1:end-1);%_____REMOVE_DUMMY_VALUE
del_p = transpose(del_p);


% ========================================================================
% UPDATE_THE_CURRENT_PARAMETERS
% ========================================================================


for i=1:N_PARAMS_SPLINE%_____PARAMETER_UPDATE_LOOP
PARAMETER_CURRENT_SPLINE(i)=PARAMETER_CURRENT_SPLINE(i)+del_p(i);
end


%-------------------------------------------------------------------------

%% ======DECISION_FLOW_FOR_CONSTRAINT_CHECK===========================%


% ========================================================================
```

```matlab
% DETECTING_IF_PARAMETERS_EXCEED_K_MAX_CONSTRAINT
% =======================================================================


MAX_OF_PARAMETERS =max(PARAMETER_CURRENT_SPLINE);

if((MAX_OF_PARAMETERS)<(K_MAX))
   %-------------DO-NOTHING----------------------------------------------%

  else
   % =======================================================================
   % DECREMENT_AND_CHECK_FOR_ERROR
   % =======================================================================
     HALVING_COUNT=HALF_ITER_MAX ;%_____REGISTER_FOR_HALVING_COUNT
     PARAMETER_CURRENT_SPLINE=
PARAMETER_CURRENT_BACKUP_SPLINE;%__RELOAD_PREVIOUS_PARAMETERS
           % =============================================================
           % DECREMENT_AND_CHECK_FOR_ERROR
           % =============================================================
           while (HALVING_COUNT ~= 0)%_____EXIT_IF_MAX_ITERATIONS_DONE
                    HALVING_COUNT=HALVING_COUNT-1 ;%_____DECREMENT_REGISTER
                    del_p=(1/2)*del_p;
                    TEMP_PARAMETERS=PARAMETER_CURRENT_SPLINE+del_p;

                    [~,~,~,~,RIPPLE_MAX_DEFAULT,~] =SPECTRUM_ANALYZER_SPLINE(
PARAMETER_CURRENT_SPLINE,0);
                    [~,~,~,~,RIPPLE_MAX_HALVED,~]  =SPECTRUM_ANALYZER_SPLINE(
TEMP_PARAMETERS,0);
                              if((RIPPLE_MAX_HALVED+HALF_ITER_TOLERANCE) <=
RIPPLE_MAX_DEFAULT)%_____EXIT_WHILE_LOOP_IF_ERROR_IMPROVED

PARAMETER_CURRENT_SPLINE=PARAMETER_CURRENT_SPLINE+del_p;%_____UPDATE_PARA
METER_IF_ERROR_IMPROVED
                                        break
                                  end

OPTIMIZER_SPLINE();%_____CALL_OPTIMIZER_TO_FURTHER_REFINE
         end
end

%----------------------------------------------------------------------


%% ======CALL_OPTIMIZER_FOR_REFINING===================================%

ITER_COUNT_SPLINE
=ITER_COUNT_SPLINE+1;%_____INCREMENT_ITERATION_COUNT
OPTIMIZER_SPLINE();%_____CALL_STANDARD_OPTIMIZER


%----------------------------------------------------------------------


end
```

```matlab
function [ K_INTERPOLATED_NUMERICAL_SPLINE ] = K_SPLINE_TO_NUMERIC_CONVERTER( PARAMETER_SPLINE )
%THIS_FUNCTION_COMPUTES_POWER_COUPLING_RESPONSE_FOR_A_GIVEN_K(U)_PROFILE

global   N_SEC  N_QW_LEN   K_MAX
global  N_PARAMS_SPLINE INTERPOLANT_MODE

%% =====SPLINE_INTERPOLATION_OF_PARAMETERS=============================%

PARAMETER_CURRENT_SPLINE=PARAMETER_SPLINE ;
% =========================================================================
% CREATE_INDEX_FOR_QUERY_POINTS_TO_EVALUATE_SPLINE
% =========================================================================


u_index = double(ones(N_QW_LEN,N_SEC));
for n =1:(N_QW_LEN)
u_index(n,:) = (linspace((n-1),n,(N_SEC)));
end


% =========================================================================
% CONCATENATE_U_INDEXES_FOR_QUERY_POINTS
% =========================================================================


U_INDEX = u_index(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_QW_LEN-1)
U_INDEX_TEMP= u_index((n),(1:(N_SEC-1)));
U_INDEX = horzcat(U_INDEX,U_INDEX_TEMP);
end
U_INDEX_RIGHT = horzcat(U_INDEX,u_index(N_QW_LEN,(1:(N_SEC))));%___END_INDEX

% =========================================================================
% GET_MIRRORED_INDEX
% =========================================================================

U_INDEX_MIRRORED=-U_INDEX_RIGHT;
U_INDEX_LEFT = fliplr(U_INDEX_MIRRORED);

U_INDEX_COMPLETE =horzcat(U_INDEX_LEFT(1:end-1),U_INDEX_RIGHT);%___THE_COMPLETE_INDEX

% =========================================================================
% CONCATENATE_U_INDEXES_AND_KNOWN_PARAMETERS_FOR_QUERY_POINTS
% =========================================================================

PARAMETER_SPLINE_RIGHT=[K_MAX,PARAMETER_CURRENT_SPLINE,0];
PARAMETER_SPLINE_LEFT=fliplr(PARAMETER_SPLINE_RIGHT);
SYMMETRIC_PARAMETER_SPLINE=horzcat(PARAMETER_SPLINE_LEFT(1:end-1),PARAMETER_SPLINE_RIGHT);%___THE_COMPLETE_INDEX

% =========================================================================
% GET_MIRRORED_KNOWN_PARAMETER_INDEX
% =========================================================================
```

168

```matlab
PARAM_INDEX_RIGHT_ALL = linspace(0,N_QW_LEN,(N_PARAMS_SPLINE+2));
PARAM_INDEX_RIGHT=PARAM_INDEX_RIGHT_ALL;
PARAM_INDEX_LEFT  = -fliplr(PARAM_INDEX_RIGHT);
KNOWN_PARAMETER_INDEX=horzcat(PARAM_INDEX_LEFT(1:end-
1),PARAM_INDEX_RIGHT);%___THE_COMPLETE_INDEX


if
((strcmp(INTERPOLANT_MODE,'TRUE')))%_PROCEED_WITH_SPLINE_INTERPOLATION_IF_TRU
E
    K_INTERPOLATED_FULL =
spline(KNOWN_PARAMETER_INDEX,SYMMETRIC_PARAMETER_SPLINE,U_INDEX_COMPLETE);
else
    K_INTERPOLATED_FULL =
pchip(KNOWN_PARAMETER_INDEX,SYMMETRIC_PARAMETER_SPLINE,U_INDEX_COMPLETE);
end


half=ceil(length(K_INTERPOLATED_FULL)/2);
K_INTERPOLATED_NUMERICAL_SPLINE=K_INTERPOLATED_FULL(half:end);

%------------------------------------------------------------------------

end


function [     ] = OPTIMIZER_SPLINE( )
%THIS_FUNCTION_GENERATES_OPTIMIZED_COUPLING_PROFILE_FOR_A_STANDARD_PARAMETRIC
_FORM
tic;
global N_PARAMS_SPLINE  RIPPLE_OPTIM     ITER_MAX_SPLINE     ITER_COUNT_SPLINE
N_QW_LEN
global PARAMETER_CURRENT_SPLINE     SPLINE_INITIAL_PARAMS
INITIAL_ITERATION_SPLINE     SPLINE_OPTIMIZATION_COMPLETE
K_SPLINE_OPTIMIZED    K_SPLINE_EXTREMUM

%% ======PARAMETER_SUBSTITUTION=========================================%

% =======================================================================
% SUBSTITUTION_OF_PARAMETERS_IN_SYMBOLIC_FUNCTION
% =======================================================================

if(strcmp(INITIAL_ITERATION_SPLINE,'TRUE'))%___EVALUATE_WITH_INITIAL_PARAMETE
RS_IF_INITIAL_CONDITION_TRUE
        PARAMETER_CURRENT_SPLINE=SPLINE_INITIAL_PARAMS;
        INITIAL_ITERATION_SPLINE='FALSE';%___SET_INITIAL_ITERATION_TO_FALSE
end

%------------------------------------------------------------------------

%% ======SPECTRUM_ANALYZER=============================================%

% =======================================================================
```

```matlab
% EVALUATE_POWER_COUPLING_RESPONSE_OF_EVALUATED_K(U)_NUMERICAL
% =======================================================================


[EXTREMUM,~,~,~,RIPPLE_MAX,~]
=SPECTRUM_ANALYZER_SPLINE(PARAMETER_CURRENT_SPLINE,1 );

m=length(EXTREMUM);%_____NUMBER_OF_EXTREMUMS


% =======================================================================
% SAVE_SPECTRUMS_IN_CURRENT_FOLDER
% =======================================================================

fig = gcf;
savefig(fig,['SPLINE/ITERATIVE_SPECTRUM_IMAGES_SPLINE/ITERATION__',num2str(IT
ER_COUNT_SPLINE),'_OUT_OF_',num2str(ITER_MAX_SPLINE),'_HALF_QW_LENGTH_',num2s
tr(N_QW_LEN),'.fig'],'compact');
%__SAVE_THIS_FIGURE



%-----------------------------------------------------------------------
%% ======OPTIMIZATION_DECISION=========================================%

if(ITER_COUNT_SPLINE >= ITER_MAX_SPLINE
)%___EXIT_OPTIMIZATION_IF_MAXIMUM_ITERATIONS_DONE
  %-------------DO-NOTHING---------------------------------------------%
  SPLINE_OPTIMIZATION_COMPLETE='SPLINE_OPTIMIZATION_MAXED_ITERATION';
elseif (RIPPLE_MAX <= RIPPLE_OPTIM
)%END_OPTIMIZATION_IF_ERROR_BELOW_TOLERANCE
  %-------------DO-NOTHING---------------------------------------------%
  SPLINE_OPTIMIZATION_COMPLETE='SPLINE_OPTIMIZATION_MET_CONDITIONS';
elseif (RIPPLE_MAX >= RIPPLE_OPTIM)%_____NOT_OPTIMIZED


% =======================================================================
% DECIDE_THE_MODE_OF_OPTIMIZATION
% =======================================================================


        if (m ==
(N_PARAMS_SPLINE+1))%_____EXCATLY_DETERMINED_OPTIMIZATION
EXCATLY_DETERMINED_OPTIMIZER_SPLINE( ) ;
        else%_____RECTANGULAR_OPTIMIZATION
RECTANGULAR_OPTIMIZER_SPLINE( ) ;
        end
end


%-----------------------------------------------------------------------
%% ======RETURNS_THE_SPLINE_PROFILE_IF_OPTIMIZATION_COMPLETED=============%

 close all;%_____CLOSES_ALL_OPEN_FIGURES

 [~,~,~,~,~,~] =SPECTRUM_ANALYZER_SPLINE( PARAMETER_CURRENT_SPLINE,1);
 K_SPLINE_OPTIMIZED  = K_SPLINE_TO_NUMERIC_CONVERTER(PARAMETER_CURRENT_SPLINE
);
 K_SPLINE_EXTREMUM =EXTREMUM;
```

170

```matlab
%--------------------------------------------------------------------------
end


function [ ] = RECTANGULAR_OPTIMIZER_SPLINE(   )
%THIS_FUNCTION_PERFORMS_RECTANGULAR_MODE_OF_OPTIMIZATIONS

global DELTA      HALF_ITER_MAX      ITER_COUNT_SPLINE
PARAMETER_CURRENT_SPLINE     N_PARAMS_SPLINE;
global K_MAX        HALF_ITER_TOLERANCE


%--------------------------------------------------------------------------
%% ======CALCULATION_OF_THE_JACOBIAN_MATRIX_D==========================%
% ========================================================================
% CREATE_DELTA_PARAMETER_SPACING_MATRIX
% ========================================================================

clearvars EXTREMUM_NEW EXTREMUM_CURRENT;%_CLEARING_VARIABLES_FOR_SIMPLICITY

DELTA_parameter= double(DELTA.*ones(1,N_PARAMS_SPLINE));




% ========================================================================
% CALCULATE_THE_INITIAL_EXTREMUM_ERROR
% ========================================================================

[EXTREMUM_CURRENT_INITIAL,~,SIGMA,D_AVG,~,~] =SPECTRUM_ANALYZER_SPLINE(
PARAMETER_CURRENT_SPLINE,0);
EXTREMUM_CURRENT=transpose(EXTREMUM_CURRENT_INITIAL);
EXTREMUM_CURRENT = repmat(EXTREMUM_CURRENT,1,N_PARAMS_SPLINE);




% ========================================================================
% CALCULATE_THE_DELTA_EXTREMUM_ERROR_FROM_CURRENT_PARAMETER_POINTS
% ========================================================================

INCREMENTED_PARAMETER=PARAMETER_CURRENT_SPLINE   ;
INCREMENTED_PARAMETER = repmat(INCREMENTED_PARAMETER,N_PARAMS_SPLINE,1);

for j =1:(N_PARAMS_SPLINE)%_____MATRIX_FOR_INCREMENTED_PARAMETERS
    INCREMENTED_PARAMETER(j,j)=
INCREMENTED_PARAMETER(j,j)+DELTA_parameter(1,j);
end

% ========================================================================
% CALCULATION_OF_DIFFERENTIAL_ERROR_CURVE
% ========================================================================



for j =1:(N_PARAMS_SPLINE)%_____ERROR_CHANGE_CALCULATIONS_FOR_ALL_EXTREMUMS
```

171

```
    DELTA_PARAMETER_CHANGE= INCREMENTED_PARAMETER(j,:);
     [EXTREMUM_NEW(j,:),~,~,~,~,~] =SPECTRUM_ANALYZER_SPLINE(
DELTA_PARAMETER_CHANGE,0);

end

EXTREMUM_NEW=transpose(EXTREMUM_NEW);


% ========================================================================
% CALCULATE_THE_JACOBIAN
% ========================================================================

D=EXTREMUM_NEW-EXTREMUM_CURRENT;%_____CALCULATION_OF_DIFFERENTIAL_ERROR
D=D./DELTA;


%------------------------------------------------------------------------
%% ======CALCULATION_OF_DESIRED_ERROR_CHANGE===========================%

DESIRED_ERROR_CHANGE= (1/2)*SIGMA*D_AVG -EXTREMUM_CURRENT_INITIAL;
DESIRED_ERROR_CHANGE=transpose(DESIRED_ERROR_CHANGE);


%------------------------------------------------------------------------

%%
======CALCULATION_OF_PSEUDO_PENROSE_INVERSE_FOR_PARAMETER_CHANGE======%%CHEC
K_LATER_FOR_WHICH_METHOD_TO_USE_FOR_PSEUDO_INVERSE

PARAMETER_CURRENT_BACKUP_SPLINE=PARAMETER_CURRENT_SPLINE;%_____PARA
METER_BACKUP


% ========================================================================
% CALCULATE_THE_PENROSE_PSEUDO_INVERSE
% ========================================================================

del_p = pinv(D)*DESIRED_ERROR_CHANGE ;

del_p = transpose(del_p);


% ========================================================================
% CALCULATE_THE_LINEAR_CONSTRAINED_INVERSE
% ========================================================================

%  options = optimoptions('lsqlin','Algorithm','trust-region-reflective');
%
% lb=+6-(double(PARAMETER_CURRENT_SPLINE));
%
% ub=double((ones(length(PARAMETER_CURRENT_SPLINE),1))-
(transpose(PARAMETER_CURRENT_SPLINE)));

% del_p = lsqnonneg(D,DESIRED_ERROR_CHANGE);
```

```matlab
% del_p = lsqlin(D,DESIRED_ERROR_CHANGE,[],[],[],[],[],[]);

% del_p = lsqlin(D,DESIRED_ERROR_CHANGE,[],[],[],[],lb,ub,[]);

% del_p = transpose(del_p);


% ========================================================================
% UPDATE_THE_CURRENT_PARAMETERS
% ========================================================================


for i=1:N_PARAMS_SPLINE%_____PARAMETER_UPDATE_LOOP
PARAMETER_CURRENT_SPLINE(i)=PARAMETER_CURRENT_SPLINE(i)+del_p(i);
end


%-------------------------------------------------------------------------

%% ======DECISION_FLOW_FOR_CONSTRAINT_CHECK===========================%


% ========================================================================
% DETECTING_IF_PARAMETERS_EXCEED_K_MAX_CONSTRAINT
% ========================================================================


MAX_OF_PARAMETERS =max(PARAMETER_CURRENT_SPLINE);

if((MAX_OF_PARAMETERS)<=(K_MAX))
    %-------------DO-NOTHING---------------------------------------------%

    else
    % ====================================================================
    % DECREMENT_AND_CHECK_FOR_ERROR
    % ====================================================================
        HALVING_COUNT=HALF_ITER_MAX ;%_____REGISTER_FOR_HALVING_COUNT
        PARAMETER_CURRENT_SPLINE=
PARAMETER_CURRENT_BACKUP_SPLINE;%__RELOAD_PREVIOUS_PARAMETERS
            % ==========================================================
            % DECREMENT_AND_CHECK_FOR_ERROR
            % ==========================================================
            while (HALVING_COUNT ~= 0)%_____EXIT_IF_MAX_ITERATIONS_DONE
                    HALVING_COUNT=HALVING_COUNT-1 ;%_____DECREMENT_REGISTER
                    del_p=(1/2)*del_p;
                    TEMP_PARAMETERS=PARAMETER_CURRENT_SPLINE+del_p;

                    [~,~,~,~,RIPPLE_MAX_DEFAULT,~] =SPECTRUM_ANALYZER_SPLINE(
PARAMETER_CURRENT_SPLINE,0);
                    [~,~,~,~,RIPPLE_MAX_HALVED,~]  =SPECTRUM_ANALYZER_SPLINE(
TEMP_PARAMETERS,0);
                            if((RIPPLE_MAX_HALVED+HALF_ITER_TOLERANCE) <=
RIPPLE_MAX_DEFAULT)%_____EXIT_WHILE_LOOP_IF_ERROR_IMPROVED

PARAMETER_CURRENT_SPLINE=PARAMETER_CURRENT_SPLINE+del_p;%_____UPDATE_PARA
METER_IF_ERROR_IMPROVED
                                        break
                            end
```

173

```matlab
        OPTIMIZER_SPLINE_JUMPER();%_____CALL_OPTIMIZER_TO_FURTHER_REFINE
            end
end


%-------------------------------------------------------------------------

%% ======CALL_OPTIMIZER_FOR_REFINING===================================%

ITER_COUNT_SPLINE
=ITER_COUNT_SPLINE+1;%_____INCREMENT_ITERATION_COUNT
OPTIMIZER_SPLINE();%_____CALL_SPLINE_OPTIMIZER


%-------------------------------------------------------------------------

end


function [EXTREMUM,OMEGA,SIGMA,D_AVG,RIPPLE_MAX,OMEGA_INDEX] =
SPECTRUM_ANALYZER_SPLINE(PARAMETER_SPLINE
,PLOT_THE_FIGURE_SPLINE)%_____CHECK_FOR_ERRORS_IN_FIND_PEAKS_SOMETIMES
%THIS_FUNCTION_COMPUTES_POWER_COUPLING_RESPONSE_FOR_A_GIVEN_K(U)_PROFILE
global F_MIN    F_MAX     N_F     V_PROP     N_SEC     N_QW_LEN     K_MAX
F_CENTER     P_NOMINAL
global N_PARAMS_SPLINE     INTERPOLANT_MODE     ITER_COUNT_SPLINE
ITER_MAX_SPLINE

%-------------------------------------------------------------------------
%% ======SPLINE_INTERPOLATION_OF_PARAMETERS=========================%

PARAMETER_CURRENT_SPLINE=PARAMETER_SPLINE ;
% =======================================================================
% CREATE_INDEX_FOR_QUERY_POINTS_TO_EVALUATE_SPLINE
% =======================================================================


u_index = double(ones(N_QW_LEN,N_SEC));
for n =1:(N_QW_LEN)
u_index(n,:) = (linspace((n-1),n,(N_SEC)));
end

% =======================================================================
% CONCATENATE_U_INDEXES_FOR_QUERY_POINTS
% =======================================================================


U_INDEX = u_index(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_QW_LEN-1)
U_INDEX_TEMP= u_index((n),(1:(N_SEC-1)));
U_INDEX = horzcat(U_INDEX,U_INDEX_TEMP);
end
U_INDEX_RIGHT = horzcat(U_INDEX,u_index(N_QW_LEN,(1:(N_SEC))));%___END_INDEX

% =======================================================================
```

```matlab
% GET_MIRRORED_INDEX
% =======================================================================

U_INDEX_MIRRORED=-U_INDEX_RIGHT;
U_INDEX_LEFT = fliplr(U_INDEX_MIRRORED);

U_INDEX_COMPLETE =horzcat(U_INDEX_LEFT(1:end-
1),U_INDEX_RIGHT);%___THE_COMPLETE_INDEX


% =======================================================================
% CONCATENATE_U_INDEXES_AND_KNOWN_PARAMETERS_FOR_QUERY_POINTS
% =======================================================================

PARAMETER_SPLINE_RIGHT=[K_MAX,PARAMETER_CURRENT_SPLINE,0];
PARAMETER_SPLINE_LEFT=fliplr(PARAMETER_SPLINE_RIGHT);
SYMMETRIC_PARAMETER_SPLINE=horzcat(PARAMETER_SPLINE_LEFT(1:end-
1),PARAMETER_SPLINE_RIGHT);%___THE_COMPLETE_INDEX


% =======================================================================
% GET_MIRRORED_KNOWN_PARAMETER_INDEX
% =======================================================================

PARAM_INDEX_RIGHT_ALL = linspace(0,N_QW_LEN,(N_PARAMS_SPLINE+2));
PARAM_INDEX_RIGHT=PARAM_INDEX_RIGHT_ALL;
PARAM_INDEX_LEFT  = -fliplr(PARAM_INDEX_RIGHT);
KNOWN_PARAMETER_INDEX=horzcat(PARAM_INDEX_LEFT(1:end-
1),PARAM_INDEX_RIGHT);%___THE_COMPLETE_INDEX



if
((strcmp(INTERPOLANT_MODE,'TRUE')))%_PROCEED_WITH_SPLINE_INTERPOLATION_IF_TRU
E
    K_INTERPOLATED_FULL =
spline(KNOWN_PARAMETER_INDEX,SYMMETRIC_PARAMETER_SPLINE,U_INDEX_COMPLETE);
else
    K_INTERPOLATED_FULL =
pchip(KNOWN_PARAMETER_INDEX,SYMMETRIC_PARAMETER_SPLINE,U_INDEX_COMPLETE);
end



half=ceil(length(K_INTERPOLATED_FULL)/2);
K_INTERPOLATED=K_INTERPOLATED_FULL(half:end);

%-----------------------------------------------------------------------
%% ======NUMERIC_K_TO_SPECTRUM_ANALYSIS=================================%
k=K_INTERPOLATED;
U=U_INDEX;
del_U =U(2)-U(1);


% =======================================================================
% EVALUATION_OF_PARAMETERS_NEEDED_FOR_CALCULATIONS
% =======================================================================
```

```matlab
F_eval =linspace(F_MIN,F_MAX,N_F);%_____FREQUENCY_POINTS_FOR_ANALYSIS
QW_LENGTH = V_PROP/(4*F_CENTER);%_____QUARTER_WAVELENGTH
W_EVAL =2*pi*F_eval;%_____ANGULAR_FREQUENCY_IN_RADIANS_PER_SECOND
N_S= length(U);%_____TOTAL_NUMBER_OF_SECTIONS_FOR_ANALYSIS
del_L=del_U*QW_LENGTH;%_DIFFERENTIAL_LENGTH_FOR_ANALYSIS_IN_TERMS_OF_METERS**
*************CHECK THIS PROBLEM
beta = W_EVAL/V_PROP;%_____PROPAGATION_CONSTANT_OF_THE_GIVEN_MEDIUM
theta = beta*del_L;%_____ANGULAR_RESOLUTION_FOR_ANALYSIS




%-------------------------------------------------------------------------
%% ======INITIAL_MATRICES_AND_VALUES====================================%


% =========================================================================
% PRE_ALLOCATION_OF_MEMORY_NEEDED_FOR_CALCULATIONS
% =========================================================================

rho = (ones(1,N_S));
sigma = (ones(1,N_S));
pie = (ones(1,N_S));
k_p = (ones(1,N_S));
k_k = (ones(1,N_S));
PIE_NS= (ones(1,N_F));
C_SQUARED_MAG= (ones(1,N_F));




% =========================================================================
% EVALUATION_OF_INITIAL_VALUES_NEEDED_FOR_CALCULATIONS
% =========================================================================

rho(1) = (1);
sigma(1)= (1/(sqrt(1-(k(1).^2))));
pie(1) =k(1)*sigma(1);

% =========================================================================
% PRECALCULATION_OF_REPETATIVE_VALUES_FOR_FAST_CALCULATION
% =========================================================================




SINE = (sin(theta));%_____SINUSOIDAL_FUNCTION_PRE_CALCULATION
SINE_SQUARED =(SINE.^2);%_____SINUSOIDAL_SQUARED_PRE_CALCULATION

for s =1:N_S
  k_p(s) = 1/(sqrt(1-(k(s).^2)));%_____PREMULTIPLIED_FACTOR_1
  k_k(s) = k_p(s).*k(s);%_____PREMULTIPLIED_FACTOR_2
end


%-------------------------------------------------------------------------
%% ======RECURSIVE_MULTIPLIPLICATION====================================%
```

```matlab
for n=1:N_F
 for s =2:N_S
    rho(s) = rho(s-1) -((2*SINE_SQUARED(n)).*( rho(s-1) + ((k_p(s)).*sigma(s-
1)) - ((k_k(s)).*(pie(s-1))))) ;
    sigma(s) = (sigma(s-1)) + ((k_p(s)).*(rho(s) + rho(s-1)));
    pie(s) =    (pie(s-1))  + ((k_k(s)).*(rho(s) + rho(s-1)));
  end
PIE_NS(n)=pie(N_S );
C_SQUARED_MAG_NUM= (((PIE_NS(n)).^2).*(SINE_SQUARED(n)));
C_SQUARED_MAG_DEN=(1+(((PIE_NS(n)).^2).*(SINE_SQUARED(n))));
C_SQUARED_MAG(n) =((C_SQUARED_MAG_NUM)./(C_SQUARED_MAG_DEN));
end


P = 10*log10(C_SQUARED_MAG);



%-------------------------------------------------------------------------

%% ======DETERMINE_ERROR_CURVE=========================================%

error_curve=P-P_NOMINAL;

if(PLOT_THE_FIGURE_SPLINE)
figure;

subplot(1,2,1);
plot((F_eval/1e9),error_curve);
title(['' num2str(N_QW_LEN) ' -QW-LENGTH SPLINE: "'
num2str(ITER_COUNT_SPLINE) '" OF "' num2str(ITER_MAX_SPLINE) '"
ITERATIONS']);
xlabel("FREQUENCY IN GHz");
ylabel("ERROR");
% =========================================================================
% DRAW_REFERENCE_LINE
% =========================================================================
hold on;
y = [0 0];
x = [(F_MIN/1E9) (F_MAX/1E9)];
line(x,y,'Color','red','LineStyle','--')
hold off;


subplot(1,2,2);
plot(K_INTERPOLATED);
xlabel("QUARTER-WAVE-SECTIONS-INDEX");
ylabel("COUPLING-PROFILE");

% =========================================================================
% DRAW_REFERENCE_LINE_COUPLING_PROFILE
% =========================================================================

hold on;
```

```matlab
y = [0 0];
x = [1 length(K_INTERPOLATED)];
line(x,y,'Color','red','LineStyle','--')
hold off;


end


%-------------------------------------------------------------------------

%% ======PICK_EXTREMUM_POINTS===========================================%

[ ~ ,LOCS_MAXIMA] = findpeaks(error_curve );
[~ ] = -findpeaks(-error_curve);
[~ ,LOCS_MINIMA] = findpeaks(-error_curve);



column=horzcat(LOCS_MAXIMA,LOCS_MINIMA);

B = sort(column);
EXTREMUM=error_curve(B);
OMEGA=F_eval(B);



% =======================================================================
% CALCULATION_OF_SIGMA_MATRIX
% =======================================================================
SIGMA = zeros(1 ,length(F_eval ));
SIGMA(LOCS_MAXIMA)=1;
SIGMA(LOCS_MINIMA)=-1;
SIGMA=SIGMA(B);


%-------------------------------------------------------------------------

%% ======CALCULATE_D_AVG================================================%

% =======================================================================
% CALCULATION_OF_D_AVERAGE
% =======================================================================

m = length(OMEGA);
OMEGA_INDEX=1:1:m;
D_AVG=(1/m)*sum(abs(EXTREMUM));

%-------------------------------------------------------------------------

%% ======CALCULATE_PEAK_TO_PEAK_RIPPLE==================================%

% =======================================================================
% CALCULATION_OF_MAXIMUM_RIPPLE
% =======================================================================

 ripple_deviation =abs(EXTREMUM);
 RIPPLE_MAX= max(ripple_deviation);
```

```matlab
%------------------------------------------------------------------------
end



function [ ] = SPLINE_INITIAL_PARAMETER_GENERATOR(  )
%THIS_FUNCTION_GENERATES_INITIAL_PARAMETERS_FOR_SPLINE_OPTIMIZATIONS

global N_PARAMS_SPLINE    N_SEC  N_QW_LEN   SPLINE_INITIAL_PARAMS
K_STD_OPTIMIZED;

%% ======GET_THE_PARAMETERS=============================================%




% ========================================================================--
----check
% CREATE_PIECE_WISE_INDEX__ALONG_HALF_COUPLER_LENGTH
% ========================================================================

u_index = double(ones(N_QW_LEN,N_SEC));
for n =1:(N_QW_LEN)
u_index(n,:) = (linspace((n-1),n,(N_SEC)));
end



% ========================================================================
% CONCATENATE_U_INDEXES
% ========================================================================



U_INDEX = u_index(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_QW_LEN-1)
U_INDEX_TEMP= u_index((n),(1:(N_SEC-1)));
U_INDEX = horzcat(U_INDEX,U_INDEX_TEMP);
end
U_INDEX = horzcat(U_INDEX,u_index(N_QW_LEN,(1:(N_SEC))));%___END_INDEX

U_INDEX_SPACING=U_INDEX(2)-U_INDEX(1);
% ========================================================================
% INDEX_TO_PICK_OUT_PARAMETERS
% ========================================================================

U_REQUIRED_ALL = linspace(0,N_QW_LEN,(N_PARAMS_SPLINE+2));
U_REQUIRED=U_REQUIRED_ALL(2:end-1);%_____REQUIRED_INDICES_TO_PICK_OUT

% ========================================================================
% PICK_OUT_INITIAL_PARAMETERS
% ========================================================================


for n =1:(N_PARAMS_SPLINE)
```

```matlab
index_found(n) = find(abs(U_INDEX-U_REQUIRED(n)) <
(0.75*U_INDEX_SPACING),1,'first');
end

index_found = sort(index_found);

SPLINE_INITIAL_PARAMS=K_STD_OPTIMIZED(index_found);


end


function [   ] = EXCATLY_DETERMINED_OPTIMIZER( OMEGA_INDEX,  D_AVG , m ,
EXTREMUM ,k_std_form ,SIGMA ,P_BAR)
%THIS_FUNCTION_PERFORMS_EXACTLY_DETERMINED_MODE_OF_OPTIMIZATION

global DELTA     HALF_ITER_MAX ITER_COUNT PARAMETER_CURRENT N_PARAMS;
global  STD_PARAMETER_MIN_BOUND  STD_PARAMETER_MAX_BOUND   HALF_ITER_TOLERANCE

% m=6;% remove_after_use


%% ======CALCULATION_OF_THE_JACOBIAN_MATRIX_D===========================%

% ========================================================================
% CREATE_DELTA_PARAMETER_SPACING_MATRIX
% ========================================================================

clearvars EXTREMUM_NEW EXTREMUM_CURRENT;%_CLEARING_VARIABLES_FOR_SIMPLICITY

DELTA_parameter= double(DELTA.*ones(1,N_PARAMS));




% ========================================================================
% CALCULATE_THE_INITIAL_EXTREMUM_ERROR
% ========================================================================

CURRENT_FUNCTION= subs(k_std_form,P_BAR,PARAMETER_CURRENT);
[EXTREMUM_CURRENT,~,SIGMA,~,~,~] =SPECTRUM_ANALYZER_STANDARD(
CURRENT_FUNCTION);
EXTREMUM_CURRENT=transpose(EXTREMUM_CURRENT);
EXTREMUM_CURRENT = repmat(EXTREMUM_CURRENT,1,N_PARAMS);


% ========================================================================
% CALCULATE_THE_DELTA_EXTREMUM_ERROR_FROM_CURRENT_PARAMETER_POINTS
% ========================================================================

INCREMENTED_PARAMETER=PARAMETER_CURRENT   ;
INCREMENTED_PARAMETER = repmat(INCREMENTED_PARAMETER,N_PARAMS,1);

% ========================================================================
```

180

```matlab
% CALCULATION_OF_DIFFERENTIAL_ERROR_CURVE
% ========================================================================

for j =1:(N_PARAMS)%_____MATRIX_FOR_INCREMENTED_PARAMETERS
    INCREMENTED_PARAMETER(j,j)=
INCREMENTED_PARAMETER(j,j)+DELTA_parameter(1,j);
end

for j =1:(N_PARAMS)%_____ERROR_CHANGE_CALCULATIONS_FOR_ALL_EXTREMUMS

 DELTA_FUNCTION= subs(k_std_form,P_BAR,INCREMENTED_PARAMETER(j,:));
 [EXTREMUM_NEW(j,:),~,~,~,~,~] =SPECTRUM_ANALYZER_STANDARD( DELTA_FUNCTION);

end

EXTREMUM_NEW=transpose(EXTREMUM_NEW);

% ========================================================================
% CALCULATE_THE_JACOBIAN
% ========================================================================

D=EXTREMUM_NEW-EXTREMUM_CURRENT;%_____CALCULATION_OF_DIFFERENTIAL_ERROR
D=D./DELTA;

D=D((1:(N_PARAMS+1)),(1:N_PARAMS));
%------------------------------------------------------------------------
%% ======REFORMATTING_THE_MATRIX_FOR_LINEAR_SYSTEM======================%


% ========================================================================
% FORMING_OF_THE_KNOWN_OF_LEFT_HAND_SIDE
% ========================================================================

SIGMA_APPEND= transpose(-SIGMA);
SIGMA_APPEND=SIGMA_APPEND((1:(N_PARAMS+1)),:);
LHS_KNOWN=horzcat(D,SIGMA_APPEND);

% ========================================================================
% FORMING_OF_THE_KNOWN_OF_RIGHT_HAND_SIDE
% ========================================================================

RHS_KNOWN=(-
EXTREMUM_CURRENT((1:(N_PARAMS+1)),1));%____RIGHT_HAND_SIDE_OF_LINEAR_SYSTEM_T
O_SOLVE



%------------------------------------------------------------------------
%% ======SOLVING_LINEAR_SYSTEM_FOR_CHANGE_IN_PARAMETERS=================%

PARAMETER_CURRENT_BACKUP=PARAMETER_CURRENT;%_____PARAMETER_BACKUP


% ========================================================================
% DETERMINE_CHANGE_IN_PARAMETERS_USING_LINEAR_SYSTEM
% ========================================================================
```

```matlab
del_p = (pinv(LHS_KNOWN))*RHS_KNOWN;
del_p=del_p(1:end-1);%_____REMOVE_DUMMY_VALUE
del_p = transpose(del_p);


% ========================================================================
% UPDATE_THE_CURRENT_PARAMETERS
% ========================================================================


for i=1:N_PARAMS%_____PARAMETER_UPDATE_LOOP
PARAMETER_CURRENT(i)=PARAMETER_CURRENT(i)+del_p(i);
end


%-------------------------------------------------------------------------


%% ======DECISION_FLOW_FOR_CONSTRAINT_CHECK==============================%


% ========================================================================
% DETECTING_MAXIMUM_AND_MINIMUM_PARAMETER
% ========================================================================


MIN_OF_PARAMETERS =min(PARAMETER_CURRENT);
MAX_OF_PARAMETERS =max(PARAMETER_CURRENT);

if(((MIN_OF_PARAMETERS)>(STD_PARAMETER_MIN_BOUND))&&((MAX_OF_PARAMETERS)<(STD
_PARAMETER_MAX_BOUND)))
    %------------DO-NOTHING--------------------------------------------------%

  else
    % ====================================================================
    % DECREMENT_AND_CHECK_FOR_ERROR
    % ====================================================================
    HALVING_COUNT=HALF_ITER_MAX ;%_____REGISTER_FOR_HALVING_COUNT
    PARAMETER_CURRENT= PARAMETER_CURRENT_BACKUP;%__RELOAD_PREVIOUS_PARAMETERS
          % ============================================================
          % DECREMENT_AND_CHECK_FOR_ERROR
          % ============================================================
          while (HALVING_COUNT ~= 0)%_____EXIT_IF_MAX_ITERATIONS_DONE
                  HALVING_COUNT=HALVING_COUNT-1 ;%_____DECREMENT_REGISTER
                  del_p=(1/2)*del_p;
                  TEMP_PARAMETERS=PARAMETER_CURRENT+del_p;
                  k_std_form_substituted=
subs(k_std_form,P_BAR,PARAMETER_CURRENT);
                  k_std_form_substituted_half=
subs(k_std_form,P_BAR,TEMP_PARAMETERS);

                  [~,~,~,~,RIPPLE_MAX_DEFAULT,~]
=SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted);
                  [~,~,~,~,RIPPLE_MAX_HALVED,~]
=SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted_half);
                              if((RIPPLE_MAX_HALVED+HALF_ITER_TOLERANCE) <=
RIPPLE_MAX_DEFAULT)%_____EXIT_WHILE_LOOP_IF_ERROR_IMPROVED
```

```matlab
PARAMETER_CURRENT=PARAMETER_CURRENT+del_p;%_____UPDATE_PARAMETER_IF_ERROR_IMP
ROVED
                                    break
                                end
                end
end


%------------------------------------------------------------------------

%% ======CALL_OPTIMIZER_FOR_REFINING====================================%

ITER_COUNT =ITER_COUNT+1;%_____INCREMENT_ITERATION_COUNT
OPTIMIZER_STANDARD(k_std_form,P_BAR);%_____CALL_STANDARD_OP
TIMIZER


%------------------------------------------------------------------------


end


function [ K_NUMERICAL ] = K_STANDARD_TO_NUMERIC_CONVERTER(
k_std_form_substituted )
%THIS_FUNCTION_COMPUTES_POWER_COUPLING_RESPONSE_FOR_A_GIVEN_K(U)_PROFILE
global   N_SEC N_PARAMS
syms u;

%% ======SYMBOLIC_FUNCTION_TO_NUMERIC_UPDATE===========================%



% ========================================================================
% CREATE_PIECE_WISE_INDEX_FOR_NUMERICAL_EVALUATION_OF_FUNCTION
% ========================================================================

u_index = ones(N_PARAMS,N_SEC);
for n =1:(N_PARAMS)
u_index(n,:) = sym(linspace((n-1),n,(N_SEC)));
end

% ========================================================================
% EVALUATE_FUNCTION_AT_INDICES
% ========================================================================

 k_eval_piece_double = ones(N_PARAMS,N_SEC);
for n =1:(N_PARAMS)
k_eval_piece_double(n,:)=double(subs(
k_std_form_substituted(n),u,u_index(n,:)));
end


% ========================================================================
% CONCATENATE_U_INDEXES
% ========================================================================
```

```matlab
u_index=double(u_index);
U_INDEX = u_index(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_PARAMS-1)
U_INDEX_TEMP= u_index((n),(1:(N_SEC-1)));
U_INDEX = horzcat(U_INDEX,U_INDEX_TEMP);
end
U_INDEX = horzcat(U_INDEX,u_index(N_PARAMS,(1:(N_SEC))));%___END_INDEX


% =========================================================================
% CONCATENATE_K(U)_FUNCTION_NUMERICALLY_EVALUATED
% =========================================================================

K_NUMERICAL= k_eval_piece_double(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_PARAMS-1)
K_NUMERICAL_TEMP= k_eval_piece_double((n),(1:(N_SEC-1)));
K_NUMERICAL = horzcat(K_NUMERICAL,K_NUMERICAL_TEMP);
end
K_NUMERICAL =
horzcat(K_NUMERICAL,k_eval_piece_double(N_PARAMS,(1:(N_SEC))));%___END_INDEX

% =========================================================================
% EVALUATE_K(U)_AT_0_WITH_LIMITS_THIS_DETERMINES_K_MAX
% =========================================================================

limiting_function =k_std_form_substituted(1);
limit_val=limit(limiting_function, u, 0);
k_max_limit_double=double(limit_val);
K_NUMERICAL(1)=k_max_limit_double;



%-------------------------------------------------------------------------
end



function [    ] = OPTIMIZER_STANDARD( k_std_form,P_BAR)
%THIS_FUNCTION_GENERATES_OPTIMIZED_COUPLING_PROFILE_FOR_A_STANDARD_PARAMETRIC
_FORM
tic;
global N_PARAMS P_NOMINAL   RIPPLE_OPTIM    ITER_MAX     ITER_COUNT    N_QW_LEN
;
global PARAMETER_CURRENT  INITIAL_ITERATION   STD_OPTIMIZATION_COMPLETE
K_STD_OPTIMIZED K_STD_EXTREMUM
syms u;
%% ======PARAMETER_SUBSTITUTION=========================================%


% =========================================================================
% SUBSTITUTION_OF_PARAMETERS_IN_SYMBOLIC_FUNCTION
% =========================================================================
if(strcmp(INITIAL_ITERATION,'TRUE'))%___EVALUATE_WITH_INITIAL_PARAMETERS_IF_I
NITIAL_CONDITION_TRUE
        P_BAR_EVAL_INITIAL = ones([1 N_PARAMS]);
        for n =1:(N_PARAMS)
            P_BAR_EVAL_INITIAL(n) =(P_NOMINAL*((log(10))/20));
        end
```

```matlab
        k_std_form_substituted= subs(k_std_form,P_BAR,P_BAR_EVAL_INITIAL);
        PARAMETER_CURRENT=P_BAR_EVAL_INITIAL;
        INITIAL_ITERATION='FALSE';%____SET_INITIAL_ITERATION_TO_FALSE
else

        k_std_form_substituted= subs(k_std_form,P_BAR,PARAMETER_CURRENT);

end




%-------------------------------------------------------------------------

%% ======SPECTRUM_ANALYZER=======================================%

% ========================================================================
% EVALUATE_POWER_COUPLING_RESPONSE_OF_EVALUATED_K(U)_NUMERICAL
% ========================================================================

[EXTREMUM,~,SIGMA,D_AVG,RIPPLE_MAX,OMEGA_INDEX] =SPECTRUM_ANALYZER_STANDARD(
k_std_form_substituted);

m=length(EXTREMUM);%_____NUMBER_OF_EXTREMUMS

fig = gcf;
savefig(fig,['STANDARD/ITERATIVE_SPECTRUM_IMAGES_STANDARD/ITERATION__',num2st
r(ITER_COUNT),'_OUT_OF_',num2str(ITER_MAX),'_HALF_QW_LENGTH_',num2str(N_QW_LE
N),'.fig'],'compact');%__SAVE_THIS_FIGURE
%-------------------------------------------------------------------------
%% ======OPTIMIZATION_DECISION=======================================%

if(ITER_COUNT >= ITER_MAX )%___EXIT_OPTIMIZATION_IF_MAXIMUM_ITERATIONS_DONE
    %------------DO-NOTHING----------------------------------------------%
    STD_OPTIMIZATION_COMPLETE='STD_OPTIMIZATION_MAXED_ITERATION';
elseif (RIPPLE_MAX <= RIPPLE_OPTIM
)%END_OPTIMIZATION_IF_ERROR_BELOW_TOLERANCE
    %------------DO-NOTHING----------------------------------------------%
    STD_OPTIMIZATION_COMPLETE='STD_OPTIMIZATION_MET_CONDITIONS';
elseif (RIPPLE_MAX >= RIPPLE_OPTIM)%_____NOT_OPTIMIZED

% ========================================================================
% DECIDE_THE_MODE_OF_OPTIMIZATION
% ========================================================================

        if (m == (N_PARAMS+1))%_____EXCATLY_DETERMINED_OPTIMIZATION
EXCATLY_DETERMINED_OPTIMIZER( OMEGA_INDEX,  D_AVG , m , EXTREMUM ,k_std_form
,SIGMA ,P_BAR) ;
        else%_____RECTANGULAR_OPTIMIZATION
RECTANGULAR_OPTIMIZER( OMEGA_INDEX,  D_AVG , m , EXTREMUM ,k_std_form ,SIGMA
,P_BAR) ;
        end
end
%-------------------------------------------------------------------------
%% ======RETURNS_THE_STD_PROFILE_IF_OPTIMIZATION_COMPLETED==============%
```

```matlab
 close all;%_____CLOSES_ALL_OPEN_FIGURES
 k_std_form_substituted=subs(k_std_form,P_BAR,PARAMETER_CURRENT);
 [~,~,~,~,~,~] =SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted);
 K_STD_OPTIMIZED  = K_STANDARD_TO_NUMERIC_CONVERTER( k_std_form_substituted
);
 K_STD_EXTREMUM =EXTREMUM;
%----------------------------------------------------------------------
end


function [ ] = RECTANGULAR_OPTIMIZER( OMEGA_INDEX,  D_AVG , m , EXTREMUM
,k_std_form ,SIGMA ,P_BAR )
%THIS_FUNCTION_PERFORMS_UNDER_DETERMINED_MODE_OF_OPTIMIZATION
global DELTA  K_MAX  HALF_ITER_MAX ITER_COUNT PARAMETER_CURRENT N_PARAMS;
global  STD_PARAMETER_MIN_BOUND  STD_PARAMETER_MAX_BOUND  HALF_ITER_TOLERANCE
 m=5;%____remove aftre_use

%% ======CALCULATION_OF_THE_JACOBIAN_MATRIX_D===========================%


% =========================================================================
% CREATE_DELTA_PARAMETER_SPACING_MATRIX
% =========================================================================

clearvars EXTREMUM_NEW EXTREMUM_CURRENT;%_CLEARING_VARIABLES_FOR_SIMPLICITY
DELTA_parameter= double(DELTA.*ones(1,N_PARAMS));


% index= min(m,N_PARAMS);%INDEX_DETERMINES_THE_DIMENSION_OF_JACOBIAN_MATRIX_D

% =========================================================================
% CALCULATE_THE_INITIAL_EXTREMUM_ERROR
% =========================================================================

CURRENT_FUNCTION= subs(k_std_form,P_BAR,PARAMETER_CURRENT);
[EXTREMUM_CURRENT,~,SIGMA,~,~,~] =SPECTRUM_ANALYZER_STANDARD(
CURRENT_FUNCTION);
EXTREMUM_CURRENT=transpose(EXTREMUM_CURRENT);
EXTREMUM_CURRENT = repmat(EXTREMUM_CURRENT,1,N_PARAMS);
% EXTREMUM_CURRENT = EXTREMUM_CURRENT((1:index),(1: index)) ;


% =========================================================================
% CALCULATE_THE_DELTA_EXTREMUM_ERROR_FROM_CURRENT_PARAMETER_POINTS
% =========================================================================

INCREMENTED_PARAMETER=PARAMETER_CURRENT  ;
INCREMENTED_PARAMETER = repmat(INCREMENTED_PARAMETER,N_PARAMS,1);


% =========================================================================
% INSERTION_OF_THE_DELTA_CHANGE_IN_PARAMETERS
% =========================================================================

for j =1:(N_PARAMS)%_____MATRIX_FOR_INCREMENTED_PARAMETERS
```

```matlab
    INCREMENTED_PARAMETER(j,j)=
INCREMENTED_PARAMETER(j,j)+DELTA_parameter(1,j);
end


% ==========================================================================
% CALCULATION_OF_DIFFERENTIAL_ERROR_CURVE
% ==========================================================================


for j =1:(N_PARAMS)%_____ERROR_CHANGE_CALCULATIONS_FOR_ALL_EXTREMUMS

 DELTA_FUNCTION= subs(k_std_form,P_BAR,INCREMENTED_PARAMETER(j,:));
 [EXTREMUM_NEW(j,:),~,~,~,~,~] =SPECTRUM_ANALYZER_STANDARD( DELTA_FUNCTION);

end




EXTREMUM_NEW=transpose(EXTREMUM_NEW);
% EXTREMUM_NEW = EXTREMUM_NEW((1:index),(1: index)) ;
% ==========================================================================
% CALCULATE_THE_JACOBIAN
% ==========================================================================

D=EXTREMUM_NEW-EXTREMUM_CURRENT;%_____CALCULATION_OF_DIFFERENTIAL_ERROR
D=D./DELTA;


%--------------------------------------------------------------------------
%% ======CALCULATION_OF_DESIRED_ERROR_CHANGE============================%

DESIRED_ERROR_CHANGE= (1/2)*SIGMA*D_AVG -EXTREMUM;
DESIRED_ERROR_CHANGE=transpose(DESIRED_ERROR_CHANGE);
%--------------------------------------------------------------------------

%%
======CALCULATION_OF_PSEUDO_PENROSE_INVERSE_FOR_PARAMETER_CHANGE======%%CHEC
K_LATER_FOR_WHICH_METHOD_TO_USE_FOR_PSEUDO_INVERSE
PARAMETER_CURRENT_BACKUP=PARAMETER_CURRENT;%_____PARAMETER_BACKUP


% ==========================================================================
% CALCULATE_THE_PENROSE_PSEUDO_INVERSE
% ==========================================================================

del_p = pinv(D)*DESIRED_ERROR_CHANGE ;

del_p = transpose(del_p);

% ==========================================================================
% UPDATE_THE_CURRENT_PARAMETERS
% ==========================================================================

for i=1:N_PARAMS%_____PARAMETER_UPDATE_LOOP
PARAMETER_CURRENT(i)=PARAMETER_CURRENT(i)+del_p(i);
```

```matlab
end

%-----------------------------------------------------------------------

%% ======DECISION_FLOW_FOR_CONSTRAINT_CHECK============================%

% ======================================================================
% DETECTING_MAXIMUM_AND_MINIMUM_PARAMETER
% ======================================================================

MIN_OF_PARAMETERS =min(PARAMETER_CURRENT);
MAX_OF_PARAMETERS =max(PARAMETER_CURRENT);

if(((MIN_OF_PARAMETERS)>(STD_PARAMETER_MIN_BOUND))&&((MAX_OF_PARAMETERS)<(STD
_PARAMETER_MAX_BOUND)))
    %-------------DO-NOTHING----------------------------------------------%

    else
    % ==================================================================
    % DECREMENT_AND_CHECK_FOR_ERROR
    % ==================================================================
      HALVING_COUNT=HALF_ITER_MAX ;%_____REGISTER_FOR_HALVING_COUNT
      PARAMETER_CURRENT= PARAMETER_CURRENT_BACKUP;%__RELOAD_PREVIOUS_PARAMETERS
            % ============================================================
            % DECREMENT_AND_CHECK_FOR_ERROR
            % ============================================================
            while (HALVING_COUNT ~= 0)%_____EXIT_IF_MAX_ITERATIONS_DONE
                    HALVING_COUNT=HALVING_COUNT-1 ;%_____DECREMENT_REGISTER
                    del_p=(1/2)*del_p;
                    TEMP_PARAMETERS=PARAMETER_CURRENT+del_p;
                    k_std_form_substituted=
subs(k_std_form,P_BAR,PARAMETER_CURRENT);
                    k_std_form_substituted_half=
subs(k_std_form,P_BAR,TEMP_PARAMETERS);

                    [~,~,~,~,RIPPLE_MAX_DEFAULT,~]
=SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted);
                    [~,~,~,~,RIPPLE_MAX_HALVED,~]
=SPECTRUM_ANALYZER_STANDARD( k_std_form_substituted_half);
                            if((RIPPLE_MAX_HALVED+HALF_ITER_TOLERANCE) <=
RIPPLE_MAX_DEFAULT)%_____EXIT_WHILE_LOOP_IF_ERROR_IMPROVED

PARAMETER_CURRENT=PARAMETER_CURRENT+del_p;%_____UPDATE_PARAMETER_IF_ERROR_IMP
ROVED
                                        break
                            end
            end
end

%-----------------------------------------------------------------------

%% ======CALL_OPTIMIZER_FOR_REFINING===================================%

ITER_COUNT =ITER_COUNT+1;%_____INCREMENT_ITERATION_COUNT
```

```matlab
OPTIMIZER_STANDARD(k_std_form,P_BAR);%_____CALL_STANDARD_OP
TIMIZER


%--------------------------------------------------------------------------


end



function [EXTREMUM,OMEGA,SIGMA,D_AVG,RIPPLE_MAX,OMEGA_INDEX] =
SPECTRUM_ANALYZER_STANDARD(
k_std_form_substituted)%_____CHECK_FOR_ERRORS_IN_FIND_PEAKS_SOMETIMES
%THIS_FUNCTION_COMPUTES_POWER_COUPLING_RESPONSE_FOR_A_GIVEN_K(U)_PROFILE
global F_MIN    F_MAX    N_F    V_PROP    N_SEC    N_QW_LEN    F_CENTER
P_NOMINAL    ITER_COUNT    ITER_MAX
syms u;




%% ======SYMBOLIC_FUNCTION_TO_NUMERIC_UPDATE=============================%



% ========================================================================--
----check
% CREATE_PIECE_WISE_INDEX__ALONG_HALF_COUPLER_LENGTH
% ========================================================================

u_index = double(ones(N_QW_LEN,N_SEC));
for n =1:(N_QW_LEN)
u_index(n,:) = (linspace((n-1),n,(N_SEC)));
end

% ========================================================================
% EVALUATE_FUNCTION_AT_INDICES
% ========================================================================

 k_eval_piece_double = double(ones(N_QW_LEN,N_SEC));
for n =1:(N_QW_LEN)
k_eval_piece_double(n,:)=(subs( k_std_form_substituted(n),u,u_index(n,:)));
end



% ========================================================================
% CONCATENATE_U_INDEXES
% ========================================================================

 u_index=double(u_index);
U_INDEX = u_index(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_QW_LEN-1)
U_INDEX_TEMP= u_index((n),(1:(N_SEC-1)));
U_INDEX = horzcat(U_INDEX,U_INDEX_TEMP);
end
U_INDEX = horzcat(U_INDEX,u_index(N_QW_LEN,(1:(N_SEC))));%___END_INDEX

% ========================================================================
```

189

```matlab
% CONCATENATE_K(U)_FUNCTION_NUMERICALLY_EVALUATED
% =======================================================================

K_NUMERICAL= k_eval_piece_double(1,(1:(N_SEC-1)));%_START_INDEX
for n =2:(N_QW_LEN-1)
K_NUMERICAL_TEMP= k_eval_piece_double((n),(1:(N_SEC-1)));
K_NUMERICAL = horzcat(K_NUMERICAL,K_NUMERICAL_TEMP);
end
K_NUMERICAL =
horzcat(K_NUMERICAL,k_eval_piece_double(N_QW_LEN,(1:(N_SEC))));%___END_INDEX


% =======================================================================
% EVALUATE_K(U)_AT_0_WITH_LIMITS_THIS_DETERMINES_K_MAX
% =======================================================================

limiting_function =k_std_form_substituted(1);
limit_val=limit(limiting_function, u, 0);
k_max_limit_double=double(limit_val);
K_NUMERICAL(1)=k_max_limit_double;



%-----------------------------------------------------------------------
%% ======NUMERIC_K_TO_SPECTRUM_ANALYSIS===============================%
k=K_NUMERICAL;
U=U_INDEX;
del_U =U(2)-U(1);


% =======================================================================
% EVALUATION_OF_PARAMETERS_NEEDED_FOR_CALCULATIONS
% =======================================================================



F_eval =linspace(F_MIN,F_MAX,N_F);%_____FREQUENCY_POINTS_FOR_ANALYSIS
QW_LENGTH = V_PROP/(4*F_CENTER);%_____QUARTER_WAVELENGTH
W_EVAL =2*pi*F_eval;%_____ANGULAR_FREQUENCY_IN_RADIANS_PER_SECOND
N_S= length(U);%_____TOTAL_NUMBER_OF_SECTIONS_FOR_ANALYSIS
del_L=del_U*QW_LENGTH;%_DIFFERENTIAL_LENGTH_FOR_ANALYSIS_IN_TERMS_OF_METERS**
*************CHECK THIS PROBLEM
beta = W_EVAL/V_PROP;%_____PROPAGATION_CONSTANT_OF_THE_GIVEN_MEDIUM
theta = beta*del_L;%_____ANGULAR_RESOLUTION_FOR_ANALYSIS
% X_LEN=U*QW_LENGTH;%_____HALF_LENGTH_OF_THE_COUPLER_IN_METERS


%-----------------------------------------------------------------------
%% ======INITIAL_MATRICES_AND_VALUES===============================%


% =======================================================================
% PRE_ALLOCATION_OF_MEMORY_NEEDED_FOR_CALCULATIONS
% =======================================================================

rho = (ones(1,N_S));
sigma = (ones(1,N_S));
pie = (ones(1,N_S));
k_p = (ones(1,N_S));
```

```matlab
k_k = (ones(1,N_S));
PIE_NS= (ones(1,N_F));
C_SQUARED_MAG= (ones(1,N_F));




% ========================================================================
% EVALUATION_OF_INITIAL_VALUES_NEEDED_FOR_CALCULATIONS
% ========================================================================

rho(1) = (1);
sigma(1)= (1/(sqrt(1-(k(1).^2))));
pie(1) =k(1)*sigma(1);


% ========================================================================
% PRECALCULATION_OF_REPETATIVE_VALUES_FOR_FAST_CALCULATION
% ========================================================================



SINE = (sin(theta));%_____SINUSOIDAL_FUNCTION_PRE_CALCULATION
SINE_SQUARED =(SINE.^2);%_____SINUSOIDAL_SQUARED_PRE_CALCULATION

for s =1:N_S
  k_p(s) = 1/(sqrt(1-(k(s).^2)));%_____PREMULTIPLIED_FACTOR_1
  k_k(s) = k_p(s).*k(s);%_____PREMULTIPLIED_FACTOR_2
end

%-------------------------------------------------------------------------
%% ======RECURSIVE_MULTIPLIPLICATION=====================================%



for n=1:N_F
 for s =2:N_S
    rho(s) = rho(s-1) -((2*SINE_SQUARED(n)).*( rho(s-1) + ((k_p(s)).*sigma(s-
1)) - ((k_k(s)).*(pie(s-1)))))  ;
    sigma(s) = (sigma(s-1)) + ((k_p(s)).*(rho(s) + rho(s-1)));
    pie(s) =     (pie(s-1))  + ((k_k(s)).*(rho(s) + rho(s-1)));
 end
PIE_NS(n)=pie(N_S );
C_SQUARED_MAG_NUM= (((PIE_NS(n)).^2).*(SINE_SQUARED(n)));
C_SQUARED_MAG_DEN=(1+(((PIE_NS(n)).^2).*(SINE_SQUARED(n))));
C_SQUARED_MAG(n) =((C_SQUARED_MAG_NUM)./(C_SQUARED_MAG_DEN));
end


P = 10*log10(C_SQUARED_MAG);



%-------------------------------------------------------------------------

%% ======DETERMINE_ERROR_CURVE==========================================%
```

```matlab
error_curve=P-P_NOMINAL;
figure;
plot((F_eval/1e9),error_curve);
title(['ERROR CURVE STANDARD: "' num2str(ITER_COUNT) '" OF "'
num2str(ITER_MAX) '" ITERATIONS']);
xlabel("FREQUENCY IN GHz");
ylabel("DEVIATION-ERROR");


% ========================================================================
% DRAW_REFERENCE_LINE
% ========================================================================
hold on;
y = [0 0];
x = [(F_MIN/1E9) (F_MAX/1E9)];
line(x,y,'Color','red','LineStyle','--')
hold off;




%----------------------------------------------------------------------------

%% ======PICK_EXTREMUM_POINTS====================================================%

[ ~ ,LOCS_MAXIMA] = findpeaks(error_curve );
[~ ] = -findpeaks(-error_curve);
[~ ,LOCS_MINIMA] = findpeaks(-error_curve);



column=horzcat(LOCS_MAXIMA,LOCS_MINIMA);

B = sort(column);
EXTREMUM=error_curve(B);
OMEGA=F_eval(B);



% ========================================================================
% CALCULATION_OF_SIGMA_MATRIX
% ========================================================================
SIGMA = zeros(1 ,length(F_eval ));
SIGMA(LOCS_MAXIMA)=1;
SIGMA(LOCS_MINIMA)=-1;
SIGMA=SIGMA(B);

%----------------------------------------------------------------------------

%% ======CALCULATE_D_AVG====================================================%

% ========================================================================
% CALCULATION_OF_D_AVERAGE
% ========================================================================

m = length(OMEGA);
OMEGA_INDEX=1:1:m;
D_AVG=(1/m)*sum(abs(EXTREMUM));
```

192

```matlab
%--------------------------------------------------------------------------
%% ======CALCULATE_PEAK_TO_PEAK_RIPPLE================================%


% =========================================================================
% CALCULATION_OF_MAXIMUM_RIPPLE
% =========================================================================

 ripple_deviation =abs(EXTREMUM);
 RIPPLE_MAX= max(ripple_deviation);
%--------------------------------------------------------------------------
end



function [ K,u,P_BAR] = STANDARD_PARAMETRIC_FORM_GENERATOR(   )
%THIS_FUNCTION_GENERATES_STANDARD_PARAMETRIC_FORM_FROM_THE_DESIGN_REQUIREMENT
S

global N_PARAMS ;

%--------------------------------------------------------------------------
%% ======EVALUATE_PIECEWISE_INTEGRAL_FUNCTION ==========================%

syms u x ;
assume( u >=0);
P_BAR = sym('P_BAR',[1 N_PARAMS]);%_____TUNING_PARAMETERS_SYMBOLIC_MATRIX
W = sym('W',[1 N_PARAMS]);%_____TUNING_WEIGHTS
% =========================================================================
% WEIGHING_FUNCTION_GENERATION
% =========================================================================

for n =1:(N_PARAMS)
W =exp(P_BAR);
end

% =========================================================================
% BASIC_SINE_SQAURED_FUNCTION_S(U))
% =========================================================================

sine_function =(sin(pi*x));
sine_squared_function=((sine_function^2)/x);

% =========================================================================
% EVALUATION_OF_WEIGHTED_INTEGRALS
% =========================================================================


fun = sym('fun',[1 N_PARAMS]);
function_of_u = sym('function_of_u',[1 N_PARAMS]);

for v =1:(N_PARAMS)
fun(v) =W(v)*sine_squared_function;
end
```

```matlab
for v =1:(N_PARAMS)
function_of_u(v) =int(fun(v),x,u,N_PARAMS);
end
%-------------------------------------------------------------------------
%% ======EVEN_MODE_IMPEDANCE_FUNCTION_GENERATION=========================%

Z_O_1 = sym('Z_O_1',[1 N_PARAMS]);
Z_O = sym('Z_O',[1 N_PARAMS]);
Z_O_SQUARED = sym('Z_O_SQUARED',[1 N_PARAMS]);

for v =1:(N_PARAMS)
Z_O_1(v) =(2/pi)*function_of_u(v);
end

for v =1:(N_PARAMS)
 Z_O(v) =exp(Z_O_1(v));
 Z_O_SQUARED(v) =(Z_O(v)^2);
end



%-------------------------------------------------------------------------
%% ======PIECE_WISE_COUPLING_SYMBOLIC_FUNCTION_GENERATION===============%

K = sym('K',[1 N_PARAMS]);

for v =1:(N_PARAMS)
K(v) = ((Z_O_SQUARED(v))-1)/((Z_O_SQUARED(v))+1);
end

K=(pi/2)*K;%_____REMOVE_THIS_SCALING_FACTOR


% P_BAR_EVAL=[-0.91826 -0.98728 -1.09078 -1.25054 -1.47631 -1.37331];
%  k_std_form_substituted= subs(K,P_BAR,P_BAR_EVAL);
%  fplot((k_std_form_substituted),[-2 7]);
%  ylim([0 1]);



%-------------------------------------------------------------------------
End
function[ TRACE_WIDTH ,TRACE_OFFSET ]= FIND_DIMENSION(k_target)
%% ======FIND_DIMENSIONS=================================================%

global Z_O

% =========================================================================
% INITIAL_ESTIMATE_LOOK_UP_TABLE
% =========================================================================

[k, Z,TRACE_WIDTH_INIT,TRACE_OFFSET_INIT]=
LOOK_UP_QUERY(k_target);%_____LOOK_UP_FOR_THE_CLOSEST_DIMENSIONS
```

```matlab
[k_estimate,Z_O_estimate] =LAPLACE_SOLVER(TRACE_WIDTH_INIT ,TRACE_OFFSET_INIT
);


% =========================================================================
% DESICION_LOOP_FOR_OPTIMIZATION
% =========================================================================

if((k_estimate < k_target) && (Z_O_estimate <
Z_O))%_____IF_K_ESTIMATE_AND_Z_O_ESTIMATE_IS_LESS_THAN_TARG
ET


  %conditional_actions_1



elseif((k_estimate < k_target) && (Z_O_estimate >
Z_O))%_____IF_K_ESTIMATE_IS_LESS_THAN_TARGET_BUT_Z_O_ESTIMATE_I
S_GREATER

    %conditional_actions_2

elseif((k_estimate > k_target) && (Z_O_estimate <
Z_O))%_____IF_K_ESTIMATE_IS_GREATER_THAN_TARGET_BUT_Z_O_ESTIMAT
E_IS_LESSER
    %conditional_actions_3


elseif((k_estimate > k_target) && (Z_O_estimate >
Z_O))%_____IF_K_ESTIMATE_AND_Z_O_ESTIMATE_IS_GREATER_THAN_TARGE
T
  %conditional_actions_4
end




end

%-------------------------------------------------------------------------

end


function [K_MAX_APPROXIMATED] =K_MAX_APPROXIMATOR( )
%THIS_FUNCTION_GENERATES_PHYSICAL_DIMENSIONS_FOR_A_GIVEN_SYMMETRIC_COUPLING_P
ROFILE

global   E_R   s   b   Z_O

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

195

```matlab
%% ======DETERMINE_FRINGING_CAPACITANCE=================================%


%
% ========================================================================
% DETERMINE_FRINGING_CAPACITANCE_FROM_APPROXIMATE_FORMULA
% ========================================================================



syms r  k w;
r_sub=s/b;%_____RATIO_OF_CONDUCTOR_TO_GROUND_SPACING
k_sub=w/b;%_____RATIO_OF_CONDUCTOR_WIDTH_TO_GROUND_SPACING




% ========================================================================
% DETERMINE_EVEN_FRINGING_CAPACITANCE_TO_ABSOLUTE_PERMITIVITY
% ========================================================================

EVEN_DEN = (r/(1-r))*log(1/r);
EVEN_DEN = EVEN_DEN +log(1/(1-r));
EVEN_DEN= 0.4413+((1/pi)*EVEN_DEN);



% ========================================================================
% DETERMINE_ODD_FRINGING_CAPACITANCE_TO_ABSOLUTE_PERMITIVITY
% ========================================================================

ODD_DEN = (r/(1-r))*log(1/r);
ODD_DEN = ODD_DEN +log(1/(1-r));
ODD_DEN=(1/r)*(1/pi)*ODD_DEN;

%------------------------------------------------------------------------
%% ======DETERMINE_EVEN_AND_ODD_MODE_IMPEDANCE==========================%

ZO_e_num = 188.3/sqrt(E_R);
ZO_e_den =( k/(1-r))+ EVEN_DEN;
ZO_e= ZO_e_num/ZO_e_den;

ZO_o_num = 188.3/sqrt(E_R);
ZO_o_den =( k/(1-r))+ (w/s)+ODD_DEN;
ZO_o= ZO_o_num/ZO_o_den;

eqn=Z_O==sqrt(ZO_e*ZO_o);
eqn=subs(eqn,k,(k_sub));
eqn=subs(eqn,r,(r_sub));

%------------------------------------------------------------------------
%% ======DETERMINE_WIDTH_OF_CENTER_SECTION==============================%

width = vpasolve(eqn,w);

%------------------------------------------------------------------------
%% ======DETERMINE_K_MAX_REQUIRED=======================================%
```

```matlab
ZO_e=subs(ZO_e,k,(k_sub));
ZO_e=subs(ZO_e,r,(r_sub));
ZO_e=double(subs(ZO_e,w,width));
rho_max=((ZO_e/Z_O)^2);

K_MAX_APPROXIMATED=((ZO_e^2)-1)/((ZO_e^2)+1);

 K_MAX_APPROXIMATED=(sqrt(rho_max)-1)/(sqrt(rho_max)+1);


%----------------------------------------------------------------------
end
```

8.2 Matlab code for estimating the power handling capability of a silicon shunt diode for a given intrinsic region thickness, applied negative bias and duty cycle.

```matlab
%% ======PIN_DIODE_REVERSE BIAS_VOLTAGE_CALCULATOR======================%


% ========================================================================
% CLEARS_PREVIOUS_VARIABLES_AND_ADDS THE_FOLDER_PATH_TO_THE_FUNCTION
% ========================================================================
clc;
clearvars;
currentFolder = pwd;
addpath(genpath(currentFolder));

%_____LOCATION_OF_THE_PATH_TO_THE_FUNCTION_AND_ITS_SUBFUNCTIONS
%----------------------------------------------------------------------


% ========================================================================
% DIODE_PARAMETERS
% ========================================================================

global F_MHz   D  W_mil

F_MHz=2000;%_____Frequency in MHz

Tp=200e-6;%_____Pulse period

m=3;%_____duty cycle values

Td=linspace(50e-6,150e-6,m);%_____Pulse duration multiple
```

```matlab
% Td=100e-6;%_____Pulse duration

D=Td/Tp;%_____Duty cycle

W_micron=80;%_____Intrinsic region width in um

W_mil=W_micron*5/127;%_____Intrinsic region width in mils

n=1000;  %_____calculation points

P=linspace(0,300,n); %_____power range in watts

dBm=10*log10(P)+30;%_____power in dBm

V_rf=10.^((dBm-10)/20);%_____Peak applied RF voltage


% ========================================================================
% PLOT POWER_VS_REVERSE BIAS VOLTAGE MINIMUM
% ========================================================================

for l=1:m

D_C= D(l);

for k =1:n

a= (0.056*V_rf(k)*sqrt(D_C))/W_mil;

b= 1+sqrt(1+ (a.^2));

c= (0.0142*F_MHz *(W_mil.^2))/(V_rf(k)*sqrt(D_C)*b);

d= sqrt(1+(c^2));

V_dc(k) =V_rf(k)/d;


end

plot(P,-V_dc);
hold on;
end

title("MINIMUM REVERSE BIAS FOR 200us PULSE PERIOD (HIGH ALTITUDE
APPLICATIONS)");
xlabel("PULSED POWER IN WATTS");
ylabel(" MINIMUM REVERSE BIAS VOLTAGE IN VOLTS");

hold on;
y = [-20 -20];
```

```
x = [0 P(n)];
line(x,y,'Color','red','LineStyle','--')
ylim([-25 0]);
xlim([0 70])
hold off;
```