VLA Interference Memo # 18

# Radio Frequency Interference Satellite Tracking Station (STS)

**New Mexico Tech**
**May 5, 2001**


**Group Members:**
**Lenard Noice**
**Ryan Schmidt**
**Edward Szpindor**
**Herald Vahle**

**Customer:**
**NRAO,**
**Raul Armendariz**

**Project Advisor:**
**Dr. Mathur**

# Satellite Tracking Station
# 2000 – 2001 Senior Design Project

Lenard Noice
Ryan Schmidt
Edward Szpindor
Herald Vahle

# Table of Contents


Satellite Tracking Station ........ 1
Section I: STS Quick Reference Guide ........ 4
Nova ........ 4
LabVIEW ........ 4
STS Control Program Setup ........ 4
STS Motor Control setup ........ 5
Satellite Selection ........ 6
Shutting down the system needs to be preformed in the following order. ........ 7
Section II: Motion Control Manual ........ 8
*1.0 Setting Up The Communication Parameters* ........ 10
1.0.1 Successful Communication ........ 10
1.0.2 Break Program screen ........ 11
1.0.3 Failed Communication ........ 11
*2.0 Navigating the BDS5 MotionLink Plus Main GUI* ........ 12
2.1.0 BDS5 Terminal ........ 12
2.1.1 Running the Program ........ 12
2.1.2 Editing The Program In The Terminal Screen ........ 13
2.1.3 Monitor Mode ........ 13
2.1.3.1 Printing Variables In The Monitor Mode ........ 14
2.1.3.2 Changing Variables ........ 14
2.1.3.3 Killing the Program ........ 14
*3.0 Programming the BDS5 Controllers* ........ 14
3.1 Writing and Compiling Code ........ 14
3.2 Programming the BDS5 ........ 15
3.2.1 Opening a .BDS File ........ 15
3.2.2 Saving Changes ........ 16
3.2.3 Sending Source Code to the BDS5 ........ 16
3.2.4 Receiving Source Code From the BDS5 ........ 16
3.2.4 Terminal Screen ........ 17
*4.0 Software and Setup References* ........ 17
4.1 Contact Information ........ 17
4.2 Error Messages ........ 17
4.3 Coding ........ 17
4.4 Tuning, and Motor Profiles ........ 18
Section III: Technical Documentation ........ 19
Section 3.1 – Systems Engineering ........ 19
3.1.1 Zero-order estimate for tracking rate requirements ........ 19
3.1.2 IPG Lightening Protection Assessment ........ 21
3.1.3 STS Antenna/feed Calibration and Pattern Measurement ........ 24
3.1.4 STS Calibration files ........ 26

3.1.5 Calibration Technique for Signal in Receiver ........ 28
3.1.6 Dynamic Data Exchange (DDE) Specifications for Nova for Windows ........ 29
Dish Tracking Osculation Problem ........ 30
The Problem ........ 30
Testing ........ 30
Solution ........ 30
Conclusion ........ 31
Section 3.3 – Motion Control Subsystem ........ 31
3.3.1 BDS-5 Controller Software ........ 31
Introduction to the BDS-5 Software ........ 31
1.0 Constants Explained ........ 31
2.0 Handshaking ........ 36
2.1 The Variable$ Label and Subroutine 20$ ........ 36
3.0 Without Handshaking ........ 38
3.3.2 BDS-5 Controller SoftwareUPDATE NOTICE ........ 45
Appendix C ........ 49
Appendix D ........ 51
3.3.3 Azimuth and Elevation Calibration ........ 52
DESIRED ANGLE * 398,996/1000 ........ 55
DESIRED ANGLE * 130,355/100 ........ 57
Appendix C ........ 57
Appendix D ........ 59
3.4 – Antenna and RF Data Collection Subsystem ........ 60
3.4.1 Satellite Tracking Design Group ........ 60
3.4.2 Undersampling Test of Spectrum Analyzer ........ 87
3.4.3 Amplifier Power Line Application ........ 89
3.4.4 Front End Composition ........ 90
3.4.5 Excel PFD Calibration File: ........ 92
3.4.6 Plots of the RF System Budget (component measurement) ........ 95

# Section I: STS Quick Reference Guide

The Following procedure needs to be followed in the order listed below in order for system to operate properly.

## Nova

**Nova** is the software used for tracking a satellite chosen by the operator, this program must be started first in order to setup the DDE (Dynamic Data Exchange) communications between Nova and the LabVIEW code.

1) Open **Nova** by double clicking on the icon on the desktop. Nova
2) Setup **Nova** per Nova manual.

## LabVIEW

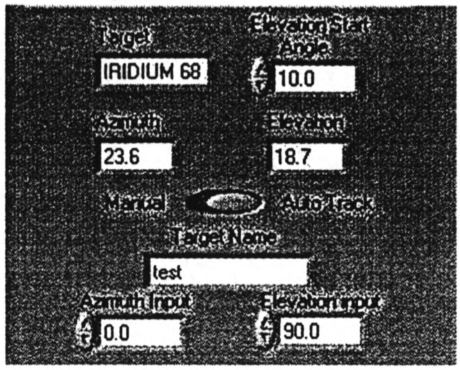
**LabVIEW** is the software used to run the program to interface and control the Motion Control System and the Receiver, and used to collect data, process and display the PFD data.

## STS Control Program Setup

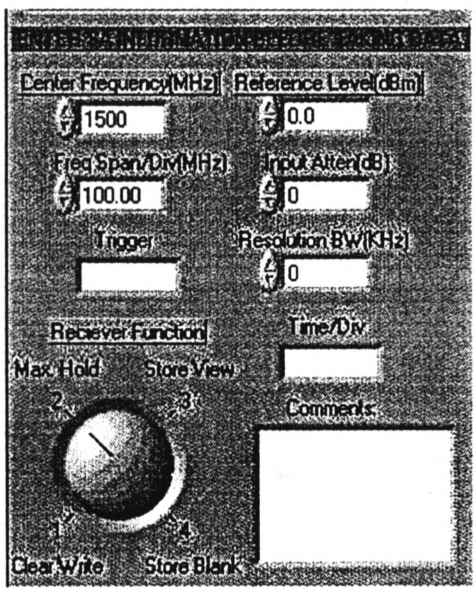
1) Start **LabVIEW** by double clicking on the icon on the desktop. LabVIEW.exe
2) Select **Open Vi**.
3) Go to directory **D:\STStracking**.
4) Select and open **STSControl.vi**.
5) Select **Run mode**. Default = Run mode.
6) Input path where the data will be saved in the **Path** block. Default = D:\data\.
7) Select **Save File** if file is to be saved. Default = not saved
   With Save File selected, in Auto Track mode the file will not be save until Minimum Elevation Angle has been reached and will stop when elevation has gone below the Minimum Elevation Angle. In Manual Mode this is 0 deg.


8) Choose **Manual** or **Auto Track**.
   Skip to #11 if **Auto Track** is selected.
9) Enter target name into the **Target Name** block.
10) Enter desired Azimuth and Elevation into the **Azimuth Input** and **Elevation Input** Default = 0 deg. and 90 deg.
11) Enter desired **Elevation Minimum Angle** Default = 10 deg.

In **Manual** mode the minimum angle is 0 deg. regardless of **Elevation Minimum Angle** setting.




12) Setup Spectrum Analyzer (SA) per mission requirements.
13) Enter all information on SA setup into appropriate blocks.
14) Enter any comments into the **Comment** block.
15) Select appropriate **Receiver Function**.

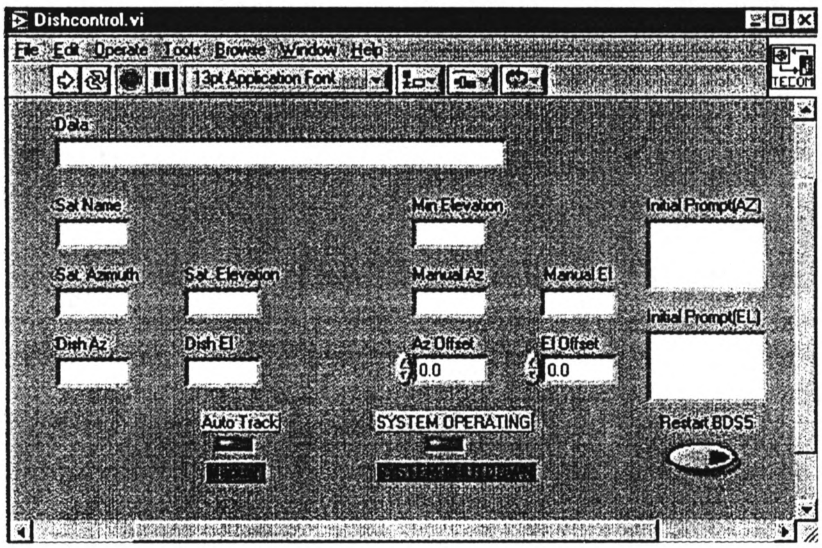



16) Click on **White Arrow (Run)** at the top of the program, this will start the program running.


## STS Motor Control setup

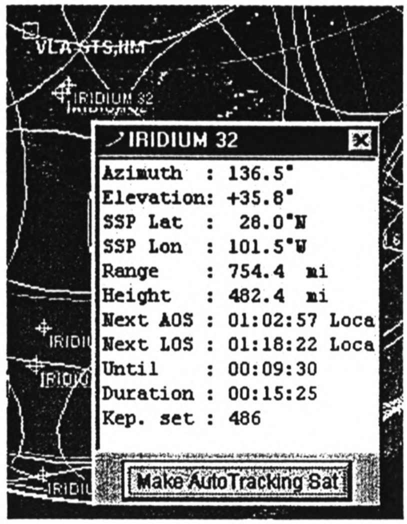
1) Select **File** button at top of LabVIEW program.

2) Select **Open** button.
3) Go to Directory **D:\STStracking**.
4) Select and open **Dishcontrol.vi**.
5) Adjust **Az Offset** and **El Offset** only if Calibration is required.
6) Click on **White Arrow (Run)** at the top of the program, this will start the program. If program was properly shutdown the last time it was used, in the **Az initial** block and the **El initial** block should display BDS5, and the dish will go through an initial zero alignment procedure. If there is a → in the two block try pushing the **Restart BDS5** button. If this is unsuccessful then shutdown the LabVIEW Program and reset the power to the BDS5's and start the setup procedure over.




## Satellite Selection

1) Return to **Nova** program.
2) Select the desired Satellite from the selected Constellation that is running by clicking on the satellite itself.
3) A box will appear showing satellite information and a button at the bottom of the box, click on it to make that satellite autotrack.
4) Close this box by clicking on the X. Do not kill the Nova software.
5) Return to LabVIEW STS Control program.




Shutting down the system needs to be preformed in the following order.

**STScontrol program**

1) Position select button to **Shutdown/Stow**.
2) Program has halted when Arrow turns white.
3) Program then can be closed.


**STS Motor Control program**

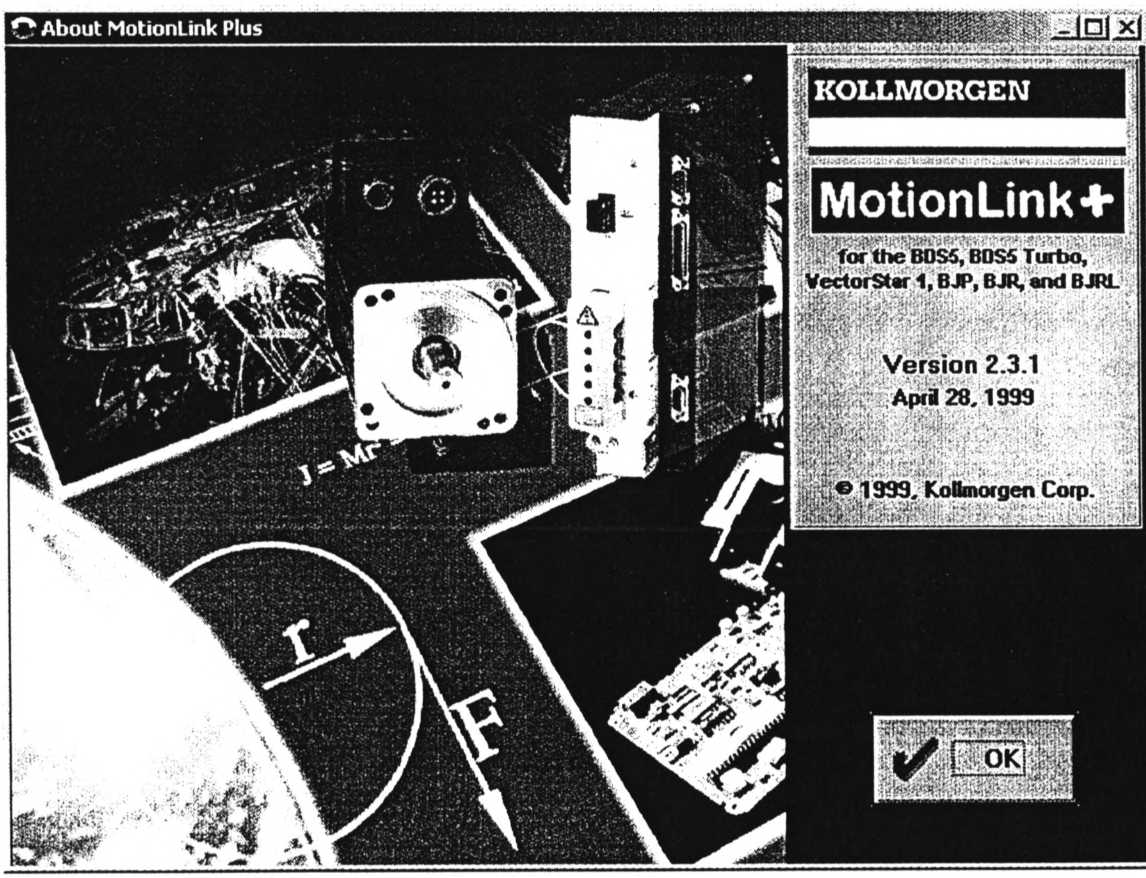
1) There is nothing to do here except to wait for the arrow to turn white. It will take approximately 45 seconds for program to shutdown. This delay is used to allow enough time for dish to stow. Once arrow turns white the program can be closed. Closing program before arrow turns white can cause problems when program is restarted.

**Nova**

1) At this time Nova can be closed.

# Section II: Motion Control Manual

BDS5 Operations Manual



BDS5 Operations Manual ........ 8
1.0 Setting Up The Communication Parameters ........ 10
1.0.1 Successful Communication ........ 10
1.0.2 Break Program screen ........ 11
1.0.3 Failed Communication ........ 11
2.0 Navigating the BDS5 MotionLink Plus Main GUI ........ 12
2.1.0 BDS5 Terminal ........ 12
2.1.1 Running the Program ........ 12
2.1.2 Editing The Program In The Terminal Screen ........ 13
2.1.3 Monitor Mode ........ 13
2.1.3.1 Printing Variables In The Monitor Mode ........ 14
2.1.3.2 Changing Variables ........ 14
2.1.3.3 Killing the Program ........ 14
3.0 Programming the BDS5 Controllers ........ 14
3.1 Writing and Compiling Code ........ 14
3.2 Programming the BDS5 ........ 15
3.2.1 Opening a .BDS File ........ 15
3.2.2 Saving Changes ........ 16
3.2.3 Sending Source Code to the BDS5 ........ 16
3.2.4 Receiving Source Code From the BDS5 ........ 16
3.2.4 Terminal Screen ........ 17
4.0 Software and Setup References ........ 17
4.1 Contact Information ........ 17
4.2 Error Messages ........ 17
4.3 Coding ........ 17
4.4 Tuning, and Motor Profiles ........ 18

The purpose of this document is to explain how to program and operate the BDS5 motor controllers through MotionLink Plus Version 2.3.1. Some of the material in this document was taken right out of the online help in MotionLink Plus.

## *1.0 Setting Up The Communication Parameters*

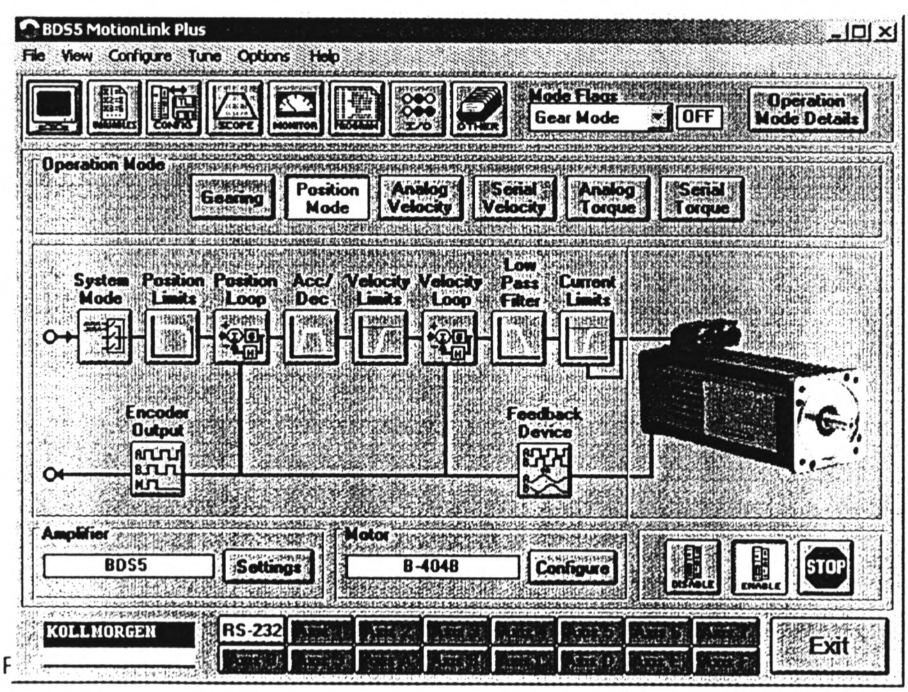
The first step that needs to be done before talking to the BDS5 controllers is setting up the communication ports. To setup the communication ports, do the following:

1. Click on the OK button on the startup screen
2. The "Start MotionLink Plus" screen will pop
3. Click on the COM SETUP" button
4. The "BDS5 Communications Settings" screen will popup
5. Click on which communication port to communicate with
6. Click Com 2 to communicate with the Azimuth controller or Click on Com 3 to communicate with the Elevation controller
7. Under the "Baud Rate" section, click on 19200
8. Click on OK

After clicking on OK, MotionLink will pop up a "MotionLink Plus Drive Scan" window. At this point, MotionLink is trying to communicate with the BDS5 controller.

### 1.0.1 <u>Successful Communication</u>

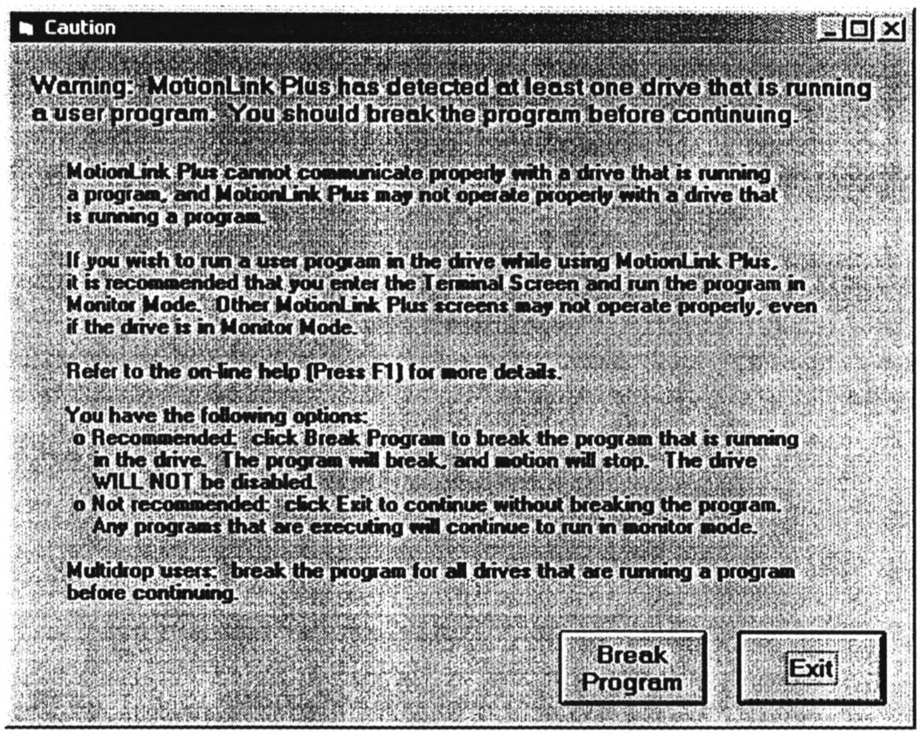
If MotionLink successfully communicates with the BDS5 controllers, the "BDS5 MotionLink Plus" window will pop, as shown below. At this point, the BDS5 controller is ready to be operated or programmed.




● F

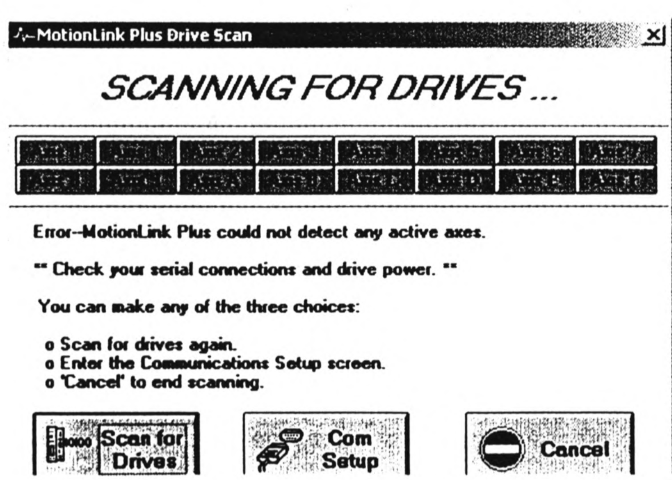
### 1.0.2 Break Program screen

If the controllers are running a program from a previous operation, a “Caution” screen will popup, as shown below. Simply click on the “BREAK PROGRAM” button to proceed. After doing this, the “BDS5 MotionLink Plus” will appear and the BDS5 controller is ready for use.




### 1.0.3 Failed Communication

Sometimes MotionLink will fail to communicate with the BDS5 controllers. The below screen will pop up.



In this case, do the following:

1. Make sure the serial lines are properly plugged into the computer.
2. Make sure that LabVIEW and Nova are not running. If they are, shut down each of these programs.
3. Repeat section 1.0 and check to make sure the communication settings are still setup correctly.
4. Turn off the power to the BDS5 controller, wait 10 seconds, and the turn the power back on. Step through section 1.0 again to communicate with the controller.

## *2.0 Navigating the BDS5 MotionLink Plus Main GUI*

Not all the menus will be explained. The only menus this manual will cover are the two menus that will allow the user to operate and program the BDS5 controller.

### 2.1.0 BDS5 Terminal

To get to the BDS5 terminal, click on the  icon.

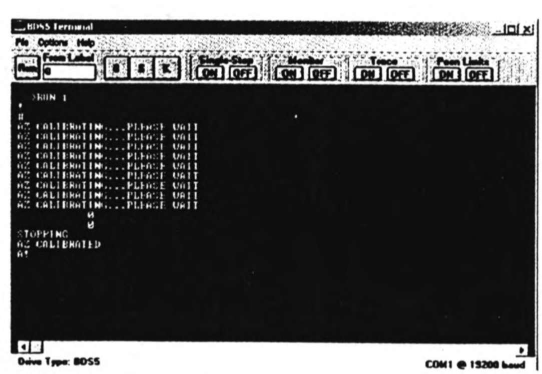
This screen emulates a computer terminal, allowing the user to type commands and variables and have them transmitted directly to the drive. This enables the user to exercise great flexibility and control over the active axis, as well as making it easy to investigate the settings of variables and flags without having to move from screen to screen within MotionLink's windows environment.

In addition, the user can also use this screen to test the user program. Here in the terminal screen, the can run the program, break it, stop it, kill it and run it in single-step, monitor, and trace modes.

The area of concern in the terminal screen is the prompt. If the user is interested in the control bars, pressing F1 will display the help menu. Once in the terminal, as mentioned, the user has the flexibility to run, edit, and kill the current program.

### 2.1.1 Running the Program

To run the current program stored in the BDS5 controller, simply type RUN 1 at the prompt

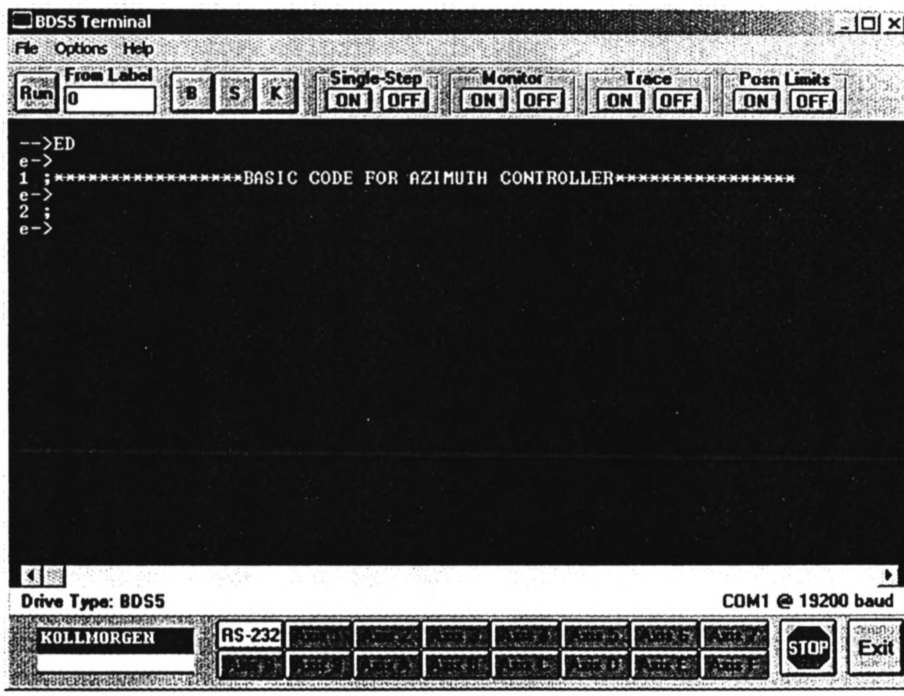


(→). This will begin running the program as depicted in the illustration.

### 2.1.2 Editing The Program In The Terminal Screen

This allows the user to edit the current user program. It is not recommended that the user edit the program from the terminal only because the user may edit the wrong line of code. The primary function of the edit command should be used to make sure that the program was successfully stored inside the BDS5.

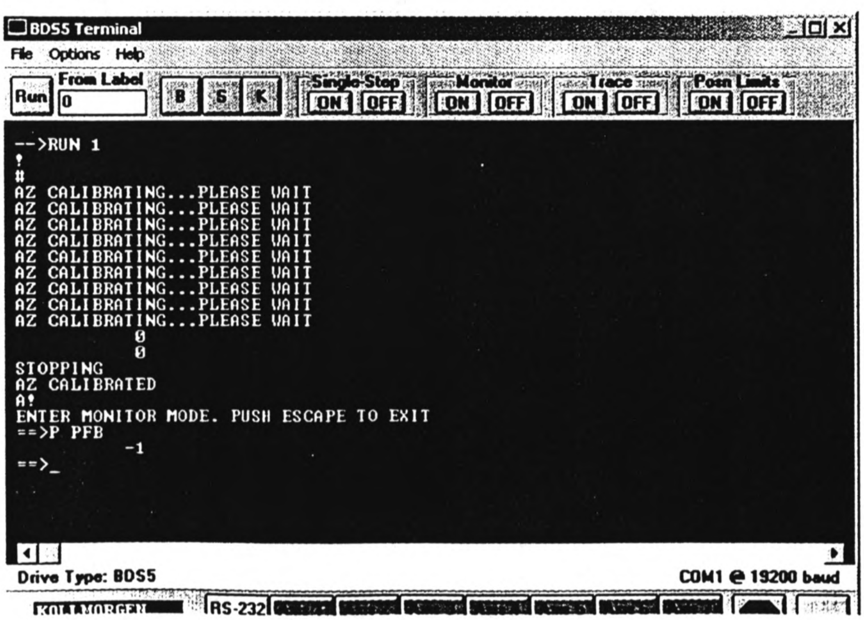
To enter the edit mode do the following:




1. If the program is running, kill the program.
2. At the prompt (→) screen type ED
3. Press enter

This will enter into the edit mode. To browse a new line of code, press enter. Below is an illustration to what the user will see while in this mode.

To exit the editor mode, simply press the Esc key on the keyboard.




### 2.1.3 Monitor Mode

When the BDS5 is in the monitor mode, the prompt transmits (==>) which is different that the (→)

prompt. While in this mode, the user can monitor different variables by printing out their values, changing values on the fly, and the user is able to kill the program while in this mode.

To enter the monitor mode, simply press the Esc key on the keyboard.

#### 2.1.3.1 Printing Variables In The Monitor Mode

While in the monitor mode, it is sometimes convenient to see what values of some variables are. To print out any variable, simply type, at the prompt (==>), the following:

1. P <variable name>
2. Press ENTER

#### 2.1.3.2 Changing Variables

While in the monitor mode, it is sometimes convenient to change certain variables. To change any variable, simply type, at the prompt (==>), the following:

1. <variable name> <new value>
2. Press ENTER

#### 2.1.3.3 Killing the Program

While in the monitor mode, it is sometimes convenient to kill the current process of the user program. To kill the current program, simply type, at the prompt (==>), the following:

1. K
2. Press ENTER

## *3.0 Programming the BDS5 Controllers*

This section of the document will show the user how to program the BDS5 controllers.

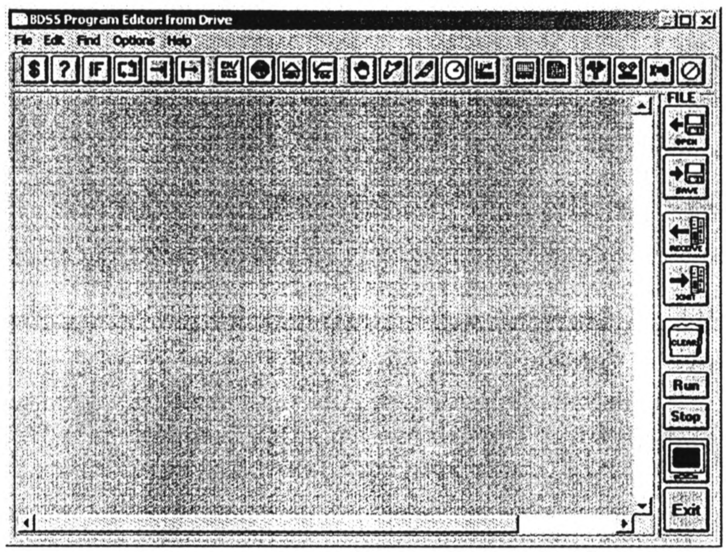
### 3.1 Writing and Compiling Code

The BDS5 use BASIC as the coding language. To write code, simply open any text editor, write the desired code, and save the file as a .BDS file. There is no need to compile the new written code since the BDS5 compile the code when it is run.

### 3.2 Programming the BDS5

From the "DS5 MotionLink Plus" main GUI, click on the 

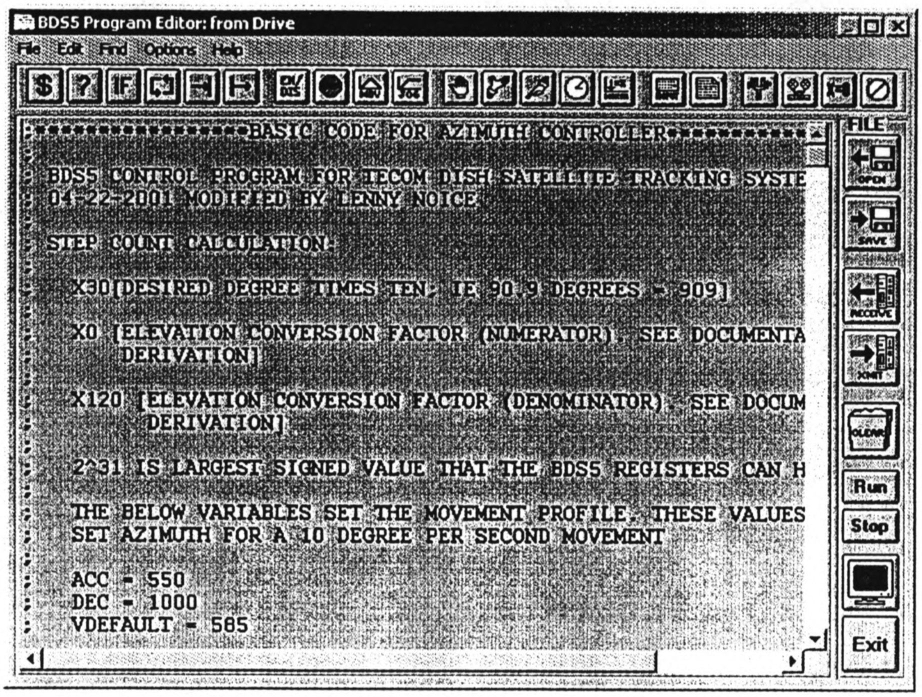
icon. This will bring up the "BDS5 Program Editor" window, as shown in the illustration.




#### 3.2.1 Opening a .BDS File

To open a .BDS file, click on the 

icon. Find the .BDS file of choice and double click on this file. The program editor will display your source code as shown below.




### 3.2.2 Saving Changes

Once the source code is displayed in the program editor, the user has the ability to make changed to the code. After modifying the code, the next step would be to save the newly changed code.

To save the file, click on the 

icon. Then save the file as a .BDS file to the desired directory.

### 3.2.3 Sending Source Code to the BDS5

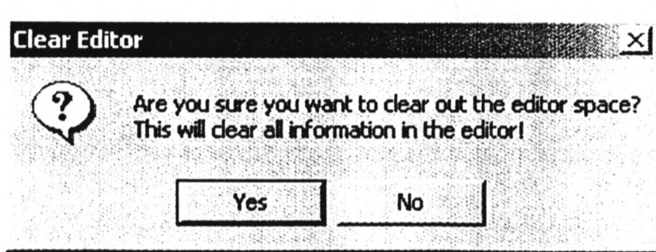
With the source code displayed inside the program editor, click on the 

icon. This will send the code into the BDS5 controller. As the program is being sent, the cursor will spin, once the cursor stops spinning, the program has been transferred to the controller.

### 3.2.4 Receiving Source Code From the BDS5

If the program editor is not cleared, do so by clicking on the 

icon. The following dialog box will appear. Click on YES to proceed.



With the editor clear, click on the  icon. The cursor will begin spinning, once it stops, the program that was in the BDS5 will be displayed in the program window editor.

### 3.2.4 Terminal Screen

From the program editor, the user can get into the terminal screen to run the new program by simply clicking on the  icon.

## *4.0 Software and Setup References*

As stated before, not all the functions of MotionLink were explained. The functions that were explained will help the user to operate and program the BDS5 controller.

### 4.1 Contact Information

If there are problems that require further assistance with the BDS5 controllers, the below information are the contact names to the engineers who can help.

http://www.motionvillage.com/customer_service/tech_services/emergency.html

It was our experience that the technical services were not helpful. So if you can't find a reliable source, you'll know why.

### 4.2 Error Messages

Sometimes the user may have run into error messages when trying to run a new program. In most cases, there are programming errors. All the errors that will be displayed on the screen can be referenced back to the BDS5 manual in Appendix C entitled Error Codes.

### 4.3 Coding

The BDS5 manuals provide a good source on how to program in BASIC. So for any coding questions refer to the BDS5 User's Manual chapters three and four.

### 4.4 Tuning, and Motor Profiles

The BDS5 manuals provide a good source on how to tune and set the motor profiles. So for any tuning and motor profile questions refer to Section 3-20 in the BDS5 User's Manual.

Note: The tuning variables and motion profiles were set by manually tweaking the variables for the optimum tracking performance as we saw fit. Refer to the systems parameter document for the tuning variables and motion profiles used.

# Section III: Technical Documentation

## Section 3.1 – Systems Engineering

3.1.1 Zero-order estimate for tracking rate requirements

**Assumptions**

1. **SOR and VLA are close enough that the error in tracking rate prediction is insignificant.**
2. **Iridium is the constellation or satellite of interest with the lowest mean altitude of 780 km.**
3. **SOR algorithm has been verified and tested.**
4. **The simplifications to the equations of motion have no significant effect on the az/el rates.**

III. Givens

A. Transmitter Data

$\Psi_T := 34.964\,\mathrm{deg}$

$\Omega_T := 253.54\,\mathrm{deg}$

$P_T := DU_{geo} + 1850\,\mathrm{m}$

$\lambda := 527 \cdot \mathrm{nm}$

$d_\Gamma := 15 \cdot \mathrm{cm}$

$BQ_\Gamma := 2.0$

$\tau_1 := 0.78$

$Q_L := 40\,\mathrm{J}$

$\sigma_J := 1.5 \cdot \mu\mathrm{rad}$

B. Receiver Data

$\Psi_V := 34.02\,\mathrm{deg}$

$\Omega_V := 253.01 \cdot \mathrm{deg}$

$P_V := DU_{geo} + 5900\,\mathrm{ft}$

$\tau_V := 0.5$

$Mag := 280$

$Bin := 10$

$d_{pix} := 20 \cdot \mu\mathrm{m}$

$\sigma_{RN} := 7 \cdot \mathrm{elec}$

$QE := .8 \cdot \mathrm{elec}$

C. Satellite Data

$Name_S :=$ "Iridum13"

$ID_S :=$ "24840/97030E"

$h_{S.a} := 780\,\mathrm{km}$

$h_{S.p} := 774\,\mathrm{km}$

$i_S := 86.4 \cdot \mathrm{deg}$

$d_{SX} := 3.3 \cdot \mathrm{m}$

$d_{SN} := 1.5 \cdot \mathrm{m}$

$\rho_S := 0.2$

$P_S := DU_{geo} + \frac{h_{S.a} + h_{S.p}}{2}$

D. Atmosphere

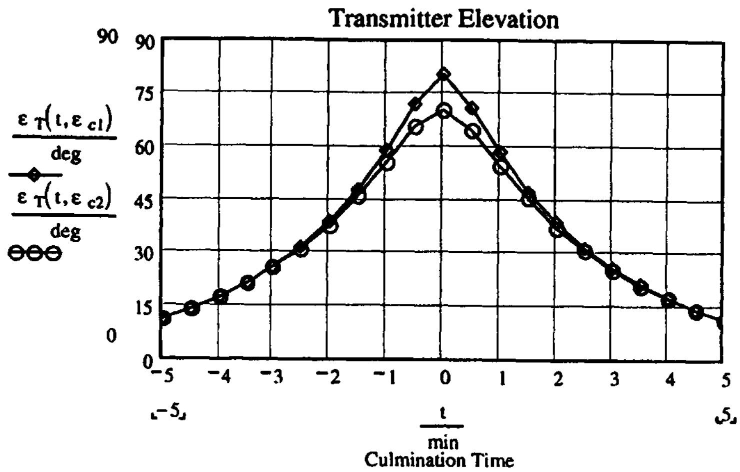
$r_{0.500} := 5 \cdot \mathrm{cm}$

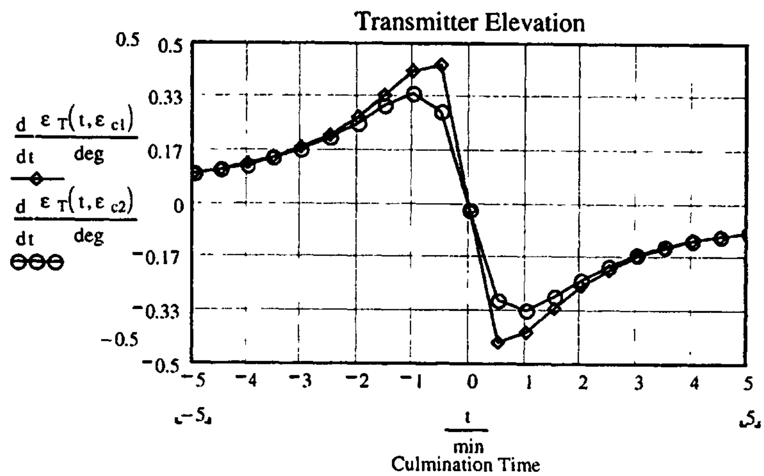
$\tau_{A\,zen} := 0.95$

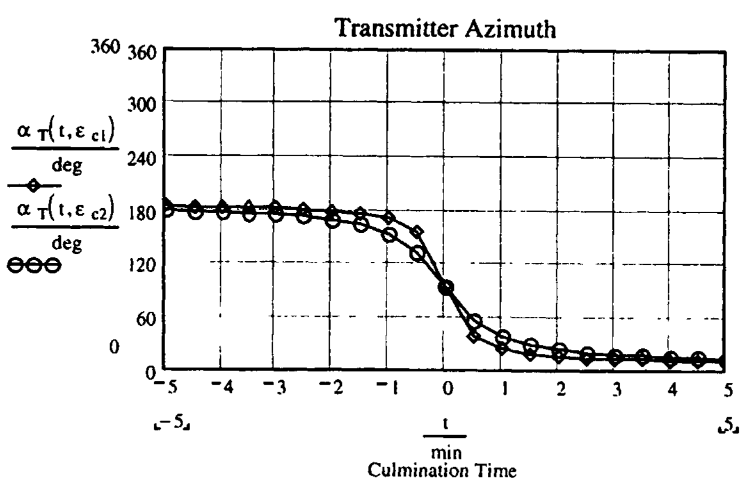
E. Elevation Angles

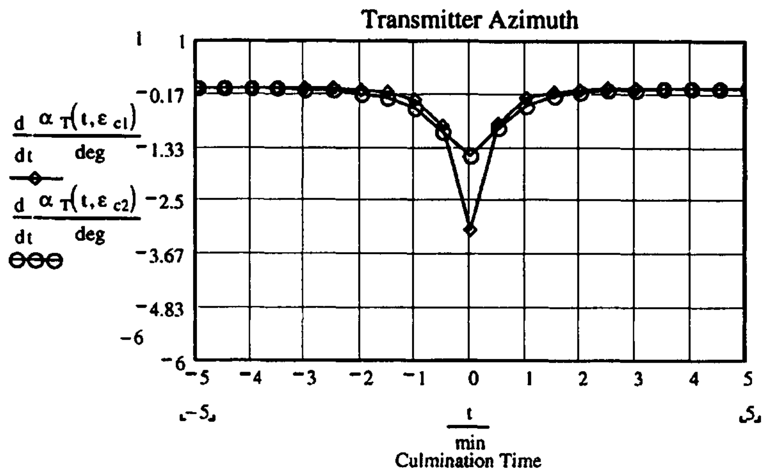
$\varepsilon_{c1} := 80 \cdot \mathrm{deg}$

$\varepsilon_{c2} := 70 \cdot \mathrm{deg}$


Transmitter Elevation
90
90
75
60
45
30
15
0
0
$\frac{\varepsilon_T(t, \varepsilon_{c1})}{deg}$
$\frac{\varepsilon_T(t, \varepsilon_{c2})}{deg}$
−5 −4 −3 −2 −1 0 1 2 3 4 5
−5
5
t / min
Culmination Time


Transmitter Elevation
0.5
0.5
0.33
0.17
0
−0.17
−0.33
−0.5
−0.5
$\frac{d}{dt}\frac{\varepsilon_T(t, \varepsilon_{c1})}{deg}$
$\frac{d}{dt}\frac{\varepsilon_T(t, \varepsilon_{c2})}{deg}$
−5 −4 −3 −2 −1 0 1 2 3 4 5
−5
5
t / min
Culmination Time


Transmitter Azimuth
360
360
300
240
180
120
60
0
0
$\frac{\alpha_T(t, \varepsilon_{c1})}{deg}$
$\frac{\alpha_T(t, \varepsilon_{c2})}{deg}$
−5 −4 −3 −2 −1 0 1 2 3 4 5
−5
5
t / min
Culmination Time




## 3.1.2 IPG Lightening Protection Assessment

Both EMS and STS design team have worked together to assess the lightening protection of IPG facility at the VLA site. FIG. 1 is a cabling of the current IPG facility. Both the STS and EMS interfaces to the IPG Operations Building are identified in the drawing. Standard ANSI reference designators were assigned to each cable. Table-1 provides a list of all cables shown in the drawing. Each cable is referenced to the drawing by its reference designator. The 'current protection' column provides a summary of the assessment for each cable. The next column provides a detailed description for recommended action to properly protect the cables and connected quipment from lightening damage.

---

In addition to the specific recommendations for individual cables, the following general recommendations are a result of the evaluation of the IPG facility.

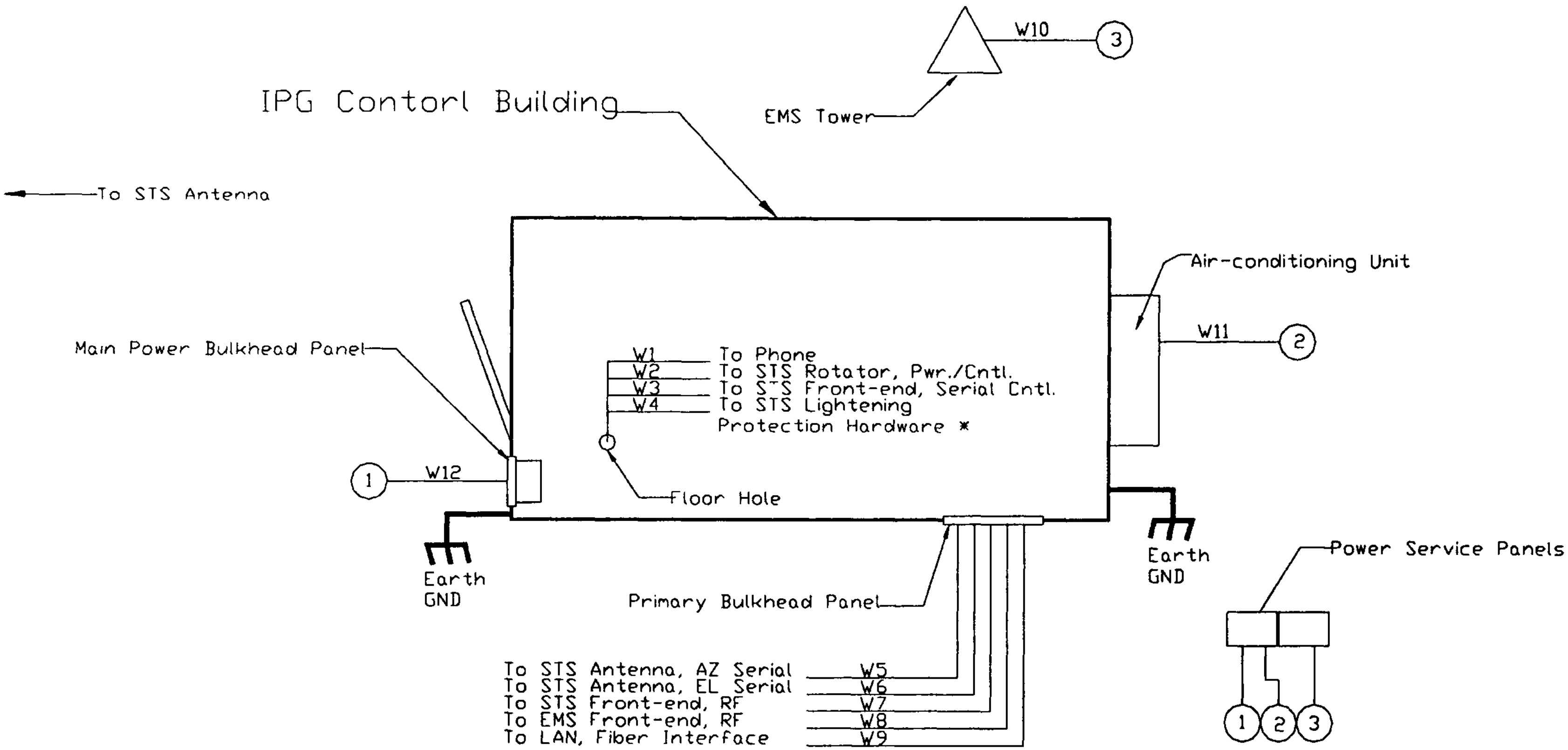
1. Insure all cables (communications and RF) that pass thru the primary bulkhead panel are grounded to the panel. This may require scraping paint.
2. Insure that all bulkhead panels are grounded to the building. Again, the may require scraping paint.
3. A thin wire grid should place above the three existing lightening rods on top of the EMS tower. A low resistance connection must be made from this grid to the existing lightening rods.

The remaining two recommendations provide protects equipment and **personnel**; therefore, these two recommendations should take the highest priority.

1. A safety ground should be available inside the building and especially to the equipment rack.
2. Add a grid to each earth ground connection on the building. The grid should be buried 4 to 6 inches underground. Ideally the grid would cover a 4 to 6 foot perimeter around the building and EMS tower.

The STS Antenna subsystem was only evaluated to insure that mount was properly grounded.

**FIG. 1 – IPG Facilities Cabling Drawing (prior to lightening protection modifications)**



## Table 1 – IPG Facilities Cable List (with Lightening Protection Recommendations)

| Reference Designator | Description | Current Protection | Recommended Modification |
|---|---|---|---|
| W1 | Phone Line | not adequate | Move cable from floor hole to the bulkhead panel; Add surge suppressor at the bulkhead panel; |
| W2 | Rotator Control Cable (EMS) | not adequate | Move cable from floor hole to the bulkhead panel; use 100% metal shielded conduit to route cable to a point as close as possible to the rotator; Terminate conduit in a metal junction box; Add surge suppressor in the junction box; |
| W3 | Serial Cable, front-end (EMS) | not adequate | Move cable from floor hole to the bulkhead panel; use 100% metal shielded conduit to route cable to a point as close as possible to the front-end; Terminate conduit in a metal junction box; Add surge suppressor in the junction box; |
| W4 | Power, lightening protection hardware (EMS) | not adequate | Hardware no longer in use; cable will be removed |
| W5 | Serial Cable, Mount Control, AZ (STS) | not adequate | Terminate cable at bulkhead panel; ground shields to bulkhead panel |
| W6 | Serial Cable, Mount Control, EL (STS) | not adequate | Terminate cable at bulkhead panel; ground shields to bulkhead panel |
| W7 | RF Cable (STS) | not adequate | Terminate cable at bulkhead panel; |
| W8 | RF Cable (EMS) | **Adequate** | N/A (should determine the nature of the component attached between the outside of the bulkhead panel and the cable) |
| W9 | Fiber Pair (LAN) | **Adequate** | N/A (does not appear to have metal shield, but requires opening the termination box to confirm) |
| W10 | Power, Front-end (EMS) | not adequate | Insure existing conduit is metal and 100% shielded; Insure flex-conduit section is making continuity with the rigid-conduit section |
| W11 | AC Power to Air-conditioning Unit | not adequate | Insure existing conduit is metal and 100% shielded; Insure that the air-conditioning unit housing has continuity to the building and to earth ground; this power drop should be isolated from the main power drop to the building (W12) by an isolation transformer to keep air-conditioner noise and surge transients from entering the building. |
| W12 | AC Power to Building | not adequate | Insure the power connection to the building is 100% shielded and that the shield is terminated at the bulkhead panel; Insure the bulkhead panel has continuity to earth ground; Add hybrid surge suppression modules in the panel box just behind the power bulkhead panel; Add isolation transformer between the power entry to the building and the power source. |

## 3.1.3 STS Antenna/feed Calibration and Pattern Measurement

**Introduction**

The purpose of this memo is to outline the test procedure for the STS antenna and feed assembly. The antenna and feed assembly will be tested to determine its gain for L-Band (1-2 GHz) missions as well as its relative gain as a function of pointing angle.

After evaluating typical procedures for antenna gain and pattern test, an approach was selected is one that will minimize errors in the measurement. Antenna test ranges throughout the antenna test and measurement industry use the same approach. The method is called Gain Transfer and it is the most frequently used technique.

**Theoretical Approach**

The device under test (DUT), in this case the STS antenna and feed assembly, will be located a distance *R* from a transmitting antenna and a signal source as shown in FIG. 1. The distance *R* is such that the spherical phase front deviation across the DUT aperture is held to λ/16, and this distance is given by the approximation:

$$R \geq \frac{2D^2}{\lambda} \quad (1)$$

where: D is the diameter of the receive antenna (3.08 m) and, λ is the wavelength of the highest operational frequency (2.0 GHz)

Under these conditions, the range *R* will be greater than 415 ft.

A known gain standard who's calibration is traceable to NIST will be used to determine the gain of the DUT. The gain standard will be mounted on the back of the DUT as shown in FIG. 1. This configuration allows the DUT and gain standard to have the same phase center axis. The general form for the signal at the receiver is expressed as:

$$P_{rx} = P_{out} \cdot G_{tx} \cdot G_{rx} \cdot \left(\frac{\lambda}{4\pi R}\right)^2 \quad (2)$$ in linear form,

where: $P_{rx}$ is the signal available at the output terminals of the receive antenna

$P_{out}$ is the sign level injected into the transmit antenna,

$G_{tx}$ is the gain of transmit antenna,

$G_{rx}$ is the gain of the receive antenna, and

$\left(\frac{\lambda}{4\pi R}\right)^2$ accounts for the spreading loss and relationship between gain and effective aperture.

And in log form as:

$$P_{rx(dB)} = P_{out} + G_{tx} + G_{rx} - 20\log\left(\frac{4\pi R}{\lambda}\right) \quad (3)$$

If we let $S_r = 2\log\left(\frac{4\pi R}{\lambda}\right)$, then the signal received by the DUT is expressed as:

$$P_{DUT} = P_{out} + G_{tx} + G_{DUT} - S_r \quad (4)$$

where: $P_{DUT}$ is the signal available at the output terminals of the receive antenna

$P_{out}$ is the sign level injected into the transmit antenna,

$G_{tx}$ is the gain of transmit antenna, and

$G_{DUT}$ is the gain of the receive antenna

The signal received by the gain standard is expressed as:

$$P_{STD} = P_{out} + G_{tx} + G_{STD} - S_r \quad (5)$$

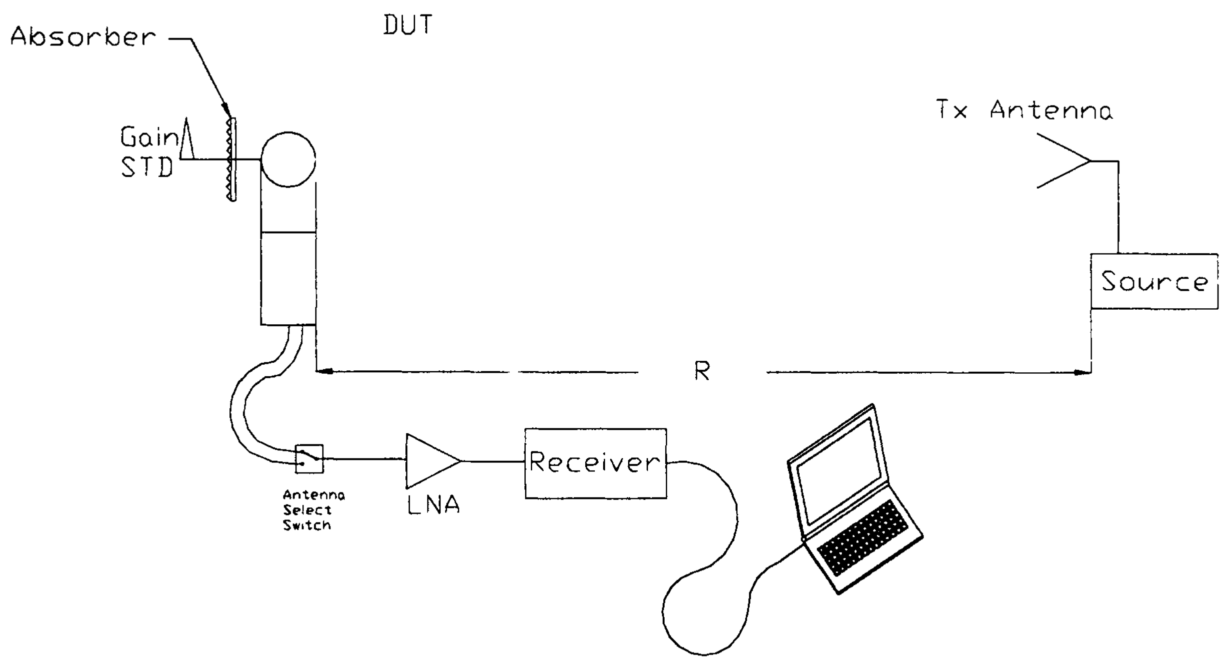
where: $P_{STD}$ is the signal available at the output terminals of the gain standard
$P_{out}$ is the sign level injected into the transmit antenna,
$G_{tx}$ is the gain of transmit antenna, and
$G_{STD}$ is the gain of the receive antenna

Since $P_{out}$, $G_{tx}$ and $S_r$ are all the same, subtracting 5 from 4 will yield;

$$G_{DUT} = (P_{DUT} - P_{STD}) + G_{STD} \quad (6)$$

Since the gain of the standard is known, the technique simply requires measuring $P_{DUT}$ and $P_{STD}$ then subtracting the two values.

FIG. 1 – Test Setup




**Issues:**

- An attenuator will be placed at the input to the spectrum analyzer when measuring $P_{DUT}$. This attenuator will reduce possible errors caused by non-linearity in the analyzer over the expected 30-40 dB gain of the DUT over the standard.

**Equipment List:**

1. Spectrum Analyzer
2. Computer
3. Gain Standard
4. Transmit antenna
5. Signal Source
6. Lumber: 2" x 4" x 8', 3' x 3' x 1/8" thk plywood
7. 30 dB attenuator (calibrated); misc. RF adapters

## 3.1.4 STS Calibration files

Ryan and Ed met to discuss requirements for amplitude calibration of the STS. We also discussed options for implementation of the calibration. The following is the result of those discussions.

**Concepts of STS Amplitude Calibration**
We discussed having four cal files one for each of the following four components:

(1) Receiver/Spectrum Analyzer, (2) Cable Assembly, (3) Receiver Front-end, (4) Antenna/Feed Assembly.

Each of the devices will be calibrated independently as the STS is re-assembled. The calibration consists of sweeping each device from 1 to 2 GHz, and recording its gain or loss as a function of frequency. All calibration will be referenced to a single known piece of test equipment if possible. The data from the calibration of each device will be stored in a file that we will call a "Frequency Table", because all data is in the table is referenced by frequency. There will be four Frequency Tables – one for each device.
Since the calibration of STS data requires no real signal processing functions, we intend to perform calibration of raw receiver data on the STS just before the data is stored to disk. It is typical in test instruments to calibrated or normalize data before it is stored or displayed. We are aware that some instruments such as the VLA do not calibrate data in real-time, but in the STS case we feel it is appropriate.
There are typically three compelling reasons not to perform real-time calibrations:

1. It requires technology not yet available or the technology is too expensive for the project.
2. There are currently available, or expected to be in the future, alternative methods for calibration that could be used, and one does not want to limit the calibration process.
3. The calibration algorithm is complex, has not been fully test, and/or has a high probability of error, so using it could produce errors in the calibration that are not recoverable if raw data is not stored.

At least one if not all of these reasons most likely apply to the first light operating conditions of the VLA, but none of these apply to the simple process of calibration of the STS.

**Proposed Calibration Process**
The following scenario describes the operation STS. The first step in the STS data processing will be to assign a specific frequency to each data point in a sweep as the data point is stored in local memory. Each data point will then have two values: frequency and amplitude (dBm).
The next step is to look up the frequency of each data point in the frequency tables. All four frequency tables must have entries at the same frequency. For example, the 19$^{th}$ element in each of the four frequency tables must have been measured at 1.192GHz. If the frequency of a given data point does not exactly match the frequencies listed in the frequency table, an algorithm will determine which cal frequency is closest to the measured data point. The closest cal frequency will then be used from each of the four tables. The process is completed by adding or subtracting the appropriate value in each of the four frequency tables to or from the amplitude recorded by the spectrum analyzer. The frequency and calibrated amplitude is then written to disk.
The frequency tables should be updated as necessary, perhaps annually, for those components subject change over time. The front-end amplifiers and the spectrum analyzer are the most likely to change over time. Obviously if any of the four calibrated components are changed out or modified the components will have to be calibrated and new frequency tables generated. Since the frequency tables are stored separately and since there is one for each device, re-calibrating components and updating frequency tables will be a simple process.

**Limiting Overall Error in STS Data Calibration**
The discussion between Ryan and I then turned to methods of calibration restricted by maintaining user flexibility and limiting the size of the Frequency Tables all the while maintaining accuracy in the calibration process. The spectrum analyzer turned out to be the source on not only the user flexibility but also the primary source of potential error and limitations.
The most obvious error in the STS calibration process is associated with assigning calibration values to points that do not exactly match the frequencies in the frequency tables. This error could be sufficiently

reduced by making multiple narrow span sweeps of each component to create a higher resolution frequency table. The problems with this approach are:

1. Determining how small the span should be may require many addition measurements.
2. Actually collecting the data for all four components will take a considerable amount of time.
3. If for instance the desired resolution is 10 kHz, the frequency tables become very large - in this case 400,000 points.

An alternative approach is to collect only 481 points from 1-2 GHz of calibration data from each device. The number of points matches the limiting device (spectrum analyzer). The calibration data from each device will be analyzed to determine which device has the largest variation in magnitude across a given bandwidth. The data from that device will be used to determine the overall STS calibration error. If this error is acceptable, 481 points in the frequency tables will be sufficient. If the error is unacceptable, efforts will be made to recalibrate each device with high frequency resolution before the end of the semester. From the data we have collected to date, 481 points appear to be sufficient.

## Additional Error Sources and Mitigation

From the front panel of the spectrum analyzer, the user has control of the center frequency, the RF bandwidth (span), the resolution bandwidth, the sweep time across the span, and video filtering. Regardless of the selected span the analyzer only provides 481 points evenly spaces of digitized samples. Since most spectrum analyzers, especially older units, are average power devices it is not possible to miss a true signal in the digitized assuming ample SNR. This has been confirmed on the STS receiver's spectrum analyzer through lab tests. Filtering before digitization provides analog integration, this will store the energy since the last sample. Although the signal is not lost, it may be distorted in digital form.
The first type of distortion is frequency errors. Since the energy is stored between samples the frequency resolution between samples is now lost. The associated error is simply:

$$F_{error} = \frac{BW_{span}}{N_{pts}}$$

where; $BW_{span}$ is the span bandwidth, and

$N_{pts}$ is the number of points (481)

Simply reducing the span reduces the error. Assuming the full span operating condition of the current STS of 1-2 GHz, the worst case frequency error is 2.079 MHz.
The second type of error is amplitude, and it is related to the sweep time across the span. The error manifests itself in the response time of the filters. If sample-and-hold circuitry is used in the spectrum analyzer, this can also contribute to amplitude error that is a function of the sweep time. It is extremely difficult to predict the error with out knowing the inner working of the analyzer. This type of error is also difficult to test in the lab due to the large set of permutations to test. To mitigate the errors we recommend the following course of action:

4. Make following statement in the user manual for STS : "As the spectrum analyzer manufacturer suggests, all parameters should be selected such that they are defined as coupled. If measurements are made with parameters uncoupled, unknown errors may be introduced in the measurement."
5. With time permitting, we will evaluate the amplitude error at various coupled parameters sets. Depending on the results of the measurements, we may make additional recommendations.

### 3.1.5 Calibration Technique for Signal in Receiver

The signal $S_R$ available at the input to theLNA is given by:

$$S_R = S \cdot A_{ef} \quad (1)$$

Where: $S$ (W*m$^{-2}$) is the signal irradiance (flux density) at the antenna
$A_{ef}$(m$^2$) is the effective aperture of the antenna

The effective aperture of the antenna is given by:

$$A_{ef} = \frac{G_R \cdot \lambda^2}{4\pi} \quad (2)$$ where: $G_R$ (dB) is the antenna gain

Substituting equation two into one yields:

$$S_R = \frac{S \cdot G_R \cdot \lambda^2}{4\pi} \quad (3)$$

Solving for $S$ gives the signal irradiance at the antenna.

$$S = \frac{S_R \cdot 4\pi}{G_R \cdot \lambda^2} \quad (4)$$

The signal power $S_R$ in equation (4) above can be derived in decibels by adding the appropriate calibration factor to the signal measured by the spectrum analyzer $S_{SA}$.

$$S_R(dBW) = S_{SA} + C \quad (5)$$

The Calibration term $C$ at a single frequency is given in decibels by:

$$C = C_{SA} + C_{cable} + C_{LNA} + K \quad (6)$$

where: $C_{SA}$ (dB) is the calibration constant for the spectrum analyzer (+/- dB)
$C_{cable}$ (dB) is the calibrated cable loss (always + dB)
$C_{LNA}$ (dB)is the calibrated LNA gain (always – dB)
$K$= - 30 dB is constant used to convert between dBm and dBW

Since the antenna gain can also be measured, it will also be added to the calibration table. Therefore, equation (5) can be written as:

$$10 \cdot \log \frac{S_R}{G_R} = S_{SA} + C_{SA} + K + C_{cable} + C_{LNA} + C_{ANT} \quad (7)$$

where: $C_{ANT}$ is the antenna gain (- dB in cal table)

The calibration constants must be measured at multiple frequencies to develop a calibration table. If we let

$$C_n = C_{SA_n} + K + C_{cable_n} + C_{LNA_n} + C_{ANT_n} \quad (8)$$ represent a single entry for a single frequency in the frequency table, then equation (7) can be rewritten as:

$$10 \cdot \log \frac{S_R}{G_R}(f_n) = S_{SA_n} + C_n \quad (9)$$

Finally equation (4) can be written as follows:

$$S(f_n) = 10 \cdot \log \frac{S_R \cdot 4\pi}{G_R \cdot \lambda^2} = S_{SA_n} + C_n + 10 \cdot \log(4\pi) - 20\log(\lambda) \quad {}_{(10)} \quad dBW \bullet m^{-2}$$

$EIRP = S(f_n) + A_s$     where: $A_s$ is equal to 10 log ($4\pi$*$R^2$)

Therefore,

$$EIRP = S(f_n) + 10 \cdot \log(4\pi \cdot R^2) \quad {}_{(12)} \quad where: 500km \leq R \leq 40Mm$$

## 3.1.6 Dynamic Data Exchange (DDE) Specifications for Nova for Windows

Nova for Windows (NfW can act as a DDE server.

The program can supply satellite tracking and range-rate data to any standard DDE client application; this information may be used to implement customized interfaces for antenna tracking or radio tuning.

Following are general specification for DDE as implemented in NfW. Note that all names and strings are case sensitive.

DDE Server name : NFW_SERVER
DDE conversation : NFW_DATA

DDE is initiated when the server (NfW) receives the string 'TUNE ON'.
The AutoTracking Status Box must be displayed in NfW although autotracking does not have to be activated.

When DDE has been activated, a single string containing the following data are "transmitted" 2-10 times per second (depending on CPU speed and loading) via DDE:

SatName AZ:Azimuth EL:Elevation RR:RangeRate AH:x

where
- SatName = the current autotracking satellite name as derived from the NfW database, maximum 12 characters;
- Azimuth = current satellite azimuth, 0.1 degree precision, no sign;
- Elevation = current satellite elevation, 0.1 degree precision, signed;
- RangeRate = satellite rate of change of distance from the observer expressed as a signed floating point number in units of 1/the speed of light.
- AH = indicates whether the satellite is above the horizon. Possible values are Y and N. NOTE: This indication is in reference to the horizon table associated with the AutoTracking Observer. See AutoTracking/AutoTracking Observer from the Main Menu to fill in the horizon table. If the horizon table is not filled in, a horizon of 0 degrees is assumed.

One space separates each field; no space after (:) labels. Make no assumptions about field lengths or total string length. Values are referred to the current AutoTracking Observer in NfW at the current UTC time and date.

Sample DDE string:
Mir AZ:116.4 EL:-21.0 RR:-1.4396996895 AH:N

DopplerHz = -FreqHz * RR * (1.0/299792.458)

M.R. Owen
Northern Lights Software Associates
nlsa@nlsa.com
http://www.nlsa.com
12-18-97

## Section 3.2 – System Control Station

Dish Tracking Osculation Problem

### The Problem

A serious problem was found with the azimuth and elevation tracking while using LabVIEW. When an angle greater than 10° was sent continuously (approximately every 33ms) to either the azimuth or elevation controllers, the motors would go to the inputted angle, appear to shift the input angle one digit to the left, and proceed to the remaining digit. For example, with an input angle of 10°, the motors would go to the inputted angle, proceed to 0°, return to the inputted angle and so-on-and-so-forth. For angles greater than 99°, the same was true. As another example, we entered a desired angle of 128° again through LabVIEW. The motors went to the 128° mark, proceeded to 28°, then to 8°, and finally back to the 128° inputted angle and would do so indefinitely.

### Testing

We began testing the system to try to determine which piece of the system was causing the problem. Through various tests and by using the logic analyzer, we determined that the problem was not due to any of the computer's hardware or the software used for tracking (LabVIEW and Nova). What we saw coming from the logic analyzer was conclusive evidence that the data being sent to the BDS-5 controllers, from LabVIEW, was not corrupted in any sense. Seeing this, it was determined that the problem was in the BDS-5 controllers.

### Solution

One solution we implanted was a type of software handshaking between LabVIEW and the BDS-5 controllers. LabVIEW would only send an angle to the controllers once the controllers were ready to receive a new angle. This temporary fix still did not completely rid the oscillations that were haunting this project. Through some thorough testing, we found that setting delays in certain parts of LabVIEW stopped the oscillating problem. What we finally found was that there only needed to be a single 20ms delay in between the control V and the data in the Labview code for elevation and azimuth. We determined that the microprocessors in the BDS-5 need a certain amount of time between initiating the interrupt before sending the data.

One quick note, the software in the BDS-5 does not print any characters to the terminal. Printing anything to the terminal takes more processing time, meaning that the delay in LabVIEW will have to be set higher (approximately 10ms higher) to avoid oscillations.

**Conclusion**

It is believed through the extensive testing, the delay that was put into the LabVIEW code is the fix to make the system stable and track properly without a recurrence of the oscillating problem described above. Further calculation will be made to see what kind of satellite tracking limitations the 20ms delay will put on the system. There will be a note in the software as to not to remove the 20ms delay from the Labview code.

## Section 3.3 – Motion Control Subsystem

### 3.3.1 BDS-5 Controller Software

#### Introduction to the BDS-5 Software

The purpose of this document is to explain the BDS-5 software that is being used in the dual axis controllers.

#### 1.0 Constants Explained

To begin, the constants that were derived for the software will be explained first.
1.1 Registers

Each of the controllers have $2^{31}$ registers equivalent to 2,147,483,648 bits. So one of the most important factors that must be avoided is overflowing these registers.

Each of the controllers, both Azimuth and Elevation, have 250 user registers from X0 thru X250. So for clarity purposes, the same register numbers were used for each controller.
1.2 Input Angle

To start, X30 was used as the storage register for the input angle. Since the controllers are not able to handle non-integer values, the input angle is multiplied by 10. Referring to the code, this is implemented as X30[1]. The [] indicates the power of ten one wishes to multiply the value of the register specified. For example, X30[1] is equivalent to X30 * 10^1, X30[2] is equivalent to X30 * 10^2, X30[3] is equivalent to X30 * 10^3, and so-on-and-so-forth. Multiplying the input angle by ten enables us to handle angles to a tenth of a degree. Handling angles to a tenth of a degree was picked because Nova is only able to output angles to LabVIEW to a tenth of a degree.

### 1.3 Angle to Resolver Count Conversion

This section of the document will explain how the constant X10 was derived.

#### 1.31 Gear Ratios

The first aspect that needs to be mentioned is the different gear ratio used for elevation and azimuth control. For elevation control, a 600:1 gear ratio is used, and for azimuth control, a 350:1 gear ratio is used.

There are two separate DC brushless motors being used to drive the elevation and azimuth control. Although these motors differ in power ratings, both of them have on shaft resolver counters. So, for one revolution on the motor shaft equals 4,096 resolver counts.

##### 1.311 Azimuth

Starting with azimuth, since there is a 350:1 gear ratio, the DC motor driving azimuth will have to rotate 350 times to turn the outer gear 360°. Since the DC motor has to rotate 350 times, it will take (350 * 4096) 1,433,600 resolver counts to realize this position.

Knowing that it will take 1,433,600 resolver counts to go 360° in azimuth, we can now accurately determine how many resolver counts it will take for each degree on the outer gear. For example, to determine how many resolver counts it would take to go 45° we would do the following calculation. Since we know that 45° is 1/8 of 360° (45 / 360), and we know that 360° is equivalent to 1,433,600 resolver counts, we simply multiply 1/8 times 1,433,000 resolver counts and get 179,000 resolver counts to go 45°. Just for clarification, let's determine how many resolver counts it would take to go 180°. Since we know that 180° is half of 360°, we simply multiply 0.5 times 1,433,600. This results in 716,000 resolver counts to go 180°.

As a quick note, as mentioned above, the input angle is being multiplied by ten to handle angles to a tenth of a degree. So for example, when the user inputs 45.1°, the program sees this as 451°. To handle this, instead of dividing by 360°, we multiply that by ten to get 3600°. By doing this, 451°/3600° = 0.125277 conserves the true angle while being able to handle angles up to a tenth of a degree.

To sum up, the following equation is used to convert angles to resolver counts.

$$X10 = \frac{(\text{Desired Angle} * 10)*(\text{Gear Ratio} = 350)*(4096)}{(360° * 10)}$$

1.312 Elevation

Moving on to elevation, there is a 600:1 gear ratio. So, the DC motor must rotate 600 times in order to rotate the outer gear 360°. Rotating 600 times would take (600*4096) 2,457,600 resolver counts. So, to find how many resolver counts it would take to go 45°, you would divide 45° by 360° to get 1/8. You would then multiply 1/8 times 2,457,600 to get 307,200 resolver counts.

As it was mentioned in the azimuth section, the input angle is multiplied by ten, so again we divide by 3600° to conserve the true angle to handle angles up to a tenth of a degree. To sum up, the following equation is used to convert angles to resolver counts.

$$X10 = \frac{(\text{Desired Angle} * 10)*(\text{Gear Ratio} = 600)*(4096)}{(360° * 10)}$$

1.4 Register X0

The next register that we need to understand is register X0. The X0 register stores the unique constant number for derived for both axes.

1.41 Azimuth

X0 is a constant number that was found by realizing that part of the azimuth equation always remains constant, in particular the below part of the equation always remains constant.

$$X0 = \frac{(\text{Gear Ratio} = 350)*(4096)}{(360° * 10)} = 398.2222$$

Since the controller cannot handle 398.2222, the number is rounded down by the BDS-5 firmware, and so X0 = 398 for the azimuth controller.

1.42 Elevation

X0 is a constant number that was found by realizing that part of the elevation equation always remains constant, in particular the below part of the equation always remains a constant:

$$X0 = \underline{(\text{Gear Ratio} = 600)*(4096)} = 682.6666$$

$$(360° * 10)$$

Since the controller cannot handle 682.6666, the number is rounded up by the BDS-5 firmware,
and so X0 = 683 for the elevation controller

## 1.5 Order of Operation

As mentioned, the registers in the dual axis controllers can only handle number less than 2,147,483,648.

### 1.51 Azimuth & Elevation

Referring back to the azimuth equation to convert angle to resolver counts, if we were to solve the numerator of the equation first, and then divide, we would exceed the register limit at 149.8° because 149.8*10*350*4096 = 2,147,532,800. This value would have to be stored first before being divided by 3600. Since the above number exceeds the maximum limit to be stored in the register this is not realizable in the BDS-5 controllers.

This is where the constant stored in X0 pays off. As shown before, X0 takes the constant parts of the equation and crunches it to a three-digit number. Now we can input 360°, which is 360*10*683 = 1,432,800 resolver counts. As we can see, we do not have to worry about overflowing any of the registers in the azimuth controller.

The elevation calculations run into the same problem of overflowing the registers. To prevent this from happening, the constant value derived above from the elevation equation is used and overflowing is not a problem for angles up to 360°.

## 1.6 Error Compensator

One of the problems with using the constant numbers stored in the registers X0 is that the numbers are rounded. Since the numbers are not exact, a constant error is introduced.

Since there is a constant error between the number of resolver counts, we can multiply the final number of resolver counts times a constant number to get the correct number of resolver counts.

### 1.61 Azimuth

To find the error for azimuth, the first thing that must be done is to calculate the number of resolver counts for several angles using both methods. Once the values are found using both methods, we divide both values by each other and find the constant error value. Below are examples for clarity.

| DESIRED | ROUNDED | ERROR | ROUNDED * ERROR |
|---|---|---|---|
| $\frac{(45^\circ*10)*(350*4096)}{(360^\circ 10)} = \mathbf{179{,}200}$ | $(45^\circ*10*398) = \mathbf{179{,}100}$ | $\frac{179{,}200}{179{,}100} = \frac{\mathbf{1792}}{\mathbf{1791}}$ | **179,200** |
| $\frac{(90^\circ*10)*(350*4096)}{(360^\circ*10)} = \mathbf{358{,}400}$ | $(90^\circ*10*398) = \mathbf{358{,}200}$ | $\frac{358{,}400}{358{,}200} = \frac{\mathbf{1792}}{\mathbf{1791}}$ | **358,400** |
| $\frac{(180^\circ*10)*(350*4096)}{(360^\circ*10)} = \mathbf{716{,}800}$ | $(180^\circ*10*398) = \mathbf{716{,}400}$ | $\frac{716{,}800}{716{,}400} = \frac{\mathbf{1792}}{\mathbf{1791}}$ | **716,800** |
| $\frac{(360^\circ*10)*(350*4096)}{(360^\circ*10)} = \mathbf{1{,}433{,}600}$ | $(360^\circ*10*398) = \mathbf{1{,}432{,}800}$ | $\frac{1{,}433{,}600}{1{,}432{,}800} = \frac{\mathbf{1792}}{\mathbf{1791}}$ | **1,433,600** |

As we can see, by converting angle to resolver counts using the constant rounded value, there is a constant error in the resolver counts. This is easily fixed by multiplying the rounded value by the error constant found, which gives us the exact number of resolver counts we need in azimuth.

As a quick note, the values of the error compensator are stored in two separate registers. The error compensator denominator is stored in register X10 and the numerator value of the error compensator is stored in register X20. The reason for storing the error compensator into two separate registers instead of one register is because of the rounding method that the BDS-5 controllers use. This will be explained later in this document.

### 1.62 Elevation

To find the error for elevation, the first thing that must be done is to calculate the number of resolver counts for several angles using both methods. Once the values are found using both methods, we divide both values by each other and find the constant error value. Below are examples for clarity.

| DESIRED | ROUNDED | ERROR | ROUNDED * ERROR |
|---|---|---|---|
| $\frac{(45^\circ*10)*(600*4096)}{(360^\circ 10)} = \mathbf{307{,}200}$ | $(45^\circ*10*683) = \mathbf{307{,}350}$ | $\frac{307{,}350}{307{,}200} = \frac{\mathbf{2048}}{\mathbf{2049}}$ | **307,200** |
| $\frac{(90^\circ*10)*(600*4096)}{(360^\circ*10)} = \mathbf{614{,}400}$ | $(90^\circ*10*683) = \mathbf{614{,}700}$ | $\frac{614{,}700}{614{,}400} = \frac{\mathbf{2048}}{\mathbf{2049}}$ | **614,400** |
| $\frac{(180^\circ*10)*(600*4096)}{(360^\circ*10)} = \mathbf{1{,}228{,}800}$ | $(180^\circ*10*683) = \mathbf{1{,}229{,}400}$ | $\frac{1{,}229{,}400}{1{,}228{,}800} = \frac{\mathbf{2048}}{\mathbf{2049}}$ | **1,228,800** |
| $\underline{(360^\circ*10)*(600*4096)} = \mathbf{2{,}457{,}600}$ | $(360^\circ*10*683) = \mathbf{2{,}458{,}800}$ | $\underline{2{,}458{,}800} = \underline{\mathbf{2048}}$ | **2,457,600** |

| | | |
|---|---|---|
| (360°*10) | 2,457,600 | 2049 |

As we can see, by converting angle to resolver counts using the constant rounded value, there is a constant error in the resolver counts. This is easily fixed by multiplying the rounded value by the error constant found, which gives us the exact number of resolver counts we need for elevation. As a quick note, the values of the error compensator are stored in two separate registers. The error compensator numerator is stored in register X10 and the denominator value of the error compensator is stored in register X20. The reason for storing the error compensator into two separate registers instead of one register is because of the rounding method that the BDS-5 controller use. This will be explained later in this document.

1.7 Register X40

This register is the register which stores the final converted resolver counts. This register contains the equation to convert the desired angle to resolver counts.

1.71 Angle to Resolver Equation

The angle to resolver equation must be implemented in the following way:

$$\frac{(\text{Desired Angle} * 10) * (\text{Controller Constant})}{\text{Error Compensator Denominator}} * \text{Error Compensator Numerator}$$

As noted earlier, the BDS-5 cannot store non-integer values, and uses a rounded scheme to handle any non-integer values it may come across. For example, if a calculation comes out to be 34.4, the BDS-5 will round down the number to the nearest whole number, 34. If a calculation comes out to be 34.5, the BDS-5 will round up the number to the nearest whole number, 35.

Since the error compensators for the azimuth and elevation equal to about 1.00 and 0.99 respectively, both error compensators would round to the integer value of 1. This would defeat the purpose of even having the error compensator. So, by first dividing by the error compensator's numerator and then multiplying by the error compensator's numerator, we avoid any rounding errors and get the desired number of resolver counts to within one resolver count.

## 2.0 Handshaking

The purpose of this section will to explain the main parts of the program to implement a successful handshake with the PC.

### 2.1 The Variable$ Label and Subroutine 20$

The Variable$ label and subroutine 20$ are the areas where the BDS-5 software spends all of its processing time.
2.1.1 Variable$ Label

The Variable$ label is where the user enters the desired angle. To enter the Variable$ label, the user must enter a control V (^V).

In the Variable$ label the following tasks are performed: The input angle is stored, the previous angle is stored (for later comparison), and the input angle is converted from degrees to resolver counts.

As a quick note, the amount of time that the BDS-5 spends in the Variable$ label is about 13.1 ms for elevation and 15.08 ms for azimuth. The reason for the two different times is because that the elevation controller is running a newer version of firmware (3.0) and the azimuth is running an older version of the firmware (2.0). The amount of time it takes to complete the individual commands in the Variable$ labels were found using the table in Appendix F-2 in the users manual and on page 40 of the BDS-5 reference guide. The final values used are approximate values because the exact versions of the firmware that are in the controllers are not known exactly. For example, we know that the firmware version in the elevation controller is 3, but it is not known if it is 3.0.1 or 3.0.5, etc. To summarize, the version of the firmware affects execution times for each command. Refer to Appendix A to see execution times for each controller.

2.2 20$

The 20$ subroutine is where the BDS-5 controller will spend most of its time. The code for this subroutine may be seen in Appendix A.

The first thing that happens in this code is that the BDS-5 compares the previous angle entered to the new angle (The old angle is stored in register X100 and the new angle is stored in X40.). If these two registers are not equal to each other, then the BDS-5 moves to the new angle using the MA command. The MA will move to the specified position at the specified speed set by the ACC, DEC and VDEFAULT values.

After the MA command, a wait command is implemented. This command will wait for a specified motion profile to start before continuing to start before continuing program execution. In short, the program will not execute any other commands until the motor has come to a complete stop. This command was used to aid in the software handshaking between the PC and the controllers.

Once the motors have stopped, a # is sent to the PC to let LabVIEW know that the controller is ready for a new angle.

If the old angle is equal to the new angle, then a stop command (S) is executed and a # is sent to the PC to let LabVIEW know that the controller is already at that angle. So if

LabVIEW keeps sending the same angle over and over, the controllers ignore the same angle sent and wait for a different angle to be sent.

The program will remain in this subroutine because of the GOTO 20 command. The only time the program leaves this subroutine is when a CTRL V is issued. Once CTRL V has been entered, the program jumps into the VARIABLE$ label. Once the angle is converted to resolver counts, the program jumps out of the VARIABLE$ label and returns to the 20$ subroutine and continues to execute the commands mentioned.

The 20$ takes approximately 22.41ms to execute for the azimuth controller and 20.31ms to execute for the elevation controller. Again, these are estimates based off the BDS-5's literature provided. The execution times for each command can be referred to in Appendix A.

## 3.0 Without Handshaking

Since handshaking takes more time to implement, we had to find an alternative way to implement successful communication between the LabVIEW, we were able to eliminate handshaking all together. The only modification to eliminate handshaking was to remove some code from subroutine 20$.

### 3.1 Subroutine 20$

To get rid of the handshaking, we removed the wait command, the print commands, the stop command and the else if. Below is the code that is left in the subroutine 20$.

```
20$

IF X40 NE X100    ; COMPARES NEW ANGLE WITH OLD ANGLE
MA X40            ; NOT EQUAL, MOVE TO NEW ANGLE

GOTO 20           ; LOOP BACK UP

END               ; END SUBROUTINE
```

So, all this code now does is move to new angles and ignores all repeated angles. If a new angle is detected, the controllers move to that new angle, if the angle being requested is the same angle the controllers are sending, then the old angle is omitted. The amount of time to execute this subroutine is cut in half for each controller. Appendix B shows the new execution times for this subroutine.

# Appendix A – Elevation/Azimuth Command Execution Times With Handshaking

| Code | | Time (ms) Firmware Version 2.0.5 (Az.) | 3.0.5 (El.) |
|---|---|---|---|
| 20$ | | 1.00 | 1.03 |
| IF X40 NE X100 | ; COMPARES NEW ANGLE WITH OLD ANGLE | 2.01 | 2.05 |
| MA X40 | ; NOT EQUAL, MOVE TO NEW ANGLE | 5.00 | 4.37 |
| W 0 | ; WAIT TILL MOTOR HAS STOPPED | 2.00 | 2.05 |
| P "#" | ; ACSII FLAG TO TELL LABVIEW THAT IT THE BDS5 IS ; READY TO ACCEPT ANOTHER COMMAND | 3.50 | 3.28 |
| ELSE | ; NEW ANGLE EQUAL OLD ANGLE | n/a | n/a |
| S | ; STOP MOTOR UNTIL NEW ANGLE NOT EQUAL ;TO OLD ANGLE | 2.00 | 2.05 |
| P "#" | ; BDS5 READY FOR NEW COMMAND FROM LABVIEW | 3.50 | 3.28 |
| ENDIF | | n/a | n/a |
| GOTO 20 | ; LOOP BACK UP | 3.40 | 2.20 |
| END | | n/a | n/a |
| | **Total Time:** | **22.41** | **20.31** |

# Appendix C – Elevation Controller Code

```
;******************BASIC CODE FOR ELEVATION CONTROLLER*********************
;
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING
; 01-01-01 MODIFIED BY LENNY NOICE;
;
;GENERAL COMMENTS ABOUT CODE AND BDS5:
;
;  X30[1] [DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X0 [ELEVATION CONVERSION FACTOR: (600 * 4096)/3600]
;
;  X10 [ELEVATION NUMERATOR VALUE FOR ERROR COMPENSATION]
;
;  X20 [ELEVATION DENOMINTOR VALUE FOR ERROR COMPENSATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
; ***************************************************************
; * NOTE: THERE ARE NO PHYSICAL STOPS IN THE ELEVATION GEAR BOX  *
; ***************************************************************
;
;

POWER-UP$

;**************************************************************
;
;* THESE VARIABLES MUST BE SET MANUALLY THROUGH THE TERMINAL   *
;* MEANING WE CAN NOT SET THESE VARIABLES USING THIS SOFTWARE. *
;**************************************************************
;

ABAUD ON        ; ENABLE AUTOBAUDING
PROMPT ON       ; TURN DISPLAY PROMPTS ON
DIR ON          ; SETS THE BDS5 DIRECTION, DIR ON = CLOCKWISE DIRECTION
MULTI ON        ; ENABLE MULTI-TASKING

;************************************************************
;
;* THE FOLLOWING COMMANDS SET UP POSITION LOOP W/INTEGRATION.*
;************************************************************
;

PL ON           ; ENABLE POSITION LOOP
TQ OFF          ; ENABLES TORQUE LOOP, WHICH DISABLES VELOCITY LOOP
PROP OFF        ; DISABLES THE PROPORTIONAL VELOCITY LOOP
PLIM ON         ; ENABLES SOFTWARE TRAVEL LIMITS
                ; ALLOWS +/- 360 DEGREES OF TRAVEL,
                ; CONTROLLED WITH PMAX AND PMIN

10$             ; MAIN PROGRAM

EN              ; ENABLE CONTROL THE BDS5


X0 = 683        ; ELEVATION CONVERSION FACTOR (600 * 4096)/3600

;***************************************************************
;
;* THE BELOW TWO NUMBERS ARE USED FOR ERROR COMPENSATION. DUE TO *
;* THE BDS5'S ROUNDING SCHEME, THE NUMBER X10/X20 IS NEEDED TO   *
;* CORRECT FOR THIS ROUNDING ERROR                               *
;***************************************************************
;

X10 = 2048      ; ERROR COMPENSATOR (NUMERATOR)
```

```
X20 = 2049          ; ERROR COMPENSATOR (DENOMINTOR)

NORM 0              ; ASSUME AT 0 DEGREES (FOR TESTING ONLY)
;NORM 614400        ; INITIAL POSITION FOR ELEVATION ASSUMED TO BE
                    ; 614400 = 90 DEGREES

KP 2569             ; TUNING GAIN FOR POSITION LOOP
KPROP 5398          ; TUNE GAIN FOR PROPORTIONAL VELOCITY
                    ; LOOP. SET FOR "tune 30 3" RESPONSE
KV 9644             ; TUNING GAIN#1 FOR INTEGRATING VELOCITY LOOP
KVI 8052            ; TUNING GAIN#2 FOR INTEGRATING VELOCITY LOOP

P "!"               ; TELLS LABVIEW CONTROLLERS ARE READY TO START
P "#"               ; TELLS LABVIEW BDS5 IS READY TO RECEIVE NEW ANGLE

15$

X40 0               ; START AT 0 DEGREES (FOR TESTING ONLY)
;X40 614400         ; INITIAL START POSITION = 90 DEGREES

20$

IF X40 NE X100      ; COMPARES NEW ANGLE WITH OLD ANGLE
MA X40              ; NOT EQUAL, MOVES TO NEW ANGLE

GOTO 20             ; LOOP BACK UP
END


VARIABLE$           ; LABVIEW SENDS CTRL V TO INPUT NEW ANGLE
INPUT "" X30[1]     ; NEW POSITION

X100 = X40          ; COMPARES NEW ANGLE WITH OLD ANGLE

;**********************************************************************
;* THE BELOW EQUATION CONVERTS ANGLE TO RESOLVER COUNTS. THE SEQUENCE *
;* TO THIS EQUATION IS IMPORTANT BECAUSE IT AVOIDS OVERFLOWING THE    *
;* THE REGISTERS WHICH ARE ONLY 2^31 FOR INPUT ANLGES 0 TO 360 DEGREES *
;**********************************************************************

X40= ((X30*X0)/X20)*X10  ; SCALES DEGREES INTO RESOLVER COUNTS

END
```

# Appendix D– Azimuth Controller Code

```
;****************BASIC CODE FOR AZIMUTH CONTROLLER***************
;
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING
; 01-01-01 MODIFIED BY LENNY NOICE
;
;GENERAL COMMENTS ABOUT CODE AND BDS5:
;
;  X30[1] [DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X0 [ELEVATION CONVERSION FACTOR: (350 * 4096)/3600]
;
;  X10 [ELEVATION DENOMINTOR VALUE FOR ERROR COMPENSATION]
;
;  X20 [ELEVATION NUMERATOR VALUE FOR ERROR COMPENSATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
; ****************************************************************
; * NOTE: THERE ARE NO PHYSICAL STOPS IN THE AZIMUTH GEAR BOX    *
; ****************************************************************
;


POWER-UP$

;**************************************************************
;
;* THESE VARIABLES MUST BE SET MANUALLY THROUGH THE TERMINAL  *
;* MEANING WE CAN NOT SET THESE VARIABLES USING THIS SOFTWARE. *
;**************************************************************
;

ABAUD ON        ; TURN AUTOBAUDING ON.
PROMPT ON       ; TURN DISPLAY PROMPTS ON.
DIR ON          ; CLOCKWISE IS POSITIVE DIRECTION
MULTI ON        ; TURN MULTI-TASKING ON.

;***********************************************************
;
;* THE FOLLOWING COMMANDS SET UP POSITION LOOP W/INTEGRATION.*
;***********************************************************
;

PL ON           ; TURN POSITION LOOP ON.
TQ OFF          ; TORQUE LOOP OFF.
PROP OFF        ; PROPORTIONAL VELOCITY LOOP OFF.
PLIM ON         ; SET SOFTWARE POSITION LIMITS
                ; ALLOWS +/- 360 DEGREES OF TRAVEL,
                ; CONTROLLED WITH PMAX AND PMIN


10$             ; MAIN PROGRAM
EN              ; ENABLE THE BDS5

X0 = 398        ; ELEVATION CONVERSION FACTOR

;****************************************************************
;
;* THE BELOW TWO NUMBERS ARE USED FOR ERROR COMPENSATION. DUE TO *
;* THE BDS5'S ROUNDING SCHEME, THE NUMBER X20/X10 IS NEEDED TO  *
;* CORRECT FOR THIS ROUNDING ERROR                              *
;****************************************************************
;

X10 = 1791
X20 = 1792

NORM 0          ; ASSUME DISH PARKED POINTING DUE NORTH (0 DEGREES)
```

```
KP 5259             ; CONSTANTS PROGRAMMED FOR POWER DOWN
KPROP 5996          ; TUNING MEANS WHEN COMMNAD:
KV 13859            ; "tune 50 2" IS ISSUED TO CONTROLLER
KVI 12549           ; SO THAT THE USER DOES NOT HAVE TO.

P "!"               ; COMMUNICATION WITH HOST THAT PROGRAM IS WORKING
P "#"               ; TELLS LABVIEW BDS5 IS READY TO RECEIVE NEW ANGLE

15$
X40 0               ; INITIAL DESIRED POSITION OF 0 DEGREES

20$

IF X40 NE X100      ; COMPARES NEW ANGLE WITH OLD ANGLE
MA X40              ; NOT EQUAL, MOVE TO NEW ANGLE

GOTO 20             ; LOOP BACK UP

END


VARIABLE$           ; USER STARTS INTERRUPT WITH A CTRL V
INPUT "?" X30[1]    ; USER DEFINES NEW ANGLE TO 1/10 OF A DEGREE

X100=X40            ; STORE OLD ANGLE

;**********************************************************************
;* THE BELOW EQUATION CONVERTS ANGLE TO RESOLVER COUNTS. THE SEQUENCE  *
;* TO THIS EQUATION IS IMPORTANT BECAUSE IT AVOIDS OVERFLOWING THE     *
;* THE REGISTERS WHICH ARE ONLY 2^31 FOR INPUT ANLGES 0 TO 360 DEGREES *
;**********************************************************************

X40 = ((X30*X0)/X10)*X20

END
```

### 3.3.2 BDS-5 Controller SoftwareUPDATE NOTICE

Further analysis was done on the rounding scheme that the BDS5 controllers use. After detailed analysis, the previous equations used to convert degrees to resolver counts were deemed to be inaccurate. This document will explain the errors found and the bulletproof solution found to compensate this.

The Problem

As was noted from the previous document, the BDS5 rounds up or down to the nearest whole integer depending on the value of the decimal value. As was thought before, to fix this, all that was needed was some type of error compensator to correct for this. This was true for the angles that were shown, but when analysis was done to the tenth of degrees, the compensator was found to be useless.

For example, as we may recall, the below equation was used for the conversion:

$$\frac{(\text{Desired Angle} * 10) * (\text{Controller Constant})}{\text{Error Compensator Denominator}} * \text{Error Compensator Numerator}$$

When 45° is used, we get the following, (the values for the elevation compensator will be used for this example):

$$\frac{(45 * 10) * (683)}{2049} * 2048 = 307{,}200 \text{ resolver counts}$$

The reason this works out is because the first part of the equation divides into a nice whole integer number, 150, but when we try this same scheme with 45.1° we get:

$$\frac{(45.1 * 10) * (683)}{2049} * 2048 = 307{,}200 \text{ resolver counts}$$

In fact, we get the same number of counts, why? Well, when we do the math for the first part of the equation, we get 150.33. The BDS5 stores this as 150 before multiplying by 2048. The previous document assumed that the BDS5 would store 150.33 just long enough to be multiplied by 2048.

In fact, with this implementation, the BDS5 is only able to move every three tenths of a degree starting at 0.2°. For example, the BDS5s will not move until 0.2°, and will not move again until it sees a difference of at least 0.3° thereafter.

## The Fix

In order to fix this problem, we have to go back to the beginning. Starting with the fundamental formula that is used to convert degrees to resolver counts, we have:

$$\text{Resolver Counts} = \frac{(\text{Desired Angle} * 10)*(\text{Gear Ratio})*(4096)}{(360° * 10)}$$

As was done from the last document, we can observe that parts of the above equation will remain a constant, and we can store this constant as to eliminate calculation time.

From the above equation:

$$\frac{(\text{Gear Ratio})*(4096)}{360°*10}$$

This part of the equation is a constant for both axes. When we solve this constant for each of the axis, we get:

$$\text{Elevation} = \frac{600*4096}{360°*10} = 682.6666$$

$$\text{Azimuth} = \frac{350*4096}{360°*10} = 398.2222$$

As we can see, if we round any of these numbers, we are going to induce an error into our system, so the trick is to make each of those numbers a nice whole number.

Starting with elevation, we can easily see that 682.6666 is equivalent to 682 and 2/3 or 2048/3. So, to make this into a nice whole integer, we multiply by three, and get **2048**, this will be our new constant term for elevation which will be discussed in a bit.

Next, moving on to azimuth, we can easily see that 398.2222 is equivalent to 398 and 2/9, or simply 3584/9. Again, to make this number into a nice whole integer, we multiply this number by nine and get **3584**, which will be our new constant for azimuth.

Once we have found these constants, it's smooth sailing from here. Recall the original formula used to convert degrees to angles from above, this equation can now be reduced to the following:

$$\text{Degrees} = (\text{Desired Angle} * 10) * (\text{Axis Constant})$$

What's wrong with this equation? This conversion will be off by a constant number for each axis. As mentioned, in order to get the derived constants, we had to multiply each of the axis by a certain number. The final step for the conversion process is to divide by the number we multiplied with. The following equations are used to convert degrees to resolve counts in the elevation and azimuth controllers:

**Elevation = ((Desired Angle * 10) * (2048)) / 3**

Azimuth = ((Desired Angle * 10) * (3584)) / 9

The key to the success of the above equations is the fact that no rounding is done until the final step is completed, proving to be an effective procedure, as we will see.

**Note:**

The desired angles shown in the equations are multiplied by ten. This step was moved to LabVIEW because of the BDS5 inability to handle not integer numbers.

## Advantages with New Equations

The above equations used to convert degrees to resolver counts have three key advantages: Reduced Operations, High Accuracy, and Low Error Rate.

## Reduced Operations

Since the desired angle is already multiplied by ten in LabVIEW, there are only two operations that the BDS5 must do; one multiply and one divide. This reduces the amount of time the BDS5 has to spend on converting degrees to resolver counts. The total time it takes to convert a single angle to resolver counts is 4.6ms (found using Appendix F in BDS5 manual) using this formula compared to the 9.2ms it took for the previous equation!

## Accurate

The success of these new equations comes from how accurate they are. The below equation is the equation that will give us the exact amount of resolver counts that are needed for each angle.

$$\text{Resolver Counts} = \frac{(\text{Desired Angle} * 10) * (\text{Gear Ratio}) * (4096)}{(360^\circ * 10)}$$

The new equations that we have found produce the **EXACT** number of resolver counts from the above equation. As a quick reminder, the reason the above equation is not directly used is because that the numerator will overflow the internal registers with angles over 150° and too, there are more operations to worry about.

## Low Error Rate

As the real world would have it, errors are inevitable. You can't get around them no matter how hard you may try, but lucky for us, these errors are near insignificant as we will see.

Nearly every conversion is going to have a decimal remainder with them. As a reminder, the BDS will round down to the nearest whole integer for decimal values equal or less than 0.4, and will round up to the nearest whole integer for decimal values equal or greater than 0.5.

Thinking about this a little bit, the biggest error we are going to get is 0.5 counts. Let's say, we get a conversion of 123,456.5. The BDS5 would round this to 123,457. This is an error of 123,457 – 123,456.5 = 0.5 counts. Again, this would be the largest rounding error that could occur.

Now, when we convert 0.5 counts to how much movement that would be in the outer gears, we get

Elevation:

$$\text{Error} = \frac{360 * 10 * 0.5}{10*600*4096} = \mathbf{7.324*10^{-5}\,°}$$

Azimuth:

$$\text{Error} = \frac{360 * 10 * 0.5}{10*350*4096} = \mathbf{1.256*10^{-4}\,°}$$

As we can plainly see, having a 0.5 count error isn't going to move the outer gears a significant amount.

## Summary

The new equations found have proven to be the fix that we need for a fast and accurate conversion scheme. These equations successfully handle the rounding schemes that the BDS5 controllers use, and they handle them with errors that are next to insignificant. For clarity, the below equations are being implemented in the BDS5s.

**Elevation = ((Desired Angle * 10) * (2048)) / 3**

Azimuth = ((Desired Angle * 10) * (3584)) / 9

## Appendix A

```
;****************BASIC CODE FOR AZIMUTH CONTROLLER***************
;
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING SYSTEM
; 03-19-01 MODIFIED BY LENNY NOICE
;
; STEP COUNT CALCULATION:
;
;  X30[DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X0 [ELEVATION CONVERSION FACTOR (NUMERATOR). SEE DOCUMENTATION FOR
;     DERIVATION]
;
;  X120 [ELEVATION CONVERSION FACTOR (DENOMINATOR). SEE DOCUMENTATION FOR
;      DERIVATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
;  THE BELOW VARIABLES SET THE MOVEMENT PROFILE. THESE VALUES
;  SET AZIMUTH FOR A 10 DEGREE PER SECOND MOVEMENT
;
;  ACC = 550
;  DEC = 1000
;  VDEFAULT = 585
;
; ***************************************************************
;
; * NOTE: THERE ARE NO PHYSICAL STOPS IN THE AZIMUTH GEAR BOX   *
; ***************************************************************
;
;

POWER-UP$

ABAUD ON                        ; TURN AUTOBAUDING ON.

PROMPT ON                       ; TURN DISPLAY PROMPTS ON.
DIR ON                          ; CLOCKWISE IS POSITIVE DIRECTION
MULTI ON                        ; TURN MULTI-TASKING ON.


10$                             ; MAIN PROGRAM

; BELOW THREE VARIABLES SET UP:
; POSITION LOOP W/O INTEGRATION

PL 1                            ; TURN POSITION LOOP ON.
TQ 0                            ; TORQUE LOOP OFF.
PROP 1                          ; PROPORTIONAL VELOCITY LOOP OFF.

; BELOW THREE VARIABLES SET UP POSITIO LIMITS
; THERE WILL BE +-20 DEGREES OF FREEDOM IN EACH AXIS

PLIM ON                         ; TURN ON POSITION LIMITS
PMIN -79872                     ; MIN. LIMIT SETTING
PMAX 1517568                    ; MAX. LIMIT SETTING

EN                              ; ENABLE THE BDS5

X0 = 398996                     ; ELEVATION CONVERSION FACTOR - SEE DOCUMENTATION
X120 = 1000                     ; FOR DERIVATION OF VARIABLES

NORM 0                          ; ASSUME DISH PARKED POINTING DUE NORTH (0 DEGREES)

; THE BELOW FOUR VARIABLES SET UP THE CONSTANTS THAT RELATE
; TO THE POSITION LOOP W/O INTEGRATION
```

```
KP 5758                 ; CONSTANTS PROGRAMMED FOR POWER DOWN
KPROP 5259              ; TUNING MEANS WHEN COMMNAD:
KV 13308                ; "tune 50 2" IS ISSUED TO CONTROLLER
KVI 12549               ; SO THAT THE USER DOES NOT HAVE TO.

P "!"                   ; COMMUNICATION WITH HOST THAT PROGRAM IS WORKING
P "#"                   ; TELLS LABVIEW BDS5 IS READY TO RECEIVE NEW ANGLE

15$

X40 0                   ; INITIAL DESIRED POSITION OF 0 DEGREES (TRUE NORTH)

20$

MA X40                  ; MOVE TO ANGLE
GOTO 20                 ; LOOP BACK UP
END

VARIABLES$              ; USER STARTS INTERRUPT WITH A CTRL V

;****************************************************************
;THE INPUT ANGLE IS MULTIPLIED BY TEN IN LABVIEW. THIS WAS DONE
;SO BECAUSE THE BDS5 CANNOT HANDLE NOT INTERGER NUMBERS
;****************************************************************

INPUT "?" X30           ; USER DEFINES NEW ANGLE TO 1/10 OF A DEGREE

X40 = (X30*X0)/X120     ; CONVERTS ANGLES TO RESOLVER COUNTS

END
```

# Appendix B

```
;******************BASIC CODE FOR ELEVATION CONTROLLER*********************
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING
; 03-19-2001 MODIFIED BY LENNY NOICE
;
; STEP COUNT CALCULATION:
;
;  X30[DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X90 [ELEVATION CONVERSION FACTOR (NUMERATOR). SEE DOCUMENTATION FOR
;     DERIVATION]
;
;  X120 [ELEVATION CONVERSION FACTOR (DENOMINATOR). SEE DOCUMENTATION FOR
;      DERIVATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
;  THE BELOW VARIABLES SET THE MOVEMENT PROFILE. THESE VALUES
;  SET AZIMUTH FOR A 10 DEGREE PER SECOND MOVEMENT
;
;  ACC = 750
;  DEC = 1000
;  VDEFAULT = 1200
;
;*************************************************************
;* NOTE: THERE ARE NO PHYSICAL STOPS IN THE ELEVATION GEAR BOX  *
;*************************************************************
;

POWER-UP$

ABAUD ON                  ; ENABLE AUTOBAUDING
PROMPT ON                 ; TURN DISPLAY PROMPTS ON
DIR ON                    ; SETS THE BDS5 DIRECTION, DIR ON = CLOCKWISE DIRECTION
MULTI ON                  ; ENABLE MULTI-TASKING

10$                       ; MAIN PROGRAM

; BELOW THREE VARIABLES SET UP:
; POSITION LOOP W/O INTEGRATION

PL 1                      ; TURN POSITION LOOP ON.
TQ 0                      ; TORQUE LOOP OFF.
PROP 1                    ; PROPORTIONAL VELOCITY LOOP OFF.

; BELOW THREE VARIABLES SET UP POSITIO LIMITS
; THERE WILL BE +-5 DEGREES OF FREEDOM IN EACH AXIS

PLIM ON                   ; TURN ON POSITION LIMITS
PMIN -65195               ; MIN. LIMIT SETTING
PMAX 2412203              ; MAX. LIMIT SETTING

EN                        ; ENABLE CONTROL THE BDS5
X90 = 130355              ; ELEVATION CONVERSION FACTOR - SEE DOCUMENTATION
X120 = 100                ; FOR DERIVATION OF VARIABLES


NORM 11735196             ; INITIAL POSITION FOR ELEVATION ASSUMED TO BE
                          ; 1173196 = 90 DEGREES
```

```
; THE BELOW FOUR VARIABLES SET UP THE CONSTANTS THAT RELATE
; TO THE POSITION LOOP W/O INTEGRATION

KP 5383                 ; TUNING GAIN FOR POSITION LOOP
KPROP 14551             ; TUNE GAIN FOR PROPORTIONAL VELOCITY
                        ; LOOP. SET FOR "tune 50 3" RESPONSE
KV 31529                ; TUNING GAIN#1 FOR INTEGRATING VELOCITY LOOP
KVI 12411               ; TUNING GAIN#2 FOR INTEGRATING VELOCITY LOOP

P "!"                   ; TELLS LABVIEW CONTROLLERS ARE READY TO START


15$

X40 1173196             ; INITIAL START POSITION = 90 DEGREES

20$

MA X40                  ; MOVE TO NEW ANGLE
GOTO 20                 ; LOOP BACK UP
END


VARIABLE$               ; LABVIEW SENDS CTRL V TO INPUT NEW ANGLE

;****************************************************************
;THE INPUT ANGLE IS MULTIPLIED BY TEN IN LABVIEW. THIS WAS DONE
;SO BECAUSE THE BDS5 CANNOT HANDLE NOT INTERGER NUMBERS
;****************************************************************

INPUT "" X30            ; NEW POSITION

X40= (X30*X90)/X120     ; DEGREES TO RESOLVER COUNTS EQUATION

END
```

### 3.3.3 Azimuth and Elevation Calibration

The purpose of this document is to explain the steps taken to calibrate the azimuth and elevation motors to exact inputted angles.

## Azimuth Calibration

Referring back to the *BDS5 Software UPDATE* document, the gear ratio was assumed to be 350:1. This means for a 360° turn of the outer gear the azimuth motor would have to travel 1,433,600 resolver counts.

$$\frac{360*10*4096*350}{3600} = 1{,}433{,}600 \text{ resolver counts}$$

Once the azimuth motor was installed, the dish was positioned to the true north mark. Using Motion Link, the program in the azimuth controller was run. Once running, a desired angle of 360° was inputted. The dish rotated to the desired angle (1,433,600 resolver counts) but due to gear backlash and slop in the system, the dish stopped short of the true north mark. We then began to move the motors until the dish was aligned to the true north marking and recorded the angle and the number of resolver counts it took for the dish to reach 360°. For the dish to rotate a complete 360°, the azimuth controller needed an input angle of 360.7°. Once the angle was known and the number of resolver counts noted was 1,436,388.

Knowing how many resolver counts it takes to rotate the dish 360°, the next step was to find the new gear ratio. Also, noting how the BDS5 is known for rounding non-integer numbers, the first step taken to find the new gear ratio was to calculate the exact number of resolver counts to go 360°. Since an inputted angle of 360.7° needed to be inputted, the exact number of resolver counts that comes out to be is:

$$\frac{360.7*10*4096*351}{3600} = 1{,}436{,}387.555556$$

Knowing the exact number of resolver counts, we can use the below formula to find the new gear ratio ($G_{NR}$) to rotate the dish to 360°.

$$G_{NR} = \frac{1{,}436{,}387.5556}{3600} = 350.680556$$

Since this gear ratio cannot be used due to the BDS5s inability to handle non-integer numbers, 350.680556 is rounded up to the nearest integer 351. Rounding the gear ratio like this induces errors. For example, if we calculate the number of resolver counts it now takes to go 360°, we get the following:

$$\frac{360*10*4096*351}{3600} = 1{,}437{,}696 \text{ resolver counts}$$

This ends up being 1,436,388 – 1,437,696 = 1,308 resolver counts off the mark, which ends up being 0.327524103° of error. A complete list of errors per tenth of a degree can be seen in Appendix A, which also shows a complete error data base for each degree the azimuth will encounter.

If we divide the desired number of resolver counts (desired number of resolver counts are those calculated with the gear ratio as 350.680556) by the actual number of resolver counts (actual number of resolver counts are those calculated with the gear ratio as 351) we get a constant quotient between these angles. So, to handle some of the resolver count error we take this quotient which ends up being approximately 0.99909 (this number can be found in the data file in Appendix A) and multiply that number to the number of resolver counts determined using the 351 gear ratio. The key problem with this is that the BDS5s cannot handle the non-integer number. This is easily handled by making 0.99908 two-integer numbers, simply 99,908/100,000. If we first multiple the actual number of resolver counts by 99908 and then divide that number by 100,000 we can very accurately regain the desired number of resolver counts needed. For example, let's convert 45° to resolver counts using 350.6805556 as the gear ratio, using 351 as the gear ratio, and using 351 as the gear ratio with the error compensator.

DESIRED RESOLVER COUNTS:

$$\frac{45*10*4096*350.6805556}{3600} = 179{,}548.5 \text{ resolver counts}$$

ACTUAL RESOLVER COUNTS

$$\frac{45*10*4096*351}{3600} = 179{,}712 \text{ resolver counts}$$

Difference of 179,712 - 179,548.5 = 163.5 resolver counts

RESOLVER COUNTS WITH COMPENSATOR

$$\frac{179{,}712 * 99{,}908}{100{,}000} \cong 179{,}547 \text{ resolver counts}$$

Difference of 179,548.5 - 179,547 = 1.5 resolver counts

With this scheme of multiplying by an error compensator, we reduce the resolver count error significantly. The only problem with this is that the BDS5 can only handle numbers $2^{31} - 1 = 2{,}147{,}483{,}647$ big. Multiplying the final number of resolver counts by the error compensators numerator would exceed the $2^{31} - 1$ limit. So, the trick is to be able to multiply these set of numbers such that the registers never over flow. The below formula is the final equation that will be used:

$$\frac{\text{DESIRED ANGLE} * 4096 * \text{GEAR RATIO} * \text{ERROR COMPENSATOR NUMERATOR}}{360\ \text{DEGREES} * 10 * \text{ERROR COMPENSATOR DENOMINATOR}}$$

Looking at the above equation we can see that all but the DESIRED ANGLE variable remains constant. Multiplying the constants in the numerator and the denominator and simplifying we get the following:

$$\frac{398{,}996}{1000}$$

Once we have this number, we get the following formula to accurately convert angles to resolver counts:

**DESIRED ANGLE * 398,996/1000**

NOTE: DESIRED ANGLE = INPUT ANGLE * 10

With this new equation, we get the simplicity and high accuracy needed for tracking in the azimuth direction. Refer to Appendix A for a complete list of each angle with its associated errors.

## Elevation Calibration

Referring back to the *BDS5 Soft UPDATE* document, the gear ratio for elevation was assumed to be 600:1, which would mean that for a 90° turn of the dish would equal to 1,228,800 resolver counts.

$$\frac{90*10*4096*600}{1800} = 1{,}228{,}800$$

Once the elevation motor was installed, the dish was positioned at zenith. To position the dish at zenith, we used a high accuracy level. Once positioned, we enabled the elevation controller through Motion Link. We then proceeded to manually move the dish from zenith to 0°. The 0° position was found again by using the same high accuracy level. Once we were at the 0° position, we observed how many resolver counts it took to from zenith to 0°. This procedure was done three times and each time we got 1,173,196±1 resolver counts.

Knowing how many resolver counts it takes to rotate the dish 90°, the next step was to find the new gear ratio ($G_{NR}$).

$$G_{NR} = \frac{1{,}173{,}196 * 1800}{90*10*4096} = 572.84961$$

Since this gear ratio cannot be used due to the BDS5s inability to handle non-integer numbers, 572.84961 is rounded up to the next integer number 573. Rounding the gear ratio like this induces errors. For example, if we calculate the number or resolver counts it now takes to get to 90°, we get the following:

$$\frac{90*10*4096*573}{1800} = 1{,}173{,}504$$

This ends up being 1,173,196 – 1,173,504 = 308 resolver counts off the mark, which ends up being 0.0236216° of error. A complete list of errors per tenth of a degree can be seen in Appendix B, which also shows a complete database for each degree the elevation will encounter.

If we divide the desired number of resolver counts (desired number of resolver counts are those calculated with the gear ratio as 572.84961) by the actual number of resolver counts (actual number of resolver counts are those calculated with the gear ratio as 573) we get a constant quotient between these angles. So, to handle some of the resolver count error we take this quotient which ends up being approximately 0.99974 (this number can be found in the data file in Appendix B) and multiply that number to the number of resolver counts determined using the 573 gear ratio. The key problem with this is that the BDS5s cannot handle the non-integer number. This is easily handled by making 0.99974 two-integer numbers, simply 99,974/100,000. If we first multiple the actual number of resolver counts by 99974 and then divide that number by 100,000 we can very accurately regain the desired number of resolver counts needed. For example, let's convert 45° to resolver counts using 572.84961 as the gear ratio, using 573 as the gear ratio, and using 573 as the gear ratio with the error compensator.

DESIRED RESOLVER COUNTS:

$$\frac{45*10*4096*572.84961}{1800} = 586{,}598 \text{ resolver counts}$$

ACTUAL RESOLVER COUNTS

$$\frac{45*10*4096*573}{1800} = 586{,}752 \text{ resolver counts}$$

Difference of 586,752 – 586,598 = 154 resolver counts

RESOLVER COUNTS WITH COMPENSATOR

$$\frac{586{,}752 * 99{,}974}{100{,}000} \cong 586{,}599 \text{ resolver counts}$$

Difference of 586,599 – 586,599 = 0 resolver counts

With this scheme of multiplying by an error compensator, we reduce the resolver count error significantly. The only problem with this is that the BDS5 can only handle numbers $2^{31} - 1 = 2{,}147{,}483{,}647$ big. Multiplying the final number of resolver counts by the error compensators numerator would exceed the $2^{31} - 1$ limit. So, the trick is to be able to multiply these set of numbers such that the registers never over flow. The below formula is the final equation that will be used:

$$\frac{\text{DESIRED ANGLE} * 4096 * \text{GEAR RATIO} * \text{ERROR COMPENSATOR NUMERATOR}}{\text{360 DEGREES} * 10 * \text{ERROR COMPENSATOR DENOMINATOR}}$$

Looking at the above equation we can see that all but the DESIRED ANGLE variable remains constant. Multiplying the constants in the numerator and the denominator and simplifying we get the following:

$$\frac{130{,}355}{100}$$

Once we have this number, we get the following formula to convert angles to resolver counts:

**DESIRED ANGLE * 130,355/100**

NOTE: DESIRED ANGLE = INPUT ANGLE * 10

With this new equation, we get the simplicity and high accuracy needed for tracking in the elevation direction. Refer to Appendix B for a complete list of each angle with its associated errors.

Since Appendix A and B are lengthy, a single hard copy of these appendices will be given to Raul. Appendix C and D show how this calibrated resolver count scheme will be implemented in each of the controllers.

## Appendix C

```
;****************BASIC CODE FOR AZIMUTH CONTROLLER**************
;
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING SYSTEM
; 03-19-01 MODIFIED BY LENNY NOICE
;
; STEP COUNT CALCULATION:
;
;  X30[DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X0 [ELEVATION CONVERSION FACTOR (NUMERATOR). SEE DOCUMENTATION FOR
;     DERIVATION]
;
;  X120 [ELEVATION CONVERSION FACTOR (DENOMINATOR). SEE DOCUMENTATION FOR
;     DERIVATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
;  THE BELOW VARIABLES SET THE MOVEMENT PROFILE. THESE VALUES
;  SET AZIMUTH FOR A 10 DEGREE PER SECOND MOVEMENT
;
;  ACC = 550
;  DEC = 1000
;  VDEFAULT = 585
;
; ************************************************************
; * NOTE: THERE ARE NO PHYSICAL STOPS IN THE AZIMUTH GEAR BOX    *
; ************************************************************
;
;
POWER-UP$
```

```
ABAUD ON                    ; TURN AUTOBAUDING ON.

PROMPT ON                   ; TURN DISPLAY PROMPTS ON.
DIR ON                      ; CLOCKWISE IS POSITIVE DIRECTION
MULTI ON                    ; TURN MULTI-TASKING ON.


10$                         ; MAIN PROGRAM

; BELOW THREE VARIABLES SET UP:
; POSITION LOOP W/O INTEGRATION

PL 1                        ; TURN POSITION LOOP ON.
TQ 0                        ; TORQUE LOOP OFF.
PROP 1                      ; PROPORTIONAL VELOCITY LOOP OFF.

; BELOW THREE VARIABLES SET UP POSITIO LIMITS
; THERE WILL BE +-20 DEGREES OF FREEDOM IN EACH AXIS

PLIM ON                     ; TURN ON POSITION LIMITS
PMIN -79872                 ; MIN. LIMIT SETTING
PMAX 1517568                ; MAX. LIMIT SETTING

EN                          ; ENABLE THE BDS5

X0 = 398996                 ; ELEVATION CONVERSION FACTOR - SEE DOCUMENTATION
X120 = 1000                 ; FOR DERIVATION OF VARIABLES

NORM 0                      ; ASSUME DISH PARKED POINTING DUE NORTH (0 DEGREES)

; THE BELOW FOUR VARIABLES SET UP THE CONSTANTS THAT RELATE
; TO THE POSITION LOOP W/O INTEGRATION

KP 5758                ; CONSTANTS PROGRAMMED FOR POWER DOWN
KPROP 5259             ; TUNING MEANS WHEN COMMNAD:
KV 13308               ; "tune 50 2" IS ISSUED TO CONTROLLER
KVI 12549              ; SO THAT THE USER DOES NOT HAVE TO.

P "!"                  ; COMMUNICATION WITH HOST THAT PROGRAM IS WORKING
P "#"                  ; TELLS LABVIEW BDS5 IS READY TO RECEIVE NEW ANGLE

15$

X40 0                  ; INITIAL DESIRED POSITION OF 0 DEGREES (TRUE NORTH)

20$

MA X40                 ; MOVE TO ANGLE
GOTO 20                ; LOOP BACK UP
END


VARIABLE$               ; USER STARTS INTERRUPT WITH A CTRL V

;**************************************************************
;THE INPUT ANGLE IS MULTIPLIED BY TEN IN LABVIEW. THIS WAS DONE
;SO BECAUSE THE BDS5 CANNOT HANDLE NOT INTERGER NUMBERS
;**************************************************************

INPUT "?" X30          ; USER DEFINES NEW ANGLE TO 1/10 OF A DEGREE

X40 = (X30*X0)/X120    ; CONVERTS ANGLES TO RESOLVER COUNTS

END
```

## Appendix D

```
;*******************BASIC CODE FOR ELEVATION CONTROLLER**********************
; BDS5 CONTROL PROGRAM FOR TECOM DISH SATELLITE TRACKING
; 03-19-2001 MODIFIED BY LENNY NOICE
;
; STEP COUNT CALCULATION:
;
;  X30[DESIRED DEGREE TIMES TEN; IE 90.9 DEGREES = 909]
;
;  X90 [ELEVATION CONVERSION FACTOR (NUMERATOR). SEE DOCUMENTATION FOR
;     DERIVATION]
;
;  X120 [ELEVATION CONVERSION FACTOR (DENOMINATOR). SEE DOCUMENTATION FOR
;      DERIVATION]
;
;  2^31 IS LARGEST SIGNED VALUE THAT THE BDS5 REGISTERS CAN HANDLE
;
;  THE BELOW VARIABLES SET THE MOVEMENT PROFILE. THESE VALUES
;  SET AZIMUTH FOR A 10 DEGREE PER SECOND MOVEMENT
;
;  ACC = 750
;  DEC = 1000
;  VDEFAULT = 1200
;
; ***************************************************************
; * NOTE: THERE ARE NO PHYSICAL STOPS IN THE ELEVATION GEAR BOX  *
; ***************************************************************
;

POWER-UP$

ABAUD ON              ; ENABLE AUTOBAUDING
PROMPT ON             ; TURN DISPLAY PROMPTS ON
DIR ON                ; SETS THE BDS5 DIRECTION, DIR ON = CLOCKWISE DIRECTION
MULTI ON              ; ENABLE MULTI-TASKING

10$                   ; MAIN PROGRAM

; BELOW THREE VARIABLES SET UP:
; POSITION LOOP W/O INTEGRATION

PL 1                  ; TURN POSITION LOOP ON.
TQ 0                  ; TORQUE LOOP OFF.
PROP 1                ; PROPORTIONAL VELOCITY LOOP OFF.

; BELOW THREE VARIABLES SET UP POSITIO LIMITS
; THERE WILL BE +-5 DEGREES OF FREEDOM IN EACH AXIS

PLIM ON               ; TURN ON POSITION LIMITS
PMIN -65195           ; MIN. LIMIT SETTING
PMAX 2412203          ; MAX. LIMIT SETTING

EN                    ; ENABLE CONTROL THE BDS5
X90 = 130355          ; ELEVATION CONVERSION FACTOR - SEE DOCUMENTATION
X120 = 100            ; FOR DERIVATION OF VARIABLES


NORM 11735196         ; INITIAL POSITION FOR ELEVATION ASSUMED TO BE
                      ; 1173196 = 90 DEGREES


; THE BELOW FOUR VARIABLES SET UP THE CONSTANTS THAT RELATE
; TO THE POSITION LOOP W/O INTEGRATION
```

```
KP 5383                  ; TUNING GAIN FOR POSITION LOOP
KPROP 14551              ; TUNE GAIN FOR PROPORTIONAL VELOCITY
                         ; LOOP. SET FOR "tune 50 3" RESPONSE
KV 31529                 ; TUNING GAIN#1 FOR INTEGRATING VELOCITY LOOP
KVI 12411                ; TUNING GAIN#2 FOR INTEGRATING VELOCITY LOOP

P "!"                    ; TELLS LABVIEW CONTROLLERS ARE READY TO START


15$

X40 1173196              ; INITIAL START POSITION = 90 DEGREES

20$

MA X40                   ; MOVE TO NEW ANGLE
GOTO 20                  ; LOOP BACK UP
END


VARIABLES$               ; LABVIEW SENDS CTRL V TO INPUT NEW ANGLE

;**************************************************************
;THE INPUT ANGLE IS MULTIPLIED BY TEN IN LABVIEW. THIS WAS DONE
;SO BECAUSE THE BDS5 CANNOT HANDLE NOT INTERGER NUMBERS
;**************************************************************

INPUT "" X30             ; NEW POSITION

X40= (X30*X90)/X120      ; DEGREES TO RESOLVER COUNTS EQUATION

END
```

# 3.4 – Antenna and RF Data Collection Subsystem

3.4.1 Satellite Tracking Design Group

## HP Spectrum Analyzer Performance through GPIB Interface

(HP 853 [s/n 2404A01592] and HP 8559A [s/n 2347A07428] analyzer combo)

The characterization process used several pieces of equipment to record the values of the environment under test. The signal generator that fed into the spectrum analyzer, the frequency counter and the power meter is the Anritsu 69367B (s/n 994602) that came from the ALMA lab. The power meter set was borrowed from Bob Hayward, FE, and is the Agilent E4419B (s/n GB40201637) power meter with the HP E4413A (s/n US38482405) CW power sensor. The frequency counter was borrowed from Paul Harden in the LO/IF group and is the HP 5342A (s/n 2232A06214).

To check the attributes of the signal the signal generator's output was manually switched between the spectrum analyzer, frequency counter, and the power meter by disconnecting a common, single line of flexible coaxial cable to the various pieces of equipment.

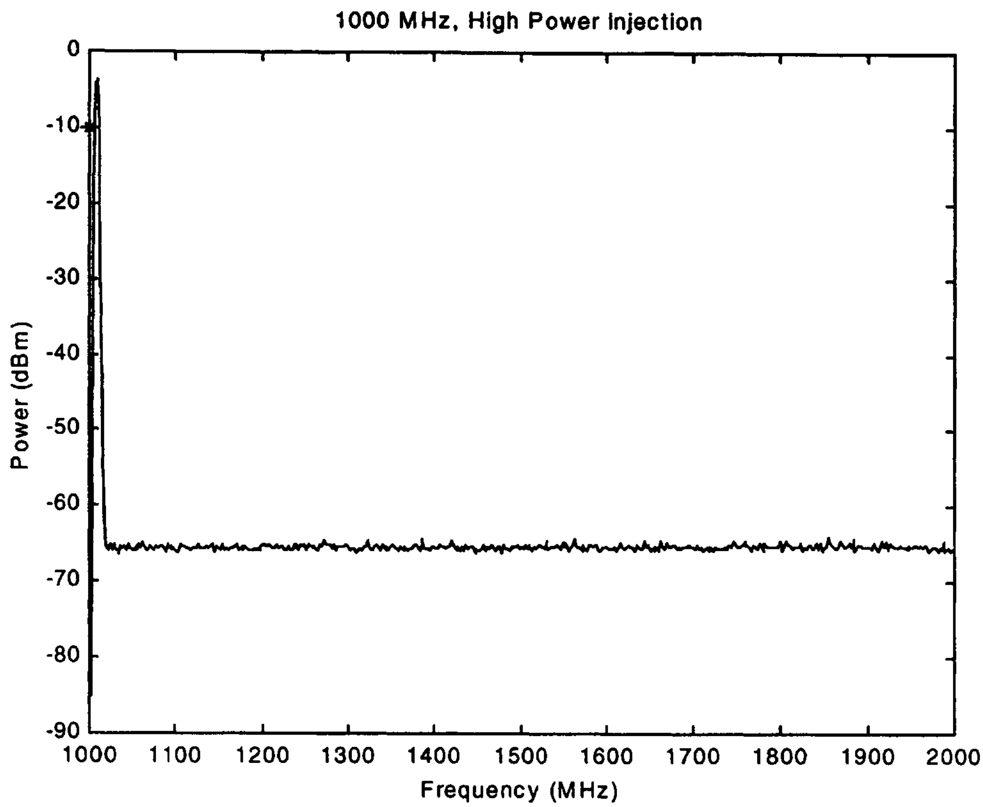
The data files were generated by communicating with the spectrum analyzer across the GPIB port from a LabVIEW code similar to the coding used to simulate a virtual front-panel.

Receiver Error

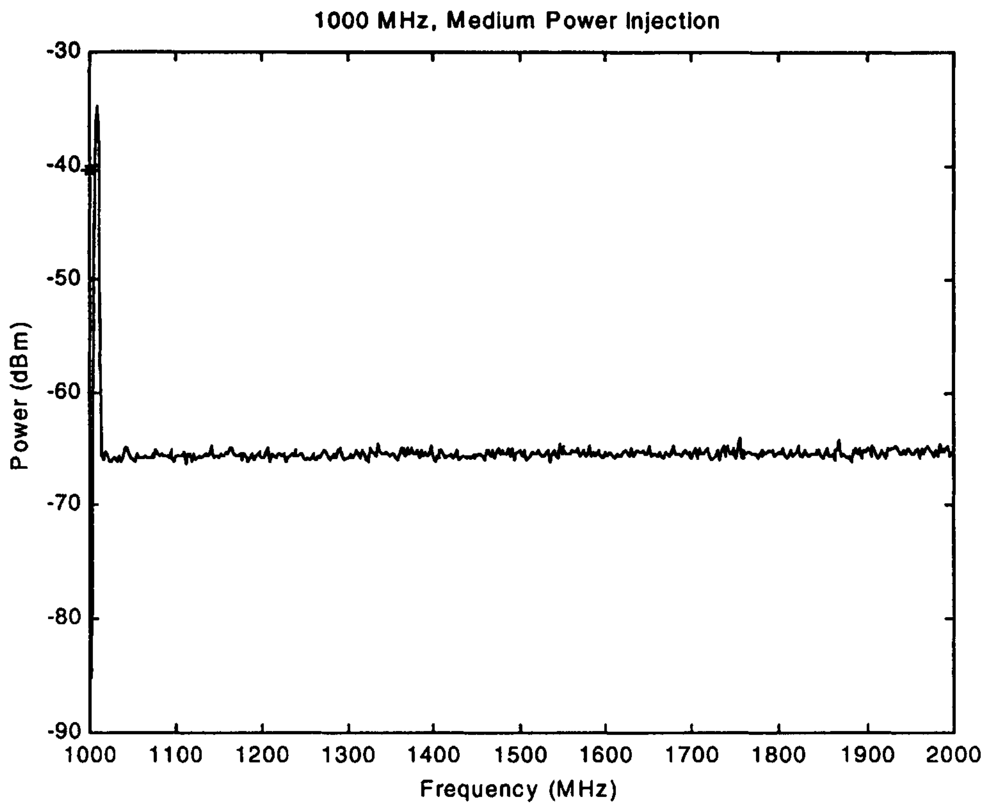
| Frequency Of Interest (MHz) | Instantaneous Frequency Error (MHz) | Instantaneous Power Error (dB) | Power Uncertainty Peak-to-Peak (dB) | Frequency Uncertanty Peak-to-Peak (MHz) |
|---|---|---|---|---|
| 1000 (Strength: Strong) | 8.3 | 6.3 | 0.3 | 0 |
| 1000 (Strength: Medium) | 8.3 | 5.6 | | |
| 1000 (Strength: Weak) | 8.3 | 4.2 | | |
| 1100 | 6.0 | 6.8 | | |
| 1200 | 5.8 | 7.0 | | |
| 1300 | 5.6 | 6.9 | | |
| 1400 | 7.5 | 6.2 | | |
| 1500 (Strength: Strong) | 7.3 | 7.2 | 0.6 & **0.7** | 2.1 & **2.1** |
| 1500 (Strength: Medium) | 7.3 | 6.0 | | |
| 1500 (Strength: Weak) | 7.3 | 5.3 | | |
| 1600 | 7.1 | 7.3 | | |
| 1700 | 8.9 | 7.1 | | |
| 1800 | 8.7 | 5.5 | | |
| 1900 (Strength: Strong) | 8.5 | 5.9 | 0.6 | 0 |
| 1900 (Strength: Medium) | 8.5 | 5.0 | | |
| 1900 (Strength: Weak) | 10.6 | 4.0 | | |
| 1991 | 9.0 | 6.8 | | |

Resolution in Frequency is 2.079 MHz and Resolution in Amplitude is 0.1 dB

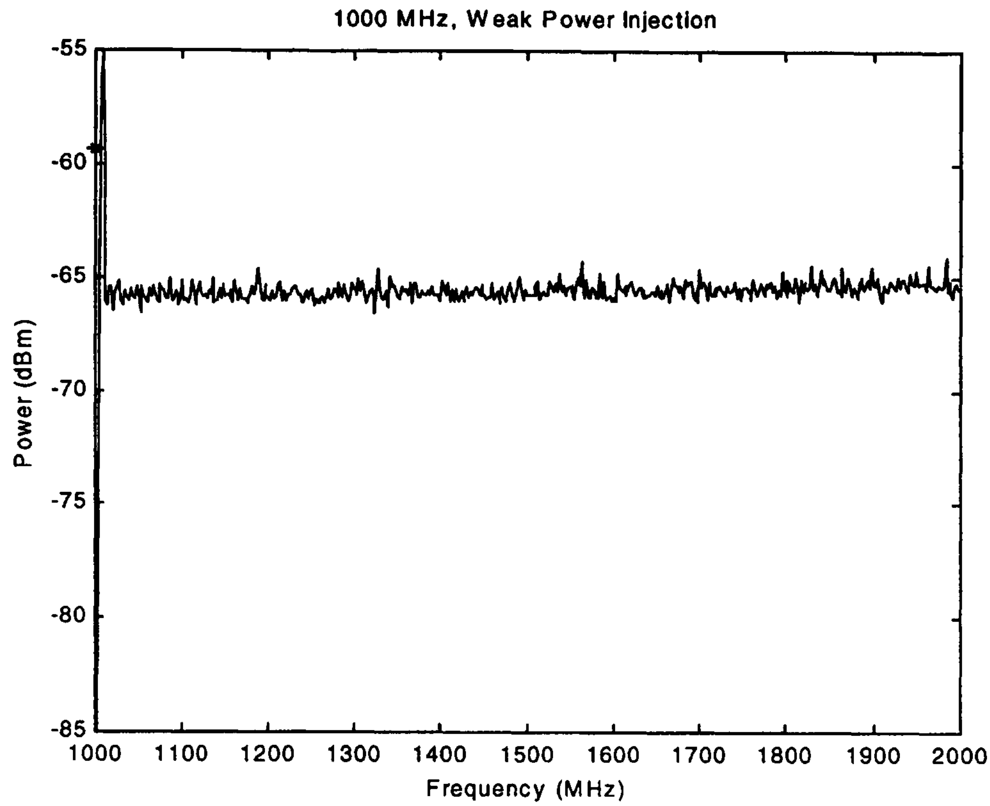
**Bold signifies the 8 hour peak-to-peak value**



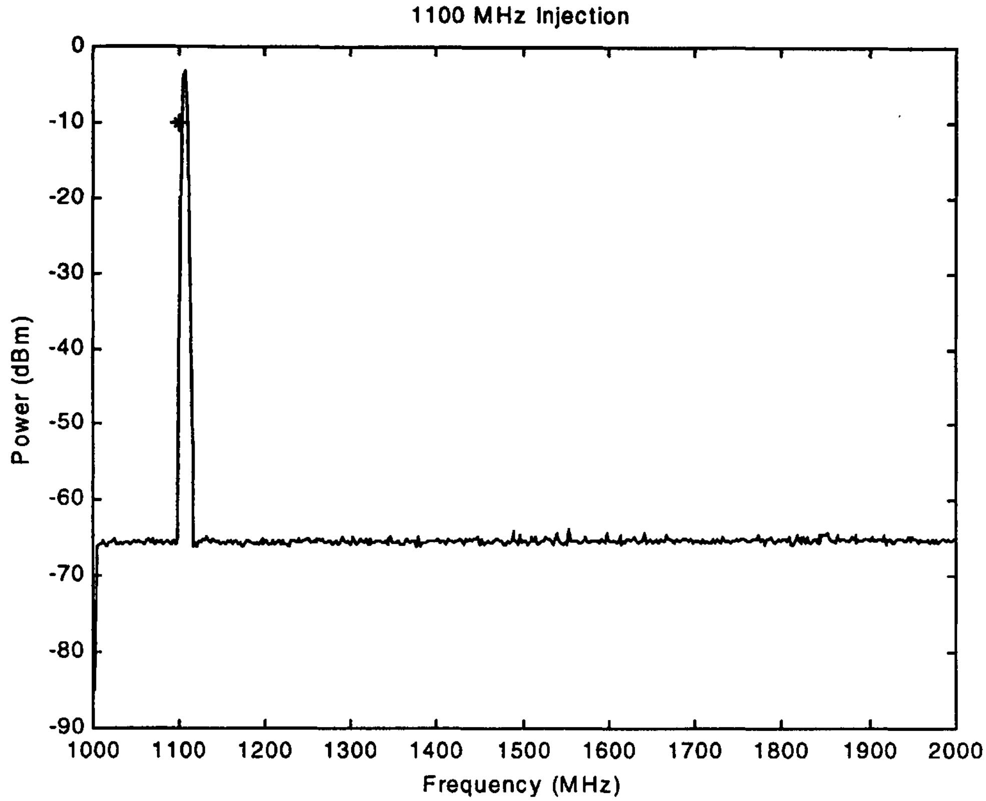

Injected Signal: -10 dBm, 1000 MHz
Detected Signal: -3.70 dBm, 1008.3 MHz



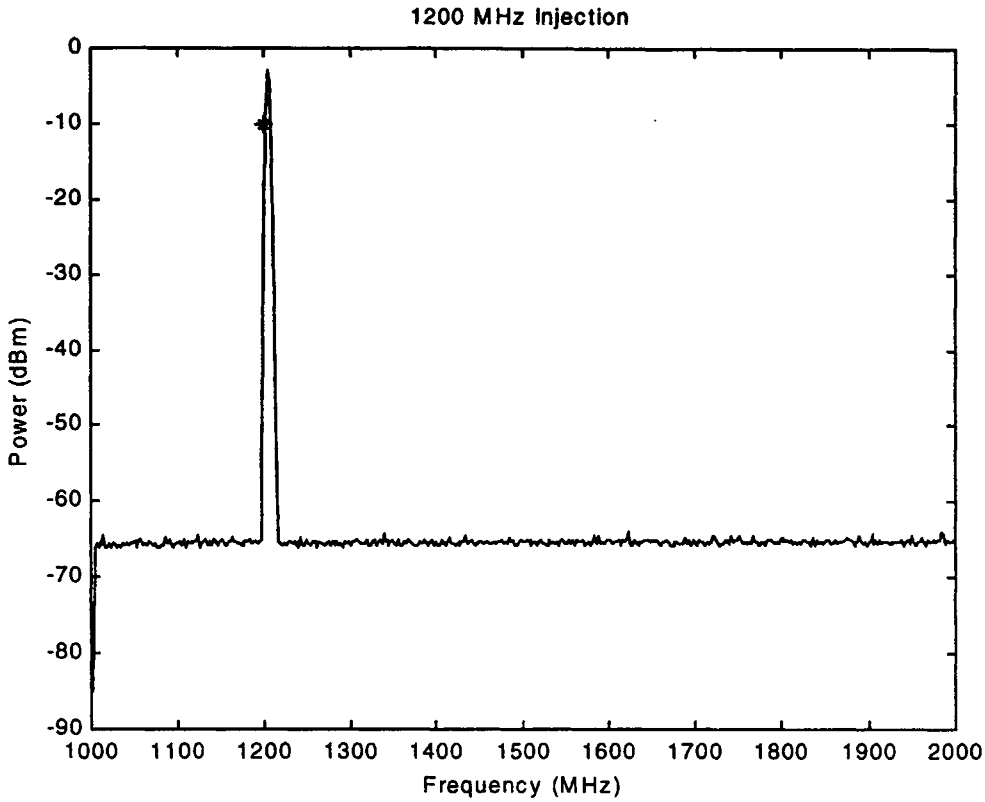

Injected Signal: -40.35 dBm, 1000 MHz
Detected Signal: -34.8 dBm, 1008.3 MHz



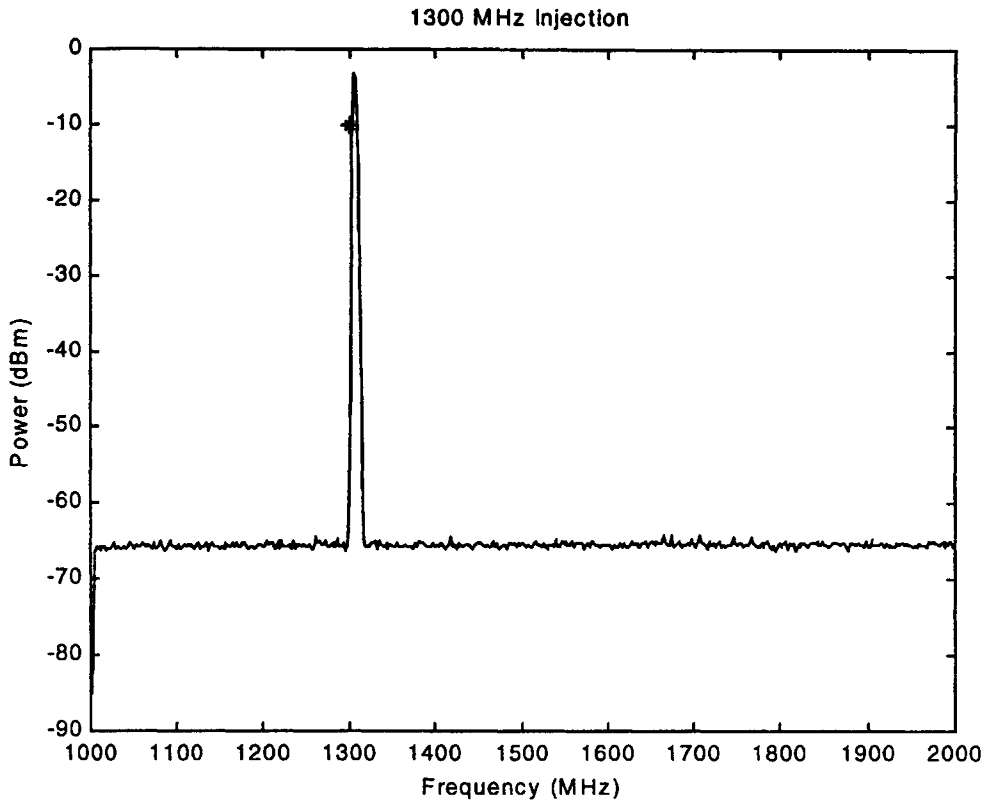

Injected Signal: -59.3 dBm, 1000 MHz
Detected Signal: -55.1 dBm, 1008.3 MHz



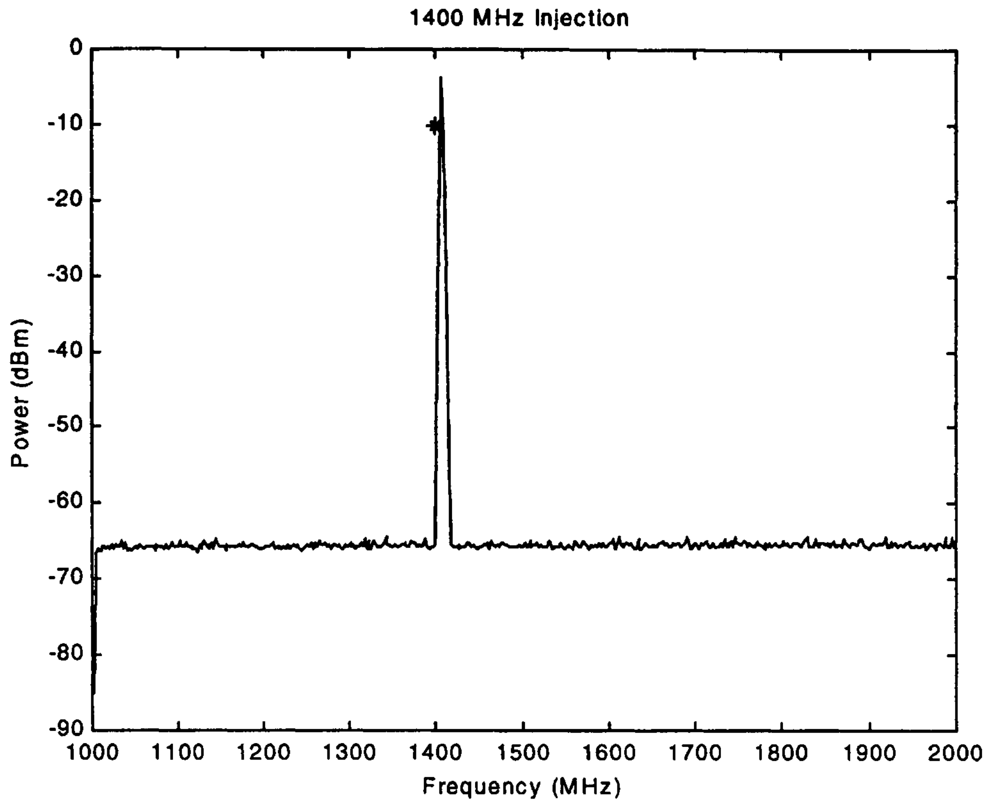

Injected Power: -10 dBm, 1100 MHz
Detected Power: -3.20 dBm, 1106.0 MHz



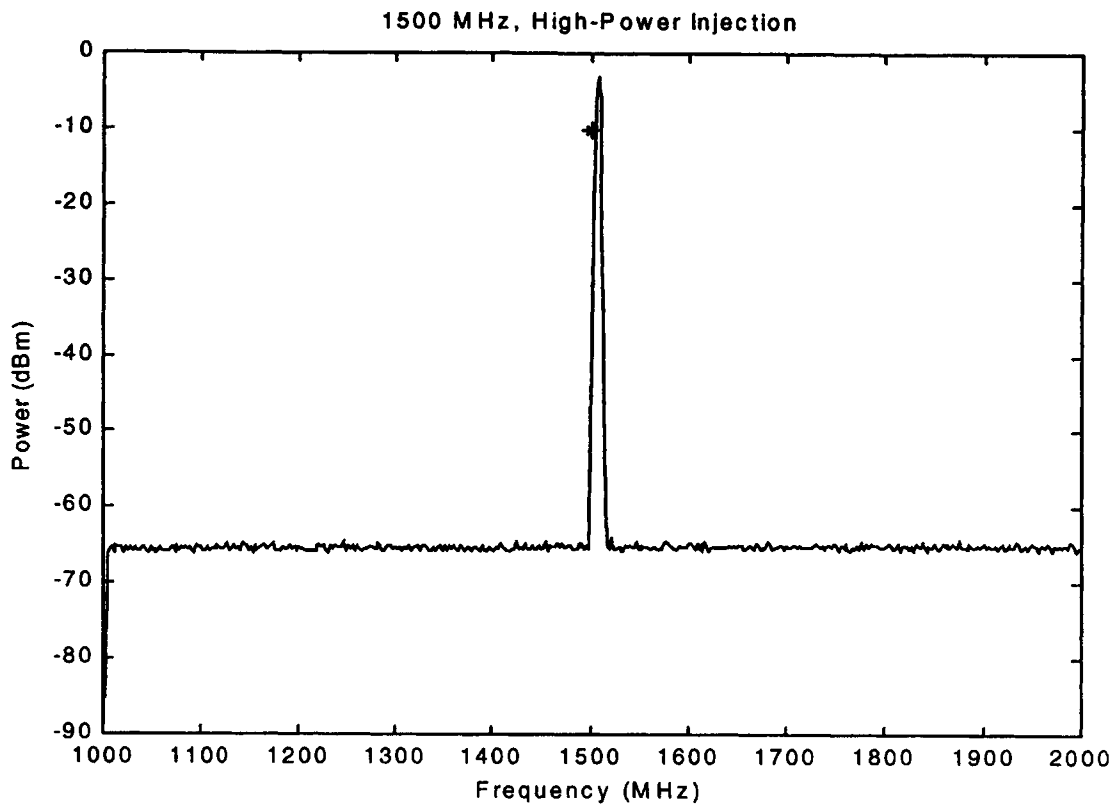

Injected Signal: -10 dBm, 1200 MHz
Detected Signal: -3.0 dBm, 1205.8 MHz



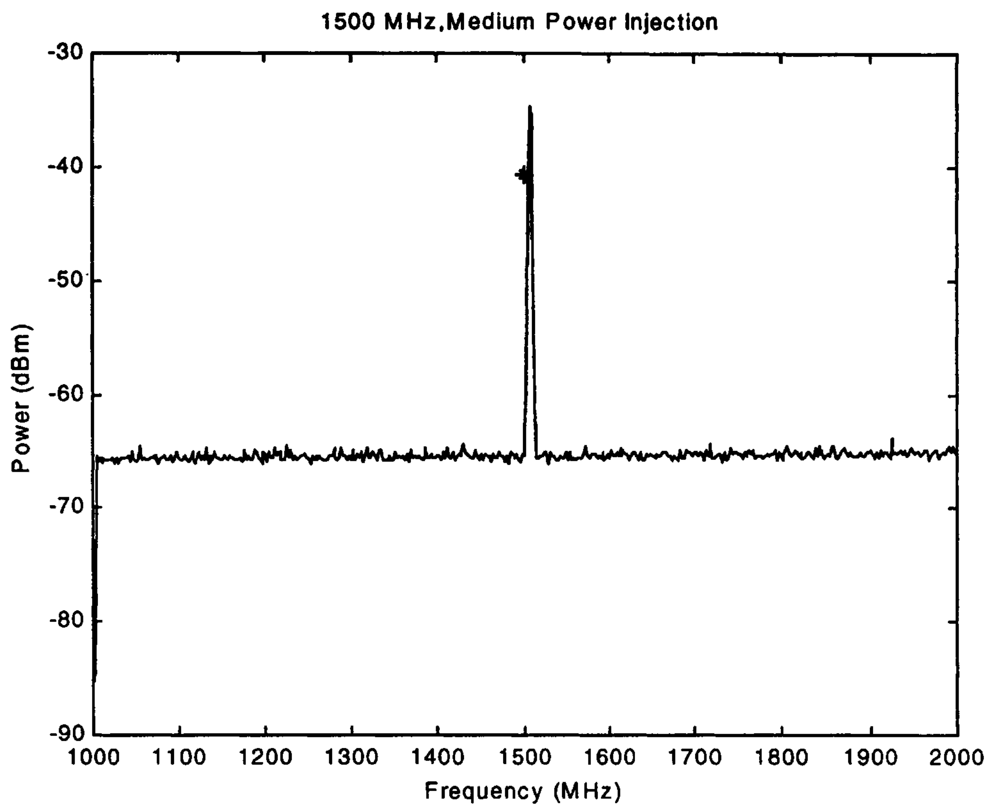

Injected Signal: -10 dBm, 1300 MHz
Detected Signal: -3.1 dBm, 1305.6 MHz



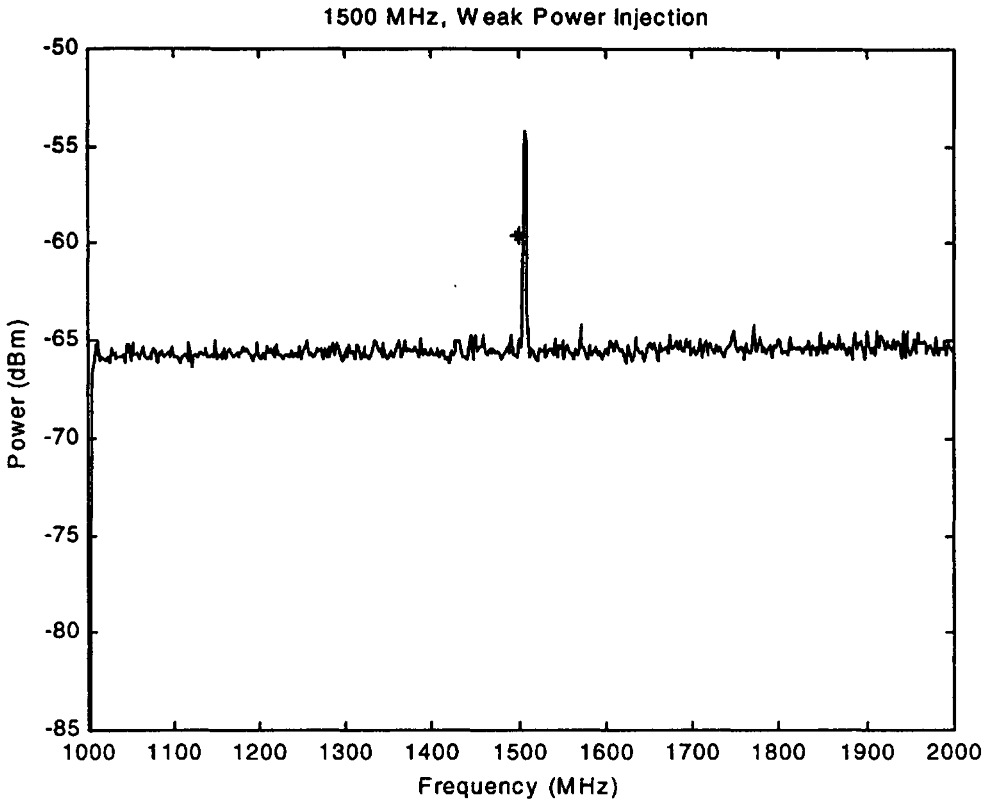

Injected Signal: -10 dBm, 1400 MHz
Detected Signal: -3.8 dBm, 1407.5 MHz



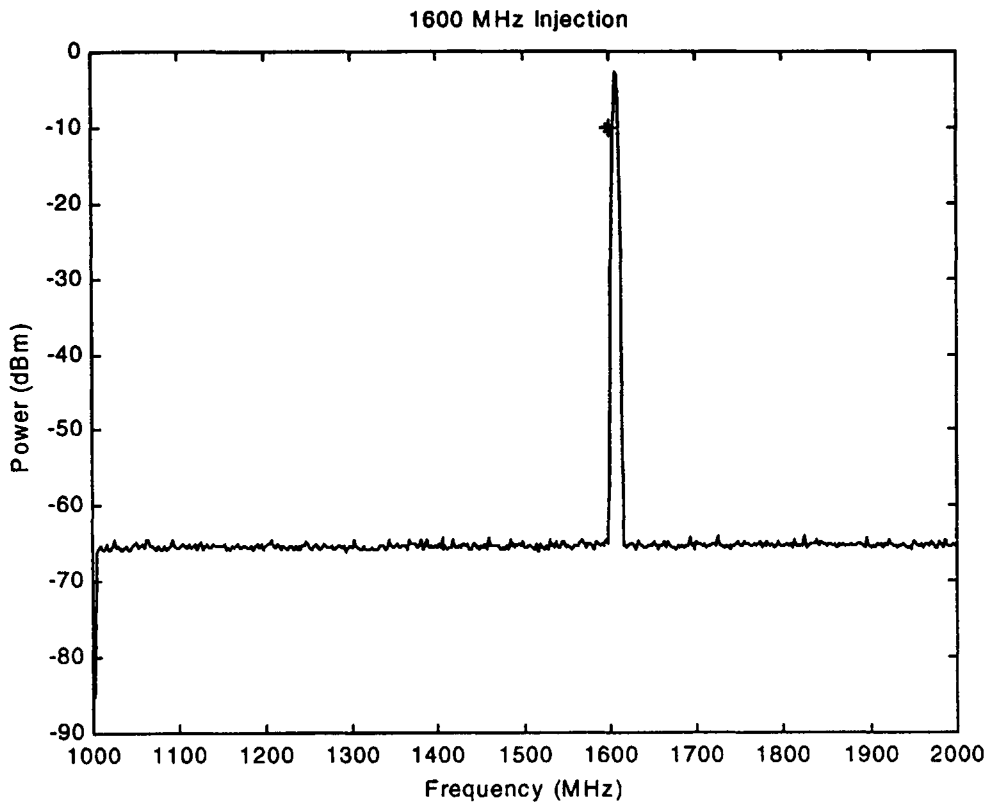

Injected Power: -10 dBm, 1500 MHz
Detected Power: -2.80 dBm, 1507.3 MHz



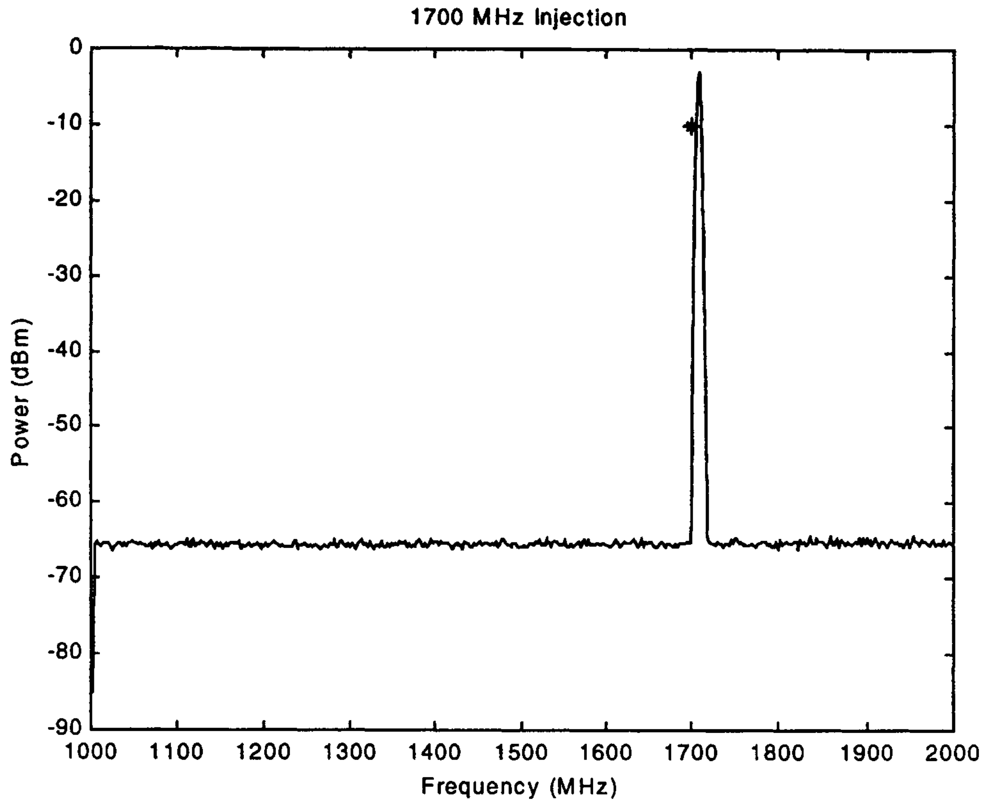

Injected Signal: -40.6 dBm, 1500 MHz
Detected Signal: -34.6 dBm, 1507.3 MHz



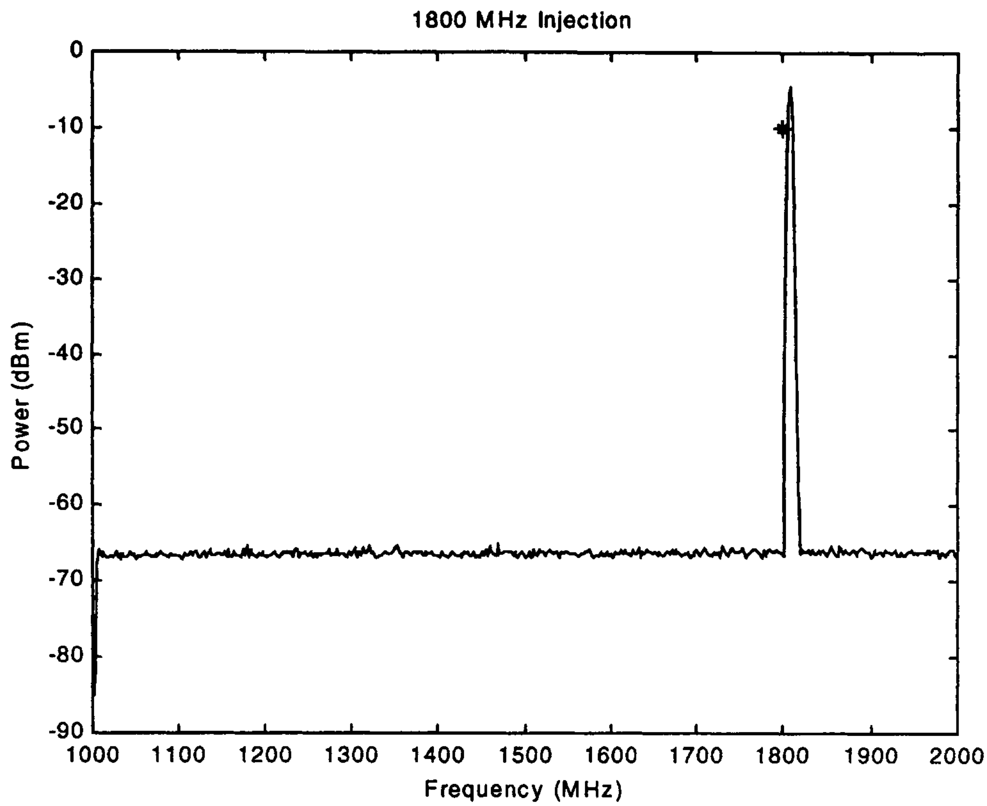

Injected Signal: -59.5 dBm, 1500 MHz
Detected Signal: -54.2 dBm, 1507.3 MHz



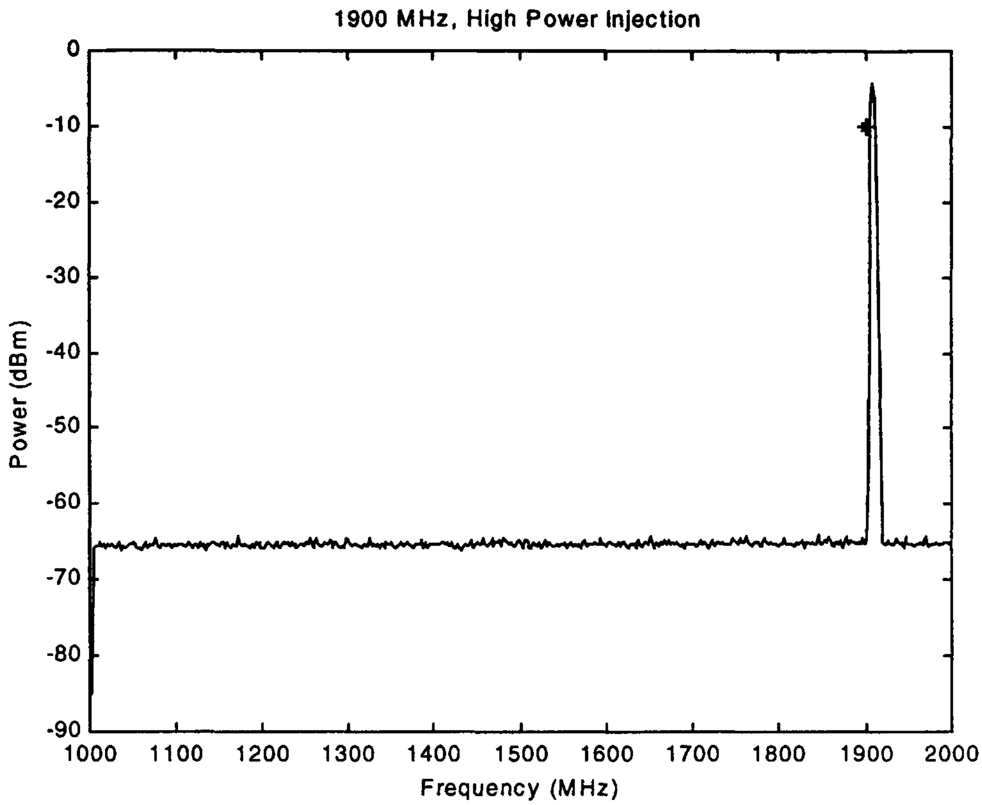

Injected Signal: -10 dBm, 1600 MHz
Detected Signal: -2.7 dBm, 1607.1 MHz



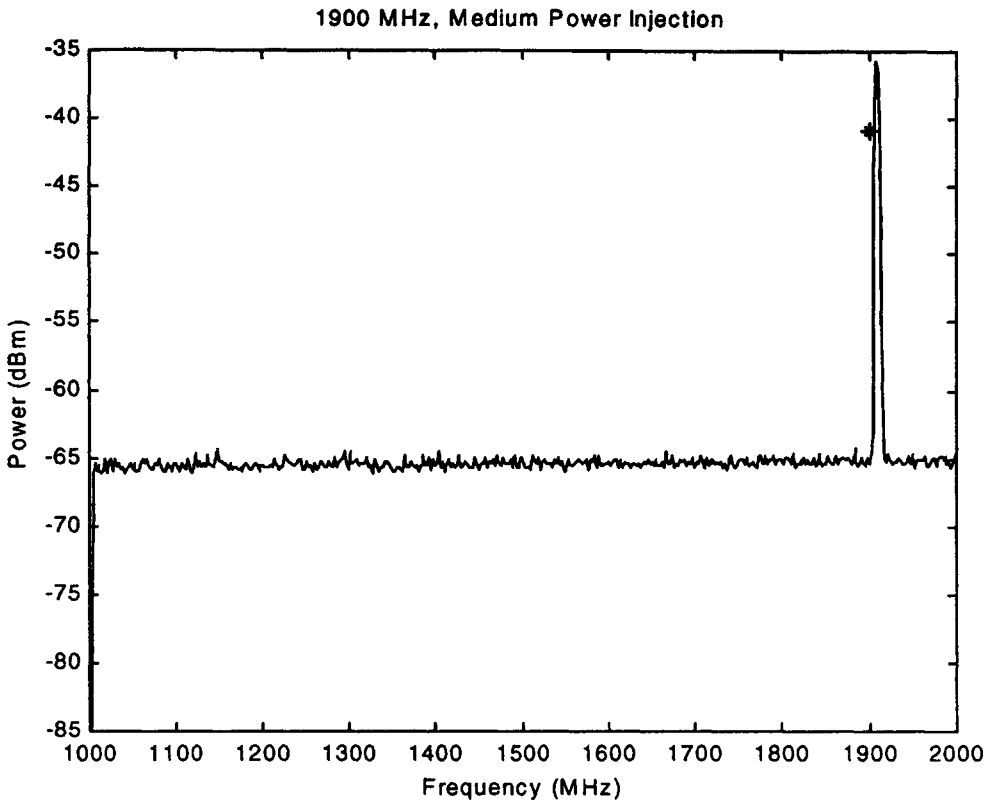

Injected Signal: -10 dBm, 1700 MHz
Detected Signal: -2.90 dBm, 1708.9 MHz



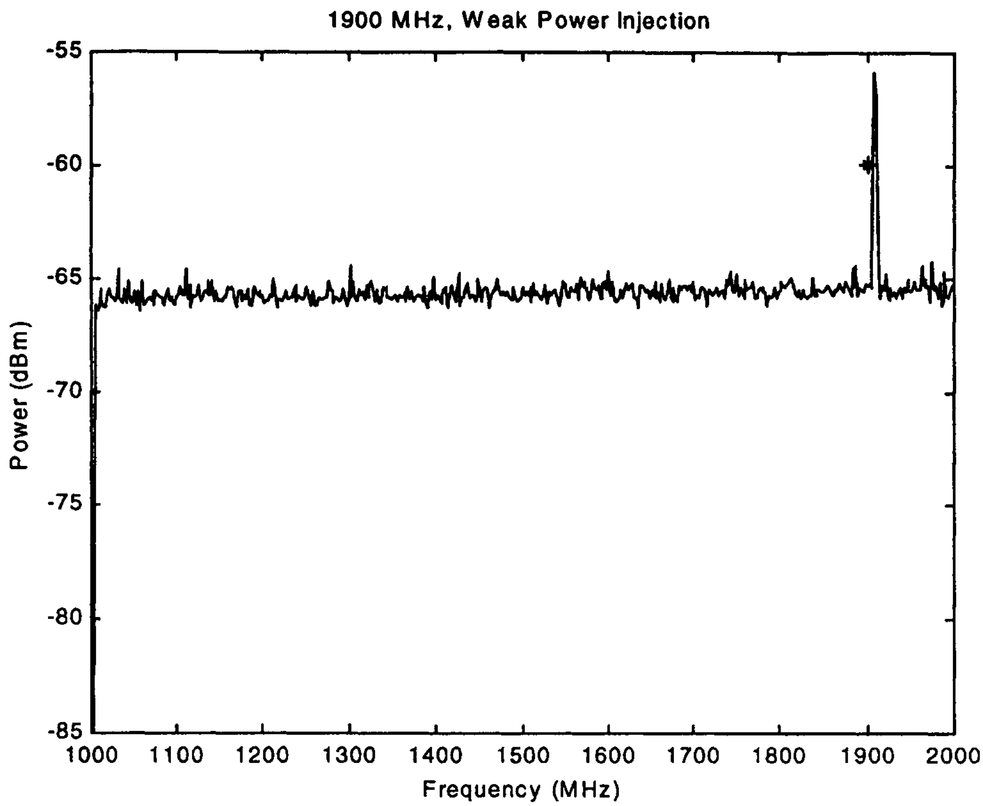

Injected Signal: -10 dBm, 1800 MHz
Detected Signal: -4.50 dBm, 1808.7 MHz



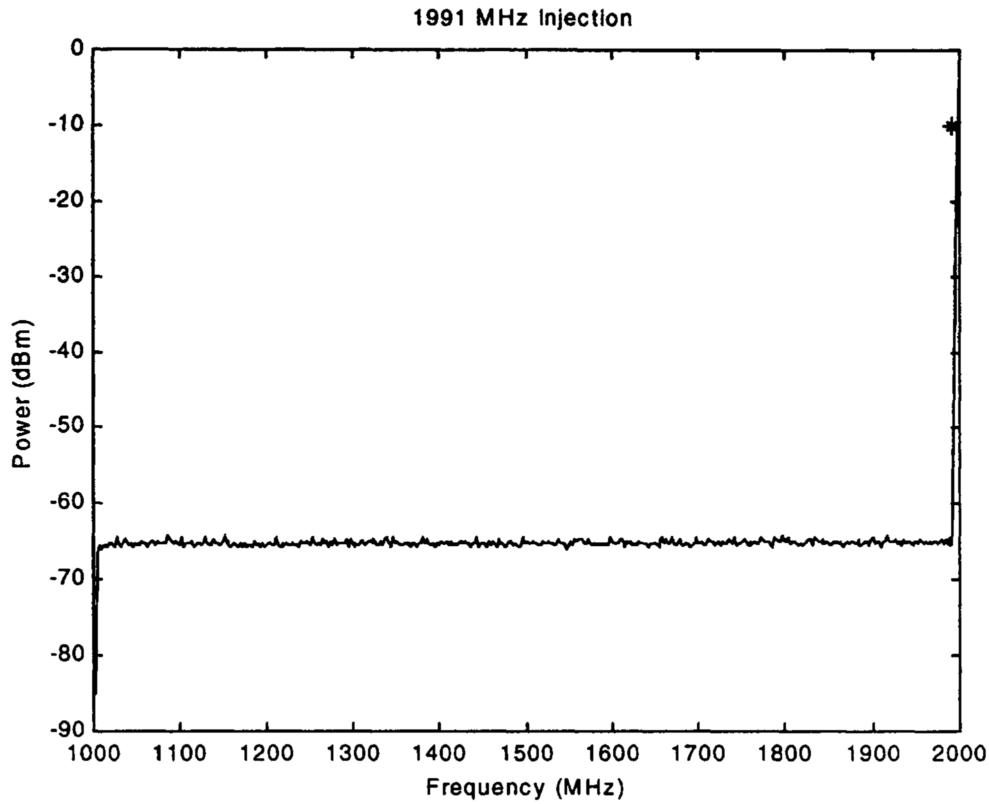

Injected Signal: -10 dBm, 1900 MHz
Detected Signal: -4.1 dBm, 1908.5 MHz



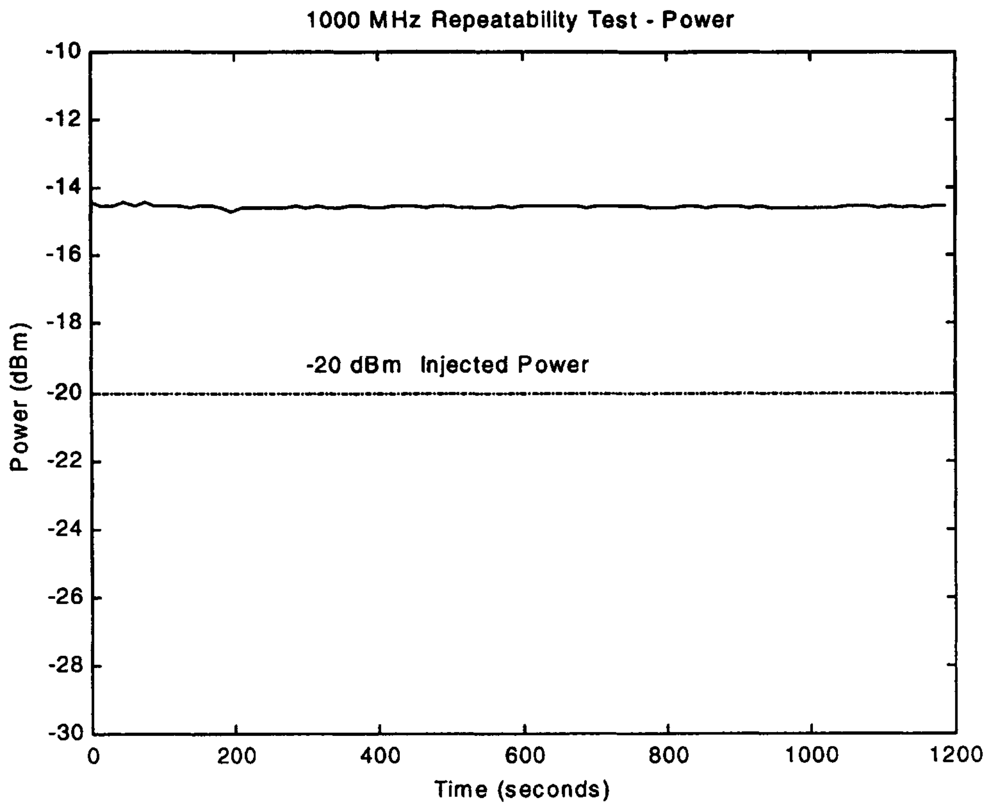

Injected Signal: -40.8 dBm, 1900 MHz
Detected Signal: -35.8 dBm, 1908.5 MHz

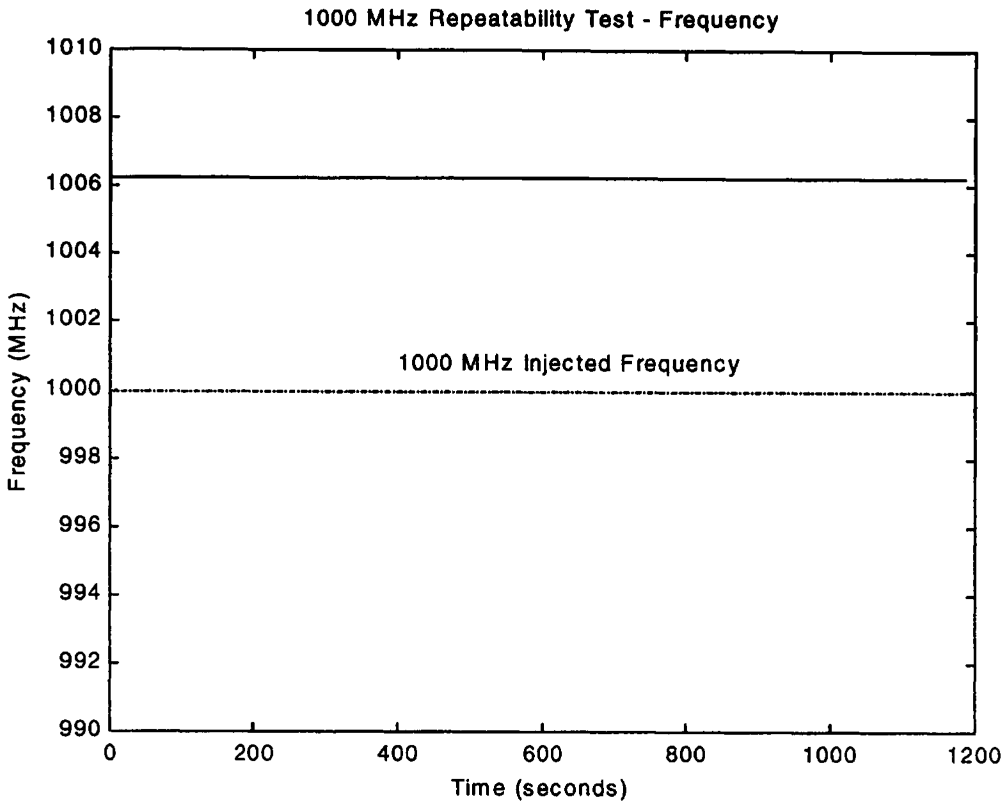


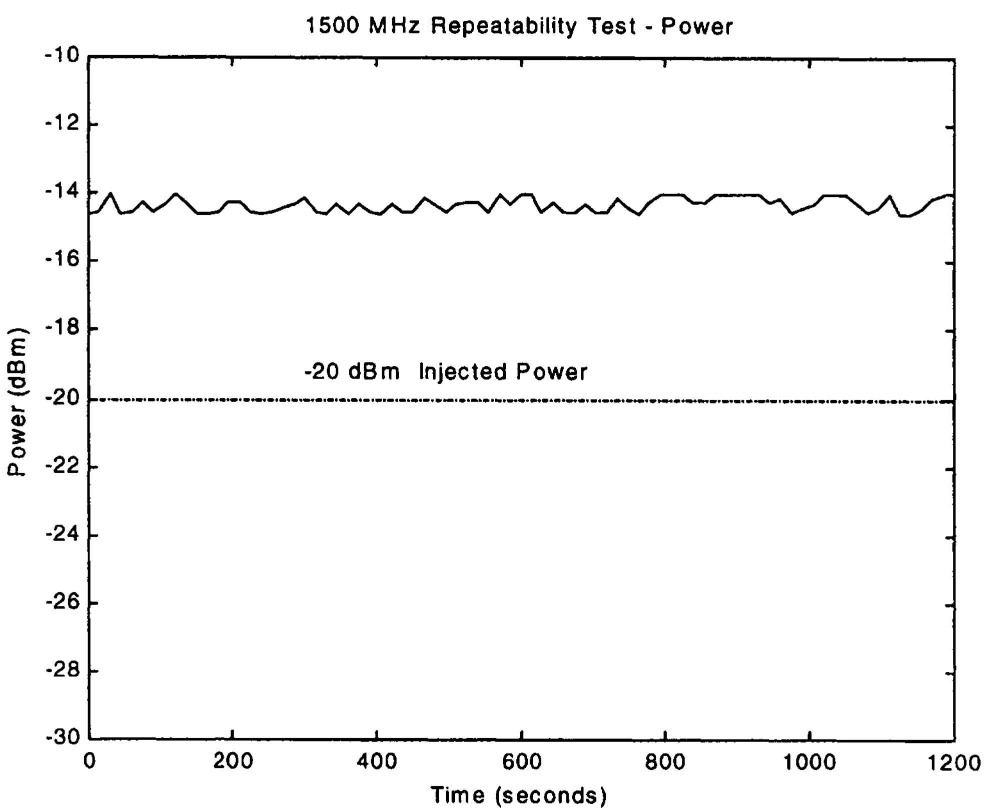

Injected Signal: -59.9 dBm, 1900 MHz
Detected Signal: -55.9 dBm, 1910.6 MHz

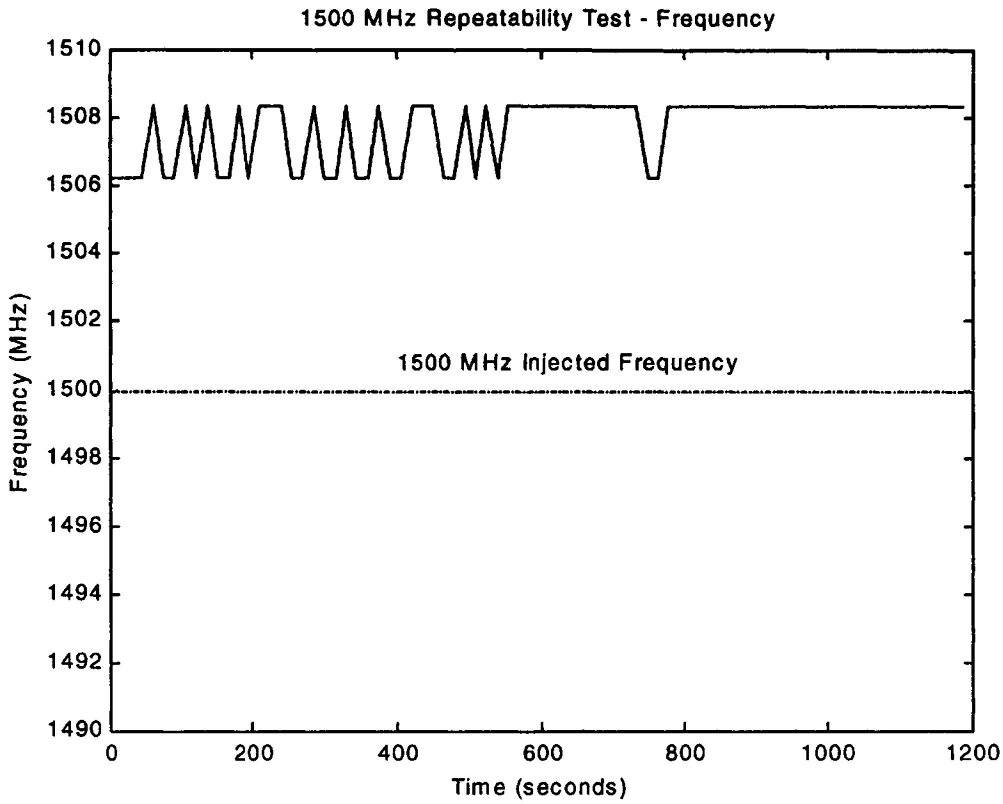


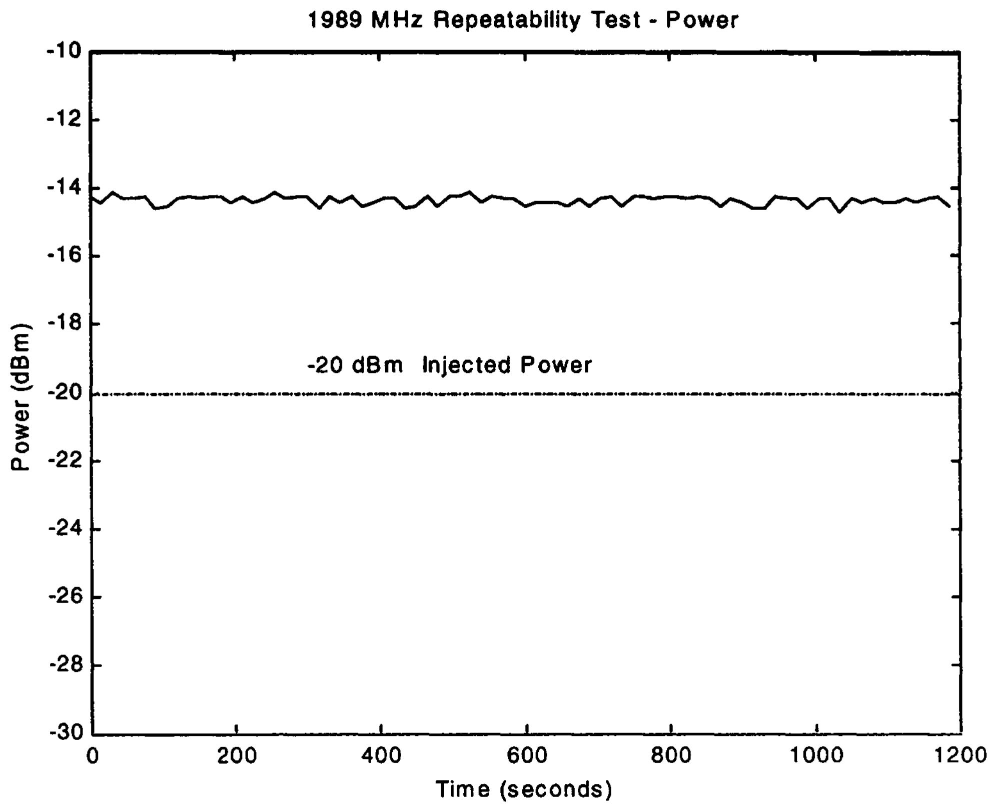

Injected Signal: -10 dBm, 1991 MHz
Detected Signal: -3.2 dBm, 2000 MHz

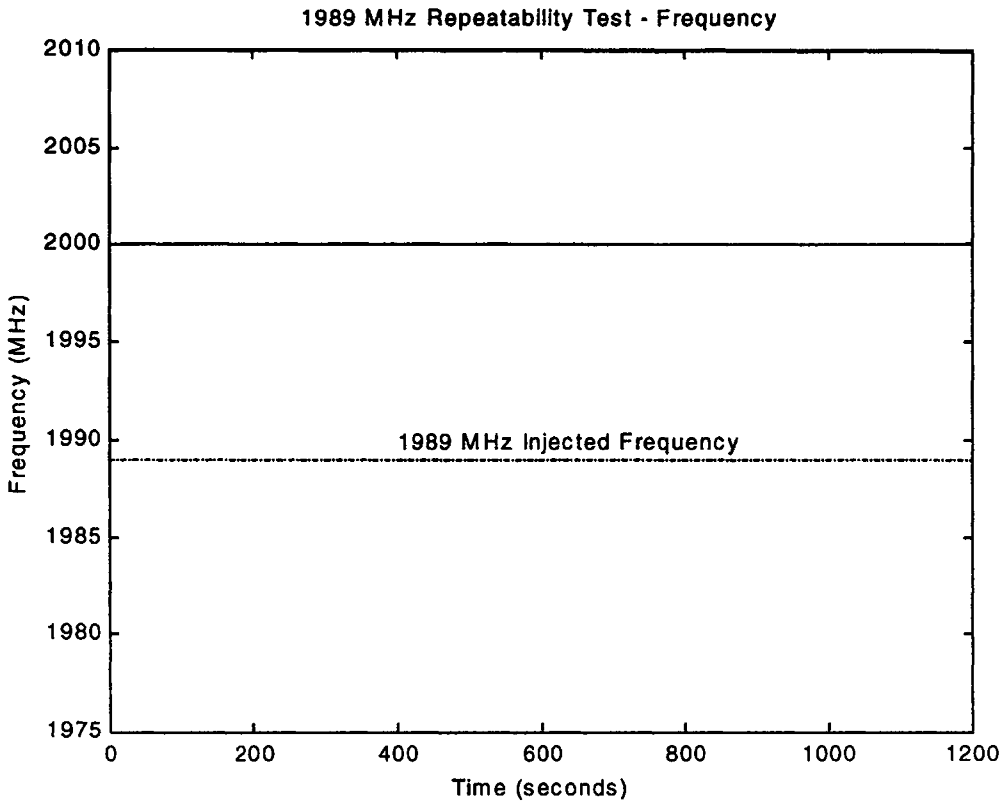

1000 MHz Repeatability Test - Power
Power (dBm)
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
-20 dBm Injected Power
0
200
400
600
800
1000
1200
Time (seconds)

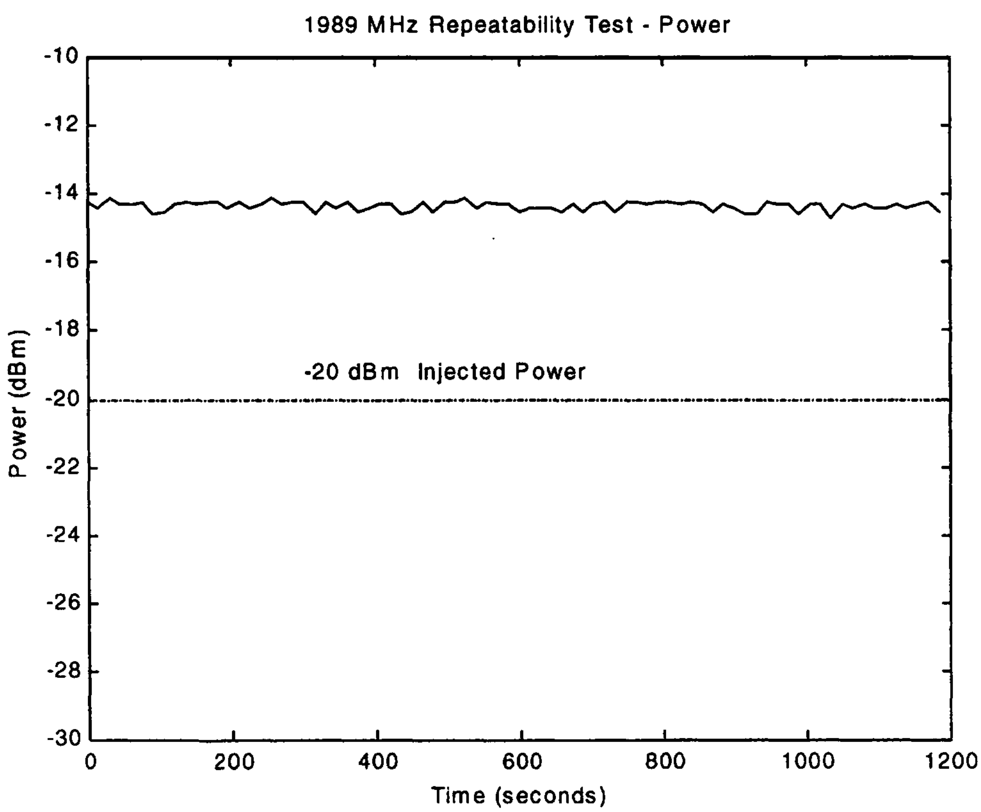

1000 MHz Repeatability Test - Frequency
Frequency (MHz)
1010
1008
1006
1004
1002
1000
998
996
994
992
990
1000 MHz Injected Frequency
0
200
400
600
800
1000
1200
Time (seconds)

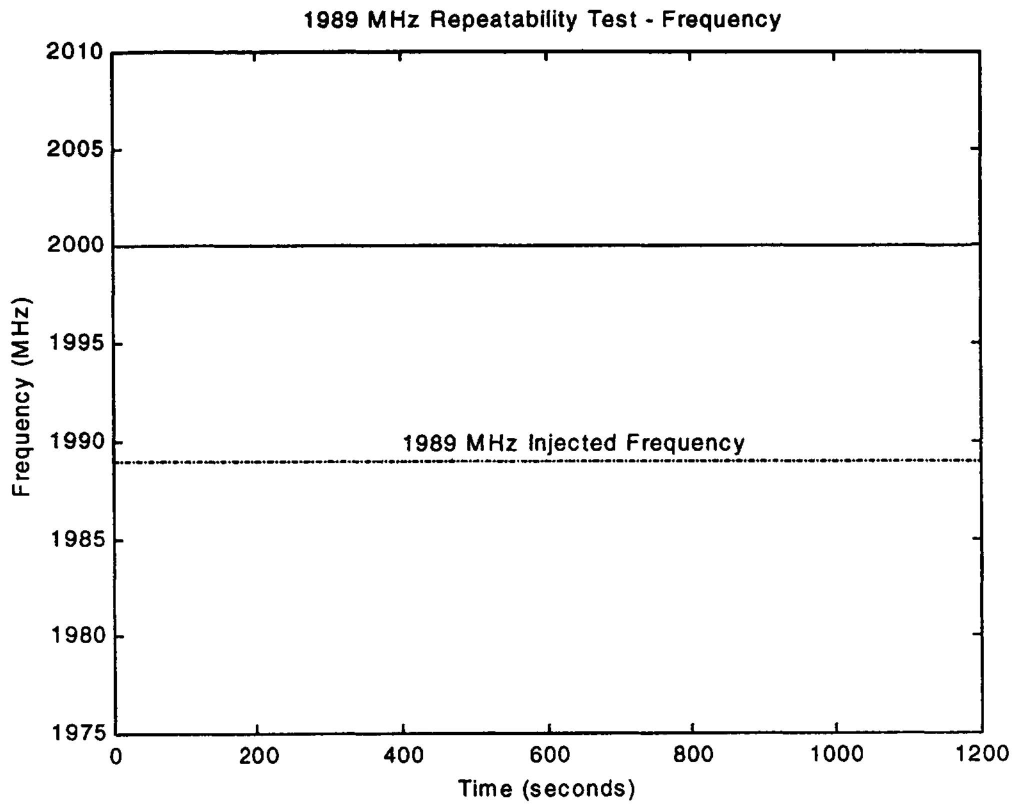

1500 MHz Repeatability Test - Power
Power (dBm)
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
-20 dBm Injected Power
0
200
400
600
800
1000
1200
Time (seconds)

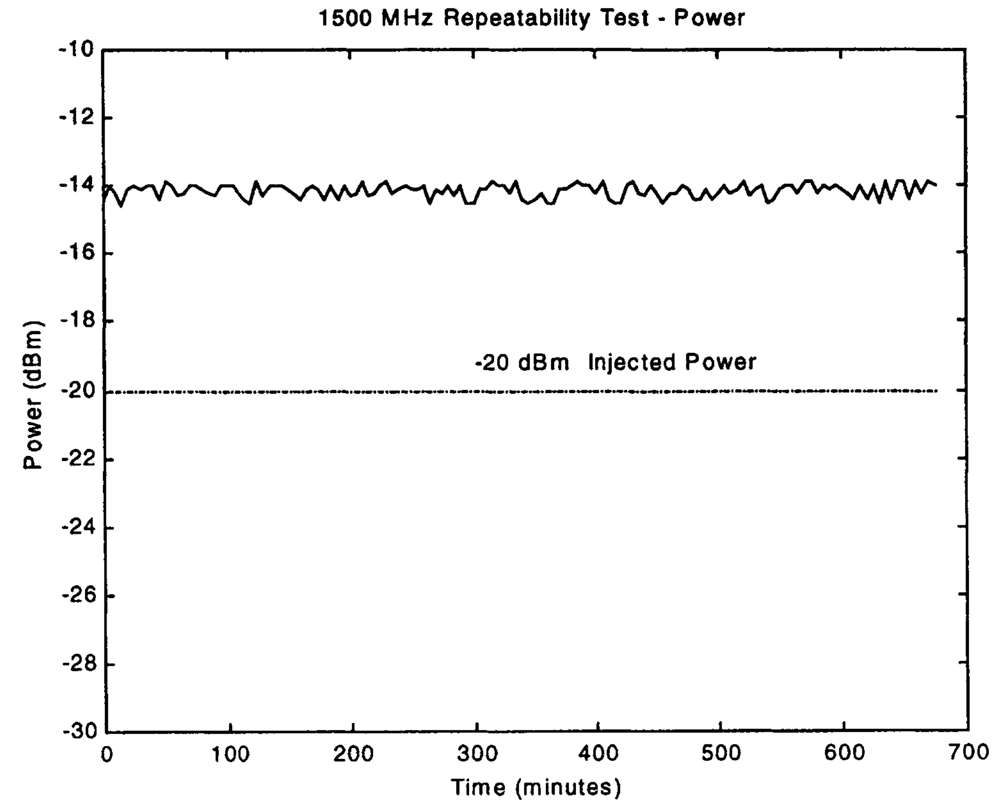

1500 MHz Repeatability Test - Frequency
Frequency (MHz)
Time (seconds)
1500 MHz Injected Frequency
1510
1508
1506
1504
1502
1500
1498
1496
1494
1492
1490
0
200
400
600
800
1000
1200

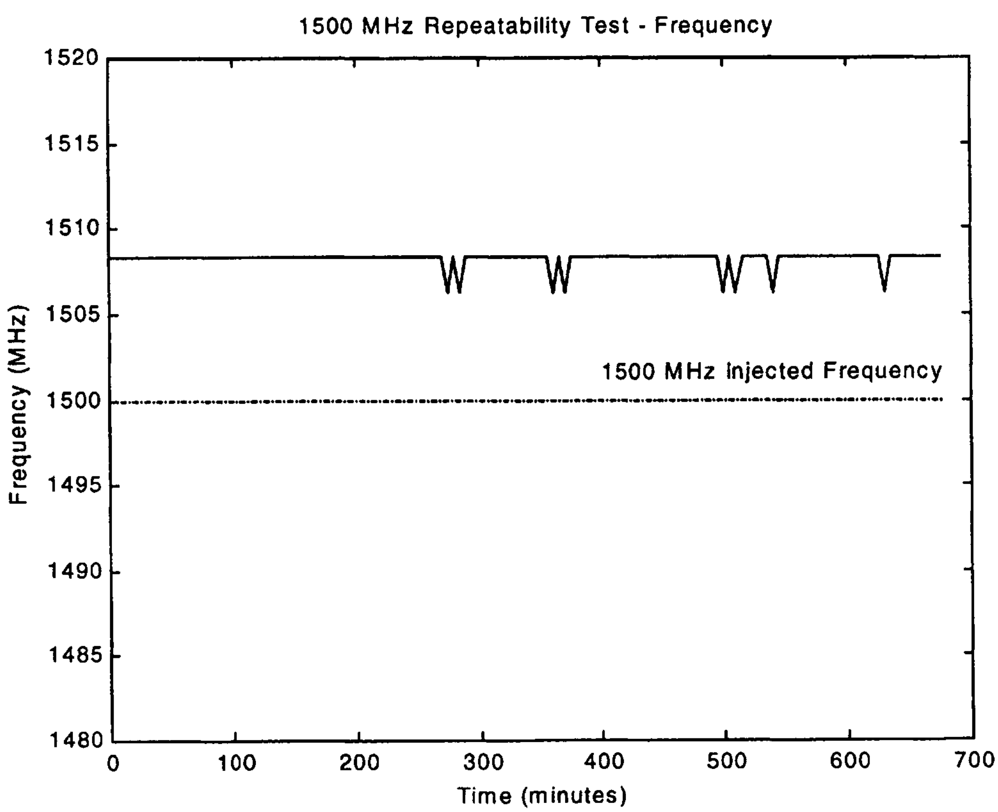

1989 MHz Repeatability Test - Power
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
Power (dBm)
-20 dBm Injected Power
0
200
400
600
800
1000
1200
Time (seconds)


1989 MHz Repeatability Test - Frequency
2010
2005
2000
1995
1990
1985
1980
1975
Frequency (MHz)
1989 MHz Injected Frequency
0
200
400
600
800
1000
1200
Time (seconds)


1989 MHz Repeatability Test - Power
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
Power (dBm)
-20 dBm Injected Power
0
200
400
600
800
1000
1200
Time (seconds)

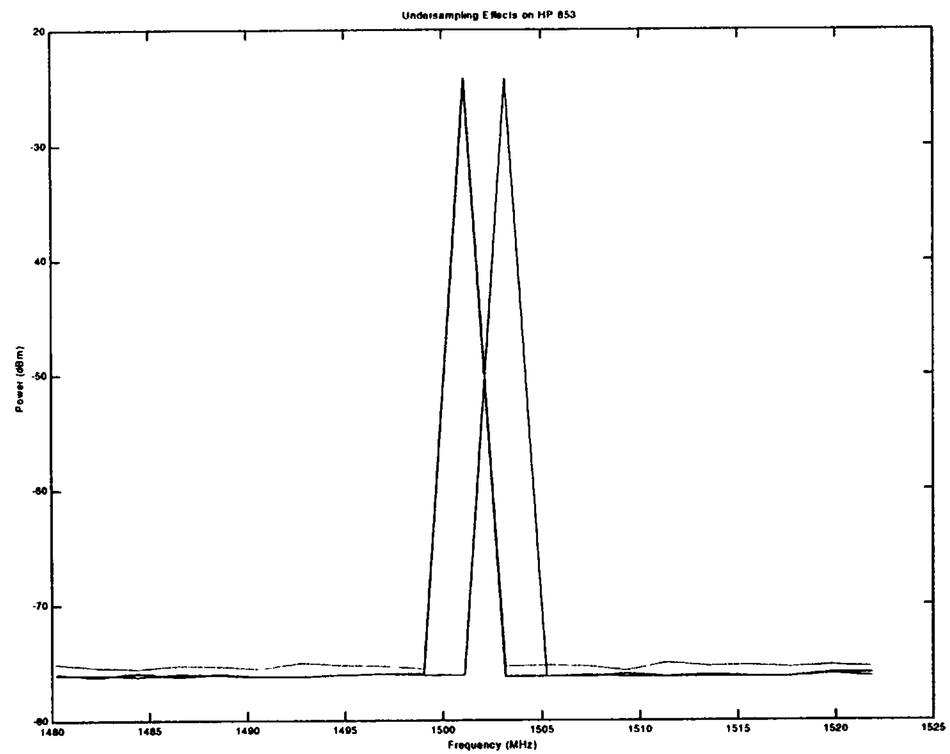

1989 MHz Repeatability Test - Frequency
2010
2005
2000
1995
1990
1985
1980
1975
Frequency (MHz)
1989 MHz Injected Frequency
0
200
400
600
800
1000
1200
Time (seconds)


1500 MHz Repeatability Test - Power
-10
-12
-14
-16
-18
-20
-22
-24
-26
-28
-30
Power (dBm)
-20 dBm Injected Power
0
100
200
300
400
500
600
700
Time (minutes)

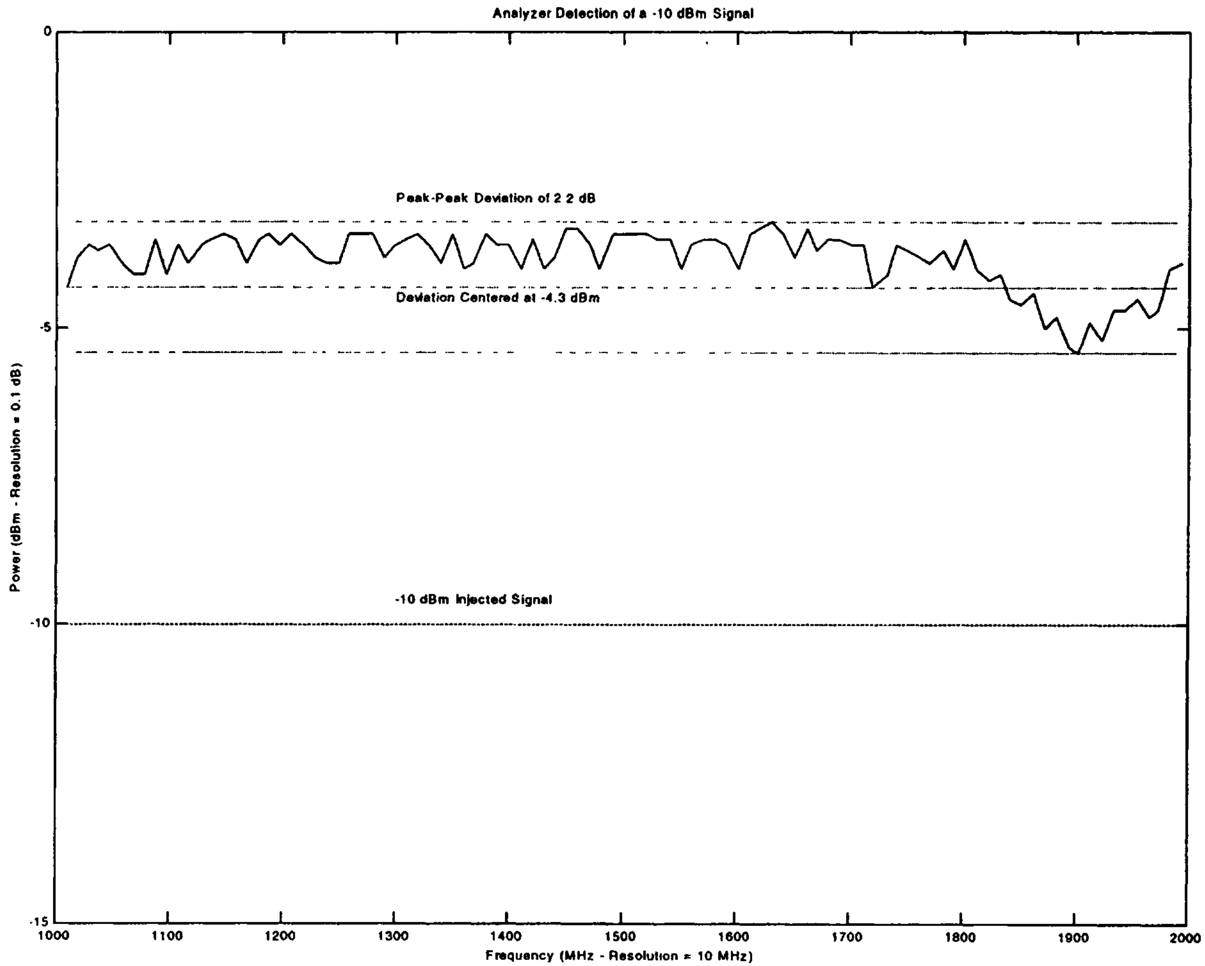

1500 MHz Repeatability Test - Frequency
1520
1515
1510
1505
1500
1495
1490
1485
1480
Frequency (MHz)
1500 MHz Injected Frequency
0
100
200
300
400
500
600
700
Time (minutes)

## 3.4.2 Undersampling Test of Spectrum Analyzer

(frequency resolution is 2.083333 MHz)




blue = 1500 MHz injection red = 1501.39 MHz injection green = 1502.79 MHz injection


Analyzer Detection of a -10 dBm Signal
Peak-Peak Deviation of 2.2 dB
Deviation Centered at -4.3 dBm
-10 dBm Injected Signal
0
-5
-10
-15
Power (dBm - Resolution = 0.1 dB)
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
Frequency (MHz - Resolution = 10 MHz)

### 3.4.3 Amplifier Power Line Application

The current design calls for the power, that the amplifiers in the front end run off of, be supplied to the front end box via a separate line than the signal path. There is another process that is available. The power can be supplied by biasing the coaxial cable that the signal travels.

The design will call for a DC block to be placed between the DC power bias "tee" (that the voltage for the amplifiers is pulled from) that will be situated "after" the output of the second amplifier – before the DC voltage gets to the second amplifier, and a second DC block will be required between the bias "tee" that the power supply will feed and the receiver. As mentioned two DC bias "tee"s will also be required. Applicable parts for this design are Pasternack Enterprises' PE 8212 DC block and PE 1601 bias "tee".

There are several advantages to this procedure:

- The cable protection of the DC line will be simplified since the DC line will be the signal path as well – which currently has a swivel joint at the top plate of the pedestal (currently protected from twisting due to excessive revolutions of the azimuth axis).
- There will be less cables to interface with the front end, and less interfaces to weatherproof.
- The signal path is a more ideal type of cable to have exposed to the environment (RFI and physical environment).

The disadvantages to this procedure are:

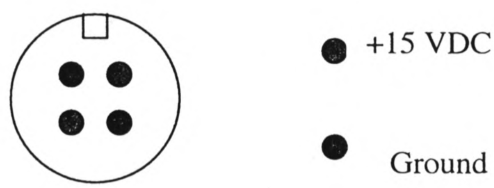
- Cost: The DC blocks cost $144.95 each (two required) and the bias "tee"s cost $139.95 each (two required). Total Cost: $569.80 + s&h.
- A small amount of receiver sensitivity will be lost (about 0.5 dB per component => 2.0 dB loss total).
- Isolation between the DC power and the RF signal will be 30 dB minimum; any RFI from the DC source could interfere with accurate observations.
- The DC blocks will operate from 10 MHz to 18 GHz, but the bias "tee"s only operate from 100 kHz to 2 GHz - another component that will limit/hinder expansion of the STS tool.

**Conclusion:**

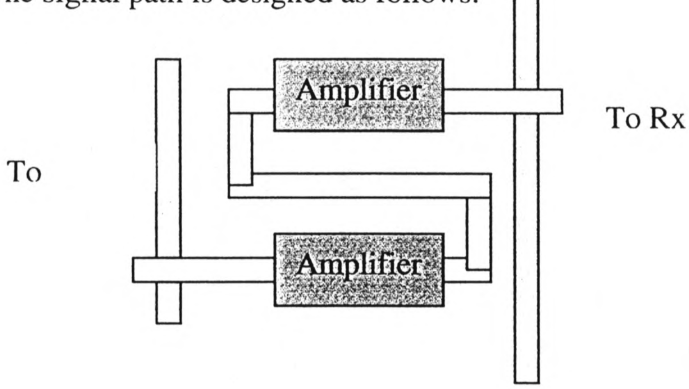
Future expansion of the DC power application will require more money and if the operational frequency range of STS is desired to be above 2 GHz then different bias "tee"s will need to be researched. For the time being the power for the amplifiers will be independently supplied to the front end on a dedicated set of power lines.

## 3.4.4 Front End Composition

The STS front end is composed to two serially linked Miteq AM-2A-1020 amplifiers. These amplifiers are specified to give 19 dB gain from 1.2 to 1.8 GHz (and will operate from approximately 300 MHz to 2.0 GHz with impacts to gain flatness and noisefigures). The signal path within the front end is 141, semi-rigid coaxial cable. The signal connectors are all SMA. SMA bulkhead feed-throughs interface the internal components to the outside. The DC power is interfaced with a KMC 9mm Bayonet Lock Connector (4 pin; Digikey # HR402-ND recepticle and HR400-ND plug). The design of the front end box allows for any maintenance to be done by unbolting the cylinder from the mounting, base plate and then sliding the cylinder off the chasis (make sure to disconnect the antenna-front end cable before doing so). Within the chasis, dc power is bussed from one side of the central plate and the signal path and amplifiers occupy the opposite side. Currently +15 VDC is the only power supplied to the front end in the following configuration (view is from looking into receptacle on the front end from outside) :




The signal path is designed as follows:



3.4.5 Excel PFD Calibration File:

| Total Loss | 4π | λ | ant gain | fe gain | lloss | s.a. offset | freq | sts power | stdgn hrn | I-J/10 | stdgn dBi | fe out | fe in | line out | line in | s.a. det. | s.a. in |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -53.8003 | 11.0 | 10.5 | -45.85 | -34.1 | 10 | -5.3 | 1000 | 694 | 351 | 34.3 | 11.55 | 597 | 256 | 750 | 850 | -4.7 | -10 |
| -54.4937 | 11.0 | 10.5 | -45.8615 | -34.1 | 9.3 | -5.3 | 1002.083 | 694 | 351 | 34.3 | 11.5615 | 597 | 256 | 764 | 857 | -4.7 | -10 |
| -54.0872 | 11.0 | 10.5 | -45.873 | -34 | 9.6 | -5.3 | 1004.167 | 694 | 351 | 34.3 | 11.573 | 590 | 250 | 750 | 846 | -4.7 | -10 |
| -52.5807 | 11.0 | 10.5 | -43.3845 | -33.9 | 9 | -5.8 | 1006.25 | 699 | 381 | 31.8 | 11.5845 | 592 | 253 | 764 | 854 | -4.2 | -10 |
| -50.3742 | 11.0 | 10.5 | -42.096 | -33.7 | 8.9 | -5 | 1008.333 | 703 | 398 | 30.5 | 11.596 | 597 | 260 | 765 | 854 | -5 | -10 |
| -50.6678 | 11.0 | 10.5 | -42.2075 | -33.8 | 8.9 | -5.1 | 1010.417 | 705 | 399 | 30.6 | 11.6075 | 596 | 258 | 764 | 853 | -4.9 | -10 |
| -49.4614 | 11.0 | 10.6 | -40.919 | -33.7 | 9.1 | -5.5 | 1012.5 | 708 | 415 | 29.3 | 11.619 | 597 | 260 | 764 | 855 | -4.5 | -10 |
| -49.9551 | 11.0 | 10.6 | -41.4305 | -34.2 | 9.3 | -5.2 | 1014.583 | 711 | 413 | 29.8 | 11.6305 | 599 | 257 | 762 | 855 | -4.8 | -10 |
| -50.5488 | 11.0 | 10.6 | -41.442 | -34.7 | 9.3 | -5.3 | 1016.667 | 714 | 416 | 29.8 | 11.642 | 601 | 254 | 764 | 857 | -4.7 | -10 |
| -51.5425 | 11.0 | 10.6 | -42.3535 | -34.5 | 9 | -5.3 | 1018.75 | 719 | 412 | 30.7 | 11.6535 | 599 | 254 | 764 | 854 | -4.7 | -10 |
| -50.8362 | 11.0 | 10.6 | -41.965 | -34.2 | 9 | -5.3 | 1020.833 | 717 | 414 | 30.3 | 11.665 | 601 | 259 | 764 | 854 | -4.7 | -10 |
| -50.83 | 11.0 | 10.7 | -42.2765 | -34.2 | 9.4 | -5.4 | 1022.917 | 721 | 415 | 30.6 | 11.6765 | 601 | 259 | 763 | 857 | -4.6 | -10 |
| -51.7238 | 11.0 | 10.7 | -42.188 | -35 | 9.2 | -5.4 | 1025 | 724 | 419 | 30.5 | 11.688 | 603 | 253 | 764 | 856 | -4.6 | -10 |
| -52.2177 | 11.0 | 10.7 | -43.0995 | -34.7 | 9.1 | -5.2 | 1027.083 | 728 | 414 | 31.4 | 11.6995 | 602 | 255 | 764 | 855 | -4.8 | -10 |
| -51.0116 | 11.0 | 10.7 | -42.411 | -34.4 | 9.3 | -5.2 | 1029.167 | 729 | 422 | 30.7 | 11.711 | 602 | 258 | 762 | 855 | -4.8 | -10 |
| -52.2055 | 11.0 | 10.7 | -43.4225 | -34.6 | 9.3 | -5.2 | 1031.25 | 729 | 412 | 31.7 | 11.7225 | 601 | 255 | 763 | 856 | -4.8 | -10 |
| -51.8995 | 11.0 | 10.7 | -43.234 | -34.2 | 9.1 | -5.3 | 1033.333 | 731 | 416 | 31.5 | 11.734 | 602 | 260 | 765 | 856 | -4.7 | -10 |
| -52.3935 | 11.0 | 10.8 | -43.4455 | -34.6 | 9.2 | -5.3 | 1035.417 | 733 | 416 | 31.7 | 11.7455 | 602 | 256 | 765 | 857 | -4.7 | -10 |
| -52.7876 | 11.0 | 10.8 | -43.457 | -34.5 | 9 | -5.6 | 1037.5 | 725 | 408 | 31.7 | 11.757 | 602 | 257 | 765 | 855 | -4.4 | -10 |
| -53.2816 | 11.0 | 10.8 | -44.4685 | -34.5 | 9.1 | -5.2 | 1039.583 | 728 | 401 | 32.7 | 11.7685 | 601 | 256 | 764 | 855 | -4.8 | -10 |
| -53.6758 | 11.0 | 10.8 | -45.18 | -34.2 | 9.3 | -5.4 | 1041.667 | 726 | 392 | 33.4 | 11.78 | 601 | 259 | 763 | 856 | -4.6 | -10 |

Total Loss = sum loss of system (negative is gain)
4pi = spreading loss
lambda = effective aperture
ant gain = (sts power – stdgnhrn)/10 + stdgn dBi
fe gain = (fe in – fe out)/10
lloss = (line in – line out)/10
s.a. off set = s.a. in – s.a. det
freq = approx. frequency of specific bin
sts power = ascii set from s.a. when recording power rx with STS
stdgn hrn = ascii set from s.a. when recording power rx with standard gain horn
I-J/10 = (sts power – stdgn hrn)/10
Stdgn dBi = isotropic gain of standard gain horn
FE out = ascii set from s.a. when recording output of front end
FE in = ascii set from s.a. when recording input power to front end
Line out = ascii set from s.a. when recording the output of signal path
Line in = ascii set from s.a. when recording input power to signal path
s.a. det. = recorded dBm level when injecting a known power to s.a.
s.a. in = known power into s.a

## PFD Theory (Code Implementation)

The equation used for the PFD implementation is:

$$PFD = P_{RX} * A_e \text{ in linear units of Watts}$$

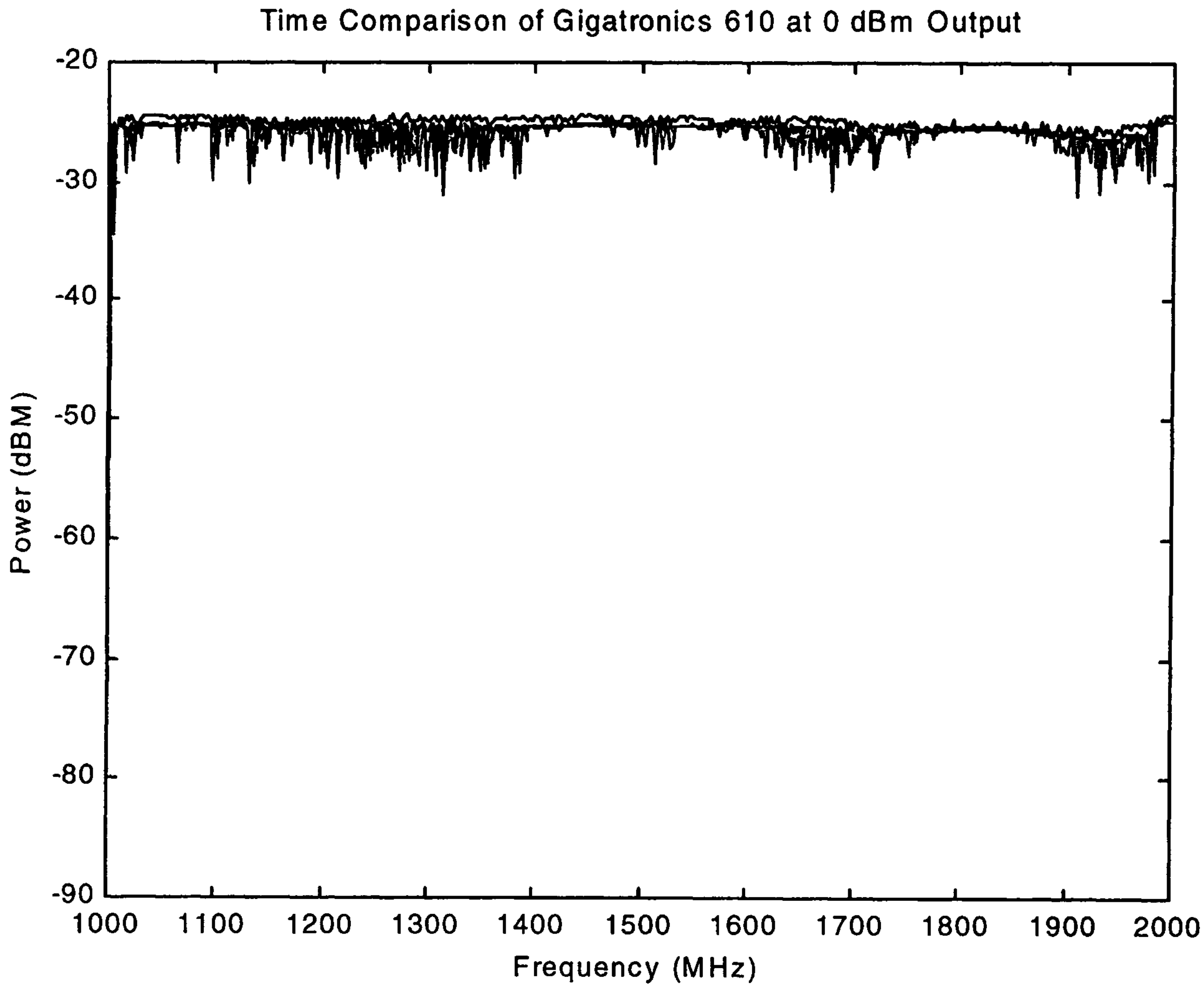
the effective aperture is:

$$A_e = \frac{4 * \pi * G_{antenna}}{\lambda^2}$$

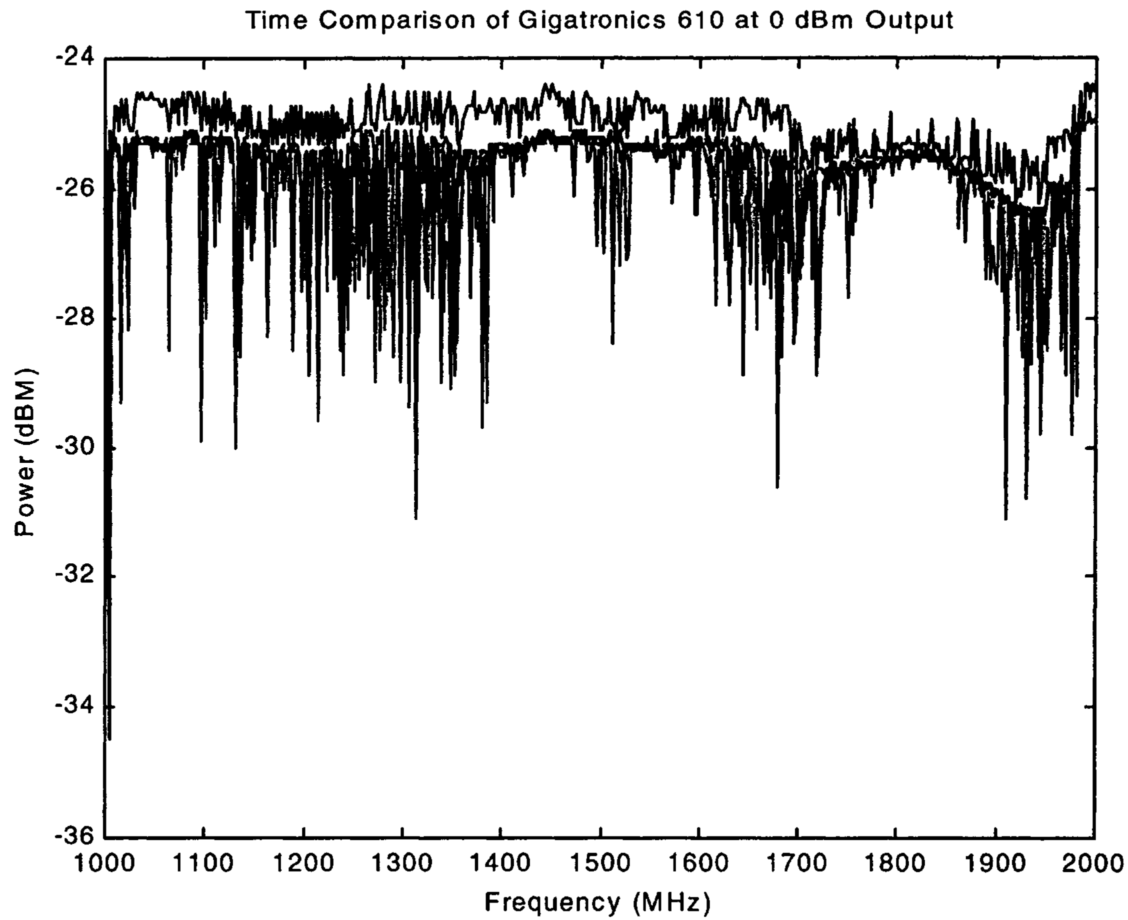
solving for PFD in dBW/m$^2$ is

$$PFD = P_{SA} - G_{antenna} - G_{FE} + Loss_{line} + Cal_{SA} = 10 * \log(4 * \pi) - 20 * \log(\lambda^2)$$

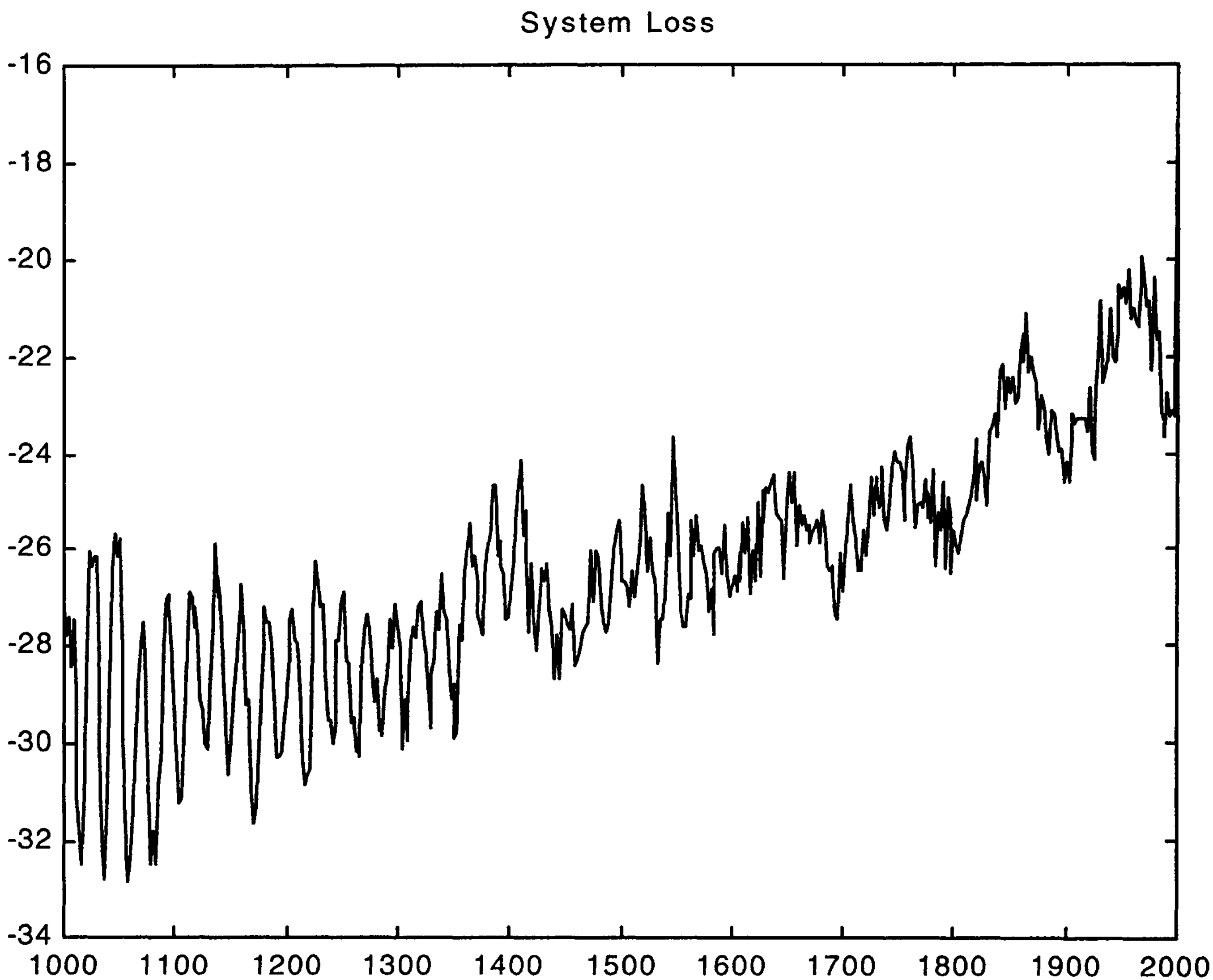
### 3.4.6 Plots of the RF System Budget (component measurement)

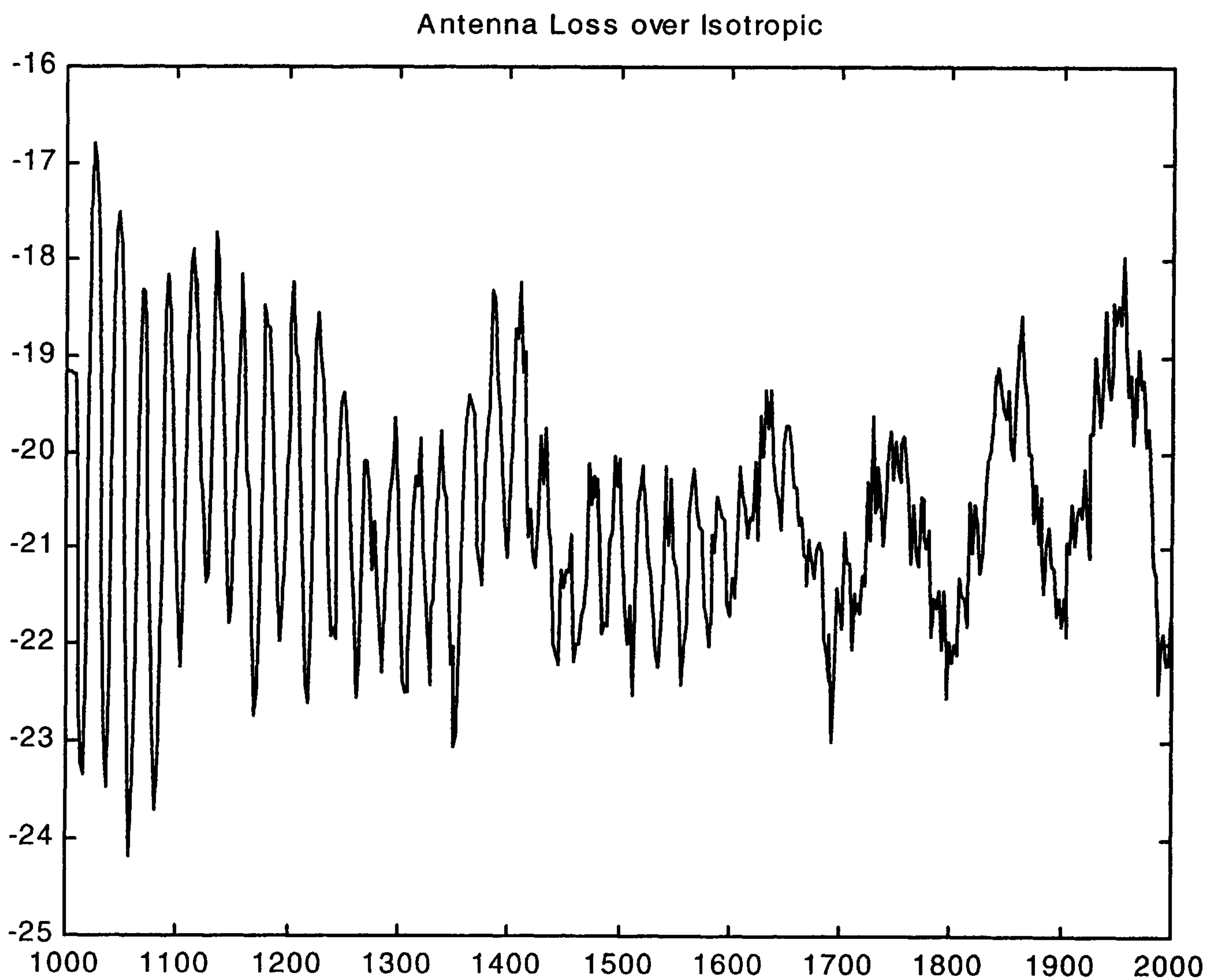


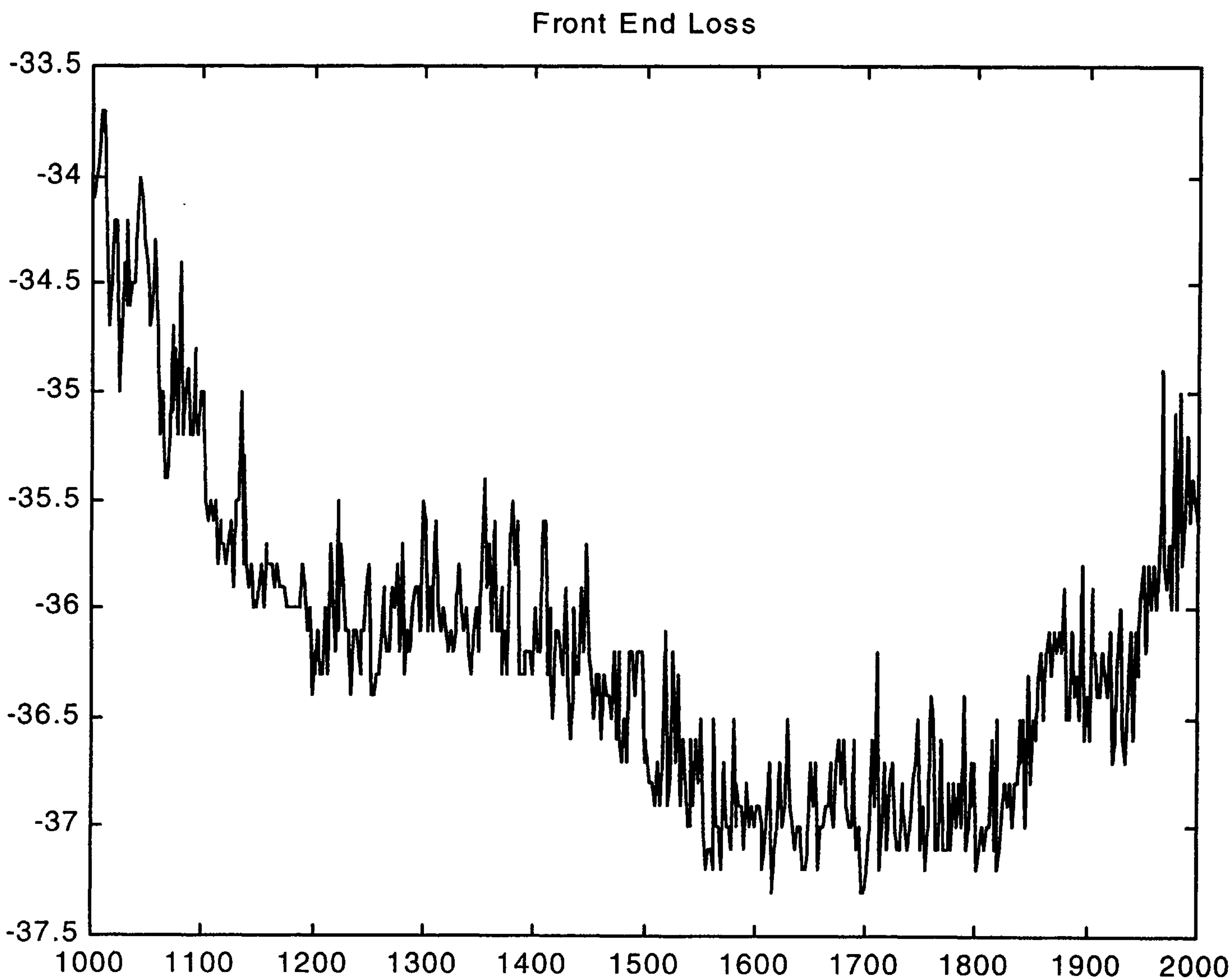

black = peak hold over several sweeps, all colors are single sweeps

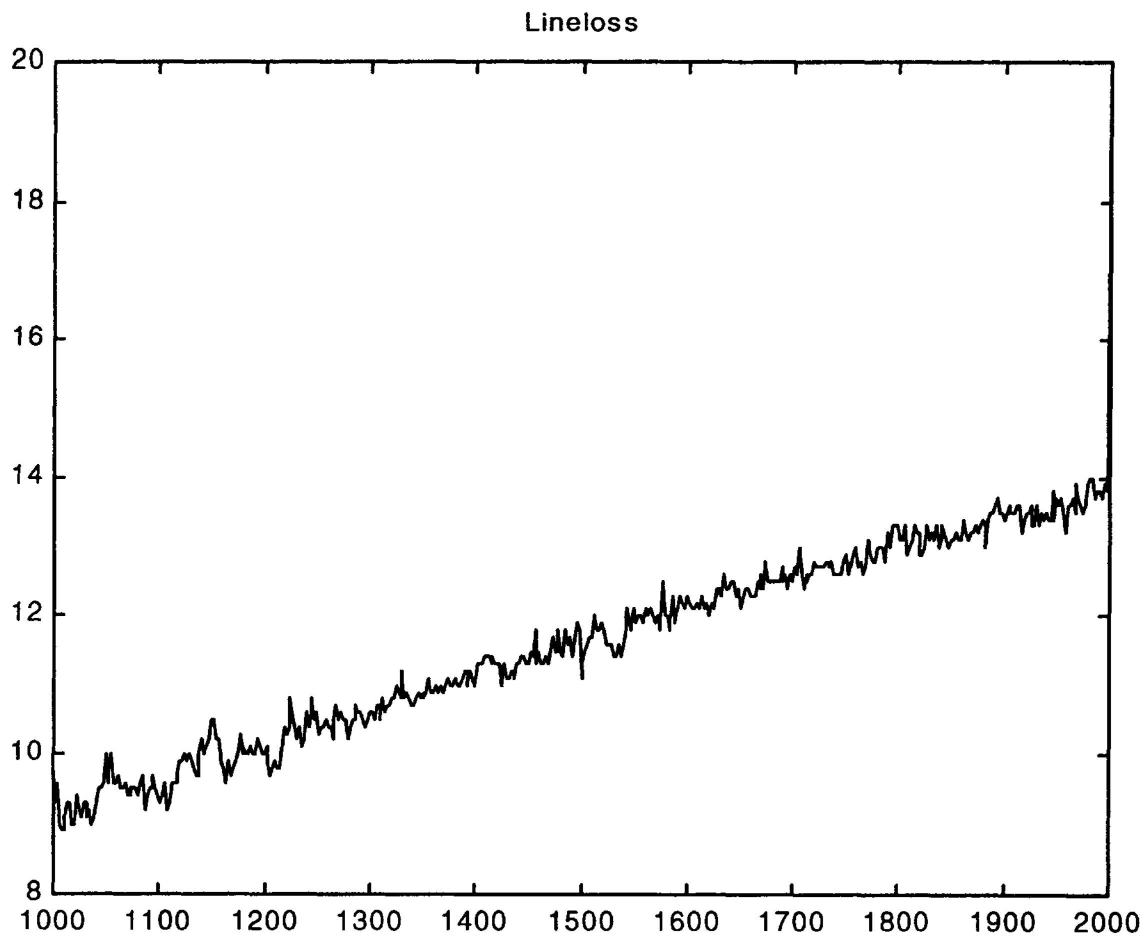


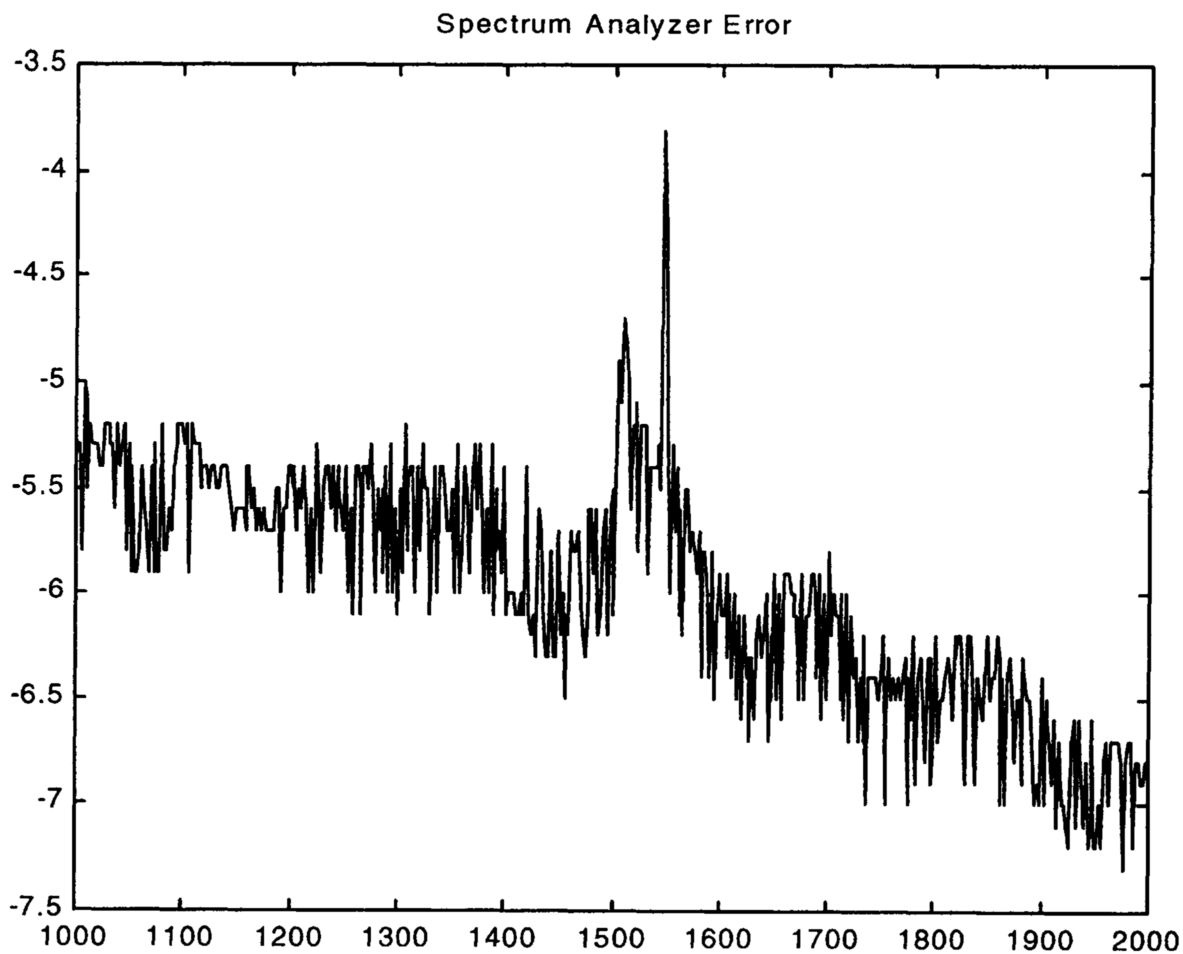

black = peak hold over several sweeps, all colors are single sweeps


System Loss
-16
-18
-20
-22
-24
-26
-28
-30
-32
-34
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000


Antenna Loss over Isotropic
-16
-17
-18
-19
-20
-21
-22
-23
-24
-25
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000


Front End Loss
-33.5
-34
-34.5
-35
-35.5
-36
-36.5
-37
-37.5
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000


Lineloss
20
18
16
14
12
10
8
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000


Spectrum Analyzer Error
-3.5
-4
-4.5
-5
-5.5
-6
-6.5
-7
-7.5
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000

# Section IV: Nova for Windows

Nova can be found at www.nlsa.com