Constructing a Radio Interferometer

Samer Atiani

A thesis submitted to the faculty of Guilford College
in partial fulfillment of the requirements
for the degree of Bachelor of Sciences

Physics Department

May 6, 2006

Adviser, Rexford Adelberger
Chair of the Committee and Professor of Physics

Steve Shapiro
Associate Professor of Physics

Thom Espinola
Glaxo Wellcome Professor of Physics

Robert Williams
Voehringer Professor of Economics

Abstract

Interferometery allows two telescopes to achieve angular resolutions comparable to one larger telescope. We constructed a radio interferometer using two 90" L-Band (1.42 GHz) SRT Radio Telescopes from Haystack Observatory, each equipped with a Global Positioning System (GPS). We have constructed a wireless communications system that allows reliable fault-tolerant connections over distances on the order of hundreds of meters in modern suburban settings, theoretically extensible to 15 miles ($\sim$24 km) in clear line of sight (LoS) conditions. This is achieved by an elaborate setup of USB relay, modified Linux operating system, and a set of shell scripts programmed to consistently check for connection reliability. With a reliable connection and a functioning GPS system, the ground is laid for a flexible baseline interferometer with the automation of calculations of signal delay and phasing by processing GPS location data and high precision time tags associated with data collected from each telescope in the baseline.

Contents

Chapter 1

Introduction

1.1 The Electromagnetic Spectrum

Most of what we know about the extraterrestrial universe comes through information in the form of electromagnetic (EM) waves, including familiar phenomena such as visible light, x-rays, and radio. Those waves travel at a speed of approximately 3×10^8 m s^{-1} in quantized packets of energy called photons. The only distinguishing characteristic among different categories of EM waves (i.e. light, radio, infrared, etc.) is their energy, with gamma rays having the most energy, radio waves having a very low energy, and visible light falling somewhere in between with energy in the order of 2 eV. The energy of those waves per photon is defined by Einstein's formula:

$$E = hf,$$

where E is the energy of the photon, h is Planck's constant, and f is the frequency of the photon. It follows that the energy of the photon directly defines its frequency; the higher the energy the higher is the frequency of the photon. The term "electromagnetic spectrum" is the physicist's jargon to refer to the frequency divisions describing the different categories of EM waves as can be seen from Figure 1-1. [1]

1.2 Radio Astronomy

An object of particular interest to this thesis project is radio waves, typically taking the range between 100 kHz to 10 GHz (the range 10^5 Hz to 10^{10} Hz in Figure 1-1).

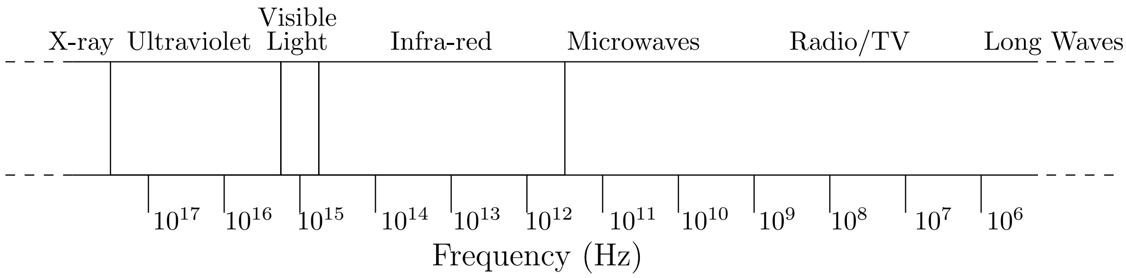


Figure 1-1: The Electromagnetic Spectrum

The earth's atmosphere and ionosphere are largely transparent to those frequencies, allowing us to detect radio signals from space using terrestrial telescopes. Radio astronomy is the branch of astronomy that specifically utilizes this atmospherical "radio window" to make useful observations of radio emitting sources.

What are radio emitting sources? Anything with a temperature of more than absolute zero naturally emits radio waves in the form of *blackbody radiation*[2] as defined by Planck's radiation law:[3]

$$B(v,T) = \frac{2hv^3}{c^2}\frac{1}{e^{hv/kT}-1}, \tag{1.1}$$

where B is brightness in W m^{-2} Hz^{-1} sterad^{-1}, h is Planck's constant, v is frequency in Hz, c is the velocity of light in m s^{-1}, k is Boltzmann's constant in J K^{-1}, and T is the temperature in K. What Equation (1.1) essentially says is that the brightness of radiation emitted by an ideal "blackbody" is both frequency and temperature dependent. As can be seen from Figure 1-2, which is a plot of Equation (1.1) on logarithmic scales, radio waves are emitted at temperatures as low as 1 K, or even below that, as long as we don't hit the theoretical absolute zero limit. Changing the temperature changes the relative intensity of emitted EM waves along the spectrum. The higher the temperature, the more relatively intense will be radiations at higher frequencies. For a celestial body such as the Sun, with outer core temperature of $\sim$ 6000 K, the frequencies at which the maximum relative intensity is achieved are within the optical range, making our Sun a good source of visible light. The Sun is also an important radio source due to its proximity and the relatively large solid angle it subtends on our sky.

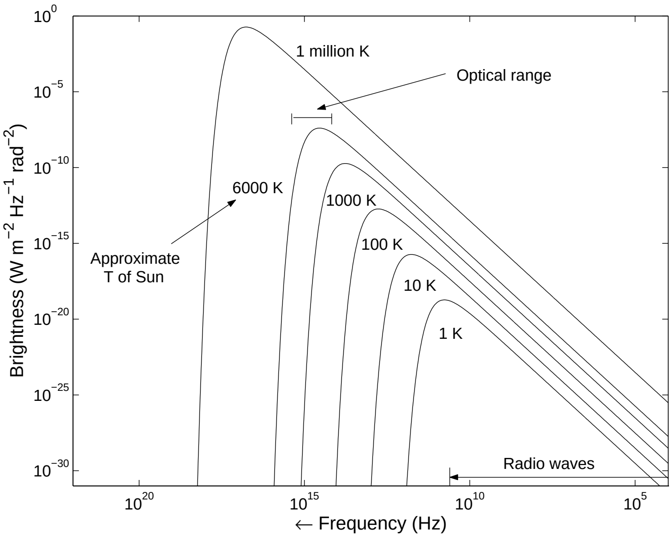


Figure 1-2: Planck-law radiation curves to logarithmic scales with brightness expressed as a function of frequency. Frequency increases to the left and wavelength increases to the right.

A fundamental result of Planck's law is that almost every object in the universe emits radio waves. In fact, we are living in a huge medium of radio noise emitted all around us. Looking at most of the objects around us with a radio wave detector and a spectroscope[4] should yield an up-sloping line on a logarithmic intensity-vs.-frequency graph in the radio range. This means that a lot of noise will be introduced during any attempt of observing radio waves due to the blackbody radiation from discrete and continuum sources. The blackbody radiation will appear as background noise wherever in the sky we look.

Another source of continuum emission is thermal emission from ionized gas. Those emissions occur as free electrons in an ionized celestial gas cloud are accelerated towards protons and deflected as they move past the protons. Since free electrons have no definite energy levels, the deflection releases photons in a continuous spectrum of energies. Those emissions are called free-free transitions, or *Bremsstrahlung*, because the electron is free before and after the interaction.[5]

Another more interesting source of radio emissions is discrete emissions from neutral atoms and molecules, which are caused by atomic and molecular processes. In contrast to continuum radiation, *spectral line emission* is generated at specific, well-defined frequencies associated with transitions between energy levels in atoms or molecules. [6] One of the most interesting line emissions is one that is generated by neutral hydrogen atoms; a single proton orbited by a single electron. The electron and proton have magnetic moments, called "spins", and can have either spins with the same direction, or the opposite direction. A hydrogen atom that has an electron and proton with spins in the same direction (parallel) has slightly less energy than one where the electron and proton have opposite spins (anti-parallel). The transition between parallel and anti-parallel spins is highly forbidden by the quantum mechanic model of the hydrogen atom, with an extremely small probability of 2.9×10^{-15} s^{-1}. This means that the that time needed for one spin transition to occur is in the order

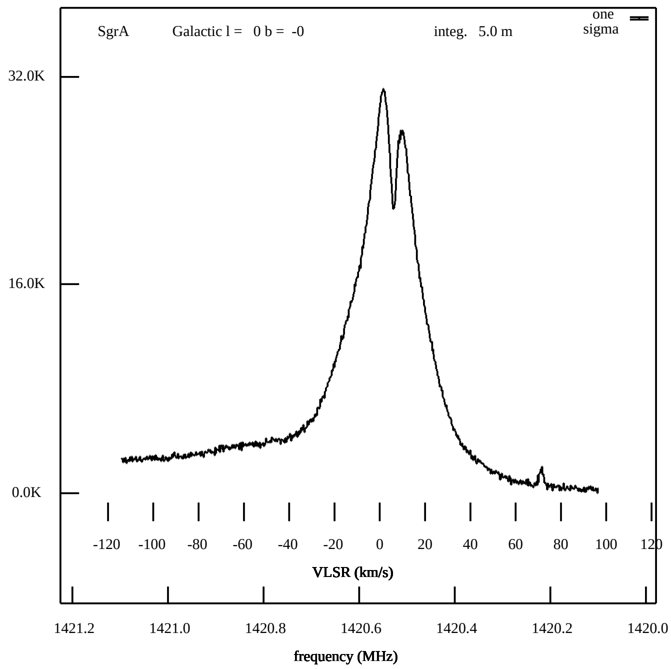


Figure 1-3: A radio graph of apparent temperature (on vertical coordinates) versus frequency of radio source Sagittarius A, taken with Guilford's Ashtarut radio telesope; one of the telescopes I worked on for my project. The graph shows two adjacent doppler-shifted spikes (line emissions) on the blackbody radiation spectrum and one miniscule spike on the left.

of 10 million years. Thus, it will be extremely difficult to reproduce such transitions in terrestrial laboratories. However, in space, neutral hydrogen is abundant in massive quantities that make neutral hydrogen emissions observable by radio telescopes at precisely 1420.41 MHz frequency. Furthermore, hydrogen atoms collisions in the space keep this process going by changing the spins of electrons and protons relative to each other. A look at Figure 1-3, which is a radio image of Sagittarius A, shows three spikes shifted to the left and to the right from the hydrogen line. The reason for the shift in spikes from the hydrogen line is due to the velocity of the source of emissions relative to the observer velocity, this phenomenon is called the Doppler Shift. The Doppler Shift is a phenomenon that affects all kinds of waves, resulting from the relative speed between the observer and the source, which leads either to the extension or compression of the wavelength along the way. The apparent change of frequency or wavelength due to this effect is constrained by the following equation:

$$v = \lambda f,$$

where v is the velocity of the wave, λ is the wavelength, and f is the frequency.

However, in the case of EM waves, the speed of the wave is constant in all frames of reference, which is the speed of light ($c \approx 3 \times 10^8$ m s^{-1}). As a result, changes in the wavelength due to the relative motion of the source to the observer will lead to a change in the frequency, the final observed fequency will thus be:[7]

$$f_o = \sqrt{\frac{1+\beta}{1-\beta}} f,$$

where f is the original frequency as emitted by the source, β is the relative speed of the source divided by the speed of light c, and f_o is the observed frequency. Thus, the deviation of the spikes seen in Figure 1-3 from the hydrogen spectral line frequency can be explained by the Doppler shift of the radio waves due to the relative motion of the source and the observer. Furthermore, the thermal motion of the gas within the emitting body will cause a distribution of Doppler shifts around the hydrogen peak, leading our spectral line to lose its sharpness and to look more like a normal distribution.

1.2.1 Interferometry

Angular Resolution

Angular resolution is an important parameter of all telescopes. To understand angular resolution, imagine a car at night with its lights on incoming from far away. If this car is a long distance away, you will not be able to distinguish the two sources of light at the front of the car but rather you will see a big blob of light. However, as the car comes closer, the angle subtended by the two sources of light to your eyes becomes larger, until it is just large enough for your eyes to start distinguishing between the two light sources. That minimal angle at which your eye is able to distinguish between two sources of light defines the angular resolution of your eye. The smaller the minimal angle, the more the angular resolution and the more powerful is the sensor, be it your eyes, an optical telescope, or a radio telescope. From diffraction theory, the minimal angle θ at which an EM telescope can resolve two close objects in the sky is proportional to λ/D_e, where λ is the wavelength of the incoming wave, and D_e is the effective diameter of the aperture area of the receiving device.[8] A more precise formula for the relation would be:

$$\theta \approx 1.22\frac{\lambda}{D_e}. \tag{1.2}$$

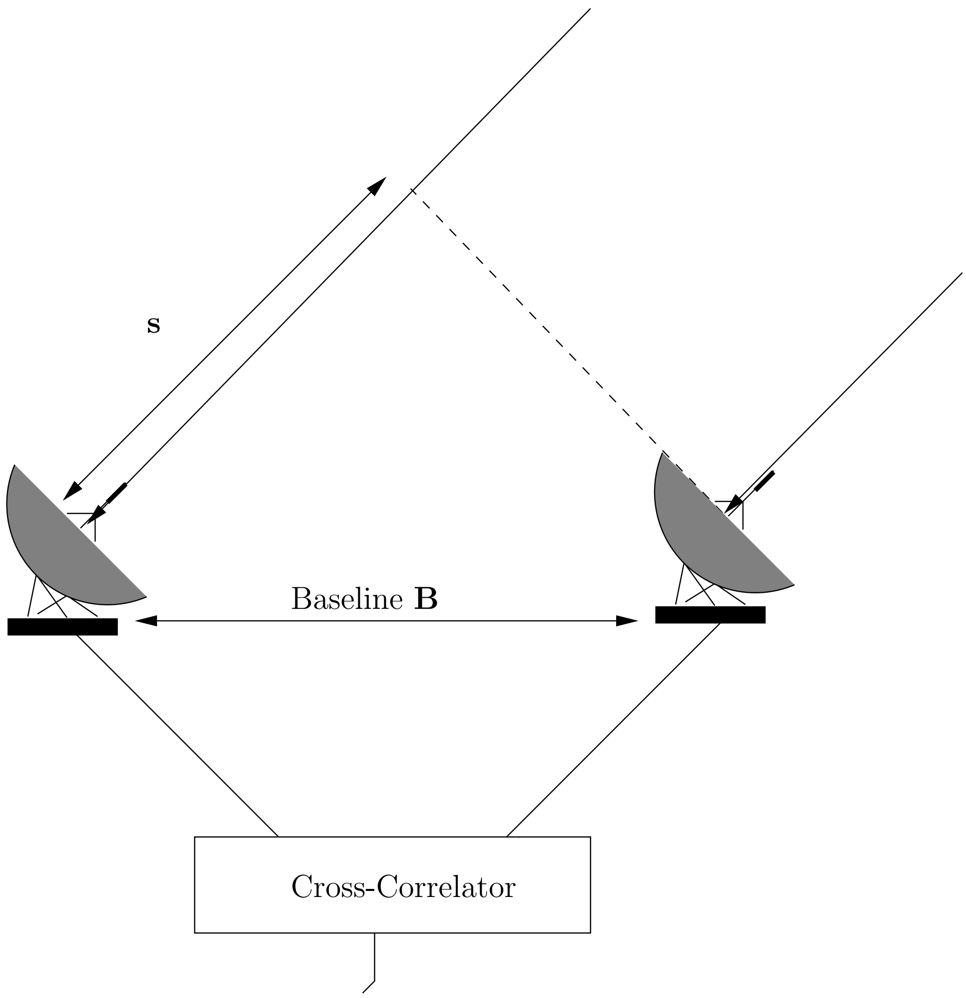
For an optical telescope of, say, a 2-m lense diameter, we would be observing light waves with wavelengthes in the order of 500 nm. If we plug the numbers in Equation (1.2), we can calculate that such a telescope has a θ in the order of 10^{-9} rad, which means that sources like binary stars will have to subtend at least 10^{-9} radians on the telescope lens for the observer to be able to distinguish them from each other. On ther other hand, the radio telescopes used in this project are ~ 2.25 m in diameter and are used to observe radio wavelengths close to 21 cm, which is the wavelength of the hydrogen line. Equation (1.2) would yield a θ of 0.11 rad for such a telescope. This is a very poor angular resolution compared to astronomical standards or even the human eye.

The quest for angular resolution

One solution to the angular resolution problem is to get the aperture area bigger and bigger. However, to get individual radio telescopes to have resolving power resembling that of optical ones we will need a huge telescope with an aperture size in the order of thousands of kilometers. While some radio telescopes were constructed as big as 300 m in diameter (by digging the ground in a parabolic manner and plating it with metal reflectors), they are still not close to the optical resolution power. Larger aperture sizes are both prohibitively expensive and a major engineering challenge.

The solution comes with interferometry. An interferometer is a pair of directional antennas pointed at the same source in the sky. To understand the added resolving power due to intereferometry, we must examine the concept of the power patterns $A(\theta, \phi)$ of each antenna, which defines the receiving power the antenna has as a function of the angle around the main beam axis (which extends from the center of the parabolic reflector through its focal point). When we use two telescopes separated by some baseline vector $\mathbf{B}$, as shown in Figure 1-4, then we could envisage one big aperture with a power pattern equal to the combination of the power patterns of the individual antennas. The final result is that we have a very big aperture with a diameter equal to the separation between the two telescopes and, thus, for large separations (in the order of thousands of kilometers), we can achieve angular resolutions closer to that of optical astronomy. The downside is that the "power pattern" of the interferometer will not reflect that of a large telescope of the same diameter. Instead, the photons collected will only be what the small telescopes inside the inteferometer collect, which leads to a trade-off between "photon collection" and angular resolution that must be taken into consideration when designing an interferometer.

A key component in an interferometer system is the *cross-correlator*. A cross-correlator adds the signals coming from both telescopes and gives an output equivalent to that in the equation described in Figure 1-4. Its output depends on the baseline separation and the signal time lag between the two telescopes. Once the differences in Doppler-shifts are removed the cross-corellator coughs up data equivalent to that coming out of one receiver, making interferometers have the same angular resolution as a radio telescope with a huge aperture.



$$R(\mathbf{B}) = \iint_S A(\mathbf{s}) I_v((s)) \exp\left[i2\pi v \left(\frac{1}{c}\mathbf{B}\cdot\mathbf{s} - \tau_i\right)\right] d\mathbf{s}dv$$

Figure 1-4: A two-element interferometer system. Signal arrives at one of the telescopes before the other, introducing a factor of time delay between data collected at each one. This time delay depends on the separation, **B**, and distance lag **s** which varies with the direction of the source.

Chapter 2

Project Background

2.1 Previous work

The history of my work with radio telescopes dates back to my freshman year in 2002, when I was assigned to work with the first Small Radio Telescope (SRT)[9] that Guilford College's Physics Department has purchased from MIT Haystack Laboratory. The SRT consists of a relatively small dish (~ 2.25 meters) with a consequently low angular resolution ($\theta \approx 0.11$ rad), connected to a control box, which communicated with a Java[10]-coded control interface via an RS232 cable connected to the PC on which the interface software was running. The SRT was designed by the Haystack Laboratory to be a low-cost radio telescope that is intended primarily to teach students about concepts of Radio Astronomy.

However, the SRT project was at its initial stages. While it was a great example of affordable teaching innovation, the software interface was not very well designed for the non-technical users. A look at the original Java source code of the operating interface gave the impression the original programmer was more interested in having a working piece of software rather than a user-friendly interface. I knew Java very well before I got into college, so my physics professor and adviser, Dr. Rex Adelberger, commissioned me to work on the operating system to improve its looks, feel, and user-friendliness to encourage more students to utilize the SRT at their disposal. Consequently, I worked on updating the interface by making telescope-dish pointing possible with just a mouse click rather than the cumbersome process of finding the appropriate control button and entering the desired coordinates numerically.

Furthermore, I added antenna tracking and halting abilities to the control interface which allowed students to entertain their long wait for the slow slewing motors with almost real time updates on the current position of the antenna.

After my work on the control interface, I moved into making software that would analyze the data received and recorded by the Java interface, making human readable graphs of average intensity-vs.-frequency that allowed students to easily see the spectral line emissions of the source without having to go through the cumbersome process of manually averaging data in spreadsheet software, creating graphs, and calculating error bars. This work was used by many students, including a senior student for his thesis project on radio astronomy.

2.2 Beginnings of thesis idea

The real impetus for my project came with the purchase of the second SRT equipped with a better digital reciever that could perform spectroscopy on much finer bin-widths (the old analog receiver had to work with relatively low number of bin-widths per bandwidth observed for it to work reasonably fast). In addition to the second SRT, the first SRT was completely revamped with the same digital receiver. Each telescope now had a Linux operated control box that communicated with the network through an embedded wireless Ethernet bridge. That meant that the box could be moved with relative freedom inside a building space without caring about the location of network ports. Furthermore, both telescope control boxes are equipped with Global Positioning Systems capable of acquiring time information to an accuracy of ± 10 nanoseconds as well as position information within ± 2 meters.

With two state-of-art radio telescopes in hand, it was only natural that I would use my previous experience with SRT and my knowledge of programming and, most importantly, physics knowledge. I wrote a thesis proposal, which was accepted the Physics Department by January 25th, 2006.

Chapter 3

Procedure

3.1 Instrumentation Description

Figure 3-1 describes the basic set of instrumentation of the autonomous SRT. Once

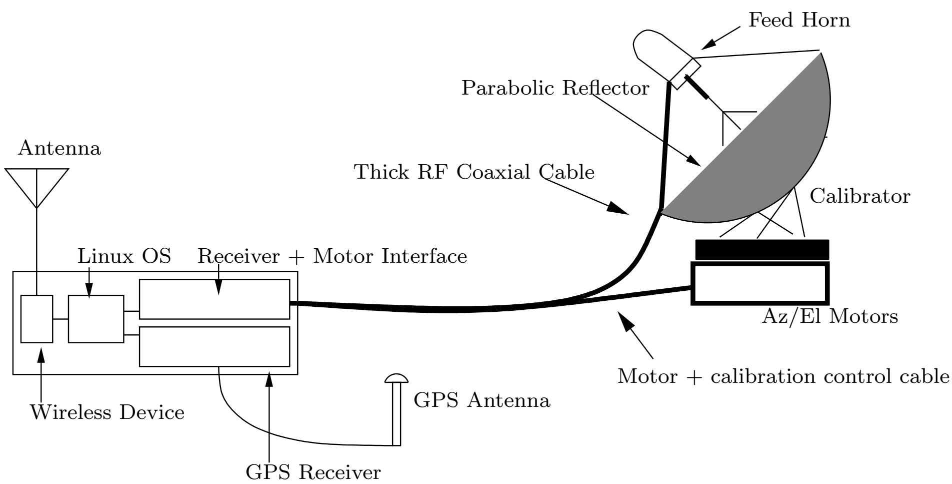


Figure 3-1: Description of Instruments for an autonomous SRT

the SRT is received from its contracted manufacturers, it is in pieces: the parabolic reflector, feed horn stands, feed horn, control box, cables, steel pieces of the telescope stand, Azimuth/Elevation motor, and the GPS dome antenna. The control box consists of a GPS receiver, an RF receiver, Linux[11] operated computer, and a wireless device connected to an antenna. Later on, a truck mounted trailer will be purchased by the physics department and the telescope, along with its control boxes, will be fully mobile. The antenna and the wireless device serve to "bridge" the ethernet connection from the campus network to the telescope so that remote control of the telescope becomes possible.

3.2 Preparations

Before the second SRT was physically set-up, the first that had to be done is to rebuild the already existing SRT on the roof of Frank, calibrate it, and make it work.[12] Then we proceeded to assemble the second SRT[13] inside a lab in Frank to test run its motors and calibrate its software to correctly communicate with the motors. To achieve this, the `srt.cat` file that defines the settings of the control interface had to be changed to let the software know that we are using a CASSI mount motor configuration rather than the AlfaSPID mount that exists on older models.

3.3 Control Procedure

The process of logging into the control boxes of the telescope and running the graphical interface requires some software to be present on the client machine. First, there must be some implementation of the Secure Shell (SSH) client software, as well as an implementation of the latest X11 server. Once the X server is running, open the server ports for remote GUI forwards by issuing the following command:

```
xhost +
```

Then log into one of the telescopes by running SSH and enabling X11 forwarding with a syntax like this:

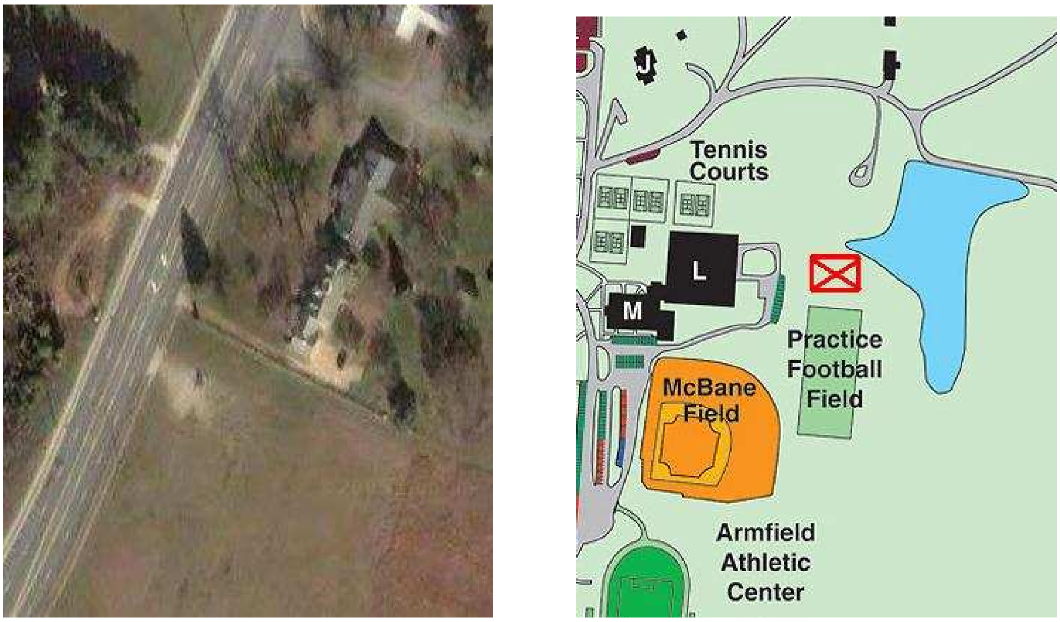
```
ssh -X srtdev@astarte.guilford.edu
```

Once you enter the password and successfully log in, find out the IP address of your client machine, and set the `DISPLAY` variable on the telescope control box. This variable lets the telescope control box know where to direct its GUI window. For example, if the client's IP adddress is `172.20.220.100` then issue the following command at the telescope shell:

```
export DISPLAY=172.20.220.100:0
```

Now you are ready to run the SRT interface by simply changing the directory to wherever the executables are and executing `./srt`

3.4 Physical Set-up



(a) (b)

Figure 3-2: Two candidate locations for the placement of the second SRT. (a) 0.5 miles north of the first SRT in Charles Womack house. (Image from Google Earth) (b) 0.5 miles east of the first SRT next to a football field overseeing the Guilford Lake (position marked with red checked box. Image from Guilford College website)

The first issue pertaining to physical set-up is where to place the SRT telescopes. The old SRT was already placed on top of the Frank Science Building, however, the second SRT was still unassembled and a location was not determined. A location had to be identified that meets the following parameters:

- A level and geologically stable ground to install the antenna base
- Access to electricity
- A weather proof structure to place the control box of the SRT

Consequently, two candidate locations were identified as meeting those conditions. As Figure 3-2 shows, one location was chosen on Guilford College campus and another was found in the backyard of the house of a Guilford physics department alumni, Charles Womack. The campus location identified had a small abandoned hut close to it, in which the control box could be placed, and it had access to electricity. The downsides, however, were its proximity to a football field which exposes the parabolic reflector to accidents and dents (which reduce the efficiency of the antenna, where the efficiency is proportional to the sum of the square of the deviations from the ideal parabolic surface). Womack's backyard location had the disadvantage of poor sky visibility due to the abundance of trees and buildings in the vicinity. However, since the plan is to establish a mobile SRT unit, the Womack house was chosen as a temporary –safe– place for the second SRT while it is assembled and the interferometry system deployed.

3.4.1 Problems and Solutions

Many problems were faced during the physical set-up, not least among them was having many parts of the telescope missing, including nuts and bolts, feed horn covers. Most of those problems were solved by improvising some fixes (used duct tapes in place of screws to place the feed horn cover) or readily buying the missing items from Home Depot. During the physical set-up, one of the RF connectors on the feedhorn accidentally hit the ground and broke, we fixed this by putting an aluminum foil over it and duct taping it tight. The aluminum foil worked well to shield the RF connector and the feed horn seems to be working.

3.5 Wireless Communication Set-up

When originally purchased, the intenion was to use the wireless configuration already installed in the control box to make the connection over the half mile between the base station and the mobile unit. The company that manufactured the wireless devices claimed that the devices had a range of 15 miles over the a clear line of sight (LoS), so we calculated that 0.5 miles in less than ideal LoS conditions would be good enough for the wireless connection to occur. This was not true, as we found the connection strength was too weak for any meaningful communication between the base station and the mobile unit to occur. Thus, we purchased two high gain antennae.

One antenna was omnidirectional with an 8 dBi gain, which means it is capable of emitting radio waves with equal gain over 360° of azimuth. However, it only had that gain for a 15° main lobe around the horizontal axis, as seen in Figure 3-3. Those two properties of antennae (i.e. azimuth and elevation lobe angles) are called horizontal and vertical beam widths of antennae, respectively. They refer to the angle subtended by the main power lobe on the horizontal and vertical planes. The general power pattern, however, has many side lobes, which are usually much weaker and are, thus, ignored. The azimuth power pattern of the omnidirectional antenna meant

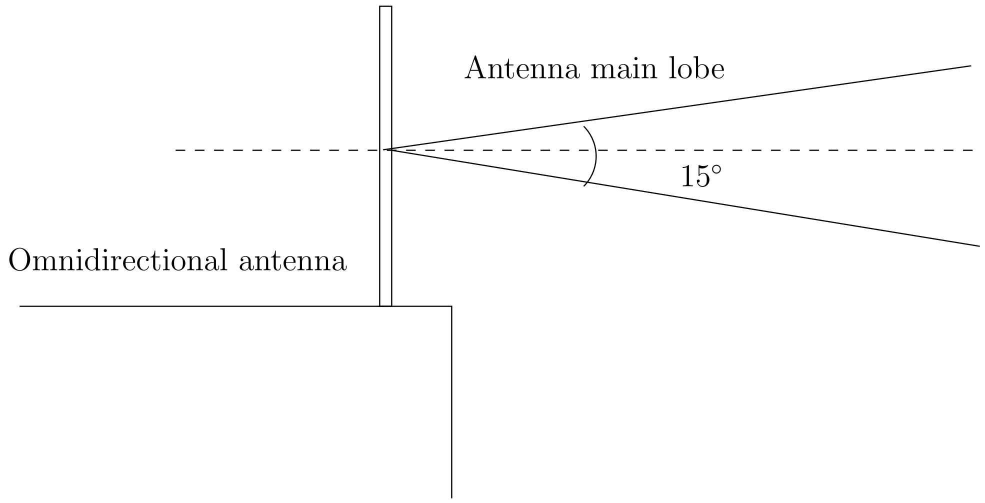


Figure 3-3: The main beam pattern of an omnidirectional antenna.

that it need not be moved or redirected whenever the mobile unit changed location, which makes it an ideal choice for the base station. However, the elevation beam width meant that there must be a certain distance/altitude configurations for the mobile station before a connection could be made, since any LoS between the two antenna terminals must fall within the elevation/azimuth antenna lobes.

The other antenna purchased is a 13 dBi gain Yagi, which is a very directional antenna. It has equal vertical and horizontal beam widths of 30°, which meant that, if it is installed on the mobile unit, it will have to be carefully redirected towards the omnidirectional antenna whenever the unit is moved.

Besides power pattern considerations, the line of sight between two antennae must be as free from obstacles as possible for a reliable wireless communication setup. More

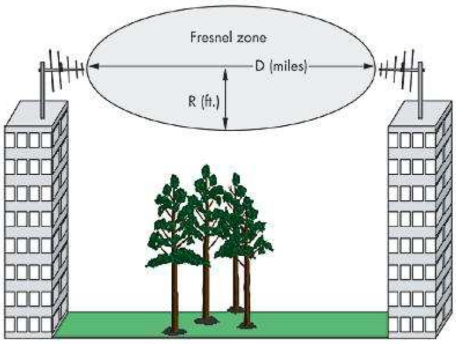


Figure 3-4: A diagram of the Fresnel Zone ellipsoid. It is geometrically defined by the maximum cross-sectional radius r and the distance between the antennae D. (Image from isa.org)

specifically, RF engineers devised a standard for assessing wireless communications called the "Fresnel Zone" clearance. The Fresnel zone is an ellipsoid that follows an EM circular diffraction pattern from the apertures of terminal antennae. As a rule of thumb, RF engineers try to keep the obstacles in the Fresnel zone to less than 20% of the volume of the ellipsoid. As can be seen from Figure 3-4, the Fresnel ellipsoid can be geometrically defined by its length D between the terminal antennae, and by the maximum cross-sectional radius r. The relationship between r in feet, D in miles and the frequency at which the antennae operate, f_{GHz}, in gigahertz is:

$$r = 72\sqrt{\frac{D}{4f_{GHz}}}$$

The antennae purchased operate at 900 MHz (0.9 GHz) and are separated by roughly 0.5 miles, which means that the maximum cross-sectional radius of the Fresnel ellipsoid is about 26 feets ($\approx$ 8 meters). So, the goal is to engineer a LoS which has at least 7-8 meters of clearance from most obstacles on the way. To achieve this goal we used a 4.5 meters antenna mast for the omnidirectional antenna and mounted it on top of the Frank Science building (for a total height of 34.5 meters above the ground) and raised the yagi antenna 5 meters over the remote unit. The purpose was to get the LoS as high as possible and hence avoiding tree and building tops.

3.5.1 Problems and Solutions

The wireless devices had manufacturing deficits that rendered my connection unreliable, albeit the devices showed full link quality. The deficit made the device randomly lock-up and stop transmitting network packets over the wireless connection, which caused the telescope to be unreachable from the college network. The only way to fix that was to go over the remote site and turn off the control box and back on. Before we realized that this problem was due to a manufacturing deficit (or a "bug" in computer industry jargon) in the wireless terminals, we hypothesized that the problem was due to a chain of events that led the Linux operating system to stop responding to network calls. The hypothesis was that the wireless link quality sometimes fluctuates and drops due to the reflection of RF waves of the metal tops of passing cars and electronic interference, which leads to a temporary outage of network access. The Linux operating system would then, hypothetically, think the network is no longer available and will turn off its network facilities. The solution would, then, be to program a script that continuously checks the status of the network devices and, if they are down, turn them back on. However, this did not fix the problem and we continued to exprience sudden cut-offs of the connection to the remote telescope.

Consequently, we contacted the company that manufactures the devices, MaxStream Corporation, and described the problem to them. They acknowledged that old models of the wireless device do experience these kinds of communication cut-offs due to a deficit in the firmware[14] and they exchanged the devices for us. However, the problem persisted and the only solution in hand at that time was still to go to the remote site and turn the control box off and back on.

However, the real reason behind turning the control box off and back on is to cut the power off from the wireless device and then restart it. So we thought one way to solve this problem is by designing a configuration that will allow the computer to turn the wireless device on/off automatically whenever network outages occur. One idea was to use the power facilities of the USB[15] ports on the computer inside the control box.

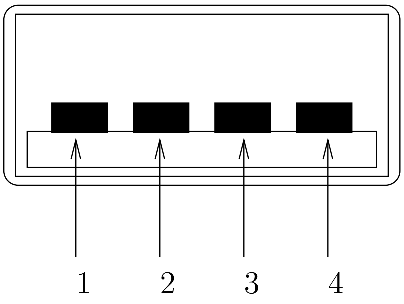


Figure 3-5: A schematic of a standard USB port, the black rectangles numbered 1-4 are metallic terminals that transmit power and data. Terminals 1 and 4 are power terminals with a maximum power output of 2.5 W at 5 volts.

Two of the metallic terminals on a standard USB port transmit a maximum power of 2.5 W at 5 volts (see Figure 3-5), which could be used to power the wireless device, whose maximum power consumption is 1 W. So, we cut a USB cable and stripped the wires corresponding to the power terminals and connected them to a power connector plug. Indeed, once the new cable (see Figure 3-6) was connected to the USB port it successfuly powered the wireless device. The next step would then be trying

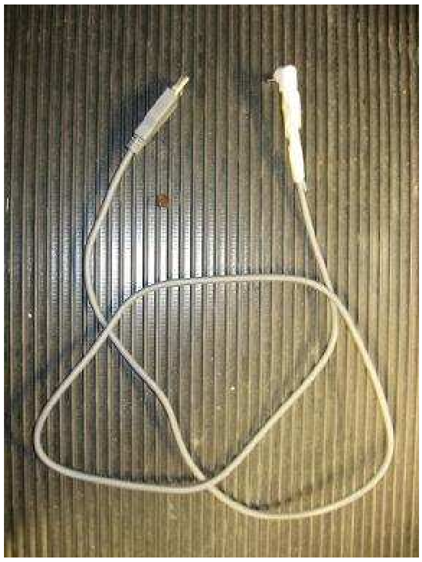


Figure 3-6: The new mutant cable! With a USB terminal and a standard power connector terminal.

to write a script that would turn the USB port on and off. To achieve that, we researched the Linux operating system's USB interface and found an experimental kernel module that would allow for USB suspend/resume and installed it (see section

3.6). We made a script that utilized this module. However, we found out that the module can only control the power output of the device if the USB hardware natively supports it, which was not the case in our computer. Therefore, although the USB-power cable worked fine to power up the device it was not very useful due to the fact that we could not control the power output of the USB port, which means that we will not be able to reset the wireless device programmatically through this “mutant cable”.

The next solution we thought of was to purchase a device called a USB relay, which is essentially a digital programmable logic controller, allowing the user to read and write customized digital input/output to any other device and open and close analog circuits. Thus, the USB relay could be set up to programmatically turn off the wireless device by opening the circuit connecting it to the power source and turn it back on by closing the circuit (see Figure 3-7).

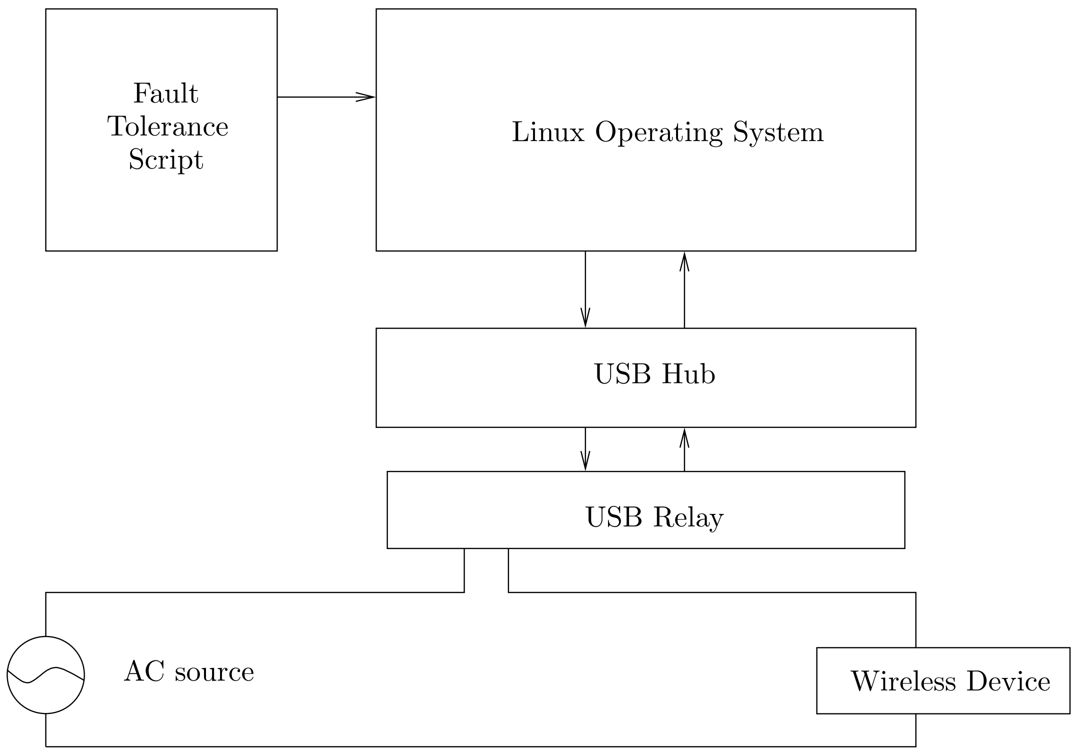


Figure 3-7: Circuit diagram for a USB relay system to control the power input of the wireless device. A fault-tolerance script commands the operating system to talk to the USB relay (through the USB hardware hub) and issue it a command that opens/closes the circuit that powers the wireless device.

We purchased the ADU200 USB relay from OnTrak Control Systems, which does the job required. However, before the USB relay was set-up, many changes to the operating system's kernel modules had to be done, and the outdated Linux driver[16] had to be updated up to the latest USB specifications for the device to work successfully. The ADU200 detailed set-up procedures can be read in section 6.1. A script was written to control the USB relay, which in turn controls the wireless device power input, and a fault-tolerant communication system, able to automatically redeem itself after wireless device failures, was finally established.

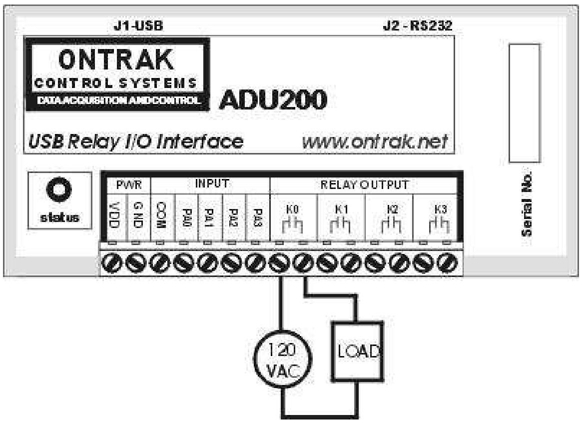


Figure 3-8: A diagram of the ADU200 USB relay device, with the circuit powering the device shown connected to the K0 port on the ADU200. (Image from ontrak.net)

3.6 Operating System Set-up

The Linux operating system set-up had to do with the wireless communications set-up: many "kernel modules" had to be added to the operating system for the fault-tolerance system discussed in section 3.5.1 to work. Kernel modules are small programs that activate certain features of the operating system and could also function as device drivers. The first kernel module we added is the USB suspend/resume module which was intended for use to turn the power input of the wireless device on/off through the USB-power cable we made. The other kernel module added was the device driver for the ADU200 USB relay. Many other existing modules had to be changed as well to enable the Linux operating system to adapt to the ADU200 module. The detailed set-up process for the ADU200, including the kernel modules set-up, is detailed in section 6.1.

For the USB suspend/resume module, it can be added through the following processes: downloading the Linux kernel sources, changing the kernel configuration, recompiling the source, and then rebooting using the new kernel:

- Downloading latest Linux sources through the `emerge` utility[17]:

  ```
  emerge --sync
  emerge -a gentoo-sources
  ```

- Letting the system know where the new Linux sources are. If you have downloaded Linux version `linux-2.6.16-gentoo-r3`, then the sources directory will be `/usr/src/linux-2.6.16-gentoo-r3`. To let Linux know the location of the new sources, the following commands should by executed:

  ```
  cd /usr/src
  ln -sfn linux-2.6.16-gentoo-r3 linux
  ```

  The reason the above commands work is because Linux looks for its kernel sources in the directory `/usr/src/linux` by default. That directory is not real, but rather a symbolic link to where the sources really are. Changing where the symbolic link points to (by issuing `ln -sfn` command) will let Linux know where the new sources are.

- Reconfiguring the kernel configuration to add the USB suspend/resume module and automatically recompiling the kernel and installing it:

  ```
  genkernel --menuconfig --udev --bootloader=grub all
  ```

  This will load a menu that will allow you to add/remove the needed kernel modules. Going through "Device drivers" menu item and then"USB support" will take you to the USB modules options. Selecting the "USB selective suspend/resume and wakeup (EXPERIMENTAL)" option will let the kernel know that we want that module added the next time we recompile.

- The `genkernel` utility will proceed to recompile the kernel using the modules specified.

The previous steps will allow for the direct USB-power solution detailed in section 3.5.1, but will still need native hardware support for turning the power off/on. Once the hardware support is there, a USB port could be turned off by simply issuing the following command from the Linux shell:

```
echo  "3" >> /sys/bus/usb/devices/3-1/power/state
```

where `3-1` is the USB port ID (listing the contents of `/sys/bus/usb/devices/` will allow you to see all the USB ID's available on the system).

Chapter 4

Results

4.1 Communication Reliability

Communication reliability is tested on the short run by calculating the average number of packet loss in normal user session with the telescope and in a heavy load session. The tests were run by executing the `ping` utility for the normal load and invoking the same utility again with `ping -A`, which turns it into the "adaptive" mode that simulates a heavy communications load with the remote SRT. Each test took 2 minutes and was run 10 times on each telescope control box; the results are detailed in the following table:

Telescope	Normal Load Packet Loss (%)	Heavy Load Packet Loss (%)
Astarte	(0 ± 1)	(2 ± 1)
Ashtarut	(0 ± 1)	(1 ± 1)

Table 4.1: Packet loss results for normal/heavy network loads

The results in Table 4.1 show that the wireless network performance is almost perfect during normal loads while it deteriorates to 1-2 % for each telescope during heavy loads. It is interesting to notice that the packet loss statistics are the same for both telescopes even though Astarte is located 0.5 miles from its base station. Furthermore, those packet loss ratios are practically invisible to the user, since the SSH utility can resend all lost packets transparently.

4.2 Data collected from Celestial Sources

Following our success with the communication system we took radio pictures of many objects in the sky with our telescopes, Figures 4-1 and 4-2 are two pictures taken by the telescopes:

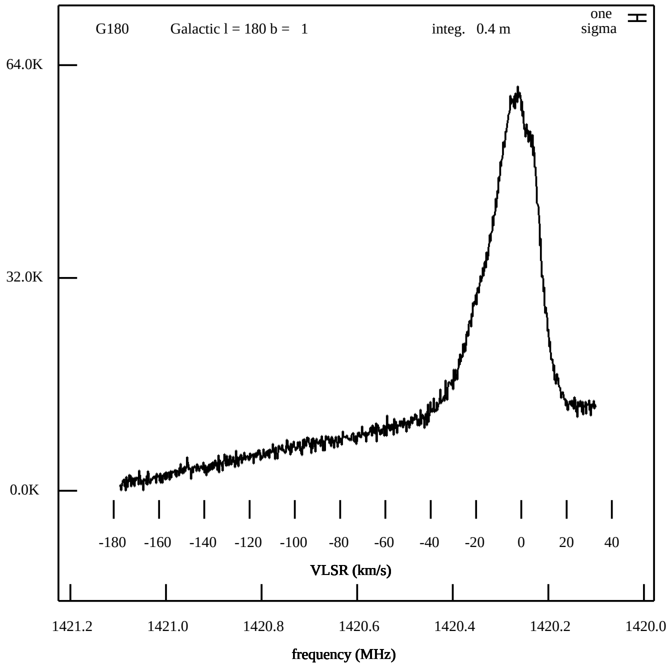


Figure 4-1: Spectrum from the center of the Milky Way galaxy taken with Ashtarut. Notice the hydrogen line peak which is Doppler shifted from its original 1420.41 MHz frequency due to the orbital motion of the earth around the center of the galaxy.

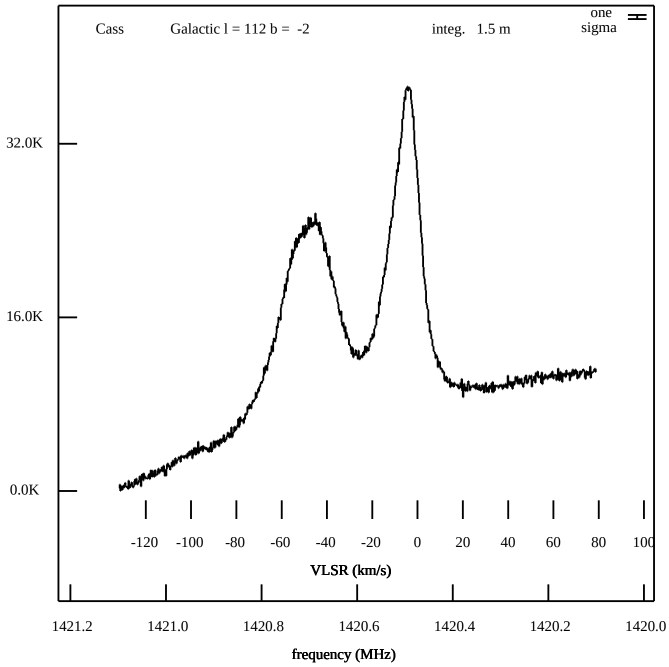


Figure 4-2: Spectrum from the Cassiopeia constellation taken with Astarte. Notice the two adjacent hydrogen line peaks with different Doppler shifts due to the relative motion of the source and the telescope. Unlike Sagittarius A's radio picture (see Figure 1-3), whose adjacent peaks are relatively similar, the peaks in the above image have different intensities and different line broadening patterns. This means that the peaks are probably due to two sources in the same direction with different velocities, rather than due to the rotational motion of a source (as in the case of Sagittarius A.)

Chapter 5

Future Work

After the completion of this thesis project, the ground was laid for the establishment of a full fledged flexible baseline interferometer with automated interferometry calculations. For this to be achieved, the following is a brief description of the steps needed.

5.1 Flexible Baseline Interferometer

Upon the purchase of a trailer, Astarte could be easily moved from the backyard of Charles Womack's house into the trailer along with its control box and antennas (see Figure 5-1. If a power generator is present, the system can be moved anywhere within a theoretical range of $\sim$ 15 miles dictated by the wireless communications antennas' manufacturer's specifications. Once the trailer is moved to its new position, the yagi antenna will need to be redirected to coincide the power pattern of the yagi with that of the omnidirectional one. Once a good signal quality is established the system will be ready for use from any computer on campus.

5.2 Automated Interferometry System

An automated interferometry system could be established by changing the source code of the already existing software or by making data recording from each telescope and feeding it into another software that would use the time tags and GPS location information to cross-correlate the data-points together.

Figure 5-1: An example of a trailer mounted telescope. (Image from MIT Haystack www.haystack.edu)

Chapter 6

Technical Notes

6.1 ADU200 Detailed Set-up Procedure

- The first step towards installing the ADU200 device is downloading the latest Linux kernel sources as described in section 3.6.

- Once the first step is done go to the directory `/usr/src/linux/drivers/usb/input/` and edit the file `hid-core.c` with your favorite text editor (i.e. `vi` or `nano`, etc.). And look for the following line:[18]

  ```
  { USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100,
   HID_QUIRK_IGNORE },
  ```

  and add the following lines after it:

  ```
  { USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100 + 20,
  HID_QUIRK_IGNORE },
  { USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100 + 30,
  HID_QUIRK_IGNORE },
  ```

  Similarly, look for the following line:

  ```
  { USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100 + 100,
  HID_QUIRK_IGNORE },
  ```

  and add the following lines after it:

```
{ USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100 + 108,
HID_QUIRK_IGNORE },
{ USB_VENDOR_ID_ONTRAK, USB_DEVICE_ID_ONTRAK_ADU100 + 118,
HID_QUIRK_IGNORE },
```

Those steps let the Linux USB-HID driver identify the ADU200 device (and other Ontrak devices) with its vendor ID, which is already stored elsewhere in the USB-HID driver.

- Go to the directory `/usr/src/linux/drivers/usb/` and edit the file `Makefile`. Look for the following lines:

```
obj-$(CONFIG_USB_USS720)       += misc/
obj-$(CONFIG_USB_PHIDGETSERVO) += misc/
obj-$(CONFIG_USB_SISUSBVGA)    += misc/
```

and add the following after them:

```
obj-$(CONFIG_USB_ADUTUX)       += misc/
```

This tells the Linux GNU C compiler, `gcc`, to configure the ADU200 driver and to look for it in the `/usr/src/linux/src/drivers/usb/misc/` directory.

- Copy the `adutux.c` file from the Thesis CD into `/usr/src/linux/src/drivers /usb/misc/` directory. This file is the modified source code for the Linux device driver originally supplied by John Homppi (see footnote 18), the original source code is out of date and it violates the new USB specifications. Mainly the `usb_class_driver adu_class struct` and `usb_driver  adu_driver struct` had to be updated to comply with the new specifications.

- Edit the file `/usr/src/linux/drivers/usb/misc/Kconfig` and look for the following paragraph:

```
config USB_LEGOTOWER
        tristate "USB Lego Infrared Tower support (EXPERIMENTAL)"
        depends on USB && EXPERIMENTAL
        help
          Say Y here if you want to connect a USB Lego
          Infrared Tower to your computer's USB port.

          This code is also available as a module ( = code
          which can be
          inserted in and removed from the running kernel
          whenever you want).
          The module will be called legousbtower. If you
          want to compile it as
          a module, say M here and read
          <file:Documentation/kbuild/modules.txt>.
```

and add the following paragraph after it:

```
config USB_ADUTUX
        tristate "ADU devices from Ontrak Control Systems
        (EXPERIMENTAL)"
        depends on USB && EXPERIMENTAL
        help
          Say Y if you want to use an ADU device from
          Ontrak Control Systems.
          To compile as a module, choose M. The module
          will be called adutux.
```

Make sure there is a line feed before and after the paragraph above (make it look similar to all the other paragraphs in the file.) What this step does is add the ADU200 drivers to the kernel configuration menu options along with a description.

- Edit /usr/src/linux/drivers/usb/misc/Makefile and look for the following lines:

```
obj-$(CONFIG_USB_LD)             += ldusb.o
obj-$(CONFIG_USB_LED)            += usbled.o
obj-$(CONFIG_USB_LEGOTOWER)      += legousbtower.o
```

  and add the following after it:

```
obj-$(CONFIG_USB_ADUTUX)       += adutux.o
```

  This allows the gcc compiler to know where to output the executable object that result from compiling adutux.c.

- Recompile the kernel as described in section 3.6, make sure the "ADU devices from Ontrak Control Systems (EXPERIMENTAL)" option under the "USB support" menu is selected by highlighting it and clicking M.

- Restart the system with the new kernel, edit the /etc/init.d/net.eth0 and look for the following lines:

```
start() {
        # These variables are set by setup_vars
        local status_IFACE vlans_IFACE dhcpcd_IFACE
        local -a ifconfig_IFACE routes_IFACE inet6_IFACE
```

  and add the following after it:

```
        mkdir /dev/usb
        mknod adutux0 c 180 179

        /jdk1.5.0_06/bin/java -cp /root/ TurnOn
```

This will create a node in the USB structure tree for the ADU200 and run the Java application that orders the ADU200 to close the circuit for the wireless device.

- In the same file, look for the following lines:

```
stop() {
        # Call user-defined predown function if it exists
        if [[ $(type -t predown) == function ]]; then
                einfo "Running predown function"
                predown ${IFACE}
        fi
```

and add the following line:

```
        /jdk1.5.0_06/bin/java -cp /root/ TurnOff
```

This will run the java application that orders the ADU200 to open the circuit powering the wireless device.

- Copy the `TurnOn.java` and `TurnOff.java` files from the Thesis CD into your `root` directory and compile them (make sure you have the latest JDK from `java.sun.com`)

- Copy the script `test` from the Thesis CD and add execution rights for the script with `chmod 700 test`, edit your crontab and schedule this script to run every 2 minutes (or whatever time you deem necessary) by typing the following command:

```
crontab -e
```

and add the following line:

```
*/2     *       *       *       *       /root/test
```

- The fault tolerance system is now ready to run!

Endnotes

[1]Giancoli (2000) and Rohlfs and Wilson (2004)

[2]A blackbody is an ideal perfect absorber of EM radiation, it follows from the laws of electrodynamics that a blackbody is also a perfect radiator (Griffiths, 1998)

[3]Kraus (1986)

[4]A spectroscope is a device that breaks an incoming EM wave into its spectral components. A glass prism, which separates white light into a rainbow of spectral components is a simple spectroscope and one of the most familiar ones.

[5]Kraus (1986)

[6]Thompson et al. (2004) and Kraus (1986)

[7]Giancoli (2000)

[8]Rohlfs and Wilson (2004)

[9]For more information refer to MIT Haystack Laboratory homepage about SRT at: `http://web.haystack.mit.edu/SRT/index.html`

[10]Java is "an object-oriented programming language developed by James Gosling and colleagues at Sun Microsystems in the early 1990s. Unlike conventional languages which are generally designed to be compiled to native code, Java is compiled to a bytecode which is then run (generally using JIT compilation) by a Java virtual machine." (Source: online Wikipedia.org article)

[11]Linux/GNU is a UNIX-like open source operating system, which allows developers to look into its source code freely and change it according to their needs.

[12]Jonathan Poe significantly contributed to those efforts.

[13]William Hahn significantly contributed to the assembly of the second SRT in the lab.

[14]A firmware is a microprogram that is stored on a programmable Read Only Memory (ROM) on a computing device. The firmware serves to manage the functioning of the device.

[15]An external peripheral interface used for communication between a computer and external peripherals over a special cable.

[16]a driver is a small piece of software that allows an operating system to talk to a device.

[17]`emerge` is a utility that is distributed with Gentoo Linux distribution, which is the one installed on the control box. It might not be found on other Linux distributions.

[18]Many of the steps described in this section are inspired by John Homppi's page on Linux Drivers for Ontrak ADU drivers. Changes were made to ensure the compliance of the code with the latest `libusb` specifications and the installation procedure was discrened into separate more detailed steps for easier installation. Access John Homppi's page at: `http://users.tyenet.com/jhomppi/index.html`

Acknowledgments

This thesis project was generously funded by the Womack Family Physics Research Fund. I would like to express my gratitude to Winslow Womack for supporting this fund and to Charles Womack for generously allowing us to put the second SRT in his backyard.

I would like to thank the distinguished faculty of the Guilford College physics department: my thesis and academic adviser Professor Rex Adelberger, for his consistent support and encouragement when I most needed it. Professors Thom Espinola, Don Smith, and Steve Shapiro for being a part of one of my most entertaining and rewarding educational experiences. I also would like to thank Professor Robert Williams, my thesis committee member and my Economics professor, for his feedback, support and understanding throughout my work on this project.

I would like to thank my family, my parents: Maha Abi Issa and Ibrahim Atiani, for their limitless support and for raising me up to become what I am now. My sister: Serin Atiani, for all the love and support through all those frustrations and difficulties of a scientific research project.

I would like to thank my friends, those who woke up in early mornings on weekends to lend me their cars or give me a ride: Yacoub Saad, William Hahn, and Bryan Cahall. Those who helped and occasionally endangered their lives with me for the sake of science: Charlie Stroup, Jeff Shamp, and Jonathan Poe. And all my other friends who stood by me through college and kept me from going insane.

My sincerest gratitude to all of them.

Bibliography

Giancoli, D. C. (2000). *Physics for Scientists and Engineers*. Prentice Hall, 3rd edition.

Griffiths, D. J. (1998). *Introduction to Electrodynamics*. Prentice Hall, 3rd edition.

Kraus, J. D. (1986). *Radio Astronomy*. Cygnus-Quasar Books, 2 edition.

Rohlfs, K. and Wilson, T. L. (2004). *Tools of Radio Astronomy*. Springer-Verlag Berlin Heidelberg New York, 4 edition.

Thompson, A. R., Moran, J. M., and Swenson Jr., G. W. (2004). *Interferometry and Synthesis in Radio Astronomy*. Wiley-VCH, 2 edition.