Searching for extra-terrestrial intelligence

IS THERE ANYBODY OUT THERE?

JENNY BAILEY

Go out on a dark, cloudless night at look up at the stars. There could be someone looking back, although of course, we couldn't see them. Their image would be history, as light from a planetary system 10 light years away would take 10 years to arrive here. A space vehicle using our current technology would take 300,000 years to get the 4 light years distance to Proxima Centauri, our nearest neighbour. So how can we find out whether or not we are

alone in the universe?

Electromagnetic radiation (light, radio waves, x-rays) is the fastest communication medium known to us at the moment. It is therefore our best hope for contact with another intelligent species.

MOther Current SETI projects

SETI@home

Launched on 17 May 1999 SETI@home uses data from the Arecibo Radio Telescope as part of Project SERENDIP.

Project Phoenix

Observations began in February 1995. Currently conducting a targeted search from Arecibo, with Jodrell Bank, Manchester checking candidate signals.

Listening for a Radio Transmission

In 1937 Nicola Tesla suggested using radio for extraterrestrial communication. Since then there have been a number of radio based SETI (Search for Extra Terrestrial Intelligence) projects to search for radio transmissions from distant solar systems, starting with Frank Drake's *Project Ozma* in 1960. Every new project boasts better sensitivity and more advanced search algorithms. *Project Argus* proposes 5000 small dishes operated by Amateur Radio Astronomers, each dish pointed in a different direction. We are developing Linux software for Project Argus stations.

Each Project Argus station is searching for a beacon signal targeted towards our solar system: a transmission powerful enough to be heard on the earth that is both narrow band and coherent and which therefore stands out above the galactic noise. Unlike Hollywood's portrayal of SETI people, we do not spend our time next to a radio telescope wearing headphones and listening to white noise. Automated searching for a signal in noise is a job for a computer.

Searching for a Needle in a Haystack

Project Argus assumes that an Alien Civilisation is trying to signal us using a radio beacon. We must make some logical deductions about this beacon based on physics, which we believe to be universal. We could try to detect wideband signals such as accidental radio leakage from, say, their domestic television transmitters or equivalent. These signals would be modulated and therefore spread over a wider band. They would look like galactic background noise.

If an alien intelligence is trying to contact another civilisation they might send out a narrow band signal to likely looking planetary systems. To hear this we must have a fully functioning SETI station pointed in the right direction, at the correct frequency, with an appropriate polarisation.

Direction

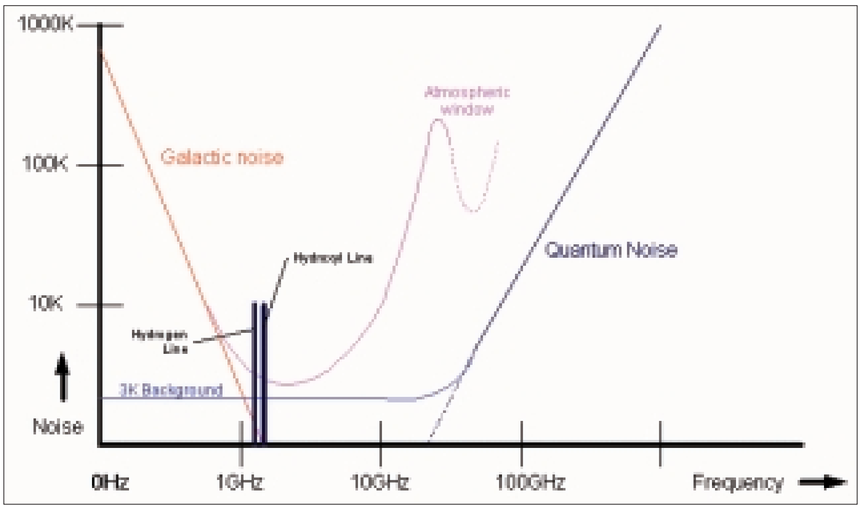
Some SETI projects are "targeted searches" where the most likely stars are monitored continuously for many hours. The telescope must track the star as it rises and sets. This type of search tend to be favoured by well-funded projects because of the cost, and noise, of a constantly tracking aerial arrangement.

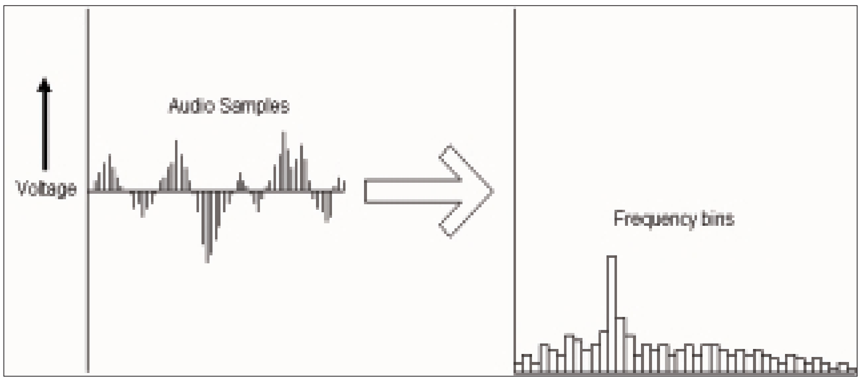
The other main search type is the "all sky search" where the radio telescope remains fixed in one position and the sky moves past. A small dish – 4m diameter – has a wider beamwidth (will see more sky) than a 100m dish, albeit with less sensitivity. To cover all the sky, all the time, over 5,000 small radio telescopes are needed, distributed all over the earth. This is the goal of Project Argus as managed by The SETI League.

Polarisation

We are trying to detect a signal which has been sent using an unknown polarisation (a term that describes the plane of the received radio waves.) To receive the signal we must be using a similar polarisation at our receiver or the signal we want will be severely attenuated.

The aerial that feeds the dish can be linearly polarised: it can use horizontal polarisation, vertical polarisation, or something in between. There are also more complex polarisations such as circular (clockwise and counter-clockwise) which are used for space communication.





[top]
Fig. 1: The best frequencies for listening to signals from space are between 1 and 10GHz

[above]
Fig. 2: An FFT routine converts audio samples into frequency bins

Frequency

The Earth's atmosphere is only transparent at certain frequency bands. The microwave window between 1GHz and 10GHz is both relatively noise free and low-loss through the atmosphere (see Figure 1.)

A signal heard throughout the Universe is the Hydrogen line. This is a narrow band, non-coherent signal generated by interstellar hydrogen gas, the most abundant molecule in the Universe. This frequency also happens to be within one of the "frequency windows" in the atmosphere, so we choose to look for signals in the band around the Hydrogen Line. The band around the Hydrogen Line also has the advantage of being protected from transmissions, so there should not be any man-made signals on these frequencies.

However, there is no "incorrect" frequency for SETI. Other projects look around frequencies such as 4461MHz (p * hydrogen line) and other mathematical constants that an intelligent civilisation might use.

Bandwidth

The sensitivity of a receiver is proportional to its bandwidth. A narrow band receiver will hear less noise than a wideband receiver, and therefore a signal will stand out better.

However, the less bandwidth we look at, the less chance we have of hearing a signal, because we don't know exactly what frequency to listen on.

Doppler shift:

This is the change in apparent frequency observed by the listener, depending on whether the transmitter is coming towards you or moving away from you. The same effect results in the change in tone of a car horn that is sounded as the car drives past.

Therefore, we need to look at as much bandwidth as possible.

The answer to these two contradictory requirements is to use a computer to split a large bandwidth into lots of narrow "frequency bins" and look for a signal in each one. The function to split the bandwidth is called a *Fourier Transform* and a cunning hack to speed up processing makes it a *Fast Fourier Transform* – FFT. When you double the size of a Fourier Transform you square the number of multiplications. When you double the size of a Fast Fourier Transform you approximately double the number of multiplications.

The computer sound card samples the audio and an FFT routine converts these samples into frequency bins (see Figure 2.)

A little FFT calculation

The sample rate is the number of samples (bytes or words) per second. With a SoundBlaster card or compatible the maximum is 44kHz. The maximum frequency we can detect at this sample rate is equal to the sample rate divided by two (also known as the Nyquist frequency.)

The sample size is the number of samples that we process in one block. Then the bin size = 2 * sample rate/sample size.

- Audio bandwidth from receiver = 20KHz (DC .. 20KHz).
- Maximum Sample rate (SoundBlaster 16) 44KHz
- Data rate (assuming 16 bit samples) = 88Kbytes/second
- Target bin size = 10Hz
- Bin Size = sample frequency/FFT length = 44000/4096 = 10.7Hz
- Number of FFTs per second = sample frequency/FFT length = 10.7 FFTs per second.
- Maximim time for each FFT cycle = 93ms.

Why CPU Power is important

Calculating an FFT involves many complex-number calculations. Narrow bin size – and therefore better sensitivity – will result from more CPU time. Therefore any SETI program should minimise CPU use elsewhere (such as for the display). So we now have the situation where the sensitivity of a SETI station can be improved by optimising the FFT routine!

A soundcard (such as a SB16) can record two channels simultaneously: stereo. Therefore one sound card can support the output from two receivers as long as the CPU can process fast enough. Linux can support more than one soundcard.

There is, however, a limit to narrowing the bin size beyond which valid signals might be missed. Narrow band signals will start to spread as they pass though the Interstellar Medium. This alone will limit the minimum bin size to greater than 0.1Hz.

Bin levels tend to be averaged over a number of FFT calculations to further enhance sensitivity. During this time the signal can move from one bin into an adjacent one due to **Doppler shift**. The averaging may then lose the signal. Doppler shift occurs because of the changing relative velocities of the transmitting station (a rotating alien world orbiting a distant Sun) and the receiving station.

Programs like SETI@HOME have the time to "chirp" the data, i.e. they move the frequency of the received data to compensate for Doppler shift. We can't predict the Doppler shift and so SETI@HOME FFTs and averages to match the many possible Doppler shifts up to ±10Hz. This is a luxury you only have when processing off-line.

Why Linux?

The SETI software runs 24 hours a day, 7 days a week, usually unattended. A robust operating system with the ability to handle large data throughput and gigabyte logfiles is an essential requirement. Another reason for using Linux is that documentation on, for instance, programming the soundcard is widely available on the Internet, in various *HOWTOs* and excellent O'Reilly books. If all else fails, then you can just dive inside the kernel source and see what is going on yourself.

Once you have discovered the *sndconfig* command or its counterpart in your chosen distribution, installing a sound card for Linux couldn't be easier. The */dev/dsp* and */dev/mixer* give consistent interfaces for many types of sound card.

Why Open Source?

The algorithms used to detect signals can be made very sensitive to catch even the faintest signal but many false alarms, or they can be set insensitive to minimise false detections and possibly miss real signals. Once the framework for SETI detection software is in place, the benefit of open source is that others can add their own improved search algorithm.

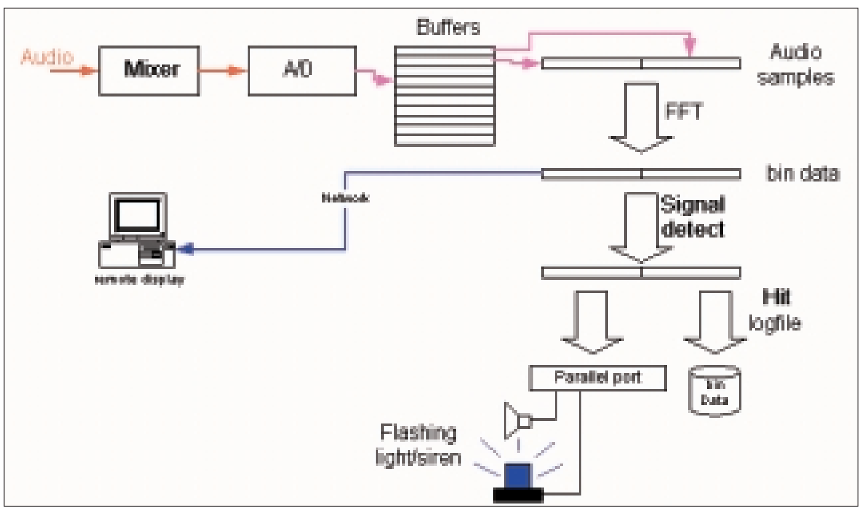


Fig. 3: The components of a Linux SETI program

The software will need to interface to many different types of hardware, like the many types of receiver to be remotely programmed. One designer cannot code for all receivers.

Once a signal is detected, it is possible to verify the signal by switching off amplifiers, or moving the aerial slightly. The sequence of events is very dependant on the SETI station configuration and an operator may want to customise the code.

Setisearch

To meet the above requirements I have been developing a Linux SETI program called *setisearch*. This is open source and modular so that development of a module can be undertaken in isolation. Figure 3 contains a diagram of the software architecture.

From the above discussion you can see that it is important to optimise a SETI program for maximum FFT CPU time. As the display might only be viewed at start-up and then during an alarm it is a waste of CPU time to keep updating a pretty display, so a text display was chosen with a user selectable level of debug.

For debugging and showing your friends, it is sometimes useful to see a graphical representation of the received noise/signal. Ideally this will be run from a different computer, taking data over the network via a sockets interface.

Sound card

A sound card's input configuration consists of a mixer followed by an A/D converter. The A/D converter samples the input audio voltage at a rate programmed by the SETI code; the mixer level is controlled by a software AGC loop to keep the input level more or less constant.

Details of programming the soundcard and mixer, and taking data from */dev/dsp* are available in the Linux Multimedia Guide by O'Reilly. Audio samples are available in either 8 bit or 16 bit form, depending on the soundcard, with various formats for the sample such as signed or unsigned. Programming the soundcard for stereo will give alternating left-right-left-right samples.

Data collection thread

The */dev/dsp* entity will give a stream of data. The process will block whilst awaiting data from */dev/dsp*. Whilst you are away processing this data it may lose samples and the sampled audio would become discontinuous and corrupted. In order to perform an FFT you will need to take a number of samples, 4096 for instance, and then process them. Therefore one process should collect data and put it into buffers and another process should take full buffers and FFT them. Hence the need for a multi-threaded program.

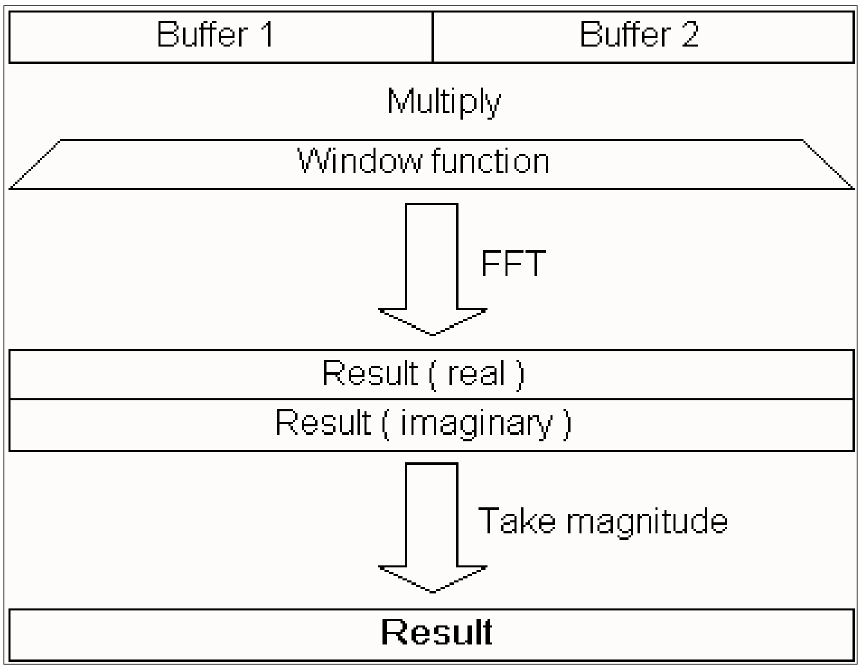


Fig. 4:How the audio samples are processed

Data processing thread

Once there are full buffers available, the processing thread will FFT the data as shown in Figure 4.

Two buffers are concatenated and then the contents multiplied by a window function. The first sample of data effectively makes a step jump from zero up to the value of sample 1. This is a false transition because of the way that we are processing blocks of data rather than continuous data, and it can cause extra noise in the results. A window function effectively creates a smooth transition rather than a step at the beginning and end of each pair of buffers, thereby reducing the noise. Unfortunately we are losing data by this process, so we process pairs of buffers, buffer 1 and 2, then 2 and 3, then 3 and 4 and so on.. All the data will then be processed.

The result of the FFT is a bit reversed complex number. The order of the bins is not linear from 0.. 4096 so this has to be re-ordered into a linear order. You could argue that for peak detection, the order is not critical. The complex number contains both magnitude and phase information of the audio stream. We calculate the magnitude and throw away the phase information.

Signal detection

Integrating many samples over a few seconds can further enhance the system sensitivity. An exponential smoothing algorithm was used as it does not require much CPU time or memory. Doppler shift is the limiting factor when integrating signals over time, as the signal may drift outside the bin limits during the period of integration.

Once the bin data is available we need to differentiate the noise from the data. A "beacon" type signal should fill one bin more

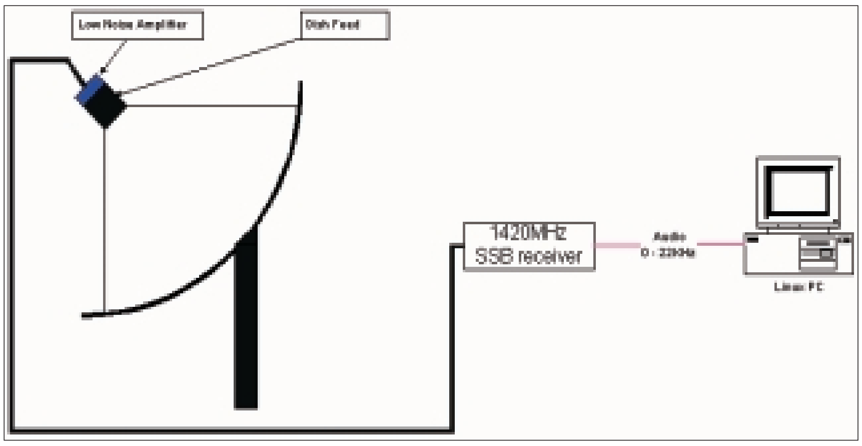


Fig. 5:A SETI station

than any other, so we are looking for bins with an above average level. The algorithm for detecting a signal in the noise should be sensitive enough to not miss valid signals, but also proof against false detection. The setting for this threshold is user-configurable via a configuration file.

On signal detection, a logfile of the 'hit' is taken whilst the signal is still present. The following data is recorded to a log file:

- Time and Date;
- Frequency of the hit;
- Position in the sky where the hit was detected;
- Bin data once a second for the duration of the hit.

The log file can be analysed using a TCL script to give a 3-D representation of the signal with Doppler shift drifting from 20KHz to DC and back.

As well as writing a logfile, hardware connected to the parallel port – such as a siren and a flashing beacon – can alert anyone in the vicinity that there is an interesting signal on the receiver. Writing data to the parallel port is relatively easy and the programming is described in the *IO Programming HOWTO*.

Typical Station

A small radio telescope suitable for Project Argus will consist of a 3 to 5 metre parabolic dish suitable for the target frequency. Many of these larger dishes are now available as people move from the old 'C' band satellite dishes to the smaller 'Sky' dishes and mini-dishes that work on 10GHz. The dish feed and low noise amolifier (LNA) units can be purchased as kits for less than £100, although it is possible for the enthusiastic amateur to build their own.

There are many 'scanner' type receivers that will work in SSB mode at 1420MHz, although the receiver bandwidth should be modified to pass 20KHz of audio. The Icom IC-R7000 is a favoured receiver and can be purchased second hand for less than £400.

What we have heard so far

Many signals have been heard so far, but the vast majority of them have been explicable as man-made interference, either due to a faulty transmitter or poor receiver design at our SETI station. The time will come when the radio bands are so full of interference that SETI stations can only operate on the dark side of the moon.

Inexplicable signals have been recorded, however. The earliest known was the 'Wow' signal as dramatised on the X-Files.

This was recorded in 1977 using the "Big Ear" Radio Telescope built by John Kraus. It shows that the signal was narrow band and that it came and went with the Gaussian response of the aerial. Unfortunately there was no follow-up Radio Telescope to confirm that signal was extra-terrestrial rather than local.

There are currently 98 Project Argus Stations in 18 countries. These stations have received many unconfirmed and unexplained signals. Co-ordination between project Argus stations will one day confirm the reception of an extraterrestrial signal. ■

Info

Setisearch website
http://www.setisearch.org/
JennyB seti website
http://www.jsquared.co.uk/seti/index.html
SetiLeague website
http://www.setileague.org/
Seti-uk website
http://www.jsquared.co.uk/seti-uk/
\Linux Multimedia Guide
(a bit dated, but the programming details are still invaluable) O'Reilly
ISBN 1-56592-219-0
Beginning Linux Programming
WROX Press
ISBN 1-861002-97-1
The Cathedral & The Bazaar
O'Reilly
ISBN 1-56592-724-9

Fig. 6:The first recorded unexplained signal from space

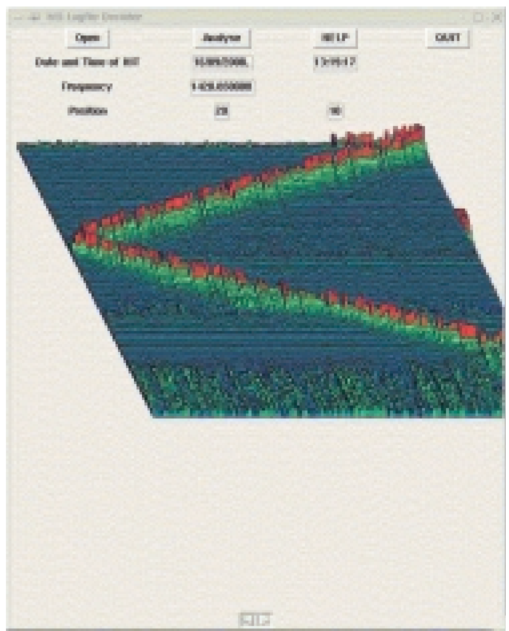


Fig. 7: A *setisearch* hit